

การผจญภัยของการเรียนรู้ของเครื่อง  
ในโลกการรู้จำรูปแบบ

The Adventures of Machine Learning  
in The World of Pattern Recognition

รัชพงศ์ กตัญญูกุล

25 มิถุนายน พ.ศ. 2564

## หน้าว่าง

## คำนำ

“การจะพัฒนาทุกสิ่งทุกอย่างให้เจริญนั้นจะต้องสร้างและเสริมขึ้นจากพื้นฐานเดิมที่มีอยู่ก่อนทั้งสิ้น ถ้าพื้นฐานไม่ดี หรือคลอนแคลนบกพร่องแล้ว ที่จะเพิ่มเติมเสริมต่อให้เจริญขึ้นไปอีกนั้น ยากนักที่จะทำได้ จึงควรจะเข้าใจให้แจ้งชัดว่า นอกจากจะมุ่งสร้างความเจริญแล้ว ยังต้องพยายามรักษาพื้นฐานให้มั่นคง ไม่บกพร่อง พร้อม ๆ กันไปด้วย”

—พระราชาดำรัสของพระบาทสมเด็จพระเจ้าอยู่หัวรัชกาลที่เก้า

หนังสือเล่มนี้ อภิปรายเนื้อหาศาสตร์การรู้จำรูปแบบด้วยวิทยาการการเรียนรู้ของเครื่อง โดยมีเนื้อหา ที่ปรับปรุง เปลี่ยนแปลง และเพิ่มเติม จากหนังสือการเรียนรู้ของเครื่องเบื้องต้นอยู่มาก. เป้าหมายหลักของหนังสือ คือ เพื่อครอบคลุมศาสตร์และศิลป์ใหม่ที่สำคัญ และ เพื่ออภิปรายศาสตร์และศิลป์ในงานการรู้จำรูปแบบ พร้อมทบทวนพื้นฐานที่สำคัญ

ถึงแม้เป้าหนึ่ง คือ การอภิปรายศาสตร์และศิลป์ในงานการรู้จำรูปแบบ แต่จากการวิเคราะห์ ทั้งในทางทฤษฎีและการประยุกต์ใช้ที่กว้างขวาง รวมถึงความก้าวหน้าที่เติบโตอย่างรวดเร็วและต่อเนื่องของศาสตร์ การจะครอบคลุมเนื้อหาทั้งหมดนั้นเป็นไปไม่ได้เลย. ดังนั้น เช่นเดียวกับหนังสือการเรียนรู้ของเครื่องเบื้องต้น หนังสือเล่มนี้จัดทำเนื้อหา โดยยึดแนวคิดในการวางแผนเพียงเป็นจุดเริ่มต้น ที่จะช่วยให้ผู้อ่านได้พอเข้าใจภาพรวม และเห็นคุณค่าของศาสตร์นี้ รวมไปจนถึงจำนวนความหลากหลายในกรณีที่ผู้อ่านต้องการแหล่งค้นคว้าเพิ่มเติม. ผู้เขียนหวังว่า ผู้อ่านจะได้เข้าใจพื้นฐานและเหตุผลเบื้องหลัง เข้าใจทฤษฎีที่สำคัญ ซาบซึ้งในความพยายาม พยายามและความคิดสร้างสรรค์เบื้องหลังการพัฒนา มองเห็นการประยุกต์ใช้ ได้ทดลองลงมือปฏิบัติ ได้ฝึกคิด ทบทวน ตั้งข้อสังเกต อภิปรายลักษณะเด่น ประเด็นสำคัญ ข้อดีข้อเสีย โอกาสและความเสี่ยง ความเกี่ยวข้อง ความเชื่อมโยงของกลไกและแนวคิดต่าง ๆ มองเห็นความท้าทาย ตลอดจนรู้สึกเพลิดเพลิน เกิดแรงบันดาลใจ ที่จะศึกษา และทำงานกับศาสตร์ด้านนี้ต่อไป

แนวคิดในการจัดทำเนื้อหานี้ ได้ออกแบบจากประสบการณ์การสอน และเพื่อใช้ประกอบการเรียน วิชาโครงข่ายภาษาไทย วิชาการเรียนรู้ของเครื่อง และวิชาการรู้จำรูปแบบ ระดับปริญญาตรีและบัณฑิตศึกษา ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น. เนื้อหาจึงถูกเรียบเรียงจากสมมติฐานว่า ผู้อ่านมีพื้นฐานคณิตศาสตร์ ซึ่งรวมถึงพีชคณิตเชิงเส้น ความน่าจะเป็น และแคลคูลัส รวมถึงมีประสบการณ์การเขียนโปรแกรมคอมพิวเตอร์มาบ้าง. อย่างไรก็ตาม ผู้เขียนหวังว่า เนื้อหานี้จะเป็นประโยชน์กับผู้อ่านที่สนใจทั่วไปเช่นกัน

## รูปแบบการเขียน

เนื่องจากเนื้อหาของศาสตร์การเรียนรู้ของเครื่องและการรู้จำรูปแบบ เกี่ยวข้องกับคณิตศาสตร์และโปรแกรมคอมพิวเตอร์ มีคำศัพท์เฉพาะจำนวนมาก อาทิ ประวัติการ แนวคิด และแนวทางปฏิบัติที่หลาย ๆ ครั้งถูกพัฒนาขึ้นจากการสร้างสรรค์และประสบการณ์ ซึ่งบางโอกาส ผู้อ่านอาจพบว่าขัดกับสัญชาตญาณ ประดิษฐ์ต่าง ๆ เหล่านี้ อาจทำให้ผู้อ่านเข้าใจข้อความที่ผู้เขียนพยายามสื่อสารได้ยาก หรือเข้าใจไม่ถูกต้อง. ดังนั้นเพื่อช่วยให้เนื้อหาอ่านง่ายขึ้น ข้อความที่ต้องการสื่อสารขัดเจนไม่คลุมเครือ ลดความกำกวມ รวมถึงการใช้ถ้อยคำไม่ฟุ่มเฟือยเยินเย้อ หรือไม่มีการใช้ย่อหน้าที่มากจนเกินไป ผู้เขียนใช้มหัพภาค เพื่อบ่งบอกการจบประโยค และใช้จุลภาค เพื่อค้นรายการต่าง ๆ รวมถึงบางครั้งอาจใช้พอนต์ตัวอังกฤษ เพื่อเน้นคำศัพท์หรือกลุ่มคำให้ชัดเจนขึ้น เช่น “วิธีที่ดีที่สุดในการเรียนรู้การรู้จำรูปแบบและการเรียนรู้ของเครื่องก็คือการลงมือทำ.” ทั้งนี้ การใช้มหัพภาคและจุลภาค แม้จะแตกต่างจากธรรมเนียมที่นิยมปฏิบัติเดิมไปบ้าง แต่เมื่อได้ผิดจากหลักการใช้ภาษาไทยที่ถูกต้องแต่อย่างใด ดังสะท้อนจากข้อความต่อไปนี้ ที่ยกจากราชบัณฑิตและสำนักงานราชบัณฑิตสถาบันฯ

- “มหัพภาค (มะ-หับ-พาก) คือ เครื่องหมาย จุด ซึ่งเขียนแสดงการจบประโยค. ในภาษาอังกฤษเรียกเครื่องหมายนี้ว่า full stop และการแสดงการจบประโยคโดยสมบูรณ์. ในภาษาไทยใช้เครื่องหมายมหัพภาคเพื่อแสดงว่าจบประโยคได้เช่นเดียวกับในภาษาอังกฤษ.”
  - เครื่องหมายมหัพภาค โดย ศ. ดร.กาญจนा นาคสกุล ราชบัณฑิต ประเภทวรรณศิลป์ สาขาวิชาภาษาไทย สำนักศิลปกรรม <http://www.royin.go.th> (คลังความรู้ เครื่องหมายมหัพภาค. สีบคัน 21 กันยายน พ.ศ. 2563)
- “จุลภาค (comma) หรือ จุดลูกน้ำ ซึ่อเครื่องหมายวรรคตอน รูปดังนี้ , มีหลักเกณฑ์การใช้ดังต่อไปนี้
  ๑. ใช้แยกวลีหรืออนุประโยคเพื่อกันความเข้าใจสับสน  
[ตัวอย่าง ...]
  ๒. ใช้คั่นคำในรายการ ที่เขียนต่อ ๆ กัน ตั้งแต่ ๓ รายการขึ้นไป โดยเขียนคั่นแต่ละรายการ ส่วนหน้าคำ “และ” หรือ “หรือ” ที่อยู่หน้ารายการสุดท้ายไม่จำเป็นต้องใส่เครื่องหมายจุลภาค  
[ตัวอย่าง ...]
  ๓. ใช้ในการเขียนบรรณานุกรม บรรณานุกรม และนามานุกรม เป็นต้น  
[ตัวอย่าง ...]
  ๔. ใช้คั่นจำนวนเลขนำจากหลักหน่วยไปทีละ ๓ หลัก”

— เครื่องหมายจุลภาค โดย สำนักงานราชบัณฑิตสภा

[http://www.royin.go.th/?page\\_id=10392](http://www.royin.go.th/?page_id=10392) (สืบค้น 21 กันยายน พ.ศ. 2563)

- “ปรัศนี (question mark) ชื่อเครื่องหมายวรรคตอน รูปดังนี้ ? มีหลักเกณฑ์การใช้ดังต่อไปนี้

๑. ใช้เมื่อสิ้นสุดความหรือประโยคที่เป็นคำนาม หรือใช้แทนคำนาม

[ตัวอย่าง ...]

๒. ใช้หลังข้อความเพื่อแสดงความสงสัยหรือไม่แน่ใจ มักเขียนอยู่ในวงเล็บ

[ตัวอย่าง ...”]

— ปรัศนี โดย สำนักงานราชบัณฑิตสภा

[http://www.royin.go.th/?page\\_id=10418](http://www.royin.go.th/?page_id=10418) (สืบค้น 28 กันยายน พ.ศ. 2563)

ภาษาไทยอาศัยการตีความในระดับวัจنبัญบัติศาสตร์อยู่มาก. การตีความในระดับวัจنبัญบัติศาสตร์อาศัยสามัญสำนึก ความเข้าใจในบริบท และความรู้เดิม. การอธิบายแนวคิดที่แตกต่างจากบริบทหรือความรู้เดิมอย่างมาก อาจทำให้ผู้อ่านไม่สามารถตีความในระดับวัจnbัญบัติศาสตร์ได้. ปัจจัยต่าง ๆ ที่ถูกอภิปราย[21] ว่าเป็นส่วนหนึ่ง ที่ทำให้ข้อความภาษาไทยหลาย ๆ ครั้งมีความก้าวกระโดดอย่างมาก ได้แก่ การเป็นภาษาคำโดยที่มีรูปแบบไวยากรณ์ที่ดูง่าย แต่ซับซ้อนด้วยการใช้คำวิเศษณ์โดยไม่มีการผันวิภาคปัจจัย (นั่นคือ การสร้างคำที่มีความหมายซับซ้อนจากการประกอบคำบรรยายต่าง ๆ เข้าด้วยกัน ที่อาจทำให้เกิดความสับสนระหว่างนามวลีและประโยค. ทั้งนามวลีและประโยคอาจอยู่ในรูป นาม-กริยา-นามได้ เช่น คนภาคถนน อาจเป็นนามวลี หมายถึง บุคคลผู้ทำหน้าที่ทำความสะอาด หรืออาจเป็นประโยคก็ได้), การไม่มีข้อบ่งบอกขอบเขตของคำที่แน่นอน และการไม่มีข้อบ่งบอกขอบเขตของประโยคที่ชัดเจน เป็นต้น. การใช้มหัพภาค เพื่อบอกการจบประโยคจะช่วยแก้ปัญหาของเขตของประโยค. ในขณะที่ การใช้ฟอนต์ตัวเอียงเพื่อเน้นคำศัพท์หรือกลุ่มคำ จะช่วยบรรเทาปัญหาของเขตของคำลง.

``Podrán cortar todas las flores,  
pero no podrán detener la primavera."

(in spanish) ---Pablo Neruda

“คุณอาจจะตัดดอกไม้ออกได้หมดทุกดอก  
แต่คุณหยุดฤดูใบไม้ผลิที่กำลังมาไม่ได้หรอก.”

—พาโบล เนรูดา

## รูปแบบอักษร

เนื่องจากเนื้อหาเกี่ยวกับอัตราส่วนของความยาวที่ต้องคำนึงถึงค่าพิเศษ เช่น ตรีโกณมิติ ความเร็ว แรง ฯลฯ ทำให้ต้องใช้สัญลักษณ์ทางคณิตศาสตร์ อย่างเช่น  $\sin$ ,  $\cos$ ,  $\tan$ ,  $\pi$ ,  $e$  เป็นต้น ซึ่งในภาษา Python ได้มีการนำสัญลักษณ์เหล่านี้มาใช้งานแล้ว เช่น `math.sin(x)`, `math.cos(x)`, `math.tan(x)` ฯลฯ สำหรับการคำนวณทางคณิตศาสตร์ แต่ในภาษา Python ไม่สามารถเขียนสัญลักษณ์ทางคณิตศาสตร์โดยตรง ต้องใช้ตัวอักษรแทน เช่น `math.pi` แทน  $\pi$ , `math.e` แทน  $e$  ฯลฯ

ตัวอักษรภาษาอังกฤษที่ใช้ในการคำนวณทางคณิตศาสตร์ เช่น  $\sin$ ,  $\cos$ ,  $\tan$ ,  $\pi$ ,  $e$  เป็นต้น ต้องมีความหมายที่เฉพาะเจาะจง เช่น  $\sin$  หมายความว่า "sinus" ในภาษาไทย แต่ในภาษาอังกฤษ หมายความว่า "sine" ซึ่งเป็นคำศัพด์ทางคณิตศาสตร์ที่มีความหมายเฉพาะเจาะจง ตัวอักษรภาษาไทย เช่น  $\pi$ ,  $e$ ,  $\phi$  เป็นต้น ไม่มีความหมายที่เฉพาะเจาะจง เช่น  $\pi$  หมายความว่า "pi" แต่ในภาษาไทย หมายความว่า "pi" ซึ่งเป็นตัวอักษรภาษาไทยที่ใช้แทนค่า pi

รูปแบบที่ 1 พังก์ชันเรเดียลเบซิส (radial basis function) สัญกรณ์  $rbf(\mathbf{x}, \mathbf{x}_c, \gamma)$

```
import numpy as np
def rbf(x, xc, gamma):
    return np.exp(-gamma*np.linalg.norm(x - xc)**2)
```

หรือรูปแบบที่ 2 (รูปแบบนี้ บางครั้งอาจแสดงเลขบรรทัดออกมากด้วย)

---

```
import numpy as np
def rbf(x, xc, gamma):
    return np.exp(-gamma*np.linalg.norm(x - xc)**2)

# gamma is a scalar:  $\gamma \in \mathbb{R}$ , but x is a vector:  $x \in \mathbb{R}^D$ 
```

นอกจากนี้ เพื่อช่วยอธิบายโปรแกรมคอมพิวเตอร์ คำอธิบายภายในส่วนข้อคิดเห็น (comment หรือ code comment) ซึ่งมีไวยากรณ์ที่เด่นชัดตามภาษาโปรแกรมคอมพิวเตอร์ รวมถึงอาจได้จัดทำให้เห็นชัดเจนขึ้น (เช่นข้ออภิบายสีที่แตกต่าง) อาจจะใช้รูปแบบอักษรอื่นตามความเหมาะสม เพื่อให้การอธิบายทำได้ชัดเจนขึ้น ดังตัวอย่างข้างต้น

รูปแบบการเขียนตัวเลข อาจขึ้นในรูปแบบปกติ เช่น พ.ศ. 2563 หรือในรูปคณิตศาสตร์ เช่น จำนวนพิกเซล  $h = 1200$  พิกเซล หรือในรูปรหัสโปรแกรม เช่น `width = 2400` ตามความเหมาะสม ทั้งนี้ การใช้รูปแบบที่แตกต่างกันนี้จะช่วยแยกแยะความหมายได้ชัดเจนขึ้น และช่วยลดความสับสนคลุมเครื่องลงได้มาก

``Form follows function."

---Louis Sullivan

“รูปลักษณะมาทีหลังประโภชน์หน้าที”

—หลุยส์ ซัลลิแวน

## สัญลักษณ์

ศาสตร์การรู้ จำรูปแบบและการเรียนรู้ของเครื่องอาศัยพื้นฐานทางคณิตศาสตร์ ดังนั้นเพื่อให้ลดความสับสน ของตัวแปรคณิตศาสตร์ที่ใช้ สัญลักษณ์ของตัวแปรคณิตศาสตร์จะใช้ตามแนวทางดังตารางข้างล่าง ยกเว้นแต่ จะระบุเป็นอื่น

ชนิดตัวแปร	แบบอักษร	ตัวอย่างอักษรلاتิน	ตัวอย่างอักษรกรีก
สเกลาร์	พิมพ์ธรรมดา	$x y z X Y Z$	$\theta \phi \omega \Theta \Phi \Omega$
เวกเตอร์	พิมพ์เล็กหนา	$\mathbf{x} \mathbf{y} \mathbf{z}$	$\boldsymbol{\theta} \boldsymbol{\phi} \boldsymbol{\omega}$
เมทริกซ์ หรือ เทนเซอร์	พิมพ์ใหญ่หนา	$\mathbf{X} \mathbf{Y} \mathbf{Z}$	$\boldsymbol{\Theta} \boldsymbol{\Phi} \boldsymbol{\Omega}$

ตัวอย่าง สเกลาร์  $x \in \mathbb{R}$  เช่น  $x = 32.4$

$$\text{เวกเตอร์ } \mathbf{x} \in \mathbb{R}^4 \text{ เช่น } \mathbf{x} = \begin{bmatrix} 10.0 & 0.75 & -44.6 & 1203.8 \end{bmatrix}^T$$

หมายเหตุ ถ้าไม่ระบุเป็นอย่างอื่น เวกเตอร์จะหมายถึงเวกเตอร์แนวตั้ง

เมทริกซ์  $\mathbf{X} \in \mathbb{R}^{2 \times 3}$  เช่น

$$\mathbf{X} = \begin{bmatrix} 1.2 & 3.5 & -0.48 \\ 0.63 & 0.0 & 123.0 \end{bmatrix}$$

และ เทนเซอร์  $\mathbf{X} \in \mathbb{R}^{2 \times 2 \times 3}$  เช่น

$$\mathbf{X} = \left[ \begin{bmatrix} 1.3 & 4.2 \\ 5 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 6.2 \\ 1.8 & 0.3 \end{bmatrix}, \begin{bmatrix} 4.5 & -3.3 \\ 2.9 & 1.7 \end{bmatrix} \right]$$

หมายเหตุ สัญลักษณ์ที่ปรากฏในภาพประกอบ อาจไม่ได้ใช้รูปแบบอักษรดังนี้ ทั้งนี้คำบรรยายภาพ จะช่วยให้ความกระจ่างแทน

``Four basic premises of writing: clarity, brevity, simplicity, and humanity.''

---William Zinsser

“ข้อตั้งสี่ประการของการเขียน คือ ความชัดเจน ความกระชับ ความเรียบง่าย และความเป็นมนุษย์”

—วิลเลียม ซินเซอร์

# สารบัญ

## สารบัญ

### สารบัญรูป

### สารบัญตาราง

i การเรียนรู้ของเครื่อง	1
1 บทนำ	3
1.1 รูปแบบ . . . . .	3
1.2 การรู้จำรูปแบบและการเรียนรู้ของเครื่อง . . . . .	4
1.3 การรู้จำตัวเลขลายมือ . . . . .	9
1.4 ประเภทของการเรียนรู้ของเครื่อง . . . . .	13
1.5 การเรียนรู้ของเครื่องและศาสตร์ที่เกี่ยวข้อง . . . . .	15
1.6 อภิธานศัพท์ . . . . .	17
1.7 แบบฝึกหัด . . . . .	19
2 พื้นฐาน	29
2.1 พีชคณิตเชิงเส้น . . . . .	29
2.2 ความน่าจะเป็น . . . . .	45
2.3 การหาค่าดีที่สุด . . . . .	62
2.4 อภิธานศัพท์ . . . . .	79
2.5 แบบฝึกหัด . . . . .	83
3 การเรียนรู้ของเครื่องและโครงข่ายประสาทเทียม	113
3.1 การปรับเส้นโค้งด้วยฟังก์ชันพุ่นนำ . . . . .	113
3.2 คุณสมบัติความทั่วไปและการเลือกแบบจำลอง . . . . .	120

3.3	โครงข่ายประสาทเทียม . . . . .	130
3.4	การประยุกต์ใช้โครงข่ายประสาทเทียม . . . . .	149
3.5	คำแนะนำสำหรับการใช้แบบจำลองทำนาย . . . . .	152
3.6	อภิธานศัพท์ . . . . .	159
3.7	แบบฝึกหัด . . . . .	162
<b>4</b>	<b>การเรียนรู้ของเครื่องในโลกกว้าง</b>	<b>223</b>
4.1	การวิเคราะห์พฤติกรรมลูกค้า . . . . .	223
4.2	ชัพพร์ตเวกเตอร์แมชชีน . . . . .	244
4.3	อภิธานศัพท์ . . . . .	260
4.4	แบบฝึกหัด . . . . .	265
<b>ii</b>	<b>การเรียนรู้เชิงลึก</b>	<b>277</b>
<b>5</b>	<b>การเรียนรู้เชิงลึก</b>	<b>279</b>
5.1	ปัญหาการเลือนหายของเกรเดียนต์ . . . . .	281
5.2	การฝึกที่ละเอียด . . . . .	284
5.3	เทคนิคการตอกออก . . . . .	288
5.4	การทำหนดค่าน้ำหนักเริ่มต้น . . . . .	293
5.5	กลไกช่วยการฝึก . . . . .	296
5.6	อภิธานศัพท์ . . . . .	305
5.7	แบบฝึกหัด . . . . .	306
<b>6</b>	<b>โครงข่ายคอนโวโลชัน</b>	<b>359</b>
6.1	ชั้นคอนโวโลชัน . . . . .	362
6.2	ชั้นดีร์รอม . . . . .	373
6.3	เกรเดียนต์ของโครงข่ายคอนโวโลชัน . . . . .	375
6.4	สรุปการคำนวณของโครงข่ายคอนโวโลชันสองมิติ . . . . .	386
6.5	โครงข่ายคอนโวโลชันที่สำคัญ . . . . .	391
6.6	อภิธานศัพท์ . . . . .	392
6.7	แบบฝึกหัด . . . . .	394
<b>7</b>	<b>การเรียนรู้เชิงลึกในโลกการรู้จำทัศนรูปแบบ</b>	<b>413</b>
7.1	การตรวจจับวัตถุในภาพ . . . . .	414
7.2	การซ้อม เสริม และก่อกำเนิดภาพ . . . . .	420

## สารบัญ

7.3 อภิธานศัพท์ . . . . .	440
7.4 แบบฝึกหัด . . . . .	443
<b>iii การรู้จำรูปแบบเชิงลำดับ</b>	<b>447</b>
<b>8 แบบจำลองสำหรับข้อมูลเชิงลำดับ</b>	<b>449</b>
8.1 ข้อมูลเชิงลำดับ . . . . .	449
8.2 แบบจำลองมาร์คอฟ . . . . .	451
8.3 แบบจำลองมาร์คอฟซ่อนเร้น . . . . .	456
8.4 อภิธานศัพท์ . . . . .	467
8.5 แบบฝึกหัด . . . . .	469
<b>9 การรู้จำรูปแบบเชิงลำดับในโลกการประมวลผลภาษาธรรมชาติ</b>	<b>473</b>
9.1 การประมวลผลภาษาธรรมชาติ . . . . .	473
9.2 โครงข่ายประสาทเวียนกลับ . . . . .	476
9.3 โครงข่ายประสาทเวียนกลับสองทาง . . . . .	485
9.4 แบบจำลองความจำระยะสั้นที่ยาว . . . . .	490
9.5 การใช้งานโครงข่ายประสาทเวียนกลับ . . . . .	492
9.6 กลไกความใส่ใจ . . . . .	495
9.7 อภิธานศัพท์ . . . . .	501
9.8 แบบฝึกหัด . . . . .	503
<b>บรรณานุกรม</b>	<b>507</b>
บรรณานุกรมภาษาไทย . . . . .	531
บรรณานุกรมภาษาอังกฤษ . . . . .	542

# สารบัญรูป

1.1	ตัวอย่างรูปตัวเลขจากลายมือเขียน . . . . .	10
1.2	แผนภาพแสดงระบบฐานรากตัวเลขลายมือ . . . . .	10
1.3	แผนภาพการค้นหาฟังก์ชันฐานรากตัวเลขลายมือ . . . . .	11
1.4	การค้นหาค่าพารามิเตอร์ของแบบจำลอง . . . . .	12
1.5	แบบฝึกหัดโน๊ตดนตรีในระดับเสียงเต็มรูป . . . . .	28
2.1	การฉายภาพ . . . . .	41
2.2	ตัวอย่างการฉายเวกเตอร์ . . . . .	42
2.3	ตัวอย่างความน่าจะเป็น ลังไส่ลูกบอล . . . . .	49
2.4	ผลจากจำลองสุ่มหยิบลูกบอล . . . . .	49
2.5	การถูเข้าของอัตราส่วนการหยิบได้สีเขียว . . . . .	49
2.6	ตัวอย่างความน่าจะเป็นแบบมีเงื่อนไข . . . . .	50
2.7	ตัวอย่างเพิ่มเติม ความน่าจะเป็นแบบมีเงื่อนไข . . . . .	56
2.8	ความเกี่ยวเนื่องของสองเหตุการณ์ . . . . .	57
2.9	ความเกี่ยวเนื่องของสามเหตุการณ์ . . . . .	57
2.10	การแจกแจงเก้าส์เชียน . . . . .	63
2.11	ความหนาแน่นความน่าจะเป็นและการแจกแจงความน่าจะเป็นสะสม . . . . .	63
2.12	ปัญหาค่ามากที่สุดกับ ปัญหาค่าน้อยที่สุด . . . . .	66
2.13	ค่าทำให้น้อยที่สุดต่าง ๆ . . . . .	67
2.14	ภาพคอนทัวร์และภาพสามทัศนวิติของฟังก์ชันจุดประสงค์ . . . . .	71
2.15	เส้นทางการหาค่าทำให้น้อยที่สุด . . . . .	73
2.16	ความก้าวหน้าในการหาค่าทำให้น้อยที่สุด . . . . .	74
2.17	ตัวอย่างปัญหาที่มีข้อจำกัด . . . . .	77
2.18	ตัวอย่างฟังก์ชันลงโทษ . . . . .	77
2.19	ตัวอย่างฟังก์ชันสูญเสียที่ถูกลงโทษ . . . . .	77
2.20	ตัวอย่างแสดงความต่างของค่าแปรปรวนและเอนໂทรปี . . . . .	84
2.21	ฟังก์ชันบวกอ่อน ฟังก์ชันซิกมอยด์ และฟังก์ชันเก้าส์เชียน . . . . .	91

## สารบัญ

2.22	ตัวอย่างภาพและผลการแก้ค่าพิเศษ . . . . .	92
2.23	ผลการจำลองปัญหามอนตี้홀 . . . . .	93
2.24	ความสัมพันธ์ของความคลาดเคลื่อนการคำนวณกับขนาดข้อมูล . . . . .	94
2.25	ผลการทำงานวิธีลงเกรเดียนต์ ที่ค่าขนาดก้าวต่าง ๆ . . . . .	99
2.26	ผลการทำงานวิธีลงเกรเดียนต์ ที่ใช้ค่าขนาดก้าวต่าง ๆ ในรอบคำนวณตัน ๆ . . . . .	100
2.27	ผลลัพธ์จากวิธีลงเกรเดียนต์ ที่ใช้ค่าขนาดก้าวต่าง ๆ . . . . .	100
2.28	ฟังก์ชันจุดประสงค์ของปัญหาหลายภาวะ . . . . .	101
2.29	ตัวอย่างแสดงผลการแก้ปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด เมื่อใช้วิธีการลงโทช . . . . .	102
2.30	ตัวอย่างปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด และเงื่อนไขครูซคุณทักษอร . . . . .	105
2.31	ตัวอย่างภาวะคู่กัน . . . . .	109
2.32	ผลการวาดกราฟฟังก์ชันไข่ชน ที่ดูต่างจากความคาดหวัง . . . . .	110
2.33	ผลจากการวาดกราฟ ซึ่งกราฟที่ได้ดูเปลกจากที่คาด . . . . .	111
3.1	ตัวอย่างจุดข้อมูล . . . . .	114
3.2	ฟังก์ชันพหุนามต่าง ๆ . . . . .	115
3.3	ความผิดพลาดของการคำนวณ . . . . .	116
3.4	พหุนามระดับขั้นหนึ่งที่ฝึกแล้ว . . . . .	118
3.5	ผลของพหุนามระดับขั้นต่าง ๆ . . . . .	121
3.6	พฤติกรรมทำงานกับธรรมชาติจริงของข้อมูล . . . . .	122
3.7	ค่าผิดพลาดชุดฝึกกับค่าผิดพลาดชุดทดสอบ . . . . .	122
3.8	แบบจำลองพหุนามเมื่อใช้จำนวนจุดข้อมูลมากขึ้น . . . . .	123
3.9	พหุนามระดับขั้นเก้า กับการทำค่าน้ำหนักเสื่อม . . . . .	126
3.10	การทำน้ำหนักเสื่อมด้วยลากกรานจ์ค่าต่าง ๆ . . . . .	126
3.11	วิธีรอสวัลลิเดชั่นห้ามบ . . . . .	127
3.12	ใช้ແນປ່ງ . . . . .	129
3.13	ເຊີລ໌ປະສາທ . . . . .	130
3.14	ເພອ້ເຊປຕຣອນ . . . . .	131
3.15	ตัวอย่างโครงข่ายເພອ້ເຊປຕຣອນສອງชั้น . . . . .	133
3.16	ฟังก์ชันจำกัดแข็งกับฟังก์ชันซิกมอยด์ . . . . .	138
3.17	ผลจากโนนดที่ส่งผ่านโนนดอืน ๆ ในชั้นคำนวณถัดไป . . . . .	141
3.18	ฟังก์ชันสูญเสีย . . . . .	148
3.19	โครงข่ายປະສາທເຖິມກับขนาดของອິນພຸຕ . . . . .	150
3.20	ฟังก์ชันซิกมอยด์ . . . . .	151
3.21	การฝึกหลาย ๆ สมัย . . . . .	153

3.22 เส้นโคงเรียนรู้ . . . . .	156
3.23 ตัวอย่างพฤติกรรมของแบบจำลองที่ความซับซ้อนต่าง ๆ . . . . .	157
3.24 ตัวอย่างเส้นโคงเรียนรู้ . . . . .	158
3.25 ผลเมื่อสุมำทำนค่าน้ำหนักเริ่มต้น . . . . .	173
3.26 ผลเมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีเหียงวิดโดร์ . . . . .	173
3.27 ผลการทดลองช้า กับวิธีกำหนดค่าน้ำหนักเริ่มต้น . . . . .	174
3.28 แผนภูมิกล่องผลการทดลองช้า . . . . .	175
3.29 ความสำคัญของการทำช้า เมื่อสุมำทำนค่าเริ่มต้น . . . . .	176
3.30 ชุดข้อมูลเรียวยอชต์ . . . . .	179
3.31 ผลทำนายชุดข้อมูลเรียวยอชต์ . . . . .	180
3.32 ชุดข้อมูลเรียวยอชต์ เปรียบเทียบการฝึกแบบหมู่ และการฝึกแบบออนไลน์ . . . . .	182
3.33 ข้อมูลชุดภาพเอ็กซเรย์เต้านม ลักษณะสำคัญมิติแรก . . . . .	189
3.34 ข้อมูลชุดภาพเอ็กซเรย์เต้านม ลักษณะสำคัญมิติที่สอง . . . . .	189
3.35 ข้อมูลชุดภาพเอ็กซเรย์เต้านม ลักษณะสำคัญต่าง ๆ ที่เป็นค่าแทนชื่อ . . . . .	190
3.36 ผลการทำนายข้อมูลภาพเอ็กซเรย์เต้านม . . . . .	191
3.37 ผลเปรียบเทียบ วิธีการจัดการกับข้อมูลขาดหายแบบต่าง ๆ . . . . .	193
3.38 การแจกแจงของข้อมูลเอมนิสต์ . . . . .	194
3.39 อินพุตของเอมนิสต์ . . . . .	195
3.40 ตัวอย่างภาพตัวเลขข้อมูลเอมนิสต์ . . . . .	195
3.41 ตัวอย่างภาพที่สับสนของเอมนิสต์ . . . . .	198
3.42 การแจกแจงของข้อมูลไม่เลกุล . . . . .	204
3.43 เอาร์พุตของแบบจำลอง สำหรับลิแกนต์และตัวหลอก . . . . .	206
3.44 ค่าคะแนนเอฟต่อระดับค่าขีดเบ่ง . . . . .	207
3.45 กราฟอาร์โซซีการทำนายการจับตัวของโนเมเลกุลขนาดเล็กกับโปรตีน . . . . .	208
3.46 แผนภูมิกล่องค่าเอาร์พุต หลังแก้ข้อมูลไม่สมดุล . . . . .	210
3.47 ค่าคะแนนเอฟ ที่ระดับค่าขีดเบ่งต่าง ๆ หลังแก้ข้อมูลไม่สมดุล . . . . .	211
3.48 กราฟอาร์โซซีการทำนายการจับตัวของโนเมเลกุล หลังปรับปรุงด้วยวิธีสุมเกิน . . . . .	211
 4.1 แผนภาพความร้อน . . . . .	224
4.2 วิธีการลบจากหลัง . . . . .	227
4.3 วิธีการตรวจจับภาพตัดตัดด้วยเทคนิคหน้าต่างเลื่อน . . . . .	228
4.4 วิธีหน้าต่างเลื่อน . . . . .	229
4.5 วิธีหน้าต่างเลื่อนกับการตรวจจับภาพเป้าหมาย . . . . .	231
4.6 เกรเดียนต์ของภาพ . . . . .	233

## สารบัญ

4.7	ตัวอย่างการทำซอฟต์แวร์ . . . . .	234
4.8	ตัวอย่างการทำซอฟต์แวร์ล็อก . . . . .	235
4.9	ตัวอย่างลักษณะสำคัญของซอฟต์แวร์ . . . . .	235
4.10	วิธีการประมาณความหนาแน่นแก่น . . . . .	239
4.11	ตัวอย่างแสดงผลลัพธ์การตรวจสอบและผลเฉลย . . . . .	241
4.12	ความเที่ยงตรงและการเรียกกลับและพื้นที่ได้กราฟ . . . . .	242
4.13	อภิรนาบ . . . . .	245
4.14	อภิรนาบและค่าพารามิเตอร์ . . . . .	246
4.15	ซ่องว่างของการแบ่งกลุ่ม . . . . .	247
4.16	ระยะทางจากอภิรนาบไปจุดข้อมูล . . . . .	249
4.17	ผลลัพธ์ของซัพพอร์ตเวกเตอร์แมชชีน . . . . .	253
4.18	ข้อมูลที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์ . . . . .	254
4.19	ผลการฝึกซัพพอร์ตเวกเตอร์แมชชีนกรณีทั่วไป . . . . .	257
4.20	พฤติกรรมของซัพพอร์ตเวกเตอร์แมชชีนที่ค่า C ต่าง ๆ . . . . .	257
4.21	ซัพพอร์ตเวกเตอร์แมชชีน ด้วยเก้าส์เซียนเครื่องanel ที่ค่า σ ต่าง ๆ . . . . .	261
4.22	ซัพพอร์ตเวกเตอร์แมชชีน ด้วยเครื่องanel เซ็นเซ็น เพื่อเปรียบเทียบกับเก้าส์เซียน . . . . .	262
4.23	ตัวอย่างการประมาณค่าความหนาแน่นความน่าจะเป็นสำหรับข้อมูลสองมิติ . . . . .	266
4.24	ตัวอย่างข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น . . . . .	269
4.25	ตัวอย่างอภิรนาบที่หาได้จากรูปปั๊ม . . . . .	269
4.26	ตัวอย่างข้อมูลที่ไม่สามารถแบ่งแยกสมบูรณ์ได้เชิงเส้น . . . . .	276
5.1	ค่าฟังก์ชันสูญเสียต่อสมัยฝึกที่ความลึกต่าง ๆ . . . . .	282
5.2	การฝึกโครงข่ายลึกล้มเหลว . . . . .	282
5.3	ปัญหาการเลือนหมายของเกรเดียนต์ . . . . .	282
5.4	การเลือนหมายของเกรเดียนต์เป็นสาเหตุของการฝึกโครงข่ายลึกล้มเหลว . . . . .	283
5.5	ฟังก์ชันกระตุนต่าง ๆ . . . . .	284
5.6	ความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบชั้น เมื่อใช้ฟังก์ชันกระตุนrelu . . . . .	285
5.7	การเลือนหมายของเกรเดียนต์บรรเทาลงด้วยการเปลี่ยนฟังก์ชันกระตุน . . . . .	285
5.8	ฟังก์ชันกระตุนreluช่วยให้การฝึกโครงข่ายลึกดีขึ้น . . . . .	285
5.9	ตัวอย่างเวลาการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ . . . . .	288
5.10	ตัวอย่างคุณภาพการทำนายเมื่อใช้ขนาดหมู่เล็กต่าง ๆ . . . . .	288
5.11	ตัวอย่างผลจากการใช้วิธีจัดหมู่เล็กแบบต่าง ๆ . . . . .	289
5.12	ภาพประกอบแสดงแนวคิดของการตกลอก . . . . .	291
5.13	ภาพแสดงพฤติกรรมซิกแซกของวิธีลงเกรเดียนต์ . . . . .	298

5.14	ภาพแสดงการทำงานของกลไกโนเมนตัม . . . . .	298
5.15	แนวทางที่นิยมดำเนินการกับค่าน้ำหนักจากการฝึกก่อน . . . . .	303
5.16	ตัวอย่างข้อมูลสำหรับแสดงปัญหาการเลือนหายของเกรเดียนต์ . . . . .	306
5.17	ขนาดของเกรเดียนต์ชั้นที่หนึ่ง ปัญหาการเลือนหายของเกรเดียนต์ . . . . .	313
5.18	ตัวอย่างจุดข้อมูล สำหรับโครงข่ายประสาทเทียมเพื่อทำการแจกแจง . . . . .	340
5.19	ผลลัพธ์การเรียนการแจกแจง . . . . .	344
5.20	ผลลัพธ์การเรียนค่าเบี่ยงเบนมาตรฐาน . . . . .	344
5.21	ตัวอย่างการแจกแจงค่าเริ่มต้นของใบอัส . . . . .	345
5.22	ผลการกระตุนระหว่างการฝึก . . . . .	348
5.23	ผลเปรียบเทียบฟังก์ชันกระตุนและวิธีกำหนดค่าเริ่มต้น . . . . .	349
5.24	การแจกแจงของค่าการกระตุน . . . . .	350
5.25	ค่าฟังก์ชันสัญเสียงต่อสมัยฝึก เมื่อไม่ใช้และใช้แบบอร์ม . . . . .	352
5.26	ค่าผิดพลาดเมื่อไม่ใช้และใช้แบบอร์ม . . . . .	353
5.27	ค่าผิดพลาดเมื่อไม่ใช้และใช้แบบอร์ม กับอัตราเรียนรู้ต่าง ๆ . . . . .	353
5.28	ค่าเฉลี่ยค่าผิดพลาด ระหว่างการฝึก เมื่อใช้และไม่ใช้แบบอร์ม ที่ขนาดหมู่เล็กต่าง ๆ . . . . .	353
5.29	แบบอร์มและขนาดของหมู่เล็ก . . . . .	354
5.30	แบบอร์มเปลี่ยนความสัมพันธ์ของการฝึกและขนาดหมู่เล็ก . . . . .	354
5.31	คุณภาพของแบบอร์มกับการเลื่อนของความแปรปรวนร่วมเกี่ยวกายนอก . . . . .	354
6.1	โครงสร้างของมิติ . . . . .	361
6.2	ขั้นการเชื่อมต่อแบบต่าง ๆ . . . . .	363
6.3	การเชื่อมต่อของขั้นตอนโวลุชั่น . . . . .	364
6.4	ฟิลเตอร์ขนาดต่าง ๆ ของขั้นตอนโวลุชั่น . . . . .	365
6.5	การเติมเต็มด้วยศูนย์ . . . . .	366
6.6	ขนาดกว้าง . . . . .	367
6.7	การคำนวณขั้นตอนโวลุชั่น . . . . .	369
6.8	การคำนวณค่าคอนโวลุชั่นในข้อมูลที่มีโครงสร้างมิติ . . . . .	371
6.9	โครงสร้างมิติ . . . . .	373
6.10	การแบ่งกระจายย้อนกลับเป็นชั้น ๆ . . . . .	378
6.11	การแบ่งกระจายย้อนกลับจากชั้นเอต์พูด . . . . .	378
6.12	การแบ่งกระจายย้อยกลับจากชั้นซ่อน . . . . .	379
6.13	การแบ่งกระจายย้อยกลับผ่านโครงสร้างการต่อเชื่อม . . . . .	381
6.14	โครงสร้างของอเล็กซ์เน็ต . . . . .	392
6.15	ตัวอย่างการจัดแทนเซอร์ไวอุปในรูปที่การคุณเมทริกซ์เสริมื่องการคำนวณแทนเซอร์ . . . . .	398

## สารบัญ

7.1	การรู้จำประเภทของวัตถุหลักในภาพ และการตรวจจับวัตถุในภาพ . . . . .	414
7.2	โอลอี้เออร์พุตที่มีโครงสร้างเทียบเท่าการแบ่งส่วนภาพอินพุต . . . . .	416
7.3	ตัวอย่างแสดงกรณีสำหรับเทคนิคล่องสมอ . . . . .	416
7.4	โอลอี้เลือกกล่องสมอเพื่อรับผิดชอบวัตถุ . . . . .	417
7.5	การขยายความละเอียดภาพด้วยชั้นดีคอนโนวูลูชั้น . . . . .	422
7.6	วิธีการฝึกแบบปรปักษ์ . . . . .	424
7.7	การฝึกโครงสร้างปรปักษ์เชิงสร้างแบบมีเงื่อนไข . . . . .	426
7.8	การทำพีซคณิตเวกเตอร์กับเวกเตอร์ค่าสุ่มของโครงสร้างปรปักษ์เชิงสร้าง . . . . .	428
7.9	โครงสร้างของใบแกนและการอนุมานที่เรียนเชิงปรปักษ์ . . . . .	429
7.10	ภาพแสดงสมมติฐานการเรียนรู้การแยกแจงข้อมูลของโครงข่ายปรปักษ์เชิงสร้าง . . . . .	431
7.11	ค่อนโนวูลูชั้นก้าวเศษช่วยขยายขนาดแผนที่ลักษณะสำคัญ . . . . .	432
7.12	การแยกແຍະໜູ້ເລີກ . . . . .	434
8.1	ตัวอย่างภาระกิจการรู้จำรูปแบบเชิงลำดับ . . . . .	452
8.2	ตัวอย่างสมมติฐานแบบต่าง ๆ ของความสัมพันธ์ระหว่างจุดข้อมูล . . . . .	454
9.1	ภาพรวมของการประมวลผลภาษาธรรมชาติ . . . . .	476
9.2	ตัวอย่างอินพุตเออร์พุตของภาระกิจการระบุหมวดคำ . . . . .	476
9.3	ตัวอย่างโครงข่ายภาษาเวียนกลับ . . . . .	477
9.4	ตัวอย่างโครงข่ายภาษาเวียนกลับ พิรุณตัวอย่างชุดข้อมูลลำดับ . . . . .	478
9.5	แผนภาพโครงสร้างโดยรวมของโครงข่ายภาษาเวียนกลับ . . . . .	478
9.6	แผนภาพคลี่ลำดับของโครงข่ายภาษาเวียนกลับ . . . . .	479
9.7	แผนภาพคลี่ลำดับของโครงข่ายภาษาเวียนกลับสองทาง . . . . .	486
9.8	แผนภาพโครงสร้างบล็อกความจำของแบบจำลองความจำระยะสั้นที่ยาว . . . . .	490
9.9	แผนภาพคลี่ลำดับของแบบจำลองความจำระยะสั้นที่ยาว . . . . .	492
9.10	แผนภาพคลี่ลำดับ กรณีที่ทั้งอินพุตและเออร์พุตเป็นชุดลำดับ . . . . .	492
9.11	แผนภาพคลี่ลำดับของโครงข่ายภาษาเวียนกลับ กรณีการจำแนกลำดับ . . . . .	493
9.12	แผนภาพคลี่ลำดับ กรณีที่อินพุตไม่ใช่ชุดลำดับ . . . . .	493
9.13	แผนภาพคลี่ลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส . . . . .	494
9.14	โครงข่ายภาษาเวียนกลับแบบลีก . . . . .	496
9.15	โครงข่ายภาษาเวียนกลับแบบลีก ที่ใช้ชั้นคำนวนไม่เวียนกลับจำนวนมาก . . . . .	496
9.16	สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส ที่ใช้รหัสเนื้อความประกอบอินพุตของตัวถอดรหัส . . . . .	500
9.17	แผนภาพแสดงโครงสร้าง เมื่อใช้กลไกความใส่ใจ . . . . .	500
9.18	แผนภาพคลี่ลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส . . . . .	503

# สารบัญตาราง

2.1	ภาษาเรื่องเขตภับเรื่องความน่าจะเป็น . . . . .	48
2.2	อัตราส่วนของการหยີບໄດ້ສີເຢີວ . . . . .	48
2.3	สรุปค่าความน่าจะเป็นร่วม . . . . .	56
2.4	คุณสมบัติที่มักสับสนของตัวแปรสุ่มต่อเนื่อง . . . . .	61
3.1	ค่าพารามิเตอร์ของแบบจำลองพหุนาม กับการทำค่าน้ำหนักเสื่อมที่ลากرانจ์ค่าต่าง ๆ . . . . .	125
3.2	ตรรกะເອັກຊ່ອວິຣີ . . . . .	132
3.3	ตัวอย่างการทำงานของເພອງເຊປຽຕຣອນ . . . . .	134
3.4	สรุปคำแนะนำสำหรับปรับปรุงแบบจำลอง . . . . .	159
3.5	ค่าสถิติของผลการຝຶກແບບจำลอง จากการทำทดลองชໍາ . . . . .	174
3.6	ตัวอย่างผลວິທີຈັດກັບຂໍ້ມູນລາດໜາຍ . . . . .	192
3.7	ເມທຣິກຊ່ອວິຣີ . . . . .	198
4.1	ค่าเฉลี่ยค่าประมาณความเที่ยงตรง . . . . .	243
6.1	ความສັນພັນຮ່ອງດ້ານນ່ວຍຄໍານວນ . . . . .	383

# สารบัญรายการ

2.1	วิธีทางทริกซ์ผกผันด้วยวิธีเก้าส์จอร์เดน . . . . .	87
2.2	วิธีลงเกรเดียนต์ . . . . .	95
2.3	วิธีลงเกรเดียนต์ที่มีเงื่อนไขการจบ . . . . .	96
2.4	วิธีลงเกรเดียนต์ เมื่อตัวแปรเป็นเวกเตอร์ . . . . .	98
3.1	แบบจำลองพหุนาม . . . . .	164
3.2	โปรแกรมฝึกพหุนามระดับขั้นหนึ่ง . . . . .	164
3.3	ตัวอย่างโปรแกรมการปรับเส้นโค้ง . . . . .	164
3.4	โครงข่ายประสาทเทียม . . . . .	166
3.5	ฟังก์ชันจำกัดแข็ง . . . . .	166
3.6	ตัวอย่างโครงข่ายประสาทสำหรับตระรากเอิกซ์ขอร์ . . . . .	166
3.7	โปรแกรมฝึกโครงข่ายประสาทเทียม . . . . .	167
3.8	โปรแกรมฟังก์ชันซิกมอยด์และอนุพันธ์ . . . . .	169
3.9	โปรแกรมฟังก์ชันเอกลักษณ์ . . . . .	170
3.10	โปรแกรมกำหนดค่าหน้าหนักเริ่มต้นด้วยการสุ่ม . . . . .	170
3.11	ค่าผิดพลาดกำลังสอง . . . . .	170
3.12	การกำหนดค่าหน้าหนักเริ่มต้นเหี้ยวนิวต์โอด์ . . . . .	172
3.13	โปรแกรมนอร์มออนไลน์อินพุต . . . . .	177
3.14	โปรแกรมฝึกโครงข่ายประสาทเทียมแบบออนไลน์ . . . . .	180
3.15	ฟังก์ชันสูญเสียครอสเอนโทรปีทวิภาค . . . . .	185
3.16	โปรแกรมแทนค่าขาดหายด้วยทุกค่าที่เป็นไปได้ . . . . .	191
3.17	ฟังก์ชันซอฟต์แมกซ์ . . . . .	196

3.18	พังก์ชันสูญเสียครอสเซ็นโทรปี . . . . .	197
3.19	โปรแกรมโหลดข้อมูลสารประกอบ . . . . .	199
3.20	ตัวอย่างโปรแกรมเลือกลักษณะสำคัญของไม้เลกุล . . . . .	201
3.21	ตัวอย่างโปรแกรมนับอะตอมและนับพันธะ . . . . .	202
3.22	โปรแกรมหาเกรเดียนต์เชิงเลข . . . . .	220
4.1	วิธีการประมาณความหนาแน่นแก่น . . . . .	265
4.2	โปรแกรมค้นหาองค์ประกอบในปัญหาปัญหามของชั้พพอร์ตเวกเตอร์แมชีน . . . . .	267
4.3	โปรแกรมหาจุดที่มีค่าคงที่ . . . . .	268
4.4	ชั้พพอร์ตเวกเตอร์แมชีน ปัญหาปัญหาม กรณีแบ่งแยกได้โดยสมบูรณ์ . . . . .	270
4.5	ชั้พพอร์ตเวกเตอร์แมชีน . . . . .	273
4.6	เครื่องเรียนรู้แบบเชิงเส้น . . . . .	276
5.1	คลาสโครงข่ายประสาทเทียม . . . . .	308
5.2	พังก์ชันกระตุ้น เขียนด้วยไฟฟอร์ช . . . . .	315
5.3	ตัวอย่างโปรแกรมรันโครงข่ายประสาทเทียมที่เขียนด้วยไฟฟอร์ช . . . . .	316
5.4	คลาสโครงข่ายประสาทเทียมแบบไฟฟอร์ช . . . . .	317
5.5	โปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ชและการหาเกรเดียนต์อัตโนมัติ . . . . .	322
5.6	โปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn . . . . .	325
5.7	ตัวอย่างคำสั่งการฝึกและทดสอบโครงข่ายประสาทเทียม ที่เขียนด้วยมอดูล nn . . . . .	325
5.8	โปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn.Module . . . . .	327
5.9	คลาส MyDataset เพื่อใช้กับ utils.data.DataLoader . . . . .	329
5.10	ตัวอย่างคำสั่งฝึกแบบจำลอง ด้วยมอดูล optim . . . . .	331
5.11	ตัวอย่างโปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn . . . . .	332
5.12	ตัวอย่างโปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn . . . . .	332
5.13	ตัวอย่างการฝึกและทดสอบโครงข่ายประสาทเทียมที่ใช้การตกลอกที่เขียนขึ้นเอง . . . . .	333
5.14	ตัวอย่างโปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn . . . . .	334
5.15	ตัวอย่างโปรแกรમชั้นสัญญาณรับทราบ . . . . .	336
5.16	โปรแกรมหาจุดที่มีค่าคงที่โดยใช้ไฟฟอร์ช nn . . . . .	340

## สารบัญตาราง

5.17	โปรแกรมคำนวณค่าลับของการพิมพ์ชั้นควรจะเป็น . . . . .	341
5.18	โปรแกรมโครงข่ายประสาทเทียม เพื่อทำนายการแจกแจง . . . . .	342
5.19	การฝึกโครงข่ายประสาทเทียม เพื่อทำนายการแจกแจง . . . . .	342
5.20	ตัวอย่างการกำหนดค่าเริ่มต้นให้โครงข่ายประสาทเทียม . . . . .	345
5.21	ฟังก์ชันกำหนดค่าน้ำหนักเริ่มต้นเซเวียร์ . . . . .	346
5.22	ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้แบบอร์ม . . . . .	348
5.23	คลาสแบบอร์ม . . . . .	351
6.1	ตัวอย่างโปรแกรมชั้นคำนวณคอนโวลูชัน . . . . .	397
6.2	ตัวอย่างการเรียกใช้ชั้นคำนวณคอนโวลูชัน . . . . .	401
6.3	ตัวอย่างการฝึกโครงข่ายคอนโวลูชัน . . . . .	402
6.4	ตัวอย่างการทดสอบโครงข่ายคอนโวลูชัน . . . . .	402
6.5	ตัวอย่างโปรแกรมชั้นเชื่อมต่อเต็มที่ . . . . .	403
6.6	ตัวอย่างโปรแกรมชั้นเชื่อมต่อเต็มที่ ที่เขียนการแพร่กระจายย้อนกลับเอง . . . . .	404
6.7	ตัวอย่างโปรแกรมชั้นคอนโวลูชัน . . . . .	405
6.8	ตัวอย่างโปรแกรมชั้นคอนโวลูชัน ที่เขียนการแพร่กระจายย้อนกลับเอง . . . . .	408
6.9	ตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด . . . . .	409
6.10	ตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด ที่เขียนการแพร่กระจายย้อนกลับเอง . . . . .	411

## ภาค i

การเรียนรู้ของเครื่อง



# บทที่ 1

## บทนำ

การประยุกต์ใช้ที่ทำให้ศาสตร์การเรียนรู้ของเครื่อง เป็นที่รู้จักอย่างกว้างขวาง คือ การรู้จำภาพ การรู้จำคำพูด การประมวลผลภาษาธรรมชาติ การประยุกต์ใช้เหล่านี้แม้แต่ต่างอย่างมาก ในเชิงสิ่งที่แสดงออก การรู้จำภาพ สัมผัสการมองเห็น การรู้จำคำพูด สัมผัสการได้ยิน การประมวลผลภาษาธรรมชาติ สัมผัสภาษา ซึ่งเป็นตัวแทนของความคิด แต่ทั้งหมดล้วนมีจุดร่วมกันที่สำคัญ คือ ทั้งหมดเป็นการรู้จำรูปแบบ.

สำหรับการเรียนรู้ การอ่านทฤษฎี เป็นวิธีที่ดีที่สุด ที่(อาจ)จะช่วยให้รู้เรื่อง แต่ไม่เข้าใจ การลงมือทำโดยไม่สนใจทฤษฎี เป็นวิธีที่เร็วที่สุด ที่(อาจ)จะช่วยให้ทำได้ แต่ไม่รู้เรื่อง การสังเกตและไตร่ตรอง เป็นวิธีที่ธรรมชาติที่สุด ที่(อาจ)จะช่วยให้เข้าใจ แต่อาจผิด. ศาสตร์การเรียนรู้ของเครื่องและการรู้จำรูปแบบ ไม่ได้ต่างจากศาสตร์อื่น ๆ เลย ในแง่ที่วิธีที่ดีที่สุดในการเรียนรู้ คือ การหาสมดุลระหว่างการศึกษาทฤษฎี การลงมือปฏิบัติ และการสังเกตและไตร่ตรอง.

เจเรมี โฮเวิร์ด (Jeremy Howard) หนึ่งในผู้เชี่ยวชาญทางด้านการเรียนรู้ของเครื่อง ได้ระบุสี่คุณสมบัติที่สำคัญสำหรับผู้ที่เหมาะสมกับงานการเรียนรู้ของเครื่อง ได้แก่ หนึ่ง ความอดีด (tenacity), สอง ความอยากรู้อยากเห็น (curiosity), สาม ความคิดสร้างสรรค์ (creativity) และสุดท้ายสี่ ทักษะ (skills) ซึ่งหมายถึงคณิตศาสตร์ และการเขียนโปรแกรมคอมพิวเตอร์. คุณสมบัติทั้งสี่เรียกตามลำดับ นั่นคือ ความอดีดสำคัญที่สุด.

อย่างไรก็ตาม บางคนอาจเริ่มด้วยคุณสมบัติที่เหมาะสมกับงาน แต่บางคนอาจใช้งานเป็นแรงจูงใจในการพัฒนาคุณสมบัติให้เกิดขึ้นในตัวเอง ดังคำกล่าวของ เจมส์ แอนโธนี ฟรูด ว่า “ความคิดฝันไม่อาจเนรมิตตัวคุณให้เป็นคนที่คุณนับถือได้ คุณต้องมุ่งมั่นบากบั้นพัฒนาตัวให้เป็นให้ได้” (James Anthony Froude: “You cannot dream yourself into a character; you must hammer and forge yourself one.”)

### 1.1 รูปแบบ

“There are only patterns, patterns on top of patterns, patterns that affect other patterns. Patterns hidden by patterns. Patterns within patterns. If you watch close, history does nothing but repeat itself. What we call chaos is just patterns we haven't recognized. What we call random is just patterns we can't decipher. What we can't understand we call nonsense. What we can't read we call gibberish.”

--Chuck Palahniuk

“รูปแบบเท่านั้น รูปแบบของรูปแบบ รูปแบบที่มีผลกับรูปแบบอื่น รูปแบบที่ซ่อนในรูปแบบ รูปแบบซ่อนในรูปแบบ ถ้าคุณดูดี ๆ ประวัติศาสตร์ไม่มีอะไร นอกจากซ้ำๆ ตัวมันเอง สิ่งที่เราเรียก ความยุ่งเหยิง ก็เป็นแค่รูปแบบที่เรายังมองไม่ออก สิ่งที่เราเรียก ไร้แบบแผน ก็เป็นแค่รูปแบบที่เรายังอ่านไม่ออก อะไรที่เราไม่เข้าใจ เราว่าไร้สาระ อะไรที่เราอ่านไม่ออก เราว่าไม่มีความหมาย。”

—ชัก ปลาหนึ่นอุค

รูปแบบมีอยู่ในทุก ๆ อย่าง ไม่ว่าจะธรรมชาติ เอกภพ ชีวิต หรือจิตปัญญา. ไม่ว่าจะประวัติศาสตร์ สมคราม การเจ้าตัวรอด กีฬา การเต้นรำ การเคลื่อนไหว การคิด ดนตรี ศิลปะ ความรู้ หรือภาษา ล้วนมีรูปแบบอยู่. รูปแบบ (pattern) หรือการซ้ำเชิงโครงสร้าง (structural repetition) ช่วยทำให้เราเข้าใจความเป็นไปต่าง ๆ ช่วยทำให้เราจดจำผู้คน อาหาร อันตราย วิธีเจ้าตัวรอด ภาษา สถานที่ สัญลักษณ์ วัตถุ แนวคิด ไปจนถึง เรื่องราวต่าง ๆ ได้.

การที่เรารู้ว่าภาพที่เห็นเป็นภาพของอะไร มีวัตถุอะไรอยู่บ้าง อยู่ที่ไหน หรือภาพบอกเล่าเหตุการณ์อะไร เป็นเพราะมีรูปแบบของภาพที่เราจำได้หรือรู้จักอยู่. การที่เราเข้าใจเสียงที่เดินินว่าเป็นเสียงของอะไร เสียงพูดของใคร กำลังพูดภาษาอะไร สำเนียงของที่ไหน พูดถึงอะไร อารมณ์เป็นอย่างไร เป็นเพราะมีรูปแบบของเสียงของภาษาที่เราจำได้อยู่. การที่เราได้ฟังหรืออ่านข้อความของภาษา แล้วเข้าใจความหมาย เป็นเพราะมีรูปแบบของภาษา ของความหมายที่เราจำได้อยู่ รวมถึงมีรูปแบบของการเชื่อมความสัมพันธ์ต่าง ๆ เข้าด้วยกัน และรูปแบบการสร้างรูปแบบใหม่ ที่เราปรับรูปแบบ ไม่ว่าจะรับรู้ในระดับจิตสำนึก หรือระดับจิตใต้สำนึก. ดังนั้นอาจกล่าวได้ว่า การรู้จำรูปแบบ เป็นความสามารถที่สำคัญที่สุดอย่างหนึ่งของสติปัญญา.

## 1.2 การรู้จำรูปแบบและการเรียนรู้ของเครื่อง

การรู้จำรูปแบบ (pattern recognition) โดยทั่วไป หมายถึง ภารกิจ เพื่อรับ�� ข้อมูลที่สำรวจมีสิ่งที่สนใจอยู่ หรือไม่ หรือ เพื่อรับ知 ข้อมูลที่สำรวจเป็นสิ่งที่สนใจประเภทใด หรือ เพื่อรับ知 สิ่งที่สนใจอยู่ที่ได้ในข้อมูลที่สำรวจ หรือ เพื่อประเมินค่าที่สนใจออกจากข้อมูลที่สำรวจ เป็นต้น.

การรู้จำรูปแบบนั้น มีอยู่ทั่วไปในธรรมชาติ เป็นความสามารถของสิ่งมีชีวิต เป็นส่วนสำคัญในพฤติกรรม

ทางสังคม แต่ในที่นี้ การรู้จำรูปแบบ จะเจาะจงเฉพาะกับ วิธีการที่จะทำให้คอมพิวเตอร์ มีความสามารถในการรู้จำรูปแบบ. การรู้จำรูปแบบด้วยคอมพิวเตอร์ อาจทำได้หลายแนวทาง. แนวทางหนึ่งคือ แนวทางการกำหนดเกณฑ์ที่ชัดเจน (criteria-based or rule-based approach) รวมไปถึง การจับคู่กับแผ่นแบบ (template matching). สำหรับบางงาน เกณฑ์แม่จะชัดเจน แต่อาจจะไม่เจาะจงที่ตัวรูปแบบเอง ตัวอย่าง เช่น การค้นหารูปแบบที่ปรากฏบ่อย ๆ ในศึกษาด้านพัฒนกรรม บางครั้งอาจต้องการหาโมโนทิฟ (motif) หรือ ลำดับของสารพัฒนกรรมสายยาว ๆ ที่พบได้บ่อยที่สุด ซึ่งเกณฑ์อาจจะระบุชัดเจน เรื่องความยาวของสาย พัฒนกรรม และเรื่องความถี่ที่ปรากฏ แต่ไม่ได้ระบุลำดับของรหัสพัฒนกรรมที่ค้นหา.

อย่างไรก็ตี รูปแบบซึ่งเป็นการซ้ำเชิงโครงสร้าง อาจเป็นผลมาจากการความสัมพันธ์สำคัญ ที่เชื่อมโยงข้อมูล กับรูปแบบนั้น. ดังนั้น การรู้จำรูปแบบ ก็เปรียบเสมือนการเรียนรู้ความสัมพันธ์สำคัญ ที่เชื่อมโยงระหว่างข้อมูล นำเข้าและรูปแบบที่สนใจนั้น. แนวทางหลักของการรู้จำรูปแบบที่จะอภิปรายในที่นี้ คือ แนวทางของการเรียนรู้ของเครื่อง. วิธีการเรียนรู้ของเครื่อง จะไม่ได้พึ่งกฎหรือเกณฑ์ที่ชัดเจน ดังแนวทางที่กล่าวไปข้างต้น แต่ใช้ ความสามารถในการทำงานกับข้อมูลมาก ๆ ของคอมพิวเตอร์ ประกอบกับแบบจำลองทางคณิตศาสตร์ เพื่อ ช่วยในการค้นหา หรือช่วยในการเรียนรู้ความสัมพันธ์ระหว่างข้อมูลนำเข้า และรูปแบบที่มักเรียกว่า ข้อมูลนำออก โดยเฉพาะสำหรับความสัมพันธ์ที่มีลักษณะซับซ้อน และยากที่จะกำหนดเป็นกฎหรือเกณฑ์ดังกล่าว.

ลักษณะเด่นของวิธีการเรียนรู้ของเครื่อง อาจปรากฏชัดอยู่ในตัวอย่างงานของอาร์瑟เรอ์ ชาเมล (Arthur Samuel) ในปี ค.ศ. 1959 ที่ เขาเขียนโปรแกรมคอมพิวเตอร์เพื่อเล่นหมากออร์ส[174] โดยที่ ตัวชาเมล เองเล่นหมากออร์สไม่เก่งเลย. หากชาเมลเขียนโปรแกรมด้วยแนวคิดดังเดิม เขายังต้องหัดเล่นหมากออร์ส ให้เก่ง และโปรแกรมวิธีเดินมากอย่างละเอียดให้กับคอมพิวเตอร์. ชาเมลไม่ได้เลือกทำแบบนั้น เขายังเลือก ที่จะโปรแกรมคอมพิวเตอร์ ให้เล่นแข่งกันเอง และให้คอมพิวเตอร์เก็บผลว่า ตำแหน่งของหมากอย่างไรที่เป็น ตำแหน่งที่ดี ซึ่งนำไปสู่ชัยชนะ หรือตำแหน่งไหนเป็นตำแหน่งไม่ดี และมักจะทำให้แพ้ แล้วให้โปรแกรมเลือก เดินหมากตามผลที่เก็บนั้น. หลังจากชาเมลปล่อยให้โปรแกรมเล่นแข่งกันเองหลายมื่นกระดาน โปรแกรม เล่นหมากออร์สของชาเมลก็สามารถเล่นหมากออร์สได้ดีมาก และเล่นได้ดีกว่าตัวของชาเมลเอง. ซึ่ง กรณีเช่นนี้ แทบจะเป็นไปไม่ได้เลยกับวิธีการเขียนโปรแกรมแบบดั้งเดิม. ดังนั้น ณ ตอนนั้น วิธีการสร้าง โปรแกรมเล่นหมากออร์สของชาเมล เป็นเหมือนการปฏิวัติแนวทางใหม่เลยที่เดียว. และนี่คือลักษณะเด่น ของการเรียนรู้ของเครื่อง คือแทนที่จะเขียนโปรแกรมวิธีทำอย่างละเอียดให้คอมพิวเตอร์ กลับเขียนโปรแกรม ให้คอมพิวเตอร์มีความสามารถในการเรียนรู้วิธีทำ สร้างสิ่งแวดล้อมให้คอมพิวเตอร์ได้เรียนรู้ แล้วปล่อยให้ คอมพิวเตอร์เรียนรู้วิธีทำเอง.

อาร์เรอร์ ชามูเอล[174] ได้นิยามการเรียนรู้ของเครื่อง (machine learning) ไว้ว่า การเรียนรู้ของเครื่อง คือ ศาสตร์ของการทำให้คอมพิวเตอร์มีความสามารถที่จะเรียนรู้ได้ โดยที่ไม่ต้องเขียนโปรแกรมวิธีการทำตรงๆ. ทอม มิทเซล[131] ได้ช่วยขยายความ โดยนิยามว่า โปรแกรมคอมพิวเตอร์จะเรียกได้ว่า มีการเรียนรู้ จากประสบการณ์  $E$  ซึ่งเกี่ยวข้องกับภารกิจ  $T$  และสมรรถนะของผลลัมภ์ที่ที่วัดได้  $P$  ก็ต่อเมื่อสมรรถนะของภารกิจ  $T$  ที่วัดด้วย  $P$  ปรับปรุงขึ้นได้จากการประสบการณ์  $E$ .

จากตัวอย่าง โปรแกรมเล่นหมากรุสของชามูเอล ประสบการณ์  $E$  คือ การได้เล่นแข่งเล่นแข่งกันเองภารกิจ  $T$  คือการเล่นหมากรุส และสมรรถนะ  $P$  วัดได้จากการที่โปรแกรมเล่นชนะ.

ปัจจุบัน การเรียนรู้ของเครื่อง ถูกประยุกต์ใช้อย่างกว้างขวาง ในวงการธุรกิจ อุตสาหกรรม การทหาร วิทยาศาสตร์ วิศวกรรม การแพทย์ การเกษตร บันเทิง ศิลปะ การกีฬา รวมถึงการประยุกต์ใช้ชีวิตประจำวัน ตัวอย่างเช่น การค้นหาข้อมูลบนเวป (web search), การกรองข้อมูลบนสื่อสังคมออนไลน์ (content filtering on social media), การตรวจสอบหารูปแบบการใช้บัตรเครดิตที่ผิดปกติ[165] ซึ่งอาจเนื่องมาจากการที่บัตรถูกขโมยไป, การบริหารการลงทุนทางการเงิน[198], งานแอพพลิเคชันที่ไม่สามารถโปรแกรมตั้งแต่ (หรือ ทำได้ยากมาก) ได้แก่ ระบบอ่านลายมือเขียน[116], การควบคุมไฮไลต์ในรีนัคบิน[42], การควบคุมหุ่นยนต์ที่มีการเครื่องไฟฟ้าที่ซับซ้อน[3], การบริหารจัดการทรัพยากรน้ำ[30], การปรับตั้งค่าของเวอร์ชัร์แมชชีน[158], การพัฒนาระบบดูแลเด็กในครadle-to-the-grave[226], การติดตามลักษณะโครงสร้างได้รับอัตโนมัติ[124], การระบุหารังสีแกรมม่าจากข้อมูลกล้องโทรทัศน์[20], ระบบตรวจสอบการสั่นสะเทือนของแผ่นดินไหว[169], การหารูปแบบในข้อมูลชีวสารสนเทศ[107], การแปลภาษาอัตโนมัติ[44], ระบบรู้จำคำพูด[176], ระบบรู้จำความก้าวหน้าของคอร์ดดนตรี[222], การประยุกต์ใช้กับงานศิลปะ[46], การประยุกต์ใช้กับกีฬา[90], ระบบรู้จำใบหน้า[10], ระบบตรวจสอบความผิดปกติของสัญญาณคลื่นไฟฟ้าหัวใจ[118], การแยกอีเมล์ที่เป็นสแปม[17], ระบบแนะนำหนังสือ เพลง วิดีโอ หรือสินค้า[73], การวิเคราะห์พฤติกรรมลูกค้า[106], การจำแนกหรือระบุหัวข้อสำหรับข้อความ[18], การเพิ่มประสิทธิภาพของงานของระบบควบคุมหรือระบบตัดสินใจที่ซับซ้อน[5, 104, 105, 102, 103, 33] ไปจนถึง การช่วยปรับปรุงคุณภาพชีวิตผู้พิการ[135].

**เก้าอี้ความรู้ รูปแบบของลูกค้าเมียและยาธิกษา** เรียบเรียงจากบางส่วนของ [29]. รูปแบบมักชื่อนความสัมพันธ์หรือกลไกที่สำคัญอยู่เบื้องหลัง. เจนเนต โรวลี่ (Janet Rowley) คุณแม่ลูกสี่ เลี้ยงลูกเป็นหลัก และทำงานเสริมกับโรงพยาบาลวิจัยมะเร็งอาร์กอน. เธอทำงานศึกษาตัวอย่างเซลล์จากคนไข้ที่ป่วยด้วยโรคเลือดต่าง ๆ แล้วในช่วงต้นปี ค.ศ. 1972 เธอกีสังเกตพบรูปแบบผิดปกติในเซลล์ของคนไข้ที่ป่วยด้วยโรคลูกค้าเมียเฉียบพลันชนิดไมลล์อยด์ คือ ดูเหมือนว่า มีบางส่วนของโครโนโซมที่แบด และบางส่วนของโครโนโซม

## 1.2. การรื้อจำรูปแบบและการเรียนรู้ของเครื่อง

7

ที่ยังสืบสืบทอดกัน (เรียกว่า การสับเปลี่ยนตำแหน่งทางพันธุกรรม หรือ translocation). โรвлีดีใจมาก และคิดว่าเรื่องอาจจบสิ้นลงแล้ว แต่ก็ยังไม่มีใครพบรูปแบบที่เด่นชัด และส่วนใหญ่ (ณ ตอนนั้น) ก็เชื่อกันว่า โครโนไซม์ที่แปลงไปเป็นผลมาจากการเรียง ไม่ใช่เป็นสาเหตุของมะเร็ง. โรвлีได้เขียนบทความรายงานเรื่องนี้ไปที่วารสารการแพทย์นิวอิงแลนด์ ซึ่งเป็นวารสารชั้นนำ แต่กลับถูกปฏิเสธ ด้วยเหตุที่วารสารชี้แจงว่า สิ่งที่โรвлีพับไม่สำคัญ. โรвлีส่งบทความนั้นไปที่วารสารเล็ก ๆ แทน.

หลังจากนั้นไม่นาน โรвлี ก็ได้ศึกษา เซลล์มะเร็งจากคนไข้ ที่ป่วยด้วยโรคลูคีเมียเรื้อรังชนิดไม่อิลอลอยด์ (Chronic Myelogenous Leukemia คำย่อ CML). แม้จะเป็นงานเสริม โรвлี ก็สนุกกับงานที่ทำมาก. เรือนำภาพถ่ายของโครโนไซม์ จากเซลล์ของคนไข้ กลับไปดูที่บ้านด้วยในวันหยุด. ภาพถ่ายของโครโนไซม์ เป็นคู่ ๆ เมื่อนอน平放 ท่องโก้ มีจุดร่วมกันตรงกลาง ๆ และมีแขนยื่นข้างบน มีขาข้างล่าง. โรвлีว่างภาพถ่ายกระจายเต็มโต๊ะอาหารที่บ้าน จนลูก ๆ ของเรอแซวว่า แม่กำลังเล่นกับตุ๊กตากระดาษอยู่. โรвлีดูภาพถ่ายเหล่านั้นอย่างละเอียด ซึ่งเป็นภาพที่ถ่ายเซลล์ที่ผ่านนิริย้อมแบบใหม่ เรอพบว่าโครโนไซม์ที่เก้าในเซลล์มะเร็งยากกว่า โครโนไซม์ที่เก้าที่พบในเซลล์ปกติ. ก่อนหน้านั้นมีนักวิจัยจากฟิลาเดลเฟียพบว่า โครโนไซม์ที่ยังสืบทอดจะสั้นผิดปกติ ในเซลล์จากผู้ป่วยลูคีเมียเรื้อรังชนิดไม่อิลอลอยด์ จนโครโนไซม์ที่ยังสืบทอดที่สั้นผิดปกติ ถูกเรียกว่า ฟิลาเดลเฟียโครโนไซม์.

สำรวจต่อ เเรอพบว่า โครโนไซม์ที่ยังสืบทอด กับโครโนไซม์ที่เก้า มีการสับเปลี่ยนส่วนกัน และชิ้นส่วนที่สับกันหายไม่เท่ากัน จึงทำให้โครโนไซม์ที่เก้ายากขึ้น และโครโนไซม์ที่ยังสืบทอดสั้นลง. โรвлีตรวจสอบเซลล์อื่น ๆ ที่ปกติของคนไข้ พบว่า เซลล์ปกติไม่ได้มีการสับเปลี่ยนส่วนกันแบบนี้ มีแบบแต่เฉพาะในเซลล์มะเร็ง. คราวนี้ โรвлีส่งบทความไปตีพิมพ์กับวารสารเนชันแนล自然 (Nature) ซึ่งเป็นวารสารวิชาการทางวิทยาศาสตร์ชั้นนำ แม้วารสารเนชันแนลจะปฏิเสธโรвлีในครั้งแรก แต่ด้วยหลักฐานที่เพิ่มขึ้น ที่สุด วารสารเนชันแนลก็ยอมรับตีพิมพ์รายงานของโรвлี. ต่อจากนั้น โรвлีก็พับการสับเปลี่ยนตำแหน่งทางพันธุกรรมระหว่างโครโนไซม์ที่สืบทอด และโครโนไซม์ที่สืบสืบทอด ในเซลล์มะเร็งจากผู้ป่วยลูคีเมียเฉียบพลันชนิดโปรไมอิโลไซติก.

โรвлีสังสัยว่า การสับเปลี่ยนตำแหน่งทางพันธุกรรมน่ามีส่วนในการก่อมะเร็ง แต่ว่า ณ ตอนนั้น มันยากที่จะพิสูจน์ประเด็นนี้ แม้ган ก่อนหน้านี้ของ เพย์ตัน รูส (Peyton Rous) ที่พับว่า ไวรัสสามารถก่อให้เกิดมะเร็งชาร์โคมาในไก่ ก็ยังไม่ได้รับการยอมรับเท่าที่ควร (ณ ตอนนั้น). อย่างไรก็ดี การค้นพบไวรัสก่อมะเร็งในสัตว์ เป็นเรื่องสำคัญ ที่จะช่วยไขปริศนาต้นกำเนิดมะเร็ง เพราะว่า ไวรัสที่รูสศึกษา ซึ่งเรียกว่า รูสชาร์โคมาไวรัส (Rous sarcoma virus คำย่อ RSV) รูสชาร์โคมาไวรัสมียืนอยู่แค่สี่ยีน ทำให้พอจะมีแนวทางค้นหา ว่ามีส่วนในการก่อมะเร็ง. สตีฟ มาρติน (Steve Martin) นักศึกษาจากมหาวิทยาลัยแคลิฟอร์เนีย ที่เบริคลี ได้แยกรูสชาร์โคมาไวรัสที่กล้ายพันธุ์กับไวรัสที่ได้ขยายพันธุ์กับไวรัสของมนุษย์ เชลล์ที่ได้กลับไม่เป็นเซลล์มะเร็ง เพราะมีการกลายพันธุ์ในไวรัสของมนุษย์ติน. ตำแหน่งที่กล้ายพันธุ์ในไวรัสของมนุษย์ติน อยู่ในยีนที่เรียกว่า ชาร์ค (src). ยีนชาร์คที่สมบูรณ์จะทำให้เกิดเซลล์มะเร็ง ชาร์คจึงถูกเรียกว่า ยีนมะเร็ง (oncogene). การค้นพบยีนมะเร็งในรูสชาร์โคมาไวรัส อาจจะช่วยอธิบายและยืนยันการค้นพบของรูส แต่ยังไม่ได้ช่วยอธิบายสาเหตุของมะเร็งในมนุษย์สักเท่าไร

การค้นพบชาร์ค ทำให้ทีมวิจัยของชาร์ล็อต 华爾默斯 (Harold Varmus) และเจ ไมเคิล บิชอฟ (J. Michael Bishop) ศึกษา และสังสัยว่า แล้วชาร์คไปอยู่ในไวรัสได้อย่างไร ในเมื่อตัวไวรัสเองไม่ได้ต้องการยึดตัวไวรัสไม่ได้ต้องการชาร์คเพื่อการยึดเซลล์ ไวรัสไม่ได้ต้องการชาร์คเพื่อการแปลงเซลล์. 华爾默สและบิชอฟคิดว่า ไวรัสน่าจะได้ชาร์โคมาโดยบังเอิญ จากเซลล์ในหนังสักเซลล์ที่มันเคยไปยึดมา ถ้าเป็นแบบนั้นจริง มันก็น่าจะมีชาร์คปราภกฏอยู่ในเซลล์ปกติตัว.

แต่ก็เกือบ ๆ สิปีหลังจากนั้น กว่าที่มีการพบร่องชาร์คในเซลล์ปกติ ชาร์คในเซลล์ปกตินี้ เรียกว่า ซีชาร์ค (c-src สำหรับ cellular src) เพื่อระบุให้ต่างจาก วีชาร์ค (v-src) ที่เป็นยีนมะเร็ง. ปราภกฏ ยีนซีชาร์คไม่ได้มีเฉพาะในไก่ แต่มีการพบในสัตว์หลาย ๆ ชนิด รวมถึงมนุษย์ด้วย. การค้นพบนี้ ทำให้ 华爾默สและบิชอฟคิดต่อไปว่า ซีชาร์คคงมีหน้าที่สำคัญอย่างในเซลล์ปกติ และตอนที่ไวรัสได้ชาร์คไป อาจจะไปเปลี่ยนแปลงบางอย่างในชาร์ค จนทำให้มันกลายเป็นยีนมะเร็ง.

ชาร์คเป็นยีนแรก แล้วหลังจากนั้นก็มีการค้นพบยีนมะเร็งจากไวรัสอื่น ๆ และเช่นเดียวกับวีชาร์ค ที่มีซีชาร์ค หลาย ๆ ยีนมะเร็ง ก็พบว่ามียีนแบบนี้ ๆ ได้ในเซลล์ปกติตัว. และพบในสัตว์หลายชนิด รวมถึงมนุษย์ด้วย. ยีนเหล่านี้ เช่น ยีนเอมวาายซี (myc) ยีโนเอบีเอล (abl) ยีนอาร์เออเอส (ras) ช่วยยืนยันว่า แนวคิดว่า ยีนมะเร็งของไวรัส มีที่มาจากยีนของเซลล์ปกติ. ยีนแบบเดียวกับยีนมะเร็ง แต่พับในเซลล์ปกติ จะเรียกว่า ยีนก่อนมะเร็ง (proto-oncogene).

วีเอบีเอล (v-abl) ในไวรัสของหนู เป็นหนึ่งในยีนมะเร็งที่ค้นพบหลังจากยีนชาร์ค และยีนก่อนมะเร็ง ที่เป็นคู่ของมัน คือ ซีเอบี

เอล (c-abl) กีพบได้ในเซลล์ปกติของหนู และกีบยังพบได้ในเซลล์ปกติของมนุษย์ด้วย. ยินซีเอบีเอล พบในโครโน่โซมที่เก้าของมนุษย์ ซึ่ง เป็นโครโน่โซมเดียวกับที่ร่วลี่พบ การสลับตำแหน่งทางพันธุกรรมในผู้ป่วยลูกคีเมียชีบพลบชนิดไม้อลอลอยด์. เรื่องนี้ทำให้มีนักวิจัย สงสัยและสืบต่อไปที่โครโน่โซมที่ยังสืบสอง จนพบว่า ในเซลล์มนุษย์เร็ง ยินซีเอบีเอลได้ย้ายจากโครโน่โซมที่เก้า ไปอยู่โครโน่โซมที่ยังสืบสอง และยังย้ายไปอยู่ตำแหน่งเดียวกันหมด ในเซลล์จากผู้ป่วยทั้งสิบเจ็ดคนที่ตรวจสอบ. ตำแหน่งที่ย้ายไป คือ ยินซีเอบีเอล ย้ายไป ต่อจากยินบีซีอาร์ (bcr) และรวมกัน (เป็น บีซีอาร์ต่อเอบีเอล หรือ bcr-abl) ซึ่งเมื่อเซลล์นำยืนไปสร้างโปรตีน จะได้โปรตีนที่ผิดปกติ โดยต่อสายโปรตีนจากบีซีเอล เข้ากับสายโปรตีนจากบีซีอาร์. ผลคือโปรตีนสายยาวพิเศษจากบีซีอาร์ต่อเอบีเอล.

หมายเหตุ ชีววิทยาจัดหลักว่า กลไกหลักของชีวิตคือ ดีเอ็นเอจะถูกถอดรหัสเป็นอาร์เอ็นเอ และอาร์เอ็นเอจะถูกแปลรหัสเพื่อ ไปสร้างโปรตีน. และโปรตีนเป็นเครื่องมือและกลไกหลักในการทำงานของชีวิต. ยิน ซึ่งเป็นลักษณะที่ถ่ายทอดทางพันธุกรรม จะถูก บันทึกไว้ด้วยดีเอ็นเอ. ถ้าเปรียบดีเอ็นเอเปรียบเหมือนโปรแกรมคอมพิวเตอร์ จะประกอบด้วยตระกูลของ โปรแกรม ไวยากรณ์ของภาษา รวมถึงข้อคิดเห็น หรือ code comments) ยินก็จะเปรียบเหมือนตระกูลของโปรแกรม.

**กลไกเบื้องหลังรูปแบบที่แสดง.** จากการศึกษาพฤติกรรมของโปรตีน โปรตีนจากบีซีเอบีเอล จะเป็นเอนไซม์ไทโรซีนคินเนส (tyrosine kinase). เอนไซม์ไทโรซีนคินเนสทำหน้าที่เพิ่มฟอสเฟตให้กับโปรตีน. การเพิ่มหรือลดฟอสเฟตกับโปรตีน เป็นส่วนของการ เปิดหรือปิดการทำงานของโปรตีน แต่เปิดหรือปิดขึ้นกับชนิดของโปรตีน. โปรตีนจากบีซีเอบีเอล (โปรตีนจากเซลล์ปกติ) จะไม่ค่อย ทำงาน ในขณะที่ โปรตีนจากบีซีอาร์ต่อเอบีเอล (โปรตีนจากเซลล์มนุษย์เร็ง) จะทำงานเกือบตลอดเวลา ทำงานเพิ่มฟอสเฟต. โปรตีน จากบีซีอาร์ต่อเอบีเอล จะเพิ่มฟอสเฟตไปให้กับ อาร์บีโปรตีน (Rb protein) ซึ่ง การเพิ่มฟอสเฟตมาก ๆ ให้กับอาร์บีโปรตีน เมื่อ ทำการปิดการทำงานของอาร์บีโปรตีน. อาร์บีโปรตีน ทำหน้าที่สำคัญในกระบวนการแบ่งตัวของเซลล์. เซลล์จะแบ่งตัวโดย การทำ สำเนาดีเอ็นเอก่อน แล้วค่อยแบ่งตัว. กระบวนการนี้จะถูกควบคุมอย่างเป็นระเบียบ. อาร์บีโปรตีน ทำหน้าที่หยุดการสำเนาดีเอ็นเอ ในเซลล์. เอนไซม์ไทโรซีนคินเนส ทำงานมากเกินไป ส่งผลเท่ากับ การปิดการทำงานของอาร์บีโปรตีน. การปิดการทำงานของอาร์บี โปรตีน ส่งผลเท่ากับ การปิดกลไกควบคุมการแบ่งตัวของเซลล์. รูปแบบที่ผิดปกติในโครโน่โซมที่ร่วลี่พบบนตัวกินข้าว เป็นสาเหตุ ของลูกคีเมีย. การสลับตำแหน่งทางพันธุกรรมทำให้เกิดยินผิดปกติ. ยินผิดปกติส่งผลให้เกิดเอมไซม์ผิดปกติ. เอนไซม์ผิดปกติส่งผลให้ เกิดการปิดกลไกหยุดการแบ่งตัวของเซลล์ และท้ายสุด ส่งผลให้เกิดโรคลูกคีเมีย.

**ความเข้าใจปัญหานำไปสู่ริบแก๊กที่ดีกว่า.** วิธีการที่การแพทย์ใช้กับมนุษย์ แต่เดิมมีอยู่สามแนวทางหลัก คือ การผ่าตัด การฉีด รังสี และการจัดยาสารพัดอย่าง เพื่อจะฆ่าเซลล์มนุษย์. ก่อนความรู้เกี่ยวกับมนุษย์เร็งข้างต้นนี้ วิธีการจัดยานี้ ตัวยาไม่ได้จะเฉพาะ กับเซลล์มนุษย์ ดังนั้นผลลัพธ์ก็ต่าง ๆ กันไป และมีผลข้างเคียงสูง.

ด้วยความเข้าใจกลไกและสาเหตุของลูกคีเมียเรื่องของนิรบัณฑุ์ นักวิทยาศาสตร์ที่บริษัทไซบากี สวิตเซอร์แลนด์ ได้แก่ นิก ไลดอน (Nick Lydon) และอเล็กซ์ มาตเตอร์ (Alex Matter) คิดว่า ถ้ายืนมาระเริงสร้างเอมไซม์ผิดปกติออกมานะ เป็นสาเหตุของโรค. เออมไซม์ผิดปกตินี้ ทำงานมากเกินไป ดังนั้น สารที่ยับยั้งเอนไซม์นี้ได้ อาจช่วยหยุดการทำงานเติบโตของมนุษย์ได. แทนที่ ไลดอนกับมาต เตอร์จะใช้วิธีค้นหาสารนี้จากรรรมชาติ หรือใช้วิธีลองผิดลองถูก ตามวิถีของการค้นหายาน ยุคหนึ่น ไลดอนกับมาตเตอร์ใช้วิธีการ ออกแบบตามเหตุผล (rational design) ใช้กระบวนการทางเคมี เพื่อสังเคราะห์โมเลกุล ที่จะเข้าไปจับตำแหน่งออกฤทธิ์ (active site) ของเอมไซม์ไทโรซีนคินเนสที่ผิดปกติ และหยุดการทำงานของไทโรซีนคินเนสที่ผิดปกติ.

หลังจากหลายปีที่ทำการทดลองทางเคมีและทดสอบ ไลดอนกับมาตเตอร์ก็ได้สารต่าง ๆ ที่ผ่านการคัดเลือกเบื้องต้น มาจำนวน หนึ่ง และทั้งคู่ได้ติดต่อกับนายแพทย์ไบรอัน ดรุกเกอร์ (Brian Druker) ซึ่งทำงานที่มหาวิทยาลัยวิทยาศาสตร์สุขภาพโอเรกอน เพื่อ ทดสอบกับเซลล์จากผู้ป่วย. ดรุกเกอร์พบว่า มีสารอยู่ตัวหนึ่งจากที่ไลดอนกับมาตเตอร์นำมา สามารถฆ่าเซลล์มนุษย์ได้ โดยไม่ฆ่า เซลล์ปกติ เมื่อใช้ที่ความเข้มข้นต่ำ ๆ

กว่าจะได้เป็นยา. ดรุกเกอร์ ไลดอน กับมาตเตอร์ ดีใจมากกับผลที่ได้ แต่ผู้บริหารของไซบากีกลับไม่ค่อยสนใจมาก ไลดอน กับมาตเตอร์ ใช้เวลาเกือบปีในการโน้มน้าวผู้บริหาร ให้อนุมัติการทดลองต่อในสัตว์. แต่ผลทดสอบพิชวิทยาในสุนัข ทำให้นักพิช วิทยาค่อนข้างเป็นห่วง เรื่องความปลอดภัยในมนุษย์. หลังจากนั้นไม่นาน บริษัทไซบากีกีความกิจการ เข้ากับบริษัทชานโดซ รวมเป็น บริษัทโนวาร์ทิส. ไลดอนลาออกจาก โนวาร์ทิสทดสอบยาอีครั้งในสัตว์ แต่นักพิชวิทยาก็ยังไม่สนับสนุน การทดลองยาในมนุษย์.

ดรุกเกอร์มองจากมนุษย์ที่ต่างไป. โอกาสครอบของคนเข้าที่ดรุกเกอร์เห็น มันรีบเริ่มมาก คนเข้าร้าว ๆ 25 ถึง 50 เปอร์เซ็นต์ ตายภายใน หนึ่งปี หลังจากพบว่าเป็นมะเร็ง และสิ่งที่ดรุกเกอร์ทำได้ ก็แค่ยื้อเวลาเท่านั้น ไม่สามารถรักษาได้เลย. ดรุกเกอร์คิดว่าพิษจากยา น่า

จะพojัดการได้โดยการติดตามผลที่ตัวคนไข้ และการปรับขนาดยา. ดรุกเกอร์ขอร้องมาตเตอร์ และมาตเตอร์กี้ยืนยันกับโนوارทิส ถึงความต้องการของยา จนในที่สุด ผู้บริหารยอมให้มีการทดสอบทางคลินิกในมนุษย์. การทดสอบเริ่มในปี ค.ศ. 1998 เกือบห้าปี หลังจากที่ดรุกเกอร์ได้ทดสอบยา กับคนไข้ลูกค้าเมียเรือรังชนิดไม่อิลอยด์จำนวนหนึ่ง โดยค่อย ๆ เพิ่มขนาดยา พร้อมกับ ติดตามอาการ ของโรคและผลข้างเคียง อย่างใกล้ชิด. ประสิทธิผลของยา วัดได้จากการลดจำนวนลงของเซลล์เม็ดเลือดขาว. ในคนปกติ เซลล์เม็ดเลือดขาวจะอยู่ที่ 4000 ถึง 6000 เซลล์ต่อเลือดหนึ่งไมโครลิตร แต่ในผู้ป่วยลูกค้าเมียเรือรังชนิดไม่อิลอยด์ เซลล์เม็ดเลือดขาวจะอยู่ที่ 100000 ถึง 500000 เซลล์ต่อเลือดหนึ่งไมโครลิตร. ที่ยาบริ�านน้อย ๆ ทีมของดรุกเกอร์ไม่เห็นผลที่แตกต่าง แต่เมื่อเพิ่มปริมาณยาขึ้น คนไข้บางคน เริ่มมีจำนวนเม็ดเลือดลดลงสูงมาก. พอนำเลือดของคนไข้ไปตรวจสอบ ก็พบว่า สัดส่วนของเซลล์ที่มีฟิลาเดลเฟียโครโนไซม์กลัดลงด้วย. ยาทำงานได้ดี.

โนوارทิสสนับสนุนยาในขั้นตอนต่อมาอย่างเต็มที่. คนไข้ในโครงการถูกติดตามผลต่อไปอีกหลายเดือน. เก้าสิบเจ็ดเปอร์เซ็นต์ ของคนไข้ที่ได้รับยาในขนาดสูงสุด มีจำนวนเม็ดเลือดขาวกลับสูงระดับปกติ ภายในเวลาสี่สัปดาห์. เมื่อตรวจสอบเซลล์จากคนไข้ ก็พบว่า คนไข้สามในสี่คน ไม่มีฟิลาเดลเฟียโครโนไซม์อีกแล้ว. ผลลัพธ์ดีเยี่ยม และดีที่สุด ในประวัติของการรักษามะเร็งด้วยการจัดยา. โนوارทิสยืนยันให้กับประเทศไทย ในชื่อ กลีเวค (Gleevec ชื่อสามัญ Imatinib) กับสำนักงานอาหารและยาของสหรัฐอเมริกา และยาได้รับการรับรองในปี ค.ศ. 2001.

กลีเวคเปลี่ยนสถานะการณ์ การรักษาลูกค้าเมียเรือรังชนิดไม่อิลอยด์หน้ามือเป็นหลังมือ. โอกาสสรอดระยะยาว (มากกว่า 8 ปี) เพิ่มขึ้นจากราว 45 เปอร์เซ็นต์ก่อนการรับรองกลีเวค ไปถึงเกือบ 90 เปอร์เซ็นต์ด้วยการใช้กลีเวค.

ปัจจุบัน เชื่อว่า ร่างกายมนุษย์ ประกอบด้วย เซลล์สองร้อยกว่าชนิด จากเซลล์ทั้งหมด จำนวนกว่าสามสิบเจ็ดล้านล้านเซลล์. จากนี่ทั้งหมดราว ๆ สองหมื่นยี่นของมนุษย์ มีรา ฯ หนึ่งร้อยสี่สิบยี่นที่มีภัณฑ์พันธุ์ และอาจก่อให้เกิดมะเร็ง. ร้อยสี่สิบยี่นเหล่านี้ เป็นส่วนหนึ่งในกระบวนการที่ควบคุมการเปลี่ยนสภาพ หรือการอยู่รอดของเซลล์. มะเร็งส่วนใหญ่จะเกี่ยวข้องกับการกลายพันธุ์ ส่องถึงแปดยี่น จากนี่ทั้งร้อยสี่สิบยี่น. การรู้ว่ามีอะไรในนั้น ก็คือว่ามีอะไรในน้ำของอุ้ง胞 หรือมะเร็งชนิดไหน จะช่วยให้เราสามารถจับแกนนิดเนื้องอก ชนิดมะเร็ง ตามเงื่อนไขทางพันธุกรรม จะช่วยให้เราเข้าใจ และสามารถเชื่อมโยงการกลายพันธุ์ กับพฤติกรรมของมะเร็ง ไปจนถึงความสามารถในการรักษาตามการกลายพันธุ์นั้น ๆ ได้. ปัจจุบัน มียกว่าสามสิบชนิด ที่รักษามะเร็งตามการกลายพันธุ์ และก็ยังมีอีกมากที่อยู่ในกระบวนการวิจัย.

เจเน็ต โรวลี่ เสียชีวิตในปี ค.ศ. 2013 จากโรคมะเร็งรังไข่. ก่อนเสียชีวิต เธอได้ทำการนัดการผ่าตัวตรวจสอบเซลล์จากที่เธอเสียชีวิต เพื่อที่นักวิจัยจะได้ศึกษาโรคต่อไป.

“[T]he most critical thing we have learned about human life at the molecular level is that everything is regulated.”

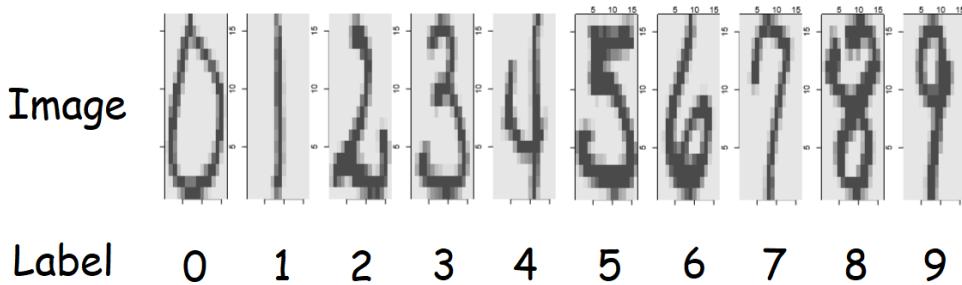
---Sean B. Carroll

“สิ่งสำคัญที่สุดที่เรารู้ได้เรียนรู้เกี่ยวกับชีวิตมนุษย์ ในระดับโมเลกุล คือ ทุก ๆ อย่างถูกควบคุมจัดระเบียบอย่างดี.”

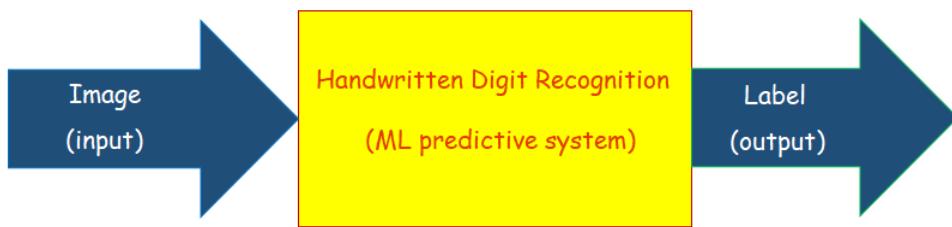
—ฉบับ ปี คาโรล

## 1.3 การรู้จำตัวเลขลายมือ

โปรแกรมรู้จำตัวเลขลายมือ<sup>[115]</sup> เป็นตัวอย่างการรู้จำรูปแบบด้วยวิธีการเรียนรู้ของเครื่อง ที่นิยมอ้างถึงกันมาก เพราะการกิจช่วยให้เข้าใจภาพรวมได้ดี และงานไม่ซับซ้อนเกินไป มีข้อมูลเข้าถึงได้ง่าย สามารถใช้เป็นตัวอย่างทดลองปฏิบัติได้. การรู้จำตัวเลขลายมือ (handwritten digit recognition) มีภารกิจ  $T$  คือ จากภาพ (ข้อมูลสำหรับ) ซึ่งคอมพิวเตอร์มองเห็นเป็นค่าความเข้มของพิกเซลต่าง ๆ แล้วให้โปรแกรมทาย ว่าภาพ



รูปที่ 1.1: ตัวอย่างรูปตัวเลขจากลายมือเขียน. แควบນแสดงตัวอย่างข้อมูลนำเข้า ซึ่งเป็นภาพ. และล่างแสดงเฉลยของแต่ละภาพ ซึ่งเป็นฉลากของแต่ละรูปแบบ



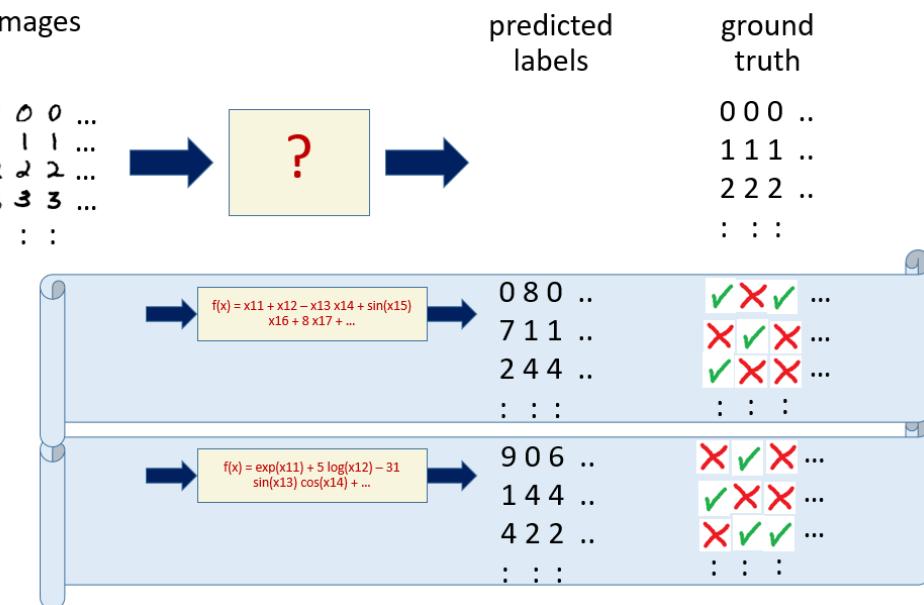
รูปที่ 1.2: แผนภาพแสดงระบบรู้จำตัวเลขลายมือ โดยมีระบบประมวลผล (ที่ทำนายฉลาก โดยสร้างตามแนวทางของการเรียนรู้ของเครื่อง) รับข้อมูลนำเข้าเป็นภาพ และให้ข้อมูลนำออก ซึ่งเป็นฉลาก.

นั้นเป็นภาพแทนตัวเลขของไร (ระบุประเภทรูปแบบ) โดยภาพของตัวเลข เป็นภาพลายมือเขียนตัวเลขต่าง ๆ จากเลข 0 ถึงเลข 9 ดังแสดงในรูปที่ 1.1. รูปตัวอย่างต่าง ๆ พร้อมเฉลย สามารถนำมาใช้ช่วยพัฒนาโปรแกรมได้ (ประสบการณ์  $E$ ). สมรรถนะ  $P$  วัดได้จากจำนวนรูปภาพที่ tally ได้ถูกต้อง.

รูปที่ 1.1 แควบ แสดงตัวอย่างรูปภาพ ที่เป็นข้อมูลนำเข้า (หรืออินพุต input) ของโปรแกรมรู้จำตัวเลขลายมือ. และล่างแสดงตัวอย่างฉลาก ของเฉลยสำหรับข้อมูลนำเข้าที่อยู่ด้านบน. ฉลาก (label) จะระบุประเภทของรูปแบบที่สนใจ ในกรณีนี้ มีสิบรูปแบบ. รูปแบบของเลขศูนย์ รูปแบบของเลขหนึ่ง รูปแบบของเลขสอง ไปจนถึง รูปแบบของเลขเก้า. เฉลย (ground truth) คือฉลากที่ถูกต้อง. เฉลย มีประโยชน์มาก โดยเฉพาะช่วยให้การวัดสมรรถนะทำได้ง่าย. สำหรับภาพใดก็ตาม หากฉลากที่ tally ตรงกับเฉลย ก็คือหมายถูก และในทางตรงกันข้าม หากไม่ตรง ก็คือหมายผิด.

ฉลากที่ tally จากโปรแกรม บางครั้งอาจเรียกในชื่อที่ทั่วไปกว่า ว่า ข้อมูลนำออก (หรือเอาต์พุต output). จากมุมมองของระบบแล้ว ภาพ คือข้อมูลนำเข้า ระบบ(โปรแกรมการรู้จำตัวเลขลายมือ) รับข้อมูลนำเข้า ประมวลผล และให้ค่าข้อมูลนำออก ซึ่งคือฉลากของรูปแบบเลขออกมานะ. รูปที่ 1.2 แสดงแผนภาพโปรแกรมการรู้จำตัวเลขลายมือจากมุมมองระบบ.

จากมุมมองนี้ ระบบประมวลผล  $f$  ทำหน้าที่แปลง ข้อมูลนำเข้า  $x$  ที่เป็นภาพ ไปเป็นข้อมูลนำออก  $y$  ที่



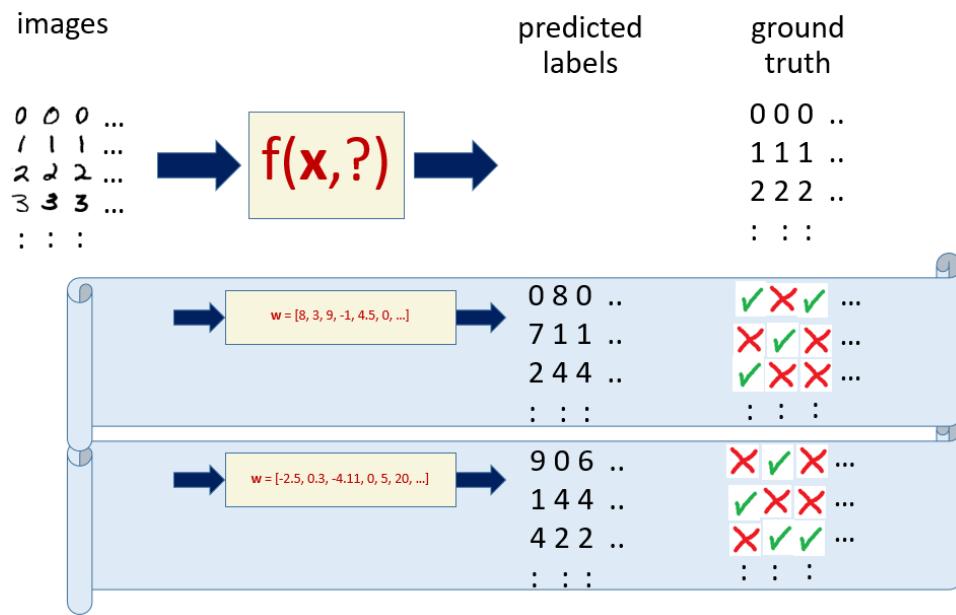
รูปที่ 1.3: แผนภาพการค้นหาฟังก์ชันรู้จำตัวเลขลายมือ. ด้วยข้อมูลตัวอย่าง โปรแกรมค้นหาฟังก์ชันคณิตศาสตร์ ที่สามารถแปลงข้อมูลนำเข้า ไปเป็น ข้อมูลนำออกที่ตรงกับเฉลยมากที่สุด.

เป็นผลลัพธ์. และ เพื่อให้เห็นภาพชัดเจน พิจารณาการรู้จำตัวเลขลายมือ ที่ออกแบบสำหรับชุดข้อมูลเอมนิสต์. ชุดข้อมูลเอมนิสต์[115] (MNIST) เป็นข้อมูลขนาดใหญ่ของภาพพร้อมเฉลยของตัวเลขลายมือเยี่ยน ข้อมูลชุดนี้นิยมใช้ สำหรับทั้งศึกษาพัฒนาระบบประมวลผลภาพ และการเรียนรู้ของเครื่อง โดยข้อมูลได้ปรับปรุงจากข้อมูลของสถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (National Institute of Standards and Technology) สหรัฐอเมริกา. ข้อมูลประกอบด้วย ภาพตัวเลขลายมือเยี่ยนจำนวน 70,000 ภาพ<sup>1</sup> พร้อมเฉลย แต่ละภาพมีขนาด  $28 \times 28$  พิกเซล และเป็นภาพขาวดำสองระดับค่า (bi-level image) นั่นคือ แต่ละพิกเซลมีค่าเป็น 0 หรือ 1. ดังนั้น หากเยี่ยน ระบบรู้จำตัวเลขลายมือเอมนิสต์นี้ เป็นฟังก์ชันคณิตศาสตร์ จะได้  $f : x \mapsto y$  โดย  $x \in \{0, 1\}^{28 \times 28}$  และ  $y \in \{0, 1, 2, \dots, 9\}$ .

จากมุมมองนี้ ปัญหาการสร้างระบบรู้จำตัวเลขลายมือ ถูกจำกัดกรอบลงมาเป็นการค้นหาฟังก์ชันคณิตศาสตร์  $f$  แทน. แนวทางการเรียนรู้ของเครื่อง ที่อาจทำได้คือ ใช้โปรแกรมค้นหาฟังก์ชัน  $f$  นี้. จากตัวอย่างข้อมูลภาพและเฉลยจำนวนมากที่มี โปรแกรมจะหาฟังก์ชันคณิตศาสตร์ที่สามารถแปลงจากภาพในตัวอย่างไปเป็นผลลัพธ์ที่ถูกต้องได้มากที่สุด. รูป 1.3 แสดงภาพตามแนวคิดนี้.

อย่างไรก็ตาม การค้นหาฟังก์ชันคณิตศาสตร์ได ๆ นั้นมีปริภูมิค้นหา (search space) ที่กว้างขวางมาก จนในทางปฏิบัติแล้ว วิธีนี้ทำงานไม่ได้เลย. วิธีแก้ปัญหาคือ แทนที่จะค้นหาฟังก์ชันคณิตศาสตร์ได ๆ แนวทางการเรียนรู้ของเครื่องที่ใช้งานได้ผล คือ จะเลือกฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์ (parametric model) ที่ พฤติ-

<sup>1</sup> 60,000 ภาพสำหรับชุดฝึกหัด และ 10,000 ภาพสำหรับทดสอบ.



รูปที่ 1.4: แผนภาพการค้นหาฟังก์ชันรู้จำตัวเลขลายมือ. ด้วยข้อมูลตัวอย่าง โปรแกรมค้นหาค่าพารามิเตอร์ของแบบจำลอง แทนการค้นหาฟังก์ชันคณิตศาสตร์.

กรรมการแปลงสามารถควบคุมได้ จากค่าของพารามิเตอร์ที่เลือกใช้ และใช้โปรแกรมค้นหาค่าของพารามิเตอร์แทน. นั่นคือ สมมติว่าฟังก์ชันคณิตศาสตร์  $f(\mathbf{x}, \mathbf{w})$  ที่พฤติกรรมการแปลงค่าข้อมูลนำเข้า  $\mathbf{x}$  ไปเป็นข้อมูลนำออก  $y$  เปลี่ยนแปลงและควบคุมได้จากค่าพารามิเตอร์ (parameter)  $\mathbf{w}$ . ดังนั้นแทนที่จะให้โปรแกรมค้นหาสมการคณิตศาสตร์ใด ๆ ที่เป็นได้ (ซึ่งมีจำนวนเกินคาดเดา) และการค้นหาไม่โอกาสสำเร็จน้อยมาก เปลี่ยนมาเป็นให้โปรแกรมค้นหาค่าของพารามิเตอร์  $\mathbf{w}$  แทน จะช่วยลดขนาดของปริภูมิค้นหาลงมหาศาล และเพิ่มโอกาสสำเร็จขึ้นมาก. นอกจากนั้น หากฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์ ที่มักเรียกว่า แบบจำลอง (model) มีความสามารถในการปรับการแปลงมาก ๆ การเลือกค่าพารามิเตอร์ ก็สามารถจะให้ผลได้ใกล้เคียงกับการค้นหาฟังก์ชันคณิตศาสตร์ใด ๆ ภายใต้ปริบทของภารกิจที่ทำงานอยู่. รูป 1.4 แสดงแนวทางของการใช้ฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์.

เนื่องจาก แบบจำลอง ทำหน้าที่แปลงจากค่าอินพุตไปหาค่าที่จะทำนายสำหรับเอาร์พุต ดังนั้น จึงอาจมองว่า แบบจำลองทำนายค่าเอาร์พุต จากค่าอินพุตได้. การรู้จำรูปแบบ ก็อาจมองจากมุมนี้ได้ว่า คือ การทำนาย (prediction ซึ่งบางครั้งเรียกว่า การอนุมาน inference) รูปแบบที่สนใจ (เอาร์พุต) จากข้อมูล (อินพุต).

รายละเอียดของแบบจำลองสำหรับการรู้จำตัวเลขลายมือเขียน และวิธีการหาค่าพารามิเตอร์ที่ทำงานได้ จะอภิปรายโดยละเอียดในบทที่ 3.

## 1.4 ประเภทของการเรียนรู้ของเครื่อง

จากตัวอย่างการรู้จำตัวเลขลายมือเขียน การวัดสมรรถนะสามารถทำได้ตรงมาตรงไป เพราะว่า รูปแบบของภาพตัวเลขลายมือเขียนมีเฉลย การประยุกต์ใช้การเรียนรู้ของเครื่อง ไม่ได้จำกัดอยู่เฉพาะกับภารกิจที่มีเฉลยเท่านั้น. แต่การที่มีเฉลย ช่วยทำให้การวัดสมรรถนะสามารถทำได้อย่างตรงมาตรงไป. ภารกิจชนิดที่มีเฉลยมาให้ด้วย จะเรียกว่า **การเรียนรู้แบบมีผู้สอน** (supervised learning). การเรียนรู้แบบมีผู้สอนเอง ก็ยังอาจจำแนกออกได้เป็นหลายประเภท ส่วนใหญ่นิยมจำแนกตามลักษณะข้อมูลนำออก. หากข้อมูลนำออกเป็นการทายฉลาก หรือทายค่าวิยุต (discrete value) ที่มีจำนวนจำกัด นั่นคือ ข้อมูลนำออก  $y \in \{\alpha_1, \alpha_2, \dots, \alpha_K\}$  เมื่อ  $K$  แทนจำนวนค่าวิยุตทั้งหมดที่เป็นไปได้ และ  $\alpha_i$  แทนค่าวิยุตต่าง ๆ ( $i = 1, \dots, K$ ) ดังเช่น การทายฉลากของตัวเลขลายมือ  $y \in \{0, 1, \dots, 9\}$  กลุ่มนี้จะเรียกว่า **ภารกิจการจำแนกกลุ่ม** (classification). แต่หากข้อมูลนำออกเป็นการทายค่าต่อเนื่อง (continuous value) นั่นคือ ข้อมูลนำออก  $y \in \mathbb{R}$  ดังเช่น การทายค่าดัชนีการเติบโตทางเศรษฐกิจ ซึ่งอาจเป็น 3.2 หรือ 4.5 หรือ ค่าจำนวนจริงใด ๆ (ซึ่งคงไม่เกิน 20 และหวังว่าจะไม่เป็น 0 หรือติดลบ) กลุ่มนี้จะเรียกว่า **ภารกิจการหาค่าคาดถอย** (regression). ค่าเฉลยสำหรับภารกิจการหาค่าคาดถอย ก็จะเป็นค่าจำนวนจริงใด ๆ (ไม่ใช่ฉลาก แบบการรู้จำตัวเลขลายมือ).

หากภารกิจที่ทำไม่มีเฉลยจริง ๆ เลย ประเภทนี้เรียกว่า **การเรียนรู้แบบไม่มีผู้ช่วยสอน** (unsupervised learning). การเรียนรู้แบบไม่มีผู้ช่วยสอน มีลักษณะที่หลากหลาย และวิธีการวัดสมรรถนะก็แตกต่างไปตามลักษณะเฉพาะ. ตัวอย่างภารกิจต่าง ๆ ที่มีลักษณะแบบนี้ ได้แก่ การจัดกลุ่มข้อมูล (clustering) ซึ่งคือ การจัดค่าข้อมูลต่าง ๆ ที่มีลักษณะคล้ายกัน ให้อยู่ในกลุ่มเดียวกัน และค่าข้อมูลต่าง ๆ ที่มีลักษณะต่างกัน ให้อยู่ต่างกลุ่มกัน, การประมาณความหนาแน่นของข้อมูล (density estimation) ซึ่งคือการเรียนรู้ค่าความน่าจะเป็นของข้อมูล, การจัดลำดับข้อมูล (ranking) ซึ่งคือ การเรียงลำดับข้อมูลตามเงื่อนไขที่ต้องการ, การสร้างแบบจำลองหัวข้อ (topic modeling) ซึ่งคือ การหาหัวข้อ (หรือตัวแทนหัวข้อ) ที่เหมาะสมกับเนื้อหาข้อมูล, การลดมิติของข้อมูล (dimension reduction) ซึ่งคือ การลดจำนวนตัวแปรหรือส่วนประกอบของแต่ละจุดข้อมูลลง เพื่อให้การประมวลผลสามารถดำเนินการได้สะดวกเร็วขึ้น, การอนุมานข้อมูล ขึ้นใหม่ (generative model) ที่สามารถใช้สร้างข้อมูลขึ้นมาใหม่ในรูปแบบเดิม หรือเปลี่ยนรูปแบบใหม่ในบริบทเดิม ซึ่งนำไปสู่การซ้อม การสร้าง หรือการตัดแปลง ภาพ เพลง ข้อความ ไปจนถึงวิดีโอต่าง ๆ ซึ่งกำลังได้รับความสนใจอย่างมากจากการศึกษา ดนตรี บันเทิง และการออกแบบ แต่ก็เป็นส่วนที่สร้างความกังวลไม่น้อยให้กับวงการสื่อสารมวลชน กฎหมาย และการพิสูจน์หลักฐาน, การเรียนรู้คุณลักษณะตัวแทน (representation learning[14]), การตรวจหารูปแบบใหม่ (novelty detection[152]) และการตรวจหารูป

แบบผิดปกติ (anomaly detection[31]) เป็นต้น. ภารกิจที่ไม่มีเฉลยนี้ครอบคลุมกว้างขวางมาก ซึ่งอาจรวมไปถึง การหาค่าดีที่สุดด้วยวิธีการค้นหาเชิงคีกษาสำนึก (optimization with heuristic search) ด้วย. การหาค่าดีที่สุดด้วยวิธีการค้นหาเชิงคีกษาสำนึกของ ก็มีการประยุกต์ใช้อย่างกว้างขวางมาก และมีรูปแบบวิธีการที่หลากหลาย อาทิ วิธีซิมูเลทเต็ดแอนนิลลิ่ง (simulated annealing[112]) และ จีเนติกอัลกอริทึม (genetic algorithm[216]) เป็นต้น.

นอกจากการเรียนรู้แบบมีผู้สอนที่มีเฉลยขัดเจน และการเรียนรู้แบบไม่มีผู้สอนที่ไม่มีเฉลยเลย ยังมีภารกิจอีกหลายประเภทที่ไม่อาจจัดอยู่ในสองกลุ่มนี้ง่ายๆ ได้ เช่น การเรียนรู้แบบกึ่งมีผู้ช่วยสอน (semi-supervised Learning) ที่เป็นลักษณะภารกิจที่มีเฉลย แต่ข้อมูลที่ได้ มีทั้งส่วนที่มีเฉลย และส่วนที่ไม่มีเฉลย และยังต้องการใช้ข้อมูลที่มีอยู่ให้คุ้มค่า โดยไม่ทิ้งข้อมูลที่ไม่มีเฉลยไปเฉย ๆ. การเรียนรู้การแนะนำลิ้นค้า (recommendation learning[166, 191]) ที่สามารถใช้ผลการประเมินความพึงใจจากตัวลูกค้าเอง กับสินค้าบางรายการ ประกอบกับ ผลประเมินจากลูกค้าคนอื่น ๆ เพื่อประเมินความชอบของลูกค้า กับสินค้ารายการที่ลูกค้าไม่ได้ประเมิน. การเรียนรู้การแนะนำลิ้นค้า อาจมีลักษณะคล้าย ๆ การเรียนรู้คุณลักษณะตัวแทน ที่พยายามเรียนรู้คุณลักษณะภายในต่าง ๆ ของสินค้าที่ลูกค้าชอบ แต่การเรียนรู้การแนะนำลิ้นค้า มีการใช้ค่าเฉลยของบากคน กับบางรายการ และไม่ได้มีค่าเฉลยของทุกคนทุกรายการ เพื่อไปทำนายความพึงใจของลูกคันทุกคน ในทุกรายการที่ไม่มีผลเฉลยได้. การเรียนรู้แบบเสริมกำลัง (reinforcement learning) ที่เป็นภารกิจการตัดสินใจ ในแต่ละคาบเวลา ซึ่งอาจจะสามารถเห็นผลระยะสั้นได้ (การเรียนรู้แบบเสริมกำลังแบบสังเกตได้สมบูรณ์ หรือ fully observable reinforcement learning) หรืออาจจะต้องประมาณผลระยะสั้นด้วย ซึ่งอาจจะประมาณผลระยะสั้นบางส่วน หรืออาจจะต้องประมาณผลระยะสั้นทั้งหมดเลย (การเรียนรู้แบบเสริมกำลังแบบสังเกตได้บางส่วน หรือ partially observable reinforcement learning) แต่เป้าหมายของการกิจจริง ๆ คือการได้ผลประโยชน์ระยะยาวที่ดี หรืออาจเป็นการหาสมดุลที่ดีระหว่างผลประโยชน์ระยะสั้น และผลประโยชน์ระยะยาว ซึ่งแม้จะมีผลลัพธ์ระยะสั้นมาให้สังเกตได้ แต่การประเมินผลประโยชน์ระยะยาวก็ไม่ได้ตรงมาตรฐานไป และไม่มีเฉลยจริง ๆ ของการตัดสินใจต่าง ๆ ให้ตรวจสอบ. การเรียนรู้แบบเสริมกำลังที่ดี จะต้องรักษาสมดุลระหว่างการเลือกการกระทำเพื่อที่จะได้ผลที่ดูเหมือนดีที่สุด กับการเลือกการกระทำเพื่อเรียนรู้ผลจากการทำต่าง ๆ ในสถานะการณ์ต่าง ๆ. ประเด็นความสมดุลนี้เรียกว่า ประเด็นของการใช้งานและการเรียนรู้ (issue of exploitation and exploration). ลักษณะเด่นชัดอีกอย่าง ก็คือการที่ระบบการเรียนรู้แบบเสริมกำลัง มีปฏิสัมพันธ์กับสิ่งแวดล้อม หรือกล่าวได้ว่า ผลของการกระทำที่ระบบเลือกมีผลต่อประสบการณ์ที่ระบบจะเรียนรู้ (ดู [104] หรือ [194] สำหรับรายละเอียดเพิ่มเติม).

## 1.5 การเรียนรู้ของเครื่องและศาสตร์ที่เกี่ยวข้อง

การเรียนรู้ของเครื่องมักถูกเชื่อมโยงกับปัญญาประดิษฐ์ (artificial intelligence หรือ คำย่อ AI) เป็นศาสตร์ที่เป้าหมายคือการสร้างคอมพิวเตอร์ที่มีเหตุมิผล เพื่อการกิจเป้าหมาย โดยคอมพิวเตอร์จะสามารถเลือกการกระทำที่ช่วยให้การกิจมีโอกาสสำเร็จมากที่สุด บนพื้นฐานของสถานการณ์ที่รับรู้ และความรู้เดิมที่มี แม้จะมีความไม่แน่นอนเกี่ยวข้องอยู่.

รัสเซลและนอร์วิก[172] ได้ยกตัวอย่างศาสตร์ต่าง ๆ ที่จัดอยู่ภายใต้ขอบเขตของปัญญาประดิษฐ์ ได้แก่ ศาสตร์การเรียนรู้ของเครื่อง ศาสตร์การแทนความรู้ (knowledge representation) ศาสตร์การประมวลผลภาษาธรรมชาติ (natural language processing) ศาสตร์คอมพิวเตอร์วิทัศน์ (computer vision) และศาสตร์วิทยาการหุ่นยนต์ (robotics) เป็นต้น.

แม้ว่าปัจจุบัน โดยเฉพาะในวงการธุรกิjmักใช้ คำว่าการเรียนรู้ของเครื่องและคำว่าปัญญาประดิษฐ์แทนกัน. อย่างไรก็ตาม ปัญญาประดิษฐ์ เน้นที่เป้าหมาย แต่ไม่ได้กำหนดวิธีการ และวิธีการหลาย ๆ อย่างของปัญญาประดิษฐ์ ไม่ได้สามารถจัดเป็นการเรียนรู้ของเครื่อง ในขณะที่การเรียนรู้ของเครื่อง มีความหมายที่เน้นถึงแนวทางวิธีการที่จะทำการกิจที่ต้องการ. และแม้ศาสตร์และศิลป์ปัจจุบันของการเรียนรู้ของเครื่อง จะได้สร้างความตื่นตัวอย่างมากกับสังคม แต่ก็ยังไม่อาจนำปัญญาประดิษฐ์ไปสู่ศักยภาพสูงสุด ซึ่งคือ การสร้างสติปัญญาระดับเดียวกับมนุษย์ ได้โดยเฉพาะ เรื่องสามัญสำนึก (common sense) เรื่องการเข้าใจภาษาธรรมชาติ (natural language understanding) เรื่องการเข้าใจความหมายระดับสูง เข้าใจสิ่งที่เป็นนามธรรม เป็นต้น.

การทำเหมืองข้อมูล (data mining) เป็นกระบวนการค้นหารูปแบบจากฐานข้อมูลขนาดใหญ่ ซึ่งมีหลายแห่งที่คล้ายกับการเรียนรู้ของเครื่อง โดยเฉพาะหลาย ๆ วิธีการของการทำเหมืองข้อมูลก็เป็นวิธีการเดียวกับวิธีการที่ใช้ในศาสตร์การเรียนรู้ของเครื่อง. ในมุมมองหนึ่ง การทำเหมืองข้อมูลจะเน้นที่ รูปแบบที่จะได้มาจากการเรียนรู้ของเครื่องจะเน้นที่วิธีการ หรือมักเรียกว่า ขั้นตอนวิธี (algorithm) มากกว่า. อย่างไรก็ตาม ในหลาย ๆ ภารกิจ มันก็อาจยากที่จะวางแผนแบ่งที่ชัดเจนได้ แต่ก็มีงานบางอย่างที่แสดงลักษณะเด่นของการทำเหมืองข้อมูล เช่น การหากว่าความสัมพันธ์ (association rules) และงานบางอย่างที่แสดงลักษณะเด่นของการเรียนรู้ของเครื่อง เช่น การเรียนรู้แบบเสริมกำลัง.

อีกประดิษฐ์หนึ่งที่อาจเป็นจุดต่างที่สำคัญ คือ ในขณะที่การเรียนรู้ของเครื่อง จะเน้นที่การค้นหารูปแบบโดยอัตโนมัติอย่างชัดเจน แต่การทำเหมืองข้อมูลนั้นอาจทำโดยอาศัยมนุษย์เป็นหลัก หรือใช้มนุษย์อยู่ในกระบวนการทำเหมืองข้อมูลอย่างมากได้. ตัวอย่างเช่น ในการหากว่าความสัมพันธ์นั้น ขั้นตอนวิธีการหากว่า

ความสัมพันธ์ อาจจะช่วยให้ความสัมพันธ์ระหว่างสินค้าต่าง ๆ ที่ซื้อด้วยกันได้ จากความถี่ที่สินค้าเหล่านั้น ปรากฏอยู่บ่อย ๆ ในรายการซื้อเดียวกัน. แต่หากจะใช้ให้ความสัมพันธ์ระหว่างคุณลักษณะของพนักงาน กับ พฤติกรรมการทำงาน อาจจะต้องอาศัยมนุษย์ช่วยกลั่นกรองความสัมพันธ์ที่ไม่เป็นสาระ (trivial association) ออก เช่น ความสัมพันธ์ที่พบว่าพนักงานที่ทำงานน้อยกว่าสามเดือนไม่เคยลาภิจ ซึ่งเหตุผลจริง ๆ เป็นเพราะว่า เขายังไม่มีสิทธิลา แต่หากสรุปผลไปผิดว่า พนักงานใหม่ขยันกว่า เพราะไม่เคยลาภิจเลย ซึ่งอาจทำให้เกิดการ เข้าใจผิดได้ หรือ ความสัมพันธ์ที่พบว่าพนักงานที่ลากคลอดทั้งหมดเป็นผู้หญิง ซึ่งแม้เป็นความจริง แต่ก็ไม่ได้มี สาระประโยชน์อะไร จึงจำเป็นต้องอาศัยมนุษย์ช่วยกลั่นกรองรูปแบบความสัมพันธ์ที่พบ.

นอกจากปัญญาประดิษฐ์ และการทำเหมืองข้อมูลแล้ว ยังมีศาสตร์อื่นอีกที่มีความหมายทับซ้อนคลุมเครือ กับการเรียนรู้ของเครื่อง. วิทยาการข้อมูล (data science) รวมศาสตร์ต่าง ๆ เพื่อวิเคราะห์ข้อมูล ทำความเข้าใจเรื่องราว และทำนายประเด็นที่สนใจ ไปจนถึงแสดงข้อมูล แสดงมุมมองและนำเสนอผลวิเคราะห์. ด้วยลักษณะของวิทยาการข้อมูล วิทยาการข้อมูลครอบคลุมเนื้อหาส่วนหนึ่งของสถิติศาสตร์, การเรียนรู้ของ เครื่อง, การทำเหมืองข้อมูล, การจัดการฐานข้อมูล, และการสร้างมโนภาพสำหรับข้อมูลและสารสนเทศ (data and information visualization) รวมถึงเทคโนโลยีต่าง ๆ ที่ใช้จัดการข้อมูลปริมาณมหาศาล เช่น แมปเรดิวซ์ (MapReduce).

หมายเหตุ การแบ่งแยกหรือจัดฉลากสำหรับศาสตร์ต่าง ๆ เหล่านี้ไม่ได้มีเส้นแบ่งที่ชัดเจน และในทาง ปฏิบัติก็ไม่ได้มีเส้นแบ่ง หรือไม่ได้มีข้อจำกัด หรือไม่ได้มีความจำเป็นใดที่ต้องแบ่งให้เด็ดขาด. ภารกิจที่ทำ เป็น สิ่งสำคัญที่สุด. นั่นหมายถึงว่า เทคนิคใด ๆ ก็ตามที่เป็นประโยชน์ ที่ใช้งานได้ ที่เหมาะสมกับงาน ถือว่าดีทั้ง นั้น ไม่ว่ามันจะเรียกหรือจัดเป็นศาสตร์ใด หรือแม้แต่ มันจะเป็นแนวทางใหม่ที่อาจยากที่จะถูกจัดให้อยู่ภายใต้ ศาสตร์ใดก็ตาม.

บางครั้ง การเรียนรู้ของเครื่อง ถูกสับสนกับการเรียนรู้เชิงลึก. การเรียนรู้เชิงลึก (deep learning) เป็น การเรียนรู้ของเครื่อง ที่เน้นการใช้แบบจำลอง ที่มีความสามารถในการแปลงข้อมูลสูง (model with high representative power) โดยใช้การประมวลผลเป็นลำดับชั้น เรียกว่า แบบจำลองเชิงลึก (deep model) หรือโครงข่ายเชิงลึก (deep network). ความสามารถของแบบจำลองเชิงลึก ได้มาจากการที่แบบจำลองมี โครงสร้างที่มีการคำนวนในลักษณะเป็นขั้น ๆ ลำดับชั้น. ผลจากขั้นหนึ่งส่งไปคำนวนต่อที่อีกขั้นหนึ่ง และ ทำการคำนวนเช่นนี้ต่อไปหลาย ๆ ขั้น (ที่มากของคำว่า ลึก). บท 5 อภิปรายการเรียนรู้เชิงลึก ในรายละเอียด.

นอกจากศาสตร์ต่าง ๆ ดังกล่าวแล้ว ประเด็นของข้อมูลหัก (big data) เป็นอีกหนึ่งเรื่องที่มักถูกสับสน กับการเรียนรู้ของเครื่อง. ข้อมูลหัก คือ ถึงชุดข้อมูลที่มีปริมาณข้อมูลขนาดใหญ่ และมีความหลากหลาย

ของชนิดข้อมูล โดยลักษณะสำคัญของข้อมูลที่เป็นข้อมูลนั้น คือ ข้อมูลมีปริมาณมาก ข้อมูลเพิ่มขึ้นอย่างรวดเร็ว และข้อมูลมีความหลากหลายมาก (ซึ่งมักถูกอ้างถึง โดยย่อว่า 3 Vs สำหรับ high volume, high velocity, และ high variety). เมื่อเปรียบเทียบข้อมูลนั้นกับการเรียนรู้ของเครื่อง กล่าวโดยง่าย คือ ในขณะที่ข้อมูลนั้นเน้นที่ลักษณะและความท้าทายของการจัดการกับข้อมูลในเชิงปริมาณ การเรียนรู้ของเครื่องเน้นที่การกิจที่จะทำ โดยมักใช้ข้อมูลประกอบ เพื่อการบรรลุภารกิจ. การทำข้อมูลนั้น อาจต้องการเพียงชาร์ดแวร์ระบบฐานข้อมูล รวมถึงโครงสร้างข้อมูลที่มีประสิทธิภาพนี้ เพื่อรองรับความท้าทายเชิงปริมาณของข้อมูล. การทำข้อมูลนั้น อาจจะใช้หรือไม่ใช้แนวทางการเรียนรู้ของเครื่องก็ได้ ขึ้นกับจุดประสงค์. การเรียนรู้ของเครื่องเอง เมื่อใช้งานกับข้อมูลที่มีลักษณะข้อมูลนั้น อาจต้องการเทคนิคและกลไกที่ช่วยจัดการความท้าทายเชิงปริมาณ และอาจต้องการขั้นตอนวิธีใหม่ ที่เหมาะสมกับปริมาณ ความเร็ว และความหลากหลายของข้อมูลนั้น. หากเปรียบเทียบข้อมูลนั้นเป็นถนนรุกรังที่ยาวมาก ๆ การเรียนรู้ของเครื่องก็อาจเปรียบเป็นรถยนต์บนถนน. บางครั้งก็ทำงานด้วยกัน บางครั้งก็ไม่. แต่เมื่อทำงานด้วยกัน หากเปลี่ยนเป็นยางสำหรับถนนรุกรังเปลี่ยนช่วงล่างให้ทนทานขึ้น และเตรียมน้ำมันเชื้อเพลิงเพื่อไว้ให้เพียงพอ อาจจะช่วยให้ขับผ่านไปสู่เป้าหมายได้โดยสวัสดิภาพ.

เนื้อหาของตำราเล่มนี้ เน้นพื้นฐาน และศาสตร์และศิลป์ที่สำคัญ ของการเรียนรู้ของเครื่อง ซึ่งแม้หลาย ๆ เรื่อง จะเป็นเนื้อหาของศาสตร์อื่น ๆ เช่นกัน แต่ตำนานี้ไม่ได้มีจุดประสงค์เพื่อ ครอบคลุมปัญญาประดิษฐ์ การทำเหมืองข้อมูล หรือศาสตร์อื่น ๆ ที่เกี่ยวข้อง. ผู้อ่านที่สนใจศาสตร์ที่เกี่ยวข้องเหล่านี้ สามารถศึกษาได้จากตำรา และแหล่งเรียนรู้เฉพาะของแต่ละศาสตร์.

## 1.6 อภิธานศัพท์

**รูปแบบ (pattern):** การซ้ำเชิงโครงสร้าง

**การรู้จำรูปแบบ (pattern recognition):** การทายค่าหรือระบุลักษณะของรูปแบบ จากข้อมูลนำเข้า

**การเรียนรู้ของเครื่อง (machine learning):** ศาสตร์ของการทำให้คอมพิวเตอร์มีความสามารถที่จะเรียนรู้ที่จะทำนาย หรือตัดสินใจได้ โดยที่ไม่ต้องเขียนโปรแกรมวิธีการทำตรง ๆ

**การรู้จำตัวเลขลายมือ (handwritten digit recognition):** โปรแกรมทายภาพ ว่าภาพนั้นแทนตัวเลขอะไร โดยภาพของตัวเลข เป็นภาพลายมือเขียนตัวเลขต่าง ๆ จากเลข 0 ถึงเลข 9

**ข้อมูลนำเข้า หรืออินพุต (input):** ตัวแปรต้น หรือข้อมูลที่โปรแกรมรับเข้า

**ข้อมูลนำออก หรือเอาต์พุต (output):** ตัวแปรตาม หรือค่าข้อมูลที่โปรแกรมต้องให้ออกมา

**ฉลาก (label):** ตัวแปรตาม หรือข้อมูล ที่ระบุประเภท หรือชื่อของรูปแบบที่สนใจ

**เอ็ม尼สต์ (MNIST):** ข้อมูลขนาดใหญ่ของภาพพร้อมเฉลยของตัวเลขลายมือเขียน ซึ่งนิยมใช้ทดสอบระบบการรู้จำตัวเลขลายมือ

**แบบจำลอง (model):** สมการคณิตศาสตร์ที่ใช้คำนวณค่าข้อมูลนำออก จากค่าข้อมูลนำเข้า และค่าพารามิเตอร์ที่เลือกใช้ ซึ่งข้อมูลนำออกมักเป็นค่าที่นายของสิ่งที่สนใจ

**พารามิเตอร์ (parameter):** ตัวแปรที่ค่าของมัน สามารถปรับเปลี่ยนการทำนายของแบบจำลอง โดยเปลี่ยนพฤติกรรมการแปลงค่าข้อมูลนำเข้าไปเป็นข้อมูลนำออก

**การเรียนรู้แบบมีผู้สอน (supervised learning):** การกิจการทำนายหรือตัดสินใจ ที่มีเฉลยที่ถูกต้องให้

**การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning):** การกิจการทำนายหรือตัดสินใจ ที่ไม่มีเฉลยที่ถูกต้องให้

**การจำแนกกลุ่ม (classification):** การกิจการทำนายฉลาก หรือทำนายค่าวิมุตที่มีจำนวนจำกัด

**การหาค่าทดแทน (regression):** การกิจการทำนายค่าที่เป็นจำนวนจริง

## 1.7 แบบฝึกหัด

``Sail away from the safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover."

---Mark Twain

“ออกเรือไปจากอ่าวที่ปลอดภัย กางใบปีกับลมสำเภา ออกสำรวจ ออกผัน ออกค้นพบ.”

—マーク・吐温

เพื่อเป็นการทบทวนทักษะการเขียนโปรแกรม แบบฝึกหัดทบทวนการเขียนโปรแกรมทั่วไป. แม้ ตำรา จะ อภิปรายเนื้อหา ศาสตร์การรู้ จำกัดแบบและการเรียนรู้ของเครื่อง โดยทั่วไป แต่เพื่อให้ผู้อ่านเข้าใจอย่างชัดเจน ตัวอย่างโปรแกรมที่ใช้จะแสดงด้วยภาษาไพธอน (เวอร์ชันสาม). แบบฝึกหัดเขียนโปรแกรมนี้ออกแบบมา เพื่อ ทบทวนทักษะการเขียนโปรแกรมด้วยภาษาไพธอน

### แบบฝึกหัด 1.1

จงเขียนโปรแกรมเพื่อพิมพ์ข้อความต่อไปนี้ ออกมาก็หน้าจอ โดยให้มีการขึ้นบรรทัดตามที่แสดง

Bruce Lee:

**Knowing is not enough, we must apply.**

**Willing is not enough, we must do.**

คำใบ้ ลองคำสั่ง **print**

### แบบฝึกหัด 1.2

จงเขียนโปรแกรมเพื่อรับตัวเลขจำนวนเต็มจากผู้ใช้ และพิมพ์ตัวเลขนั้น พร้อมค่ากำลังสองของมัน ดัง ตัวอย่าง

**Enter a number: 4**

**4 is squared to 16**

เมื่อ 4 ในท้ายบรรทัดแรกเป็นอินพุตจากผู้ใช้

คำใบ้ (1) ลองคำสั่ง **input**, (2) เปรียบเทียบผลลัพธ์ของ "3"+"5" กับของ **int("3") + 5** และ (3) ลองคำสั่ง **5\*\*2**

### แบบฝึกหัด 1.3

จากภาพยนตร์เรื่องคนหลุดโลก (Cast Away ก.ศ. 2000) ซัค โนแลนด์ รอดชีวิตจากเครื่องบินตก และติดอยู่ที่เกาะร้าง เขาลองคำนวณหาโอกาส ที่ทีมค้นหาจะพบเขาที่เกาะร้าง ดังนี้ (1) เครื่องบินด้วยความเร็ว  $v$  เมล์ต่อชั่วโมง. (2) เครื่องบินติดต่อ กับ หอควบคุมการบินไม่ได้ เป็นเวลา  $T$  ชั่งโมงก่อนจะตก. ซัคต้องการคำนวณหาพื้นที่ที่ทีมค้นหานำไปตั้งค้นหา.

จงเขียนโปรแกรม เพื่อคำนวณพื้นที่ค้นหา โดยรับความเร็วเครื่องบิน  $v$  เมล์ต่อชั่วโมง และเวลา  $T$  ชั่วโมง จากที่ขาดการติดต่อจนถึงเครื่องตก. โปรแกรมรายงานออกมาระบุเป็นพื้นที่ตารางเมล์ และเปรียบเทียบกับพื้นที่ของประเทศไทย โดยพื้นที่ประเทศไทย มีขนาดประมาณ 513120 ตารางกิโลเมตร หรือ 198120 ตารางเมล์.

ตัวอย่างโปรแกรม

**Plane speed (mph):** 475

**Time from the last contact to crash (h):** 1

**Search area =** 708821.84 sq.mi.

**That is 3.58 times the size of Thailand.**

เมื่อ 475 ในบรรทัดแรก และ 1 ในบรรทัดที่สอง เป็นอินพุตจากผู้ใช้ และ 708821.84 กับ 3.58 เป็นผลการคำนวณ

คำใบ้ (1) พื้นที่ค้นหา  $a$  ตารางเมล์ คำนวณได้จาก  $a = \pi r^2$  เมื่อ  $r = v \cdot T$ . (2) คำสั่ง **round** สามารถใช้ช่วยปัดเศษได้ เช่น **round(21.842, 2)** จะให้ผลลัพธ์เป็น 21.84 (ปัดเป็นเลขทศนิยมสองตำแหน่ง). (3) มодูล **math** มีฟังก์ชันและค่าคงที่ทางคณิตศาสตร์ต่าง ๆ ที่มีประโยชน์. มодูล **math** จะถูกนำเข้ามาใช้งานได้ โดยคำสั่ง **import math** และค่า  $\pi$  สามารถเรียกได้จาก **math.pi**

## แบบฝึกหัด 1.4

จงเขียนฟังก์ชันคำนวณเวลาที่ลูกเห็นนิสิ่งจากหน้าไม้ของผู้เชิร์ฟไปถึงท้ายสนามเห็นนิสิ่งผู้รับ และคำนวณพลังงานที่ใช้ในการเชิร์ฟ ในหน่วยจูล (Joules ตัวย่อ J) และในหน่วยแคลอรี่ (calories ตัวย่อ cal) โดยฟังก์ชันรับ ค่าน้ำหนักของลูกบอล  $m$  กรัม ค่าความเร็วสูงสุดของลูกบอล  $v$  ในหน่วยกิโลเมตรต่อชั่วโมง และความยาวของสนามเห็นนิส  $d$  เมตร. สมมติว่าไม่มีแรงต้านทางอากาศ ไม่มีผลกระทบแรงดึงดูดของโลก ไม่มีผลกระทบกระเด้งที่ผิวนานม และคิดประมาณระยะทางเฉพาะในแนวราบทิศทางความยาวสนาม.

ตัวอย่างการเรียกใช้ฟังก์ชัน

```
time, energy, cal = serve(200, 23.8, 58)
print(time, 's')
```

```
print(energy, 'J')
print(cal, 'cal')
```

เมื่อ 200 คือความเร็วสูงสุดของลูกบอล ในหน่วยกิโลเมตรต่อชั่วโมง 23.8 คือความยาวสนาม ในหน่วยเมตร 58 คือน้ำหนักลูกบอล ในหน่วยกรัม และ **serve** คือฟังก์ชันที่ใช้คำนวน. ผลลัพธ์คือ

**0.8568 s**

**89.51 J**

**21.39 cal**

คำใบ้ (1) ระยะที่ลูกบอลเดินทางประมาณจากความยาวสนาม. (2) เวลาที่ลูกบอลวิ่ง  $t$  คำนวนจาก  $d = v_0 + \frac{1}{2} \cdot a \cdot t^2$  เมื่อ  $v_0$  คือความเร็วต้น (ประมาณเป็นศูนย์ ขณะลูกกระแทบที่น้ำแข็ง) และ  $a$  เป็นความเร่งเฉลี่ยของลูกบอล ในหน่วย เมตรต่อวินาทีกำลังสอง. (3) ความเร่งเฉลี่ยของลูกบอล  $a$  ประมาณได้จาก  $a = v/t$ . (4) แรงเฉลี่ยที่ใช้  $f$  ในหน่วยนิวตัน คำนวนได้จาก  $f = m \cdot a$ . (5) พลังงานที่ใช้  $e$  ในหน่วยจูล ประมาณได้จาก  $e = f \cdot d$ . (6) หนึ่งแคลอรีเท่ากับ 4.184 จูล. (7) โปรแกรมกำหนดฟังก์ชันด้วยไวยากรณ์

```
def func_name(arg1, arg2, arg3):
    # function body
    ...
    return output1, output2
```

(8) แนวทางปฏิบัติที่ดีในการเขียนโปรแกรมไพธอน คือ ส่วนของโปรแกรมหลักจะเขียนอยู่ในรูปแบบ

```
if __name__ == '__main__':
    # main program
    ...
```

## แบบฝึกหัด 1.5

บริษัทขนส่งแห่งหนึ่ง คิดค่าบริการซึ่งประกอบด้วย ค่าบริการส่ง (คิดตามพื้นที่) และค่าส่งของ (คิดตามน้ำหนัก) โดย ค่าบริการส่ง คิด 50 บาท ถ้าส่งในเขตจังหวัดขอนแก่น และคิด 100 บาท ถ้าส่งนอกเขตจังหวัดขอนแก่น. ค่าส่งของ คิดดังนี้ (1) คิด 8 บาทต่อกิโลกรัม สำหรับของน้ำหนักไม่เกิน 10 กิโลกรัม (2) คิด 12 บาทต่อกิโลกรัม สำหรับของน้ำหนักเกิน 10 กิโลกรัม แต่ไม่เกิน 20 กิโลกรัม และ (3) คิด 15 บาทต่อกิโลกรัม สำหรับของน้ำหนัก 20 กิโลกรัมขึ้นไป.

จะเขียนฟังก์ชันรับที่อยู่ และน้ำหนักของ แล้วคำนวณค่าส่งของบริษัทแห่งนี้.

ตัวอย่างการเรียกใช้ฟังก์ชัน

```
cost = delivery_kk("Khon Kaen", 14)
print(cost)
```

เมื่อ "Khon Kaen" คือพื้นที่ส่ง (อยู่ในเขตจังหวัดขอนแก่น) 14 คือน้ำหนักของที่ต้องการส่ง และฟังก์ชัน

`delivery_kk` ทำหน้าที่คำนวณค่าจัดส่ง. ผลลัพธ์คือ 218 ซึ่งคือค่าจัดส่ง  $50 + 14 \cdot 12 = 218$  บาท.

คำใบ้ ไฟรอนใช้ไวยากรณ์เงื่อนไข ดังนี้

```
if cond:
    # if body
    ...
# statement after condition
```

หากเป็นเงื่อนไขทางเลือก ใช้ไวยากรณ์ดังนี้

```
if cond:
    # if body
    ...
else:
    # else body
    ...
# statement after condition
```

หากเป็นเงื่อนไขทางเลือกหลายทาง ใช้ไวยากรณ์ดังนี้

```
if cond:
    # if body
    ...
elif cond:
    # elif body
    ...
else:
    # else body
    ...
# statement after condition
```

## แบบฝึกหัด 1.6

จงเขียนโปรแกรมเพื่อคำนวณค่ารากกำลังสองเฉลี่ย (root mean square คำย่อ RMS) โดยรับจำนวนของค่าที่ต้องการคำนวณ และรับค่าเหล่านั้นทีละค่าจนครบ และคำนวณค่ารากกำลังสองเฉลี่ย เมื่อได้รับค่าต่าง ๆ ครบตามจำนวนแล้ว.

ตัวอย่างโปรแกรม

**Number of values:** 4

**value 1:** 10

**value 2:** 2

**value 3:** 0.4

**value 4:** 3.8

**RMS =** 5.445181356024793

เมื่อ 4 ในบรรทัดแรก เป็นอินพุตที่ผู้ใช้ระบุจำนวนค่า และค่า 10 ค่า 2 ค่า 0.4 และ 3.8 เป็นอินพุตที่ผู้ใช้ป้อน ส่วน **value 1** ไปจนถึง **value 4** เป็นสิ่งที่โปรแกรมพิมพ์ออกไปหน้าจอ และ 5.445181356024793 เป็นผลลัพธ์การคำนวณ  $\sqrt{\frac{10+2+0.4+3.8}{4}} = 5.445181356024793$ .

คำให้ (1) ค่ารากกำลังสองเฉลี่ย **rms** คำนวณจาก  $rms = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$  เมื่อ  $N$  เป็นจำนวนค่า และ  $x_i$  เป็นค่าต่าง ๆ ที่ต้องการคำนวณ. (2) modulus **math** มีฟังก์ชัน **math.sqrt** เพื่อใช้คำนวณค่าราก. (3) ตัวอย่างรูปแบบไวยากรณ์เพื่ออนสำหรับการวนซ้ำ คือ

```
for i in range(num):
    # statement to be repeated
    ...
# statement after for loop
```

เมื่อ **num** เป็นจำนวนครั้งที่ต้องการวนซ้ำ และตัวแปร **i** เป็นดัชนีของการวนซ้ำ. (4) ไพรอนมีวิธีจัดรูปแบบข้อมูลสายอักขระ (string) ได้หลายแบบ (4.1) ใช้ตัวดำเนินการ % เช่น "**value %d**"%8 ซึ่งจะแสดงผลเป็น **value 8** หรือ (4.2) ใช้เมท็อด **format** ของข้อมูลสายอักขระ เช่น "**value {}**".**format(8)** ซึ่งจะแสดงผลเป็น **value 8** เช่นกัน. (5) ตัวดำเนินการ = เป็นตัวดำเนินการกำหนดค่า (assignment operator) ซึ่งทำงานโดย ประเมินค่าจากนิพจน์ที่อยู่ทางซ้ายมือ และนำค่าไปเก็บไว้ในตัวแปรที่อยู่ทางขวา เช่น **x = 3 + 4** คือ การกำหนดค่าให้ตัวแปร **x** เป็นค่า 7 ซึ่งได้จากการประเมินนิพจน์ **3+4**. ทำนอง

เดียวกัน  $x = x + 1$  คือ การกำหนดค่าให้ตัวแปร  $x$  เป็น ค่าจากการประเมินนิพจน์  $x + 1$  ดังนั้นหากรันคำสั่ง  $x = x + 1$  นี้แล้ว ตัวแปร  $x$  จะมีค่าเพิ่มจากเดิมขึ้นหนึ่ง.

### แบบฝึกหัด 1.7

จงเขียนฟังก์ชัน เพื่อประมาณค่าความน่าจะเป็นของเหตุการณ์ต่าง ๆ จากจำนวนครั้งที่พบ.

ตัวอย่างการเรียกใช้ฟังก์ชัน

```
count = [0, 8, 20, 4, 12, 1, 5]
p = est_prob(count)
print(p)
```

เมื่อ **count** คือ ตัวแปรของข้อมูลชนิดลิสต์ (list) ที่เก็บจำนวนครั้งของเหตุการณ์ 7 เหตุการณ์ โดย ตัวเลขในแต่ละตำแหน่ง แทนจำนวนครั้งที่พบเหตุการณ์นั้น เช่น เหตุการณ์ที่ 1 ไม่พบเลย เหตุการณ์ที่ 2 พบร 8 ครั้ง. ส่วน **est\_prob** คือฟังก์ชันที่ประมาณความน่าจะเป็น. ผลลัพธ์คือ

**[0.0, 0.16, 0.4, 0.08, 0.24, 0.02, 0.1]**

ซึ่งหมายถึง ความน่าจะเป็นที่คำนวณได้ สำหรับเหตุการณ์ต่าง ๆ ตามลำดับ เช่น เหตุการณ์ที่ 1 มีความน่าจะเป็น เป็น 0 เหตุการณ์ที่ 2 มีความน่าจะเป็น เป็น 0.16.

คำใบ้ (1) ความน่าจะเป็น  $p_i$  ประมาณได้จาก  $p_i = \frac{c_i}{\sum_{j=1}^N c_j}$  เมื่อ  $c_i$  คือจำนวนครั้งที่พบเหตุการณ์  $i$  และ  $N$  คือจำนวนเหตุการณ์ทั้งหมด. (2) แต่ละค่าของลิสต์สามารถนำอกมาได้โดยการใช้ดัชนี เช่น **count[2]** จะได้ค่า 20 ออกมาก (ดัชนีแรก เริ่มที่ 0). (3) ฟังก์ชัน **len** สามารถช่วยนับจำนวนรายการทั้งหมดในลิสต์ได. (4) คำสั่ง **for** สามารถทำงานกับลิสต์ได้โดยตรง เช่น

```
for c in count:
    print(c)
```

(5) ลิสต์ว่าง สามารถสร้างได้ เช่น **prob = []** กำหนดตัวแปร **prob** ให้มีค่าเป็นลิสต์ว่าง. (6) ลิสต์สามารถเพิ่มรายการเข้าไปได้ เช่น **prob.append(0.1)** เป็นการเพิ่มรายการ 0.1 เข้าไปในลิสต์ของตัวแปร **prob**.

### แบบฝึกหัด 1.8

จงเขียนฟังก์ชันที่รับข้อความ และนับความถี่ของคำต่าง ๆ ในข้อความ แล้วส่งผลการนับความถืออกมา.

ตัวอย่างการเรียกใช้ฟังก์ชัน

```
txt = "Evil is done by oneself; " + \
"by oneself is one defiled. "+ \
"Evil is left undone by oneself; " + \
"by oneself is one cleansed. "
```

```
wf = word_freq(txt)
print(wf)
```

เมื่อ `txt` คือ ตัวแปรที่เก็บข้อความ และ `word_freq` คือฟังก์ชันที่นับความถี่ของคำ ในข้อความของ `txt`. ผลลัพธ์คือ

```
{'is': 4, 'left': 1, 'done': 1, 'Evil': 2, 'one': 2,
'cleansed': 1, 'oneself': 4, 'undone': 1, 'defiled': 1,
'by': 4}
```

ซึ่งอยู่ในรูปของเพรอนดิกชันนารี (dictionary).

คำใบ้ (1) ใช้ฟังก์ชันข้างล่าง เพื่อจัดการคำต่าง ๆ ให้เรียบร้อย

```
def clean_txt(msg):
    msg = msg.replace('.', ' ')
    msg = msg.replace(';', ' ')
    msg = msg.replace('\n', ' ')
    msg = msg.replace(' ', ' ')

return msg
```

(2) เมท็อด `split` ของข้อมูลสายอักขระ สามารถช่วยแยกคำต่าง ๆ ออกมากจากข้อความได้สะดวก เช่น `"Evil is left".split()` จะให้ลิสต์ `['Evil', 'is', 'left']` ออกมา.

(3) เมท็อด `strip` ช่วยตัดซองว่ารอบคำออกได้สะดวก. (4) ดิกชันนารีว่า สามารถสร้างได้ เช่น `w = {}` จะสร้างดิกชันนารีว่า ให้กับตัวแปร `w`. (5) การอ้างอิงรายการของดิกชันนารี จะใช้กุญแจด้วย ซึ่งเป็นชื่อเช่นเดียว กับด้วยของลิสต์ เพียงแต่ กุญแจด้วยของดิกชันนารีสามารถใช้เป็นสายอักขระได้ เช่น `w['Evil'] = 1` เป็นการกำหนดค่า 1 ให้กับรายการที่มีกุญแจด้วยเป็น '`Evil`' ซึ่งหากยังไม่มีรายการของกุญแจอยู่ ไฟ-รอนจะสร้างขึ้นมาใหม่ แต่หากมีอยู่แล้วค่า 1 ก็จะไปแทนที่ค่าเดิมของรายการนี้. กลไกนี้ทำให้ดิกชันนารี สะดวกมากกับการใช้นับความถี่คำในลักษณะเช่นนี้. (6) เช่นเดียวกับตัวแปรเดียว รายการของตัวนี้สามารถใช้

ในลักษณะการเปลี่ยนแปลงค่าได้ เช่น `w['Evil'] += 1` จะเป็นการเพิ่มค่าของรายการของกุญแจดังนี้ 'Evil' จากเดิม ขึ้นไปหนึ่ง.

### แบบฝึกหัด 1.9

ต่อรหัสดีเอ็นเอ. ดีเอ็นเอประกอบด้วย ฐานนิวคลีโอไทด์ (nucleotide bases) สี่ชนิด ได้แก่ อะดีนีน (adenin ตัวย่อ A) ไซโตซีน (cytosine ตัวย่อ C) กัวานีน (guanine ตัวย่อ G) และ ไฮมีน (thymine ตัวย่อ T). ลำดับของฐานนิวคลีโอไทด์ต่าง ๆ จะเป็นข้อมูลที่เซลล์นำไปใช้ในกระบวนการสร้างโปรตีน. นั่นคือ ลำดับของฐานนิวคลีโอไทด์สามตัว จะบอกชนิดของกรดอะมิโน (amino acid) ที่จะเซลล์จะสร้างเพื่อไปประกอบเป็นโปรตีน (หรืออาจจะเป็นรหัส เพื่อบอกการจบของลำดับสายกรดอะมิโน). ชุดของฐานนิวคลีโอไทด์สามตัว จะเรียกว่า โคดอน (codon). โคดอน จะถูกอ่านตามลำดับ และจะไม่มีอ่านดีเอ็นเอซ้อนกัน เช่น 'AAGGGC' จะอ่านเป็นโคดอนสองชุด คือ 'AAG' และ 'GGC'.

จงเขียนฟังก์ชัน เพื่อแปลงจากลำดับดีเอ็นเอ ไปเป็นโปรตีน ซึ่งคือ สายของกรดอะมิโน โดย ฟังก์ชันรับไฟล์ตารางโคดอน ที่เป็นตารางการแปลงโคดอนไปเป็นกรดอะมิโน และรับไฟล์ลำดับดีเอ็นเอ แล้วผลลัพธ์คือ

ตัวอย่างไฟล์ตารางโคดอนและตัวอย่างไฟล์ดีเอ็นเอ สามารถดาวน์โหลดได้จาก <http://degas.en.kku.ac.th/coewiki/doku.php?id=pr:advbook> (ภายใต้หัวข้อ ข้อมูลประกอบแบบฝึกหัด). ตัวอย่างการเรียกใช้ฟังก์ชัน

```
protein = codon('codons.txt', 'homo_sapiens_mitochondrion.txt')
print(protein)
```

เมื่อ `codons.txt` คือ ชื่อไฟล์ตารางแปลงโคดอน `homo_sapiens_mitochondrion.txt` คือชื่อไฟล์ลำดับของดีเอ็นเอ ที่ต้องการแปลง และ `codon` คือฟังก์ชันที่แปลงโคดอนเป็นโปรตีน. ผลลัพธ์คือ

```
['Lysine', 'Glycine', 'Leucine', 'Alanine', 'stop', 'Leucine',
'Lysine', 'Tryptophan', 'Leucine', 'Isoleucine', 'Cysteine',
'Valine', 'Glutamine', 'Leucine', 'Methionine', 'Glutamine',
'Serine', 'Glycine', 'Valine', 'Leucine', 'Glutamine',
'Serine', 'Leucine']
```

ซึ่งอยู่ในรูปของลิสต์.

คำใบ้ (1) เปิดดูเนื้อหาในไฟล์ก่อน เพื่อเข้าในรูปแบบของข้อมูลที่เก็บ. (2) พรอนใช้ไวยากรณ์ดังนี้ในการเปิดอ่านไฟล์

```
with open('filename', 'r') as f:  
    file_content = f.read()  
    # ... process file_content
```

โดย 'filename' แทนชื่อไฟล์ที่ต้องการเปิดอ่าน (ระบุด้วย 'r') และใช้ตัวแปร **f** เป็นตัวจัดการไฟล์ (file handle). เมื่ออ่าน **read** ใช้อ่านเนื้อหาทั้งหมดของไฟล์ออกมานะ. (3) การอ่านดีเอ็นเอมาทีละชุด ชุดละสาม สามารถทำได้หลายวิธี หนึ่งในเทคนิคที่สะดวกคือ (3.1) ใช้ **range(0, len(dna), 3)** เพื่อหาตำแหน่งเริ่มต้นของแต่ละชุดโคดอน เมื่อ **dna** เป็นข้อมูลสายอักขระที่เก็บลำดับของดีเอ็นเอ (3.2) ใช้เทคนิคการตัด (slicing) เช่น **dna[i:(i+3)]** เพื่อดึงโคดอนออกมานะ เมื่อ **i** เป็นตำแหน่งเริ่มของโคดอน. (4) ถ้าอ่านตารางโคดอนและจัดทำเป็นดิกชันนารีไว้ก่อน จะทำให้การแปลงสะดวกมาก.

## แบบฝึกหัด 1.10

โน้ตดนตรีในระดับเสียงเต็มรูป (diatonic notes) คือ โน้ตดนตรี 7 ตัวโน้ตในระดับเสียง (scale). ตัวโน้ตทั้งเจ็ดนี้ จะนิยามต่างกันไปสำหรับแต่ละกุญแจเสียง. ตัวอย่าง เช่น ระดับเสียงหลัก (major scale) ของกุญแจเสียง C (key of C) จะมีโน้ต C, D, E, F, G, A และ B. ระดับเสียงหลักของกุญแจเสียง G (key of G) จะมีโน้ต G, A, B, C, D, E และ F#. ระดับเสียงหลัก นิยามระดับเสียงเต็มรูป ตามเกณฑ์ดังนี้

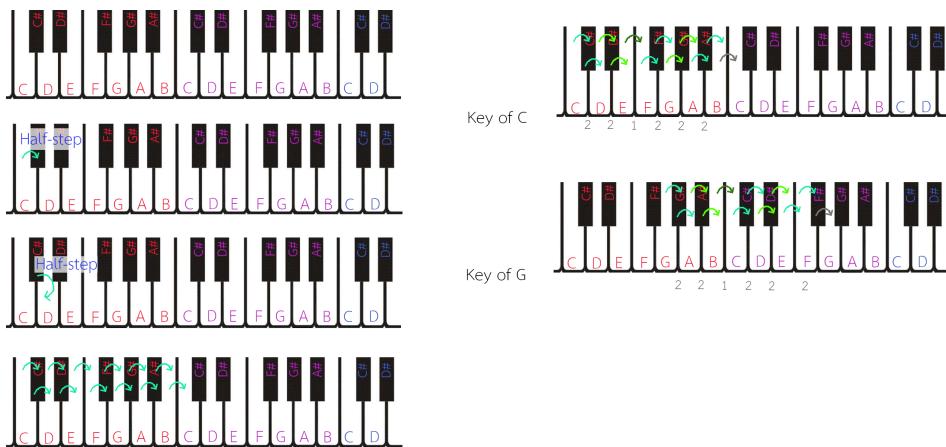
โน้ตดนตรีในระดับเสียงเต็มรูป	กุญแจเสียง C	กุญแจเสียง G
ตัวแรก เป็นโน้ตของกุญแจเสียง	C	G
ตัวที่สอง เสียงสูงขึ้น 2 ครั้งขึ้น จำกตัวแรก	D	A
ตัวที่สาม เสียงสูงขึ้น 2 ครั้งขึ้น จำกตัวที่สอง	E	B
ตัวที่สี่ เสียงสูงขึ้น 1 ครั้งขึ้น จำกตัวที่สาม	F	C
ตัวที่ห้า เสียงสูงขึ้น 2 ครั้งขึ้น จำกตัวที่สี่	G	D
ตัวที่หก เสียงสูงขึ้น 2 ครั้งขึ้น จำกตัวที่ห้า	A	E
ตัวที่เจ็ด เสียงสูงขึ้น 2 ครั้งขึ้น จำกตัวที่หก	B	F# (2 ครั้งขึ้น นั่นคือ E → F → F#)

หมายเหตุ ครั้งขึ้น (half-step) กล่าวโดยง่าย คือ ห่างกัน 1 ก้านดีดเปย์โน (รวมก้านดีดทั้งสี่ขาและดำเนินรูป 1.5 ประกอบ).

จงเขียนฟังก์ชัน **diatonic** ที่รับอาร์กิวเม้นต์ **scale\_key** สำหรับกุญแจเสียง แล้วให้ค่าโน้ตดนตรี ในระดับเสียงเต็มรูปօอกมา โดยใช้เลขจำนวนเต็มแทนโน้ตดนตรีต่าง ๆ ดังนี้ เลข 1 แทนโน้ต C, เลข 2 แทนโน้ต C#, เลข 3 แทนโน้ต D, เลข 4 แทนโน้ต D# เป็นต้น.

คำໃບ້ ມອດຸໂລ (modulo) ອໍາວິກາຮາຮາເອາເສີ່ງ ຈຶ່ງໃຫ້ຕັດນຳເນີນການ % ອາຈ່າຍໃຫ້ທຸກອຍ່າງຈ່າຍຂຶ້ນ  
ຕ້ວຍຢ່າງຜລກາຮາທຳການ

```
>>> diatonic(1)
(1, 3, 5, 6, 8, 10, 12)
>>> diatonic(5)
(5, 7, 9, 10, 12, 2, 4)
>>> diatonic(10)
(10, 12, 2, 3, 5, 7, 9)
```



ຮູບທີ 1.5: ກາພແຄວບນສຸດຊ້າຍ ແສດງກໍານົດເປີຍໂນ ພຣ້ອມໂນ້ຕດນົດ. ກາພຊ້າຍແຄວສອງ ແຄວສາມ ແລະ ແຄວສີ ແສດງລູກສະບຸຮະບຸຮະດັບ ເສີຍງຄົງຂຶ້ນ. ສັງເກດ E → F ແລະ B → C ເພີ່ມຮະດັບເສີຍແຄ່ຄົງຂຶ້ນ (ໄມ່ມີກໍານົດເດືອຍຸ່ງຕຽກຄາງ). ກາພບນທາງໝາວ ແສດງກໍານົດເປີຍໂນ ພຣ້ອມໂນ້ຕດນົດ ແລະ ລູກສະແສດງການເພີ່ມຮະດັບເສີຍທີ່ລະຄົງຂຶ້ນ ຈາກກຸງແຈເສີຍ C ເບີ່ຍົບເທື່ອບກັບກາພລ່າງທາງໝາວ ທີ່ແສດງ ກາພທາໂນ້ຕິໃນຮະດັບເສີຍເຕີມຮູປ ເມື່ອໃໝ່ກຸງແຈເສີຍ G.

## บทที่ 2

### พื้นฐาน

``Divide each difficulty into as many parts as is feasible and necessary to resolve it."

---René Descartes

“แบ่งปัญหาออกเป็นส่วนย่อย ๆ เท่าที่จะทำได้และจำเป็นที่จะแก้มันได้”

—เรอเน่ เดการ์ต

ศาสตร์การรู้จำรูปแบบและการเรียนรู้ของเครื่อง อาศัยพื้นฐานจากหลาย ๆ ศาสตร์ การทำความเข้าใจศาสตร์นี้ และพัฒนาการ จำเป็นต้องอาศัยศาสตร์พื้นฐาน. บทนี้จะทบทวนศาสตร์พื้นฐานที่สำคัญ คือ พีชคณิตเชิงเส้น ความน่าจะเป็น และการหาค่าดีที่สุด.

#### 2.1 พีชคณิตเชิงเส้น

การรู้จำรูปแบบและการเรียนรู้ของเครื่อง เกี่ยวข้องกับข้อมูลและตัวแปรจำนวนมาก. พีชคณิตเชิงเส้น<sup>1</sup> มีเครื่องมือและทฤษฎีต่าง ๆ ที่ช่วยอำนวยความสะดวก ในการทำงานกับตัวแปรจำนวนมาก ดังนั้น จึงเป็นเป็นพื้นฐานที่สำคัญ

สเกลาร์ (scalar) หมายถึง ตัวเลขเดียว เช่น ตัวเลข 3 ตัวเลข 0 ตัวเลข  $-0.42$  ตัวเลข  $168.79$ . กำหนดให้  $\mathbb{R}$  แทนเขตของจำนวนจริง. ดังนั้น สัญกรณ์ เช่น  $x \in \mathbb{R}$  ระบุว่า ตัวแปร  $x$  เป็นสเกลาร์ของจำนวนจริง.

เวกเตอร์ (vector) หมายถึง ลำดับของตัวเลข. เวกเตอร์ ในพีชคณิตเชิงเส้น มีสองชนิด คือ เวกเตอร์แนวอน แล้วเวกเตอร์แนวตั้ง. เวกเตอร์แนวอน แสดงด้วยลำดับในแนวอน เช่น  $[103.4, -28.6, 0, 9.99]$  เป็นเวกเตอร์แนวอน ที่เป็นลำดับของตัวเลขสี่ตัว. คำราเล่นนี้จะใช้เวกเตอร์แนวตั้งเป็นหลัก นั่นคือ หากกล่าวถึง เวกเตอร์ โดยไม่ได้ระบุเฉพาะเจาะจงแล้ว จะหมายถึง เวกเตอร์แนวตั้ง ที่

<sup>1</sup>เนื้อหาในหัวข้อนี้ได้รับอิทธิพลหลักจาก [76] [39] และ [190]

แสดงด้วยลำดับในแนวตั้ง ดังแสดงใน สมการ 2.1

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.1)$$

เมื่อ  $x_1, \dots, x_n$  แทนตัวเลขสเกลาร์ และ  $x_i$  เรียกว่าเป็น ส่วนประกอบที่  $i$  ของเวกเตอร์  $\mathbf{x}$ . สัญกรณ์ เช่น  $\mathbf{x} \in \mathbb{R}^n$  ระบุว่า ตัวแปร  $\mathbf{x}$  เป็นเวกเตอร์ ที่มีส่วนประกอบจำนวน  $n$  ตัว ซึ่งส่วนประกอบแต่ละตัวเป็นจำนวนจริง. หมายเหตุ สัญลักษณ์  $\mathbf{0}$  หมายถึง เวกเตอร์ที่ส่วนประกอบทุกตัวเป็นศูนย์. นั่นคือ  $\mathbf{0} = [0, 0, \dots, 0]^T$ .

เมตริกซ์ (matrix) หมายถึง โครงสร้างสองมิติของลำดับของตัวเลข ดังแสดงในสมการ 2.2

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \quad (2.2)$$

เมื่อ  $\mathbf{A}$  เป็นเมตริกซ์ขนาดมิติ  $m \times n$  และ  $A_{11}, \dots, A_{mn}$  แทนตัวเลขสเกลาร์ และ  $A_{ij}$  เป็น ส่วนประกอบของเมตริกซ์ ที่ได้ชนิดตามนั้น แถว  $i$  และส عم��  $j$ . ดังนี้ อาจเขียนโดยใช้ตัวห้อย โดยมีเครื่องหมายจุลภาค คัน เช่น  $A_{i,j}$  หรือเครื่องหมายจุลภาคคันอาจถูกละไว้ได้ เช่น  $A_{ij}$  ในกรณีที่ความหมายชัดเจน. เมตริกซ์ สามารถถูกระบุขนาดมิติ ได้จาก สัญกรณ์  $\mathbf{A} \in \mathbb{R}^{m \times n}$ .

นอกจากนี้ ศาสตร์การเรียนรู้ของเครื่อง นิยมใช้สัญกรณ์จุดคู่ ดังปฏิบัติใน [76]. นั่นคือ กำหนดให้ สัญกรณ์  $\mathbf{A}_{i,:}$  หมายถึง เมตริกซ์ย่อย ที่ได้จากส่วนประกอบที่แถว  $i$  ของเมตริกซ์  $\mathbf{A}$  คือ  $\mathbf{A}_{i,:} = [A_{i1}, A_{i2}, \dots, A_{in}]$  เรียกว่า แถว  $i$  ของ  $\mathbf{A}$ . ทำนองเดียวกัน สัญกรณ์  $\mathbf{A}_{:,j}$  หมายถึง เมตริกซ์ย่อย ที่ได้จากส่วนประกอบที่ส عم馬  $j$  ของเมตริกซ์  $\mathbf{A}$  เรียกว่า ส عم馬  $j$  ของ  $\mathbf{A}$ .

การสลับเปลี่ยน (transpose) เป็นการดำเนินการจัดเรียงลำดับใหม่. สำหรับเวกเตอร์ การสลับเปลี่ยน ของเวกเตอร์แนวอน จะได้เวกเตอร์แนวตั้ง และการสลับเปลี่ยนของเวกเตอร์แนวตั้ง จะได้เวกเตอร์แนว อน และใช้สัญกรณ์ เช่น  $\mathbf{x}^T$  คือ การสลับเปลี่ยนของเวกเตอร์  $\mathbf{x}$  เช่น จากสมการ 2.1 เวกเตอร์  $\mathbf{x}^T = [x_1, x_2, \dots, x_n]$  หรือ ในทางกลับกัน สมการ 2.1 อาจเขียนใหม่ได้ในรูป  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$

สำหรับเมตริกซ์ การสลับเปลี่ยนของเมตริกซ์ ซึ่งใช้สัญกรณ์ เช่น  $\mathbf{A}^T$  คือ การเปลี่ยนตำแหน่งของส่วน ประกอบ โดย ส่วนประกอบที่ตำแหน่ง  $(i, j)$  จะถูกเปลี่ยนไปอยู่ตำแหน่ง  $(j, i)$  ซึ่งผลที่ได้เสมือนกับการสลับ

ตำแหน่งรอบแนวทางແຍງມູນຂອງເມທຣິກ່າ. ສມກາຣ 2.3 ແສດກາຣສລັບເປີ່ຍນຂອງເມທຣິກ່າ **A**.

$$\mathbf{A}^T = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} \\ A_{12} & A_{22} & \cdots & A_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{mn} \end{bmatrix} \quad (2.3)$$

ກາຣສລັບເປີ່ຍນຂອງເມທຣິກ່າ ຈະທຳໃຫ້ຂາດມືຕີຂອງເມທຣິກ່າເປີ່ຍນໄປ ເຊັ່ນ ຈາກຕ້ວອຍ່າງ ຂາດມືຕີຂອງເມທຣິກ່າ **A** ຄືອ  $m \times n$  ແຕ່ ຂາດມືຕີຂອງເມທຣິກ່າ  $\mathbf{A}^T$  ຄືອ  $n \times m$  (ສັງເກດຈາກ  $\mathbf{A}^T$  ມີ  $n$  ແລະ  $m$  ສດມກໍ)

**ມືຕີ ແລະ ລຳດັບໜັ້ນ.** ໂຄງສຮ້າງຂອງເວກເຕັກ ມີໜຶ່ງມືຕີຂອງລຳດັບ ໃນລັກໝນະທີ່ລຳດັບຂອງຂໍ້ມູນດຳເນີນໄປໄດ້ ແນວເດືອຍ. ໂຄງສຮ້າງຂອງເມທຣິກ່າ ມີສອງມືຕີ ໃນລັກໝນະທີ່ລຳດັບຂອງຂໍ້ມູນດຳເນີນໄປໄດ້ສອງແນວ. ປາພຂາວ ດຳ<sup>2</sup> ທີ່ວິກາພສເກລເທາ (gray-scale image) ກົມືລັກໝນະໂຄງສຮ້າງສອງມືຕີຂອງລຳດັບ ນັ້ນຄືອ ມີໂຄງສຮ້າງສອງມືຕີ ຂອງລຳດັບຄ່າພິກເໜລ ຕາມລຳດັບແນວຕັ້ງ ແລະ ຕາມລຳດັບແນວນອນ. ກາຣເຂົ້າໂປຣແກຣມຈະໃຫ້ໂຄງສຮ້າງຂໍ້ມູນ ເຊັ່ນ ອາຣີເຣີ (array) ຂາດສອງມືຕີ ໃນກາຣແທນຂໍ້ມູນກາພສເກລເທານີ້.

ຂໍ້ມູນລາຈມີໂຄງສຮ້າງມືຕີຂອງລຳດັບທີ່ໜັ້ນມີກຳນົດກຳນົດ ເຊັ່ນ ປາພສີ (color image) ມີໂຄງສຮ້າງສາມມືຕີຂອງລຳດັບ ຄ່າພິກເໜລ ຕາມລຳດັບແນວຕັ້ງ ຕາມລຳດັບແນວນອນ ແລະ ຕາມໜຸດຂອງໜ່ອງສີ ແດ້ ເຊິ່ງ ນໍ້າເຈີນ. (ແມ່ໜ່ອງສີໄມ້ໄດ້ມີ ຄວາມສົມພັນຮູນໃນເຂົ້າລຳດັບ ໃນລັກໝນະທີ່ໜ່ອງສີແດ່ງ ໄນຈະເປັນຕົ້ນມີກ່ອນໜ່ອງສີເຂິ່ງ ເປັນຕົ້ນ ແຕ່ຂໍ້ມູນຂອງໜ່ອງສີ ຕ່າງໆ ແກ່ອກອກເປັນລັກໝນະຂອງຕ້າວເອງ). ກາຣເຂົ້າໂປຣແກຣມຈະໃຫ້ໂຄງສຮ້າງຂໍ້ມູນ ເຊັ່ນ ອາຣີເຣີ ຂາດສາມມືຕີ ໃນກາຣແທນຂໍ້ມູນປາພສີ. ຂໍ້ມູນລົວດີໂອ (video) ເປັນຂໍ້ມູນເປັນລັກໝນະໂຄງສຮ້າງສົມມືຕີຂອງລຳດັບຄ່າພິກເໜລ ຕາມ ລຳດັບແນວຕັ້ງ ຕາມລຳດັບແນວນອນ ຕາມໜຸດຂອງໜ່ອງສີ ແລະ ຕາມລຳດັບເວລາ. ກາຣເຂົ້າໂປຣແກຣມຈະໃຫ້ໂຄງສຮ້າງ ຂໍ້ມູນ ເຊັ່ນ ອາຣີເຣີ ຂາດສົມມືຕີ ໃນກາຣແທນຂໍ້ມູນລົວດີໂອ ແລະ ຂໍ້ມູນລົວດີໂອ ທີ່ກ່າວເຖິງນີ້ ຍັ້ງໄມ້ໄດ້ຮັມຂໍ້ມູນຂອງເສີຍ ດ້ວຍ ຜົ່ງທາກເປັນໜ່ອງເສີຍເດືອຍ (monophonic sound channel) ກົມືຕີຂອງອາຣີເຣີອີກໜຶ່ງມືຕີ ຢ່ອງ ແກ່ອກເປັນໜ່ອງເສີຍສເຕອຣີໂອ (stereophonic sound channel) ກົມືຕີຂອງອາຣີເຣີອີກສອງມືຕີ ຢ່ອງທາກ ແກ່ອກໜ່ອງເສີຍພຸດ ອອກຈາກເສີຍປະກອບອື່ນ ຖ້າ ຢ່ອມີຂໍ້ມູນຄຳບຽຍພາຫາຕ່າງໆ ປະກອບ ກົມືຕີຂອງອາຣີເຣີເພີ່ມຂຶ້ນອີກ.

<sup>2</sup> ປາພຂາວດຳ ໃນບຣິບທີ່ໄປ ແມ່ຍົງ ປາພສເກລເທາ ທີ່ໃນປາພສາມຮັດແສດກຄ່ານ້ຳໜັກຮັດສີຕ່າງໆ ໄດ້ຕັ້ງແຕ່ຂາວ ໄປຈົນຄົງດຳ. ອຳຍ່າງເຮັດ ກົມືຕີ ໃນຮະບບໍຂໍ້ມູນ ມີຂໍ້ມູນປາພສີທີ່ເປັນຄ່າທິວກາດ ນັ້ນຄືອ ປາພຈະສາມຮັດແສດກສີໄດ້ແກ່ສອງສີ ນັ້ນຄືອ ແຕ່ລະພິກເໜລ ຈະສາມຮັດແສດກໄດ້ແກ່ສີຂາວ ຢ່ອສີດຳ ເຫັນນັ້ນ ໄນສາມຮັດແສດກຮະດັບສີອື່ນ ຖ້າ ຮະຫວ່າງກາລາໄດ້. ຕັ້ນນັ້ນ ເພື່ອລັດຄວາມສັບສນ ໃນທີ່ຈະໃຫ້ຄໍາວ່າ ປາພສເກລເທາ ແທນຄໍາວ່າ ປາພຂາວ ດຳ.

อย่างไรก็ตาม หากกล่าวถึง “มิติ” นั้นจะสังเกตว่า ภาพสเกลเทาสองมิติ เป็นข้อมูลโครงสร้างลำดับสองมิติ. ภาพสีสองมิติ เป็นข้อมูลโครงสร้างลำดับสามมิติ. วิดีโอ (ของภาพสองมิติไม่รวมข้อมูลเสียง) เป็นข้อมูลโครงสร้างลำดับสี่มิติ. นอกจากนี้ เมื่อกล่าวถึง ปริภูมิค่า (vector space) ซึ่ง ใช้บอกถึง ขนาดความเป็นไปได้ของลักษณะข้อมูล และ มิติของปริภูมิจะใช้บอกขนาด และความซับซ้อนของปริภูมินั้น ๆ เช่น ข้อมูลสเกลาร์ที่เป็นค่าจริง จะมีปริภูมิค่า เป็น  $\mathbb{R}$  เป็นปริภูมิค่าหนึ่งมิติ. แต่ละจุดข้อมูล อ้างถึงได้ด้วยตัวเลขตัวเดียว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาบนเส้นจำนวนจริง. ข้อมูลเวกเตอร์ที่มีส่วนประกอบเป็นค่าจริงสี่ค่า จะมีปริภูมิค่า เป็น  $\mathbb{R}^4$ . แต่ละจุดข้อมูล อ้างถึงได้ด้วยตัวเลขสี่ตัว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาในปริภูมิขนาดสี่มิติ. ข้อมูลเมทริกซ์ขนาด  $2 \times 2$  จะมีปริภูมิค่า เป็น  $\mathbb{R}^{2 \times 2}$ . แต่ละจุดข้อมูล อ้างถึงได้ด้วยตัวเลขสี่ตัว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาบนปริภูมิขนาดสี่มิติ เช่นเดียวกับ ข้อมูลเวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง. นั่นคือ เวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง มีความสามารถในการแทนข้อมูลเทียบเท่ากับ เมทริกซ์ขนาด  $2 \times 2$  เพียงแต่ ข้อมูลที่เก็บในเวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง ไม่ได้มีโครงสร้างลำดับ เหมือนกับ ข้อมูลที่เก็บในเมทริกซ์ขนาด  $2 \times 2$  และโครงสร้างลำดับเช่นนี้ โดยเฉพาะหากเป็นโครงสร้างตามธรรมชาติของข้อมูล สามารถนำมาใช้ในประโยชน์ และช่วยในการรู้จำรูปแบบอย่างมีประสิทธิภาพได้ (ดังเช่น ที่จะได้อธิบายในบท 5 ต่อไป)

อย่างไรก็ตาม เพื่อลดความสับสน จานี้ไป เมื่อกล่าวถึง “มิติ” จะมีการระบุอย่างชัดเจนว่า หมายถึง มิติ ในความหมายใด เช่น คำว่า มิติ ใช้ในความหมาย มุมมอง ซึ่งเป็นเป็นความหมายกว้าง ๆ ของมิติ และใช้ในความหมายของมิติโดยทั่วไป. คำว่า มิติปริภูมิค่า ใช้ในความหมายของ มิติของปริภูมิค่า และคำว่า ลำดับชั้น (rank) ใช้ในความหมายของ มิติของโครงสร้างลำดับ. ตัวอย่าง ภาพสเกลเทาขนาด  $600 \times 800$  พิกเซล เป็นภาพสองมิติ ที่เป็นข้อมูลลำดับชั้นสอง (มีลำดับตามแนวตั้ง และตามแนวนอน) หรือสามารถแทนด้วย เมทริกซ์ขนาดมิติ  $600 \times 800$  และมีมิติปริภูมิค่า เป็น  $480000$ . สัญกรณ์  $\{0, \dots, 255\}^{600 \times 800}$  จะระบุชัดเจนทั้งจากมุมมองของลำดับชั้น และมิติปริภูมิค่า. นอกจากนั้น สัญกรณ์นี้ยังระบุช่วงค่าที่เป็นไปได้ของข้อมูลด้วยว่า แต่ละค่าเป็นจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 255.

แทนเชอร์ (tensor) หมายถึง โครงสร้างลำดับชั้นของตัวเลข. สเกลาร์ คือ แทนเชอร์ ลำดับชั้นศูนย์ (rank-0 tensor). เวกเตอร์ คือ แทนเชอร์ ลำดับชั้นหนึ่ง (rank-1 tensor). เมทริกซ์ คือ แทนเชอร์ ลำดับชั้นสอง (rank-2 tensor). ข้อมูลที่แทนด้วยอาร์เรย์ขนาด  $n$  มิติ คือ แทนเชอร์ ลำดับชั้น  $n$  (rank- $n$  tensor). ตัวเลข

แต่ละตัวในแทนเชอร์ เป็น ส่วนประกอบของแทนเชอร์ และอ้างอิงได้โดยใช้ตัวนี้ ตามลำดับขั้น.

ตัวอย่าง เทนเชอร์ลำดับขั้นสี่  $\mathbf{A} \in \mathbb{R}^{1 \times 2 \times 2 \times 3}$  มีค่า

$$\mathbf{A} = \left[ \begin{bmatrix} [[1.2], [3]] & [[-8.7], [6]] \end{bmatrix} \quad \begin{bmatrix} [[0.9], [-1]] & [[4], [1]] \end{bmatrix} \quad \begin{bmatrix} [[11], [5]] & [[0.1], [0]] \end{bmatrix} \right]$$

และ  $A_{1,1,1,1} = 1.2; A_{1,1,1,2} = 0.9; \dots A_{1,2,2,2} = 1; A_{1,2,2,3} = 0.$

หมายเหตุ รูปแบบอักษรที่ใช้ คือ ฟังก์ชันจะใช้สัญลักษณ์ เช่น  $f$  หรือ  $\sigma$  หรือ  $\text{calc}$  โดยไม่มีการทำแบบอักษรตัวหนา ไม่ว่าฟังก์ชันจะให้ค่าออกมาเป็นสเกลาร์ หรือเมทริกซ์ หรือแทนเชอร์ เช่น  $f: \mathbb{R} \rightarrow \mathbb{R}$  และ  $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1 \times m_2 \times m_3}$  และ  $\text{calc}: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^q$ . สัญกรณ์ เช่น  $f: \rho_1 \rightarrow \rho_2$  จะบุ่าว่าฟังก์ชัน  $f$  รับค่าตัวแปรที่อยู่ในเซต  $\rho_1$  เพื่อไปทำการคำนวณ และให้ผลลัพธ์การคำนวณออกมาเป็นค่าที่อยู่ในเซต  $\rho_2$ . ตัวอย่าง เช่น  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  จะบอกว่า ฟังก์ชัน  $g$  รับตัวแปรเข้าเป็นเวกเตอร์ที่มี  $n$  ส่วนประกอบ และจะให้ผลลัพธ์ออกมาเป็นสเกลาร์.

**การคำนวณเมทริกซ์.** การบวกลบเมทริกซ์กับเมทริกซ์ จะทำได้ ก็ต่อเมื่อ เมทริกซ์มีขนาดมิติเท่ากัน และผลลัพธ์คำนวณได้จากค่าของส่วนประกอบทั้งสองที่ตำแหน่งเดียวกัน เช่น  $\mathbf{C} = \mathbf{A} + \mathbf{B}$  โดย  $C_{ij} = A_{ij} + B_{ij}$ . การคำนวณสเกลาร์กับเมทริกซ์ กำหนดให้ เป็นการคำนวณค่าสเกลาร์นั้น ๆ กับส่วนประกอบของเมทริกซ์แต่ละตัว เช่น  $\mathbf{D} = a \cdot \mathbf{B} + c$  โดย  $D_{ij} = a \cdot B_{ij} + c$ . การคำนวณเวกเตอร์กับเวกเตอร์ และการคำนวณสเกลาร์กับเวกเตอร์ ก็ทำในทำนองเดียวกัน. (ดูตัวอย่าง จากแบบฝึกหัด 2.5)

**การคูณกันของเมทริกซ์.** การคูณเมทริกซ์ (matrix product) จะดำเนินการได้ เมื่อเมทริกซ์สองเมทริกซ์ที่จะคูณกัน ต้องมีขนาดมิติที่เข้ากันได้ นั่นคือ จำนวนสدمภ์ของเมทริกซ์ตัวหน้า เท่ากับ จำนวนแຄวของเมทริกซ์ตัวหลัง และใช้สัญกรณ์ เช่น  $\mathbf{AB}$  หรือ  $\mathbf{A} \cdot \mathbf{B}$  โดย เมทริกซ์  $\mathbf{A}$  มีขนาดมิติ  $m \times p$  เมทริกซ์  $\mathbf{B}$  มีขนาดมิติ  $p \times n$  และ ผลลัพธ์  $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$  จะมีขนาดมิติ  $m \times n$  และ  $C_{ij} = \sum_k A_{ik} \cdot B_{kj}$ .

นอกจากการคูณเมทริกซ์แล้ว การดำเนินการ การคูณแบบตัวต่อตัว (element-wise product หรือ Hadamard product) ก็มีการใช้อย่างกว้างขวาง. การคูณแบบตัวต่อตัว จะดำเนินการได้ ก็ต่อเมื่อ เมทริกซ์สองเมทริกซ์ที่จะคูณกัน ต้องมีขนาดมิติที่เท่ากัน และใช้สัญกรณ์ เช่น  $\mathbf{A} \odot \mathbf{B}$  โดย เมทริกซ์  $\mathbf{A}$  มีขนาดมิติ  $m \times n$  เมทริกซ์  $\mathbf{B}$  มีขนาดมิติ  $m \times n$  และ ผลลัพธ์  $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$  จะมีขนาดมิติ  $m \times n$  และ  $C_{ij} = A_{ij} \cdot B_{ij}$ . การคูณกันของเวกเตอร์ จะแสดงเหมือนกับการดำเนินการเมทริกซ์ เช่น  $z = \mathbf{x}^T \cdot \mathbf{y}$  เมื่อ  $\mathbf{x}$  และ  $\mathbf{y}$  มีสัดส่วนเท่ากัน และ  $z = \sum_i x_i \cdot y_i$ . สังเกต  $z = \mathbf{x}^T \cdot \mathbf{y} = \mathbf{y}^T \cdot \mathbf{x}$ . (ดูตัวอย่าง จากแบบฝึกหัด 2.6)

คุณสมบัติของการคูณเมตริกซ์. การคูณเมตริกซ์ มีคุณสมบัติการกระจาย (distributive properties) เช่น  $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$  มีคุณสมบัติการเปลี่ยนกลุ่ม (associative properties) เช่น  $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$  แต่ การคูณเมตริกซ์ไม่มีคุณสมบัติการสลับที่.

## ระบบสมการ

เมตริกซ์ เวกเตอร์ และการดำเนินการต่าง ๆ ที่กล่าวมา นอกจากจะช่วยการอ้างถึง และการจัดการกับข้อมูล ทำได้อย่างสะดวกแล้ว ยังอำนวยความสะดวกในการอ้างถึง และจัดการกับปัญหาระบบสมการ หรือปัญหาที่มีตัวแปรไม่ทราบค่า และสมการที่เกี่ยวข้องจำนวนมาก ตัวอย่าง เช่น

$$x + y + z = 6 \quad (2.4)$$

$$2x + 2y - z = 3 \quad (2.5)$$

$$y + z = 5 \quad (2.6)$$

เป็นระบบสมการเชิงเส้น และสามารถเขียนได้กราฟทั้งรด ด้วยสัญกรณ์ ของเมตริกซ์ และเวกเตอร์ ดังนี้

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 5 \end{bmatrix}$$

และกราฟทั้งรดขึ้นอีก เป็น

$$\mathbf{Ax} = \mathbf{b} \quad (2.7)$$

หาก นิยาม

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \\ 0 & 1 & 1 \end{bmatrix} \text{ นิยาม } \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ และนิยาม } \mathbf{b} = \begin{bmatrix} 6 \\ 3 \\ 5 \end{bmatrix} \quad (2.8)$$

เมตริกซ์เอกลักษณ์ (identity matrix) ที่นิยมใช้สัญญาณลักษณ์  $\mathbf{I}$  (หรือ  $\mathbf{I}_n$  เมื่อต้องการระบุขนาดมิติ  $n \times n$ ) เป็นเมตริกซ์ที่มีค่าตามแนวทะแยงมุมเป็นหนึ่ง และค่าอื่น ๆ เป็นศูนย์ เช่น เมตริกซ์เอกลักษณ์ ขนาดมิติ

$3 \times 3$  คือ

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

เมทริกซ์เอกลักษณ์ มีคุณสมบัติสำคัญ คือ เมื่อคูณกับเมทริกซ์ใด หรือคูณกับเวกเตอร์ใด แล้วจะได้เมทริกซ์นั้น หรือเวกเตอร์นั้น ตัวเดิม. นั่นคือ  $\mathbf{A} \cdot \mathbf{I} = \mathbf{A}$ .

เมทริกซ์ผกผัน (matrix inverse) คือ เมทริกซ์คู่คูณ ที่เมื่อคูณกับคู่ของมันแล้ว ผลลัพธ์จะได้เป็นเมทริกซ์เอกลักษณ์ ใช้สัญญาณลักษณ์เป็นเมทริกซ์ที่มีตัวยกลบหนึ่ง เช่น  $\mathbf{A}^{-1}$  หมายถึง เมทริกซ์ผกผัน ที่เป็นคู่ผกผันกับ  $\mathbf{A}$  นั่นคือ  $\mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}$ . จากตัวอย่าง เมทริกซ์ผกผัน ที่เป็นคู่ของ  $\mathbf{A}$  ในสมการ 2.8 คือ

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ -0.6666667 & 0.3333333 & 1 \\ 0.6666667 & -0.3333333 & 0 \end{bmatrix} \quad (2.9)$$

จากเมทริกซ์ผกผัน สมการ 2.7 สามารถแก้ได้โดย  $\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$  ซึ่งเมื่อแทนค่าเข้าไป จะได้  $\mathbf{x} = [1, 2, 3]^T$  ซึ่งหมายถึง ตัวแปร  $x = 1$  ตัวแปร  $y = 2$  และตัวแปร  $z = 3$ . (ตัวอย่างวิธีการหาเมทริกซ์ผกผัน สามารถดูได้จากแบบฝึกหัด 2.7)

**ความเป็นอิสระเชิงเส้น.** สังเกตว่า ระบบสมการในตัวอย่างข้างต้น (สมการ 2.4 ถึง 2.6) มีสามสมการ และมีตัวแปรที่ไม่ทราบค่า (unknown variables) สามตัว ซึ่งทำให้เมทริกซ์สัมประสิทธิ์  $\mathbf{A}$  (สมการ 2.8) มีจำนวนแ亶เท่ากับจำนวนสตดมภ์ ซึ่งเรียกว่า เมทริกซ์จตุรัส (square matrix).

โดยทั่วไปแล้ว ถ้าหากมีจำนวนสมการน้อยกว่า (จำนวนแ亶ของเมทริกซ์ น้อยกว่า จำนวนสตดมภ์ หรือ เมทริกซ์ที่มีสัดส่วนตี้ยกว้าง) คำตอบของระบบสมการจะมีได้หลายค่า เช่น ตัวอย่างระบบสมการ

$$x + y + z = 6$$

$$2x + 2y - z = 3$$

จะได้

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \end{bmatrix} \text{ เวกเตอร์ตัวแปร } \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ และเวกเตอร์ค่าคงที่ } \mathbf{b} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

ซึ่งมีคำตอบหลายชุด เช่น  $x = 1, y = 2, z = 3$  หรือ  $x = 2, y = 1, z = 3$  หรือ  $x = 3, y = 0, z = 3$  หรือ  $x = 2.5, y = 0.5, z = 3$  เป็นต้น. (จริง ๆ คือ ทุกชุดค่า ที่  $x + y = 3$  และ  $z = 3$  สามารถเป็นคำตอบได้ทั้งหมด)

แต่หากมีจำนวนสมการน้อยกว่า (จำนวนแคลของเมทริกซ์ มากกว่า จำนวนสต็อก หรือ เมทริกซ์ที่มีสัดส่วนสูงแคบ) อาจจะไม่สามารถหาคำตอบของระบบสมการได้ เช่น ตัวอย่างระบบสมการ

$$x + y + z = 6 \quad (2.10)$$

$$2x + 2y - z = 3 \quad (2.11)$$

$$y + z = 5 \quad (2.12)$$

$$x + z = 5 \quad (2.13)$$

จะได้

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \text{ เวกเตอร์ตัวแปร } \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ และเวกเตอร์ค่าคงที่ } \mathbf{b} = \begin{bmatrix} 6 \\ 3 \\ 5 \\ 5 \end{bmatrix}$$

ซึ่งไม่สามารถหาคำตอบของสมการได้ (สังเกตว่า  $x = 1, y = 2, z = 3$  ที่เป็นคำตอบของสามสมการแรก 2.10 ถึง 2.12 ขัดแย้งกับสมการที่สี่ 2.13)

**เมทริกซ์เอกฐาน.** ถึงแม้ เมทริกซ์จะเป็นเมทริกซ์จตุรัส ก็ไม่ใช่ทุกเมทริกซ์จตุรัส ที่จะสามารถหาเมทริกซ์ผกผันคู่ของมันได้. เมทริกซ์ที่หาคู่ผกผันไม่ได้ จะเรียกว่า เมทริกซ์เอกฐาน (singular matrix). เช่น ตัวอย่างระบบสมการ

$$x + y + z = 6 \quad (2.14)$$

$$2x + 2y + 2z = 12 \quad (2.15)$$

$$y + z = 5 \quad (2.16)$$

จะได้

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{เวกเตอร์ตัวแปร } \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{และเวกเตอร์ค่าคงที่ } \mathbf{b} = \begin{bmatrix} 6 \\ 12 \\ 5 \end{bmatrix}$$

เมทริกซ์สัมประสิทธิ์ที่ได้จะเป็นเมทริกซ์เอกฐาน. สังเกต สมการแรก 2.14 กับสมการที่สอง 2.15 จะเห็นว่า สมการที่สอง มีค่าเท่ากับสมการแรกคูณสอง ซึ่งหมายความว่า ถึงแม้จะมีสองสมการ แต่ก็ให้ข้อมูลเทียบเท่ากับ การมีแค่สมการเดียว และสัมประสิทธิ์ของสมการ ทำให้  $\mathbf{A}_{1,:}$  และ  $\mathbf{A}_{2,:}$  ไม่เป็นอิสระเชิงเส้นแก่กัน. (ดูแบบฝึกหัด 2.1 สำหรับตัวอย่างของสอดคล้องไม่เป็นอิสระเชิงเส้นแก่กัน)

เซตของเวกเตอร์  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$  จะเป็นอิสระเชิงเส้นกัน (linearly independent) ถ้า

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_k\mathbf{v}_k = \mathbf{0}$$

ก็ต่อเมื่อ ทุก ๆ ค่าสัมประสิทธิ์  $a_1, \dots, a_k$  ต้องเป็นศูนย์ทั้งหมดเท่านั้น.

นอกจากความสัมพันธ์เชิงเส้นระหว่างเวกเตอร์แล้ว ถ้ามีสเกลาร์  $a_1, a_2, \dots, a_k$  ที่ทำให้  $\mathbf{v} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_k\mathbf{v}_k$  แล้วจะเรียกว่า  $\mathbf{v}$  เป็นผลรวมเชิงเส้น ของ  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ . การที่เราได้ของเมทริกซ์ เป็นผลรวมเชิงเส้นของถาวร อีก ก็จะทำให้เมทริกซ์เป็นเอกฐาน.

**ดีเทอร์มิแนนต์.** ลักษณะเฉพาะที่สำคัญอย่างหนึ่งของเมทริกซ์จตุรัส คือ **ดีเทอร์มิแนนต์** (determinant). ค่าของดีเทอร์มิแนนต์สามารถช่วยบอกได้ว่าเมทริกซ์จตุรัสนั้นเป็นเอกฐานหรือไม่.

ดีเทอร์มิแนนต์ของเมทริกซ์จตุรัส  $\mathbf{A}$  ขนาดมิติ  $n \times n$  เป็นค่าสเกลาร์ และใช้สัญลักษณ์ เช่น  $|\mathbf{A}|$  หรือ  $\det \mathbf{A}$  โดย ดีเทอร์มิแนนต์ มีคุณสมบัติ

- ดีเทอร์มิแนนต์ ของเมทริกซ์  $\mathbf{A}$  เป็นฟังก์ชันเชิงเส้นของแต่ละสอดคล้อง  $\mathbf{A}$ . นั่นคือ ถ้า  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  เมื่อ  $\mathbf{a}_i$  เป็นสอดคล้องที่  $i^{th}$  ของเมทริกซ์ แล้ว

$$\begin{aligned} \det [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \alpha\mathbf{a}_k + \beta\mathbf{v}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \\ = \alpha \cdot \det [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{a}_k, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \\ + \beta \cdot \det [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{v}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \end{aligned} \tag{2.17}$$

เมื่อ  $\alpha$  และ  $\beta$  เป็นจำนวนจริงใด ๆ และ  $\mathbf{v} \in \mathbb{R}^n$  เป็นเวกเตอร์ที่มีส่วนประกอบเป็น  $n$  ค่าจริงใด ๆ

2. ถ้ามีสอดคล้องกันค่าเดียวกัน ค่าดีเทอร์มิแนนต์จะเป็นศูนย์. นั่นคือ ถ้ามี  $\mathbf{a}_i = \mathbf{a}_j$  โดย  $i \neq j$  แล้ว

$$\det [\mathbf{a}_1, \dots, \mathbf{a}_i, \dots, \mathbf{a}_j, \dots, \mathbf{a}_n] = \det [\mathbf{a}_1, \dots, \mathbf{a}_i, \dots, \mathbf{a}_i, \dots, \mathbf{a}_n] = 0 \quad (2.18)$$

3. ค่าดีเทอร์มิแนนต์ของเมทริกซ์เอกลักษณ์ เป็นหนึ่ง. นั่นคือ

$$\det \mathbf{I} = 1 \quad (2.19)$$

จากคุณสมบัติพื้นฐานทั่วสาม ทฤษฎีการขยายบวกจัจย์รวม (cofactor expansion theorem) ถูกพัฒนาขึ้น และ ค่าดีเทอร์มิแนนต์ของเมทริกซ์  $\mathbf{A}$  ขนาดมิติ  $n \times n$  สามารถคำนวณได้จาก

$$\begin{aligned} \det \mathbf{A} &= A_{k1} \cdot C_{k1} + A_{k2} \cdot C_{k2} + \dots + A_{kn} \cdot C_{kn} \\ &= A_{1k} \cdot C_{1k} + A_{2k} \cdot C_{2k} + \dots + A_{nk} \cdot C_{nk} \\ &= \sum_j A_{kj} \cdot C_{kj} = \sum_j A_{ik} \cdot C_{ik} \end{aligned} \quad (2.20)$$

เมื่อ  $k$  เป็นดัชนีที่เลือกตรึงให้คงที่ ( $k \in \{1, \dots, n\}$ ) และ บจจย์รวม (cofactor)  $C_{ij} = (-1)^{i+j} \cdot M_{ij}$  โดย  $M_{ij}$  เป็นค่าดีเทอร์มิแนนต์ของเมทริกซ์ย่อยจาก  $\mathbf{A}$  โดยการตัดแกร่งที่  $i$  และตัดสอดคล้องที่  $j$  ออก. ตัวอย่าง

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 3 \\ 8 & 2 & 7 \\ 0 & 5 & 9 \end{bmatrix} \text{ เมื่อเลือกตรึงแล้ว } k = 1 \text{ จะได้ } \det \mathbf{A} = 1 \cdot \begin{vmatrix} 2 & 7 \\ 5 & 9 \end{vmatrix} - 4 \cdot \begin{vmatrix} 8 & 7 \\ 0 & 9 \end{vmatrix} + 3 \cdot \begin{vmatrix} 8 & 2 \\ 0 & 5 \end{vmatrix}$$

และ

$$\begin{vmatrix} 2 & 7 \\ 5 & 9 \end{vmatrix} = 2 \cdot |9| - 7 \cdot |5| = -17. \quad \begin{vmatrix} 8 & 7 \\ 0 & 9 \end{vmatrix} = 8 \cdot |9| - 7 \cdot |0| = 72. \quad \begin{vmatrix} 8 & 2 \\ 0 & 5 \end{vmatrix} = 8 \cdot |5| - 2 \cdot |0| = 40.$$

ดังนั้น  $\det \mathbf{A} = -17 - 4 \cdot 72 + 3 \cdot 40 = -185.$

คุณสมบัติที่ตามมาของค่าดีเทอร์มิแนนต์ มีหลายอย่าง แต่ที่สำคัญ คือ  $\det \mathbf{A} = \det \mathbf{A}^T$  ดังนั้น การที่ระบบสมการมีสมการที่ไม่เป็นอิสระเชิงเส้นต่อกัน (แกร่งของเมทริกซ์ ไม่เป็นอิสระเชิงเส้นต่อกัน) จะทำให้ เมทริกซ์สัมประสิทธิ์ มีค่าดีเทอร์มิแนนต์เป็นศูนย์ ซึ่งหมายถึง เมทริกซ์เป็นเอกภูต.

**นอร์ม.** ดีเทอร์มิแนต์ บอกคุณสมบัติที่สำคัญของเมตริกซ์จัตุรัส. เวกเตอร์ก็ต้องการค่าบวกคุณสมบัติ. นอร์ม (norm) เป็นการวัดขนาดของเวกเตอร์. จำนวนส่วนประกอบของเวกเตอร์ อาจจะบอกขนาดมิติปริภูมิของเวกเตอร์ แต่นอร์มจะบอกขนาดของเวกเตอร์ และเป็นคุณสมบัติที่สำคัญอันหนึ่งของเวกเตอร์ สามารถใช้เปรียบเทียบสองเวกเตอร์ที่อยู่ปริภูมิค่าเดียวกัน (เวกเตอร์ที่อยู่ปริภูมิค่าเดียวกัน จะมีจำนวนส่วนประกอบเท่ากัน แต่อาจมีขนาดไม่เท่ากัน). นอร์มอาจวัดได้หลายวิธี แต่วิธีที่นิยม คือ  $L^p$  นอร์ม โดย  $L^p$  นอร์ม กำหนดเป็น

$$\|\mathbf{x}\|_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}} \quad (2.21)$$

เมื่อ  $p \in \mathbb{R}, p \geq 1$ .

งานส่วนใหญ่มักเลือกใช้  $L^2$  นอร์ม ( $p = 2$ ) หรือเรียกว่า ยูคลีเดียนนอร์ม (Euclidean norm) หรือ ระยะทางยูคลีเดียน (Euclidean distance) ที่ใช้สัญกรณ์  $\|\mathbf{x}\|_2$ . ความนิยมของ  $L^2$  นอร์ม ทำให้ปอยครัง สัญกรณ์อาจจะตัวห้อยออก เป็น  $\|\mathbf{x}\|$ . นอกจากนั้น บางครั้ง เพื่อความสะดวกและลดการคำนวน ค่า  $L^2$  นอร์ม กำลังสอง ได้แก่  $\|\mathbf{x}\|_2^2 = \sum_i x_i^2 = \mathbf{x}^T \cdot \mathbf{x}$  ก็นิยมใช้วัดขนาดเวกเตอร์

ตัวอย่างเช่น  $\mathbf{v} = [0.5, -8, 11]^T$  จะมี  $L^2$  นอร์มเป็น  $\|\mathbf{v}\| = \sqrt{\sum_i v_i^2} = \sqrt{0.25 + 64 + 121} = 13.61$ . สังเกตว่า ส่วนประกอบที่มีค่าใกล้ ๆ ศูนย์ จะมีขนาดผลเล็กลง เมื่อคำนวนด้วย  $L^2$  นอร์ม (สังเกต ส่วนประกอบ 0.5 มีขนาดผลลดลงเป็น 0.25) ดังนั้น สำหรับบางภาระกิจที่ไม่ต้องการพุตติกรรม เช่นนี้ แต่ต้องการ พุตติกรรมที่ผลของส่วนประกอบมีอัตราส่วนคงที่ ซึ่งจะช่วยให้เห็นความแตกต่างระหว่างศูนย์ กับค่าที่ไม่เป็นศูนย์ได้ดีกว่า ภาระกิจเหล่านั้นอาจเลือกใช้  $L^1$  นอร์ม ตัวอย่างเช่น  $\|\mathbf{v}\|_1 = \sum_i |v_i| = 0.5 + 8 + 11 = 19.5$ .

เวกเตอร์ใดก็ตามที่มี  $L^2$  นอร์มเป็นหนึ่ง จะเรียกว่า เวกเตอร์หนึ่งหน่วย (unit vector) นั่นคือ ถ้า  $\|\mathbf{u}\|_2 = 1$  จะเรียกว่า  $\mathbf{u}$  เป็นเวกเตอร์หนึ่งหน่วย. ตัวอย่าง เช่น เวกเตอร์  $\mathbf{v}_1 = [1, 0, 0]^T$  เวกเตอร์  $\mathbf{v}_2 = [0, 0, 1]^T$  และเวกเตอร์  $\mathbf{v}_3 = [0.7428, 0.5571, -0.3714]^T$  เป็นเวกเตอร์หนึ่งหน่วย. แต่เวกเตอร์  $\mathbf{v}_4 = [1, 1, 0]^T$  (ขนาดเป็น 1.414) และเวกเตอร์  $\mathbf{v}_5 = [0.7, 0.2, 0.1]^T$  (ขนาดเป็น 0.735) ไม่ใช่เวกเตอร์หนึ่งหน่วย.

เวกเตอร์ใด ๆ  $\mathbf{v} \in \mathbb{R}^n$  จะมีเวกเตอร์หนึ่งหน่วย  $\mathbf{u} \in \mathbb{R}^n$  ที่ซึ่งไปทิศทางเดียวกับมัน และ

$$\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|} \quad (2.22)$$

จากตัวอย่างข้างต้น เวกเตอร์  $\mathbf{v}_4 = [1, 1, 0]^T$  มีเวกเตอร์หนึ่งหน่วย  $[0.7071, 0.7071, 0]^T$  ที่ซึ่งไปทาง

เดียวกับมัน และเวกเตอร์  $\mathbf{v}_5 = [0.7, 0.2, 0.1]^T$  มีเวกเตอร์หนึ่งหน่วย  $[0.9526, 0.2722, 0.1361]^T$  ที่ซึ่งไปทางเดียวกับมัน.

## ภาคฉายเชิงตั้งฉาก

การเปลี่ยนมุมมอง รวมถึงการแยกปัจจัยประกอบออก เป็นหนึ่งในวิธีที่ช่วยในการทำความเข้าใจเรื่องราวต่าง ๆ รวมถึงการทำความเข้าใจกลุ่มของตัวแปรต่าง ๆ และทำความเข้าใจข้อมูล. การทำภาคฉายเชิงตั้งฉาก เป็นเสมือนการเปลี่ยนมุมมองวิธีหนึ่ง

รูป 2.1 แสดงตัวอย่างการทำภาคฉายเชิงตั้งฉาก. เวกเตอร์  $\mathbf{x} = [3, 4]^T$  เมื่อฉายลงบนทิศทางของของเวกเตอร์  $\mathbf{u} = [1, 0]^T$  แล้ว จะมีขนาดเป็น 3 บนทิศทางของ  $\mathbf{u}$  และเมื่อฉายลงบน  $\mathbf{v} = [0, 1]^T$  จะมีขนาดเป็น 4 บนทิศทางของ  $\mathbf{v}$ . ขนาดของการฉายเวกเตอร์ได้  $\mathbf{z}$  ลงบนทิศทางของเวกเตอร์หนึ่งหน่วย  $\mathbf{u}$  จะคำนวณได้จาก

$$z_u = \mathbf{z}^T \mathbf{u} \quad (2.23)$$

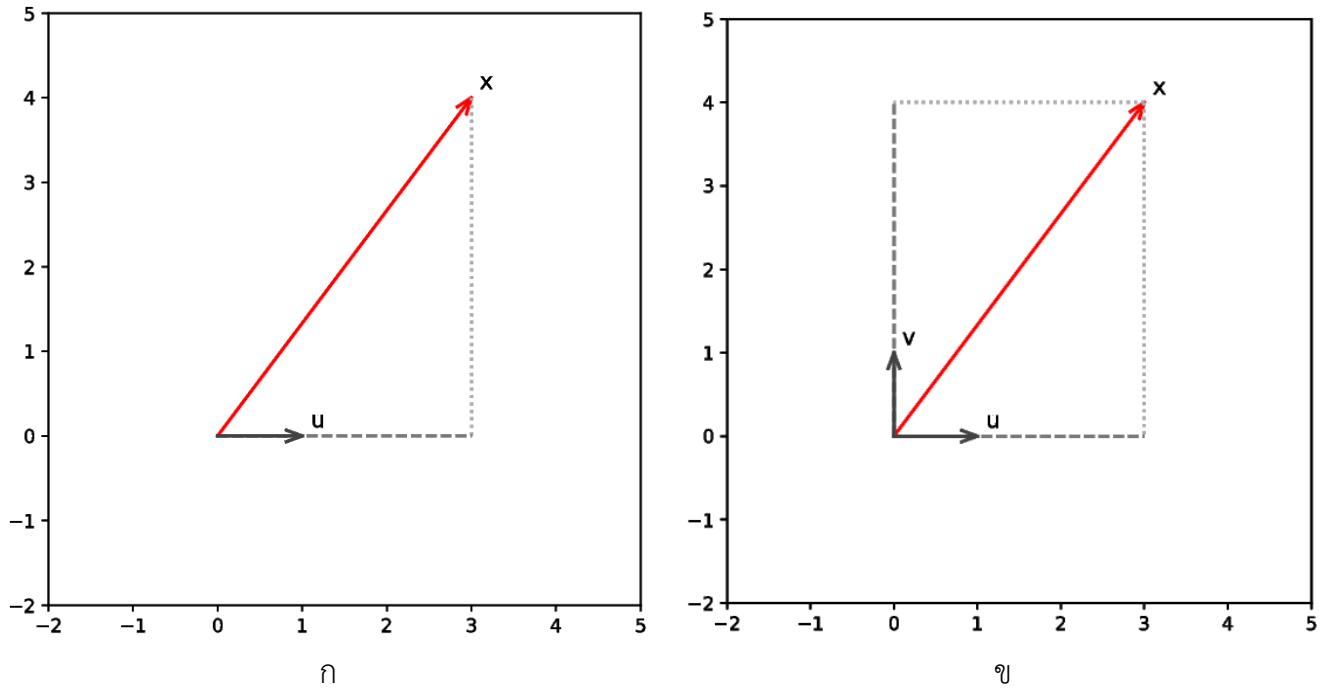
เมื่อ  $\mathbf{z}_u$  เป็นขนาดของเวกเตอร์  $\mathbf{z}$  ที่ฉายลงบนทิศทางของเวกเตอร์หนึ่งหน่วย  $\mathbf{u}$ .

รูป 2.2 แสดงตัวอย่างการทำภาคฉายเชิงตั้งฉากของเวกเตอร์เดิม  $\mathbf{x} = [3, 4]^T$  แต่ฉายลงบนเวกเตอร์  $\mathbf{u} = [0.9578, 0.2873]^T$  และเวกเตอร์  $\mathbf{v} = [-0.2873, 0.9578]^T$ . ดังนั้น ขนาดของ  $\mathbf{x}$  ที่ฉายลงบน  $\mathbf{u}$  จะเป็น  $\mathbf{x}^T \mathbf{u} = 4.023$  และขนาดที่ฉายลงบน  $\mathbf{u}$  จะเป็น  $\mathbf{x}^T \mathbf{u} = 2.969$ . สังเกตว่า ในตัวอย่างนี้ เวกเตอร์  $\mathbf{v}$  ตั้งฉากกับเวกเตอร์  $\mathbf{u}$  และเวกเตอร์ที่ตั้งฉากกัน (orthogonal vectors) จะมีผลคูณเวกเตอร์เป็นศูนย์ หรือ  $\mathbf{v}^T \mathbf{u} = 0$ .

การแปลงเชิงเส้น (linear transformation) ก็คือการคูณตัวแปรที่ต้องการแปลง ด้วยเมตริกซ์แปลง (transformation matrix). นั่นคือ การแปลงเชิงเส้น  $T(\mathbf{x}) = \mathbf{Ax}$ . การหาเวกเตอร์ตั้งฉาก อาจมองเป็นการแปลงเวกเตอร์ไปเป็นเวกเตอร์ที่ตั้งฉากกับต้นฉบับ ซึ่งตัวอย่างข้างต้นนี้ใช้เมตริกซ์แปลง

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

และจากตัวอย่าง  $\mathbf{u} = [0.9578, 0.2873]^T$  ซึ่งเมื่อทำการแปลงเชิงเส้นด้วย  $\mathbf{Au} = [-0.2873, 0.9578]^T$  ผลลัพธ์คือเวกเตอร์ที่ตั้งฉากกับเวกเตอร์ต้นฉบับ.



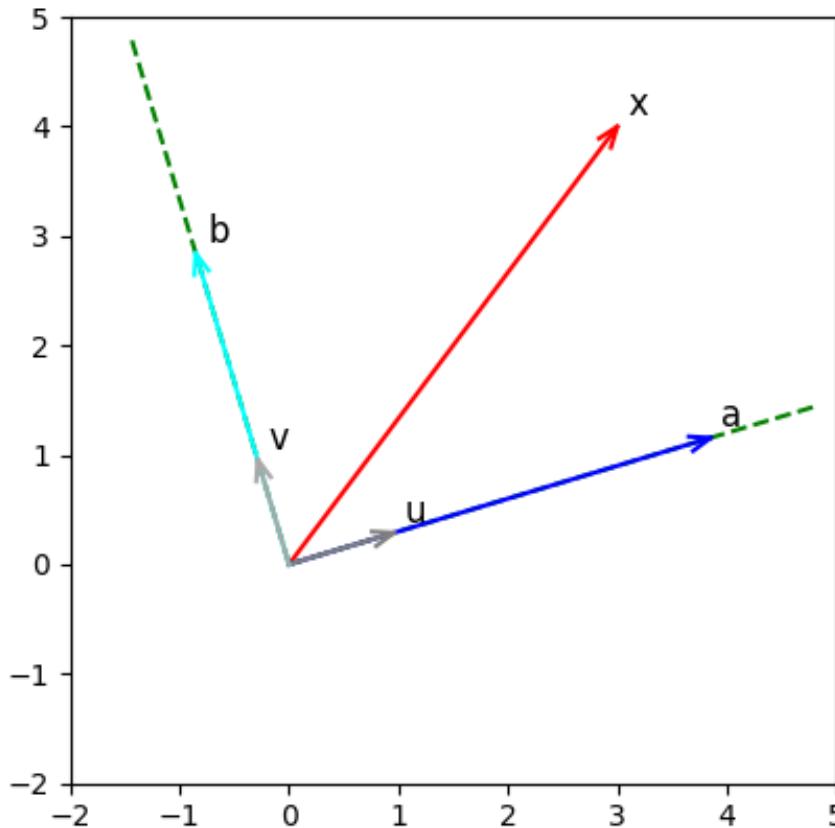
**รูปที่ 2.1:** การฉายภาพ. (ก) การฉายภาพของเวกเตอร์  $\mathbf{x} = [a, b]^T$  ลงบนเวกเตอร์  $\mathbf{u} = [1, 0]^T$  คือ การฉายภาพจากเวกเตอร์  $\mathbf{x}$  ที่สินใจ ในแนวตั้งฉากไปลงบน แนวของกับเวกเตอร์ที่ฉายลง และ (ข) การฉายภาพของเวกเตอร์  $\mathbf{x} = [a, b]^T$  ลงบน เวกเตอร์  $\mathbf{u} = [1, 0]^T$  กับเวกเตอร์  $\mathbf{v} = [0, 1]^T$  โดยมีขนาดเป็น  $a$  เมื่อฉายบนเวกเตอร์  $\mathbf{u}$  และมีขนาดเป็น  $b$  เมื่อฉายบนเวกเตอร์  $\mathbf{v}$ . ทั้ง  $\mathbf{u}$  และ  $\mathbf{v}$  เป็นเวกเตอร์หนึ่งหน่วย.

เวกเตอร์  $\mathbf{x} \in \mathbb{R}^n$  เรียกว่า  $\mathbf{x}$  ออยู่ในปริภูมิ  $\mathbb{R}^n$  นั่นคือ ถ้า  $\mathbb{R}^n$  เป็นเส้นมีอนแม่นที่ เวกเตอร์  $\mathbf{x}$  ได ๆ ที่อยู่ใน ปริภูมิ ก็เปรียบเสมือนเป็นจุด ๆ หนึ่งบนแม่นที่นั้น. จุดใด ๆ ในปริภูมิ  $\mathbb{R}^n$  จะสามารถอ้างอิงถึงได้โดยใช้ตัวเลข  $n$  ตัว (นั่นคือ  $n$  มิติปริภูมิค่า)

**ปริภูมิย่อย.** เซตย่อย (subset)  $\rho$  ของ  $\mathbb{R}^n$  จะเป็นปริภูมิย่อย (subspace) ของ  $\mathbb{R}^n$  เมื่อ  $\rho$  เป็นเซตปิด (closed set) ภายใต้การดำเนินการบวกของเวกเตอร์ และการคูณกับสเกลาร์. นั่นคือ ถ้า  $\mathbf{a}$  และ  $\mathbf{b}$  เป็นเวกเตอร์ในปริภูมิย่อย  $\rho$  และ  $\mathbf{a} + \mathbf{b}$  และ  $\alpha\mathbf{a}$  ก็ต้องอยู่ในปริภูมิย่อย  $\rho$  สำหรับทุก ๆ ค่า  $\alpha$ .

ตัวอย่าง เช่น จากรูป 2.2 เวกเตอร์  $\mathbf{x}$  ออยู่ใน  $\mathbb{R}^2$  และเวกเตอร์  $\mathbf{u}$  และเวกเตอร์  $\mathbf{v}$  และเวกเตอร์  $\mathbf{a}$  และ เวกเตอร์  $\mathbf{b}$  ก็อยู่ในปริภูมิ  $\mathbb{R}^2$  ด้วย. สมมติให้  $\rho_1$  เป็นปริภูมิย่อยของ  $\mathbb{R}^2$  โดยมี เวกเตอร์  $\mathbf{u}$  ออยู่ใน  $\rho_1$  แต่ ไม่มีเวกเตอร์  $\mathbf{v}$  ในปริภูมิย่อย  $\rho_1$ . นั่นคือ ปริภูมิย่อย  $\rho_1$  เป็นเซตปิด สำหรับ เวกเตอร์ในแนว  $\mathbf{u}$  เช่น  $\mathbf{u}$  และ  $\mathbf{a}$  ออยู่ใน  $\rho_1$ .

มองจากอีกมุมหนึ่ง ถ้ากำหนดให้  $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$  เป็นเซตของเวกเตอร์ได ๆ ในปริภูมิ  $\mathbb{R}^n$  แล้ว ผลรวมเชิงเส้นทุกแบบของเวกเตอร์เหล่านี้ จะเป็นเซตของเวกเตอร์ ที่เรียกว่า **การແພທ້ວ (span)** ของเวกเตอร์



รูปที่ 2.2: เวกเตอร์  $\mathbf{x} = [3, 4]^T$  ฉายลงบน เวกเตอร์หนึ่งหน่วย  $\mathbf{u} = [0.9578, 0.2873]^T$  และ  $\mathbf{v} = [-0.2873, 0.9578]^T$  โดย  $\mathbf{a}$  เป็นเวกเตอร์ตามทิศทาง  $\mathbf{u}$  แต่มีขนาดเท่ากับที่เวกเตอร์  $\mathbf{x}$  ฉายลงบน  $\mathbf{u}$  และทำนองเดียวกัน  $\mathbf{b}$  ก็เป็นเวกเตอร์ที่ได้จากการฉาย  $\mathbf{x}$  ลงบน  $\mathbf{v}$ . นั่นคือ  $\mathbf{a} = 4.023\mathbf{u} = [3.853, 1.156]^T$  และ  $\mathbf{b} = 2.969\mathbf{v} = [-0.853, 2.844]^T$ .

$\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$  และใช้สัญกรณ์<sup>3</sup>

$$\text{span}[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k] = \left\{ \sum_{i=1}^k \alpha_i \mathbf{c}_i : \alpha_1, \dots, \alpha_k \in \mathbb{R} \right\} \quad (2.24)$$

และ ปริภูมิย่อยคือ การແຜ່ทົ່ວຂອງເຊົາດຂອງເວກເຕັມ. ຕ້າຍ່າງເຊັ່ນ ປະລິມີຢ່າຍ  $\rho_1$  ໃນຕ້າຍ່າງຂ້າງຕັ້ນ ກີ່ຄືການແຜ່ທົ່ວຂອງ  $\{\mathbf{u}\}$  ແລະ ປະລິມີ  $\mathbb{R}^2$  ກີ່ຄືການແຜ່ທົ່ວຂອງ  $\{\mathbf{u}, \mathbf{v}\}$ .

ສັງເກດວ່າໃນຂະໜາດ  $\mathbb{R}^2$  ການອ້າງຄືການແຜ່ທົ່ວຂອງເຊົາດປະລິມີ  $\mathbb{R}^2$  ຕ້ອງການຕັ້ງເລີກ 2 ຕ້າ (ສອນມິຕີປະລິມີຄ່າ) ແຕ່ການອ້າງຄືການແຜ່ທົ່ວຂອງປະລິມີຢ່າຍ  $\rho_1$  ຕ້ອງການຕັ້ງເລີກແຕ່ຕັ້ງເດືອນ (ປະລິມີຢ່າຍ ມີໜຶ່ງມິຕີປະລິມີຄ່າ). ຈຳນວນມິຕີປະລິມີຄ່າຂອງປະລິມີໄດ້ 1 (ຮວມຄືການປະລິມີຢ່າຍໄດ້ 1) ຈະເທົ່າກັບຈຳນວນຂອງເວກເຕັມທີ່ເປັນອີສະເໜີເສັ້ນກັນ ທີ່ອູ້ໃນປະລິມີ. ຕ້າຍ່າງເຊັ່ນ ປະລິມີ  $\mathbb{R}^2$  ມີສອນເວກເຕັມທີ່ເປັນອີສະເໜີເສັ້ນ ໄນວ່າຈະເລືອກ  $\mathbf{u}$  ກັບ  $\mathbf{v}$  ຮຸ່ວໂມງເລືອກຊຸດ  $[1, 0]^T$  ກັບ

<sup>3</sup>ສัญกรณ์ເຫັນ  $\{a_i : c(a_i)\}$  ສໍາຫັບ  $i = 1, \dots, k$  ພາຍໃຕ້ ເຊົາດທີ່ມີສາມາດເປັນ  $a_i$  ຢ່າງ  $a_i$  ຜ່ານເຈື່ອນໄຂທີ່ຮະບຸລັງເຄື່ອງໝາຍຈຸດຄຸ່ງ. ນັ້ນຄືການ  $a_i$  ຜ່ານເຈື່ອນໄຂ  $c(a_i)$  ແລ້ວ ອ່ານ  $a_i$  ຈະເປັນສາມາດຂອງເຊົາດ ແຕ່ ຢ່າງ  $a_i$  ໄນຜ່ານເຈື່ອນໄຂ  $c(a_i)$  ແລ້ວ ອ່ານ  $a_i$  ຈະໄນ້ເປັນສາມາດຂອງເຊົາດ ໂດຍພິຈາລະນາ  $a_i$  ສໍາຫັບ  $i = 1, \dots, k$ .

$[1, 1]^T$  หรือเลือกชุด  $[1, -1]^T$  กับ  $[0, 1]^T$  หรือชุดอื่น ๆ ของเวกเตอร์สองตัวที่ตั้งฉากกัน. ดังนั้น  $\mathbb{R}^2$  จึงมีสองมิติ(ปริภูมิค่า). ปริภูมิ  $\rho_1$  มีแค่เวกเตอร์เดียวที่เป็นอิสระเชิงเส้น ดังนั้น  $\rho_1$  จึงมีหนึ่งมิติ(ปริภูมิค่า).

**การแยกส่วนประกอบเชิงตั้งฉาก.** ถ้ากำหนด  $\rho$  เป็นปริภูมิย่อยของ  $\mathbb{R}^n$  และกำหนดให้  $\rho^\perp$  เป็นส่วนเติมเต็มเชิงตั้งฉาก (orthogonal complement) ของ  $\rho$  โดย  $\rho^\perp$  ประกอบด้วยเวกเตอร์ทั้งหมด ที่แต่ละเวกเตอร์ตั้งฉากกับทุกเวกเตอร์ใน  $\rho$ . นั่นคือ

$$\rho^\perp = \{\mathbf{x} : \mathbf{x}^T \mathbf{v} = 0 \text{ for all } \mathbf{v} \in \rho\} \quad (2.25)$$

เซต  $\rho^\perp$  เองก็เป็นปริภูมิย่อยของ  $\mathbb{R}^n$  ด้วย. เวกเตอร์ทั้งหมดจากทั้งสองเซต  $\rho$  และ  $\rho^\perp$  แผ่ทั่ว  $\mathbb{R}^n$ . นั่นคือ เวกเตอร์ใด ๆ  $\mathbf{x} \in \mathbb{R}^n$  สามารถแยกส่วนประกอบออกได้

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$$

เมื่อ  $\mathbf{x}_1 \in \rho$  และ  $\mathbf{x}_2 \in \rho^\perp$ . การแยกส่วนประกอบแบบนี้ จะเรียกว่า การแยกส่วนประกอบเชิงตั้งฉาก (orthogonal decomposition) ของ  $\mathbf{x}$  ตาม  $\rho$  และ  $\mathbf{x}_1$  และ  $\mathbf{x}_2$  เป็นภาพฉายตั้งฉาก (orthogonal projections) ของ  $\mathbf{x}$  ลงบนปริภูมิย่อย  $\rho$  และ  $\rho^\perp$ .

จากตัวอย่างในรูป 2.2 จะเห็นว่า

$$\begin{aligned} \mathbf{x} &= [3, 4]^T = \mathbf{a} + \mathbf{b} \\ &= 4.023\mathbf{u} + 2.969\mathbf{v} \end{aligned}$$

หรือ  $\mathbf{x}$  ถูกแยกออก เป็น ส่วนประกอบแรก 4.023 และส่วนประกอบที่สอง 2.969. หัวข้อ ?? อภิรายวิธีการวิเคราะห์ส่วนประกอบหลัก (Principal Component Analysis) ซึ่งเป็นวิธีที่ใช้แนวคิดหลัก จากการฉายภาพเชิงตั้งฉาก เพื่อลดมิติปริภูมิของข้อมูล ซึ่งจะช่วยการประมวลผล และช่วยการนำเสนอภาพของข้อมูลได้

## เวกเตอร์ลักษณะเฉพาะและค่าลักษณะเฉพาะ

การแปลงข้อมูลเชิงเส้นมีการใช้อย่างกว้างขวาง และเครื่องมือสำคัญที่นิยมใช้ในการวิเคราะห์การแปลงข้อมูล เชิงเส้น คือ เวกเตอร์ลักษณะเฉพาะ และ ค่าลักษณะเฉพาะ.

กำหนดให้  $\mathbf{A}$  เป็น  $n \times n$  เมทริกซ์. ค่าสเกลาร์  $\lambda$  และเวกเตอร์  $\mathbf{v}$  โดย  $\mathbf{v} \neq \mathbf{0}$  จะเรียกว่า ค่าลักษณะเฉพาะ (eigenvalue) และเวกเตอร์ลักษณะเฉพาะ (eigenvector) ของ  $\mathbf{A}$  เมื่อ

$$\mathbf{Av} = \lambda \mathbf{v} \quad (2.26)$$

ตัวอย่าง เช่น

$$\mathbf{A} = \begin{bmatrix} 9.8 & 3.6 \\ 3.6 & 2.5 \end{bmatrix}$$

มีค่าลักษณะเฉพาะค่าหนึ่ง คือ  $\lambda_1 = 11.277$  และมีเวกเตอร์ลักษณะเฉพาะที่คู่กัน  $\mathbf{v}_1 = [-0.925, -0.379]^T$ .

เมื่อตรวจสอบ จะพบว่าทางซ้ายมือ  $(\mathbf{A}\mathbf{v})$  จะได้

$$\begin{bmatrix} 9.8 & 3.6 \\ 3.6 & 2.5 \end{bmatrix} \cdot \begin{bmatrix} -0.925 \\ -0.379 \end{bmatrix} = \begin{bmatrix} -10.433 \\ -4.279 \end{bmatrix}$$

ซึ่ง เท่ากับทางขวา มือ คือ  $\lambda_1 \cdot \mathbf{v}_1$ . ดังนั้น  $\lambda_1$  และ  $\mathbf{v}_1$  คือ ค่าลักษณะเฉพาะ และเวกเตอร์ลักษณะเฉพาะ ของ  $\mathbf{A}$ .

การหาค่าลักษณะเฉพาะและเวกเตอร์ลักษณะเฉพาะ. จาก  $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$  ดังนั้น  $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$  เมื่อ  $\mathbf{I}$  เป็นเมตริกซ์เอกลักษณ์. กรณีที่  $\mathbf{v} = \mathbf{0}$  เป็นกรณีที่ชัดเจนแต่ไม่น่าสนใจ (trivial). พิจารณากรณีที่  $\mathbf{v} \neq \mathbf{0}$  สำหรับกรณีนี้  $\det[\mathbf{A} - \lambda\mathbf{I}] = 0$  ( เพราะว่า ถ้าดีเทอร์มิแนต์นี้ไม่เท่ากับศูนย์ จะแก้สมการได้เป็น  $\mathbf{v} = [\mathbf{A} - \lambda\mathbf{I}]^{-1}\mathbf{0} = \mathbf{0}$  ซึ่งจะเป็นกรณีแรก ที่ชัดเจนแต่ไม่น่าสนใจ)

จาก  $\det[\mathbf{A} - \lambda\mathbf{I}] = 0$  แทนค่าและแก้สมการหาค่า  $\lambda$  แล้วใช้ค่า  $\lambda$  ที่ได้แทนค่าลงใน  $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$  และแก้สมการหาค่าของเวกเตอร์  $\mathbf{v}$ .

เช่น จากตัวอย่างข้างต้น

$$\begin{aligned} \det \begin{bmatrix} 9.8 - \lambda & 3.6 \\ 3.6 & 2.5 - \lambda \end{bmatrix} &= 0 \\ (3.6)(3.6) - (9.8 - \lambda)(2.5 - \lambda) &= 0 \\ \lambda^2 - 12.3\lambda + 11.54 &= 0 \end{aligned}$$

จะได้เป็นสมการพหุนาม (polynomial equation) ของตัวแปร  $\lambda$  ที่ในตัวอย่างนี้เป็นสมการกำลังสอง (quadratic equation) ซึ่งมีรูปทั่วไปคือ  $ax^2 + bx + c = 0$  และผลเฉลยของสมการ มีสองค่าคือ  $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$  และ  $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$  และเมื่อแทนค่า  $a = 1, b = -12.3, c = 11.54$  แล้วจะได้ค่าลักษณะเฉพาะสองค่าคือ  $\lambda_1 = 11.28$  และ  $\lambda_2 = 1.02$ . นำค่าลักษณะเฉพาะแต่ละค่า ไปหาเวกเตอร์ที่คู่กัน ได้แก่ สำหรับ

$\lambda_1 = 11.28$  คำนวณหาเวกเตอร์ลักษณะเฉพาะ  $\mathbf{v}_1$  จาก

$$\underbrace{\begin{bmatrix} 9.8 & 3.6 \\ 3.6 & 2.5 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{v}_1} = \underbrace{11.28}_{\lambda_1} \cdot \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{v}_1}$$

และจะได้สมการ

$$9.8a + 3.6b = 11.28a$$

$$3.6a + 2.5b = 11.28b$$

ซึ่งสองสมการนี้ไม่เป็นเชิงเส้นต่อ กัน (เพราะว่า นี่คือ กรณี  $\det[\mathbf{A} - \lambda \mathbf{I}] = 0$ ) ดังนั้น คำตอบที่ได้จะไม่ได้มีแค่หนึ่งเดียว. สมมติเลือกสมการแรกมา  $9.8a + 3.6b = 11.28a$  และแก้สมการจะได้  $b = 0.41a$  ดังนั้น เวกเตอร์ลักษณะเฉพาะคือ  $[1, 0.41]^T$  หรือ  $[2, 0.82]^T$  หรือ  $[3, 1.32]^T$  หรือ เวกเตอร์ใด ๆ ที่อยู่ในรูป  $\alpha[1, 0.41]^T$  เมื่อ  $\alpha$  เป็นจำนวนจริงใด ๆ ที่  $\alpha \neq 0$ . สังเกตว่า เวกเตอร์ทั้งหมดที่ได้จะซึ่งเป็นพิเศษทางเดียว กัน (หรือตรงกันข้ามก็ได้ แต่ไม่ซึ่งเป็นพิเศษทางอื่น) แต่ต่างขนาดกันเท่านั้น.

เพื่อความสะดวก เวกเตอร์ลักษณะเฉพาะมักจะเลือกให้เป็นเวกเตอร์ขนาดหนึ่งหน่วย. ดังนั้น จากเวกเตอร์  $[1, 0.41]^T$  จะได้เวกเตอร์หนึ่งหน่วยเป็น  $\frac{1}{\sqrt{1^2+0.41^2}}[1, 0.41]^T = [0.93, 0.38]^T$ . นั่นคือ ได้เวกเตอร์ลักษณะเฉพาะ  $\mathbf{v}_1 = [0.93, 0.38]^T$  คู่กับค่าลักษณะเฉพาะ  $\lambda_1 = 11.28$  และในทำนองเดียวกัน ได้ เวกเตอร์ลักษณะเฉพาะ  $\mathbf{v}_2 = [0.38, -0.93]^T$  คู่กับค่าลักษณะเฉพาะ  $\lambda_2 = 1.02$ . หมายเหตุ ถึงแม้ว่าจะเลือกเวกเตอร์ลักษณะเฉพาะ ให้เป็นเวกเตอร์หนึ่งหน่วยแล้วก็ตาม แต่เวกเตอร์ลักษณะเฉพาะก็ยังอาจเลือกได้ หลายแบบอยู่ เช่น  $\mathbf{v}_1 = [-0.93, -0.38]^T$  หรือ  $\mathbf{v}_2 = [-0.38, 0.93]^T$  ที่อยู่ในแนวเดียวกัน มีขนาด เป็นหนึ่งเหมือนกัน เพียงแต่มีทิศตรงกันข้าม.

## 2.2 ความน่าจะเป็น

ปัจจัยสำคัญเรื่องหนึ่ง โดยเฉพาะกับงานการรู้จำรูปแบบและการเรียนรู้ของเครื่อง คือความไม่แน่นอน (uncertainty). ความไม่แน่นอนอาจมาจากหลายสาเหตุ เช่น ความไม่เที่ยงของเครื่องมือ หรือวิธีการวัด หรือวิธีการเก็บข้อมูล หรืออาจมาจากการสัญญาณรบกวน หรือมาจากการขนาดของข้อมูลที่จำกัด หรือแม้แต่ธรรมชาติความหลากหลาย และความแปรปรวนของข้อมูลเอง. ทฤษฎีความน่าจะเป็น (probability theory) เป็นแนวทางหนึ่งที่ให้กรอบ

วิธีการสำหรับการวัด และการจัดการกับความไม่แน่นอน. ดังนั้น ทฤษฎีความน่าจะเป็น<sup>4</sup> จะเป็นพื้นฐานที่สำคัญสำหรับการเรียนรู้ของเครื่องและการรู้จำรูปแบบ.

## เซต

ทฤษฎีความน่าจะเป็นจะมองรูปแบบต่าง ๆ เป็นเหตุการณ์ (event) และทฤษฎีความน่าจะเป็นจะใช้เซตในการอธิบายความหมายของเหตุการณ์.

เซต (set) แทนกลุ่มของค่าต่าง ๆ ที่สนใจ. เซต อาจแสดงด้วยสัญกรณ์ เช่น  $\{24, 98, 16, 53\}$  ที่หมายถึง เซตที่มีสมาชิกสี่ตัว ได้แก่ ค่า 24 ค่า 98 ค่า 16 และค่า 53 โดยลำดับของสมาชิกที่ปรากฏในเซตไม่ได้มีความหมาย ซึ่งต่างจากลำดับของส่วนประกอบในเวกเตอร์ เมทริกซ์ หรือเทนเซอร์ ที่อภิปรายในหัวข้อ 2.1. นอกจากนั้นสมาชิกของเซต ก็ไม่ได้จำกัดเฉพาะตัวเลข ตัวอย่างเช่น  $\{\text{'ช'}, \text{'ท'}, \text{'ณ'}, \text{'ต'}, \text{'ด'}\}$  เป็นเซตของพยัญชนะห้าตัว. สัญกรณ์ เช่น  $a \in A$  ระบุว่า  $a$  เป็นสมาชิกของเซต  $A$ .

ถ้าเซต  $A$  มีสมาชิกที่ทุกตัว เป็นสมาชิกของเซต  $B$  แล้วจะเรียกว่า เซต  $A$  เป็นเซตย่อย (subset) ของเซต  $B$  และใช้สัญกรณ์  $A \subset B$ . นั่นคือ ถ้า  $A \subset B$  และ  $a \in A$  หมายถึง  $a \in B$  ด้วย แต่อาจมีสมาชิกของ  $B$  ที่ไม่ได้เป็นสมาชิกของ  $A$  ก็ได้. สัญกรณ์  $b \notin A$  หมายถึง ค่า  $b$  ไม่ได้เป็นสมาชิกของเซต  $A$ . สัญลักษณ์  $\emptyset$  จะใช้แทนเซตว่าง (empty set) หรือเซตที่ไม่มีสมาชิกอยู่. นั่นคือ  $\emptyset = \{\}$ . สัญลักษณ์  $\Omega$  นักใช้แทนเซตของค่าที่เป็นไปได้ทั้งหมด.

**การดำเนินการเซต.** อินเตอร์เซกชัน (intersection) แทนด้วยสัญกรณ์ เช่น  $A \cap B$  ซึ่งหมายถึง การดำเนินการที่ผลลัพธ์จะเป็นเซตที่มีสมาชิกที่เป็นทุกสมาชิกของ  $A$  และ  $B$  ที่มีค่าเหมือนกัน. นั่นคือ ถ้า  $a \in A$  และ  $b \in B$  และ  $a = b$  แล้ว  $a \in A \cap B$  และในทางกลับกัน ถ้า  $c \in A \cap B$  แล้ว  $c \in A$  และ  $c \in B$ .

ยูเนียน (union) แทนด้วยสัญกรณ์ เช่น  $A \cup B$  ซึ่งหมายถึง การดำเนินการที่ผลลัพธ์จะเป็นเซตที่มีสมาชิกทั้งหมดของ  $A$  และสมาชิกทั้งหมดของ  $B$ . นั่นคือ ถ้า  $a \in A$  และ  $a \in A \cup B$  และถ้า  $b \in B$  และ  $b \in A \cup B$  และในทางกลับกัน ถ้า  $c \in A \cup B$  แล้ว  $c \in A$  หรือ  $c \in B$  หรือทั้ง  $c \in A$  และ  $c \in B$ .

ผลต่างเซต (set difference) แทนด้วยสัญกรณ์ เช่น  $A \setminus B$  ซึ่งหมายถึง การดำเนินการที่ผลลัพธ์จะเป็นเซตที่สมาชิกทั้งหมด เป็นสมาชิกของ  $A$  แต่ไม่มีสักตัวที่เป็นสมาชิกของ  $B$ . นั่นคือ ถ้า  $c \in A \setminus B$  และ  $c \in A$  และ  $c \notin B$  และในทางกลับกัน ถ้า  $a \in A$  และ  $a \notin B$  และ  $a \in A \setminus B$ .

<sup>4</sup>เนื้อหาในส่วนนี้ได้รับอิทธิพลหลักจาก [15] [61] [133] และ [80].

ส่วนเติมเต็ม (complement) แทนด้วยสัญกรณ์ เช่น  $A^c$  ซึ่งหมายถึง เซตของค่าทั้งหมดที่เป็นไปได้ แต่ไม่ได้เป็นสมาชิกของ  $A$ . นั่นคือ  $A^c = \Omega \setminus A$ .

ตัวอย่างเช่น ถ้า  $A = \{1, 3, 8, 9, 12, 16, 20\}$  และ  $B = \{7, 8, 12, 20, 32\}$  แล้วจะได้  $A \cap B = \{8, 12, 20\}$  และ  $A \cup B = \{1, 3, 7, 8, 9, 12, 16, 20, 32\}$  และ  $A \setminus B = \{1, 3, 9, 16\}$  และ  $B \setminus A = \{7, 32\}$ .

## ความน่าจะเป็น

ความน่าจะเป็น (probability) เป็นค่าที่ใช้ประมาณโอกาสที่เหตุการณ์ที่สนใจจะเกิดขึ้น. ทฤษฎีความน่าจะเป็น พิจารณาเหตุการณ์ ในบริบทของผลลัพธ์ต่าง ๆ (outcomes) ทั้งหมดทุกแบบที่อาจเกิดขึ้นได้.

เซตของผลลัพธ์แบบต่าง ๆ ที่เป็นไปได้ทั้งหมด จะเรียกว่า ปริภูมิตัวอย่าง (sample space). เหตุการณ์ (event) คือเซตย่อยของปริภูมิตัวอย่าง หรือกล่าวง่าย ๆ เหตุการณ์ คือ กลุ่มของผลลัพธ์ที่เป็นไปได้ (ผลลัพธ์ที่กล่าว คือผลลัพธ์ในเรื่องที่สนใจ).

ความน่าจะเป็น อาจอธิบายง่าย ๆ จากตัวอย่าง<sup>5</sup> หากสมมติว่า การทดลองทำซ้ำ ๆ เป็นจำนวน  $N$  ครั้ง โดยให้สภาพแวดล้อมเหมือนเดิมมากที่สุด. กำหนดให้  $A$  เป็นเหตุการณ์ที่สนใจ โดย  $A$  อาจจะเกิดขึ้นหรือไม่เกิดก็ได้ ในแต่ละซ้ำ. หากสิ่งที่พบคือ เมื่อจำนวนทำซ้ำ  $N$  ใหญ่ขึ้น อัตราส่วนของจำนวนครั้งที่จะเกิด  $A$  ในแต่ละซ้ำ จะมีค่าเข้าใกล้ค่า ๆ หนึ่งมากขึ้น. ค่าที่อัตราส่วนเข้าใกล้มากขึ้นเมื่อจำนวนซ้ำมากขึ้น คือค่าความน่าจะเป็นของ  $A$ .

ขยายความคือ หากกำหนดให้  $N(A)$  แทนจำนวนครั้งที่จะเกิดเหตุการณ์  $A$  ในการทำซ้ำทั้งหมด  $N$  ครั้ง อัตราส่วน  $\frac{N(A)}{N}$  จะค่อย ๆ ลุ่เข้าสู่ค่าค่าหนึ่ง เมื่อ  $N$  เพิ่มขึ้น. ค่าที่อัตราส่วนลุ่เข้า จะเรียกว่า ความน่าจะเป็นที่เหตุการณ์  $A$  จะเกิดขึ้น. ค่าความน่าจะเป็นของเหตุการณ์  $A$  แทนด้วยสัญลักษณ์  $\Pr(A)$  และความน่าจะเป็นจะมีค่าอยู่ระหว่าง 0 กับ 1. นั่นคือ  $\Pr(A) \in [0, 1]$  โดย 0 หมายถึงเหตุการณ์นั้นไม่มีโอกาสเกิดขึ้นเลย และ 1 หมายถึงเหตุการณ์นั้นเกิดขึ้นอย่างแน่นอน.

ตัวอย่างเช่น ลังใส่ลูกบอลสีต่าง ๆ ดังแสดงในรูป 2.3 หากสุ่มหยิบลูกบอล ออกมากลาง 1 ลูก และกำหนดให้  $A$  เป็นเหตุการณ์ที่หยิบได้ลูกบอลสีเขียว. สมมติทำการทดลอง(สุ่มหยิบ)ซ้ำ  $N = 10$  ผลการจำลอง

<sup>5</sup> ตัวอย่างนี้อธิบายตามมุมมองแบบความถี่นิยม (frequentist). ความน่าจะเป็น อาจถูกมองได้ตามมุมมองแบบความถี่นิยม หรือตามมุมมองแบบเบส (bayesian). มุมมองแบบความถี่นิยม จำกัดความน่าจะเป็น เป็นความถี่จากการทำซ้ำดังที่อธิบายในตัวอย่าง. แต่มุมมองแบบเบส จะมองความน่าจะเป็น เป็นการประมาณโอกาสที่เหตุการณ์จะเกิด โดยเน้นที่รرمชาติของโอกาสเอง ไม่ได้มองเป็นความถี่จากการทำซ้ำ. มุมมองแบบเบส จะกว้างกว่า มุมมองแบบความถี่นิยมที่คำอธิบายจะเข้าได้ยากกับเหตุการณ์ที่ทำซ้ำไม่ได้ เช่น การประมาณอายุของดาว ที่ไม่เกี่ยว กับการทำซ้ำ แต่เป็นการประมาณ ที่เนื่องมีหลักฐาน หรือข้อมูลเข้ามาใหม่ ความน่าจะเป็นของอายุของดาว ที่สามารถจะถูกประเมินเป็นค่าใหม่ได้.

เหตุการณ์ (simulation) ดังแสดงในรูป 2.4 ชี้งบอกได้ว่า อัตราส่วนที่หยิบได้ลูกบอลสีเขียว เป็น  $\frac{N(A)}{N} = \frac{8}{10} = 0.8$ . หากเพิ่มจำนวนการทำซ้ำ  $N$  จาก 10 เป็น 100 และเพิ่มเป็น 1000 และเป็น 10000 และทำต่อ ๆ ไป จะเริ่มเห็นว่าอัตราส่วน  $\frac{N(A)}{N}$  ลู่เข้าสู่ค่าค่าหนึ่ง ดังแสดงในตาราง 2.2. เมื่อนำค่าต่าง ๆ ไปวาดกราฟ จะได้ดังรูป 2.5 ซึ่งจะเห็นว่าค่าที่ อัตราส่วน  $\frac{N(A)}{N}$  ลู่เข้าหาคือ 0.75. นั่นคือ ความน่าจะเป็นของการสุ่มหยิบได้ลูกสีเขียว  $\Pr(A) = 0.75$ . มองจากอีกมุมหนึ่ง ในลังมีลูกบอล 12 ลูก และเป็นลูกสีเขียวอยู่ 9 ลูก หากสุ่มหยิบอย่างยุติธรรม (แต่ละลูกมีโอกาสสูงเท่า ๆ กัน) โอกาสที่จะหยิบได้ลูกสีเขียวจะเป็น  $\frac{9}{12} = 0.75$  ซึ่งค่าที่คำนวนนี้สอดคล้องกับค่าความน่าจะเป็นที่ได้จากการจำลองเหตุการณ์ข้างต้น.

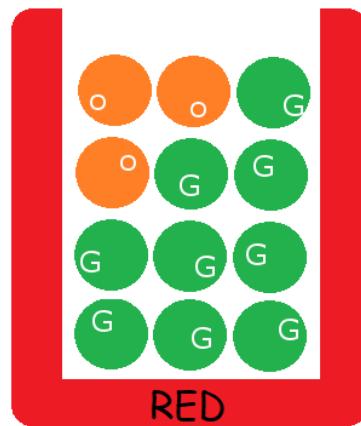
มุมมองของความน่าจะเป็น. เหตุการณ์อาจมองได้ในหลาย ๆ ระดับ เช่น เหตุการณ์ระดับล่าง ได้แก่ เหตุการณ์ที่หยิบได้ลูกบอลลูกที่หนึ่ง  $a_1$  (ซึ่งเป็นสีเขียว) ไปจนถึง เหตุการณ์ที่หยิบได้ลูกบอลลูกที่สิบสอง  $a_{12}$  (ซึ่งเป็นสีส้ม) หรือเหตุการณ์ระดับที่สูงขึ้น ได้แก่ เหตุการณ์ที่หยิบได้ลูกบอลสีเขียว  $A$  เหตุการณ์ที่หยิบได้ลูกบอลสีส้ม  $B$  เป็นต้น. ทฤษฎีความน่าจะเป็นมองเชตจากมุมมองที่ต่างกันไป ดังแสดงในตาราง 2.1.

ตารางที่ 2.1: ภาษาเฉพาะที่ใช้ในเรื่องเชตกับเรื่องความน่าจะเป็น

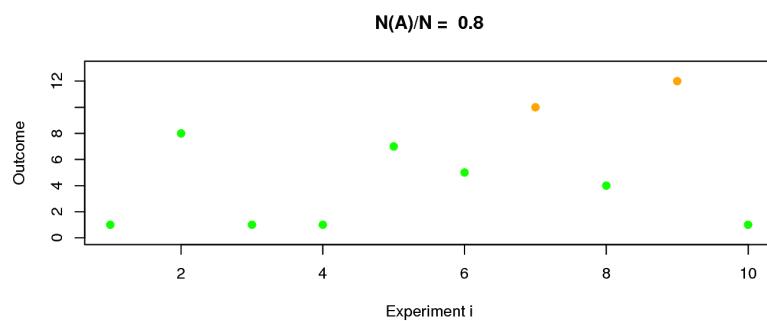
สัญลักษณ์ทั่วไป	ภาษาเฉพาะในเรื่องเชต	ภาษาเฉพาะในเรื่องความน่าจะเป็น
$\Omega$	กลุ่มของค่าที่เป็นไปได้ทั้งหมด	ปริภูมิตัวอย่าง หรือผลลัพธ์ทั้งหมดที่เป็นไปได้
$a \in \Omega$	ค่าหนึ่งที่เป็นไปได้	รูปแบบของผลลัพธ์ของเรื่องที่สนใจ หรือเหตุการณ์พื้นฐาน
$A \subset \Omega$	เซตย่อยของ $\Omega$	เหตุการณ์ที่ผลลัพธ์ใน $A$ เกิดขึ้น
$A^c$	ส่วนเติมเต็มของ $A$	เหตุการณ์ที่ผลลัพธ์ใน $A$ ไม่เกิด
$A \cap B$	อินเตอร์เซกชัน	เหตุการณ์ที่มีผลลัพธ์ทั้งใน $A$ และ $B$ เกิดขึ้น
$A \cup B$	ยูเนียน	เหตุการณ์ที่มีผลลัพธ์ใน $A$ หรือ $B$ หรือในทั้งคู่เกิดขึ้น
$A \setminus B$	ผลต่างเซต	เหตุการณ์ที่ผลลัพธ์ใน $A$ เกิดขึ้น แต่ผลลัพธ์ใน $B$ ไม่เกิด
$\emptyset$	เซตว่าง	เหตุการณ์ที่เป็นไปไม่ได้

ตารางที่ 2.2: อัตราส่วนของการสุ่มได้ลูกบอลสีเขียว เมื่อจำนวนการทำซ้ำเพิ่มขึ้น

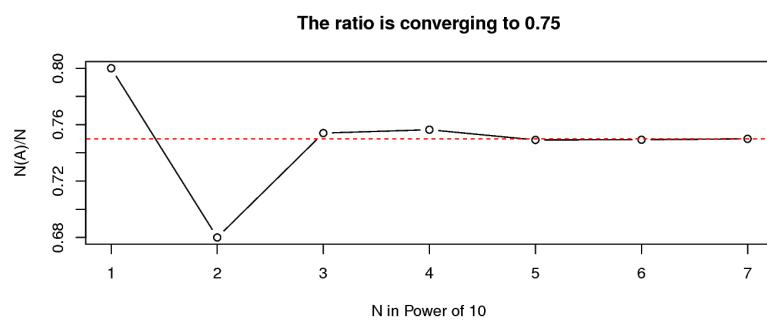
$N$	10	100	1000	$10^4$	$10^5$	$10^6$	$10^7$
$\frac{N(A)}{N}$	0.8	0.68	0.754	0.7564	0.74917	0.749291	0.7499472



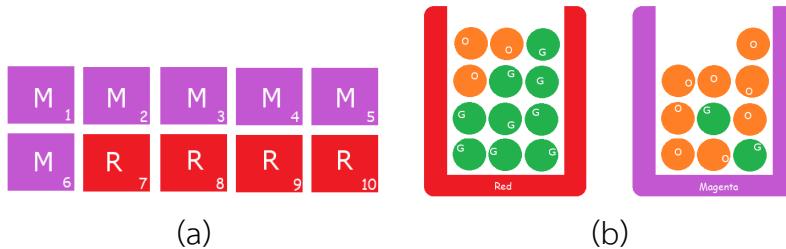
รูปที่ 2.3: ลังใส่ลูกบอล ซึ่งมีลูกบอลอยู่ภายใน 12 ลูก เป็นลูกบอลสีสัมสานลูก และที่เหลือเป็นสีเขียว



รูปที่ 2.4: ผลจากจำลองเหตุการณ์สุ่มหยิบลูกบอล 10 ครั้ง จากกล่องลูกบอลที่แสดงในรูป 2.3. ลูกที่ 1–9 สีเขียว ลูกที่ 10–12 สีส้ม. จากการสุ่มทำ 10 ครั้ง มีครั้งที่ 7 และ 9 ที่หยิบได้ลูกบอลสีส้ม. ดังนั้น อัตราส่วนจำนวนครั้งที่หยิบได้ลูกบอลสีเขียว คือ 0.8 (ระบบที่ด้านบนของภาพ)



รูปที่ 2.5: อัตราส่วน  $\frac{N(A)}{N}$  ลู่เข้าหา  $\Pr(A) = 0.75$  เมื่อ  $N$  เพิ่มขึ้น (แสดงด้วยเส้นประสีแดง)



รูปที่ 2.6: ตัวอย่างลังบรรจุลูกบอลสี. ภาพ (a) แสดงจำนวนลังสีฟ้ากับลังสีแดง (มีลังสีแดงอยู่ 4 ลัง ที่เหลือเป็นสีฟ้า). ภาพ (b) แสดงลูกบอลสีภายในลัง โดย ลังซ้ายสีแดงมีลูกบอลสีส้มอยู่ 3 ลูก ที่เหลือสีเขียว และลังขวาสีฟ้ามีลูกบอลสีเขียวอยู่ 2 ลูก ที่เหลือสีฟ้า

**คุณสมบัติของความน่าจะเป็น.** ความน่าจะเป็นมีคุณสมบัติที่น่าสนใจหลายอย่าง เช่น ความน่าจะเป็นที่จะเกิดผลลัพธ์จากกลุ่มผลลัพธ์ทั้งหมดทุกแบบที่เป็นไปได้ คือ ต้องพบแน่นอน ความน่าจะเป็นมีค่าสูงสุด. นั่นคือ  $\Pr(\Omega) = 1$ . ความน่าจะเป็นที่จะพบเหตุการณ์ที่เป็นไปไม่ได้ คือ ต้องไม่พบแน่นอน ความน่าจะเป็นมีค่าต่ำสุด. นั่นคือ  $\Pr(\emptyset) = 0$ . ความสัมพันธ์ระหว่างความน่าจะเป็นของเหตุการณ์  $A$  กับส่วนเติมเต็ม  $A^c$  คือ  $\Pr(A) = 1 - \Pr(A^c)$  และ  $\Pr(A \cup A^c) = 1$  และ  $\Pr(A \cap A^c) = 0$ . ความน่าจะเป็นของยูเนียน คือ  $\Pr(A \cup B) = \Pr(A \setminus B) + \Pr(B \setminus A) + \Pr(A \cap B)$  หรือ  $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$ . ความน่าจะเป็นของผลต่าง คือ  $\Pr(A \setminus B) = \Pr(A) - \Pr(A \cap B)$ .

**เหตุการณ์ที่ไม่มีส่วนร่วมกัน.** ถ้า  $A \cap B = \emptyset$  และจะเรียกว่า เหตุการณ์  $A$  และเหตุการณ์  $B$  ไม่มีส่วนร่วมกัน (disjoint) และยูเนียนของเหตุการณ์ที่ไม่มีส่วนร่วมกัน จะมีความน่าจะเป็น  $\Pr(A \cup B) = \Pr(A) + \Pr(B)$ .

**กฎของการรวมความน่าจะเป็น.** ถ้า  $B_1 \cup B_2 \cup \dots \cup B_n = \Omega$  และ  $B_i \cap B_j = \emptyset$  สำหรับทุกๆค่าของ  $i \neq j$  ตั้งแต่  $1, \dots, n$  แล้ว กฎของความน่าจะเป็นรวม (law of total probability) กล่าวว่า

$$\Pr(A) = \sum_{i=1}^n \Pr(A \cap B_i) \quad (2.27)$$

จากกฎของความน่าจะเป็นรวม กรณีพิเศษ คือ  $\Pr(A) = \Pr(A \cap B) + \Pr(A \cap B^c)$ .

**ตัวแปรสุ่ม.** เพื่อความสะดวก เหตุการณ์อาจถูกระบุด้วย ตัวแปรสุ่ม เช่น จากตัวอย่าง เหตุการณ์ที่หยิบได้ลูกบอลสีเขียว และเหตุการณ์ที่หยิบได้ลูกบอลสีฟ้า จะสามารถถูกอ้างถึงได้สะดวก และชัดเจนกว่า ถ้ากำหนดตัวแปรสุ่ม  $B$  แทนสีของลูกบอลที่ถูกสุ่มหยิบขึ้นมา. เหตุการณ์ที่หยิบได้ลูกบอลสีเขียว สามารถเขียนเป็น

$B = 0$  เมื่อ 0 แทนสีเขียว และ เหตุการณ์ที่ทิบได้ลูกบอลสีส้ม สามารถ เขียนเป็น  $B = 1$  เมื่อ 1 แทนสีส้ม. ตัวแปรสุ่ม (random variable) อาจถูกนิยามว่า เป็นตัวแปรที่ค่าของมันขึ้นกับผลลัพธ์ของเรื่องที่สนใจ เมื่อเรื่องที่สนใจเป็นกระบวนการที่มีความไม่แน่นอนอยู่.

หมายเหตุ ตัวแปรสุ่มเป็นการแทนเหตุการณ์ด้วยค่าตัวเลข โดยสำหรับธรรมชาติของเหตุการณ์ที่ไม่ได้เป็นตัวเลข การใช้งานตัวแปรสุ่มนี้อาจทำได้โดยการกำหนดความหมายให้กับตัวเลข เช่น 0 แทนสีเขียว และ 1 แทนสีส้ม. แต่ทั้งๆ คั่งเพื่อความสะดวกและชัดเจน อาจมีการใช้สัญลักษณ์ เช่น ‘ $\sigma$ ’ แทนตัวเลข 0 ในกรณีที่ระบุถึงสีเขียว.

ฟังก์ชันการแจกแจง (distribution function) ของตัวแปรสุ่ม  $X$  คือฟังก์ชัน  $F : \mathbb{R} \rightarrow [0, 1]$  โดย

$$F(x) = \Pr(X \leq x) \quad (2.28)$$

เมื่อ  $x$  เป็นค่าของตัวแปรสุ่ม  $X$ .

สังเกตว่า เนื่องจากฟังก์ชันการแจกแจง สื่อถึงค่าความน่าจะเป็นของเหตุการณ์ต่าง ๆ ที่ตัวแปรสุ่มมีค่า จนถึงค่า  $x$  ที่กำหนด ดังนั้นฟังก์ชันการแจกแจงนี้ บางครั้งอาจเรียก ฟังก์ชันการแจกแจงสะสม (cumulative distribution function คำย่อ cdf).

ตัวแปรสุ่มอาจมีได้หลายแบบขึ้นกับลักษณะของค่าของมัน ซึ่งค่าของมันก็คือลักษณะของผลลัพธ์ที่เป็นไปได้. ตัวแปรสุ่มวิญญาณ (discrete random variable) คือตัวแปรสุ่มที่ค่าของมัน อยู่ในเซตจำกัด (finite set) หรืออยู่ในเซตไม่จำกัดแต่นับได้ (countably infinite set). ตัวอย่าง ตัวแปรสุ่มสีของลูกบอล  $B$  นี่ เป็นตัวแปรสุ่มวิญญาณ เนื่องจากค่าของมันมาจากเซตจำกัด ได้แก่  $\{0, 1\}$  (มีจำนวนสมาชิกน้อยกว่าค่าอนันต์  $\infty$ ). ตัวแปรสุ่มจำนวนต้นทุเรียน ก็เป็นตัวแปรสุ่มวิญญาณ เนื่องจากค่าของมันมาจากเซตไม่จำกัดแต่นับได้ ได้แก่  $\{0, 1, 2, 3, \dots\}$ . แต่ตัวแปรสุ่มปริมาณน้ำในอ่างเก็บน้ำ ไม่ใช่ตัวแปรสุ่มวิญญาณ เพราะว่า ค่าของมันมาจาก  $\mathbb{R}^+$  ตัวแปรสุ่มปริมาณน้ำในอ่างเก็บน้ำ จะเป็นตัวแปรสุ่มต่อเนื่อง. หัวข้อ 2.2 อภิปรายตัวแปรสุ่มต่อเนื่องเพิ่มเติม.

ตัวแปรสุ่มวิญญาณ  $X$  จะมีฟังก์ชันมวลความน่าจะเป็น (probability mass function คำย่อ pmf)  $f : \mathbb{R} \rightarrow [0, 1]$  โดย  $f(x) = \Pr(X = x)$ .

ค่าคาดหมาย (expectation หรือ expected value) เป็นค่าเฉลี่ยของตัวแปรสุ่ม และใช้สัญกรณ์ เช่น  $E[X]$  สำหรับค่าคาดหมายของตัวแปรสุ่ม  $X$ . โดยสำหรับตัวแปรสุ่มวิญญาณที่เป็นตัวเลข ค่าคาดหมายสามารถ

คำนวณได้จาก

$$E[X] = \sum_x x \cdot \Pr(X = x). \quad (2.29)$$

ความแปรปรวน (variance) ของตัวแปรสุ่ม ใช้สัญกรณ์ เช่น  $\text{var}[X]$  ซึ่งค่าความแปรปรวน คำนวณได้จาก

$$\text{var}[X] = E[(X - E[X])^2] \quad (2.30)$$

ความน่าจะเป็นร่วม. เมื่อใช้ตัวแปรสุ่มอธิบายเหตุการณ์ ในกรณีที่สนใจเหตุการณ์ที่เกี่ยวข้องกับตัวแปรสุ่มตั้งแต่สองตัวขึ้นไป ความน่าจะเป็นที่ใช้ จะเรียกว่า ความน่าจะเป็นร่วม (joint probability) และใช้สัญกรณ์ เช่น  $\Pr(X, Y)$  หรือความน่าจะเป็นร่วมของตัวแปรสุ่ม  $X$  และตัวแปรสุ่ม  $Y$  และความน่าจะเป็นร่วม  $\Pr(X, Y) = \Pr(X \cap Y)$ . นั่นคือ  $\Pr(X = x, Y = y)$  หมายถึง ความน่าจะเป็นที่ ตัวแปรสุ่ม  $X$  จะมีค่าเป็น  $x$  และตัวแปรสุ่ม  $Y$  จะมีค่าเป็น  $y$ .

ความแปรปรวนร่วมเกี่ยว (covariance) ของตัวแปรสุ่มสองตัว ใช้สัญกรณ์ เช่น  $\text{cov}[X, Y]$  ซึ่งค่าความแปรปรวนร่วมเกี่ยว คำนวณได้จาก

$$\begin{aligned} \text{cov}[X, Y] &= E_{X,Y}[(X - E[X])(Y - E[Y])] \\ &= E_{X,Y}[XY] - E[X]E[Y] \end{aligned} \quad (2.31)$$

เมื่อ  $E_{X,Y}$  หมายถึงค่าคาดหมาย ที่คิดโดยคำนึงถึงความน่าจะเป็นร่วมของ  $X$  และ  $Y$ . นั่นคือ สำหรับตัวแปรสุ่มวิยุต  $\text{cov}[X, Y] = \sum_x \sum_y (x - E[X])(y - E[Y]) \cdot \Pr(X = x, Y = y)$ .

## ความน่าจะเป็นแบบมีเงื่อนไข

ความน่าจะเป็นแบบมีเงื่อนไข (conditional probability) ประมาณโอกาสที่จะเกิดเหตุการณ์ที่สนใจ ในกรณีที่รู้ผลลัพธ์ของเงื่อนไข. ความน่าจะเป็นแบบมีเงื่อนไขจะเน้นบริบทของเงื่อนไข. สัญกรณ์ เช่น  $\Pr(A|B)$  แทนความน่าจะเป็นแบบมีเงื่อนไข ที่หมายถึง ความน่าจะเป็นของเหตุการณ์  $A$  ในกรณีที่เหตุการณ์  $B$  เป็นจริง. เหตุการณ์  $B$  เป็นเงื่อนไข และเป็นบริบทเสริม เป็นข้อมูลเสริมในการประมาณความน่าจะเป็น.

จากตัวอย่างของรูป 2.3 พิจารณาตัวอย่างที่คราวนี้ มีลังอยู่ 10 ลัง ซึ่ง เป็นลังสีแดง 4 ลัง และเป็นลังสีบานเย็น 6 ลัง ดังรูป 2.6. ถ้าสุ่มยกมาหนึ่งลัง โอกาสที่จะเป็นลังที่ 7 คือ  $1/10$  แต่ถ้าเห็นว่าลังที่สุ่มมาเป็นสี

แดง โอกาสที่จะเป็นลังที่ 7 คือ  $1/4$  เพราะว่า มีแค่ลังที่ 7 ถึงลังที่ 10 ที่เป็นสีแดงอยู่แล้ว 4 ลัง. ถ้าเห็นว่าลังที่สุ่มมาสีบานเย็น โอกาสที่จะเป็นลังที่ 7 ไม่มีเลย หรือโอกาสเป็น 0 เพราะลังที่ 7 สีแดง. ข้อมูลพิเศษ หรือบริบทเพิ่มเติมนี้ คือเงื่อนไขที่ใช้ประกอบการประมาณโอกาสของเหตุการณ์.

การคำนวณ. ความน่าจะเป็นแบบมีเงื่อนไข สามารถคำนวณได้จาก

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} \quad (2.32)$$

เมื่อ  $\Pr(B) > 0$ . จากสมการ 2.32 จะได้

$$\Pr(X, Y) = \Pr(X|Y) \cdot \Pr(Y) \quad (2.33)$$

เมื่อ  $X$  และ  $Y$  คือตัวแปรสุ่ม. สมการ 2.33 มักเรียกว่า **กฎผลคูณ** (product rule). นอกจากนั้น พิจารณาสมการ 2.27 และ 2.33 จะพบว่า

$$\Pr(X) = \sum_y \Pr(X, Y = y) = \sum_y \Pr(X|Y = y) \cdot \Pr(Y = y) \quad (2.34)$$

ซึ่ง สมการนี้มักเรียกว่า **กฎผลรวม** (sum rule). สังเกตว่า การใช้กฎผลรวม จะลดตัวแปรสุ่มลงไป ซึ่งการทำ เช่นนี้ จึงอาจถูกเรียกว่า การถลายปัจจัย (marginalization).

กฎผลคูณสามารถใช้ต่อเนื่องกัน ในลักษณะลูกโซ่

$$\begin{aligned} \Pr(X_1, X_2, \dots, X_n) &= \Pr(X_1) \cdot \Pr(X_2, \dots, X_n | X_1) \\ &= \Pr(X_1) \cdot \Pr(X_2 | X_1) \cdot \Pr(X_3, \dots, X_n | X_1, X_2) \\ &\quad \vdots \\ &= \Pr(X_1) \cdot \Pr(X_2 | X_1) \cdot \Pr(X_3 | X_1, X_2) \cdot \Pr(X_4 | X_1, X_2, X_3) \cdots \\ &\quad \cdots \Pr(X_n | X_1, \dots, X_{n-1}) \end{aligned} \quad (2.35)$$

ซึ่งสมการ 2.35 มักจะถูกเรียกว่า **กฎลูกโซ่ของความน่าจะเป็น** (chain rule of probability).

**กฎของเบส** (Bayes' rule หรือ Bayes' theorem) คือ

$$\Pr(Y|X) = \frac{\Pr(X|Y) \cdot \Pr(Y)}{\Pr(X)} \quad (2.36)$$

$$= \frac{\Pr(X|Y) \cdot \Pr(Y)}{\sum_y \Pr(X|Y = y) \cdot \Pr(Y = y)} \quad (2.37)$$

จากความสัมพันธ์ที่ได้จากการกฎของเบล์ การอนุมานค่าที่สนใจจากข้อมูล มักจะเรียกชื่อพจน์ต่าง ๆ ในสมการ 2.36 เพื่อความสะดวก ดังนี้ ถ้ากำหนดให้ตัวแปรสุ่ม  $Y$  แทนเป้าหมายของการอนุมาน และตัวแปรสุ่ม  $X$  แทนข้อมูลประกอบ แล้ว  $\Pr(Y)$  จะเรียกว่า ความน่าจะเป็นก่อน (prior probability คำย่อ prior) ซึ่งหมายถึง ก่อนการนำข้อมูลประกอบมาคิด  $\Pr(Y|X)$  จะเรียกว่า ความน่าจะเป็นภายหลัง (posterior probability คำย่อ posterior) ซึ่งหมายถึง ภายหลังการนำข้อมูลประกอบมาคิด และ หาก  $\Pr(X|Y)$  เขียนอยู่ในรูปฟังก์ชัน  $f(y) = \Pr(X|Y = y)$  ก็จะถูกเรียกว่า พังก์ชันควรจะเป็น (likelihood function คำย่อ likelihood).

ดังนั้น จากกฎของเบล์ และชื่อพจน์ต่าง ๆ อาจสรุปความสัมพันธ์ได้เป็น  $\text{posterior} \propto \text{likelihood} \cdot \text{prior}$ .

**ตัวอย่างการคำนวณ.** กลับมาที่รูป 2.6 อีกครั้ง คราวนี้จะสุมเดือกัง แล้วพอได้ลังแล้วก็จะสุมหิบลูกอลอกมา. โอกาสที่จะสุมได้ลังແดงเป็น  $\frac{4}{10}$  หรือความน่าจะเป็นที่จะได้ลังสีແดง  $\Pr(C = 'r') = 0.4$  โดย  $C = 'r'$  แทนเหตุการณ์ที่จะได้ลังสีແดง. ในทำนองเดียวกัน ความน่าจะเป็นที่จะได้ลังสีบานเย็น  $\Pr(C = 'm') = 0.6$ .

ถ้ารู้ว่าเป็นลังสีແดง เมื่อสุมหิบลูกอลoma โอกาสที่จะหิบได้ลูกอลสีเขียว คือ  $\frac{9}{12} = 0.75$  หรือเขียนเป็นสัญกรณ์ได้ว่า  $\Pr(B = 'g'|C = 'r') = 0.75$  โดย  $B = 'g'$  แทนเหตุการณ์ที่จะหิบได้ลูกอลสีเขียว. ทำนองเดียวกัน ก็จะได้ความน่าจะเป็นแบบมีเงื่อนไขอื่น ๆ ดังนี้

- $\Pr(B = 'o'|C = 'r') = 0.25,$
- $\Pr(B = 'g'|C = 'm') = 0.20,$
- $\Pr(B = 'o'|C = 'm') = 0.80.$  #

สังเกตว่า ความน่าจะเป็นที่จะหิบได้ลูกอลสีเขียวเมื่อรู้ว่าลังสีແดง  $\Pr(B = 'g'|C = 'r')$  ไม่เหมือนกับความน่าจะเป็นที่จะหิบลูกอลสีเขียวและสูมได้ลังสีແดง  $\Pr(B = 'g', C = 'r')$ . สำหรับ  $\Pr(B = 'g'|C = 'r') = 0.75$  นั้นไม่ต้องสนใจเลยว่าโอกาสที่จะได้ลังสีແดงเป็นเท่าไร. ในขณะที่  $\Pr(B = 'g', C = 'r')$  จะประกอบด้วยโอกาสที่จะได้ลังสีແดง  $\Pr(C = 'r') = 0.4$  และโอกาสที่จะหิบได้ลูกอลสีเขียวจากลังนั้น  $\Pr(B = 'g'|C = 'r') = 0.75$  ซึ่งจากกฎผลคูณ (สมการ 2.33) จะได้

$$\begin{aligned}\Pr(B = 'g', C = 'r') &= \Pr(C = 'r') \cdot \Pr(B = 'g'|C = 'r') \\ &= (0.4) \cdot (0.75) = 0.3.\end{aligned}$$

ในทำนองเดียวกันก็จะได้ค่าความน่าจะเป็นต่าง ๆ ดังแสดงในตาราง 2.3.

ทบทวน (1) ผลรวมของความน่าจะเป็นของทุก ๆ เหตุการณ์เป็น 1. นั่นคือ

$$\begin{aligned}\Pr(\Omega) &= \Pr(C = 'r', B = 'g') + \Pr(C = 'r', B = 'o') \\ &\quad + \Pr(C = 'b', B = 'g') + \Pr(C = 'b', B = 'o') \\ &= 0.3 + 0.1 + 0.12 + 0.48 = 1.\end{aligned}$$

ธรรมชาตินี้เป็นคุณสมบัติพื้นฐานของความน่าจะเป็น. ทบทวน (2) ความน่าจะเป็นของเหตุการณ์  $X$  เท่ากับ ผลรวมของความน่าจะเป็นของเหตุการณ์  $X$  และ  $Y$  สำหรับทุก ๆ ความเป็นไปได้ของ  $Y$  ดังเช่น

$$\begin{aligned}\Pr(C = 'r') &= \Pr(C = 'r', B = 'g') + \Pr(C = 'r', B = 'o') \\ &= 0.3 + 0.1 = 0.4 \\ \Pr(C = 'm') &= \Pr(C = 'm', B = 'g') + \Pr(C = 'm', B = 'o') \\ &= 0.12 + 0.48 = 0.6.\end{aligned}$$

ธรรมชาตินี้คือกฎผลบวก (สมการ 2.34).

จากกฎของการบวก จะได้ ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีเขียว และสีส้ม (โดยไม่สนใจสีของลัง)

$$\begin{aligned}\Pr(B = 'g') &= \Pr(C = 'r', B = 'g') + \Pr(C = 'm', B = 'g') \\ &= 0.3 + 0.12 = 0.42 \\ \Pr(B = 'o') &= \Pr(C = 'r', B = 'o') + \Pr(C = 'm', B = 'o') \\ &= 0.1 + 0.48 = 0.58.\end{aligned}$$

และความน่าจะเป็นของลังถ้าหากรู้สีของลูกบอลที่สูมหยิบออกมานึงลูก

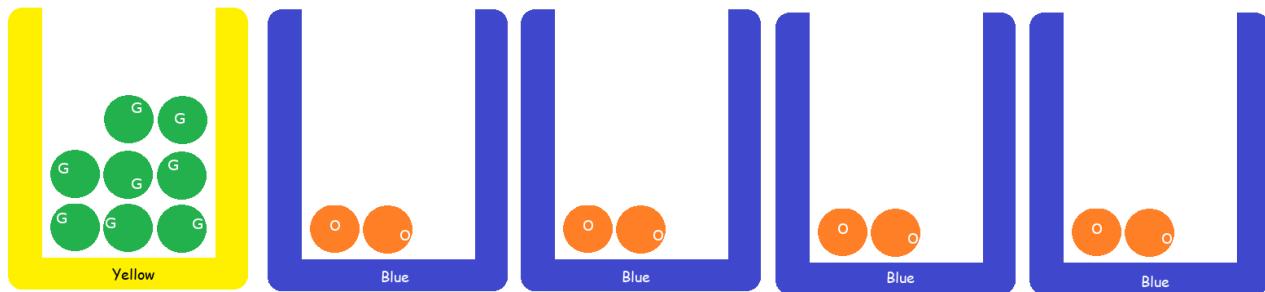
$$\begin{aligned}\Pr(C = 'r' | B = 'g') &= \frac{\Pr(B = 'g' | C = 'r') \cdot \Pr(C = 'r')}{\Pr(B = 'g')} \\ &= \frac{(0.75)(0.4)}{0.42} = 0.71.\end{aligned}$$

ความน่าจะเป็นแบบนี้เนื่องจาก ความน่าจะเป็นที่เปลี่ยนแปลงตามความน่าจะเป็นที่สนใจได้ จากค่าของ ความน่าจะเป็นอื่นที่ประเมินความน่าจะเป็นได้ง่ายกว่า เช่น  $\Pr(B|C)$  จะประเมินได้ง่าย เพราะว่า ลูกบอลอยู่ในลัง ดังนั้นจะนับได้ง่ายว่า ในลังแต่ละสี มีลูกบอลสีไหนจำนวนเท่าไร ต่อจำนวนลูกบอลทั้งหมดในลัง.

$\Pr(C)$  ก็ประเมินได้ง่าย แต่  $\Pr(C|B)$  ประเมินตรง ๆ ได้ยาก เพราะลูกบอลแต่สีกระจายไปทุก ๆ ลัง.

ตารางที่ 2.3: สรุปค่าความน่าจะเป็นของตัวอย่างการสุ่มลังและลูกบอล

ลัง <i>C</i>	ลูกบอล <i>B</i>	
	เขียว ‘g’	ส้ม ‘o’
แดง ‘r’	0.3	0.1
บานเย็น ‘m’	0.12	0.48

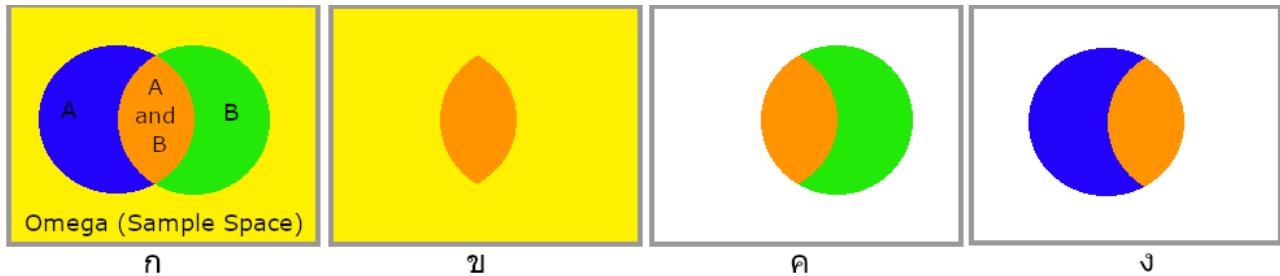


รูปที่ 2.7: ตัวอย่างเน้นความต่างระหว่างสุ่มเลือกลังแล้วสุ่มเลือกลูกบอล (ในภาพ) เปรียบเทียบกับเหลูกบอลทั้งหมดมารวมกัน และ สุ่มเลือกลูกบอล (ไม่มีภาพ)

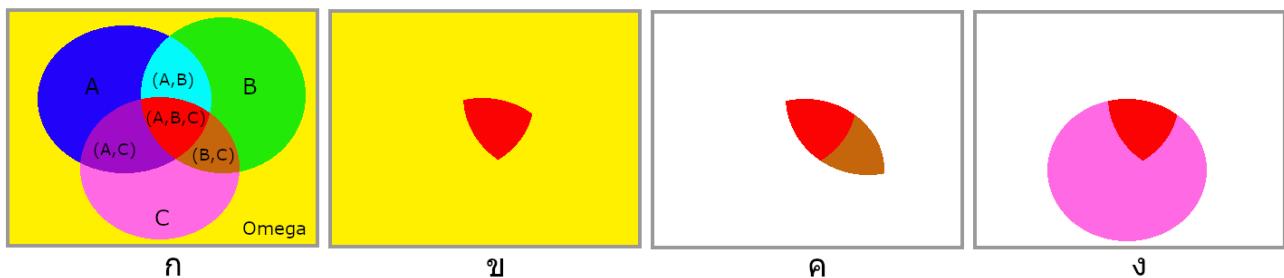
การตีความและความสับสนที่พบได้บ่อย. เพื่อหาความน่าจะเป็นที่จะได้ลูกบอลสีเขียว  $\Pr(B = 'g')$  บ่อยครั้งมักถูกคำนวณด้วย  $11/22 = 0.5$  ซึ่งได้จากการนับลูกบอลสีเขียว เทียบกับลูกบอลทั้งหมด. กรณีนี้ ค่าที่ถูกต้อง  $\Pr(B = 'g') = 0.42$  ไม่เท่ากับ  $11/22 = 0.5$  ซึ่ง  $11/22$  ได้จากการนับลูกบอล โดยสมேือนว่าไม่มีลัง. กรณีหลังนั้น คือสถานการณ์ที่เหลูกบอลทั้งหมดออกจากลัง และสุ่มหยิบลูกไหenkได้. ในขณะที่ ตัวอย่างนี้ ต้องเลือกลังก่อน ถ้าเลือกลังแล้ว ต้องสุ่มหยิบลูกจากในลังที่เลือก.

สองกรณีนี้ จะเห็นต่างกันชัดเจนมาก ถ้าพิจารณากรณี เช่น ตัวอย่างในรูป 2.7 มีลังสีเหลืองแค่ 1 ลัง มีลังสีฟ้า 4 ลัง แต่ลังสีเหลืองมีลูกบอล 8 ลูก ที่ทั้งหมดสีเขียว และลังสีฟ้ามีลูกบอล 2 ลูก ที่ทั้งหมดสีส้ม. เมื่อคิดความน่าจะเป็นแล้วจะพบว่า กรณีนี้ ถ้าเหลูกบอลทั้งหมดออกจากลัง แล้วสุ่มหยิบโอกาสที่จะได้สีเขียวเป็น  $8/16 = 1/2$  แต่ถ้าสุ่มเลือกลังก่อน ลังสีเหลืองมีโอกาสแค่  $1/5$  แล้วโอกาสได้ลูกบอลสีเขียวจากลังนี้เป็น 1 ในขณะที่โอกาสที่จะได้ลังสีฟ้าเป็น  $4/5$  แต่โอกาสได้ลูกบอลสีเขียวเป็น 0 ดังนั้นโอกาสได้ลูกบอลสีเขียวจะเป็นแค่  $(1/5) \cdot 1 + (4/5) \cdot 0 = 1/5$ .

ภาพความเกี่ยวเนื่องของเหตุการณ์ และความน่าจะเป็นร่วม และความน่าจะเป็นแบบมีเงื่อนไข แสดงในรูป 2.8 สำหรับสองเหตุการณ์ และรูป 2.9 สำหรับสามเหตุการณ์.



รูปที่ 2.8: ภาพแสดงความเกี่ยวเนื่องของเหตุการณ์ ภาพ ก แสดงเหตุการณ์  $A$  ด้วยวงกลมซ้าย และเหตุการณ์  $B$  ด้วยวงกลมขวา พื้นที่ที่ทับซ้อนกันคือ เหตุการณ์ร่วม  $A \cap B$  พื้นที่ทั้งหมดในกรอบคือผลลัพธ์ทุก ๆ แบบที่เป็นไปได้ ( $\Omega$  หรือปริภูมิตัวอย่าง). ภาพ ข ความน่าจะเป็น  $\Pr(A \cap B)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B$  จากบริบทของทุก ๆ ผลลัพธ์ที่เป็นไปได้. ภาพ ค ความน่าจะเป็น  $\Pr(A|B)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B$  จากบริบทของเหตุการณ์  $B$ . ภาพ ง ความน่าจะเป็น  $\Pr(B|A)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B$  จากบริบทของเหตุการณ์  $A$ .



รูปที่ 2.9: ภาพแสดงความเกี่ยวเนื่องของเหตุการณ์สามเหตุการณ์ ภาพ ก แสดงเหตุการณ์  $A$  เหตุการณ์  $B$  เหตุการณ์  $C$  ด้วยวงกลม พื้นที่ที่ทับซ้อนกันแทนเหตุการณ์ร่วม ฉลาก เช่น  $(A, B)$  ระบุเหตุการณ์ร่วม  $A \cap B$  และ  $(A, B, C)$  ระบุเหตุการณ์ร่วม  $A \cap B \cap C$ . พื้นที่ทั้งหมดในกรอบคือผลลัพธ์ทุก ๆ แบบที่เป็นไปได้ ( $\Omega$ ). ภาพ ข ความน่าจะเป็น  $\Pr(A \cap B \cap C)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B \cap C$  จากบริบทของทุก ๆ ผลลัพธ์ที่เป็นไปได้. ภาพ ค ความน่าจะเป็น  $\Pr(A|B,C)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B \cap C$  จากบริบทของเหตุการณ์ร่วม  $B \cap C$ . ภาพ ง ความน่าจะเป็น  $\Pr(A,B|C)$  มองโอกาสเกิดเหตุการณ์ร่วม  $A \cap B \cap C$  จากบริบทของเหตุการณ์  $C$ .

**ความเป็นอิสระต่อกัน.** เหตุการณ์  $A$  และเหตุการณ์  $B$  จะเป็นอิสระต่อกัน (independent) ก็ต่อเมื่อ  $\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$ . ดังนั้น จากกฎผลคูณ  $\Pr(A \cap B) = \Pr(A|B) \cdot \Pr(B)$  จะได้ว่า  $\Pr(A|B) = \Pr(A)$  เมื่อ  $A$  และ  $B$  เป็นอิสระต่อกัน. ความหมายคือ ถ้าเหตุการณ์  $A$  และเหตุการณ์  $B$  เป็นอิสระต่อกันแล้ว การรู้หรือไม่รู้ข้อมูลของ  $B$  ก็ไม่ได้เปลี่ยนการประมาณค่าของ  $A$ .

### ตัวอย่างการใช้งานความน่าจะเป็นแบบมีเงื่อนไข

**ปัญหาการตรวจเต้านมด้วยภาพเอ็กซเรย์.** สมมติว่า ผู้หญิงอายุสี่สิบปีคนหนึ่ง ไปทำการตรวจเต้านมด้วยภาพเอ็กซเรย์ (mammogram) และผลตรวจเป็นบวก (positive ซึ่งหมายถึง เครื่องตรวจบอกว่าเป็นมะเร็ง) โดยผลที่ผู้หญิงคนนี้จะเป็นมะเร็งจริง ๆ เป็นเท่าไร

สมมติว่า ข้อมูลประกอบ คือ วิธีการตรวจเต้านมด้วยเอ็กซเรย์มีค่าความไว (sensitivity) ที่ 80% ซึ่งหมาย-

ความว่า ถ้าคนที่เป็นมะเร็งไปทำการตรวจแล้ว โอกาสที่จะได้ผลเป็นบวก คือ 0.8. นั่นคือ  $\Pr(M = 1|C = 1) = 0.8$  เมื่อ  $M = 1$  แทนผลตรวจเป็นบวก (ถ้า  $M = 0$  คือผลตรวจเป็นลบ) และ  $C = 1$  แทนผู้รับการตรวจเป็นมะเร็งจริง ๆ (ถ้า  $C = 0$  คือผู้รับการตรวจไม่ได้เป็นมะเร็ง).

ความเข้าใจผิดอย่างหนึ่งที่พบบ่อย คือ การสรุปว่า ผู้หญิงคนนั้นมีโอกาสเป็นมะเร็ง 80% ซึ่งผิด เพราะการสรุปนี้ ไม่ได้คำนึงถึงความน่าจะเป็นก่อน นั่นคือ โอกาสที่ผู้หญิงอายุสี่สิบปี จะเป็นมะเร็งเต้านม ซึ่งจากสถิติ<sup>6</sup> คือ 17%. นั่นคือ  $\Pr(C = 1) = 0.17$ .

นอกจากนั้น การสรุปยังต้องการข้อมูลว่า วิธีการตรวจมีผลบวกผิด (false positive หรือสัญญาณหลอก false alarm) เป็นเท่าไร ถ้าวิธีการตรวจมีผลบวกผิด เป็น 10%. นั่นคือ  $\Pr(M = 1|C = 0) = 0.1$ .

ดังนั้น เมื่อร่วมหลักฐานทุกอย่างเข้าด้วยกัน โดยใช้กฎของเบส์ จะได้ว่า

$$\begin{aligned}\Pr(C = 1|M = 1) &= \frac{\Pr(M = 1|C = 1)\Pr(C = 1)}{\Pr(M = 1|C = 0)\Pr(C = 0) + \Pr(M = 1|C = 1)\Pr(C = 1)} \\ &= \frac{0.8 \cdot 0.17}{0.1 \cdot 0.83 + 0.8 \cdot 0.17} = 0.62.\end{aligned}$$

ดังนั้นคำตอบที่ถูกคือ 62%.

**ปัญหามอนตี้霍ล.** ปัญหามอนตี้霍ล (Monty Hall Problem) เป็นสถานการณ์การตัดสินใจของผู้เข้าแข่งขันเกมส์โชว์มอนตี้霍ล. ผู้เข้าแข่งขันต้องเลือกเปิดประตูหนึ่งในสามประตู. มีประตูหนึ่งที่ซ่อนรางวัลใหญ่ไว้. อิกสองประตูมีแต่ของปลอกใจ. ผู้เข้าแข่งขันจะได้อะไรก็ตามที่อยู่หลังประตูกลับบ้าน. แต่หลังจากผู้เข้าแข่งขันเลือกประตูไปแล้ว แทนที่พิธีกรจะเปิดประตูนั้นออกทันที. พิธีกรจะเดินไปที่ประตู และเลือกเปิดประตูหนึ่ง ที่ผู้เข้าแข่งขันไม่ได้เลือก. ประตูที่พิธีกรเปิด จะไม่มีรางวัลอยู่ และเสนอโอกาสให้ผู้เข้าแข่งขันเปลี่ยนใจไปเลือกประตูที่เหลืออยู่. ผู้เข้าแข่งขันควรจะเลือกยืนยันประตูเก่า หรือควรจะเลือกเปลี่ยนไปประตูใหม่

ปัญหานี้ ในมุมมองของความน่าจะเป็นแบบมีเงื่อนไข คือ การหาค่าความน่าจะเป็นที่ประตูใหม่จะมีรางวัล เปรียบเทียบกับการหาค่าความน่าจะเป็นที่ประตูเก่าจะมีรางวัล.

กำหนดให้  $\Pr(R = 3|C = 1, H = 2)$  แทน การหาค่าความน่าจะเป็นที่รางวัลจะอยู่ประตูที่สาม เมื่อผู้เข้าแข่งขันเลือกประตูที่หนึ่ง และพิธีกรเปิดประตูที่สอง โดย  $R$  แทนประตูที่มีรางวัล  $C$  แทนประตูที่เลือก และ  $H$  แทนประตูที่พิธีกรเปิด.

<sup>6</sup>จากรายงาน Breast Cancer Facts & Figures 2019-2020 ของ American Cancer Society. เนื้อหาของปัญหานี้ ดัดแปลงจาก [133] โดยปรับปรุงสถิตินี้เป็นค่าล่าสุด.

## พิจารณา กรณี

$$\Pr(R = 1|C = 1, H = 2) = \frac{\Pr(R = 1, H = 2|C = 1)}{\Pr(H = 2|C = 1)} \quad (2.38)$$

ซึ่งเป็นตัวแทนของโอกาส ในกรณีผู้เข้าแข่งขันไม่เปลี่ยนใจแล้วได้รางวัล เปรียบเทียบกับกรณี

$$\Pr(R = 3|C = 1, H = 2) = \frac{\Pr(R = 3, H = 2|C = 1)}{\Pr(H = 2|C = 1)} \quad (2.39)$$

ซึ่งเป็นตัวแทนของโอกาส ในกรณีผู้เข้าแข่งขันเปลี่ยนใจแล้วได้รางวัล.

เพื่อแก้สมการ 2.38 และ 2.39 กฎของเบล์ ต้องการข้อมูลเพิ่มเติม. โอกาสที่รางวัลจะอยู่ประดูใหญ่ มีเท่าๆ กัน. นั่นคือ  $\Pr(R = 1) = \Pr(R = 2) = \Pr(R = 3) = 1/3$  และ เพราะประดูที่มีรางวัลเป็นอิสระ กับประดูที่ผู้แข่งขันเลือก ดังนั้น  $\Pr(R) = \Pr(R|C)$ . นั่นคือ  $\Pr(R = 1|C = 1) = \Pr(R = 2|C = 1) = \Pr(R = 3|C = 1) = 1/3$ .

แต่พิธีกรต้องไม่เปิดประดูที่ผู้แข่งขันเลือก หรือไม่เปิดประดูที่มีรางวัล ดังนั้น  
 $\Pr(H = 2|C = 1, R = 1) = 1/2$  เพราะ พิธีกรเลือกเปิดประดูที่สองหรือที่สามก็ได้  
 $\Pr(H = 2|C = 1, R = 2) = 0$  เพราะ พิธีกรเปิดประดูที่มีรางวัลไม่ได้  
 $\Pr(H = 2|C = 1, R = 3) = 1$  เพราะ พิธีกรเปิดประดูที่สองได้เท่านั้น  
 จากข้อมูลประกอบเหล่านี้ อนุมานได้ว่า

$$\begin{aligned} \Pr(R = 1, H = 2|C = 1) &= \Pr(H = 2|C = 1, R = 1) \cdot \Pr(R = 1|C = 1) \\ &= (1/2)(1/3) = 1/6 \end{aligned}$$

$$\begin{aligned} \Pr(R = 2, H = 2|C = 1) &= \Pr(H = 2|C = 1, R = 2) \cdot \Pr(R = 2|C = 1) \\ &= (0)(1/3) = 0 \end{aligned}$$

$$\begin{aligned} \Pr(R = 3, H = 2|C = 1) &= \Pr(H = 2|C = 1, R = 3) \cdot \Pr(R = 3|C = 1) \\ &= (1)(1/3) = 1/3 \end{aligned}$$

ซึ่งเท่านี้ก็เพียงพอแล้ว จะสรุปได้ว่า โอกาสที่จะได้รางวัล ถ้าผู้แข่งขันเปลี่ยนประดู จะมากกว่า โอกาสถ้าผู้แข่งขันไม่เปลี่ยน (เพราะว่า สมการ 2.38 และ 2.39 มีตัวหารเท่ากัน).

อย่างไรก็ตาม ค่า  $\Pr(H = 2|C = 1)$  ที่สามารถอ่านมาได้จากกฎผลรวม. นั่นคือ

$$\begin{aligned}\Pr(H = 2|C = 1) &= \Pr(R = 1, H = 2|C = 1) + \Pr(R = 2, H = 2|C = 1) \\ &\quad + \Pr(R = 3, H = 2|C = 1) \\ &= 1/6 + 0 + 1/3 = 3/6 = 1/2.\end{aligned}$$

ดังนั้น สรุปได้ว่า

โอกาสเมื่อยืนยันประตุเดิม  $\Pr(R = 1|C = 1, H = 2) = (1/6)/(1/2) = 1/3$

โอกาสเมื่อเปลี่ยนประตุใหม่  $\Pr(R = 3|C = 1, H = 2) = (1/3)/(1/2) = 2/3.$

### ตัวแปรสุ่มต่อเนื่อง

ตัวแปรสุ่ม  $X$  จะเรียกว่า เป็นตัวแปรสุ่มต่อเนื่อง (continuous random variable) ก็ต่อเมื่อ พังก์ชันการแจกแจงของ  $X$  สามารถแสดงได้ในรูป

$$F(x) = \int_{-\infty}^x f(u)du, \quad x \in \mathbb{R} \tag{2.40}$$

สำหรับพังก์ชัน  $f : \mathbb{R} \rightarrow [0, \infty)$  ที่สามารถหาปริพันธ์ได้ (integrable function). พังก์ชัน  $f$  นี้จะเรียกว่า พังก์ชันความหนาแน่นความน่าจะเป็น (probability density function) บางครั้งอาจเรียก พังก์ชันความหนาแน่น (density function คำย่อ pdf) ของ  $X$ .

**สิ่งที่มักสับสน.** ตัวแปรสุ่มต่อเนื่อง มีคุณสมบัติหลายอย่างที่มักถูกเข้าใจผิด. ทั้งตัวแปรสุ่มวิญญาณและตัวแปรสุ่มต่อเนื่องใช้บรรยายเหตุการณ์ ซึ่งเหตุการณ์จะสามารถนำไปหาค่าความน่าจะเป็นได้. ความน่าจะเป็นยังมีคุณสมบัติเหมือนเดิม ไม่ว่า จะเป็น ความน่าจะเป็นของเหตุการณ์ที่บรรยายด้วยตัวแปรสุ่มวิญญาณ หรือ ด้วยตัวแปรสุ่มต่อเนื่อง. นั่นคือ ความน่าจะเป็น มีค่าระหว่างศูนย์ถึงหนึ่ง และผลรวมของความน่าจะเป็นทั้งหมดเป็นหนึ่ง.

แต่ตัวแปรสุ่มวิญญาณและตัวแปรสุ่มต่อเนื่องมีคุณสมบัติหลาย ๆ อย่างต่างกัน. กำหนดให้  $D$  เป็นตัวแปรสุ่มวิญญาณ และ  $C$  เป็นตัวแปรสุ่มต่อเนื่อง สำหรับตัวแปรสุ่มวิญญาณ ความน่าจะเป็นของแต่ละผลลัพธ์ คือค่าฟังก์ชันมวลความน่าจะเป็นของค่าผลลัพธ์นั้น นั่นคือ  $\Pr(D = d) = \text{pmf}(d).$  แต่สำหรับตัวแปรสุ่มต่อเนื่อง ความน่าจะเป็นของแต่ละผลลัพธ์เป็นศูนย์เสมอ ไม่ว่าค่านั้นจะเป็นเท่าไร นั่นคือ  $\Pr(C = c) = 0.$

แม้ว่า ความน่าจะเป็นของแต่ละค่าเป็นศูนย์ แต่ความน่าจะเป็นของช่วงค่าสามารถหาได้. วิธีประเมินความน่าจะเป็น ในกรณีตัวแปรสุ่มต่อเนื่อง จะใช้ฟังก์ชันการแจกแจง  $F(c) = \Pr(C \leq c)$  และความน่าจะเป็น  $\Pr(C > c) = 1 - F(c)$ . เมื่อต้องการประเมินความน่าจะเป็นเป็นช่วง ก็สามารถทำได้โดย  $\Pr(c_0 < C \leq c_1) = F(c_1) - F(c_0)$ . หากต้องการประเมินความน่าจะเป็นบริเวณรอบ ๆ ค่าใดก็สามารถทำได้โดย  $\Pr(c - \varepsilon < C \leq c + \varepsilon) = F(c + \varepsilon) - F(c - \varepsilon)$  เมื่อ  $\varepsilon$  ระบุระยะของบริเวณรอบ ๆ. ข้อควรระวัง ถ้า  $\varepsilon$  เล็กมาก ๆ และ  $\Pr(c - \varepsilon < C \leq c + \varepsilon)$  จะใกล้กับศูนย์ (ความน่าจะเป็นของค่าจุดจุดหนึ่งของตัวแปรสุ่มต่อเนื่องเป็นศูนย์).

ตัวแปรสุ่มวิภาค มี pmf แต่ไม่มี pdf. ตัวแปรสุ่มต่อเนื่อง มี pdf ไม่มี pmf. ค่าของ  $\text{pmf}(d) \in [0, 1]$  สำหรับทุก ๆ ค่า  $d$  เพราะว่าค่าของ pmf( $d$ ) คือค่าความน่าจะเป็น. แต่ค่าของ pdf( $c$ )  $\geq 0$  ซึ่งอาจจะใหญ่กว่า 1 ได้. อย่างไรก็ตาม

$$\int_{-\infty}^{\infty} \text{pdf}(c)dc = F(\infty) = \Pr(C \leq \infty) = 1$$

ตาราง 2.4 สรุปคุณสมบัติของตัวแปรสุ่มต่อเนื่อง ที่มักถูกเข้าใจผิด เปรียบเทียบกับคุณสมบัติของตัวแปรสุ่มวิภาคในประเด็นเดียวกัน.

ตารางที่ 2.4: คุณสมบัติที่มักสับสนของตัวแปรสุ่มต่อเนื่อง

ประเด็น	ตัวแปรสุ่มวิภาค $D$	ตัวแปรสุ่มต่อเนื่อง $C$
ฟังก์ชัน	pmf	pdf
ช่วงค่า	$\text{pmf} : \mathbb{R} \rightarrow [0, 1]$	$\text{pdf} : \mathbb{R} \rightarrow [0, \infty)$
ความน่าจะเป็น	$\Pr(D = d) = \text{pmf}(d)$	$\Pr(C = c) = 0$ pdf ไม่ใช่ค่าความน่าจะเป็น
ฟังก์ชันการแจกแจง	$F(d) = \Pr(D \leq d)$ $F(d) = \sum_{u \leq d} \text{pmf}(u)$	$F(c) = \Pr(C \leq c)$ $F(c) = \int_{-\infty}^c \text{pdf}(u)du$
ค่าคาดหมาย	$E[D] = \sum_d d \cdot \text{pmf}(d)$	$E[C] = \int_{-\infty}^{\infty} c \cdot \text{pdf}(c)dc$

**การแจกแจงเกาส์เชียน.** การแจกแจงแบบต่อเนื่อง ชนิดหนึ่งที่สำคัญ คือ **การแจกแจงเกาส์เชียน** (Gaussian distribution) หรืออาจเรียกว่า **การแจกแจงปกติ** (normal distribution).

การแจกแจงเกาส์เชียน อธิบายการแจกแจงของตัวแปรสุ่มต่อเนื่อง  $X$  ด้วยฟังก์ชันความหนาแน่น

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty \quad (2.41)$$

เมื่อ  $\mu$  และ  $\sigma^2$  เป็นพารามิเตอร์ของแบบจำลอง<sup>7</sup>. พังก์ชันการแจกแจงของการแยกแยะเกาล์เซียน ไม่มีรูปแบบปิด (closed form) ซึ่งในคณิตศาสตร์ หมายถึง นิพจน์ที่สามารถเขียนโดยใช้การคำนวณพื้นฐานได้ และพังก์ชันการแจกแจง มักเขียนในรูป

$$F(x) = \frac{1}{2} \left( 1 + \operatorname{erf} \left( \frac{x - \mu}{\sigma \sqrt{2}} \right) \right) \quad (2.42)$$

เมื่อ

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt. \quad (2.43)$$

รูป 2.10 แสดงความสัมพันธ์ระหว่างพารามิเตอร์  $\mu$  กับ  $\sigma$  และผลต่อค่าความหนาแน่นความน่าจะเป็นของการแจกแจงเกาล์เซียน. ค่า  $\mu$  จะควบคุมตำแหน่งที่มีความหนาแน่นสูงสุด. ค่า  $\sigma$  ควบคุมการแผ่. สังเกตว่า ความหนาแน่นความน่าจะเป็น มีค่าเกินหนึ่งได้ (ภาพล่างที่สองจากซ้าย  $\sigma = 0.2$ ).

รูป 2.11 แสดงความหนาแน่นความน่าจะเป็น (pdf) และการแจกแจงความน่าจะเป็น (cdf). สังเกตว่า การแจกแจงความน่าจะเป็น จะเป็นพังก์ชันเพิ่ม (increasing function) เพราะว่า การแจกแจงความน่าจะเป็น  $F(x) = \Pr(X \leq x)$  ดังนั้น ที่ค่ามากขึ้น ความน่าจะเป็นจะไม่มีทางน้อยลง และที่อนันต์  $F(\infty) = 1$ . การแจกแจงความน่าจะเป็น เป็นค่าปริพันธ์ (integral) ของความหนาแน่นความน่าจะเป็น ดังนั้น พื้นที่ใต้กราฟของความหนาแน่นความน่าจะเป็น จนถึง ณ จุดที่สนใจ จะเท่ากับค่าการแจกแจงความน่าจะเป็น.

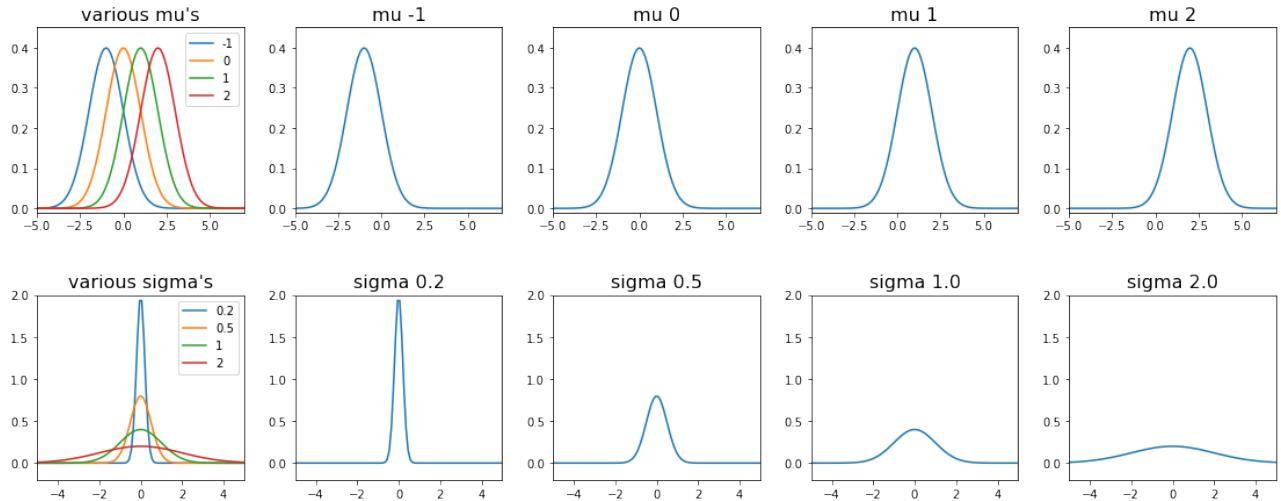
## 2.3 การหาค่าดีที่สุด

การรู้จักรูปแบบ และการเรียนรู้ของเครื่อง ถูกสร้างบนพื้นฐานของศาสตร์การหาค่าดีที่สุด<sup>8</sup> การรู้จักรูปแบบ ต้องการค้นหารูปแบบที่สนใจอย่าง แต่ต้องการให้ผลการค้นหานั้นผิดพลาดน้อยที่สุด. การเรียนรู้ของเครื่อง ต้องการที่จะทำการภารกิจที่ได้รับมอบหมาย ให้ได้สมรรถนะสูงสุด จากประสบการณ์ที่มี.

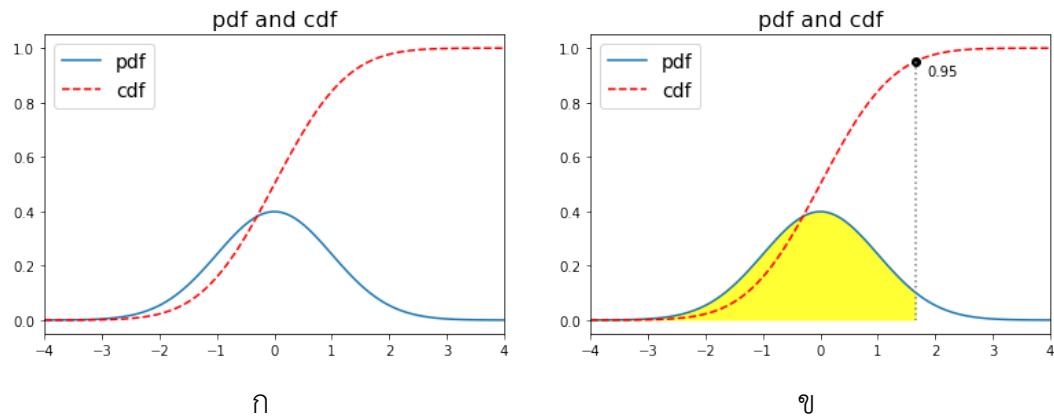
การหาค่าดีที่สุด (optimization) คือ การหาค่าของปัจจัย (แทนด้วยตัวแปร) ที่มีผลให้เป้าหมาย (แทนด้วยพังก์ชันของตัวแปร) มีค่าน้อยที่สุด (หรือมีค่ามากที่สุด ขึ้นกับเป้าหมายที่ต้องการ). ปัจจัยที่ต้องการเลือก เรียกว่า ตัวแปรตัดสินใจ (decision variable) และพังก์ชันแทนเป้าหมาย ซึ่งประมาณความสัมพันธ์ระหว่างค่าของตัวแปรตัดสินใจและเป้าหมายที่ต้องการ เรียกว่า พังก์ชันจุดประสงค์ (objective function).

<sup>7</sup> ในที่นี้ การแจกแจงเกาล์เซียน ถูกมองเป็นแบบจำลองที่ใช้ทำนายความน่าจะเป็นของ  $X$ .

<sup>8</sup> เนื้อหาในหัวข้อนี้ได้รับอิทธิพลหลักจาก [39] และ [61, App. B]



รูปที่ 2.10: ความหนาแน่นความน่าจะเป็น ของการแจกแจงเก้าส์เชียน ที่ค่าพารามิเตอร์ต่าง ๆ. ภาพในแถวบน แสดงผลของค่า  $\mu$  ต่าง ๆ ตั้งแต่ -1 ถึง 2 โดย ค่า  $\mu$  จะบออยู่ข้างบนแต่ละภาพ. ภาพข่ายสุด แสดงผลของค่า  $\mu$  ต่าง ๆ ในภาพเดียวกัน เพื่อการเปรียบเทียบได้ชัดเจน. ภาพในแถวล่าง จัดเรียงในลักษณะเดียวกัน แต่เป็น ผลของค่า  $\sigma$  ต่าง ๆ ตั้งแต่ 0.2 ถึง 2. สังเกตว่า ความหนาแน่นความน่าจะเป็น มีค่ามากกว่าศูนย์เสมอ แต่อาจมีค่ามากกว่าหนึ่งได้ เช่นแสดงในภาพล่างที่สองจากข้าง右.



รูปที่ 2.11: ภาพ ก แสดงความหนาแน่นความน่าจะเป็น (pdf) และการแจกแจงความน่าจะเป็นสะสม (cdf) ของการแจกแจงแบบเก้าส์เชียน. ภาพ ข แสดงค่าของ การแจกแจงสะสม คือพื้นที่ใต้กราฟของความหนาแน่น. นั่นคือ ณ จุดที่แสดง  $cdf(x) = 0.95$  ซึ่งเท่ากับพื้นที่ใต้กราฟของความหนาแน่น (พื้นที่แรเงาสีเหลือง).

ตัวอย่าง ปัญหาการหาค่าดีที่สุด เช่น การเลือกอุณหภูมิบ่มทุเรียน และเป้าหมายคือได้ทุเรียนสุก ซึ่งวัดจากปริมาณน้ำตาล. ถ้าปัจจัยค่าอุณหภูมิ แทนด้วยตัวแปร  $x$  และถ้ามีฟังก์ชัน  $h$  ที่สามารถใช้ในการประมาณความสัมพันธ์ ระหว่างอุณหภูมิที่บ่มกับปริมาณน้ำตาลที่ได้ ดังนั้นตัวแปร  $x$  คือตัวแปรตัดสินใจ และฟังก์ชัน  $h$  คือฟังก์ชันจุดประสงค์. ถ้าปริมาณน้ำตาลที่ได้มาก เป็นดัชนีบอกว่าทุเรียนสุกดี กรณีนี้คือ การหาค่า  $x$  ที่ทำให้ได้ค่าฟังก์ชัน  $h$  ที่มากที่สุด.

ปัญหาค่ามากที่สุด และปัญหาค่าน้อยที่สุด. การหาค่าตัวแปรตัดสินใจ ที่ทำให้ได้ฟังก์ชันจุดประสงค์ มีค่ามากที่สุด นั้นเรียกว่า ปัญหาค่ามากที่สุด (maximization problem). ตัวอย่างปัญหาการเลือกอุณหภูมิบ่มทุเรียน ข้างต้นเป็น การหาค่าดีที่สุดแบบปัญหาค่ามากที่สุด. ปัญหาค่ามากที่สุด ใช้สัญกรณ์

$$\underset{x}{\text{maximize}} \quad h(x) \quad (2.44)$$

หรือ อาจเขียนย่อเป็น  $\max_x h(x)$  ซึ่งระบุว่า ต้องการหาค่าของตัวแปรตัดสินใจ  $x$  ที่ทำให้ฟังก์ชันจุดประสงค์  $h(x)$  มีค่ามากที่สุด.

ทำงานเดียวกัน การหาค่าตัวแปรตัดสินใจ ที่ทำให้ได้ฟังก์ชันจุดประสงค์ มีค่าน้อยที่สุด นั้นเรียกว่า ปัญหาค่าน้อยที่สุด (minimization problem). ตัวอย่างปัญหาค่าน้อยที่สุด เช่น การหาเส้นทางขับรถจากalonแก่นไปร้อยเอ็ด ที่ใช้เวลาเดินทางน้อยที่สุด (ตัวแปรตัดสินใจเลือกเส้นทาง และฟังก์ชันจุดประสงค์ ประมาณเวลาเดินทาง) การหาทำเลตั้งเสาสัญญาณวิทยุ ที่ใช้บประมาณรวมน้อยที่สุด (ตัวแปรตัดสินใจเลือกตำแหน่งที่ตั้งเสาสัญญาณ และฟังก์ชันจุดประสงค์ ประเมินงบประมาณรวม) การหารูปแบบการจัดรูปร่างของโปรตีน ที่ใช้พลังงานน้อยที่สุด (ตัวแปรตัดสินใจเลือกรูปร่างของโปรตีน และฟังก์ชันจุดประสงค์ คำนวณพลังงานที่ใช้). ปัญหาค่าน้อยที่สุด ใช้สัญกรณ์

$$\underset{x}{\text{minimize}} \quad h(x) \quad (2.45)$$

หรือ อาจเขียนย่อเป็น  $\min_x h(x)$  ซึ่งระบุว่า ต้องการหาค่าของตัวแปรตัดสินใจ  $x$  ที่ทำให้ฟังก์ชันจุดประสงค์  $h(x)$  มีค่าน้อยที่สุด. บางครั้ง สัญกรณ์อาจจะระบุเซตของค่าตัวแปรตัดสินใจที่ใช้ค้นหา เช่น  $\min_{x \in \mathbb{R}} h(x)$  ซึ่งระบุว่า ค่าของตัวแปรตัดสินใจสามารถเป็นจำนวนจริงได ๆ หรือ  $\min_{x \in \mathbb{R}^2} h(x)$  ระบุว่า ค่าของตัวแปรตัดสินใจเป็นเวกเตอร์ที่มีสองส่วนประกอบจำนวนจริง.

ปัญหาค่าน้อยที่สุด และปัญหาค่ามากที่สุด จริง ๆ แล้ว เป็นสมือนเรื่องเดียวกันที่มองจากคนละมุม. ปัญหาค่าน้อยที่สุด และปัญหาค่ามากที่สุด สามารถแปลงไปมาระหว่างกันได. นั่นคือ การหาตัวแปรตัดสิน

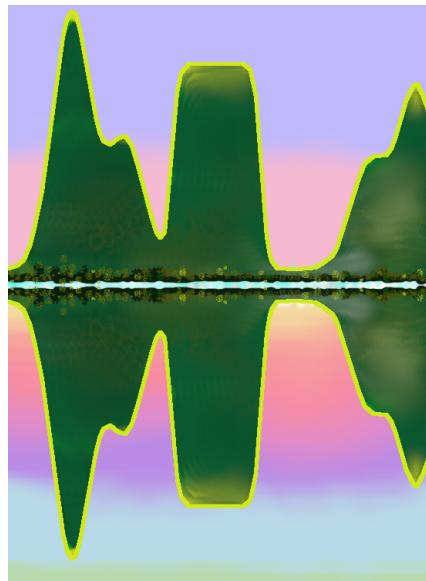
ใจ  $x$  ที่ทำให้ฟังก์ชันจุดประสงค์  $h(x)$  มีค่ามากที่สุด จะเทียบเท่ากับการหาค่า  $x$  ที่ทำให้  $-h(x)$  มีค่าน้อยที่สุด. นั่นคือ  $\max_x h(x) \equiv \min_x -h(x)$ . รูป 2.12 แสดงภาพเปรียบเทียบค่าฟังก์ชัน  $h(x)$  และ  $-h(x)$  ที่เปรียบเสมือนภาพภูเขา และเงาของภาพภูเขาที่สะท้อนน้ำ โดยค่าของฟังก์ชันจะพลิกกลับรอบ ๆ ค่าศูนย์ (ค่าวกเปลี่ยนเป็นลบ ค่าลบเปลี่ยนเป็นบวก ค่าศูนย์อยู่ที่เดิม ค่าวกมากอยู่สูงจะเปลี่ยนเป็นค่าลบมากอยู่ต่ำลง).

ดังนั้นเพื่อความสะดวก ตarrant ใจอ้างถึง ปัญหาค่าน้อยที่สุด แทนปัญหาการหาค่าดีที่สุด โดยเฉพาะ เมื่อ  
อภิปรายถึงวิธีการที่ใช้แก่ปัญหา ซึ่งเมื่อปัญหาทั้งสองแบบเทียบเท่ากัน วิธีต่าง ๆ ที่แก่ปัญหาค่าน้อยที่สุดได้ ก็  
สามารถใช้แก่ปัญหาค่ามากที่สุดได้เช่นกัน.

หมายเหตุ ฟังก์ชันจุดประสงค์ อาจถูกเรียกด้วยชื่ออื่น ๆ เช่น ฟังก์ชันค่าใช้จ่าย (cost function), ฟังก์ชันความสูญเสีย (loss function), ฟังก์ชันพลังงาน (energy function), ฟังก์ชันผลประโยชน์ (utility function) และ ฟังก์ชันคุณค่า (value function). ชื่อเหล่านี้ คือฟังก์ชันจุดประสงค์. แต่ชื่อของฟังก์ชันเหล่านี้ อาจบ่งบอกได้ชัดเจนกว่า ปัญหาเป็นปัญหาค่าน้อยที่สุด (เช่น ฟังก์ชันค่าใช้จ่าย, ฟังก์ชันความสูญเสีย และฟังก์ชันพลังงาน) หรือปัญหาเป็นปัญหาค่ามากที่สุด (เช่น ฟังก์ชันผลประโยชน์ และฟังก์ชันคุณค่า). ชื่อเหล่านี้ มีการใช้อ้างถึงความต้องการตามศาสตร์ ตามสาขาวิชา และตามงานประยุกต์ใช้งานด้านต่าง ๆ เช่น เครชชูลศาสตร์ มักใช้ฟังก์ชันผลประโยชน์, ศาสตร์การวิจัยปฏิบัติการ (operation research) มักพบคำว่า ฟังก์ชันค่าใช้จ่าย. ศาสตร์การเรียนรู้ของเครื่อง มักเลือกใช้คำว่า ฟังก์ชันความสูญเสีย. ส่วนคำว่า ฟังก์ชันพลังงาน อาจพบเห็นได้บ้าง ในงานทางด้านการประมวลผลภาพ.

ผลลัพธ์จากการหาค่าดีที่สุด คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยที่สุด. ค่าที่ได้นี้ เรียกว่า ค่าทำให้น้อยที่สุด (minimizer) และนิยมใช้สัญลักษณ์เป็นตัวแปรตัดสินใจตามด้วยตัวยกที่เป็นดาว เช่น  $x^*$  เพื่อระบุว่า กำลังกล่าวถึง ค่าทำให้น้อยที่สุดที่หามาเสร็จแล้ว ไม่ใช่  $x$  ที่เป็น ตัวแปรตัดสินใจ ที่อาจใช้ค่าใด ๆ ก็ได้. หมายเหตุ ค่าทำให้น้อยที่สุด โดยทั่วไปจะไม่ใช่ค่าที่น้อยที่สุด. รูป 2.13 แสดงแกนนอนแทนค่าของตัวแปรตัดสินใจ  $x$  และแกนตั้งแทนค่าของฟังก์ชันจุดประสงค์  $f(x)$ . ค่า  $x$  ที่น้อยที่สุด แทนด้วยสัญกรณ์  $x_{\min}$  คือ  $-\infty$  เพราะว่า  $-\infty$  เป็นค่าที่น้อยที่สุดของจำนวนจริง และไม่ได้มีข้อจำกัดค่าของ  $x$  (ดูหัวข้อ 2.3 สำหรับกรณีปัญหาแบบมีข้อจำกัด). ค่าทำให้น้อยที่สุด  $x^* = x_1$  เพราะว่า ที่ค่า  $x_1$  ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยที่สุด นั่นคือ  $f(x_1) = f_{\min}$  หรือ  $f(x_1) \leq f(x)$  สำหรับทุก ๆ ค่าของ  $x$ .

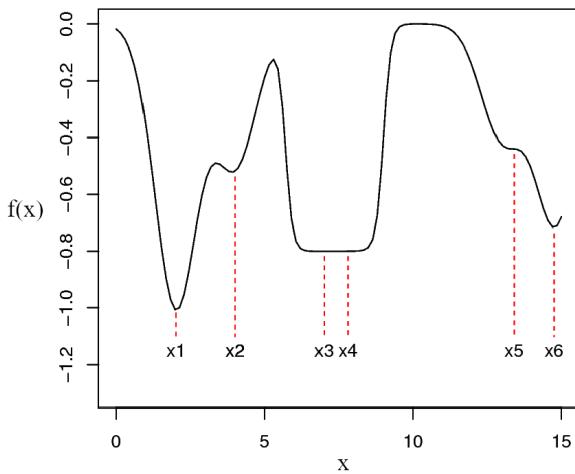
สัญกรณ์  $\min_x f(x)$  นี้ใช้เพื่อรับบุเป้าหมายและตัวแปรที่เกี่ยวข้องเท่านั้น. หากต้องการระบุความสัมพันธ์ในสมการ อาจใช้สัญกรณ์ เช่น  $v = \arg \min_x f(x)$  เพื่อระบุว่า ค่า  $v = x^*$  ที่หาได้จากการแก้ปัญหา



รูปที่ 2.12: ปัญหาค่ามากที่สุดกับ ปัญหาค่าน้อยที่สุดเป็นเรื่องเดียวกันที่มองจากคลุมมุม. การหาค่า  $x$  (เปรียบเหมือนตำแหน่งตามแนววนอน) ของ  $h(x)$  ที่มากที่สุด (เปรียบเหมือนยอดเขา) เป็นเรื่องเดียวกับ การหาค่า  $x$  ที่ทำให้  $-h(x)$  มีค่าน้อยที่สุด (ยอดเขาสูงเท่าไร เขาของยอดเขายิ่งต่ำลงมากเท่านั้น แต่ตำแหน่งตามแนววนอนเป็นที่เดิม).

$\min_x f(x)$ . หากต้องการระบุค่าฟังก์ชันจุดประสังค์ที่น้อยที่สุด อาจระบุด้วยสัญกรณ์ เช่น  $f(x^*)$  หรือ สัญลักษณ์ เช่น  $f_{\min}$  เป็นต้น.

ค่าทำให้น้อยที่สุดท้องถิ่น และค่าทำให้น้อยที่สุดทั่วหมด. จากรูป 2.13 สังเกตว่า แม้  $x_1$  จะเป็นค่าทำให้น้อยที่สุด แต่  $x_2, x_3, x_4, x_5, x_6$  ก็มีลักษณะที่น่าสนใจ. ค่า  $x_1, x_2, x_3, x_4, x_5, x_6$  ทั้งหมด จะเป็น ค่าทำให้น้อยที่สุดท้องถิ่น. ค่าทำให้น้อยที่สุดท้องถิ่น (local minimizers) คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสังค์มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประสังค์จากบริเวณรอบ ๆ. กล่าวอีกอย่างได้ว่า ค่าทำให้น้อยที่สุดท้องถิ่น คือ ค่าที่ทำให้ฟังก์ชันจุดประสังค์มีค่าน้อยที่สุดในท้องถิ่น (ไม่มีใครในล่วงแวงที่น้อยเกิน). ค่า  $x_1, x_2, x_3, x_4, x_5, x_6$  ต่างก็ทำให้ค่าฟังก์ชันจุดประสังค์น้อยกว่าหรือเท่ากับค่าจากบริเวณรอบ ๆ แต่ค่า  $x_1$  นอกจากจะทำให้  $f(x_1)$  มีค่าน้อยกว่าค่าจากบริเวณรอบ ๆ แล้ว (ซึ่งทำให้  $x_1$  เป็นค่าทำให้น้อยที่สุดท้องถิ่น) ค่า  $f(x_1)$  ยังน้อยที่สุดทุกที่ด้วย. ค่าตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสังค์มีค่าน้อยกว่าหรือเท่ากับ ฟังก์ชันจุดประสังค์ของค่าตัวแปรทุกตัวที่เป็นไปได้ เรียกว่า ค่าทำให้น้อยที่สุดทั่วหมด (global minimizer). กล่าวอีกอย่างได้ว่า ค่าทำให้น้อยที่สุดทั่วหมด คือ ค่าที่ทำให้ฟังก์ชันจุดประสังค์มีค่าน้อยที่สุดทั่วหมดทุกที่ (ไม่มีใครในหล้าที่น้อยเกิน). ค่าทำให้น้อยที่สุดทั่วหมด จะเป็นค่าทำให้น้อยที่สุดท้องถิ่นด้วยเสมอ. แต่ค่าทำให้น้อยที่สุดท้องถิ่น อาจไม่ใช่ค่าทำให้น้อยที่สุดทั่วหมด. สถานการณ์ที่ การแก้ปัญหาค่าดีที่สุด แล้วได้ค่าทำให้น้อยที่สุดท้องถิ่น แต่ไม่ใช่ค่าทำให้น้อยที่สุดทั่วหมด มักถูกอ้างถึงว่าเป็น สถานการณ์ที่ดีที่สุดท้อง



รูปที่ 2.13: ค่าทำให้น้อยที่สุดต่าง ๆ ของปัญหา  $\min_x f(x)$ . ค่าน้อยที่สุดของ  $x$  หรือ  $x_{\min} = -\infty$  แต่ค่าทำให้น้อยที่สุด  $x^* = x_1$ . ค่า  $x_2, x_3, x_4, x_5, x_6$  เป็นค่าทำให้น้อยที่สุดท้องถิ่น.

ถิ่น (local optimum).

สังเกตว่า บริเวณ  $x_3$  และ  $x_4$  จะเป็นที่เสมือนที่ร้าบ ซึ่งนอกจาก  $x_3$  และ  $x_4$  ค่าบริเวณนั้นก็จะให้ฟังก์ชันจุดประสังค์ที่เท่ากัน ค่า  $x$  บริเวณที่ราบนั้น ก็จะเรียกว่าเป็น ค่าทำให้น้อยที่สุดท้องถิ่นได้ทั้งหมด เพราะว่า รอบ ๆ ข้างไม่มีใครทำให้ฟังก์ชันจุดประสังค์น้อยกว่า  $x_3$  และ  $x_4$  ก็เป็นค่าทำให้น้อยที่สุดท้องถิ่น แต่เป็นลักษณะที่เรียกว่า จุดอานม้า (saddle point).

## การแก้ปัญหาด้วยวิธีลงเกรเดียนต์

ศาสตร์การหาค่าดีที่สุดนั้นกว้างขวาง และมีการประยุกต์ใช้ที่หลากหลาย. สำหรับการประยุกต์ใช้กับงานการรู้จำรูปแบบ และการเรียนรู้ของเครื่อง ลักษณะปัญหา มักจะถูกตีกรอบออกมายให้ตัวแปรตัดสินใจ  $v \in \mathbb{R}^n$  เมื่อ  $n$  เป็นจำนวนเต็มมีค่าตั้งแต่หนึ่งขึ้นไป และฟังก์ชันจุดประสังค์  $g$  เป็นฟังก์ชันที่สามารถหาอนุพันธ์ได้ (differentiable function). การมีฟังก์ชันจุดประสังค์ที่สามารถหาอนุพันธ์ได้ ช่วยให้สามารถใช้ขั้นตอนวิธีแก้ปัญหาค่าน้อยที่สุด ที่มีประสิทธิภาพได้.

วิธีลงเกรเดียนต์ (gradient descent algorithm) เป็นขั้นตอนวิธี (algorithm) สำหรับปัญหาค่าน้อยที่สุด. วิธีลงเกรเดียนต์ เป็นขั้นตอนวิธีที่เรียบง่าย และใช้ได้ผลดี โดยเฉพาะกับปัญหานาดไม่ใหญ่มาก. แนวคิดของวิธีลงเกรเดียนต์ คือ การใช้ค่าเกรเดียนต์ ช่วยในการหาค่าของตัวแปรตัดสินใจ โดย การเริ่มต้นด้วย ค่าของตัวแปรตัดสินใจ ค่าหนึ่ง และคำนวนค่าเกรเดียนต์ ณ จุดนั้นออกมายแล้วใช้ค่าเกรเดียนต์ที่ได้ บอกทิศทางในการปรับค่าของตัวแปรตัดสินใจ ว่าควรปรับเพิ่มหรือลด มากน้อยเท่าไร ดำเนินการปรับค่าตัวแปรตัดสินใจ

แล้วทำไปเรื่อย ๆ จนพบจุดที่เป็นค่าทำให้น้อยที่สุด.

เกรเดียนต์ ซึ่งเป็นอนุพันธ์ของฟังก์ชันจุดประสังค์ต่อตัวแปรตัดสินใจ จะบอกอัตราการเปลี่ยนค่าของฟังก์ชันจุดประสังค์ เมื่อตัวแปรตัดสินใจเพิ่มค่าขึ้น. ดังนั้นหากเกรเดียนต์เป็นบวกและมีค่ามาก นั่นหมายถึง ถ้าเพิ่มค่าตัวแปรตัดสินใจขึ้น แล้วฟังก์ชันจุดประสังค์จะมีค่าเพิ่มขึ้นมาก. หากค่าของเกรเดียนต์เป็นบวกแต่ มีขนาดเล็ก การเพิ่มค่าตัวแปรตัดสินใจขึ้น จะไปเพิ่มค่าฟังก์ชันจุดประสังค์ขึ้น แต่ไม่มาก และหากเกรเดียนต์ เป็นลบ เมื่อเพิ่มค่าตัวแปรตัดสินใจขึ้น ค่าฟังก์ชันจุดประสังค์จะลดลง. ดังนั้นเกรเดียนต์จึงสามารถใช้เป็น เสมือนเงื่อนงำ ที่บอกทิศทางที่จะปรับค่าตัวแปรตัดสินใจ เพื่อลดค่าฟังก์ชันจุดประสังค์ลงไปเรื่อย ๆ จนไปถึง ค่าฟังก์ชันจุดประสังค์ที่น้อยที่สุดได้. สมการ 2.46 แสดงการคำนวณค่าตัวแปรตัดสินใจ ด้วยวิธีลิงเกรเดียนต์

$$\boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)} - \alpha \nabla g(\boldsymbol{v}^{(k)}) \quad (2.46)$$

เมื่อ ตัวแปร  $\boldsymbol{v}^{(k+1)}$  เป็นค่าใหม่ของตัวแปรตัดสินใจ (ที่ได้จากการคำนวณครั้งที่  $k + 1$ ) และ ตัวแปร  $\boldsymbol{v}^{(k)}$  เป็นค่าเดิมของตัวแปรตัดสินใจ (ที่ได้จากการคำนวณครั้งที่  $k$ ) และ เกรเดียนต์  $\nabla g(\boldsymbol{v}^{(k)})$  คือค่าเกรเดียนต์ ของฟังก์ชันจุดประสังค์ ที่ค่าเดิมของตัวแปรตัดสินใจ (ค่าตัวแปรตัดสินใจที่ได้จากการคำนวณครั้งที่  $k$ ) และค่า สเกลาร์  $\alpha > 0$  เรียกว่า ขนาดก้าว (step size) เป็นค่าที่ใช้ควบคุมอัตราเร็วในการปรับค่าของตัวแปรตัดสิน ใจ.

แนวคิดนี้ อุปมาเหมือน คนที่อยู่บนยอดเขาสูง ที่มองลงมาจนมองอะไรไม่เห็น และต้องการกลับบ้าน ที่อยู่พื้นล่าง. เปรียบฟังก์ชันจุดประสังค์ เป็นเหมือนระดับความสูงของพื้นที่ ณ จุดที่ยืนอยู่ และตัวแปรตัดสิน ใจเป็นเหมือนตำแหน่งเส้นรุ้งเส้นแรง (latitude-longitude location) ณ จุดที่ยืนอยู่ ตัวแปร  $\boldsymbol{v}^{(k)}$  ก็เสมือน ตำแหน่งที่ยืนอยู่ปัจจุบัน และเกรเดียนต์  $\nabla g(\boldsymbol{v}^{(k)})$  ก็เหมือนความชัน ณ ตำแหน่งที่ยืนปัจจุบัน ที่บอกว่า ทิศทางไหนที่รู้สึกว่าพื้นชันขึ้น. ถ้าต้องการกลับบ้าน หรือไปตำแหน่งที่ระดับความสูงที่ต่ำที่สุด วิธีคือ ขยับ จากจุดที่ยืนปัจจุบันไปจุดใหม่ โดยขยับไปในทิศทางลง (ซึ่งคือ ทิศทางตรงข้ามกับทิศที่พื้นชันขึ้น ได้แก่ ทิศ  $-\nabla g(\boldsymbol{v}^{(k)})$ ) โดยต้องค่อย ๆ เดิน ค่อย ๆ ก้าวเล็ก ๆ เพราะถ้าก้าวยาวเกินไป อาจก้าวข้ามทางเดินลงมา แคบ ๆ ที่จะกลับบ้านได้ และขนาดก้าว  $\alpha$  ก็เป็นค่าที่ใช้คุณไม่ให้ก้าวยาวเกินไป หลังจากก้าวไปแล้ว ตอนนี้ ตำแหน่งที่ยืนก็จะเปลี่ยนใหม่เป็น  $\boldsymbol{v}^{(k+1)}$  และก็ขยับแบบนี้อีก ทำซ้ำเรื่อย ๆ จนกลับถึงบ้าน.

**ตัวอย่าง.** การหาค่าทำให้น้อยที่สุดของ  $f(x) = -e^{-(x-5)^2}$  ด้วยวิธีลิงเกรเดียนต์.

- เริ่มต้นด้วยการเลือกจุดเริ่มต้น สมมติเลือก  $x^{(0)} = 6.5$  และเลือกใช้ค่าขนาดก้าว สมมติเลือกเป็น

$\alpha = 0.5$ . เกรเดียนต์สามารถวิเคราะห์การคำนวณໄວ่ได้

$$\nabla f(x) = \frac{df(x)}{dx} = -e^{-(x-5)^2} \cdot (-2x + 10).$$

- ปรับค่าตัวแปรตัดสินใจ ตามสมการ 2.46

การคำนวณครั้งแรก ( $k = 1$ )

$$\begin{aligned} x^{(1)} &= x^{(0)} - (0.5) \cdot \nabla f(x^{(0)}) = 6.5 - (0.5) \cdot \nabla f(6.5) \\ &= 6.5 - (0.5) \cdot \left( -e^{-(6.5-5)^2} \cdot (-2 \cdot (6.5) + 10) \right) = 6.3419. \end{aligned}$$

การคำนวณที่สอง ( $k = 2$ )

$$x^{(2)} = x^{(1)} - (0.5) \cdot \nabla f(x^{(1)}) = 6.3419 - (0.5) \cdot \nabla f(6.3419) = 6.1202$$

การคำนวณต่อ ๆ มา

$$x^{(3)} = 6.1202 - (0.5) \cdot \nabla f(6.1202) = 5.8009$$

$$x^{(4)} = 5.8009 - (0.5) \cdot \nabla f(5.8009) = 5.3792$$

และ  $x^{(5)} = 5.0508; x^{(6)} = 5.0001; x^{(7)} = 5.0000; x^{(8)} = 5.0000.$

- และผลลัพธ์เข้าสู่  $x = 5$  ซึ่งคือค่าทำให้น้อยที่สุด. ส่วน  $f(5) = -1$  เป็นค่าฟังก์ชันจุดประสงค์ที่น้อยที่สุด และค่าเกรเดียนต์ ณ จุดนี้คือ  $\nabla f(5) = 0$ . #

สังเกตว่า ที่ค่าทำให้น้อยที่สุด  $x^*$  จะทำให้เกรเดียนต์  $\nabla f(x^*) = 0$  และนี่คือ ธรรมชาติ<sup>9</sup> ของค่าทำให้น้อยที่สุด ซึ่งคือ ณ จุดค่าทำให้น้อยที่สุด เกรเดียนต์จะมีค่าเป็นศูนย์<sup>10</sup> ดังนั้น ถึงแม้จะคำนวณต่อ ค่าของตัวแปรตัดสินใจก็จะยังคงติดอยู่ที่ค่าทำให้น้อยที่สุด.

การใช้งานวิธีลงเกรเดียนต์. วิธีลงเกรเดียนต์ มีอภิธานพารามิเตอร์ เป็นค่าขนาดกว้าง. อภิธานพารามิเตอร์ (meta-parameter หรือ hyper-parameter) หมายถึง ปัจจัยระดับสูงของวิธีการคำนวณ แต่ละวิธี ที่ผู้ใช้ต้องเลือกให้เหมาะสม. สำหรับวิธีลงเกรเดียนต์ ผู้ใช้ต้องเลือกค่าของขนาดกว้าง. ถ้าเลือกใช้ขนาดกว้างที่ไม่ค่า

<sup>9</sup> นี่คือ เงื่อนไขจำเป็นอันดับแรก (First-Order Necessary Condition). ดู [39] สำหรับรายละเอียด.

<sup>10</sup> เกรเดียนต์  $\nabla f(x^*) = 0$  เป็นจริง ในกรณีที่ว่าไปของปัญหาค่าน้อยที่สุดแบบไม่มีข้อจำกัด. หัวข้อ 2.3 อภิปรายกรณีปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด และแบบฝึกหัด 2.4 แสดงตัวอย่างกรณีพิเศษ ที่ ณ จุดค่าทำให้น้อยที่สุด แต่ค่าเกรเดียนต์ไม่เป็นศูนย์.

เล็กพอ วิธีลิงเกรเดียนต์ จะรับประกันการลู่เข้าได้<sup>11</sup>. การลู่เข้า (convergence) หมายถึง พฤติกรรมที่ดีของผลการคำนวณ สำหรับการคำนวณลักษณะการวนซ้ำขั้ดเกลา (iterative refinement computation) นั่นคือค่าของผลการคำนวณของแต่ละรอบคำนวณ มีการเปลี่ยนแปลงน้อยลง เมื่อรอบการคำนวณเพิ่มมากขึ้น อาจกล่าวได้ว่า การลู่เข้าคือพฤติกรรมที่ ผลลัพธ์จากการคำนวณเปลี่ยนแปลงค่าเข้าหา(ลู่เข้าหา)ค่าได้ค่าหนึ่ง ในลักษณะคงที่ เมื่อรอบการคำนวณเพิ่มมากขึ้น.

ขนาดก้าวที่ใหญ่เกินไป อาจทำให้การคำนวณล้มเหลวได้ แต่ขนาดก้าวที่เล็กเกินไป อาจทำให้ต้องทำการคำนวณหลายรอบมาก ๆ ซึ่งมีผลให้ใช้เวลาในการคำนวณนาน (ดูแบบฝึกหัด 2.22 ประกอบ). ในทางปฏิบัติ เพื่อให้ไม่เสียเวลาคำนวณมากเกินไป เงื่อนไขการจบการคำนวณ (terminating condition) มักถูกใช้ประกอบ. เงื่อนไขการจบที่นิยมใช้กับวิธีลิงเกรเดียนต์ ได้แก่ เงื่อนไขจำนวนรอบสูงสุด (maximum number of iterations) และ เงื่อนไขความคลาดเคลื่อนยินยอม (tolerance).

เงื่อนไขจำนวนรอบสูงสุด จะกำหนดจำนวนรอบที่จะหยุดทำการคำนวณ ไม่ว่าการคำนวณนั้น จะดำเนินไปถึงสิ้นสุดหรือไม่ จะได้ค่าทำให้น้อยที่สุดแล้วหรือไม่. นั่นคือ กำหนด  $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \alpha \nabla g(\mathbf{v}^{(k)})$  สำหรับ  $k = 1, \dots, N_{\max}$  เมื่อ  $N_{\max}$  คือจำนวนรอบคำนวณที่มากที่สุด และการใช้เงื่อนไขนี้ จะทำให้ผู้ใช้ต้องกำหนดจำนวนรอบสูงสุดนี้ด้วย. ดังนั้นอภิธานพารามิเตอร์ จะมีจำนวนรอบสูงสุดขึ้นอีกด้วย.

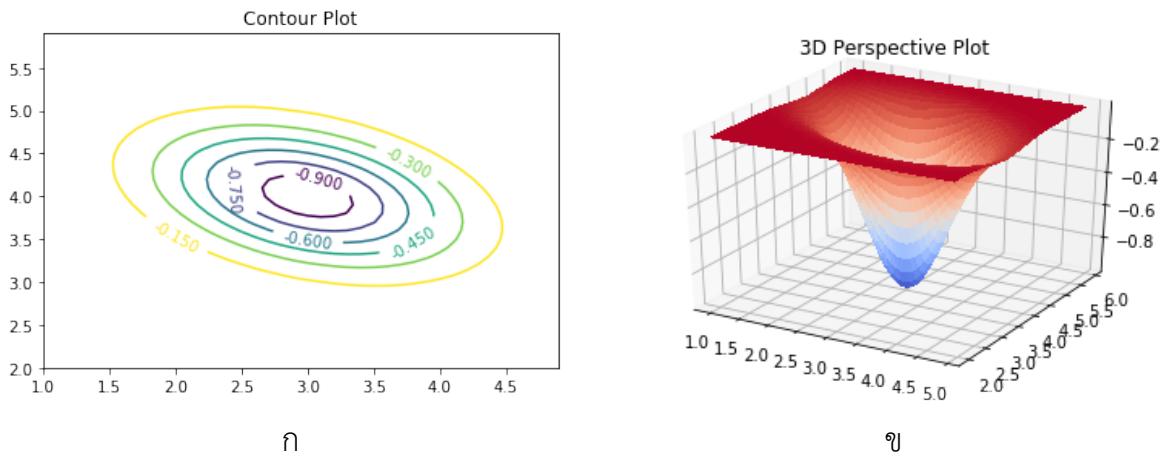
เงื่อนไขความคลาดเคลื่อนยินยอม กำหนดค่าความคลาดเคลื่อนที่ยอมรับได้ ซึ่งอาจเลือกใช้เงื่อนไข ขนาดของเกรเดียนต์  $\epsilon = \|\nabla g(\mathbf{v}^{(k+1)})\|$ . แบบฝึกหัด 2.17 แสดงตัวอย่างโปรแกรมวิธีลิงเกรเดียนต์อย่างง่าย ที่ใช้เงื่อนไขการจบแค่จำนวนรอบสูงสุด. แบบฝึกหัด 2.18 แสดงตัวอย่างโปรแกรมวิธีลิงเกรเดียนต์ที่ใช้เงื่อนไขคลาดเคลื่อนยินยอม.

วิธีลิงเกรเดียนต์ประยุกต์ใช้ได้ง่าย ต้องการแค่เกรเดียนต์ และค่าเริ่มต้น. ค่าเริ่มต้นก่อนการคำนวณของตัวแปรตัดสินใจ อาจเป็นค่าใดก็ได้<sup>12</sup> แต่โดยทั่วไป มักนิยมใช้การกำหนดค่าเริ่มต้น (initialization) ให้กับตัวแปรตัดสินใจ ด้วยค่าที่สุ่มขึ้นมา. (ดูแบบฝึกหัด 2.24)

ปัญหาที่ตัวแปรตัดสินใจมีหลาย ๆ ตัว สามารถใช้วิธีลิงเกรเดียนต์ได้ โดยการจัดหาราย ๆ ตัวแปรตัดสินใจ เป้าหมายรวมกันเป็นเวกเตอร์ตัวแปรตัดสินใจเวกเตอร์เดียว และวิธีลิงเกรเดียนต์เตรียมการมาสำหรับกรณีนี้ อญ্তแล้ว (สมการ 2.46). สังเกตว่า ตัวแปรตัดสินใจเขียนเป็นเวกเตอร์ และอัตราการเปลี่ยนก็ใช้เกรเดียนต์ แต่

<sup>11</sup> วิธีลิงเกรเดียนต์ รับประกันการลู่เข้าสู่ค่าทำน้อยที่สุดทั้งถี่ เมื่อเลือกใช้ขนาดก้าวที่มีค่าเล็กพอ แต่ไม่ได้รับประกันว่า จะเจอค่าทำน้อยที่สุดทั่วหมด หรือไม่ได้รับประกันว่า จะไม่เปิดดือยุคด้านม้า เป็นต้น.

<sup>12</sup> วิธีลิงเกรเดียนต์ทันทันต่อค่าเริ่มต้นต่าง ๆ ในเมื่อที่ว่า โดยทั่วไปแล้ว (ถ้าปัญหาไม่ยากเกินไป) วิธีลิงเกรเดียนต์จะสามารถหาค่าทำให้น้อยที่สุดได้ เมื่อเลือกค่าเริ่มต้นต่าง กัน แต่การเลือกค่าเริ่มต้นที่ดี จะช่วยให้วิธีลิงเกรเดียนต์ทำงานได้เร็วขึ้น และสำหรับราย ๆ กรณี ค่าเริ่มต้นต่าง กันอาจนำไปสู่ค่าทำให้น้อยที่สุดท้องถิ่นและตัว (กรณีปัญหาหลายภาวะ multi-modal problem ดูแบบฝึกหัด 2.23 เพิ่มเติม) หรือสำหรับบาง กรณี ค่าเริ่มต้นบางค่า อาจทำให้วิธีลิงเกรเดียนต์ไม่สามารถทำงานได้เลย. ดูแบบฝึกหัด 2.23 ประกอบ.



รูปที่ 2.14: ภาพคอนทัวร์ ( $v_1$ ) และภาพสามมิติ ( $v_2$ ) ของฟังก์ชันจุดประสังค์  $g(\mathbf{v}) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$

การนำวิธีลิงเกรเดียนต์ ไปเขียนโปรแกรมเพื่อทำงานกับเวกเตอร์ จะต้องระวังเรื่องของตัวแปรเป็นพิเศษ. (ดูแบบฝึกหัด 2.21 ประกอบ) นอกจากนั้น เพื่อช่วยให้สามารถตรวจสอบความถูกต้องในการคำนวณ ให้แน่ใจว่าฟังก์ชันจุดประสังค์ที่ได้อ่านมีประสิทธิภาพ ควรตรวจสอบค่าฟังก์ชันจุดประสังค์ทุก ๆ รอบคำนวณ. ตัวอย่างต่อไปนี้ แสดงการคำนวณของวิธีลิงเกรเดียนต์ เมื่อตัวแปรตัดสินใจมีสองค่า.

ตัวอย่าง เมื่อตัวแปรตัดสินใจเป็นเวกเตอร์. ปัญหา เช่น  $\min_{\mathbf{v}} g$  เมื่อ  $\mathbf{v} = [v_1, v_2]^T$  และ  $g(\mathbf{v}) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$  ซึ่งรูป 2.14 แสดงฟังก์ชันจุดประสังค์ในรูปคอนทัวร์ (contour plot) และในรูปสามมิติ (3d perspective plot) ปัญหานี้สามารถแก้ด้วยวิธีลิงเกรเดียนต์ ดังนี้.

- เลือกอภิมานพารามิเตอร์ ได้แก่ ค่าขนาดก้าว  $\alpha = 0.01$

- เตรียมฟังก์ชันคำนวณเกรเดียนต์

$$\nabla g(\mathbf{v}) = \begin{bmatrix} \frac{\partial g}{\partial v_1} \\ \frac{\partial g}{\partial v_2} \end{bmatrix} = \begin{bmatrix} g(\mathbf{v}) \cdot (-2v_1 - v_2 + 10) \\ g(\mathbf{v}) \cdot (-4v_2 - v_1 + 19) \end{bmatrix} = g(\mathbf{v}) \cdot \begin{bmatrix} -2v_1 - v_2 + 10 \\ -4v_2 - v_1 + 19 \end{bmatrix}.$$

สังเกต ค่าฟังก์ชันจุดประสังค์  $g(\mathbf{v})$  เป็นสเกลาร์ แต่ค่าเกรเดียนต์  $\nabla g(\mathbf{v})$  เป็นเวกเตอร์ ที่มีสัดส่วน (จำนวนส่วนประกอบ) เท่ากับสัดส่วนของตัวแปรตัดสินใจ  $\mathbf{v}$ .

- สุ่มเลือกค่าเริ่มต้น สมมติว่าสุ่มได้  $\mathbf{v}^{(0)} = [2.5, 3.5]^T$

- ปรับค่าตัวแปรตัดสินใจ ตามสมการ 2.46

การคำนวณครั้งแรก ( $k = 1$ )

$$\begin{aligned}\mathbf{v}^{(1)} &= \mathbf{v}^{(0)} - (0.01) \cdot \nabla g(\mathbf{v}^{(0)}) \\ &= [2.5, 3.5]^T - 0.01 \cdot (-0.3679) \cdot [-2(2.5) - 3.5 + 10, -4(3.5) - 2.5 + 19]^T \\ &= [2.5, 3.5]^T - 0.01 \cdot [-0.5518, -0.9197]^T = [2.506, 3.509]^T\end{aligned}$$

การคำนวณที่สอง ( $k = 2$ )

$$\begin{aligned}\mathbf{v}^{(2)} &= \mathbf{v}^{(1)} - (0.01) \cdot \nabla g(\mathbf{v}^{(1)}) \\ &= [2.506, 3.509]^T - 0.01 \cdot [-0.5615, -0.9326]^T = [2.511, 3.519]^T\end{aligned}$$

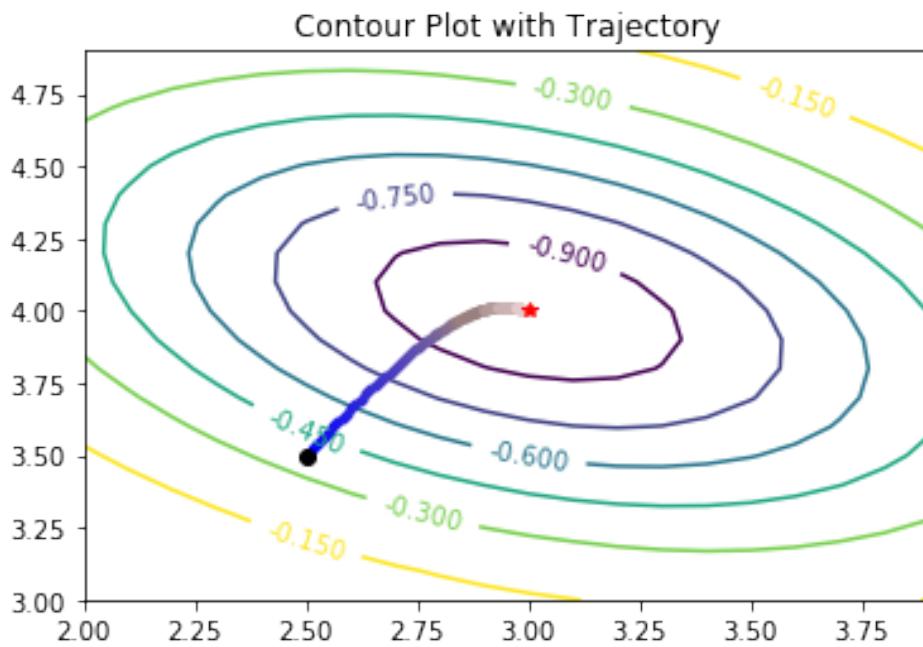
การคำนวณที่สาม ( $k = 3$ )

$$\begin{aligned}\mathbf{v}^{(3)} &= \mathbf{v}^{(2)} - (0.01) \cdot \nabla g(\mathbf{v}^{(2)}) \\ &= [2.511, 3.519]^T - 0.01 \cdot [-0.5711, -0.9452]^T = [2.517, 3.528]^T\end{aligned}$$

- และเมื่อดำเนินการคำนวณต่อไป (ดูแบบฝึกหัด 2.21 ประกอบ) จะพบว่า  $\mathbf{v}^{(300)} = [2.997, 4.001]^T$ ,  $\mathbf{v}^{(400)} = [2.999, 4.000]^T$ ,  $\mathbf{v}^{(500)} = [3.000, 4.000]^T$ ,  $\mathbf{v}^{(600)} = [3.000, 4.000]^T$ . ค่าของ  $\mathbf{v}$  ปรับน้อยมาก ๆ จนถึงแทบไม่ปรับเลยในรอบคำนวณหลัง ๆ นี้คือ  $\mathbf{v}$  ลู่เข้าหา  $[3, 4]^T$ . รูป 2.15 แสดงการปรับค่าตัวแปรตัดสินใจ ในรูปเส้นทางบนภาพคอนทัวร์<sup>13</sup> ของฟังก์ชันจุดประสงค์. รูป 2.16 แสดง ความก้าวหน้า (progress) ของการดำเนินการหาค่าทำให้น้อยที่สุด.

จากรูป 2.16 ค่าของตัวแปรตัดสินใจ  $\mathbf{v}$  ทั้งสองค่า จะปรับเข้าหา  $[3, 4]^T$  นั่นคือคำตอบลู่เข้า. เมื่อดูที่จุดลู่เข้า ดังที่ได้อธิบายมาแล้ว ขนาดของเกรเดียนต์จะเป็นศูนย์ที่จุดศักดิ์สูตร (optimal point) ซึ่งเป็นจุดที่ตัวแปรตัดสินใจปรับค่าไปอยู่ที่ค่าทำให้น้อยที่สุด. ขนาดของเกรเดียนต์จึงนิยมใช้เป็นเงื่อนไขในการจบโปรแกรม เพื่อไม่ต้องคำนวณมากรอบเกินไป เช่นกรณีนี้ จะเห็นว่าทำการคำนวณแค่ราว ๆ 500 รอบก็เท่านั้นการลู่เข้าชัดเจนแล้ว. เมื่อดูที่ฟังก์ชันจุดประสงค์ (ภาพ Loss บนซ้าย) ค่าฟังก์ชันจุดประสงค์จะน้อยลงเรื่อย ๆ ถ้าเลือกค่าขนาดก้าวไม่ใหญ่เกินไป. พฤติกรรมการทำงานของวิธีลงเกรเดียนต์จะมีลักษณะเช่นนี้ คือ ค่าฟังก์ชันจุดประสงค์จะ

<sup>13</sup>รูปเส้นทางบนภาพคอนทัวร์ แห่งนี้ สามารถแสดงได้เฉพาะกรณีที่ตัวแปรตัดสินใจ  $\mathbf{v} \in \mathbb{R}^2$  หากตัวแปรตัดสินใจอยู่ในมิติ碧ุญที่ใหญ่ขึ้น การนำเสนอด้วยภาพจะทำได้ยากมาก.



รูปที่ 2.15: เส้นทางการหาค่าทำให้น้อยที่สุด จุดวงกลมสีดำแสดงจุดเริ่มต้น (ตำแหน่ง  $[2.5, 3.5]^T$ ) และจุดดาวสีแดงแสดงจุดสุดท้าย (ตำแหน่ง  $[3, 4]^T$ ). เส้นทางระหว่างจุดทั้งสอง คือค่าต่าง ๆ ที่ตัวแปรตัดสินใจถูกปรับตั้งแต่รอบคำนวณแรก ๆ ไปจนถึงรอบคำนวณท้าย ๆ.

ลดลง เมื่อรอบคำนวณเพิ่มขึ้น. การดูค่าฟังก์ชันจุดประสงค์ต่อรอบคำนวณเข่นี้ สะท verk ความสามารถตรวจสอบได้ ไม่ว่าตัวแปรตัดสินใจจะมีจำนวนเท่าใด และถูกใช้เป็นแนวทางปฏิบัติ เพื่อติดตามความก้าวหน้า และวิเคราะห์การทำงานของการแก้ปัญหาค่าน้อยที่สุด ว่าดำเนินการได้ดีมากน้อยเพียงใด.

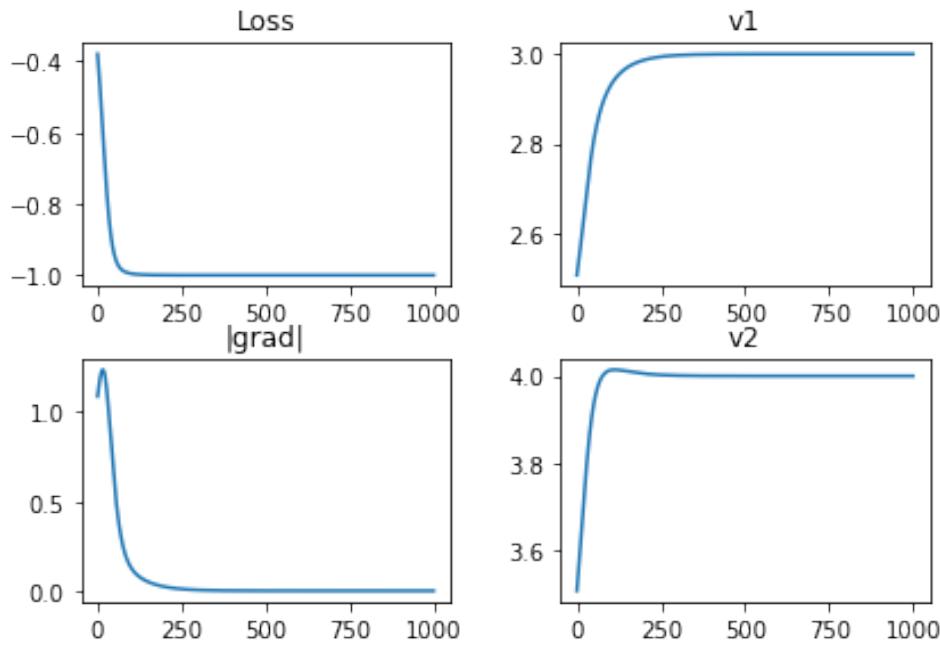
วิธีลงเกรเดียนต์:

“ไม่ว่าเราเริ่มต้นอยู่ที่ไหน ถ้าเรายับไปทิศทางที่ดีขึ้นเรื่อย ๆ

เราจะไปถึงจุดที่ดีที่สุดได้ เพียงต้องทำเรื่อย ๆ และไม่รีบเกินไป”

## การหาค่าดีที่สุดแบบมีข้อจำกัด

การหาค่าดีที่สุดที่ได้อภิปรายมา ค่าของตัวแปรตัดสินใจเป็นค่าจำนวนจริงได ๆ ซึ่งเขียนเป็นสัญกรณ์ทั่ว ๆ ไป  $\min_{\mathbf{v}} g(\mathbf{v})$  โดย  $\mathbf{v} \in \mathbb{R}^n$  เมื่อ  $n$  เป็นจำนวนเต็มที่มีค่าตั้งแต่หนึ่งเป็นตันไป การหาค่าดีที่สุดแบบนี้ ไม่ได้มีเงื่อนไขจำกัดค่าของตัวแปรตัดสินใจ (นอกจากเป็นจำนวนจริง) และการหาค่าดีที่สุดแบบนี้ จะเรียกว่า การหาค่าดีที่สุดแบบไม่มีข้อจำกัด (unconstrained optimization).



รูปที่ 2.16: ความก้าวหน้าในการหาค่าทำให้น้อยที่สุด. ภาพบนซ้าย (**Loss**) แสดงค่าฟังก์ชันจุดประสงค์ต่อรอบการคำนวน. ภาพล่างซ้าย ( $|\text{grad}|$ ) แสดงขนาดเกรเดียนต์ต่อรอบการคำนวน. ขนาดเกรเดียนต์นิยมใช้เป็นเงื่อนไขการจบ เพื่อลดเวลาคำนวนลง. ภาพขวาแสดงค่าตัวแปรตัดสินใจต่อรอบการคำนวน ภาพบนแสดง  $v_1$  และภาพล่างแสดง  $v_2$ .

กรณีที่มีข้อจำกัดของค่าของตัวแปรตัดสินใจ จะเรียกว่า **การหาค่าดีที่สุดแบบมีข้อจำกัด** (constrained optimization) และจะใช้สัญกรณ์ เพื่อระบุข้อจำกัดอย่างชัดเจน เช่น

$$\begin{aligned} & \underset{\boldsymbol{v}}{\text{minimize}} \quad g(\boldsymbol{v}) \\ & \text{subject to} \quad \boldsymbol{v} \in \Omega \end{aligned} \tag{2.47}$$

เมื่อ  $\boldsymbol{v} \in \mathbb{R}^n$  เป็นตัวแปรตัดสินใจ ฟังก์ชัน  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  เป็นฟังก์ชันจุดประสงค์ และเซต  $\Omega$  เป็นเซตย่อยของ  $\mathbb{R}^n$  ที่ระบุค่าของตัวแปรตัดสินใจในช่วงที่สนใจ หรือกลุ่มค่าที่ยอมรับได้. ค่าคำตобของ  $\boldsymbol{v}$  ที่ได้จะต้องอยู่ในเซตย่อยนี้. เซต  $\Omega$  นี้ เรียกว่า **เซตข้อจำกัด** (constraint set หรือ เซตที่เป็นไปได้ feasible set). กรณีที่  $\Omega = \mathbb{R}^n$  ปัญหานอกนี้จะเป็น การหาค่าดีที่สุดแบบไม่มีข้อจำกัด. สัญกรณ์ 2.47 อาจเขียนย่อเป็น  $\min_{\boldsymbol{v}} g(\boldsymbol{v}) \text{ s.t. } \boldsymbol{v} \in \Omega$ .

จากตัวอย่าง การหาอุณหภูมิบ่มทุเรียน ถ้ากำหนดว่า อุณหภูมิบ่มต้องอยู่ในช่วง 30 ถึง 80 องศาเซลเซียส (เพราะว่าเตาบ่มที่มี ทำอุณหภูมิได้ในช่วงแค่นั้น) ปัญหานี้จะเป็นการหาค่าดีที่สุดแบบมีข้อจำกัด และอาจเขียนเป็น สัญกรณ์  $\max_t \text{sugar}(t) \text{ s.t. } 30 \leq t \leq 80$  เมื่อ  $t$  แทนอุณหภูมิบ่มในหน่วยองศาเซลเซียส ฟังก์ชัน  $\text{sugar}$  ประเมินระดับน้ำตาลที่ได้จากการบ่มด้วยอุณหภูมิ  $t$  และข้อจำกัด  $30 \leq t \leq 80$  ระบุค่าของ  $t$  ที่ใช้ได้

ว่า ออยู่ในช่วง 30 ถึง 80. (ปัญหาค่ามากที่สุดนี้ เทียบเท่า ปัญหาค่าน้อยที่สุด  $\min_t - \text{sugar}(t)$  s.t.  $30 \leq t \leq 80$ )

ค่าทำให้น้อยที่สุดท้องถิ่น และค่าทำให้มากที่สุดทั่วหมด. ในกรณีที่มีข้อจำกัด ค่าของตัวแปรตัดสินใจจะพิจารณาจากค่าที่เป็นไปได้ (feasible) เท่านั้น. ค่าทำให้น้อยที่สุดท้องถิ่น คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประสงค์ของค่าที่เป็นไปได้ บริเวณรอบ ๆ. นั่นคือ ถ้ากำหนดให้  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  เป็นฟังก์ชันค่าจริง ที่นิยามสำหรับเซต  $\Omega \subset \mathbb{R}^n$  และ จุด  $\mathbf{v}^* \in \Omega$  เป็นค่าทำให้น้อยที่สุดท้องถิ่น ของฟังก์ชัน  $g$  บนเซต  $\Omega$  ถ้ามีค่า  $\epsilon > 0$  ที่  $g(\mathbf{v}) \geq g(\mathbf{v}^*)$  สำหรับทุกค่าของ  $\mathbf{v} \in \Omega \setminus \{\mathbf{v}^*\}$  และ  $\|\mathbf{v} - \mathbf{v}^*\| < \epsilon$ . ส่วนค่าทำให้น้อยที่สุดทั่วหมด คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสงค์ มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประสงค์ของค่าที่เป็นไปได้ อื่น ๆ ทุกค่า. นั่นคือ จุด  $\mathbf{v}^* \in \Omega$  จะเป็นค่าทำให้น้อยที่สุดทั่วหมดของฟังก์ชัน  $g$  บนเซต  $\Omega$  ถ้า  $g(\mathbf{v}) \geq g(\mathbf{v}^*)$  สำหรับ ทุกค่าของ  $\mathbf{v} \in \Omega \setminus \{\mathbf{v}^*\}$ .

วิธีแก้ปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด. มีสองแนวทางหลักที่ทั่วไปนิยมใช้แก้ปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด คือ แนวทางการแปลงมุมมอง และแนวทางการลงโทษ.

แนวทางการแปลงมุมมอง (projection approach) จะใช้ฟังก์ชัน  $F : \mathbb{R}^n \rightarrow \Omega$  เพื่อแปลงค่าของตัวแปรตัดสินใจมาเป็นค่าที่เป็นไปได้. ตัวอย่าง เช่น เมื่อใช้กับวิธีลงเกรเดียนต์ อาจทำโดย

$$\mathbf{v}^{(k+1)} = F \left( \mathbf{v}^{(k)} - \alpha \cdot \nabla g(\mathbf{v}^{(k)}) \right) \quad (2.48)$$

ในทางปฏิบัติแล้ว ฟังก์ชันแปลง  $F$  นี้ อาจจะยากที่หา หรือยากที่จำแนก ในหลาย ๆ สถานการณ์. งานการเรียนรู้ของเครื่อง มักนิยมใช้แนวทางการลงโทษมากกว่า.

แนวทางการลงโทษ (penalty approach) จะใช้การลงโทษ เมื่อค่าตัวแปรตัดสินใจไม่อยู่ในเขตข้อจำกัด โดยการปรับกรอบปัญหาจากเดิม  $\min_{\mathbf{v}} g(\mathbf{v})$  s.t.  $\mathbf{v} \in \Omega$  ไปเป็นกรอบปัญหาใหม่

$$\min_{\mathbf{v}} g(\mathbf{v}) + \lambda P(\mathbf{v}) \quad (2.49)$$

เมื่อ  $\lambda \in \mathbb{R}$  เป็นพารามิเตอร์ลงโทษ (penalty parameter) ซึ่งนิยมเรียกว่า ลากرانจ์พารามิเตอร์ (Lagrange parameter) และฟังก์ชัน  $P : \mathbb{R}^n \rightarrow [0, \infty)$  เป็นฟังก์ชันต่อเนื่อง (continuous function) เรียกว่า ฟังก์ชันลงโทษ (penalty function) โดย ฟังก์ชันลงโทษจะมีค่าเป็นศูนย์ เมื่อค่าตัวแปรตัดสินใจอยู่ในเขตข้อจำกัด นั่นคือ  $P(\mathbf{v}') = 0$  สำหรับ  $\mathbf{v}' \in \Omega$ .

ตัวอย่างเช่น ปัญหา  $\min_x \sin(x)/(1+x^2)$  s.t.  $x \geq 0$ . รูป 2.17 แสดงฟังก์ชันจุดประสงค์ และแสดงส่วนที่ไม่ผ่านเงื่อนไขด้วยแรเงาสีเทา. รูปแสดงให้เห็นว่า จุดที่ทำให้ค่าฟังก์ชันจุดประสงค์น้อยที่สุด มีค่าเป็นลบ แต่จุดนี้ลักษณะเดียวกับข้อจำกัด และใช้เป็นคำตอบไม่ได้. การหาคำตอบด้วยวิธีลงโทษ อาจเลือกใช้ฟังก์ชัน

$$P(x) = \begin{cases} 0 & \text{เมื่อ } x \geq 0, \\ -x & \text{เมื่อ } x < 0. \end{cases} \quad (2.50)$$

ซึ่งเป็นฟังก์ชันต่อเนื่อง (ทำให้สามารถหาอนุพันธ์ได้ และสามารถใช้วิธีแก้ปัญหา เช่น วิธีลงเกรเดียนต์ได้) และฟังก์ชัน  $P$  ลงโทษค่าที่ลักษณะเดียวกับข้อจำกัดตามปริมาณที่ลักษณะ. รูป 2.18 แสดงค่าของฟังก์ชันลงโทษในสมการ 2.50 และเมื่อนำฟังก์ชันลงโทษนี้ไปใช้ร่วมกับฟังก์ชันจุดประสงค์ ฟังก์ชันที่ปรับใหม่จะเปลี่ยนพฤติกรรมไป ดังแสดงในรูป 2.19.

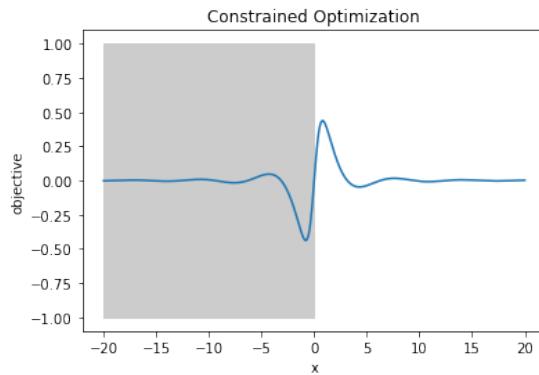
วิธีใช้การลงโทษนี้ ก็คือ การเปลี่ยนจากปัญหาแบบมีข้อจำกัด กลับมาเป็นปัญหาแบบไม่มีข้อจำกัด โดยการตัดแปลงฟังก์ชันจุดประสงค์ใหม่ เพื่อให้คำตอบ อยู่ในช่วงค่าที่เป็นไปได้. ดังนั้น ขั้นตอนการหาคำตอบต่อไป ก็สามารถดำเนินการได้ โดยใช้วิธีแก้ปัญหา แบบเดียวกับปัญหาแบบไม่มีข้อจำกัด.

รูป 2.19 แสดงให้เห็นว่า เมื่อ ค่าของ Lagrange multiplier ใหญ่มากพอ ค่าหน่วยที่สุดของฟังก์ชันจุดประสงค์ใหม่ จะถูกบังคับให้อยู่ในช่วงค่าที่เป็นไปได้. กลไกการทำงานของการลงโทษ คือ การลงโทษจะทำให้คำตอบของปัญหาค่าน้อยที่สุดใหม่ ที่รวมการลงโทษเข้าไป จะเป็น<sup>14</sup> คำตอบของปัญหาค่าน้อยที่สุดแบบมีข้อจำกัดเดิม เมื่อลagrangian หาร่วมกับค่าใหญ่มากพอ.

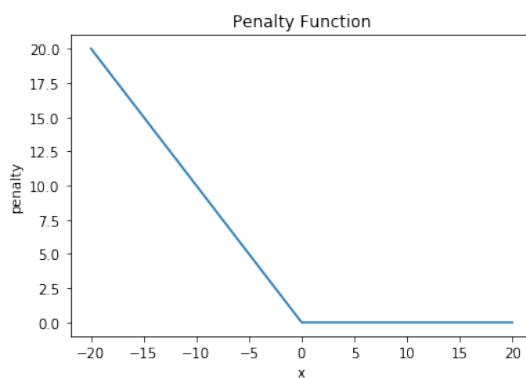
ในทางปฏิบัติ การแก้ปัญหาด้วยวิธีลงโทษ จะเลือกค่าของ Lagrange multiplier มาค่าหนึ่ง และดำเนินการแก้ปัญหา แล้วตรวจสอบค่าคำตอบที่ได้ ว่าอยู่ภายใต้ข้อจำกัดหรือไม่ ถ้าคำตอบที่ได้ อยู่ภายใต้ข้อจำกัด นี่ ก็คือ คำตอบของปัญหาที่ใช้ได้. แต่ถ้าคำตอบที่ได้ ไม่อยู่ภายใต้ข้อจำกัด จะต้องเพิ่มค่าของ Lagrange multiplier ขึ้น และดำเนินการแก้ปัญหาอีก และทำเช่นนี้ เพิ่มค่า Lagrange multiplier ขึ้น จนกว่าจะได้คำตอบที่อยู่ภายใต้ข้อจำกัด.

**เกร็ดความรู้ สติปัญญาของลิง** รายการโนวา ออกอากาศสารคดีสติปัญญาของลิงไม่มีทาง เรื่อง “Ape Genius” ทางสถานีโทรทัศน์พีบีเอส ของสหรัฐอเมริกา ในปี ค.ศ. 2008. รายการนำเสนอองค์ความรู้ทางด้านศึกษาสติปัญญาของลิงชิมแปนซีและลิงโอบโนบลaley ฯ งาน (ลิงชิมแปนซีและลิงโอบโนบลาย มีพันธุกรรมต่างจากมนุษย์แค่ประมาณ 1.2%. มนุษย์แต่ละคนมีพันธุกรรมแตกต่างกันประมาณ

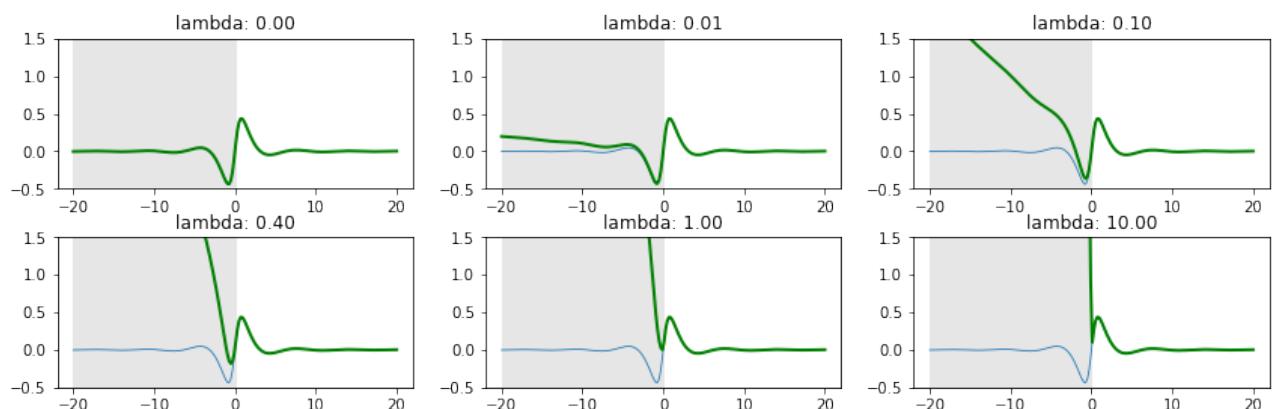
<sup>14</sup> กำหนดให้  $v^{(k)}$  เป็นคำตอบของปัญหาที่มีการลงโทษด้วยค่า  $\lambda_k$  และฟังก์ชันจุดประสงค์เดิม  $g$  เป็นฟังก์ชันต่อเนื่อง และ  $\lambda_k \rightarrow \infty$  เมื่อ  $k \rightarrow \infty$  แล้ว ลิมิต (limit) ของลำดับที่สู่เข้า  $\{v^{(k)}\}$  จะเป็นคำตอบของปัญหาแบบมีข้อจำกัดเดิม. ดู [39] สำหรับรายละเอียด.



รูปที่ 2.17: ตัวอย่างปัญหาที่มีข้อจำกัด  $\min_x \sin(x)/(1+x^2)$  s.t.  $x \geq 0$ . เส้นกราฟ แสดงค่าฟังก์ชันจุดประสงค์  $\sin(x)/(1+x^2)$  ส่วนพื้นที่แรเงา แสดงข้อจำกัดที่ค่าตัวแปรตัดสินใจไม่ได้.



รูปที่ 2.18: ตัวอย่างฟังก์ชันลงโทษ  $P(x) = 0$  เมื่อ  $x \geq 0$  นอกจางานนั้น  $P(x) = -x$ .



รูปที่ 2.19: ตัวอย่างฟังก์ชันสูญเสียที่รวมการลงโทษเข้าไป (เส้นหนาสีเขียว) ที่ใช้ลักษณะพารามิเตอร์ค่าต่าง ๆ (ตั้งแต่ 0 ถึง 10 ตามที่ระบุในแต่ละภาพ) เปรียบเทียบกับฟังก์ชันเดิม (เส้นบางสีฟ้า).

0.1%) รายการ พยายามจะตอบคำถาม ที่ถามว่า ลักษณะของสติปัญญาแห่งมุ่งได้ที่ต่างกัน และทำให้มีแบบจำลองไม่สามารถพัฒนาขึ้นมาสร้างอริธรรม เช่นเดียวกับที่มนุษย์ทำได้.

ความสามารถในการสร้างและใช้เครื่องมือ มีหลักฐานชัดเจนว่า ลิงชิมแปนซีมีการสร้างและใช้เครื่องมือ เช่น เจน ภูดดอล พบลิงชิมแปนซีในแทนซาเนีย ใช้กิ่งไม้ในการล้อมตามกิน และ จิล พริตซ์ พบลิงชิมแปนซีในปาไฟฟากอลี ในประเทศเซเนกัล ที่สร้างหอกจากกิ่งไม้ และใช้เป็นเครื่องมือในการล่าหาอาหาร

ความสามารถในการทำงานร่วมกัน มีหลักฐานหลายอย่างของพฤติกรรมการอุกลาร่วมกันของลิงชิมแปนซี และ สถาบันวิจัยลิงใหญ่ไม่มีทาง (Great Ape Research Institute) ของญี่ปุ่น พบว่า ลิงชิมแปนซีมีความสามารถในการทำงานร่วมกัน มีความสามารถในการขอความช่วยเหลือ และก็สามารถให้ความช่วยเหลือมนุษย์ได้เวลาที่ถูกร้องขอ

ความสามารถในการแก้ปัญหา การศึกษาหนึ่งทดลอง โดยใส่เมล็ดถั่วไว้ในหลอดยา ที่ลิงไม่สามารถจะล้วงเข้าไปหยิบได้ และ ตัวหลอด ก็ยืดติดกับกรงแน่น จนลิงไม่สามารถขยับได้ ลิงใช้เวลาพากหนึ่ง ก่อนจะพบวิธีแก้ปัญหา มันไปที่บ่อห้ำในกรง อมน้ำแล้ว มาพ่นใส่หลอด แล้วอาหารก็ลอยขึ้นบนน้ำ มันเติมน้ำเข้าไป จนอาหารลอยอยู่ในระดับที่เอื้อมถึงได้ สิ่งนี้ แสดงถึงความสามารถในการแก้ปัญหาของลิง.

ความสามารถในการเลียนแบบ ทีมของนักจิตวิทยาแอนดรู วิทเทนต้องการทดสอบความสามารถในการเลียนแบบของลิง ทีมสร้างเครื่องกลไกที่ลิงจะต้องทำการส่องขั้นตอน ได้แก่ หมุนจานให้พอดีซอง และโยกคันโยก เพื่อจะได้กินอาหาร และนำเครื่องไปทดสอบกับตัวลิง ลิงไม่สามารถจะหาวิธีทำนี้ได้เอง แต่ทีมงานค่อย ๆ สอนลิงขั้นมาตัวหนึ่ง จากนั้นลองให้ลิงตัวอื่นดูลิงตัวนี้ทำงานแล้วสังเกตว่าลิงตัวอื่น ๆ ก็สามารถเลียนแบบ เพื่อทำงานส่องขั้นตอนนี้ได้อย่างง่ายดาย.

ความสามารถทางตัวเลข เทหทูโร มัตซูชิรา瓦แห่งมหาวิทยาลัยเกียวโต นำเสนอผลการทดสอบลิงชิมแปนซีชื่อ ไอ ที่แสดงความสามารถทางตัวเลข ในการเข้าใจความหมายของตัวเลขของบิก และยังสามารถรู้คำคำนวณของตัวเลขได้

ความสามารถทางภาษาและการสื่อสาร ลิงโบโนบีอุคนชี เรียนรู้ภาษาอังกฤษได้เอง โดยไม่ได้ถูกสอนโดยตรง และชูชา เจร-รัมบ瓦แสดงให้เห็นว่าคนชีเข้าใจภาษาอังกฤษและสามารถทำตามคำสั่งได้อย่างถูกต้อง

**ความสามารถที่ลิงไม่มี** ความสามารถที่กล่าวมาข้างต้น เป็นความสามารถที่พบทักษานในลิงชิมแปนซีหรือโบโนบี. แต่ความสามารถที่ลิงชิมแปนซีหรือโบโนบีไม่มี และเป็นปัจจัยสำคัญที่ทำให้ลิงไม่สามารถพัฒนาอาริธรรมขึ้นมาได้ เช่นว่าคือความสามารถด้านอารมณ์ ลิงชิมแปนซีมีปัญหาที่เห็นได้ชัดเจน คือปัญหาด้านอารมณ์ ทั้งการแก่งแย่งชิงดิบัน ความรุนแรง และที่สำคัญคือ การควบคุมอารมณ์ตัวเอง.

ความสามารถในการควบคุมตัวเอง การทดลองของแซลลี่ บอยเซนมหาวิทยาลัยรัฐโอไฮโอ แสดงให้เห็นโดยให้ลิงเลือกงานอาหารระหว่างงาน ส่องงาน ที่มีขนมอยู่ไม่เท่ากัน แต่งานที่ลิงเอื้อมมือไปหา จะเป็นงานที่จะนำไปให้กับลิงอีกด้วย ถ้าเป็นขนมที่อยู่บนงาน ลิงไม่สามารถจะอดใจและเอื้อมไปที่งานที่น้อยกว่าได้ มันจะเอื้อมไปที่งานที่มันเห็นอาหารมากกว่าตลอด. แต่พอแซลลี่ บอยเซนเปลี่ยนจากการที่เอาขนมวางไว้ในงานให้เห็น กลับใช้ตัวเลขซึ่งลิงเข้าใจความหมาย วางไว้แทน. ลิงสามารถเรียนรู้ที่จะเอื้อมไปที่งานที่ตัวเลขน้อยกว่าได้. การทดลองนี้แสดงให้เห็นว่า ลิงชิมแปนซีมีปัญหาในการควบคุมอารมณ์ของตัวมันเอง. เวลาที่มันเห็นอาหารอยู่ มันไม่สามารถควบคุมตัวเพื่อเลือกทางเลือกที่ดีกว่าได้ แต่พอตัดแรงกระตุนทางอารมณ์ออก (ใช้ตัวเลขวางแผนอาหารจริง) มันสามารถเลือกทางเลือกที่ดีกว่าได้.

นอกจากความสามารถในการควบคุมตนเองแล้ว ปัจจัยสำคัญอีกสองอย่างที่รายการสรุปว่า เป็นอุปสรรคที่ทำให้สติปัญญาของลิงไม่อาจสะสม สร้างเสริมไปสู่การพัฒนาในระดับเดียวกับมนุษย์ได้ ก็คือ ความสามารถในการเรียนรู้โดยรับการถ่ายทอดจากคนอื่น (หรือลิงตัวอื่น) และความสามารถในการสอน. แม้เด็กอาจไม่ได้แสดงความสามารถในการแก้ปัญหาได้เท่ากับลิงชิมแปนซี แต่เด็ก ๆ แสดงความสามารถที่สามารถเรียนรู้จากสิ่งที่ถูกสอนได้ดีกว่า สุนัขเองก็ยังมีความสามารถในการเรียนจากการสอนของมนุษย์ได้ดีกว่าลิง.

นอกจากความสามารถในการเรียนจากการถ่ายทอด ความเต็มใจที่จะถ่ายทอด หรือความเต็มใจที่จะสอน ก็เป็นส่วนประกอบสำคัญที่ทำให้การถ่ายทอดความรู้เกิดขึ้นได้ และลิงชิมแปนซีไม่ทั้งสององค์ประกอบนี้ อาริธรรมของมนุษย์สร้างโดยการส่งผ่าน

ความรู้และปัญญาจากรุ่นสูงรุ่น. เมลิงสามารถเรียนรู้จากลิงตัวอื่นได้โดยการเลียนแบบ แต่การเรียนรู้โดยการเลียนแบบนั้นมักจะข้าและดีนั่นเอง. บางครั้งยังอาจมีการสูญเสียไปจากการเปลี่ยนรุ่นของลิงอีกด้วย. ลิงรุ่นเก่าตายไป ลิงรุ่นใหม่อาจไม่ได้เรียนรู้สิ่งที่ลิงรุ่นเก่ารู้แล้ว หลาย ๆ อย่างที่ลิงรุ่นเก่ารู้แล้ว เช่นวิธีการใช้เครื่องมือ อาจหายไปจากลิงรุ่นใหม่ และอาจใช้เวลาอีกนานกว่าที่ลิงรุ่นใหม่จะพบวิธีใช้เครื่องมืออีกรอบ.

การควบคุมตัวเอง การเรียนรู้จากการถ่ายทอด และความเต็มใจที่จะสอน เป็นคุณสมบัติที่แยกนุชร์ออกจากลิง และเป็นพื้นฐานอารยธรรมของมนุษย์.

“... just as a bird needs two wings,  
we need both wisdom and compassion.”

---Robina Courtin

“... แบบเดียวกับที่นกต้องมีสองปีก  
เราต้องมีทั้งปัญญาและเมตตา”

—โรบิน่า เคอร์ทิน

## 2.4 อภิรานศัพท์

**มิติ (dimension):** มุ่มมอง หรือหมายถึง มิติปริภูมิค่า ที่เป็นจำนวนตัวเลขที่ใช้ เพื่อระบุตำแหน่งของค่าที่สนใจในปริภูมิค่า เช่น เวกเตอร์  $v \in \mathbb{R}^{12}$  จะอยู่ใน 12 มิติปริภูมิค่า หรือหมายถึงลำดับชั้น เช่น เทคนิค  $V \in \mathbb{R}^{2 \times 3 \times 16}$  จะมี 3 ลำดับชั้น.

**เทนเซอร์ (tensor):** โครงสร้างลำดับชั้นของตัวเลข.

**นอร์ม (norm):** ขนาดของเวกเตอร์.

**ภาพฉายเชิงตั้งฉาก (orthogonal projection):** การแปลงค่าที่สนใจ ลงไปบนทิศทางใหม่ โดยการคูณกับเวกเตอร์หนึ่งของทิศทางใหม่.

**การแยกส่วนประกอบเชิงตั้งฉาก (orthogonal decomposition):** การแยกส่วนประกอบของค่าที่สนใจออกเป็นส่วน ที่แต่ละส่วนอยู่ในปริภูมิย่อยที่ตั้งฉากกัน โดยปริภูมิย่อยทั้งหมดที่ได้จากการแยกส่วนประกอบ จะແแททั่วปริภูมิเดิม.

**เวกเตอร์ลักษณะเฉพาะ (Eigenvector):** เวกเตอร์  $v \neq \mathbf{0}$  ใด ๆ ที่ทำให้สมการ  $\mathbf{Av} = \lambda v$  เป็นจริง เมื่อ  $A$  เป็นเมตริกซ์ที่สนใจ.

**ค่าลักษณะเฉพาะ (Eigenvalue):** ค่าสเกลาร์  $\lambda$  ที่คู่กับเวกเตอร์ลักษณะเฉพาะ ในการทำให้สมการ  $\mathbf{Av} = \lambda v$  เป็นจริง.

**ความน่าจะเป็น (probability):** ค่าประเมินโอกาสที่จะเกิดเหตุการณ์ที่สนใจ.

**ผลลัพธ์ (outcome):** ผลเฉลย หรือความจริงที่เกิดขึ้น สิ่งที่เกิดขึ้น หรือสิ่งที่ประจำษภัยหลัง.

**ปริภูมิตัวอย่าง sample space):** เซตของผลลัพธ์แบบต่าง ๆ ที่เป็นไปได้ทั้งหมด.

**เหตุการณ์ (event):** กลุ่มของผลลัพธ์ที่เป็นไปได้.

**ตัวแปรสุ่ม (random variable):** ตัวแปรที่ใช้อธิบายเหตุการณ์ที่สนใจ ในรูปตัวเลข.

**ฟังก์ชันการแจกแจง (distribution function):** ฟังก์ชันการแจกแจงของตัวแปรสุ่ม  $X$  คือฟังก์ชัน  $F : \mathbb{R} \rightarrow [0, 1]$  โดย  $F(x) = \Pr(X \leq x)$ .

**ตัวแปรสุ่มวิภาค (discrete random variable):** ตัวแปรสุ่มที่ค่าของมันอยู่ในเซตจำกัด หรืออยู่ในเซตไม่จำกัดแต่นับได้ เช่น  $X \in \{0, \dots, 255\}$  เซตของเลขศูนย์ถึงสองร้อยห้าสิบห้า หรือ  $Y \in \mathbb{I}$  เซตของเลขจำนวนเต็ม.

**ตัวแปรสุ่มต่อเนื่อง (continuous random variable):** ตัวแปรสุ่ม ที่ฟังก์ชันการแจกแจงของมัน สามารถเขียนได้ในรูป  $F(x) = \int_{-\infty}^x f(u)du$  เมื่อ  $f : \mathbb{R} \rightarrow [0, \infty)$  เป็นฟังก์ชันความหนาแน่น.

**ฟังก์ชันมวลความน่าจะเป็น (probability mass function):** ฟังก์ชันของตัวแปรสุ่มวิภาค ที่ค่าของมันเท่ากับความน่าจะเป็น.

**ฟังก์ชันความหนาแน่นความน่าจะเป็น (probability density function):** ฟังก์ชันของตัวแปรสุ่มต่อเนื่อง ที่ค่าของมันมากกว่าหรือเท่ากับศูนย์. ค่าของฟังก์ชันความหนาแน่น ไม่ใช่ความน่าจะเป็น แต่ค่าปริพันธ์ในช่วงขอบเขตของมัน จะเป็นความน่าจะเป็นของช่วงขอบเขตนั้น.

**ค่าคาดหมาย (expectation):** ค่าเฉลี่ยของตัวแปรสุ่ม.

**ความน่าจะเป็นแบบมีเงื่อนไข (conditional probability):** ค่าประมาณโอกาสที่จะเกิดเหตุการณ์ที่สนใจ เมื่อรู้ผลลัพธ์ของเงื่อนไข.

**ความน่าจะเป็นก่อน (prior probability):** ความน่าจะเป็นของตัวแปรสุ่มที่ต้องการอนุมาน ก่อนที่จะมีข้อมูลประกอบ.

**ความน่าจะเป็นภายหลัง (posterior distribution):** ความน่าจะเป็นของตัวแปรสุ่มที่ต้องการอนุमาน หลังจากรู้ข้อมูลประกอบแล้ว.

**ฟังก์ชันควรจะเป็น (likelihood function):** ฟังก์ชันของค่าของเงื่อนไข ของความน่าจะเป็นแบบมีเงื่อนไข.

**การถลายปัจจัย (marginalization):** การใช้กฎผลรวม เพื่อลดจำนวนตัวแปรสุ่มลง จากความน่าจะเป็นที่พิจารณา.

**การหาค่าดีที่สุด (optimization):** การหาค่าของตัวแปรตัดสินใจ เพื่อให้ได้ค่าเป้าหมายดีที่สุด.

**ตัวแปรตัดสินใจ (decision variable):** ตัวแปรที่ต้องการหาค่าในการหาค่าดีที่สุด.

**ฟังก์ชันจุดประสงค์ (objective function):** ฟังก์ชันที่ใช้ประมาณค่าเป้าหมายในการหาค่าดีที่สุด หรือบางครั้งอาจเรียกว่า ฟังก์ชันสูญเสีย.

**ปัญหาค่าน้อยที่สุด (minimization problem):** การหาค่าดีที่สุด ที่ต้องการให้ค่าฟังก์ชันจุดประสงค์น้อยที่สุด.

**ค่าทำให้น้อยที่สุด (minimizer):** ค่าของตัวแปรตัดสินใจ ที่ทำให้ค่าฟังก์ชันจุดประสงค์น้อยที่สุด.

**ค่าทำให้น้อยที่สุดท้องถิ่น (local minimizer):** ค่าที่ดีกว่า(หรือไม่แย่กว่า)ค่ารอบ ๆ ข้าง ในปัญหาค่าน้อยที่สุด ต่างจากค่าทำให้น้อยที่สุดทั่วหมดที่ดีกว่า(หรือไม่แย่กว่า)ค่าอื่น ๆ ทั่วหมด. ค่าทำให้น้อยที่สุดทั่วหมดจะเป็นค่าทำให้น้อยที่สุดท้องถิ่นด้วยเสมอ แต่ค่าทำให้น้อยที่สุดท้องถิ่นอาจไม่ใช่ค่าทำให้น้อยที่สุดทั่วหมด.

**วิธีลิงเกรเดียนต์ (gradient descent algorithm):** ขั้นตอนวิธีหนึ่งในการแก้ปัญหาค่าน้อยที่สุด โดยการคำนวณแบบวนซ้ำ และอาศัยค่าเกรเดียนต์ของฟังก์ชันจุดประสงค์เทียบกับตัวแปรตัดสินใจ.

**ขนาดก้าว (step size):** ค่าสเกลาร์ที่ใช้ควบคุมความเร็วในการปรับค่าตัวแปรตัดสินใจ ของขั้นตอนวิธีเพื่อแก้ปัญหาค่าน้อยที่สุด เช่น วิธีลิงเกรเดียนต์.

**เงื่อนไขการจบ (terminating condition):** เงื่อนไขที่ใช้หยุดการคำนวณ.

**การกำหนดค่าเริ่มต้น (initialization):** การกำหนดค่าเริ่มต้นให้กับตัวแปร.

**การลู่เข้า (convergence):** ผลลัพธ์จากการคำนวณแบบวนซ้ำที่ค่าของผลลัพธ์เข้าใกล้ค่า ๆ หนึ่งมากขึ้นเรื่อย ๆ เมื่อจำนวนรอบคำนวณเพิ่มขึ้น.

## 2.5 แบบฝึกหัด

### แบบฝึกหัดเชิงทฤษฎี

“As to methods there may be a million and then some, but principles are few. The man who grasps principles can successfully select his own methods. The man who tries methods, ignoring principles, is sure to have trouble.”

---Ralph Waldo Emerson

“วิธี การ มี เป็น ล้าน และ มาก กว่า แต่ หลัก การ มี ไม่ มาก บุคคล ผู้ ยึด ใน หลัก การ สามารถ เลือก วิธี การ ได้อย่าง ดี บุคคล ผู้ ลอง แต่ วิธี การ ละ เลย หลัก การ ย่อม แน่นอน ว่า จะ มี ปัญหา”

—ราล์ฟ วัลโล อีเมอร์สัน

#### แบบฝึกหัด 2.1

จากระบบสมการ

$$x + y + z = 6 \quad (2.51)$$

$$2x + 2y + 3z = 14 \quad (2.52)$$

$$x + y + 4z = 15 \quad (2.53)$$

จงวิเคราะห์และอภิปรายว่า ระบบสมการนี้ สามารถหาคำตอบได้หรือไม่ หรือถ้าได้ คำตอบมีชุดเดียว หรือ คำตอบมีหลายชุด. สังเกต แต่ละแผล เป็นอิสระเชิงเส้นกัน แต่สอดมวานิชและสอง (สัมประสิทธิ์ของ  $x$  และ  $y$ ) ไม่เป็นอิสระเชิงเส้นกัน จากมุ่งมองของสารสนเทศที่ได้รับ ถ้าการไม่เป็นอิสระเชิงเส้นกันของแผล หมายถึง สมการที่ได้ไม่ได้ให้สารสนเทศเพียงพอ สมการหนึ่งได้มาจากการแปลงเชิงเส้นของสมการอื่น แล้ว การไม่เป็น อิสระเชิงเส้นกันของสอดมวานิช หมายถึงอะไร อภิปราย

#### แบบฝึกหัด 2.2

ความแปรปรวน (variance) เป็นค่าประเมินการแจกแจงของข้อมูลจากค่าเฉลี่ย. ความแปรปรวน อาจถูก สับสนกับเอนโทรปี. เอนโทรปีสารสนเทศ (information entropy) หรือเรียกสั้น ๆ ว่า เอนโทรปี (entropy) เป็นการประมาณค่าคาดหมายของปริมาณสารสนเทศที่ได้รับ. กำหนดให้ตัวแปรสุ่ม  $X$  แทนข้อความที่ได้รับ. ถ้าข้อความที่ได้รับ มีความน่าจะเป็นสูง หมายถึง ข้อมูลมีสารสนเทศน้อย. ถ้า  $\Pr(X = x) = 1$  แปลว่า  $x$  ไม่มีสารสนเทศอะไรอยู่เลย เป็นเรื่องที่รู้กันอยู่แล้ว. ถ้าข้อความที่ได้รับ มีความน่าจะเป็นต่ำ หมายถึง ข้อความมีสารสนเทศมาก เป็นเรื่องใหม่ เป็นเรื่องที่น่าแปลกใจ เป็นข่าวที่คิดไม่ถึง. ปริมาณสารสนเทศของแต่ละข้อความ ถูกนิยามเป็น  $h(x) = -\log_2 \Pr(X = x)$  เมื่อ  $x$  เป็นข้อความที่ได้รับ. ลอการิทึม (logarithm) เป็นฟังก์ชัน

ทางเดียว (monotonic function) และการทำลบ ช่วยให้ได้ความสัมพันธ์ที่  $h(x)$  ค่าน้อย เมื่อ  $\Pr(X = x)$  ค่ามาก และ  $h(x)$  ค่ามาก เมื่อ  $\Pr(X = x)$  ค่าน้อย นอกจากนั้น ยังช่วยให้  $h(x)$  มีค่ามากกว่าหรือเท่ากับศูนย์ด้วย. ค่าเฉลี่ย หรือค่าคาดหมายของปริมาณสารสนเทศ

$$H[X] = - \sum_x \Pr(X = x) \cdot \log_2 \Pr(X = x) \quad (2.54)$$

จะเรียกว่า เอนโทรปีสารสนเทศ.

จงคำนวณค่าความแปรปรวนและค่าเอนโทรปี ของข้อมูลต่อไปนี้. ข้อมูลชุดที่ 1 มีความน่าจะเป็นดังนี้

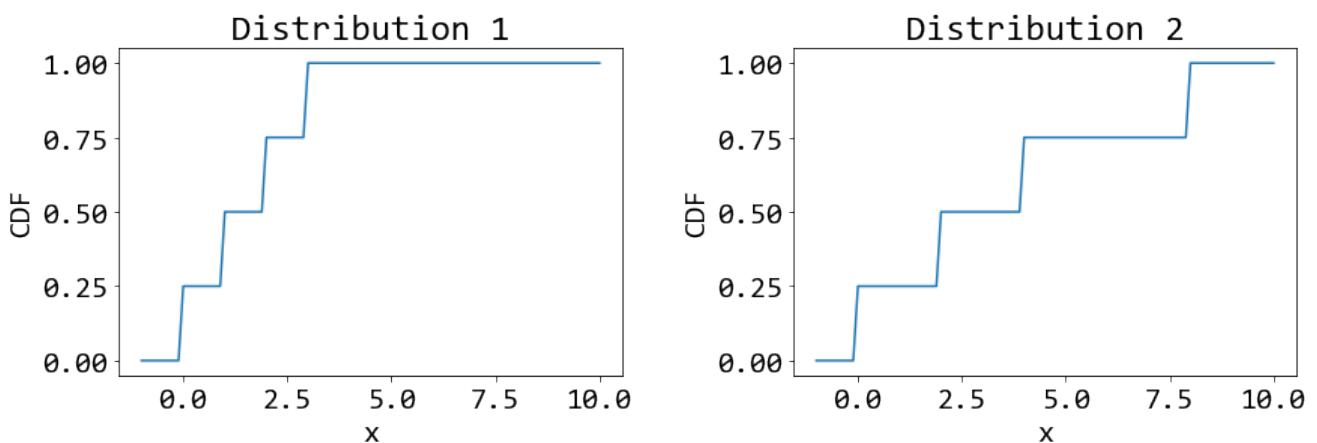
$$\Pr(X = x) = \begin{cases} \frac{1}{4} & \text{เมื่อ } x = 0 \text{ หรือ } x = 1 \text{ หรือ } x = 2 \text{ หรือ } x = 3, \\ 0 & \text{เมื่อ } x \text{ เป็นค่าอื่น ๆ.} \end{cases} \quad (2.55)$$

และข้อมูลชุดที่ 2 มีความน่าจะเป็นดังนี้

$$\Pr(X = x) = \begin{cases} \frac{1}{4} & \text{เมื่อ } x = 0 \text{ หรือ } x = 2 \text{ หรือ } x = 4 \text{ หรือ } x = 8, \\ 0 & \text{เมื่อ } x \text{ เป็นค่าอื่น ๆ.} \end{cases} \quad (2.56)$$

รูป 2.20 แสดงการแจกแจงของข้อมูลทั้งสองชุด. การแจกแจง  $F(x) = \Pr(X \leq x) = \sum_{v \leq x} \Pr(X = v)$ .

สังเกตผลลัพธ์ของค่าความแปรปรวนและค่าเอนโทรปีที่คำนวณได้. สรุปและอภิปรายค่าความแปรปรวน และค่าเอนโทรปีของข้อมูลสองชุดนี้ พร้อมอภิปรายความต่างกันของค่าความแปรปรวน และค่าเอนโทรปี โดยทั่วไป และยกตัวอย่างลักษณะของข้อมูลที่ทำให้เห็นความต่างชัดเจนประกอบ.



รูปที่ 2.20: ภาพ ก แสดงการแจกแจงของข้อมูลชุดที่ 1. ภาพ ข แสดงการแจกแจงของข้อมูลชุดที่ 2.

คำใบ้ ถ้ากำหนด  $X_1$  เป็นตัวแปรสุ่มของข้อมูลชุดที่ 1 และค่าคาดหมาย  $E[X_1] = \sum_x x \cdot \Pr(X_1 = x) = 0(1/4) + 1(1/4) + 2(1/4) + 3(1/4) = 1.5$ .

### แบบฝึกหัด 2.3

จงแก้ปัญหา  $\min_v g(v)$  s.t.  $v \leq 2$  เมื่อ

$$g(v) = 0.2 \log(1 + e^{-v-3}) + \frac{1.5}{1 + e^{-v+2}} - 2.5e^{-0.01(v-10)^2} \quad (2.57)$$

โดยใช้วิธีการลงโทษ (penalty method) พร้อมอภิปรายว่า (1) กำหนดฟังก์ชันลงโทษเป็นอะไร (2) ฟังก์ชันจุดประสงค์ที่รวมการลงโทษเข้าไปแล้วเป็นอะไร (3) เกรเดียนต์ของฟังก์ชันจุดประสงค์ เกรเดียนต์ของฟังก์ชันลงโทษ และเกรเดียนต์ของฟังก์ชันที่ถูกลงโทษ (ฟังก์ชันจุดประสงค์ที่รวมการลงโทษเข้าไปแล้ว) เป็นอะไรบ้าง และ (4) คำตอบของปัญหาคืออะไร และเลือกค่าลากរานจ์พารามิเตอร์อย่างไร ยกตัวอย่างค่าที่ทำงานได้ดู แบบฝึกหัด 2.25 สำหรับผลจากการแก้ปัญหา ด้วยโปรแกรม.

คำใบ้ พจน์แรกในฟังก์ชันจุดประสงค์ ได้แก่  $0.2 \log(1 + e^{-v-3})$  คือ ฟังก์ชันบวกอ่อน (softplus function) ซึ่งมีรูปพื้นฐานคือ  $\log(1 + e^x)$  และมีอนุพันธ์คือ  $1/(1 + e^{-x})$  ที่เรียกว่า ฟังก์ชันซิกมอยด์ (sigmoid function). พจน์ที่สองได้แก่  $1.5/(1 + e^{-v+2})$  คือ ฟังก์ชันซิกมอยด์ ที่พิงกล่าวไป และฟังก์ชันซิกมอยด์ในรูปพื้นฐาน  $\sigma(x) = 1/(1 + e^{-x})$  มีอนุพันธ์คือ  $e^x/(1 + e^x)^2$  หรือเท่ากับ  $\sigma(x) \cdot (1 - \sigma(x))$ . พจน์ที่สามได้แก่  $-2.5e^{-0.01(v-10)^2}$  คือ ฟังก์ชันเกาล์เชียน (Gaussian function) ซึ่งรูปพื้นฐาน  $e^{-x^2}$  มีอนุพันธ์เป็น  $-2xe^{-x^2}$ .

### แบบฝึกหัด 2.4

จงแก้ปัญหา  $\min_v g(v)$  s.t.  $v \leq -0.5$  เมื่อ

$$g(v) = \begin{cases} -(x+1) & \text{เมื่อ } x < -1, \\ x+1 & \text{เมื่อ } -1 \leq x < 0, \\ (-x+1) & \text{เมื่อ } 0 \leq x < 1, \\ x-1 & \text{เมื่อ } x \geq 1. \end{cases} \quad (2.58)$$

โดยใช้วิธีการลงโทษ พร้อมอภิปรายว่า (1) กำหนดฟังก์ชันลงโทษเป็นอะไร (2) ฟังก์ชันจุดประสงค์ที่รวมการลงโทษเป็นอะไร (3) เกรเดียนต์ของฟังก์ชันจุดประสงค์ เกรเดียนต์ของฟังก์ชันลงโทษ และเกรเดียนต์ของฟังก์ชันที่ถูกลงโทษ (ฟังก์ชันจุดประสงค์ที่รวมการลงโทษเข้าไปแล้ว) เป็นอะไรบ้าง และ (4) คำตอบของปัญหาคืออะไร และเลือกค่าลากรานจ์พารามิเตอร์อย่างไร ยกตัวอย่างค่าที่ทำงานได้ดู คำใบ้ อนุพันธ์ของฟังก์ชันต่อเนื่องเป็นช่วง สามารถหาได้ในแต่ละช่วง.

หมายเหตุ ฟังก์ชัน  $g$  เป็นฟังก์ชันต่อเนื่องเป็นช่วง (piecewise continuous function) ซึ่งเป็นกรณีพิเศษ ที่ ณ จุดดีที่สุด ค่าเกรเดียนต์อาจจะไม่เป็นศูนย์.

## แบบฝึกหัดการเขียนโปรแกรม

### แบบฝึกหัดไพธอนแก่น

แบบฝึกหัดต่อไปนี้ ทบทวนการเขียนโปรแกรมด้วยภาษาไพธอนพื้นฐาน โดยยังไม่แนะนำให้เรียกใช้ module อื่น ตอนนี้.

### แบบฝึกหัด 2.5

จะเขียนฟังก์ชัน **add\_matrix** ที่รับเมทริกซ์สองตัว เป็นอาร์กิวเมนต์ ซึ่งแต่ละตัวเป็นลิสต์ที่มีโครงสร้างสองลำดับมิติ (เหมือนเมทริกซ์) ตรวจสอบ ว่าขนาดของเมทริกซ์เท่ากัน ทำการบวกเมทริกซ์ และรีเทิร์นผลลัพธ์ออกมาในรูปลิสต์ ที่มีโครงสร้างแบบเดียวกัน

### แบบฝึกหัด 2.6

จะเขียนฟังก์ชัน **mult\_matrix** ที่รับเมทริกซ์สองตัว เป็นอาร์กิวเมนต์ ซึ่งแต่ละตัวเป็นลิสต์ที่มีโครงสร้างสองลำดับมิติ (เหมือนเมทริกซ์) ตรวจสอบ ว่าขนาดของเมทริกซ์สามารถทำ การคูณเมทริกซ์ได้ ทำการคูณเมทริกซ์ และรีเทิร์นผลลัพธ์ออกมาในรูปลิสต์ ที่มีโครงสร้างลักษณะเดียวกัน

### แบบฝึกหัด 2.7

การหาเมทริกซ์ผกผัน ด้วยวิธีเกาส์约ร์แดน (Gauss Jordan method) ทำได้โดยขั้นตอนต่อไปนี้. เมื่อต้องการหา  $\mathbf{A}^{-1}$  ของเมทริกซ์  $\mathbf{A} \in \mathbb{R}^{n \times n}$

- ขยายเมทริกซ์  $\mathbf{A}$  ด้วยเมทริกซ์เอกลักษณ์. เมทริกซ์ขยาย  $\mathbf{A}' \in \mathbb{R}^{n \times 2n}$  จะเป็น

$$\mathbf{A}' = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{bmatrix} \quad (2.59)$$

2. ดำเนินปฏิบัติการเชิงเส้นกับแถวของ  $\mathbf{A}'$  จนส่วนหน้ากลা�ยเป็นเมตริกซ์เอกลักษณ์ ดังนี้

$$\mathbf{B}' = \begin{bmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1n} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \quad (2.60)$$

รายละเอียดการดำเนินปฏิบัติการ คือ (1) ที่แถว  $i$  เพื่อทำให้ส่วนประกอบแนวทะแยงมุมเป็นหนึ่ง ดำเนินการ  $r_{ij} = a_{ij}/a_{ii}$  สำหรับ  $j = 1, \dots, 2n$  เมื่อ  $r_{ij}$  คือค่าส่วนประกอบภายหลังการคำนวณ ที่ตำแหน่ง  $(i, j)$ . (2) ที่แถวอื่น ๆ แถว  $k \neq i$  เพื่อทำให้ส่วนประกอบ ตำแหน่ง  $(k, i)$  นอกแนว ทะแยงมุมเป็นศูนย์ ดำเนินการ  $s_{kj} = a_{kj} - r_{ij} \cdot a_{ki}$  สำหรับ  $j = 1, \dots, 2n$  เมื่อ  $s_{ij}$  คือค่า ส่วนประกอบภายหลังการคำนวณที่ตำแหน่ง  $(i, j)$ . (3) ดำเนินการเข่นนี้ โดยทำทุกแถว จากแถว  $i = 1, \dots, n$  และผลลัพธ์จะได้  $\mathbf{B}'$ .

3. ส่วนหลังของ  $\mathbf{B}'$  (ส่วนที่  $(n+1)^{th}$  จนถึงส่วนที่  $n^{th}$ ) คือเมตริกซ์ผกผันของ  $\mathbf{A}$ .

จากโปรแกรมตัวอย่างในรายการ 2.1 จะเขียนโปรแกรมไฟรอน เพื่อหาเมตริกซ์ผกผันด้วยวิธีเกาส์จอร์เดน และทดสอบผลลัพธ์ที่ได้ โดยใช้การคูณเมตริกซ์ จากแบบฝึกหัด 2.6 และศึกษาโปรแกรม และอภิปรายข้อ จำกัด และกรณีที่อาจเสียงทำให้โปรแกรมรันผิดพลาด พร้อมเสนอการปรับปรุงแก้ไข. คำใบ้ การหารด้วยศูนย์ จะทำให้โปรแกรมล้มไปได้.

รายการ 2.1: ตัวอย่างโปรแกรม วิธีหาเมตริกซ์ผกผันด้วยวิธีเกาส์จอร์เดน

---

```

1 def gauss_jordan(mat_in):
2     ...
3     Do Gauss-Jordan Matrix Inversion.
4     A: n x n matrix in a list of lists.
5     Return: Aaug, Ainv
6         Aaug is the final augmented matrix.
7         if the front part of Aaug appears to be identity,
8             Ainv is a valid matrix inversion of A.
9         ...
10    # Get dimension
11    n = len(mat_in)
12
13    # Copy a matrix

```

```

14     A = []
15     for i in range(n):
16         A.append(mat_in[i].copy())
17
18     # Make an identity matrix I
19     Imat = [[1 * (i == j) for j in range(n)] for i in range(n)]
20
21     # Augment A with I
22     for i in range(n):
23         A[i].extend(Imat[i])
24
25     # Get the front part of A' to identity
26     for i in range(n):
27         # Make a diagonal element 1
28         anchor = A[i][i]
29         for j in range(i, 2 * n):
30             A[i][j] /= anchor
31
32         # Make an off-diagonal 0
33         for k in range(n):
34             if k != i:
35                 target = A[k][i]
36                 for j in range(2 * n):
37                     if target != 0:
38                         A[k][j] = A[k][j] - A[i][j] * target
39
40     Ainv = []
41     for i in range(n):
42         Ainv.append(A[i][n:])
43
44     return A, Ainv
45
46 if __name__ == '__main__':
47     M0 = [[4, 2, 8], [3, 0, 9], [7, 5, 6]]
48     print('A'); print(M0)
49
50     M1, M2 = gauss_jordan(M0)
51
52     print('Inverse of A'); print(M2)

```

---

## แบบฝึกหัดเพื่อสอนด้วยนัมไพ

เนื่องจากโปรแกรมการรู้จำรูปแบบ ใช้การคำนวณเชิงเลข (numerical computation) อย่างมาก ดังนั้น การใช้มอดูลการคำนวณเชิงเลขประกอบจึงมีประโยชน์มาก เพื่อช่วยลดภาระ และเพื่อจะได้ทุ่มเทไปที่ศาสตร์ของ การรู้จำรูปแบบ และการเรียนรู้ของเครื่องได้เต็มที่มากขึ้น. หัวข้อนี้ แนะนำมอดูลคำนวณเชิงเลข นัมไพ (Numpy) และแมทพล็อตลิบ (Matplotlib) เพื่อช่วยการคำนวณเชิงเลข และการนำไปสร้างภาพให้เห็น (visualization)

นอกจากมอดูลนัมไพ และแมทพล็อตลิบ มีมอดูลที่มีประโยชน์ และนิยมใช้ สำหรับงานรู้จำรูปแบบอีกมาก ซึ่ง ตำนานี้ จะได้แนะนำมอดูลอื่น ๆ อีก ตามเนื้อหาที่เหมาะสมต่อไป

ตัวอย่างโปรแกรมในแบบฝึกหัดต่อไป จะนิยมนำเข้ามอดูล numpy และ matplotlib ด้วยคำสั่ง

```
import numpy as np
from matplotlib import pyplot as plt
```

ซึ่งจะนำเข้ามอดูล numpy และ matplotlib.pyplot. มอดูล pyplot เป็นมอดูลย่อย เพื่อใช้ในการวาดกราฟ. การนำเข้านั้น ได้ตั้งชื่อใหม่ให้กับมอดูลทั้งสองเพื่อความสะดวก เป็น np และ plt.

## แบบฝึกหัด 2.8

จะสร้างตัวแปร a1 เป็น np.array จาก list เช่น [2, 4, 8, 9] และเรียนรู้คำสั่งต่อไปนี้ คำสั่ง len(a1) คำสั่ง a1.shape คำสั่ง type(a1) คำสั่ง a1.tolist() แล้วสร้าง a2 เป็น np.array จาก list สອมิติ เช่น [[2, 4], [8, 9]] และเรียนรู้คำสั่งแบบเดียวกัน รวมทั้ง ทดลอง และอภิปรายผลของคำสั่งตัดส่วน (สังเกตชนิด และสัดส่วน ลอง type และ shape) เช่น คำสั่ง a2[0] คำสั่ง a2[0, :] คำสั่ง a2[0][1] และคำสั่ง a2[0,1].

## แบบฝึกหัด 2.9

จากคำสั่งต่อไปนี้ ทดลอง ตัดแปลง สังเกต และอภิปรายสิ่งที่ได้เรียนรู้.

- คำสั่ง v1 = np.array([[1,2,3,4]])
- คำสั่ง v2 = np.array([[2,4,1,-1]])
- คำสั่ง v1 + 5 เปรียบเทียบกับคำสั่ง v1 + v2 และคำสั่ง v1 + v2.transpose()

- คำสั่ง `v1 * 5` เปรียบเทียบกับคำสั่ง `v1 * v2` และคำสั่ง `v1 * v2.transpose()`
- ทดลองและ比べยับเทียบ การคูณด้วยคำสั่งต่าง ๆ ได้แก่ คำสั่ง `v1*v2` คำสั่ง `np.dot(v1,v2)` คำสั่ง `np.multiply(v1,v2)` คำสั่ง `np.matmul(v1,v2)` รวมถึงทดลองเปลี่ยนสัดส่วนของ `v1` และ `v2` เป็นอย่างอื่น (รวมถึงเป็นเมตริกซ์ และเทนเซอร์)

### แบบฝึกหัด 2.10

จงทดลองคำสั่งการหาเมทริกซ์ผกผัน `np.linalg.inv` และหาเมทริกซ์ผกผันต่อไปนี้ และ比べยับเทียบผลกับแบบฝึกหัด 2.7.

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 2 & 4 \\ 4 & 0 & 8 \\ 7 & 3 & 5 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} -1 & 2 & 4 \\ 4 & 1 & -1 \\ 9 & 3 & 8 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 4 & 0 & 4 \\ 1 & 2 & 3 \\ 7 & 3 & 10 \end{bmatrix}$$

### แบบฝึกหัด 2.11

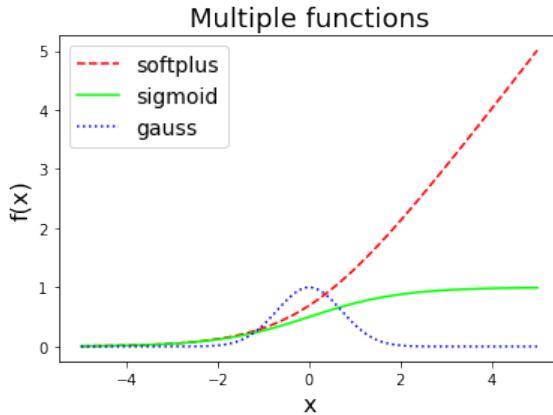
จงหาค่าและเวกเตอร์ลักษณะเฉพาะของเมทริกซ์ในแบบฝึกหัด 2.10.

คำໃບ້ ພັກໜັນ `numpy.linalg.eig` ເຮັດໃຫ້ ເລແພຄ (LAPACK) ຈຶ່ງເປັນມອດຸລີ່ໄດ້ຮັບຄວາມເຂື້ອສູ່ໃຈໃນການคำນວນค่าและเวກเตอร์ลักษณะเฉพาะ. ເລແພຄ ໃຊ້ວິທີคำນວນດ້ວຍການແຍກຕ້ວປະກອບຄ່າເອກຮູ້ານ (singular value decomposition) ທີ່ມີປະສິທິກາພາກກວ່າ ຕ້ວອຍ່າງວິທີคำນວນທີ່ກີ່ປະໄຍດ້ໃຫ້ວ່າຂອງ 2.1.

### แบบฝึกหัด 2.12

จงเขียนໂປຣແກຣມ ເພື່ອວາດກາຟຂອງພັກໜັນບາກອ່ອນ  $f(x) = \log(1 + e^x)$  ພັກໜັນຊືກມອຍດີ  $f(x) = 1/(1 + e^{-x})$  ແລະ ພັກໜັນເກາສເໜີຍນ  $f(x) = e^{-x^2}$  ໂດຍ ວາດກາຟທັງສາມໃຫ້ຢູ່ໃນກາຟເດືອກກັນ ດັ່ງແສດງໃນຮູບ 2.21.

คำใบ้ ดูคำสั่ง `np.linspace` คำสั่ง `plt.plot` โดยเฉพาะอาร์กิวเมนต์ เช่น `color=(1, 0.5, 0)` และ `linestyle='--'` คำสั่ง `plt.legend` คำสั่ง `plt.title` โดยเฉพาะอาร์กิวเมนต์ เช่น `fontsize=18`.



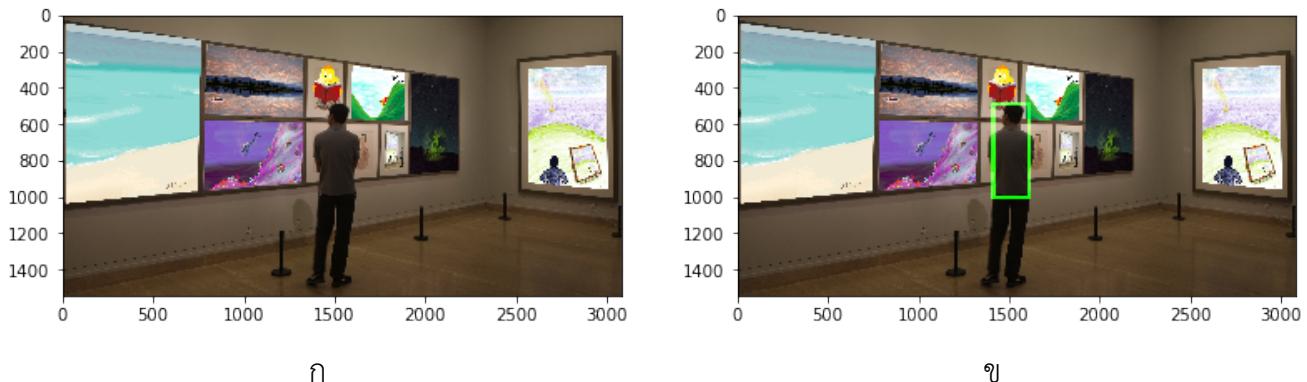
รูปที่ 2.21: กราฟแสดงพฤติกรรมของของฟังก์ชันบวกอ่อน ฟังก์ชันซิกมอยด์ และฟังก์ชันเกาส์เซียน

## แบบฝึกหัด 2.13

จะเขียนโปรแกรม โดยใช้มอดูล `matplotlib.pyplot` เพื่อนำเข้าภาพ (ซึ่งอาจจะเป็นไฟล์นามสกุล `.png`) และนำมาแสดง ดังต่อไปนี้ในรูป 2.22 (ภาพ ก). การดำเนินการนี้ ต้องอ่านภาพเข้ามาก่อน และจึงค่อยแสดงภาพออกหน้าจอ. คำสั่ง เช่น `img = plt.imread('t.png')` อ่านภาพจากไฟล์ชื่อ `'t.png'`. ตรวจสอบสัดส่วนข้อมูล และลักษณะข้อมูล เช่น ใช้คำสั่ง `print(img.shape)` ใช้คำสั่ง `print(img)`. สังเกต `plt.imread` อ่านภาพ ออกมาเป็นข้อมูลที่มีสัดส่วน 4 ลำดับขั้น สำหรับ ช่องสีแดง ช่องสีเขียว ช่องสีน้ำเงิน และช่องอัลฟ่า (alpha channel) และแต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1. การแสดงภาพทำได้ ด้วยคำสั่ง เช่น `plt.imshow(img)`.

หลังจากนั้นทดลองเปลี่ยนค่าพิกเซลของภาพ และสังเกตผล. ตัวอย่างในรูป 2.22 ภาพ ข ที่แสดงการเปลี่ยนค่าพิกเซล ด้วยคำสั่ง เช่น `img[1000:1010, 1400:1620, 1] = 1` จะเปลี่ยนค่าพิกเซลต่าง ๆ ที่ตำแหน่ง 1000 ถึง 1009 ตามแนวตั้ง และตำแหน่ง 1400 ถึง 1619 ตามแนวโนน ในช่องสีที่หนึ่ง (ช่องสีเขียว) โดยเปลี่ยนค่าเป็นหนึ่ง (ค่ามากที่สุด). เมื่อเปลี่ยนค่าเสร็จแล้ว และนำข้อมูลมาแสดงดูใหม่ จะเห็นสีเขียวปรากฏขึ้นมา (ภาพ ข มีการเปลี่ยนค่าลักษณะนี้ศิริรัง สำหรับสีเส้นที่เห็นแสดงเป็นกรอบ). หมายเหตุ ถ้ากำหนดค่าใหม่ให้กับพิกเซลแล้ว ภาพนั้นจะมีตำแหน่งนั้นคือ เส้นสีเขียวจะไม่สามารถถูกลบออกได้ (แต่ถูกเขียนทับได้). หากต้องการเก็บภาพต้นฉบับไว้ ให้ทำสำเนาไว้ก่อนที่จะมีการดัดแปลงภาพ เช่น

`marked = img.copy()` และดำเนินการเปลี่ยนค่าพิกเซลที่สำเนาแทน.



รูปที่ 2.22: ภาพ ก เป็นภาพต้นฉบับ. ภาพ ข ดัดแปลงจาก ภาพ ก โดยกำหนดค่าความเข้มของพิกเซลใหม่

### แบบฝึกหัด 2.14

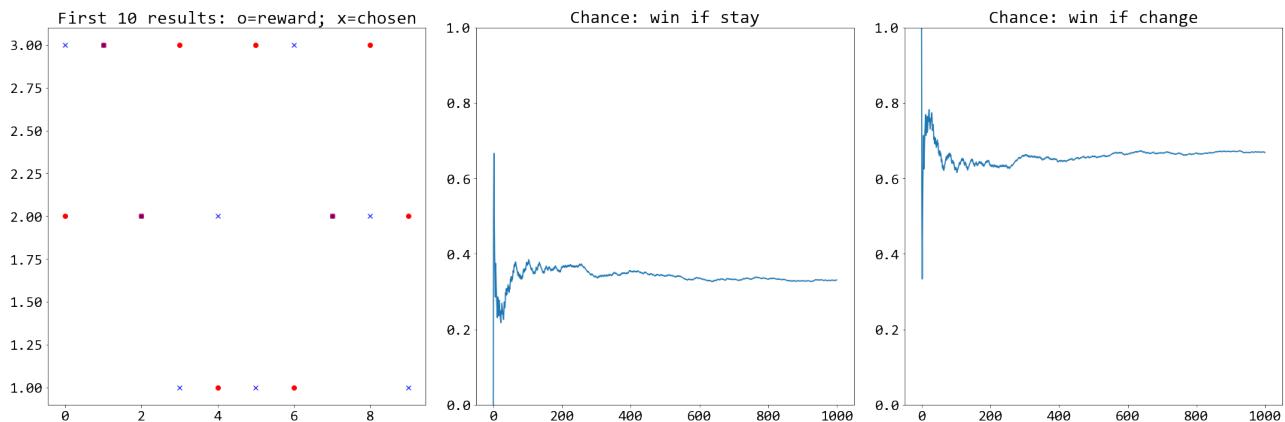
จากค่าฟังก์ชัน  $g(v) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$  จะเขียนโปรแกรม เพื่อวัดภาพคอนทัวร์ และภาพสามเหลี่ยมมิติ ดังรูป 2.14.

คำใบ้ คำสั่ง `plt.contour` จะวัดภาพคอนทัวร์ จากค่าพิกัดสามเหลี่ยมมิติ ดังนั้นเพื่อจาวัดภาพคอนทัวร์ จะต้องคำนวณค่าฟังก์ชัน  $g$  ที่  $v_1$  และ  $v_2$  ต่าง ๆ ทั้งหมดที่ครอบคลุมบริเวณที่จะวัด. ตัวอย่าง เช่น หากต้องการวัดครอบคลุมบริเวณ  $v_1$  มีค่า 1 ถึง 5 และ  $v_2$  มีค่า 2 ถึง 6 โดยวัดให้มีความละเอียดมิติละ 40 จุด จะต้องคำนวณค่า ฟังก์ชัน  $g$  ที่  $v_1$  และ  $v_2$  ในบริเวณนี้ออกมา 1600 ค่า. คำสั่ง `np.meshgrid` อาจช่วยลดภาระการจับคู่ค่า  $v_1$  และ  $v_2$  ได้.

### แบบฝึกหัด 2.15

จากปัญหามอนตี้霍ล ในหัวข้อ 2.2 จะเขียนโปรแกรมเพื่อจำลองสถานการณ์ และแสดงให้เห็นว่า โอกาสที่ผู้เข้าแข่งขันจะได้รางวัล หากเลือกที่จะเปลี่ยนประตูเป็น 2/3 และโอกาสที่ผู้เข้าแข่งขันจะได้รางวัล หากเลือกประตูเดิมเป็น 1/3.

อภิปรายโปรแกรม และผลการทดลอง พร้อมออกแบบวิธีนำเสนอผล ให้ประจักษ์ชัดเจน. รูป 2.23 แสดงตัวอย่างวิธีนำเสนอผล ซึ่งจากรูปจะเห็นว่า เมื่อจำนวนช้ำมากขึ้น โอกาสที่จะชนะเมื่อเลือกประตูเดิม (ภาพกลาง) จะประมาณ 0.33 หรือ 1/3 ในขณะที่ ถ้าเปลี่ยนไปประตูใหม่ โอกาสจะประมาณ 0.67 หรือ 2/3.



**รูปที่ 2.23:** ผลลัพธ์จากการจำลองสถานการณ์ปัญหามอนตี้霍ล. ภาพซ้าย แสดงผลลัพธ์ 10 ครั้งแรก. สัญลักษณ์ ‘o’ สีแดงแทนรางวัล สัญลักษณ์ ‘x’ สีน้ำเงินแทนประตูที่ผู้เข้าแข่งขันเลือก เช่น ครั้งแรกสุด (ครั้งที่ 0) ผู้เข้าแข่งขันเลือกประตูที่สาม แต่รางวัลอยู่ประตูที่สอง. ประตูที่พิธีกรเปิดไม่ได้แสดงในภาพ. ภาพกลางแสดงอัตราส่วนสะสมของจำนวนครั้งที่ผู้เข้าแข่งขันชนะได้รางวัล เมื่อเลือกประตูเดิม. ภาพขวาแสดงอัตราส่วนสะสมของจำนวนครั้งที่ผู้เข้าแข่งขันชนะได้รางวัล เมื่อตัดสินใจเปลี่ยนไปประตูใหม่.

คำใบ้ คำสั่ง `np.random.choice` ใช้สุ่มเลือกค่าจากลิสต์. รางวัลอาจสุ่มให้อยู่ประตูไหนก็ได้. ประตูที่ผู้เข้าแข่งขันเลือก ก็อาจอยู่ประตูไหนก็ได้. แต่ประตูที่พิธีกรเลือกเปิด จะเป็นประตูที่มีรางวัลไม่ได้ หรือจะเป็นประตูที่ผู้เข้าแข่งขันเลือกไม่ได้. การเลือกประตูของพิธีกร ต้องทำหลังจากการเลือกประตูรางวัล และการเลือกประตูของผู้เข้าแข่งขัน.

## แบบฝึกหัด 2.16

ผลการคำนวณค่าสถิติจากข้อมูลที่จำกัด จะมีความไม่แน่นอนอยู่<sup>15</sup> ในแท่งที่ว่า หากสุ่มกลุ่มข้อมูลออกมากใหม่ แล้วผลการคำนวณอาจจะเปลี่ยนไป.

หากคำนวณค่าเฉลี่ยของกลุ่มข้อมูลที่สุ่มมาจากการแจกแจงเดียวกัน และนำค่าเฉลี่ยของแต่ละการสุ่มมาวิเคราะห์ จะพบว่า[76] ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย

$$\text{SD}(\mu_N) = \sqrt{\text{var}\left[\frac{1}{N} \sum_{n=1}^N x_n\right]} = \frac{\sigma}{\sqrt{N}} \quad (2.61)$$

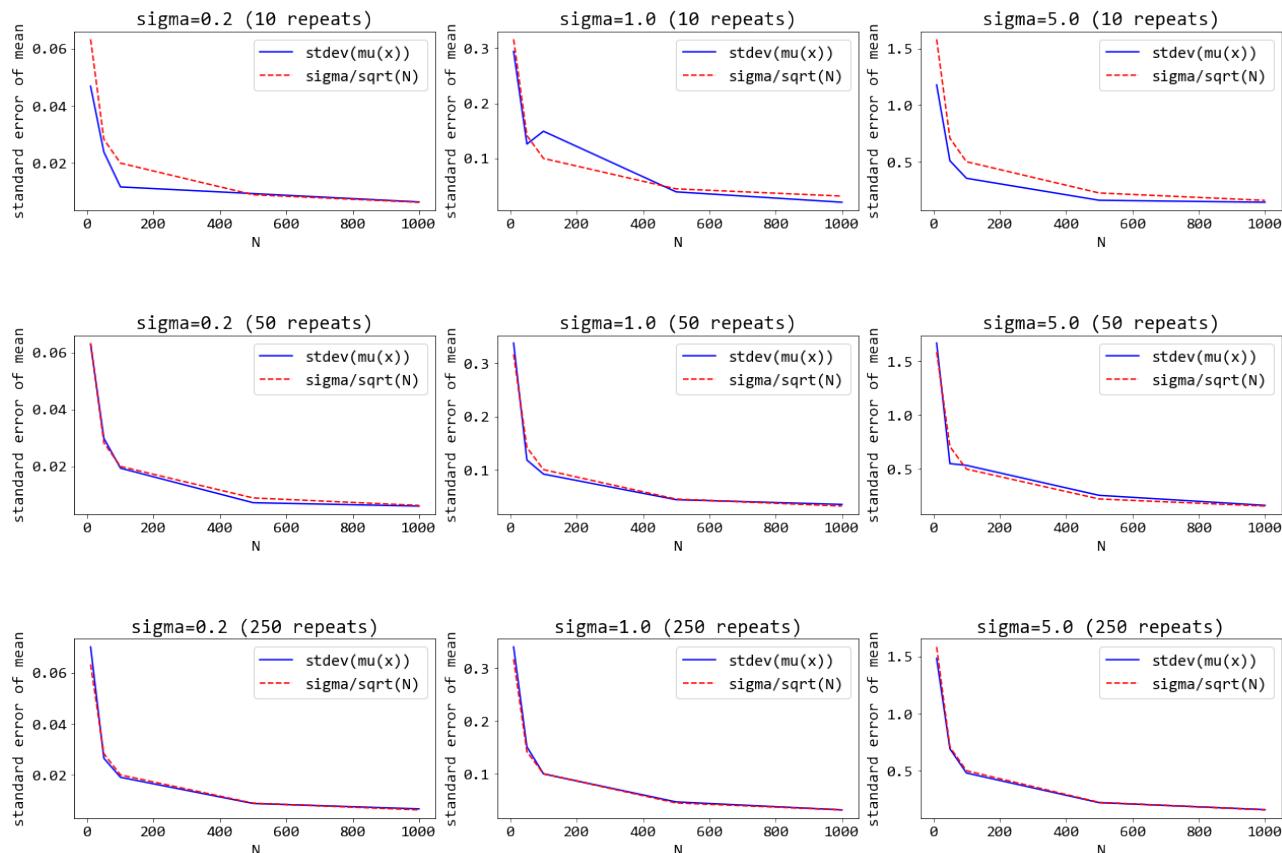
เมื่อ  $\mu_N$  คือค่าเฉลี่ยที่ของกลุ่มขนาด  $N$  และ  $x_n$  คือจุดข้อมูลในกลุ่มที่สุ่มขึ้นมา และ  $\sigma$  คือ ค่าเบี่ยงเบนมาตรฐานของการแจกแจง.

จากทฤษฎีข้างต้น ยิ่งกลุ่มข้อมูลมีขนาดใหญ่ ( $N$  มีค่ามาก) ค่าเฉลี่ยที่คำนวณได้จะยิ่งมีความแม่นยำใกล้เคียงกับค่าเฉลี่ยจริงของการแจกแจงเท่านั้น.

<sup>15</sup>เนื้อหาในแบบฝึกหัดนี้ ได้รับแรงบันดาลใจและอิทธิพลหลัก ๆ จาก [76].

จงเขียนโปรแกรม เพื่อแสดงในเห็นว่า สมการ 2.61 เป็นจริง.

ตัวอย่างการนำเสนอ แสดงในรูป 2.24. (ไม่จำเป็นต้องทำตามตัวอย่างนี้.) ผลที่ได้ มาจากการจำลอง (simulation) โดย กำหนดการแจกแจงแบบเก้าส์เชียน ที่มีค่าเฉลี่ยเป็นศูนย์ และค่าความแปรปรวน  $\sigma^2$  และ จำลองการสุ่มกลุ่มจำนวน  $N$  ขึ้นมา. ตัวอย่างนี้ ทดลองค่า  $\sigma$  สามค่าได้แก่ 0.2, 1.0, และ 5.0 แสดงด้วย ภาพชั้ย กลาง และขวา ตามลำดับ.



รูปที่ 2.24: ความสัมพันธ์ของความคลาดเคลื่อนการคำนวณกับขนาดข้อมูล. แต่ละภาพ แสดง ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ที่คำนวณจากกลุ่มข้อมูลที่สุ่มขึ้นมา (เส้นทึบ สีน้ำเงิน) กับค่าที่คำนวณทางทฤษฎี (เส้นประ สีแดง). แกนนอน แสดงจำนวนจุดข้อมูล ในกลุ่ม. ภาพต่าง ๆ ทางซ้าย แสดงผล เมื่อการแจกแจงของข้อมูลมี  $\sigma = 0.2$  ภาพต่าง ๆ แนวกลาง  $\sigma = 1$  และภาพต่าง ๆ ทางขวา  $\sigma = 5$ . ภาพต่าง ๆ ในแนวนอน แสดงผลเมื่อทำซ้ำ 10 ครั้ง และกลาง ทำซ้ำ 50 ครั้ง และขวา ทำซ้ำ 250 ครั้ง.

จากนั้น คำนวณค่า  $\mu_N$  ของกลุ่ม โดยทำซ้ำ 10 ครั้ง (ผลแสดงในภาพต่าง ๆ แนวนอน) ทำซ้ำ 50 ครั้ง (กลาง) และทำซ้ำ 250 ครั้ง (ขวา). ค่าเฉลี่ยของแต่ละซ้ำจะถูกนำมาคำนวณค่าเบี่ยงเบนมาตรฐาน. นั่นคือ  $\widehat{SE} = \sqrt{\frac{\sum_{r=1}^R m_r - \bar{m}}{R-1}}$  เมื่อ  $\widehat{SE}$  คือค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย และ  $R$  คือ จำนวนซ้ำ. ส่วน  $m_r = \frac{1}{N} \sum_{n=1}^N x_n$  คือ ค่าเฉลี่ยของกลุ่มของซ้ำที่  $r^{th}$  และ  $\bar{m} = \frac{1}{R} \sum_{r=1}^R m_r$ . ค่า  $\widehat{SE}$  (แทนด้วย เส้นทึบ สีน้ำเงิน และสัญกรณ์ `stdev(mu(x))` ในภาพ) เป็นค่าที่คำนวณจากข้อมูล ส่วน  $\sigma/N$  เป็นค่าที่คำนวณ จากทฤษฎี (แทนด้วย เส้นประสีแดง และสัญกรณ์ `sigma/sqrt(N)` ในภาพ). ในแต่ละภาพ แกนตั้ง

แสดงค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย และแคนนอน แสดงค่าขนาดของกลุ่ม  $N$  ซึ่งในตัวอย่างใช้ 10, 50, 100, 500, และ 1000.

ผลที่ได้ แสดงในเห็นว่า (1) เมื่อ  $N$  ขนาดใหญ่ขึ้น ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ยมีค่าเล็กลง. (2) ถ้า  $\sigma$  ของการแจกแจง มีค่าใหญ่ ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ก็จะมีค่าใหญ่ตามสัดส่วน. (3) ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ที่ได้จากการคำนวนกับข้อมูล เป็นไปตามทฤษฎี และจะเห็นความสอดคล้องได้ชัดเจนมากขึ้น เมื่อจำนวนข้าเพิ่มขึ้น.

หมายเหตุ ในทางปฏิบัติ เราไม่รู้ค่า  $\sigma$  ของการแจกแจง. ดังนั้น การประเมินความคลาดเคลื่อนของการคำนวน เมื่อกลุ่มข้อมูลมีขนาดจำกัด จะไม่สามารถทำได้สะอาดเช่นนี้. ในทางปฏิบัติ เราจะสามารถประเมินความคลาดเคลื่อน ต่อขนาดข้อมูลได้อย่างไร. หรือ ที่น่าสนใจกว่าคือคำถามว่า ทำอย่างไร เราถึงจะสามารถประเมิน ได้ว่าภาระกิจที่กำลังทำอยู่ ต้องการจำนวนข้อมูลเท่าไร. อภิปราย พร้อมให้เหตุผลประกอบ.

### แบบฝึกหัด 2.17

จากตัวอย่างปัญหาค่าน้อยที่สุด  $\min_x f(x)$  เมื่อ  $f(x) = -e^{-(x-5)^2}$  ในหัวข้อ 2.3 เราสามารถนำวิธีลงเกรเดียนต์สามารถนำไปเขียนโปรแกรมได้ง่าย ๆ โดยเริ่มจาก เตรียมเกรเดียนต์ของฟังก์ชันจุดประสงค์. นั่นคือ  $\nabla f(x) = -e^{-(x-5)^2} \cdot (-2x + 10)$  และไพรอนฟังก์ชัน **grad** ดังแสดงในรหัสโปรแกรมข้างล่าง จะทำหน้าที่คำนวนเกรเดียนต์นี้

```
def grad(x):
    return -np.exp(-(x - 5)**2) * (-2*x + 10)
```

และหลังจากมีเกรเดียนต์แล้ว โปรแกรมวิธีลงเกรเดียนต์ก็สามารถทำงานได้ ดังแสดงในรายการ 2.2. โปรแกรมเลือกใช้ค่าขนาดก้าว **step\_size** เป็น 0.5 (ตัวแปร  $\alpha$  ในสมการ 2.46) และใช้เงื่อนไขจบเป็นจำนวนรอบสูงสุด ซึ่งในที่นี้ใช้เป็น 8 รอบ ที่ระบุลงไปให้ลูปเลย ด้วย **for i in range(8)**

ค่าเริ่มต้นของตัวแปรตัดสินใจ เลือกกำหนดให้เป็น 6.5 ในบรรทัดที่สอง และคำสั่งในบรรทัดที่สี่ คือการคำนวนสมการ 2.46.

รายการ 2.2: ตัวอย่างโปรแกรม วิธีลงเกรเดียนต์อย่างง่าย ๆ

---

```
1 step_size = 0.5
2 x = 6.5
3 for i in range(8):
4     x = x - step_size * grad(x)
5 print('x = ', x, '; grad = ', grad(x))
```

---

จากโปรแกรมนี้ ทดลองรัน และเปรียบผลลัพธ์ที่ได้กับตัวอย่าง

### แบบฝึกหัด 2.18

การใช้งานวิธีลงเกรเดียนต์ในทางปฏิบัติ นิยมเพิ่มเงื่อนไขการจับ เพื่อให้สามารถเลือกจำนวนรอบสูงสุดไว้มาก ๆ ก่อน โดยไม่ต้องกังวลว่า จะเสียเวลา\_rันโดยไม่จำเป็น (ดูแบบฝึกหัด 2.19 ประกอบ)

จากแบบฝึกหัด 2.17 โปรแกรมวิธีลงเกรเดียนต์ง่าย ๆ ที่แสดงในรายการ 2.2 สามารถเพิ่มเงื่อนไขการจบทิ้งไปได้ ดังแสดงในรายการ 2.3. เงื่อนไขจำนวนรอบสูงสุด ควบคุมได้โดยผ่านตัวแปร **Nmax**. โปรแกรมตัวอย่างนี้ใช้เงื่อนไขความคลาดเคลื่อนยินยอม วัดค่าผ่าน **eps** โดยถ้า **eps** น้อยกว่าหรือเท่ากับค่าที่ยินยอมได้ ก็จะจบการคำนวนทันที. ค่าที่ยินยอมได้ (tolerance) กำหนดผ่าน **tol**.

---

รายการ 2.3: ตัวอย่างโปรแกรม วิธีลงเกรเดียนต์ที่มีเงื่อนไขการจบทิ้งความคลาดเคลื่อนที่ยินยอมได้

---

```

1 step_size = 0.5
2 Nmax = 500
3 tol = 0.00001
4 x = 6.5
5 for i in range(Nmax):
6     x = x - step_size * grad(x)
7     print(i, ': x = ', x, '; grad = ', grad(x))
8
9     eps = np.abs(grad(x))
10    if eps <= tol:
11        print('Reach termination criteria.')
12        break

```

---

สังเกตว่า โปรแกรมนี้ มีการคำนวนที่ซ้ำซ้อนมาก ในแต่ละรอบ เช่น **grad(x)** มีการคำนวนค่าเดียวกันถึงสามครั้ง. ความซ้ำซ้อนนี้สามารถลดลงได้ โดยการเก็บค่าไว้ในตัวแปร เช่น

```

...
x = 6.5
gradx = grad(x)
for i in range(Nmax):
    x = x - step_size * gradx
    gradx = grad(x)
    print(i, ': x = ', x, '; grad = ', gradx)
    eps = np.abs(gradx)
    if eps <= tol:
        print('Reach termination criteria.')
        break

```

โดย . . . แทนโค้ดต่าง ๆ ที่ละไว้ไม่ได้หมายถึงการพิมพ์จุดสามครั้งเข้าไปในโปรแกรม. ตัวแปร  $\text{gradx}$  ใช้เก็บค่าเกรเดียนต์ที่คำนวณไว้แล้ว และค่าเกรเดียนต์ที่คำสั่งปรับค่า  $x = x - \text{step\_size} * \text{gradx}$  จะเป็นคุณลักษณะค่ากับที่คำสั่ง  $\text{print}$  และที่กำหนดค่า  $\text{eps}$  ตั้งนั้นตำแหน่งของ  $\text{gradx} = \text{grad}(x)$  จึงต้องอยู่หลังการปรับค่า  $x$  และเพื่อให้รอบคำนวณแรกสามารถทำได้ จึงต้องมีคำสั่ง  $\text{gradx} = \text{grad}(x)$  อยู่ก่อนเข้าสู่ปัจจุบัน.

จากโปรแกรมในรายการ 2.3 จงทดลองรัน และเปรียบเทียบผล แล้วทดลองเปลี่ยนค่า  $\text{tol}$  เป็นค่าอื่น ๆ เช่น  $0, 1e-5, 1e-3$ .

คำถามเพิ่มเติม. ทำไมเงื่อนไขความคลาดเคลื่อน นั่นคือ  $\text{eps} <= \text{tol}$  ถึงเขียนด้วยการเปรียบเทียบน้อยกว่าหรือเท่ากับ ทดลองเปลี่ยนเป็นการเปรียบเทียบมากกว่า ได้แก่  $\text{eps} < \text{tol}$  และทดลองค่า  $\text{tol}$  ต่าง ๆ อีกครั้ง สังเกตและอภิปรายผล

## แบบฝึกหัด 2.19

จากแบบฝึกหัด 2.17 จงทดลองเปลี่ยนจุดเริ่มต้น (บรรทัดที่สอง) เป็นค่าต่าง ๆ เช่น  $4, 5, 6, 7, 8$ . ทดลองเพิ่มหรือลดจำนวนรอบสูงสุดถ้าจำเป็น ลองรันโปรแกรมในรายการ 2.2 ใหม่ แล้วสังเกตผลที่ได้ ว่าที่ค่าเริ่มต้นต่าง ๆ ต้องคำนวณกี่รอบ ถึงจะได้คำตอบ เช่น จากตัวอย่าง ถ้าใช้  $x = 6.5$  จะได้คำตอบที่รับที่เจ็ด (ให้ใช้ค่าขนาดกว้างเท่ากันหมวดเป็น  $0.5$  ก่อน)

อภิปรายถึงประเด็นปัญหาที่จะเกิด เมื่อนำโปรแกรมนี้ไปรันในทางปฏิบัติ แล้วทำการทดลองใหม่ โดยรันโปรแกรมในรายการ 2.3 และอภิปรายข้อดีของการใช้เงื่อนไขการจบ ที่มีในโปรแกรม 2.3.

## แบบฝึกหัด 2.20

เงื่อนไขความคลาดเคลื่อนยินยอม จะกำหนดค่าความคลาดเคลื่อนที่ยอมรับได้ ซึ่งอาจทำได้หลายวิธี เช่น ใช้ค่าความคลาดเคลื่อนของตัวแปร  $\varepsilon_v = \|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|$  ใช้ค่าความคลาดเคลื่อนของเกรเดียนต์  $\varepsilon_\nabla = \|\nabla g(\mathbf{v}^{(k+1)}) - \nabla g(\mathbf{v}^{(k)})\|$  หรือใช้ค่าความคลาดเคลื่อนของฟังก์ชันจุดประสงค์  $\varepsilon_g = |g(\mathbf{v}^{(k+1)}) - g(\mathbf{v}^{(k)})|$  หรือใช้ค่าความคลาดเคลื่อนข้างตันสมกัน. อภิปรายเงื่อนไขความคลาดเคลื่อนยินยอมแต่ละแบบ คำใบ้ พิจารณาสมการ 2.46 และความหมายของเกรเดียนต์.

## แบบฝึกหัด 2.21

จากตัวอย่าง ปัญหาค่าน้อยที่สุด เมื่อตัวแปรตัดสินใจเป็นเวกเตอร์  $\min_{\mathbf{v}} g$  เมื่อ  $\mathbf{v} = [v_1, v_2]^T$  และ  $g(\mathbf{v}) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$ . คล้ายกับแบบฝึกหัด 2.17 วิธีลงเกรเดียนต์ต้องการฟังก์ชัน

คำนวณค่าเกรเดียนต์ ซึ่งอาจทำได้โดย

```
def grad(u):
    assert type(u) == type(np.array([]))
    assert u.shape == (2,1)

    gu = g(u)
    gradu = gu * np.array([-2*u[0,0] - u[1,0] + 10,
                           [-4*u[1,0] - u[0,0] + 19]])
    return gradu
```

ฟังก์ชัน `grad` นี้ใช้คำสั่ง `assert` เพื่อจำกัดชนิดของอินพุต `u` เพื่อป้องกันปัญหาจากชนิดข้อมูล. ตัวแปร `u` แทนเวกเตอร์  $\mathbf{u} = [u_1, u_2]^T$ . ค่าส่วนประกอบ  $u_1$  และ  $u_2$  เข้าถึงได้โดย  $u[0,0]$  และ  $u[1,0]$  ตามลำดับ. นอกจ้านี้ ฟังก์ชัน `grad` ยังเรียกใช้ฟังก์ชันจุดประสงค์ ซึ่งเขียนได้ดังนี้

```
def g(u):
    assert type(u) == type(np.array([]))
    assert u.shape == (2, 1)

    loss = -np.exp(-53 - u[0,0] ** 2 - 2 * u[1,0] ** 2 - u[0,0] * u[1,0]
                  + 10 * u[0,0] + 19 * u[1,0])
    return loss
```

รายการ 2.4 แสดงตัวอย่างโปรแกรม วิธีลงเกรเดียนต์ เมื่อตัวแปรเป็นเวกเตอร์.

รายการ 2.4: ตัวอย่างโปรแกรม วิธีลงเกรเดียนต์ เมื่อตัวแปรเป็นเวกเตอร์

---

```
1 step_size = 0.4
2 Nmax = 1000
3 tol = 1e-6
4 v = np.array([[2.5], [3.5]])
5
6 losses = []
7 gradv = grad(v)
8 for i in range(Nmax):
9     v = v - step_size * gradv
10    gradv = grad(v)
11    loss = g(v)
12    losses.append(loss)
13
14    eps = np.linalg.norm(gradv)
15    if eps <= tol:
```

```

16     print('Reach termination at i={} with {}'.format(i, eps))
17     break
18
19     print('{}: v = [{:.3f} {:.3f}]; grad = [{:.4f} {:.4f}]'.←
20         format(i,v[0,0],v[1,0],gradv[0,0],gradv[1,0]))
21 print('{}: v = [{} {}]; g = {}'.format(i,v[0,0],v[1,0], loss))

```

---

สังเกตว่า โปรแกรมบันทึกค่าฟังก์ชันจุดประสงค์ หรือเรียกว่าอย่าง ๆ ว่า **ค่าสูญเสีย** (loss) ของทุกรอบการคำนวณไว้ใน **losses**. ค่าสูญเสียต่อรอบคำนวณ สามารถนำมาใช้ เพื่อตรวจสอบการทำงานแก้ปัญหาค่าน้อยที่สุดได้ ดังแสดงในภาพข้างบนของรูป 2.16. ภาพอื่น ๆ ในรูป 2.16 ก็จะสามารถทำได้ในแบบเดียวกันเพียงแต่ต้องบันทึกค่านั้น ๆ ขณะรันด้วย ซึ่งในที่นี้ ขอລາຍເປີ້ມໃຫ້ໂປຣແກຣມດູຈັບຂອນເກີນໄປ.

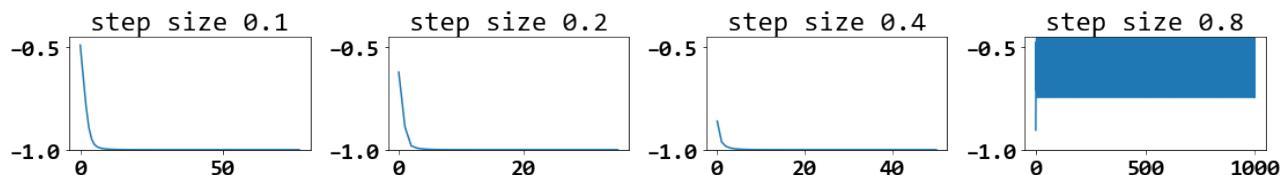
ลองเปรียบเทียบฟังก์ชัน **grad** กับการคำนวณหาค่าเกรเดียนต์ด้วยมือ และเปรียบเทียบໂປຣແກຣມในรายการ 2.4 เปรียบเทียบกับรายการ 2.3 และอภิปรายจุดแตกต่าง

ทดลองรันໂປຣແກຣມในรายการ 2.4 แล้วแก้ไขค่า **step\_size** รัน และตรวจสอบผลที่ได้ กับการคำนวณของตัวอย่างในหัวข้อ 2.3.

## แบบฝึกหัด 2.22

จากปัญหาในแบบฝึกหัด 2.21 ทดลอง เปลี่ยนค่าขนาดก้าว (ตัวแปร **step\_size**) ต่าง ๆ เช่น 0.1, 0.2, 0.4, 0.8 และสังเกตผลและอภิปราย และเตรียมหลักฐานเพื่อประกอบการอภิปราย เช่น รูป 2.25 รูป 2.26 และรูป 2.27. ทดลองค่าอภิมานพารามิเตอร์อื่น ๆ เพื่อยืนยันข้อสรุปที่ได้อภิปราย.

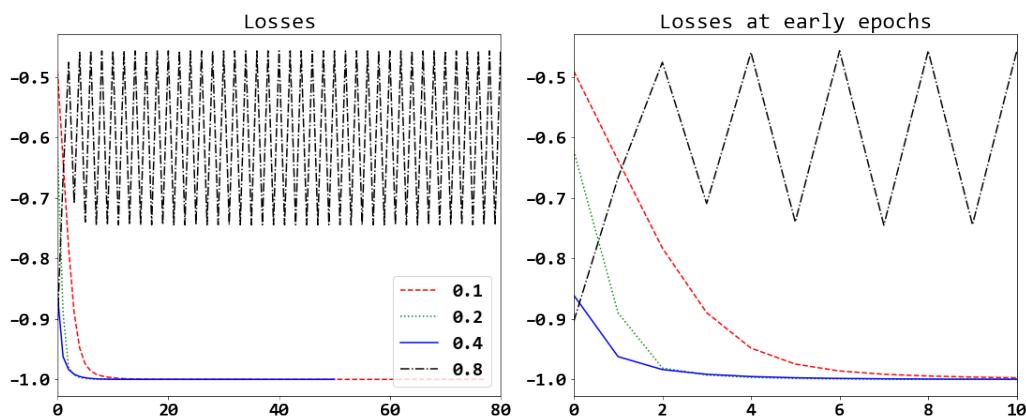
รูป 2.25 แสดงความก้าวหน้าของการแก้ปัญหา (ค่าสูญเสียต่อรอบคำนวณ) เมื่อใช้ค่าขนาดก้าวต่าง ๆ (ตามระบุเหนือภาพ) โดยค่าอภิมานพารามิเตอร์อื่น ๆ คือ ใช้จำนวนรอบสูงสุด 1000 รอบ ใช้ค่าตัวแปรติดสินใจเริ่มต้นเป็น  $[2.5, 3.5]^T$  และใช้ค่าความคลาดเคลื่อนยินยอม เป็น  $10^{-6}$ .



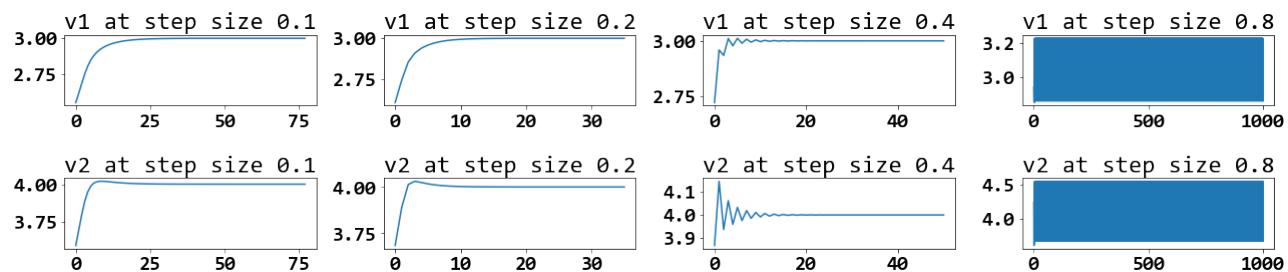
รูปที่ 2.25: ตัวอย่างแสดงผลการทำงานวิธีลิงเกรเดียนต์ เมื่อใช้ค่าขนาดก้าวต่าง ๆ.

รูป 2.26 แสดงความก้าวหน้าของการแก้ปัญหา เมื่อใช้ค่าขนาดก้าวต่าง ๆ ในภาพเดียวกัน โดยค่าขนาด

ก้าวที่ใช้ระบุด้วยสัญลักษณ์ ดังแสดงในกรอบคำอธิบายสัญลักษณ์ (legend). ภาพซ้ายแสดงในช่วง 80 รอบคำนวณแรก (เมื่อใช้ขนาดก้าวบางค่า การคำนวณจบก่อน 80 รอบ). ภาพขวาแสดงในช่วง 10 รอบคำนวณแรก เพื่อให้เห็นชัดเจนว่าที่ค่าขนาดก้าวเท่าใดถึงเข้าสู่คำตอบได้เร็วที่สุด (ค่าสูญเสียลดลงต่ำสุด ในรอบคำนวณที่น้อยที่สุด หมายถึงการถึงเข้าสู่คำตอบได้เร็วที่สุด). ตัวอย่างนี้ จะเห็นว่าขนาดก้าว 0.4 ช่วยให้วิธีลิงเกรเดียโนต์ถึงเข้าเร็วที่สุด และขนาดก้าวที่น้อยลง มีผลให้ถึงเข้าช้าลง. แต่หากใช้ขนาดก้าวที่ใหญ่เกินไป (เช่น 0.8 ในตัวอย่างนี้) อาจทำให้วิธีลิงเกรเดียโนต์ไม่สามารถถึงเข้าสู่คำตอบได้. รูป 2.27 แสดงค่าของตัวแปรตัดสินใจ  $v_1$  และ  $v_2$  สังเกตว่า เมื่อใช้ขนาดก้าว 0.1, 0.2, 0.4 ค่าของตัวแปรตัดสินใจถึงเข้าสู่ค่า 3 และ 4 ตามลำดับ โดย เมื่อใช้ขนาดก้าว 0.8 ค่าของ  $v_1$  และ  $v_2$  มีการส่ายอยู่บ้างก่อนส่ายน้อยลงจนถึงเข้าสู่คำตอบ. แต่เมื่อใช้ขนาดก้าว 0.8 ค่าของ  $v_1$  และ  $v_2$  แสดงการส่ายต่อเนื่องไปจนครบ 1000 รอบฝึกโดยไม่มีแนวโน้มจะถึงเข้าขนาดก้าว 0.8 เป็นขนาดก้าวที่ใหญ่เกินไปอย่างชัดเจน. อภิปราย พฤติกรรมนี้ พร้อมวิเคราะห์สาเหตุที่ทำให้น้อยอย่างสุด (เช่นรูป 2.15) เพื่อประกอบการอภิปราย.



รูปที่ 2.26: ผลการทำงานวิธีลิงเกรเดียนต์ ในรอบคำนวณตัน ๆ เมื่อใช้ค่าขนาดก้าวต่าง ๆ เมื่อวัดรวมกัน.



รูปที่ 2.27: ผลลัพธ์จากการวิธีลิงเกรเดียนต์ เมื่อใช้ค่าขนาดก้าวต่าง ๆ.

หมายเหตุ อย่างไรก็ตาม ค่าขนาดก้าวนี้ไม่จำเป็นต้องใช้เป็นค่าเดียวกันตลอดทุกรอบการคำนวณ อาจ

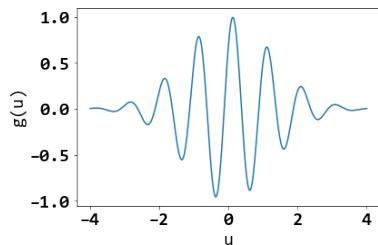
เปลี่ยนขนาดได้ตามความเหมาะสม เช่น อาจปรับให้ขนาดเล็กลง เมื่อจำนวนรอบคำนวณมาก ๆ ได้ หรืออาจใช้กลไกการปรับที่ซับซ้อนขึ้นได้ เช่น วิธีลิงเกรเดียนต์ชั้นที่สุด (steepest gradient descent method ดู [39] เพิ่มเติม สำหรับผู้ที่สนใจ).

### แบบฝึกหัด 2.23

จงแก้ปัญหา  $\min_u \cos(2\pi u - \frac{\pi}{4}) \cdot \exp\left(-\frac{u^2}{\pi}\right)$  ด้วยวิธีลิงเกรเดียนต์.

ทดลองค่าเริ่มต้นต่าง ๆ เช่น  $-2, -1, -0.7, 0, 0.12301636938191951, 0.5, 1, 1.5$ . เลือกค่าอภิมานพารามิเตอร์อื่น ๆ ให้เหมาะสม รันวิธีลิงเกรเดียนต์จนสำเร็จ และสังเกตผลลัพธ์ที่ได้จากการใช้ค่าเริ่มต้นต่าง ๆ. อภิปรายความสัมพันธ์ระหว่างค่าเริ่มต้นต่าง ๆ กับผลลัพธ์ที่ได้. เมื่อใช้ค่าเริ่มต้นเป็น  $0.12301636938191951$  ผลลัพธ์เป็นอย่างไร ทำไมถึงเป็นเช่นนั้น?

รูป 2.28 แสดงค่าฟังก์ชันจุดประสงค์  $g(u)$ . อภิปรายการใช้งานวิธีลิงเกรเดียนต์กับปัญหาลักษณะนี้ โดยเฉพาะในทางปฏิบัติ ที่มักไม่สามารถคาดการณ์ของฟังก์ชันจุดประสงค์ได้.



รูปที่ 2.28: ฟังก์ชันจุดประสงค์  $g(u) = \cos(2\pi u - \frac{\pi}{4}) \cdot \exp\left(-\frac{u^2}{\pi}\right)$ .

หมายเหตุ ปัญหานี้ในแบบฝึกหัดนี้ จะเรียกว่า ปัญหาหลายภาวะ (multi-modal problem) ซึ่งคือ ปัญหาค่าน้อยที่สุดที่มีค่าทำน้อยที่สุดท้องถิ่นหลายที่ และวิธีลิงเกรเดียนต์จะพบคำตอบที่ใกล้ที่สุด ที่ทิศทางเกรเดียนต์ของจุดเริ่มต้นขึ้นไป เมื่อใช้ขนาดก้าวเล็กพอก.

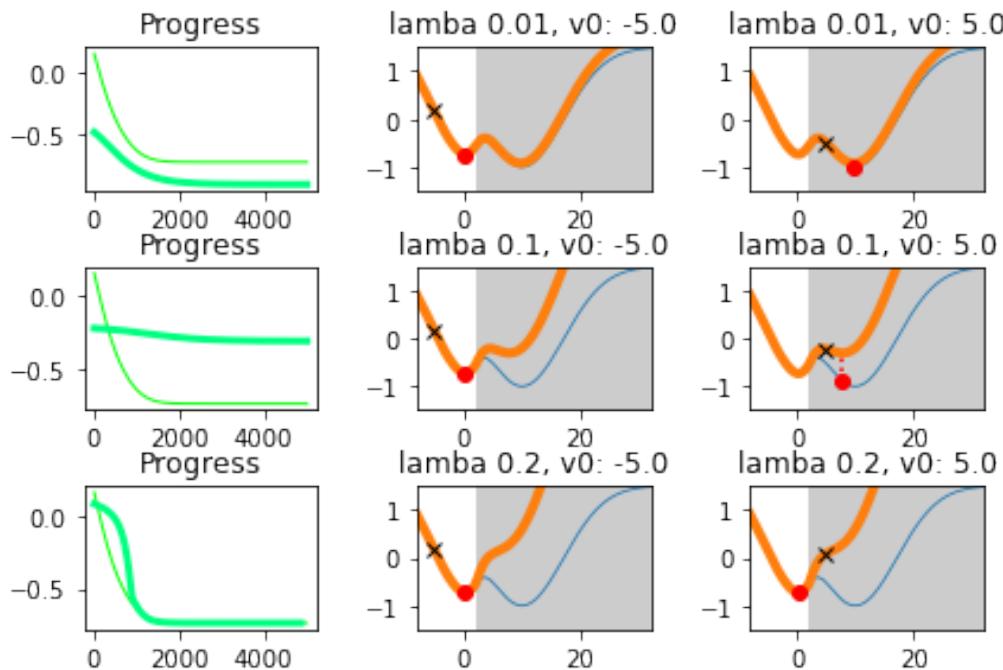
### แบบฝึกหัด 2.24

จงแก้ปัญหานี้ในแบบฝึกหัด 2.23 โดยใช้การกำหนดค่าเริ่มต้นด้วยวิธีการสุ่ม แล้วทดลอง เพิ่มจำนวนทำซ้ำให้มากขึ้นเท่าตัว. สังเกตผลลัพธ์ที่ได้ อภิปรายว่า การกำหนดค่าเริ่มต้นด้วยวิธีการสุ่ม จะช่วยบรรเทาปัญหาของการติดอยู่ในสถานการณ์ที่ดีที่สุดท้องถิ่นได้อย่างไร

คำใบ้ ลองคำสั่งการสุ่มค่า เช่น คำสั่ง `np.random.uniform`, คำสั่ง `np.random.normal`, หรือคำสั่ง `np.random.randn`.

### แบบฝึกหัด 2.25

จากแบบฝึกหัด 2.3 จะเขียนโปรแกรม เพื่อแก้ปัญหาแบบมีข้อจำกัด และเปรียบเทียบผลที่ได้ กับผลที่แสดงในรูป 2.29. ผลที่แสดงในรูป 2.29 ได้จากการคำนวณวิธีลงเกรเดียนต์ โดยใช้ รอบคำนวณสูงสุด เป็น 5000 และใช้ค่าข้าดก้าวเป็น 0.02.



รูปที่ 2.29: ตัวอย่างแสดงผลทำงานวิธีลงเกรเดียนต์ ของแบบฝึกหัด 2.25. แต่ละແກ່แสดงผล เมื่อเลือก  $\lambda$  เป็น 0.01, 0.1, 0.2 ตามลำดับ. ภาพในส่วนກraq แสดงความก้าวหน้า (ค่าฟังก์ชันจุดประสงค์ต่อรอบฝึก) โดยเส้นบางสีเขียว แสดงความก้าวหน้า เมื่อค่าเริ่มต้นเป็น  $-5$  และเส้นหนาสีน้ำเงินเขียว แสดงความก้าวหน้า เมื่อค่าเริ่มต้นเป็น  $5$ . กราฟทั้งหมดล้วนเข้าดี ยกเว้นที่  $\lambda = 0.1$  และค่าเริ่มต้นเป็น  $5$  (กราฟกลางเส้นหนาสีน้ำเงินเขียว) ที่ดูเหมือนกำลังลดลง แต่อาจต้องการรอบคำนวณเพิ่ม. ภาพในส่วนที่สอง และที่สาม แสดง ค่าฟังก์ชันจุดประสงค์ตั้งเดิม (เส้นบางสีฟ้า) ค่าฟังก์ชันจุดประสงค์ที่ถูกกลงโทษ (เส้นหนาสีส้ม) ค่าเริ่มต้นของตัวแปรตัดสินใจ (ากบาทสีดำ) และค่าสุดท้ายของการคำนวณ (จุดกลมสีแดง) พื้นที่สีเทาแสดงบริเวณที่ล่อมเมิดข้อจำกัด คำตอบได้ที่อยู่ในบริเวณนี้ถือว่าใช้ไม่ได้ โดยภาพในส่วนที่สอง แสดงผลเมื่อค่าเริ่มต้นเป็น  $-5$  และภาพในส่วนที่สาม แสดงผลเมื่อค่าเริ่มต้นเป็น  $5$ .

รูป 2.29 แสดงให้เห็นว่า เมื่อเลือกใช้ค่า  $\lambda$  ขนาดใหญ่มากพอก ไม่ว่าจะเลือกใช้ค่าเริ่มต้นที่ไหน คำตอบที่ได้จะเป็นค่าที่ใช้ได้ (feasible) ดังที่เห็นในแควรล่างสุด ภาพกลางและขวา.

### แบบฝึกหัด 2.26

ทฤษฎีบทكارุชคุนทักเกอร์ (Karush-Kuhn-Tucker theorem คำย่อ KKT)<sup>16</sup> กล่าวถึงเงื่อนไขสำหรับค่า

<sup>16</sup>เนื้อหาในส่วนนี้ เรียบเรียงจาก [39].

ทำให้น้อยที่สุดของปัญหา

$$\begin{aligned} & \text{minimize } f(\mathbf{x}) \\ & \text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & \quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

เมื่อตัวแปร  $\mathbf{x} \in \mathbb{R}^n$ . พิจารณาจุดประสงค์  $f : \mathbb{R}^n \mapsto \mathbb{R}$ . พิจารณาข้อจำกัด  $\mathbf{h} : \mathbb{R}^n \mapsto \mathbb{R}^m$ ,  $\mathbf{h} = [h_1, \dots, h_m]^T$ . และพิจารณาข้อจำกัด  $\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^p$ ,  $\mathbf{g} = [g_1, \dots, g_p]^T$ .

ทฤษฎีบทคารูชคุนท์กเกอร์ กล่าวว่า หากกำหนดให้พิจารณา  $f$ ,  $\mathbf{h}$ , และ  $\mathbf{g}$  เป็นพิจารณาที่สามารถหาอนุพันธ์ได้อย่างต่อเนื่อง (continuously differentiable) ซึ่งระบุด้วยสัญกรณ์  $f, \mathbf{h}, \mathbf{g} \in C^1$  และกำหนดให้  $\mathbf{x}^*$  เป็นจุดปกติ และเป็นค่าทำให้น้อยที่สุดท้องถิ่นของปัญหา  $\min f(\mathbf{x})$  s.t.  $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ ,  $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$  แล้วจะต้องมี  $\boldsymbol{\alpha} \in \mathbb{R}^m$  และ  $\boldsymbol{\beta} \in \mathbb{R}^p$  โดยที่

$$(\text{หนึ่ง}) \quad \boldsymbol{\beta} \geq \mathbf{0},$$

$$(\text{สอง}) \quad \nabla f(\mathbf{x}^*) + \boldsymbol{\alpha}^T \nabla \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\beta}^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0}^T, \text{ และ}$$

$$(\text{สาม}) \quad \boldsymbol{\beta}^T \mathbf{g}(\mathbf{x}^*) = 0.$$

การประยุกต์ใช้ทฤษฎีบทคารูชคุนท์กเกอร์ จะใช้เงื่อนไขทั้งสามนี้ ประกอบกับอีกสองเงื่อนไขข้อจำกัดเดิมได้แก่  $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$  และ  $\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}$  เพื่อค้นค่า  $\mathbf{x}^*$  ต่าง ๆ ที่มีโอกาสเป็นค่าทำให้น้อยที่สุด  $\mathbf{x}^*$ .

หมายเหตุ จุดปกติ (regular point) หมายถึง ค่า  $\mathbf{x}^*$  ที่สอดคล้องกับข้อจำกัดทั้งหมด และมีเกรเดียนต์ของข้อจำกัดที่ทำงานเป็นอิสระเชิงเส้นกัน. นั่นคือ ค่า  $\mathbf{x}^*$  จะเป็นจุดปกติ เมื่อ เงื่อนไขดังเดิม  $h_1(\mathbf{x}^*) = 0, \dots, h_m(\mathbf{x}^*) = 0$  และเงื่อนไขดังเดิม  $g_1(\mathbf{x}^*) \leq 0, \dots, g_p(\mathbf{x}^*) \leq 0$  และเกรเดียนต์เวกเตอร์  $\nabla h_i(\mathbf{x}^*)$ ,  $\nabla g_j(\mathbf{x}^*)$  สำหรับ  $i = 1, \dots, m$  และ  $j \in J(\mathbf{x}^*)$  เป็นอิสระเชิงเส้นต่อ กัน โดย เซตของดัชนีข้อจำกัดที่ทำงาน  $J(\mathbf{x}^*) \equiv \{j : g_j(\mathbf{x}^*) = 0\}$ .

ข้อจำกัดแบบภาวะไม่เท่ากัน  $g_j(\mathbf{x}) \leq 0$  จะเรียกว่า ทำงาน (active) ที่  $\mathbf{x}^*$  ถ้า  $g_j(\mathbf{x}^*) = 0$  และข้อจำกัด จะเรียกว่า ไม่ทำงาน (inactive) ที่  $\mathbf{x}^*$  ถ้า  $g_j(\mathbf{x}^*) < 0$ .

ความหมายของทฤษฎีบทคารูชคุนท์กเกอร์ คือ ด้วยเงื่อนไข  $\boldsymbol{\beta}_j \geq 0$  และข้อจำกัด  $g_j(\mathbf{x}^*) \leq 0$  ทำให้เงื่อนไข  $\boldsymbol{\beta}^T \mathbf{g}(\mathbf{x}^*) = \beta_1 g_1(\mathbf{x}^*) + \dots + \beta_p g_p(\mathbf{x}^*)$  สามารถนุமานได้ว่า ถ้า  $g_j(\mathbf{x}^*) < 0$  และ  $\beta_j = 0$  แต่ถ้า  $g_j(\mathbf{x}^*) = 0$  และ  $\beta_j$  อาจจะมีค่าเป็นบวกก็ได้ หรือเป็นศูนย์ก็ได้.

จวิเคราะห์ค่า  $\mathbf{x}^*$  ด้วยเงื่อนไขจากทฤษฎีบทคารูชคุนท์กเกอร์ สำหรับปัญหา  $\min f(\mathbf{x})$  s.t.  $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$  โดย  $g_1(\mathbf{x}) = (x_1 - 1.5)^2 + x_2 - 5.5$  และ  $g_2(\mathbf{x}) = 0.2x_1^2 - x_2 + 2.5 \leq 0$  และ  $f(\mathbf{x}) =$

$$1.5(x_1 + c_1)^2 + 1.5(x_2 + c_2)^2 \text{ เมื่อ}$$

- (สถานการณ์ ก)  $c_1 = 3.27$  และ  $c_2 = 4.8$ .
- (สถานการณ์ ข)  $c_1 = 3.27$  และ  $c_2 = 3.98950997289$ .
- (สถานการณ์ ค)  $c_1 = 2$  และ  $c_2 = 4$ .

พร้อมเขียนโปรแกรม เพื่อแสดงผลเช่นรูป 2.30.

ตัวอย่างการตรวจสอบเงื่อนไขかるชุดทั้กเกอร์พิจารณาสถานการณ์ ก เมื่อ  $f(\mathbf{x}) = 1.5(x_1 + 3.27)^2 + 1.5(x_2 + 4.8)^2$ . จากเงื่อนไขที่หนึ่ง  $\beta_1 \leq 0$  และ  $\beta_2 \leq 0$ . เงื่อนไขที่สอง  $\nabla f(\mathbf{x}) + \beta_1 \nabla g_1(\mathbf{x}) + \beta_2 \nabla g_2(\mathbf{x}) = 0$ . นั่นคือ

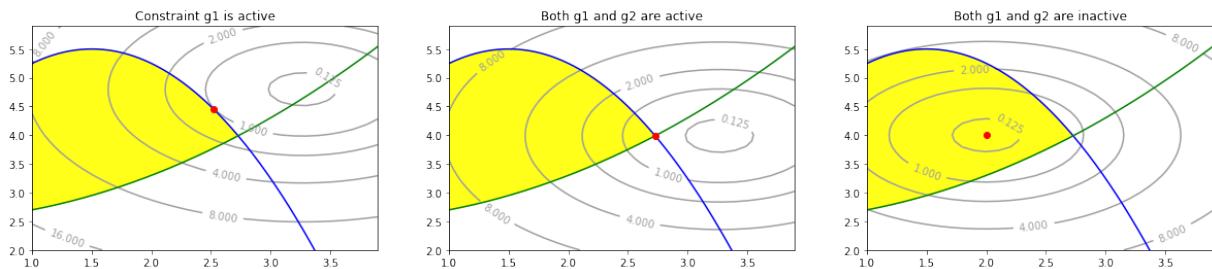
$$\begin{bmatrix} 3(x_1 - 3.27) + 2\beta_1(x_1 - 1.5) + 0.4\beta_2 x_1 \\ 3(x_2 - 4.8) + \beta_1 - \beta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

และเงื่อนไขที่สาม  $\beta_1 g_1(\mathbf{x}) + \beta_2 g_2(\mathbf{x}) = 0$ . นั่นคือ  $\beta_1(x_2 + (x_1 - 1.5)^2 - 5.5) + \beta_2(-x_2 + 0.2x_1^2 + 2.5) = 0$ .

กรณีแรก (a) ถ้า  $\beta_1 = \beta_2 = 0$ . เมื่อแทนค่า  $\beta_1$  และ  $\beta_2$  เข้าไปในเงื่อนไขที่สองแล้วจะได้ว่า  $\mathbf{x}_a = [3.27, 4.8]^T$  แต่เมื่อตรวจสอบเงื่อนข้อจำกัด  $g_1(\mathbf{x}_a) = 2.4329$  ซึ่งไม่เป็นข้อจำกัด  $g_1(\mathbf{x}) \leq 0$ . ดังนั้น  $\mathbf{x}_a$  ไม่ใช่คำตอบ.

กรณีที่สอง (b) ถ้า  $\beta_1 = 0$ . เมื่อแทนค่า  $\beta_1$  เข้าไปในเงื่อนไขที่สองและเงื่อนไขที่สาม จะได้สามสมการ ซึ่งสามารถแก้สมการเพื่อหาค่า  $x_1, x_2, \beta_2$  ออกมาได้ หลังจากแก้สมการแล้ว จะได้  $\mathbf{x}_b = [3.348, 4.742]^T$  และ  $\beta_2 = -0.175$  ซึ่งค่า  $\beta_2 < 0$  ลงทะเบียนได้ ดังนั้น  $\mathbf{x}_b$  ไม่ใช่คำตอบ.

กรณีที่สาม (c) ถ้า  $\beta_2 = 0$ . เมื่อแทนค่า  $\beta_2$  เข้าไปในเงื่อนไขที่สองและเงื่อนไขที่สาม จะได้สามสมการ ซึ่งสามารถแก้สมการเพื่อหาค่า  $x_1, x_2, \beta_1$  ออกมาได้ หลังจากแก้สมการแล้ว จะได้  $\mathbf{x}_c = [2.529, 4.440]^T$  และ  $\beta_1 = 1.079$  ซึ่งค่า  $\beta_1 > 0$  สอดคล้องกับเงื่อนไขแรก และเมื่อตรวจสอบ  $g_1(\mathbf{x}_c) = 0$  และ  $g_2(\mathbf{x}_c) = -0.661$  ซึ่งก็สอดคล้องกับข้อจำกัด  $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ . ดังนั้น  $\mathbf{x}_c$  สามารถเป็นคำตอบได้.



**รูปที่ 2.30:** ตัวอย่างปัญหาค่าน้อยที่สุดแบบมีข้อจำกัดกรณีต่าง ๆ. ภาพซ้าย แสดงสถานการณ์ ก ที่ค่าทำให้น้อยที่สุดอยู่ตำแหน่งที่ทำให้ข้อจำกัด  $g_1 = 0$  แต่  $g_2 < 0$ . ภาพกลาง แสดงสถานการณ์ ข ที่ทำให้ข้อจำกัดทั้ง  $g_1 = g_2 = 0$ . ภาพขวา แสดงสถานการณ์ ค ที่ทำให้ข้อจำกัดทั้ง  $g_1 < 0$  และ  $g_2 < 0$ . ตัวแปร  $\mathbf{x} \in \mathbb{R}^2$  และด้วยแกนนอนแทน  $x_1$  และแกนตั้งแทน  $x_2$ . สำหรับแต่ละภาพ ค่าฟังก์ชันจุดประสงค์  $f$  แสดงด้วยวดาดภาพคอนทัวร์ (เส้นระดับสีเทา). เส้นสีน้ำเงิน แสดงขอบเขตของข้อจำกัด  $g_1$  (เส้นแทน  $g_1(\mathbf{x}) = 0$ ). บริเวณค่าที่สอดคล้องกับข้อจำกัด  $g_1$  อยู่ด้านล่างของเส้นสีน้ำเงิน. เส้นสีเขียว แสดงขอบเขตของข้อจำกัด  $g_2$  (เส้นแทน  $g_2(\mathbf{x}) = 0$ ). บริเวณค่าที่สอดคล้องกับข้อจำกัด  $g_2$  อยู่ด้านบนของเส้นสีเขียว. พื้นที่แรเงาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดของ  $g_1$  และ  $g_2$ ). จุดสีแดงคือ  $\mathbf{x}^*$  ที่ถูกต้องสำหรับแต่ละกรณี.

หมายเหตุ การแก้สมการด้วยมือ เป็นการฝึกทักษะที่ดี. แต่อย่างไรก็ตาม โปรแกรมมีเครื่องมือที่สะดวกในการช่วยแก้สมการลักษณะแบบนี้. คำสั่งข้างล่างนี้ แสดงตัวอย่างการใช้ **sympy** เพื่อช่วยในการแก้สมการ (สถานการณ์ ก กรณี a)

```
from sympy.solvers import solve
from sympy import Symbol
x1, x2, beta2 = Symbol('x1'), Symbol('x2'), Symbol('beta2')
solve([3*(x1 - 3.27) + 0.4*beta2*x1,
       3*(x2-4.8) - beta2,
       beta2*(-x2 + 0.2*x1**2 + 2.5)])
```

ในทางปฏิบัติ การแก้ปัญหาแบบมีข้อจำกัด อาจจะสะดวกกว่าที่จะเลือกใช้วิธีการลงโทษ แต่ทฤษฎีบทคุณทักษะคุณทักษะ ช่วยให้ความเข้าใจเกี่ยวกับค่าตอบของปัญหา ซึ่งในหลาย ๆ กรณี ได้นำไปสู่วิธีการแก้ปัญหาที่มีประสิทธิภาพมาก. (หัวข้อ 4.2 อธิบายแบบจำลองจำแนกค่าทวิภาค ที่การพัฒนาใช้ประโยชน์จากทฤษฎีบทคุณทักษะคุณทักษะ)

## แบบฝึกหัด 2.27

จงเขียนโปรแกรม เพื่อแก้ปัญหาในแบบฝึกหัด 2.26 โดยใช้วิธีการลงโทษ.

คำใบ้ วิธีการลงโทษต้องการฟังก์ชันลงโทษ  $P_1(\mathbf{x})$  สำหรับข้อจำกัด  $g_1(\mathbf{x}) \leq 0$  อาจกำหนดเป็น  $P_1(\mathbf{x}) = \delta(g_1(\mathbf{x})) \cdot g_1(\mathbf{x})$  เมื่อ  $\delta$  เป็นฟังก์ชันขั้นบันไดหนึ่งหน่วย (unit step function).

นั่นคือ

$$\delta(a) = \begin{cases} 0 & \text{เมื่อ } a < 0, \\ 1 & \text{เมื่อ } a \geq 0. \end{cases}$$

คำสั่งข้างล่าง แสดงตัวอย่างโปรแกรมเกรเดียนต์  $\nabla P_1(\mathbf{x})$  ของฟังก์ชันลงโทษ  $P_1(\mathbf{x})$ .

```
def dPenalized_g1(x):
    g1 = x[1,0] + (x[0,0] - 1.5)**2 - 5.5
    dP = (g1 > 0) * np.array([[2*(x[0,0]-1.5)], [1]])
return dP
```

สังเกต การเขียนโปรแกรมข้างต้นใช้กลไก ( $g1 > 0$ ) ซึ่งเทียบเท่า ( $1 - \delta(-g_1)$ ). การใช้กลไกลักษณะนี้จะให้ค่าเป็นหนึ่ง (ลงโทษ) เมื่อ  $g1$  มากกว่าศูนย์ และให้ค่าเป็นศูนย์ (ไม่มีการลงโทษ) เมื่อ  $g1$  น้อยกว่าหรือเท่ากับศูนย์. เนื่องจากที่ขอบ (ที่  $g1$  เท่ากับศูนย์) จะไม่มีเกรเดียนต์. เมื่อเปรียบเทียบกับ ( $g1 \geq 0$ ) ซึ่งเทียบเท่า  $\delta(g_1)$  ผลลัพธ์อาจต่างกันเพียงเล็กน้อย แต่การรวมเงื่อนไขขอบที่ถูกต้องอาจช่วยให้การทำงานของวิธีลงโทษได้เสถียรภาพมากขึ้น.

หมายเหตุ ปัญหานี้แบบฝึกหัด 2.26 มีสองข้อจำกัด แต่ที่นี่แสดงตัวอย่างแค่สำหรับ  $g_1(\mathbf{x}) \leq 0$ .

## แบบฝึกหัด 2.28

หลาย ๆ สถานการณ์พบว่า ปัญหาการหาค่าดีที่สุดแบบมีเงื่อนไข จะมีคู่ปัญหาของมัน. และในกรณีนั้น ปัญหาการหาค่าดีที่สุดแบบมีเงื่อนไขดั้งเดิม จะเรียกว่า **ปัญหาปฐม** (primal problem) ส่วนปัญหาที่เป็นคู่ของปัญหาปฐม จะเรียกว่า **ปัญหาคู่** (dual problem).

สำหรับตัวอย่างของภาวะคู่กัน (duality) พิจารณาปัญหาเชิงเส้นที่เขียนในรูป

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

ฟังก์ชันจุดประสงค์  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$  เป็นฟังก์ชันเชิงเส้น และข้อจำกัดต่าง ๆ ก็เป็นฟังก์ชันเชิงเส้น. การหาค่าดีที่สุดแบบมีเงื่อนไข สำหรับปัญหาเชิงเส้น มักถูกอ้างถึงด้วยชื่อ **การโปรแกรมเชิงเส้น** (linear programming). การโปรแกรมเชิงเส้น เป็นการศึกษาถึงขั้นตอนวิธีต่าง ๆ ที่มีประสิทธิภาพ สำหรับการหาค่าดีที่สุดแบบมีเงื่อนไข เพื่อใช้กับปัญหาเชิงเส้น. รายละเอียดของการโปรแกรมเชิงเส้น สามารถศึกษาเพิ่มเติมได้จาก [39] หรือ [137].

จากวิธีลากرانจ์ (Lagrange method ศึกษาได้จาก [39]) ปัญหาเชิงเส้นแบบมีข้อจำกัดข้างต้น สามารถเขียนในรูปปัญหาที่ไม่มีข้อจำกัดได้เป็น

$$\min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{x} - \boldsymbol{\lambda}_1^T (\mathbf{A}\mathbf{x} - \mathbf{b}) - \boldsymbol{\lambda}_2^T \mathbf{x}$$

เมื่อ  $\boldsymbol{\lambda}_1 \geq \mathbf{0}$ ,  $\boldsymbol{\lambda}_2 \geq \mathbf{0}$ , และทั้ง  $\boldsymbol{\lambda}_1$  กับ  $\boldsymbol{\lambda}_2$  มีค่าใหญ่มากพอ. สังเกตว่า หากมีการลดเมิดข้อจำกัด เช่น  $\mathbf{A}\mathbf{x} < \mathbf{b}$  จะทำให้พจน์  $-\boldsymbol{\lambda}_1^T (\mathbf{A}\mathbf{x} - \mathbf{b})$  มีค่าเป็นบวก และเมื่อประกอบกับกลไกของลากرانจ์พารามิเตอร์  $\boldsymbol{\lambda}_1 \geq \mathbf{0}$  ที่หาก  $\boldsymbol{\lambda}_1$  มีขนาดใหญ่มากพอ จะทำให้ค่าจุดประสงค์รวมมากขึ้น และส่งผลต่อเนื่องทำให้การค้นหาค่า  $\mathbf{x}$  จะต้องปรับค่า  $\mathbf{x}$  และส่งผลเป็นการแก้ไขการลดเมิดดังกล่าว.

วิธีของวิธีลากرانจ์ จะต้องเลือกลากرانจ์พารามิเตอร์ให้เหมาะสม นั่นคือมีค่าใหญ่มากพอที่ข้อจำกัดจะไม่ถูกละเมิด. แต่การเลือกลากرانจ์พารามิเตอร์ที่มีค่าใหญ่มากเกินไป จะไปขัดขวางการค้นหาค่าที่ดีที่สุด (ผลลัพธ์ที่ได้จะไม่ละเมิดข้อจำกัด แต่จะไม่ใช่ค่าที่ดีที่สุดที่เป็นไปได้<sup>17</sup>). ดังนั้น การเลือกขนาดของลากرانจ์พารามิเตอร์เอง ก็สามารถถูกมองเป็นปัญหาการหาค่าดีที่สุดได้. นั่นคือ เลือกขนาดของลากرانจ์พารามิเตอร์ที่ใหญ่ที่สุด ที่จะไม่ทำร้ายจุดประสงค์เดิมของปัญหาปัญหา.

เพื่อความสะดวก กำหนดให้ฟังก์ชันจุดประสงค์รวม

$$L \equiv \mathbf{c}^T \mathbf{x} - \boldsymbol{\lambda}_1^T (\mathbf{A}\mathbf{x} - \mathbf{b}) - \boldsymbol{\lambda}_2^T \mathbf{x}$$

โดย  $\boldsymbol{\lambda}_1 \geq \mathbf{0}$  และ  $\boldsymbol{\lambda}_2 \geq \mathbf{0}$ .

ดังนั้นกราฟเดียนต์

$$\nabla_{\mathbf{x}} L = \mathbf{c}^T - \boldsymbol{\lambda}_1^T \mathbf{A} - \boldsymbol{\lambda}_2^T$$

และเมื่อพิจารณา ณ จุดดีที่สุด<sup>18</sup> นั่นคือ ที่  $\nabla_{\mathbf{x}} L = \mathbf{0}$  และแก้สมการจะได้  $\boldsymbol{\lambda}_2^T = \mathbf{c}^T - \boldsymbol{\lambda}_1^T \mathbf{A}$ . ดังนั้น ค่าฟังก์ชันจุดประสงค์รวม ณ จุดดีที่สุด (เมื่อแทนค่า  $\boldsymbol{\lambda}_2$  เข้าไป) จะเป็น

$$L' = \boldsymbol{\lambda}_1^T \mathbf{b}$$

โดย  $\boldsymbol{\lambda}_1 \geq \mathbf{0}$  และ  $\boldsymbol{\lambda}_2 \geq \mathbf{0}$ . และ ณ จุดดีที่สุด เงื่อนไข  $\boldsymbol{\lambda}_2 \geq \mathbf{0} \equiv \boldsymbol{\lambda}_1^T \mathbf{A} \leq \mathbf{c}^T$ . สังเกตว่า (1)  $L$  เป็นค่าฟังก์ชันจุดประสงค์รวม ที่ค่าขึ้นกับ  $\mathbf{x}$  แต่  $L'$  เป็นค่าฟังก์ชันจุดประสงค์รวม ที่ได้เลือก  $\mathbf{x}$  ให้ดีที่สุดแล้ว และ

<sup>17</sup>วิธีลากرانจ์ จะต่างจากวิธีการลงโทษ โดยวิธีลากرانจ์ ใช้ลากرانจ์พารามิเตอร์และต้องเลือกค่าให้เหมาะสม. แต่วิธีการลงโทษ ใช้ฟังก์ชันการลงโทษ ซึ่งจะลงโทษเฉพาะตอนที่ละเมิดข้อจำกัด ดังนั้นการเลือกค่าน้ำหนักในการลงโทษจะผ่อนคลายกว่า. นั่นคือ สำหรับวิธีการลงโทษ เพียงเลือกค่าน้ำหนักให้มีค่าใหญ่มากพอเท่านั้น ไม่ต้องห่วงว่ามากเกินไปจะไปรบกวนฟังก์ชันจุดประสงค์หลัก. แต่ความสะดวกนี้ ก็จะแกลกมาด้วยการเลือกใช้ฟังก์ชันลงโทษให้เหมาะสม และประสิทธิภาพการทำงาน.

<sup>18</sup>การวิเคราะห์นี้เทียบเท่าทฤษฎีบทค่ารูรุคุนทั่วโลก โดยเฉพาะเงื่อนไขที่สอง.

(2)  $L'$  ไม่ใช่ฟังก์ชันของ  $\mathbf{x}$  แต่เป็นฟังก์ชันของ  $\boldsymbol{\lambda}_1$ . การมองจากปัญหาจากมุ่งมองของ  $\boldsymbol{\lambda}_1$  จะทำให้ได้ปัญหาคู่ ซึ่งเขียนได้เป็น

$$\begin{aligned} & \underset{\boldsymbol{\lambda}_1}{\text{maximize}} \quad \boldsymbol{\lambda}_1^T \mathbf{b} \\ & \text{subject to} \quad \boldsymbol{\lambda}_1 \geq \mathbf{0}, \\ & \quad \boldsymbol{\lambda}_1^T \mathbf{A} \leq \mathbf{c}^T. \end{aligned}$$

สังเกต ข้อจำกัด  $\boldsymbol{\lambda}_1^T \mathbf{A} \leq \mathbf{c}^T$  เปรียบเสมือน เงื่อนไขที่ควบคุมไม่ให้  $\boldsymbol{\lambda}_1$  มีค่าใหญ่เกินไปจนไปรบกวนจุดประสงค์ดั้งเดิมในปัญหาปฐม.

หากเปรียบเทียบ ปัญหาปฐมเป็นเสมือนการหาค่า  $\mathbf{x}$  ที่ทำให้จุดประสงค์เดิมเล็กที่สุด แต่การดำเนินการให้จุดประสงค์เดิม  $f$  มีขนาดเล็ก ถูกควบคุมด้วยข้อจำกัดดังเดิมต่าง ๆ. ดังนั้น จุดประสงค์เดิมจะเล็กได้เท่าที่ข้อจำกัดเดิมอนุญาต. ในขณะที่ปัญหาคู่ มองจากอีกด้านของมุ่งมองจากจุดที่ปรับ  $\mathbf{x}$  ได้สูงสุดแล้ว แต่ต้องการคุณไม่ให้ลดเมิดข้อจำกัด. ปัญหาคู่ จึงเสมือนการหาค่า  $\boldsymbol{\lambda}_1$  ที่ทำให้จุดประสงค์รวม  $L'$  (ซึ่งรวมข้อจำกัดเดิมและปรับ  $\mathbf{x}$  ดีที่สุดแล้ว) มีค่ามากที่สุด เพื่อรักษาข้อจำกัดเดิมต่าง ๆ ไว้ แต่การดำเนินการให้  $L'$  ใหญ่ ถูกควบคุมไม่ให้มากเกินไปจนรบกวนจุดประสงค์ดั้งเดิม.

เมื่อแก้ปัญหาคู่เสร็จ คำตอบจะได้  $\boldsymbol{\lambda}_1^*$  และทำให้สามารถคำนวณ  $\boldsymbol{\lambda}_2^* = \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda}_1^*$ . ด้วยความเชื่อมโยงและทฤษฎีบทكار์เรอร์ คำตอบของปัญหาปฐม สามารถพิจารณาได้ดังนี้. ตรวจสอบส่วนประกอบต่าง ๆ นั่นคือ  $\boldsymbol{\lambda}_1^* = [\lambda_{11}, \dots, \lambda_{1m}]^T$  และ  $\boldsymbol{\lambda}_2^* = [\lambda_{21}, \dots, \lambda_{2n}]^T$ . ถ้า  $\lambda_{1i} > 0$  แปลว่า เงื่อนไขที่ตำแหน่ง  $i^{th}$  ทำงาน นั่นคือ  $\mathbf{A}_{i,:} \cdot \mathbf{x} = b_i$ . ถ้า  $\lambda_{2i} > 0$  แปลว่า  $x_i = 0$ . ค่าของ  $\mathbf{x}^*$  สามารถวิเคราะห์ได้จากการ  $\lambda_1^*$ .

ตัวอย่างปัญหาเชิงเส้นข้างล่าง

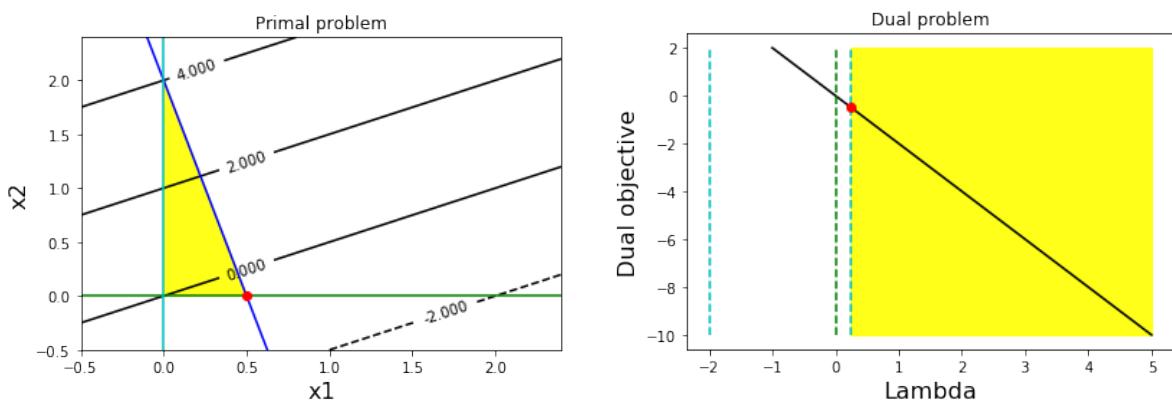
$$\begin{aligned} & \underset{x_1, x_2}{\text{minimize}} \quad 2x_2 - x_1 \\ & \text{subject to} \quad -4x_1 - x_2 \geq -2, \\ & \quad x_1 \geq 0, \\ & \quad x_2 \geq 0. \end{aligned}$$

ซึ่งอยู่ในปฐมรูป (primal form). เปรียบเทียบกับแก้ปัญหา

$$\begin{aligned}
 & \underset{\lambda_1}{\text{maximize}} && -2\lambda_1 \\
 & \text{subject to} && \lambda_1 \geq 0, \\
 & && -4\lambda_1 \leq -1, \\
 & && -\lambda_1 \leq 2
 \end{aligned}$$

ซึ่งเป็นรูปคู่' (dual form) ของปัญหาข้างต้น.

เมื่อแก้ปัญหาคู่เสริม ผลลัพธ์คือ  $\lambda_1^* = 0.25$ . ดังนั้น  $\lambda_2^* = [-1, 2]^T - [-4, -1]^T \cdot 0.25 = [0, 2.25]^T$ . เนื่องจาก  $\lambda_{22} > 0$  ดังนั้น  $x_2 = 0$ . และเนื่องจาก  $\lambda_1 > 0$  ดังนั้น  $-4x_1 - x_2 = -2$ . เมื่อวิเคราะห์ผลทั้งหมดรวมกันจะได้ว่า  $\mathbf{x}^* = [0.5, 0]^T$ . รูป 2.31 แสดงภาพของภาวะคู่กันในตัวอย่างนี้.



**รูปที่ 2.31:** ตัวอย่างภาวะคู่กัน. ภาพข่าย แสดงปัญหาปฐม (ปัญหาค่าน้อยที่สุด) ด้วยค่าฟังก์ชันจุดประสงค์ในปริภูมิของตัวแปร  $\mathbf{x}$ . ค่าฟังก์ชันจุดประสงค์  $f(\mathbf{x}) = 2x_2 - x_1$  แสดงด้วยหาดภาพคอนทัวร์. เส้นสีน้ำเงิน แสดงขอบเขตของข้อจำกัด  $-4x_1 - x_2 \geq -2$  (เส้นแทน  $-4x_1 - x_2 = -2$ ). เส้นสีเขียว แสดงขอบเขตของข้อจำกัด  $x_2 \geq 0$  (เส้นแทน  $x_2 = 0$ ). เส้นสีฟ้าเขียว แสดงขอบเขตของข้อจำกัด  $x_1 \geq 0$  (เส้นแทน  $x_1 = 0$ ). พื้นที่แรเงาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดทั้งสาม). จุดสีแดงคือ  $\mathbf{x}^*$ . ภาพขวา แสดงปัญหาคู่ (ปัญหาค่ามากที่สุด) ด้วยแกนตั้งเป็นค่าฟังก์ชันจุดประสงค์ของปัญหาคู่ และแกนนอนแสดงค่า  $\lambda$ . เส้นสีดำทิบ คือ ค่าฟังก์ชันจุดประสงค์ของปัญหาคู่  $L'(\lambda_1) = -2\lambda_1$ . เส้นประ แสดงขอบเขตของข้อจำกัด  $\lambda_1 \geq 0$  (เส้นสีเขียว) และข้อจำกัด  $-4\lambda_1 \leq -1 \equiv \lambda_1 \geq 0.25$  และ  $-\lambda_1 \leq 2 \equiv \lambda_1 \geq -2$  (ทั้งคู่แสดงด้วยเส้นสีฟ้าเขียว). พื้นที่แรเงาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดทั้งสาม). จุดสีแดง (ในทั้งสองภาพ) แทนค่าตอบที่ถูกต้อง. นั่นคือ ปัญหาปฐม  $x_1^* = 0.5$ ,  $x_2^* = 0$  และปัญหาคู่  $\lambda_1^* = 0.25$ .

จากปัญหาเชิงเส้นข้างล่าง จงแปลงเป็นรูปคู่ แก้ปัญหาทั้งในรูปปฐม และรูปคู่ และตรวจสอบคำตอบ.

$$\begin{aligned}
 & \underset{x_1, x_2}{\text{minimize}} && 2x_2 + 2x_1 \\
 & \text{subject to} && -4x_1 - x_2 \geq -2, \\
 & && x_1 \geq 0, \\
 & && x_2 \geq 0.
 \end{aligned}$$

หมายเหตุ วิธีลงเกรเดียนต์ และวิธีลงโທะสามารถใช้ช่วงหาค่าตอบได้ แต่ปัญหาเชิงเส้น เป็นกลุ่มปัญหาที่ได้รับการศึกษาอย่างกว้างขวาง และมีขั้นตอนวิธีต่าง ๆ ที่ได้พัฒนาขึ้นเฉพาะ ซึ่งมีประสิทธิภาพมากกว่าวิธีลงเกรเดียนต์มาก เช่น วิธีซิมเพล็กซ์ (simplex method) และวิธีจุดภายใน (interior-point method). เนื้อหาของปัญหาการหาค่าดีที่สุดสำหรับปัญหาเชิงเส้น เกินขอบเขตของหนังสือเล่มนี้ ผู้อ่านที่สนใจสามารถศึกษาเพิ่มเติมได้จาก [137].

## การคำนวณเชิงเลข

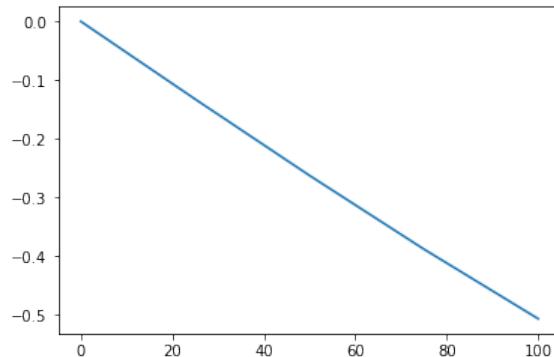
การเขียนโปรแกรมคำนวณเชิงเลข มีปัจจัยด้านข้อจำกัดที่ต้องคำนึงถึง. แบบฝึกหัดต่อไปนี้ แนะนำบางประเด็นที่ควรคำนึงถึง เวลานำทฤษฎีทางคณิตศาสตร์มาเขียนโปรแกรม.

### แบบฝึกหัด 2.29

โปรแกรมข้างล่างนี้ใช้วาดรูป 2.32.

```
x = np.linspace(0, 100, 5)
plt.plot(x, np.sin(x))
```

จงวิเคราะห์และอธิบายว่า ทำไม่รูปที่ได้ไม่เห็นเป็นรูปโค้งขึ้นลง เช่น รูปของค่าพังชันไซน์ที่คุณเคย



รูปที่ 2.32: ผลจากการคำสั่ง `plt.plot(x, np.sin(x))` จากแบบฝึกหัด 2.29.

### แบบฝึกหัด 2.30

จากโปรแกรมและผลการรันดังแสดงข้างล่างนี้ จงอภิราย่าว่าทำไม่มี  $x$  บางตัวไม่เท่ากับ  $7 \cdot y$  ทั้ง ๆ ที่  $y = x/7$ . โปรแกรมคำนวณ

```
x = np.linspace(1,10, 20)
y = x/7
print(x == 7*y)
```

และผลลัพธ์ที่ได้คือ

```
[ True  True
True False False True  True  True  True  True ]
```

ทำไม จึงมีผลบางค่าที่เป็น **False** ทั้ง ๆ ที่  $\frac{x}{7}$  มีค่าเท่ากับ  $x$ ? จงวิเคราะห์และอธิบายผลของ  $x == 7*(x/7)$  กับ  $x == 7*x/7$  ประกอบ พิรุณภาพประเด็นที่ได้เรียนรู้นี้ กับสถานการณ์ที่อาจจะเกิดขึ้น.

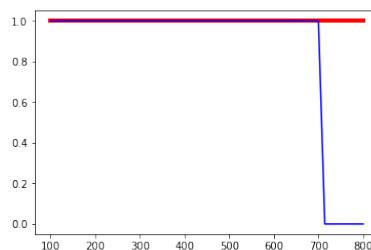
### แบบฝึกหัด 2.31

จากคณิตศาสตร์  $\log(\exp(x)) = x$  โปรแกรมข้างล่างนี้

```
xs = np.linspace(100, 800)
plt.plot(xs, xs/xs, 'r', linewidth=4)
plt.plot(xs, xs/np.log(np.exp(xs)), 'b', linewidth=1.5)
```

วาดกราฟของ  $x/x$  เปรียบเทียบกับ  $x/\log(\exp(x))$  โดย  $x$  มีค่าตั้งแต่ 100 ถึง 799 ซึ่ง ทั้ง  $x/x$  และ  $x/\log(\exp(x))$  ก็มีค่าเท่ากับ 1 เมื่อ  $x > 0$ . ดังนั้น ผลลัพธ์น่าจะเห็นเส้นตรงแนวโนนที่ค่าหนึ่ง เท่าเดิม คงที่ตลอดช่วง. แต่ผลที่ได้เป็นดังแสดงในรูป 2.33. จงสืบกรณีนี้ อธิบายสิ่งที่เกิดขึ้น และอภิปรายผลที่อาจเกิดขึ้นในทางปฏิบัติ จากประเด็นที่ได้เรียนรู้.

คำใบ้ ตรวจสอบค่า **np.exp(x)** ที่ค่า **x** ต่าง ๆ และลองสืบค้นข้อมูลเรื่อง IEEE754 จากอินเตอร์เน็ต.



รูปที่ 2.33: ผลจากการวาดกราฟ  $x/x$  (เส้นสีแดงหนา) และกราฟ  $x/\log(\exp(x))$  (เส้นสีน้ำเงินบาง) โดย  $x$  มีค่าตั้งแต่ 100 ถึง 799. เส้นกราฟ อาจดูต่างจากที่คาด. แบบฝึกหัด 2.31.

### แบบฝึกหัด 2.32

ฟังก์ชันซอฟต์แมกซ์ (softmax function) ซึ่งมักใช้สัญลักษณ์ softmax :  $\mathbb{R}^n \mapsto \mathbb{R}^n$  นิยามว่า เมื่อ อินพุตของซอฟต์แมกซ์  $\mathbf{v} = [v_1, \dots, v_n]^T$  และผลลัพธ์  $\mathbf{u} = \text{softmax}(\mathbf{v})$  โดย

$$u_i = \frac{\exp(v_i)}{\sum_{j=1}^n \exp(v_j)}$$

สำหรับ  $i = 1, \dots, n$  เมื่อ  $u_i$  เป็นส่วนประกอบของ  $\mathbf{u}$ . ฟังก์ชันซอฟต์แมกซ์นิยมใช้อย่างมาก ในงานรู้จำรูปแบบ. โปรแกรมข้างล่างนี้ เขียนการคำนวณฟังก์ชันซอฟต์แมกซ์แบบง่าย ๆ

```
def softmax(v):
    ev = np.exp(v)
    return ev/np.sum(ev)
```

จะทดสอบฟังก์ชันนี้ ด้วยค่า  $v$  ต่าง ๆ เช่น `softmax(np.array([1, 2, 5]))` (ลองผสมค่าหลาย ๆ แบบ ทั้งค่าบวก ค่าลบ และศูนย์) อภิปรายการพฤติกรรมของฟังก์ชันซอฟต์แมกซ์. แล้วลองทดสอบอีกรอบด้วยค่าขนาดใหญ่ เช่น `softmax(np.array([1000, 2000, 5000]))` สังเกตผลลัพธ์ที่ได้ อภิปรายถึงปัญหาและสาเหตุ พร้อมเสนอวิธีแก้ปัญหา.

คำใบ้ ปัญหาอยู่ที่ไหน วิธีแก้อาจใช้คณิตศาสตร์ไปช่วยบรรเทาสาเหตุ. (หัวข้อ 3.7 อภิปรายวิธีเขียนโปรแกรม ฟังก์ชันซอฟต์แมกซ์ที่ทนทาน (robust) สำหรับใช้งานในทางปฏิบัติ.)

## บทที่ 3

### การเรียนรู้ของเครื่องและโครงข่ายประสาทเทียม

“If I have seen further, it is by standing upon the shoulders of giants.”

---Isaac Newton

“ถ้าผมมองเห็นได้ไกลกว่า มันก็มาจากการยืนอยู่บนไหล่ของเหล่ายกษัตริย์”

—ไอแซค นิวตัน

วิธีการเรียนรู้ของเครื่อง มีมากมาย หลากหลายแบบ แตกต่างกันไปตามลักษณะงานที่ต้องการ. ตัวอย่าง การปรับเลี้นโครงด้วยฟังก์ชันพหุนาม ในหัวข้อ 3.1 อภิรายตัวอย่างง่าย ๆ ที่เป็นแนวทางหลัก และสะท้อนหลักการที่สำคัญของการเรียนรู้ของเครื่อง. หัวข้อ 3.2 อภิรายพื้นฐาน หลักการ และประเด็นสำคัญของศาสตร์การเรียนรู้ของเครื่อง. หัวข้อ 3.3 อภิรายโครงข่ายประสาทเทียม ซึ่งเป็นแบบจำลองที่สำคัญ ใช้งานได้กว้างขวาง และเป็นหนึ่งในศาสตร์และศิลป์ของการเรียนรู้ของเครื่อง. หัวข้อ 3.4 อภิรายการประยุกต์ใช้งานของโครงข่ายประสาทเทียม. หัวข้อ 3.5 อภิรายคำแนะนำทั้งสำหรับการใช้งานโครงข่ายประสาทเทียม และการใช้งานการเรียนรู้ของเครื่องโดยทั่วไป.

#### 3.1 การปรับเลี้นโครงด้วยฟังก์ชันพหุนาม

การทำแบบจำลอง คือการสร้างสมการคณิตศาสตร์ เพื่อคำนวณค่าคำตอบ  $y$  จากค่าคำถาม  $x$ . และหลังจากทำแบบจำลองเสร็จเรียบร้อยแล้ว แบบจำลองที่ได้ (สมการคณิตศาสตร์ที่นิยามการคำนวณครบถ้วนอย่างทุกขั้นตอน) จะสามารถนำไปใช้บนมานาหรือท่านายค่าคำตอบ สำหรับคำถามที่สงสัยได้.

พิจารณากรณีที่ทั้งอินพุตและเอาต์พุตเป็นมิติเดียว นั่นคือ คำถาม  $x \in \mathbb{R}$  และ  $y \in \mathbb{R}$ . หากต้องการจะท่านายค่า  $y$  ที่สัมพันธ์กับค่า  $x$  โดยที่มีตัวอย่างข้อมูลเป็นคู่ ๆ ของ  $(x, y)$  ได้แก่  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  ทั้งหมดจำนวน  $n$  คู่. ตัวแปรต้น  $x$  เป็นค่าที่ถามมา เพื่อหา  $y$  ที่เป็นตัวแปรตาม หรือค่าที่อยากได้ค่า

ตอบไป. แต่ละคู่  $(x_i, y_i)$  อาจเรียกว่า **จุดข้อมูล** (datapoint).

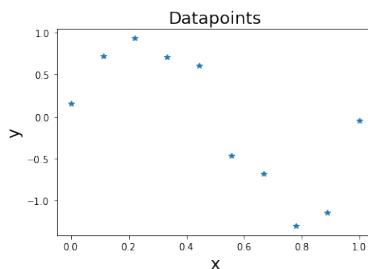
รูป 3.1 แสดงตัวอย่างจุดข้อมูล 10 จุด. ตำแหน่งของแต่ละจุดข้อมูลในภาพ ระบุจากค่า  $x$  ตามแกนนอน และค่า  $y$  ตามแกนตั้ง. ตัวแปรตน  $x$  อาจเรียก อินพุต หรือข้อมูลนำเข้า (input) และตัวแปรตาม  $y$  อาจเรียก เอ้าท์พุต หรือข้อมูลนำออก (output). จากตัวอย่างในภาพ จุดข้อมูลแรกสุด มีค่า  $x = 0$  ค่า  $y = 0.16$ .

เป้าหมายของตัวอย่างนี้คือ การคำนวณค่า  $y$  ของค่าอินพุต  $x$  ที่ส่งสัญ โดยอินพุต  $x$  อาจ จะเป็นค่าเดิม หรืออาจจะเป็นค่าใหม่ที่ไม่เคยเห็นมาก่อน. แนวทางคือ การใช้แบบจำลอง ซึ่งเป็นการคำนวณ ทางคณิตศาสตร์ ที่เป็นฟังก์ชันของตัวแปร  $x$  และใช้ค่าที่ฟังก์ชันคำนวณได้ ทายเป็นค่า  $y$ . แบบจำลองที่จะ เลือกใช้สำหรับตัวอย่างนี้ คือ **ฟังก์ชันพหุนาม** (polynomial function). ฟังก์ชันพหุนาม  $f$  คำนวณค่า  $y$  จาก  $x$  โดย

$$y = f(x, \mathbf{w}) = w_0 + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3 + \dots + w_m \cdot x^m \quad (3.1)$$

เมื่อ  $\mathbf{w} = [w_0, w_1, w_2, \dots, w_m]^T$  เป็นค่าพารามิเตอร์ของฟังก์ชันพหุนาม และ  $m$  เป็นระดับขั้น (degree) ของฟังก์ชันพหุนาม. ก่อนที่จะสามารถนำฟังก์ชันพหุนาม ไปใช้คำนวณค่า  $y$  จากค่า  $x$  ที่สามได้ ต้องคำนวณระดับขั้น  $m$  และค่าของพารามิเตอร์  $\mathbf{w}$  ให้เรียบร้อยก่อน.

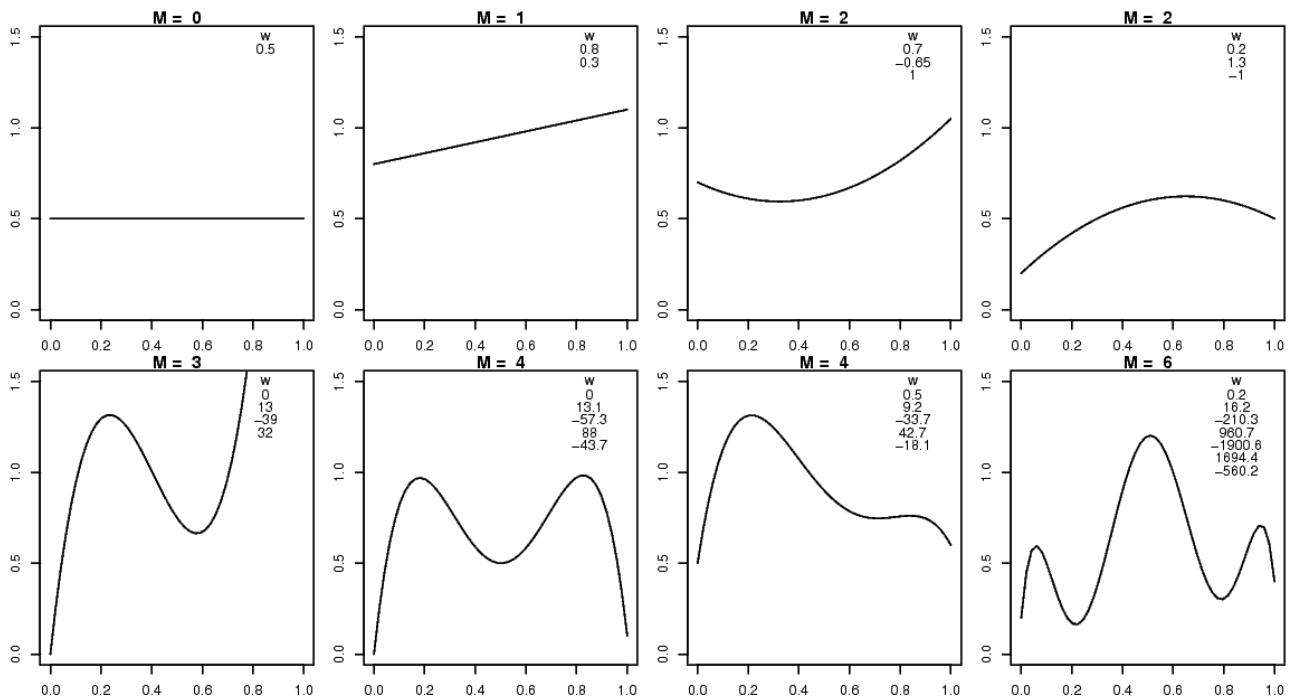
ตัวอย่างเช่น หากเลือก ระดับขั้น  $m = 2$  และค่าของพารามิเตอร์  $\mathbf{w} = [0.7, -0.65, 1]^T$  สำหรับ แบบจำลอง  $f_1$  นั้นคือ ฟังก์ชันพหุนาม  $y = f_1(x) = 0.7 - 0.65x + x^2$  แล้วที่  $x = 0.5$  จะคำนวณค่า  $y$  เป็น 0.625. ระดับขั้น และค่าของพารามิเตอร์ที่เลือกใช้ ส่งผลโดยตรงกับค่าที่คำนวณ เช่น หากเลือก ระดับขั้น  $m = 3$  และค่าของพารามิเตอร์  $\mathbf{w} = [0, 13, -39, 32]^T$  สำหรับแบบจำลอง  $f_2$  นั้นคือ ฟังก์ชันพหุนาม  $y = f_2(x) = 13x - 39x^2 + 32x^3$  แล้วที่  $x = 0.5$  จะคำนวณค่า  $y$  เป็น 0.75 ซึ่งแตกต่างจากผล คำนวณจาก  $f_1$ .



รูปที่ 3.1: ตัวอย่าง จุดข้อมูล  $(0, 0.160)$ ,  $(0.111, 0.724)$ ,  $(0.222, 0.931)$ ,  $(0.333, 0.712)$ ,  $(0.444, 0.610)$ ,  $(0.556, -0.460)$ ,  $(0.667, -0.684)$ ,  $(0.778, -1.299)$ ,  $(0.889, -1.147)$ , และ  $(1, -0.045)$  รวม 10 จุด.

ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ จะให้ผลการทำนายต่างกัน. รูป 3.2 แสดงพฤติกรรมการทำนาย ของฟังก์ชันพหุนาม เมื่อเลือกใช้ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ. พฤติกรรมการทำนาย หมายถึง ความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต. การปรับเลี้นโค้ง ใช้ประโยชน์จากการที่ พฤติกรรมการทำนายเปลี่ยนตามระดับขั้นและค่าพารามิเตอร์. ดังนั้น การปรับเลี้นโค้ง ทำได้โดยปรับระดับขั้น และค่าพารามิเตอร์ของสมการ เพื่อปรับเส้นโค้งให้ได้ผลการทำนายที่ดีขึ้น.

ระดับขั้นของสมการพหุนาม จะกำหนดจำนวนพารามิเตอร์ของฟังก์ชันพหุนาม. การเลือกระดับขั้น เป็นการเลือกความสามารถของแบบจำลองโดยรวม. ระดับขั้นสูง แบบจำลองจะมีความสามารถมาก แต่ก็เพิ่มจำนวนพารามิเตอร์ที่ต้องหาค่า เท่ากับ เพิ่มความยากในการปรับแบบจำลองขึ้น. หัวข้อ 3.2 อภิปรายวิธีการเลือกระดับขั้น. ตอนนี้ สมมติว่าระดับขั้น  $m$  ถูกเลือกมาแล้ว หากเลือกระดับขั้นเป็น  $m$  ฟังก์ชันพหุนามจะ มีจำนวนพารามิเตอร์ เป็น  $m + 1$  ตัว. การหาค่าของพารามิเตอร์เหล่านี้ จะเรียกว่า **การฝึก (training)** หรือ **การเรียนรู้ (learning)**.



รูปที่ 3.2: ฟังก์ชันพหุนาม เมื่อเลือกใช้ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ. ระดับขั้น  $M$  แสดงเหนือแต่ละภาพ และค่าพารามิเตอร์แสดงภายในแต่ละภาพ.

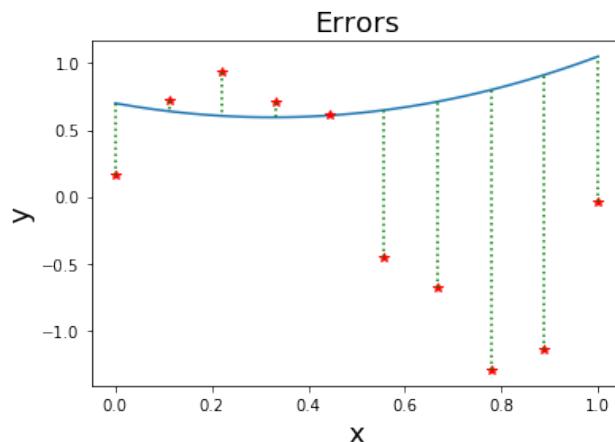
**การฝึกแบบจำลอง.** การฝึกแบบจำลอง คือการหาค่าพารามิเตอร์ของแบบจำลอง เพื่อให้แบบจำลองทำนายได้ถูกต้องมากที่สุด หรือกล่าวอีกอย่างคือ เพื่อให้แบบจำลองทำนายผิดน้อยที่สุด. การวัดว่าแบบจำลองทำนายได้ผิดมากน้อยเท่าใด สามารถใช้แนวทางของวิธีกำลังสองน้อยที่สุด (Least Square method) ได้. วิธีกำลัง

สองน้อยที่สุด วัดว่าแบบจำลองที่ทำนายได้ผิดมากน้อยเท่าใด จากผลต่างกำลังสอง ระหว่างค่าเอาต์พุตที่ทำนาย กับค่าเอาต์พุตจริง. นั่นคือ  $E_n = (\hat{y}_n - y_n)^2$  เมื่อ  $E_n$  คือความผิดพลาด (error) ของการทำนายจุดข้อมูลที่  $n^{th}$  ซึ่งวัดจากผลต่างกำลังสองระหว่าง ค่าเอาต์พุตที่ทำนาย  $\hat{y}_n$  สำหรับจุดข้อมูลที่  $n^{th}$  และค่าเอาต์พุตจริง  $y_n$  ของจุดข้อมูลที่  $n^{th}$ . ค่า  $y_n$  ที่ได้จากข้อมูล อาจเรียกว่า เอาต์พุตจริง (ground truth) หรือ ค่าเฉลย. รูป 3.3 แสดงผลต่างระหว่างค่าที่ทำนายและค่าเอาต์พุตจริง. ความผิดพลาดรวม สามารถคำนวณได้ดังสมการ 3.2.

$$E = \frac{1}{2} \sum_{n=1}^N E_n = \frac{1}{2} \sum_{n=1}^N (\hat{y}_n - y_n)^2. \quad (3.2)$$

การยกกำลังสอง ช่วยให้ความผิดพลาดจากการทายขาดไม่ไปหักล้างกับความผิดพลาดจากการทายเกิน. ค่าคงที่  $\frac{1}{2}$  ถูกใช้เพื่อความสะดวก (ที่จะได้เห็นต่อไป เมื่อทำการหาอนุพันธ์).

ด้วยวิธีวัดความผิดพลาดนี้ การฝึกฟังก์ชันพหุนามระดับขั้น  $m$  ก็สามารถทำได้โดย  $\mathbf{w}^* = \arg \min_{\mathbf{w}} E$ . เมื่อจับการฝึกแล้ว ค่าพารามิเตอร์  $\mathbf{w}^*$  จะถูกนำไปใช้กับฟังก์ชันพหุนามเพื่อทำนาย. ฟังก์ชันพร้อมด้วยค่าพารามิเตอร์ที่ได้จากการฝึก มักจะเรียกรวม ๆ ว่า แบบจำลองที่ฝึกแล้ว.



รูปที่ 3.3: ความผิดพลาดของการทำนายที่จุดข้อมูลต่าง ๆ. จุดดาวสีแดง แสดงจุดข้อมูล. เส้นทึบสีฟ้า แทนพฤติกรรมการทำนายจากแบบจำลอง. ความผิดพลาดวัดจากค่า  $x$  เดียว กับ ค่าที่แบบจำลองทำนายห่างจากค่า  $y$  ของจุดข้อมูลเท่าไร ในภาพ เน้นความห่างนี้ด้วยเส้นปรสีเขียว.

ตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง. สำหรับตัวอย่างข้อมูลในรูป 3.1 สมมติระดับขั้นที่เลือกคือ ระดับขั้นหนึ่ง ( $m = 1$ ) นั่นคือ  $\hat{y} = w_0 + w_1 x$ . ในการฝึกแบบจำลอง ซึ่งคือการทำ  $w_0$  และ  $w_1$  ที่ทำนายผิดพลาดน้อยที่สุด นั่นคือ การหา  $w_0^*, w_1^* = \arg \min_{w_0, w_1} E$  เมื่อ  $E = \frac{1}{2} \sum_{n=1}^N E_n$  และ  $E_n = (w_0 + w_1 x_n - y_n)^2$ .

ค่าความผิดพลาดต่ำสุด เกิดเมื่อ  $\frac{\partial E}{\partial w_0} = 0$  และ  $\frac{\partial E}{\partial w_1} = 0$  ซึ่งเมื่อเขียน  $E$  ในรูปฟังก์ชันของ  $w_0$  และ  $w_1$  จะได้

$$\frac{\partial \frac{1}{2} \sum_{n=1}^N \{w_0 + w_1 x_n - y_n\}^2}{\partial w_0} = 0, \quad (3.3)$$

$$\frac{\partial \frac{1}{2} \sum_{n=1}^N \{w_0 + w_1 x_n - y_n\}^2}{\partial w_1} = 0 \quad (3.4)$$

และหลังจากหาอนุพันธ์เสร็จจะได้

$$\sum_{n=1}^N \{(w_0 + w_1 x_n - y_n) \cdot (1 + 0 - 0)\} = 0, \quad (3.5)$$

$$\sum_{n=1}^N \{(w_0 + w_1 x_n - y_n) \cdot (0 + x_n - 0)\} = 0. \quad (3.6)$$

ทำการจัดรูปใหม่ โดยเรียงตามพารามิเตอร์ จะได้

$$w_0 \sum_{n=1}^N \{1\} + w_1 \sum_{n=1}^N \{x_n\} - \sum_{n=1}^N \{y_n\} = 0, \quad (3.7)$$

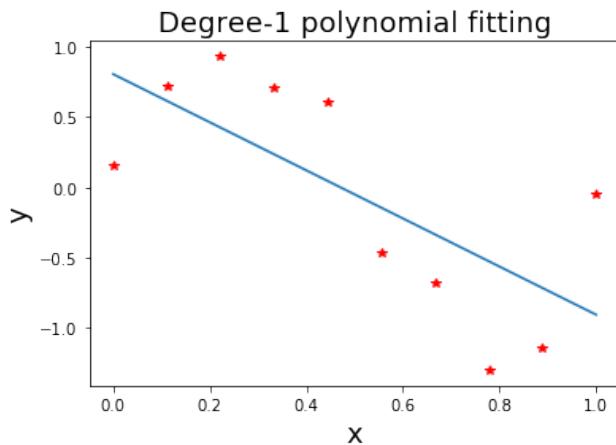
$$w_0 \sum_{n=1}^N \{x_n\} + w_1 \sum_{n=1}^N \{x_n^2\} - \sum_{n=1}^N \{y_n \cdot x_n\} = 0 \quad (3.8)$$

ซึ่งเมื่อจัดรูปสมการ 3.7 และ 3.8 ให้อยู่ในรูปเมทริกซ์จะได้

$$\begin{bmatrix} N & \sum_{n=1}^N x_n \\ \sum_{n=1}^N x_n & \sum_{n=1}^N x_n^2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N y_n \\ \sum_{n=1}^N y_n \cdot x_n \end{bmatrix}. \quad (3.9)$$

เมื่อนำค่าจุดข้อมูลในรูป 3.1 มาคำนวณ ผลจะได้ว่า  $N = 10$ ,  $\sum_{n=1}^N x_n = 5$ ,  $\sum_{n=1}^N x_n^2 = 3.519$ ,  $\sum_{n=1}^N y_n = -0.498$ , และ  $\sum_{n=1}^N y_n x_n = -1.992$ . เมื่อแก้สมการแล้วจะได้ค่า  $[w_0, w_1]^T = [0.805, -1.710]^T$ . นั่นคือ แบบจำลอง  $\hat{y} = 0.805 - 1.71x$  พฤติกรรมของแบบจำลองนี้แสดงดังในรูป 3.4.

การใช้งาน หรือการทำนายด้วยแบบจำลอง  $\hat{y} = 0.805 - 1.71x$  คือ การคำนวณโดยแทนค่า  $x$  ที่ถูกมาลงไว้ เช่น ที่  $x = 0.5$  แบบจำลองนี้ทำนาย  $\hat{y} = 0.805 - 1.71(0.5) = -0.05$ . ความสามารถของแบบจำลองนี้ ประเมินคร่าว ๆ ได้จากค่าความผิดพลาดรวม (สมการ 3.2)  $E = 1.487$ .



รูปที่ 3.4: พฤติกรรมของแบบจำลองพหุนามระดับขั้นหนึ่งที่ฝึกแล้ว (เส้นทึบสีฟ้า) กับจุดข้อมูลที่ฝึก (ดาวสีแดง).

#### เกร็ดความรู้สมองมนุษย์ (เรียบเรียงจาก [143] [82] และ [217])

โดยเฉลี่ยแล้ว สมองมนุษย์มีขนาดประมาณ 1.13 ถึง 1.26 ลิตร และหนักประมาณ 1.3 กก. ใช้ออกซิเจนประมาณ 20 เปอร์เซ็นของปริมาณทั้งหมดที่ร่างกายรับเข้าไป และใช้กำลังงานประมาณ 25 วัตต์[75].

สมองเชื่อมต่อกับส่วนอื่น ๆ ของร่างกายผ่านไขสันหลัง และระบบประสาทนอกส่วนกลาง (Peripheral Nervous System คำย่อ PNS) ไปสันหลังทำหน้าที่หลัก ๆ คือเชื่อมต่อสัญญาณควบคุมจากสมองไปยังส่วนต่าง ๆ ของร่างกาย และส่งผ่านสัญญาณรับรู้จากส่วนต่าง ๆ ของร่างกายกลับไปยังสมอง และไขสันหลังเองก็มีระบบประสาทของตัวเองที่ช่วยทำงาน เช่น การควบคุมการตอบสนองฉับพลัน. ระบบประสาทนอกส่วนกลาง มีหน้าที่หลัก คือเชื่อมต่อสัญญาณจากสมองและไขสันหลังไปสู่อวัยวะต่าง ๆ.

การทำงานของสมองมีลักษณะคล้ายคณะกรรมการของกลุ่มผู้เชี่ยวชาญจำนวนมาก นั่นคือ ส่วนต่าง ๆ ของสมองทำงานร่วมกัน แต่ว่าแต่ละส่วนของสมองมีหน้าที่เฉพาะด้าน. เราอาจมองได้ว่าส่วนของสมองมีสามส่วนใหญ่ ๆ คือ สมองส่วนบน (forebrain), สมองส่วนกลาง (midbrain), และ สมองส่วนล่าง (hindbrain).

สมองส่วนล่างนับรวมส่วนบนของไขสันหลัง ก้านสมอง (brain stem) และ เช雷เบลัม (cerebellum). สมองส่วนล่างจะควบคุมการทำงานที่เป็นพื้นฐานของการดำรงชีพ เช่น การหายใจ และการเต้นของหัวใจ. เช雷เบลัมช่วยประสานงานเรื่องการเคลื่อนไหวและ การเรียนรู้ของการเคลื่อนไหวที่เกิดจากการฝึกทำซ้ำ ๆ เช่น การเล่นเปียโนหรือการตีลูกเทนนิส จะอาศัยการทำงานของเช雷เบลัม ช่วย.

สมองส่วนกลางอยู่ด้านบนของก้านสมอง หน้าที่เกี่ยวกับการควบคุมการตอบสนองแบบฉับพลัน และเป็นส่วนหนึ่งในระบบการควบคุมการเคลื่อนไหวของดวงตาและการเคลื่อนไหวโดยสมัครใจอื่น ๆ. สมองส่วนกลางนี้มีส่วนที่ทำงานประมวลผลภาพอยู่ด้วย. สภาวะเห็นทั้งหมด (blindsight) เป็นสภาวะของผู้พิการทางสายตา ที่การพิการเกิดจากส่วนประมวลผลภาพหลักที่เปลือกสมองส่วนการเห็น (visual cortex ซึ่งจัดอยู่ในสมองส่วนบน) ไม่สามารถทำหน้าที่ได้ แต่ดวงตาและส่วนอื่น ๆ ในระบบการมองเห็น รวมถึงส่วนประมวลผลภาพของสมองส่วนกลางยังดีอยู่. สภาวะเห็นนี้ ตัวผู้พิการจะไม่รับรู้ถึงการมองเห็น แต่เมื่อมีการทดลอง โดยบังคับให้ผู้มีสภาวะเห็นทั้งหมดบรรยายรูปร่างหรือทำแท่งของวัตถุด้วยการเดา ผู้มีสภาวะเห็นทั้งหมดจะบรรยายได้ถูกต้องทั้งทั้งรูปร่าง ตำแหน่ง และการเคลื่อนไหว ซึ่งความถูกต้องแม่นยำที่ได้สูงมากเกินกว่าที่จะได้มาจากการคาดเดา. คำอธิบายสภาวะนี้ก็คือ สมองกลับไปใช้ผลการประมวลภาพจากสมองส่วนกลาง ซึ่งแม้จะไม่มีความสามารถในการประมวลผลได้เท่ากับเปลือกสมองส่วนการเห็น แต่ก็ช่วยให้เกิดการมองเห็นได้จิตสำนึกนี้เกิดขึ้นได้.

เนื่องจากระบบประมวลภาพในสมองมีทั้งที่สมองส่วนกลางและบริเวณเปลือกสมองส่วนการเห็นในสมองส่วนบน ทฤษฎีวิัฒนาการเชื่อว่า การประมวลภาพที่สมองส่วนกลางเป็นวิัฒนาการในช่วงก่อน (สัตว์หลายชนิด เช่น กบ) ใช้การประมวลภาพที่

สมองส่วนกลางเป็นหลัก) และเปลือกสมองส่วนการเห็นเป็นวิวัฒนาการในช่วงต่อมา. ผู้เชี่ยวชาญด้านประสาทวิทยาเดวิด ลินเดน (ผู้เขียนหนังสือ Accidental Mind[121]) ได้อธิบายเพิ่มเติมในการสนทนาร่วมกันว่า หากการประมวลผลภาพของสมองส่วนกลางเสียหาย แต่ส่วนอื่น ๆ ในระบบการมองเห็นยังดีอยู่ รวมถึงเปลือกสมองส่วนการเห็นที่ยังดีอยู่ ผู้ป่วยจะรับรู้ถึงการมองเห็นได้ แต่พบว่าผู้ป่วยจะมีการตอบสนองการประสานงานระหว่างมือและตา (hand-eye coordination) ที่ช้าลงอย่างชัดเจน.

สมองส่วนบนเป็นส่วนที่ใหญ่ที่สุดในสมองส่วน. สมองส่วนบนประกอบด้วยเยเรบรัม (cerebrum) และล่วนสมองใน (the inner brain). หมายเหตุ เยเรบรัม (ของสมองส่วนบน) มาจากภาษาลาติน แปลตรงตัวว่า สมอง ขณะที่ เยเรเบลัม (ของสมองส่วนล่าง) มาจากภาษาลาติน ซึ่งแปลตรงตัวว่า สมองน้อย. เยเรบรัมคือภาพของสมองที่คนทั่วไปจะนิยมเมื่อกล่าวถึงสมอง. เยเรบรัม ทำหน้าที่หลักในการรับรู้ ความจำ การวางแผน การคิด การจินตนาการ รวมถึงศีลธรรม นิสัย และบุคลิกภาพ. เมื่อมองจากด้านบน เยเรบรัมดูเหมือนจะแบ่งได้เป็นซีกซ้ายและซีกขวา โดยมีดูเหมือนมีร่องแบ่งสมองสองซีกนี้ออกจากกัน. สมองทั้งสองซีกเชื่อมต่อกันผ่านเส้นใยประสาทเรียกว่า คอร์ปัส คาโลซัม (corpus callosum). สมองทั้งสองซีกนี้ทำงานร่วมกัน แต่สมองซีกซ้ายจะควบคุมการทำงานของร่างกายซีกขวา และสมองซีกขวาจะควบคุมการทำงานของร่างกายซีกซ้าย โดยสมองซีกซ้ายจะเด่นด้านการทำงานเกี่ยวกับภาษา การวิเคราะห์รายละเอียด และทักษะเชิงรูปธรรม ในขณะที่สมองซีกขวาจะเด่นด้านการอ่านภาพรวม และทักษะเชิงนามธรรม. การทำงานไขว้ระหว่างซีกสมองกับร่างกายนั้น แม้จะยังไม่มีคำอธิบายว่าเหตุใดกลไกของร่างกายจึงเป็นเช่นนั้น แต่ข้อเท็จจริงคือสัญญาณจากสมองซีกหนึ่งจะไขว้ไปบังคับร่างกายอีกซีกหนึ่ง ดังนั้น หากสมองซีกหนึ่งเสียหาย ร่างกายอีกซีกหนึ่งจะได้รับผลกระทบ เช่น ผู้ป่วยโรคหลอดเลือดสมอง เมื่อเกิดสมองซีกขวาเสียหาย จะส่งผลให้ผู้ป่วยเป็นอัมพาตในซีกซ้ายของร่างกาย.

การศึกษาที่น่าสนใจเกี่ยวกับสมองซีกซ้ายและขวา หลายกรณีได้มาจาก การศึกษาผู้ป่วยโรคลมชักรุนแรง ที่แพทย์ต้องตัด คอร์ปัส คาโลซัมเพื่อลดความรุนแรงของการลมชักไม่ให้แพร่ขยายข้ามซีกสมองได้. หนึ่งในตัวอย่าง[204] คือ การศึกษาที่นำผู้ป่วยที่ผ่านการตัดการเชื่อมต่อระหว่างสมองซีกซ้ายและขวาออกจากกัน มาใส่คอนแทกเลนส์พิเศษเพื่อแยกการมองเห็นระหว่างตาซ้ายและตาขวาออกจากกัน. ตาซ้ายและขวาการซีกซ้ายเชื่อมโยงกับสมองซีกขวา ตาขวาและขวาการซีกขวาเชื่อมโยงกับสมองซีกซ้าย. เมื่อให้ตาขวาบีบภาพของเท้าของໄก และให้ตาซ้ายบีบภาพของบ้านที่ถูกหิมะท่วง พร้อมสั่นให้ผู้ทดลองซึ่งเลือกภาพที่เกี่ยวข้องด้วยมือซ้ายและขวา ผู้ทดลองซึ่งมีขวาไปที่ภาพตัวแม่ໄก และมีซ้ายไปที่ภาพพลวั่น ผู้ทดลอง อธิบายถึงเท้าໄกได้ แต่ไม่สามารถอธิบายภาพของบ้านที่ถูกหิมะท่วงได้ และเมื่อให้ผู้ทดลองอธิบายเหตุผลที่ซึ่งเลือกภาพแม่ໄก และพลวั่น สมองส่วนซ้าย ซึ่งไม่ได้รับรู้ภาพของบ้านที่ถูกหิมะท่วง ก็พยายามอธิบายไปว่า เท้าໄกเกี่ยวข้องกับแม่ໄก และพลวั่นเกี่ยวข้องคือเป็นเครื่องมือตักมูลໄก. กรณีนี้ ผู้เชี่ยวชาญอธิบายว่า สมองซีกซ้ายซึ่งมีความสามารถทางภาษา แต่ไม่ได้รับภาพที่สมองซีกขวาเห็น ไม่ได้รับรู้ถึงภาพบ้านหิมะท่วง แต่สมองซีกขวา แม้จะรับรู้ภาพของบ้านหิมะท่วงและยังบังคับมือซ้ายไปที่พลวั่น ซึ่งเป็นสิ่งที่มักจะเชื่อมโยงกับภาพหิมะท่วง ในกลุ่มคนที่คุ้นเคยกับสภาพหิมะ แต่สมองซีกขวาไม่มีความสามารถทางภาษา จึงไม่สามารถอธิบายออกมากเป็นคำพูดได้.

เยเรบรัมแต่ละซีกยังสามารถแบ่งเป็นส่วนย่อย ๆ ลงมาได้อีก ซึ่งแต่ละส่วนของเยเรบรัมมักจะเรียกว่ากลีบ(lobe). เยเรบรัมมีกลีบหลัก ๆ เช่น กลีบหน้า (frontal lobe), กลีบข้าง (parietal lobe), กลีบท้ายทอย (occipital lobe), และกลีบขมับ (temporal lobe). สมองกลีบหน้าจะอยู่บริเวณหลังหน้า部分ของเรา และทำหน้าที่เกี่ยวกับ การวางแผน การจินตนาการ ลึกลง แต่กลีบหน้าที่เกี่ยวกับการควบคุมการเหตุผล การควบคุมตัวเอง บุคคลิกภาพ และ ศีลธรรม. ลีกเข้าไปท้าย ๆ กลีบจะเป็นบริเวณที่ทำหน้าที่เกี่ยวกับการควบคุมการเคลื่อนไหว. ในกลีบหน้าของสมองซีกซ้ายจะมีบริเวณไบรก้า (Broca's area) ซึ่งเป็นส่วนที่ทำหน้าที่เกี่ยวกับการใช้ภาษา.

สมองกลีบข้างซึ่งอยู่ด้านหลังกลีบหน้าเข้ามา (บริเวณใต้กลางกระหม่อม) ทำหน้าที่เกี่ยวกับรัส กลิน สัมผัส รวมถึงการรับรู้ การเคลื่อนไหวของร่างกาย ความสามารถในการอ่านหนังสือและการคิดคำนวนตัวเลข ก็เกี่ยวข้องกับสมองกลีบข้าง. สมองกลีบท้ายทอยอยู่ด้านหลังกลีบข้างไปทางหน้าหลัง (บริเวณท้ายทอย) ทำหน้าที่หลักเกี่ยวกับการมองเห็น. เปลือกสมองส่วนการเห็น ซึ่งเป็นส่วนประมวลผลการมองเห็นหลัก ก็อยู่ในบริเวณกลีบท้ายทอย. สมองกลีบขมับจะอยู่ใต้กลีบหน้าและกลีบข้าง ซึ่งเมื่อเทียบกับภายนอกแล้วจะอยู่บริเวณขมับ. สมองกลีบขมับทำหน้าที่หลักเกี่ยวกับการประมวลผลเสียงต่าง ๆ และมีหน้าที่ช่วยในการรวมความจำและความรับรู้ต่าง ๆ ทั้งภาพ เสียง กลิ่น และ สัมผัส เข้าด้วยกัน.

ที่ผิวชั้นนอกของเยเรบรัมจะเป็นชั้นของเนื้อเยื่อที่หนาประมาณ 2 ถึง 4 มิลลิเมตร ซึ่งเรียกว่า เยเรบรอคortex (cerebral cortex). การประมวลผลของสมองส่วนใหญ่เชื่อกันว่าเกิดขึ้นภายในเนื้อเยื่อส่วนนี้ เนื้อเยื่อส่วนนี้จะมีสีเข้มกว่าเนื้อเยื่อส่วนด้านใน และมีถруктурอ้างถึงในชื่อของเนื้อเทา (gray matter) เปรียบเทียบกับเนื้อขาว (white matter) ซึ่งอยู่ภายใต้ใน. เนื้อเทาจะประกอบ

ด้วยเซลล์ประสาท หลอดเลือดฟอย และ เซลล์เกลีย. เซลล์ประสาทในเนื้อเทาจะมีไขมันที่เป็นจวนน้อยกว่าเซลล์ประสาทในเนื้อขาว จึงทำให้สีของเนื้อเยื่อโดยรวมดูเข้มกว่า. (ดูรายละเอียดของเซลล์ประสาท ในเกร็ดความรู้เซลล์ประสาท.) เนื่องจากเซโรลอลคอร์เทกซ์เป็นผิวของสมอง รอยหยักของสมองจะช่วยเพิ่มพื้นที่ผิวและบริมาณของเนื้อเทาซึ่งสัมพันธ์กับบริมาณของข้อมูลที่สมองสามารถประมวลผลได้.

ส่วนสมองในเป็นอีกบิเวณในสมองส่วนบน. ส่วนสมองในนี้จะเริ่มต่อไปสันหลังเข้ากับเซเรบรม. ส่วนสมองในทำหน้าที่เกี่ยวารมณ์ มีส่วนในการเปลี่ยนแปลงการรับรู้และการตอบสนองไปตามสถานะของอารมณ์ในขณะนั้น ๆ มีส่วนช่วยเริ่มการเคลื่อนไหวต่าง ๆ ที่เราทำโดยเราไม่ต้องคิดถึงการเคลื่อนไหวเหล่านั้น และมีส่วนสำคัญในการควบคุมการสร้างความจำ. ส่วนประกอบต่าง ๆ ของส่วนสมองในนี้จะมีเป็นคู่ ๆ ทางซ้ายและขวา โดยมีส่วนประกอบที่สำคัญ เช่น ไฮปोราลามัส (hypothalamus) הרามัส (thalamus) bazal ganglia (basal ganglia) อะมิกดาลา (amygdala) และ hippocampus. ไฮปอราลามัสเป็นเมืองศูนย์กลางการจัดการอารมณ์. รากสามช่วงจัดการข้อมูลที่ผ่านไปมาระหว่างเซเรบรมและสันหลัง. bazal gangliaช่วยการเริ่มและประสานงานการเคลื่อนไหวต่าง ๆ. โรคพาร์กินสันซึ่งผู้ป่วยจะมีอาการที่เด่นชัดคือมีปัญหากับการเคลื่อนไหว เช่น อาการสั่น เดินหรือเคลื่อนไหวได้ช้า เป็นโรคที่เกี่ยวพันกับเซลล์ประสาทที่เชื่อมต่อกับ bazal ganglia นี้. อะมิกดาลาทำหน้าที่เกี่ยวกับอารมณ์ ความกลัว ความก้าวร้าว และ ความจำที่เกี่ยวข้องกับอารมณ์ความรู้สึก. มีงานศึกษาที่พบความเกี่ยวข้องกันระหว่างขนาดของอะมิกดาลากับความสัมพันธ์ทางสังคมของบุคคลนั้น. อิปโปแคมปัสทำหน้าที่จัดส่งความจำใหม่ไปเก็บในตำแหน่งที่เหมาะสมในเซเรบรม และคืนหาความจำที่ต้องการจากเซเรบรม. ผู้ป่วยที่สูญเสียอิปโปแคมปัสไปจะสูญเสียความสามารถในการสร้างความทรงจำใหม่.

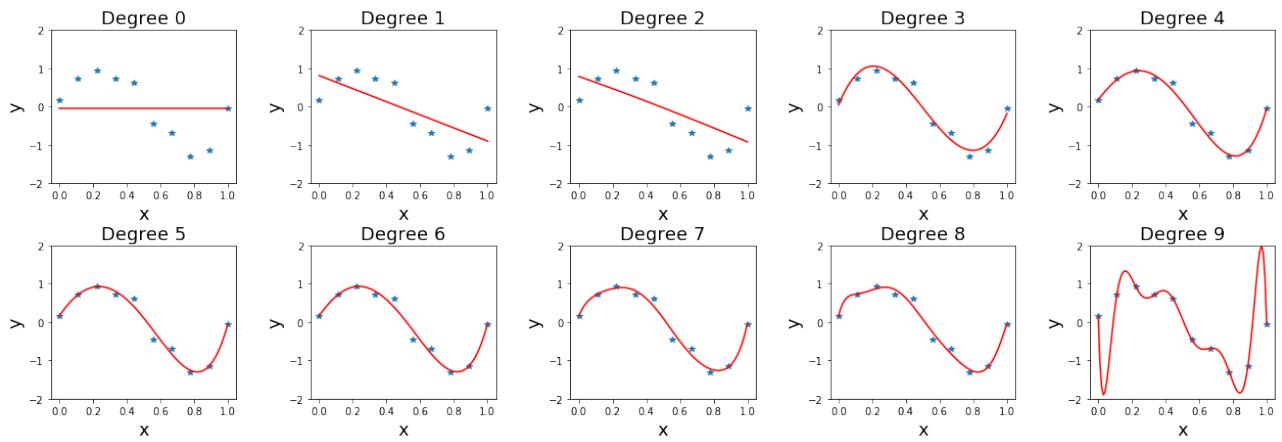
### 3.2 คุณสมบัติความทั่วไปและการเลือกแบบจำลอง

หัวข้อ 3.1 แสดงตัวอย่างของการปรับเส้นโค้ง ด้วยฟังก์ชันพหุนามระดับขั้นหนึ่ง. ระดับขั้นของฟังก์ชันพหุนาม เป็นอภิธานพารามิเตอร์ ของแบบจำลองพหุนาม. การสร้างแบบจำลองสามารถใช้ระดับขั้นใดก็ได้ แต่การเลือกระดับขั้นที่เหมาะสม เพื่อได้แบบจำลองที่ดี จะได้อภิปรายในหัวข้อนี้. รูป 3.5 แสดงพฤติกรรมการทำนายที่ระดับขั้นต่าง ๆ.

จากรูป 3.5 สังเกตว่า ระดับขั้นที่สูงขึ้นช่วยให้แบบจำลองยืดหยุ่นมากขึ้น และสามารถปรับตัวเข้าหากุจุข้อมูลได้ง่ายขึ้น และที่ระดับขั้นสูงมาก ๆ เช่น ที่ระดับขั้นเก้า ฟังก์ชันพหุนามสามารถปรับเข้าหากุจุข้อมูลได้ใกล้มาก ๆ. แต่ที่ระดับขั้นเก้า พฤติกรรมการทำนาย ระหว่างกุจุข้อมูลมีการเปลี่ยนแปลงรุนแรงมาก.

การสร้างแบบจำลองทำนาย ต้องการแบบจำลองทำนายที่มีคุณสมบัติความทั่วไป. คุณสมบัติที่แบบจำลองสามารถทำนายได้ดี แม้กับข้อมูลที่ไม่เคยเห็นมาก่อน จะเรียกว่า **คุณสมบัติความทั่วไป** (generalization).

ปกติแล้ว ข้อมูลจะมีสัญญาณรบกวนประกอบเข้ามาด้วย แบบจำลองที่ดีควรจะจับสารสนเทศที่สำคัญของข้อมูล. เมื่อแบบจำลองปรับตัวเข้ากับข้อมูลที่ใช้ฝึกมากเกินไป แบบจำลองอาจจะจับสัญญาณรบกวนเข้าไปปนกับสารสนเทศที่สำคัญ ซึ่งจะส่งผลให้แบบจำลองสามารถทำนายข้อมูลที่ใช้ฝึกได้อย่างแม่นยำ แต่อาจไม่สามารถทำนายข้อมูลใหม่ได้ดี.



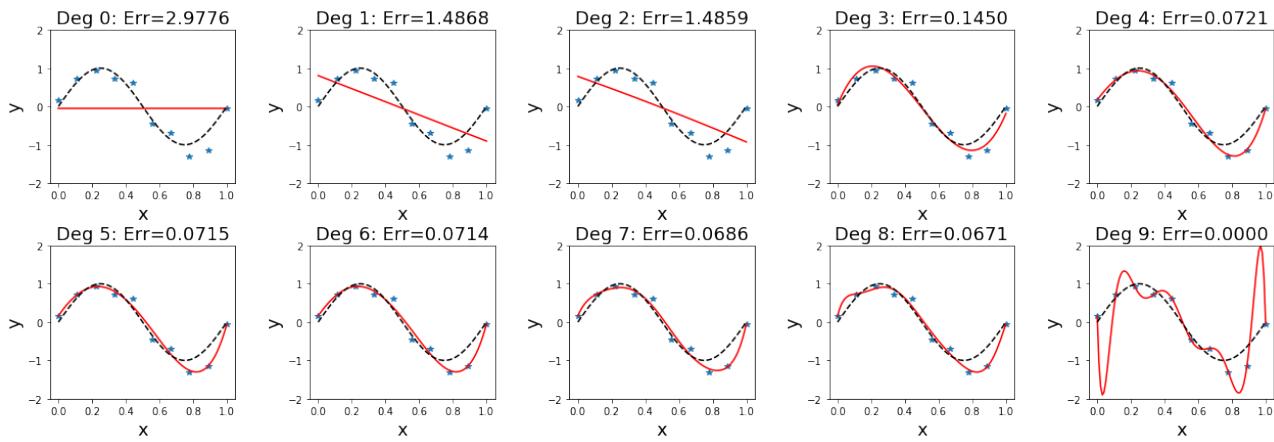
รูปที่ 3.5: พฤติกรรมการทำนายของแบบจำลองที่ระดับขั้นต่าง ๆ. เส้นกราฟสีแดง แสดงพฤติกรรมของฟังก์ชันพหุนาม. สัญลักษณ์ดาวสีฟ้า แทนจุดข้อมูล.

ตัวอย่างข้อมูลที่แสดงนี้ จริง ๆ แล้วสร้างมาจากความสัมพันธ์  $y = \sin(2\pi x) + \varepsilon$  เมื่อสัญญาณรบกวน  $\varepsilon$  สุ่มขึ้นมาจากการแจกแจงเกาส์เชียน ที่มีค่าเฉลี่ยเป็น 0 และค่าเบี่ยงเบนมาตรฐานเป็น 0.3. นั่นคือ  $\varepsilon \sim \mathcal{N}(0, 0.3)$ . รูป 3.6 แสดงพฤติกรรมการทำนาย เปรียบเทียบกับสารสนเทศที่สำคัญของข้อมูล (แสดงด้วยเส้นประสีดำ). แบบจำลองที่ดี คือแบบจำลองที่สามารถประมาณสารสนเทศที่สำคัญของข้อมูลได้ แต่ในทางปฏิบัติ การสร้างแบบจำลอง ไม่ได้รู้สารสนเทศที่สำคัญ เพราจะหากว่า ความสามารถสร้างแบบจำลอง จากสารสนเทศที่สำคัญที่รู้นั้นได้โดยตรง. ดังนั้น การใช้งานแบบจำลองทำนาย ในทางปฏิบัติ ต้องการกลไก หรือกระบวนการที่จะทราบตรวจสอบว่า แบบจำลองยังมีคุณสมบัติความทวีไปดีอยู่หรือไม่.

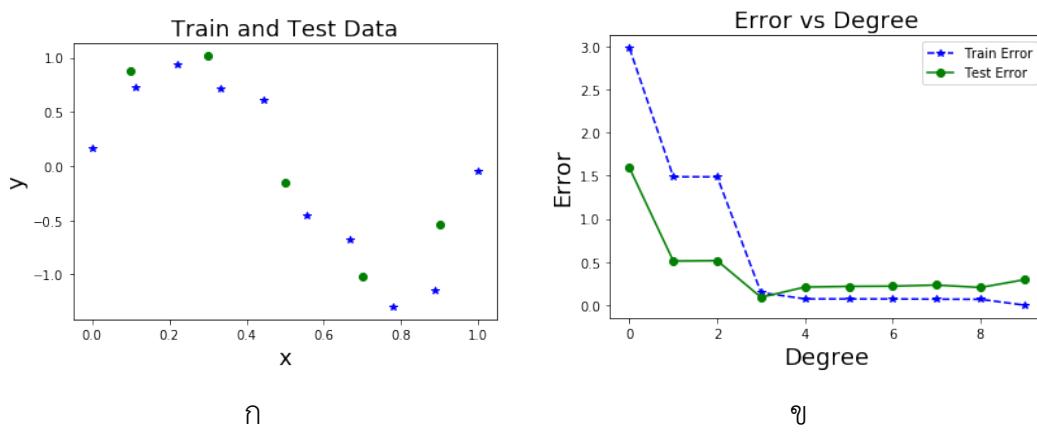
กระบวนการที่ใช้ตรวจสอบ คุณสมบัติความทวีไป ที่ตรงมาตรงไปที่สุด คือการใช้ข้อมูลทดสอบ. **ข้อมูลทดสอบ** (test data) คือข้อมูลอีกชุด ที่ไม่ได้ถูกใช้ในการวนการฝึก เป็นข้อมูลที่แบบจำลองไม่เคยเห็นเลย จะใช้เพื่อทดสอบแบบจำลองเท่านั้น. เพื่อให้สามารถจำแนกได้ชัดเจน ว่ากำลังพูดถึงข้อมูลสำหรับจุดประสงค์ได้อยู่ ข้อมูลที่ใช้ฝึกแบบจำลอง จะเรียกว่า **ข้อมูลฝึก** (training data).

จากตัวอย่างการปรับเส้นโค้งข้างต้น สมมติว่าข้อมูลทดสอบ ที่ได้แยกไว้คือ  $(0.1, 0.881), (0.3, 1.015), (0.5, -0.152), (0.7, -1.015)$ , และ  $(0.9, -0.537)$  รวม 5 จุดข้อมูล. รูป 3.7 ภาพ ก แสดงจุดข้อมูลฝึก (10 จุด) และจุดข้อมูลทดสอบ (5 จุด). ภาพ ข แสดงค่าผิดพลาดของแบบจำลองที่ระดับขั้นต่าง ๆ เมื่อทดสอบกับชุดข้อมูลฝึก และชุดข้อมูลทดสอบ.

จาก ภาพ ข สังเกตว่า ค่าผิดพลาดเมื่อทดสอบกับชุดฝึก ซึ่งค่านี้มักเรียกว่า **ค่าผิดพลาดชุดฝึก** (training error) มีค่าลดลงเรื่อย ๆ เมื่อระดับขั้นเพิ่มขึ้น. แต่ค่าผิดพลาดเมื่อทดสอบกับชุดทดสอบ ซึ่งค่านี้มักเรียกว่า **ค่าผิดพลาดชุดทดสอบ** (test error) มีค่าลดลงจนต่ำสุดที่ ในตัวอย่างนี้ เป็นระดับขั้นสาม และค่ากลับ



รูปที่ 3.6: พฤติกรรมการทำนายของแบบจำลองที่ระดับขั้นต่าง ๆ ที่แสดงความสัมพันธ์จริงด้วยเส้นประสีดำเนีก. ความสัมพันธ์จริง คือความสัมพันธ์ระหว่าง  $x$  และ  $y$  ที่ใช้ในกระบวนการสร้างจุดข้อมูลสำหรับตัวอย่าง. กระบวนการสร้างจุดข้อมูลสำหรับตัวอย่างนี้ ใช้ความสัมพันธ์จริง  $y = \sin(2\pi x)$  ประกอบกับสัญญาณรบกวน  $\epsilon$ . นั่นคือ จุดข้อมูลสร้างจาก  $y = \sin(2\pi x) + \epsilon$ . ค่าผิดพลาด (ระบุด้วยคำย่อ Err) เนื่องแต่ภาพคือ ค่าผิดพลาดของแต่ละแบบจำลอง โดยคิดจาก 10 จุดข้อมูลฝึก.



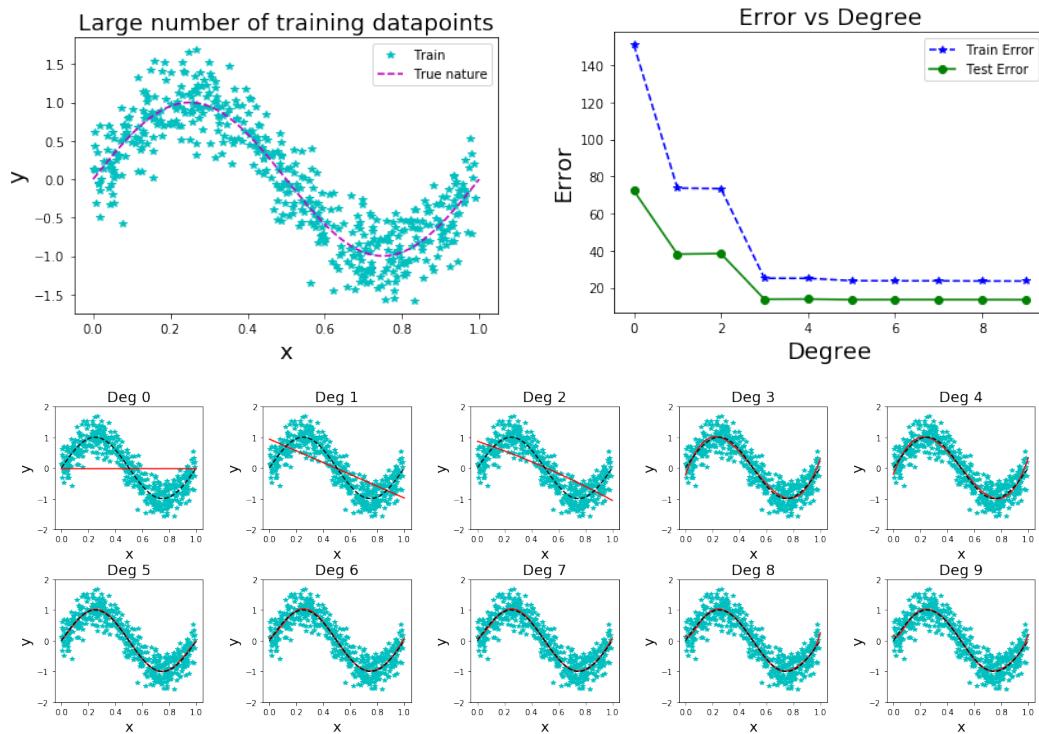
รูปที่ 3.7: ภาพ ก แสดงข้อมูลฝึก (ดาวสีฟ้า) และข้อมูลทดสอบ (วงกลมสีเขียว). ภาพ ข แสดงค่าผิดพลาดของแบบจำลอง เมื่อทดสอบกับจุดข้อมูลฝึกและจุดข้อมูลทดสอบ.

เพิ่มขึ้น เมื่อระดับขั้นเพิ่มขึ้นหลังจากนั้น. ณ จุดที่ค่าผิดพลาดชุดฝึกต่ำลง แต่ค่าผิดพลาดชุดทดสอบกลับสูงขึ้น เป็นสัญญาณป่งชี้ว่า แบบจำลองเริ่มเสียคุณสมบัติความทั่วไป ซึ่งมักเรียกว่า แบบจำลองเกิดการโอเวอร์ฟิต (overfitting).

ผลจากค่าผิดพลาดชุดทดสอบ ปัจจุบันแบบจำลองพหุนามระดับขั้นสี่ขึ้นไป เริ่มเกิดโอเวอร์ฟิต และแบบจำลองที่มีคุณสมบัติความทั่วไปดีที่สุดในการทดสอบนี้ คือ แบบจำลองพหุนามระดับขั้นสาม. หากสังเกต รูป 3.6 จะเห็นว่า ระดับขั้นสามให้การประมาณรวมชาติจริงของข้อมูลดีที่สุด (พฤติกรรมของแบบจำลอง เส้นทึบแดง มีลักษณะใกล้เคียงกับรวมชาติจริง เส้นประสีดำเนีกมากกว่าระดับขั้นอื่น ๆ).

การโอเวอร์ฟิตของแบบจำลอง สัมพันธ์โดยตรงกับข้อมูล โดยเฉพาะกับจำนวนจุดข้อมูล. ฟังก์ชันพหุนาม

ที่มีระดับขั้นสูง เรียกว่า เป็นแบบจำลองที่มี **ความซับซ้อน (complexity)** สูง. แบบจำลองที่มีความซับซ้อนสูง มีความยืดหยุ่นมาก. จำนวนจุดข้อมูลที่มีมากพอก จะช่วยให้เห็นสารสนเทศที่สำคัญ จากสัญญาณรบกวนที่ปนมาได้ชัดเจนขึ้น และช่วยการฝึกแบบจำลองที่มีความซับซ้อนสูง ให้มีคุณสมบัติความทวีไปดีขึ้นได้.



รูปที่ 3.8: ภาพบนซ้าย แสดงจุดข้อมูลจำนวนมาก สร้างจาก  $y = \sin(2\pi x) + \epsilon$  โดยสารสนเทศที่สำคัญหาด้วยเส้นประสานยืน. จุดข้อมูล 500 จุดถูกสร้างขึ้นมาเป็นข้อมูลฝึก (จุดสีน้ำเงินเขียว). จุดข้อมูล 250 จุดถูกสร้างขึ้นมาเป็นข้อมูลทดสอบ (ไม่ได้แสดงในภาพ). ภาพบนขวา แสดงค่าผิดพลาดดุลพิศ (เส้นประสานน้ำเงิน) และดุลทดสอบ (เส้นทึบสีเขียว). ด้วยจุดข้อมูลจำนวนมาก ไม่มีสัญญาณบ่งบอกถึงการโอเวอร์ฟิตให้เห็น. ภาพล่าง แสดงพฤติกรรมการทำงานของฟังก์ชันพหุนามที่ระดับขั้นต่าง ๆ (ระบุหนึ่งในภาพ ด้วยคำย่อ เช่น Deg 0 สำหรับ ระดับขั้น 0 หรือ degree 0).

รูป 3.8 แสดงให้เห็นว่า ข้อมูลจำนวนมาก สามารถช่วยลดปัญหาโอเวอร์ฟิตได้. ภาพบนซ้ายแสดง ค่าผิดพลาดดุลทดสอบลดลงจนถึงค่อนข้างคงที่หลังจากระดับขั้นที่สาม.

ผู้เชี่ยวชาญบางคนแนะนำว่า[15] จุดข้อมูลควรมีจำนวนไม่น้อยกว่า 5 เท่าของจำนวนพารามิเตอร์ของแบบจำลอง. ตัวอย่างเช่น ฟังก์ชันพหุนามระดับขั้นเก้า มีพารามิเตอร์ 10 ตัว ดังนั้นควรจะมีจุดข้อมูลไม่น้อยกว่า 50 จุดตามคำแนะนำนี้.

อย่างไรก็ตาม นอกจากการเพิ่มจำนวนจุดข้อมูล ยังมีกลไกอื่น ๆ อีก ที่สามารถช่วยลดปัญหาโอเวอร์ฟิต เมื่อใช้แบบจำลองที่มีความซับซ้อนสูงได้ เช่น การใช้แนวทางเบย์เชียน (Bayesian) หรือ การทำเรกูลาริซ์.

อีกประเด็นหนึ่งที่น่าสนใจ สังเกตระดับค่าผิดพลาดที่แสดงในรูป 3.8 เปรียบเทียบกับที่แสดงในรูป 3.7

ระดับค่าที่แสดงในรูป 3.7 (ภาพซ้าย) มีค่าค่อนข้างต่ำ (แกน  $y$  สูงสุดประมาณ 3.0) ในขณะที่ ระดับค่าที่แสดงในรูป 3.8 (ภาพซ้ายบน) มีค่าสูงมาก (แกน  $y$  สูงสุดเกิน 140). นั่นเป็นเพราะ ทั้งสองภาพวัดค่าผิดพลาดด้วย ผลรวมค่าผิดพลาด  $E = 0.5 \sum_n E_n$ . โดยปกติแล้ว เมื่อจำนวนจุดข้อมูลมากขึ้น ผลรวมค่าผิดพลาดจะมากขึ้นตามไปด้วย. ในทางปฏิบัติ การประเมินผล มีภาระงานผลด้วย **ค่าความแม่นยำ** (accuracy) ซึ่งอาจวัดด้วย **ค่าเฉลี่ยความผิดพลาดกำลังสอง** (mean square error คำย่อ MSE) หรือ **รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง** (root mean square error คำย่อ RMSE) ที่ให้ผลคงเส้นคงวามากกว่าผลรวมค่าผิดพลาด. ค่าเฉลี่ยความผิดพลาดกำลังสอง คำนวณจาก  $\text{MSE} = \frac{1}{N} \sum_{n=1}^N E_n$  เมื่อ  $N$  คือจำนวนจุดข้อมูล และค่าผิดพลาดกำลังสองของแต่ละจุดข้อมูล  $E_n = (\hat{y}_n - y_n)^2$  โดย  $\hat{y}_n$  กับ  $y_n$  คือค่าที่ทำนาย และค่าเฉลี่ย สำหรับจุดข้อมูลที่  $n^{th}$  ตามลำดับ. ในทำนองเดียวกัน รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง คำนวณจาก  $\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N E_n}$ .

**การทำเรกูลารไรซ์.** การทำเรกูลารไรซ์ (regularization) เป็นวิธีหนึ่งที่นิยมใช้เพื่อช่วยลดปัญหาการโอเวอร์ฟิต. แนวทางหนึ่ง คือ การทำค่าน้ำหนักเลื่อน (weight decay) โดย การใส่พจน์เสมือนการลงโทษ เข้าไปในฟังก์ชันเป้าหมาย เพื่อจะถ่วงดุลไม่ให้พารามิเตอร์ของแบบจำลองมีค่าใหญ่เกินไป. สมการ 3.10 แสดงฟังก์ชันเป้าหมาย ที่ประกอบด้วยพจน์ค่าผิดพลาดและพจน์ค่าน้ำหนักเลื่อน ซึ่งเป็นพจน์แรกและพจน์ที่สองทางขวา มีอัตราการลดลงเรื่อยๆ ตามลำดับ. นั่นคือ ฟังก์ชันเป้าหมาย

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (3.10)$$

เมื่อ  $\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$  และ พารามิเตอร์  $\lambda$  ควบคุมสมดุลย์ระหว่างอิทธิพลของค่าผิดพลาดจากการทำนาย และอิทธิพลจากพจน์ค่าน้ำหนักเลื่อน. จากมุ่งมองของการหาค่าน้อยที่สุดพารามิเตอร์  $\lambda$  อาจถูกเรียกเป็น ลากرانจ์พารามิเตอร์. บีชอบ[15] ชี้ว่า บ่อยครั้งที่ พจน์ค่าน้ำหนักเลื่อน จะไม่รวม  $w_0$ . หรือ ถ้ามี  $w_0$  ก็อาจจะมีลากرانจ์พารามิเตอร์เฉพาะของตัวเอง.

รูป 3.9 แสดงผลจากการทำเรกูลารไรซ์ ด้วยการใช้ค่าลากرانจ์ต่าง ๆ. ภาพซ้ายสุด  $\lambda = 0$  เทียบเท่ากับการไม่ได้ใช้วิธีค่าน้ำหนักเลื่อน. การโอเวอร์ฟิตเห็นได้ชัดในกรณีนี้. ภาพกลาง แสดงค่าลากرانจ์ที่เหมาะสมค่าลากرانจ์พารามิเตอร์ที่เหมาะสม จะช่วยบังคับแบบจำลองที่มีความซับซ้อนสูง ให้ทำตัวเสมอ มีความซับซ้อนต่ำลง. ค่าประมาณจากแบบจำลอง (แสดงด้วยเส้นทึบสีแดง) มีลักษณะใกล้เคียงกับ  $\sin(2\pi x)$  ที่ใช้สร้างจุดข้อมูล. แต่ถ้าหากใช้ค่าลากرانจ์ค่าใหญ่เกินไป ก็อาจทำให้เกิดการอันเดอร์ฟิตได้ ดังแสดงในภาพขวาสุด.

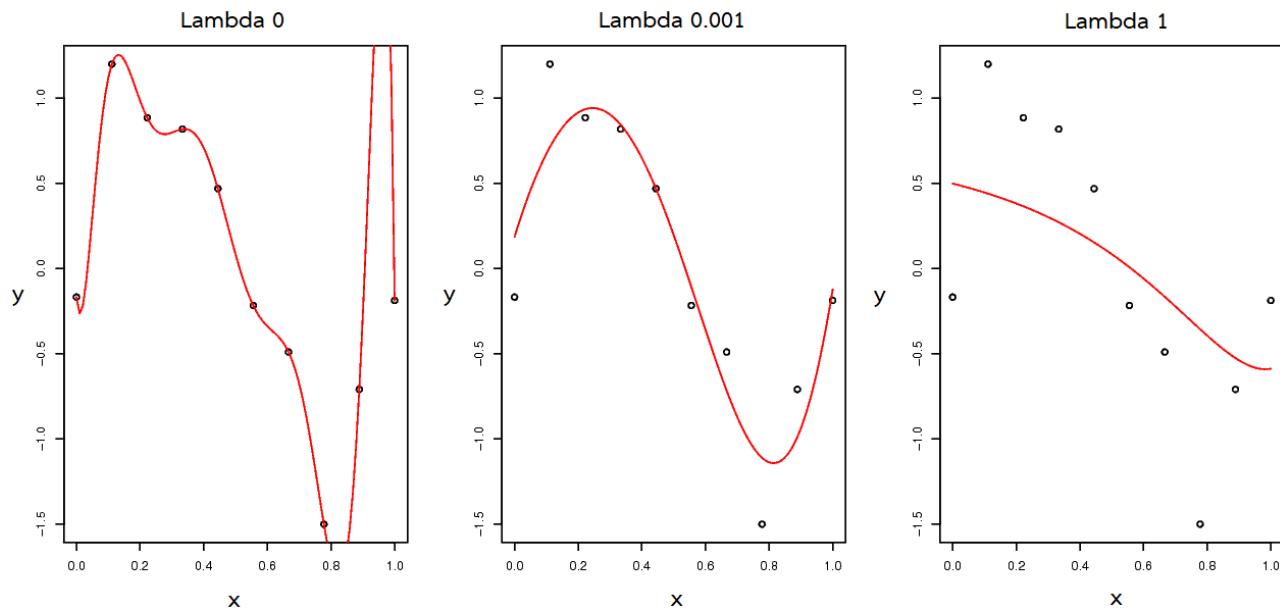
ตาราง 3.1 แสดงให้เห็นว่า ถ้าใช้ค่า  $\lambda$  ใหญ่พอดี การทำค่าน้ำหนักเสื่อม ช่วยควบคุมให้ค่าพารามิเตอร์ไม่ใหญ่เกินไปได้. แต่ถ้าใช้ค่า  $\lambda$  ใหญ่เกินไป ก็ทำให้ค่าพารามิเตอร์น้อยเกินไปได้ เช่นกัน. รูป 3.10 แสดงผลค่าผิดพลาดของแบบจำลองพหุนามระดับขั้นเก้า กับการทำค่าน้ำหนักเสื่อมที่ลากرانจ์ค่าต่าง ๆ เมื่อประเมินกับข้อมูลชุดฝึกหัดและชุดทดสอบ. สังเกตุค่าผิดพลาดของแบบจำลอง เมื่อประเมินกับชุดฝึกหัด ค่าผิดพลาดของแบบจำลองจะน้อยลง เมื่อใช้ลากرانจ์ค่าน้อย ๆ (ให้ผลคล้ายกับการใช้ฟังก์ชันพหุนามระดับขั้นสูง ๆ). ส่วนเมื่อประเมินกับชุดทดสอบ ค่าผิดพลาดของแบบจำลองจะลดลงท่าสุดที่ค่าลากرانจ์ราว ๆ 0.001 หรือ  $\log(\lambda) \approx -6.91$ .

ตารางที่ 3.1: ค่าพารามิเตอร์ของแบบจำลองพหุนาม กับการทำค่าน้ำหนักเสื่อมที่ลากرانจ์ค่าต่าง ๆ

พารามิเตอร์	$\lambda = 0$	$\lambda = 10^{-5}$	$\lambda = 1$
$w_0$	-0.17	-0.04	0.5
$w_1$	-18.6	11.85	-0.47
$w_2$	1009.96	-38.18	-0.49
$w_3$	-11723.66	37.64	-0.35
$w_4$	64085.01	-7.29	-0.2
$w_5$	-195203.42	-20.61	-0.07
$w_6$	349413.48	-0.2	0.02
$w_7$	-365010.66	20.7	0.1
$w_8$	205750.66	17.33	0.16
$w_9$	-48302.79	-21.36	0.21

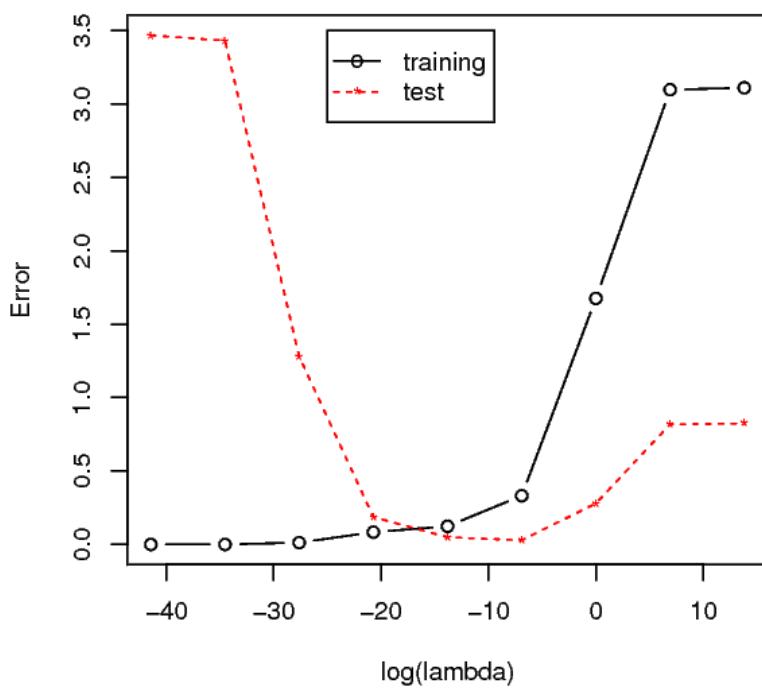
สำหรับการประเมินแบบจำลอง ปัจจัยสำคัญ คือ แบบจำลองสามารถทำงานได้ดีมาก่อน หรือ แบบจำลองมีคุณสมบัติความทวีไป. ดังนั้น เพื่อเลือกความซับซ้อนของแบบจำลอง เช่น การเลือกระดับขั้นของพหุนาม หรือการเลือกค่าลากرانจ์ของการทำค่าน้ำหนักเสื่อม จึงควรทำการวัดคุณสมบัติความทวีไปของแบบจำลองที่ความซับซ้อนต่าง ๆ กัน. วิธีที่ง่ายและตรงไปตรงมาที่สุด ก็คือ การแบ่งข้อมูลออกเป็น 2 ชุด ได้แก่ ข้อมูลชุดฝึก ที่ใช้ฝึกแบบจำลอง นั่นคือใช้หาค่าของพารามิเตอร์  $\mathbf{w}$  และชุดตรวจสอบ ที่ใช้เลือกความซับซ้อนของแบบจำลอง เช่น  $M$  หรือ  $\lambda$ .

หลังจากเลือกแบบจำลองเสร็จแล้ว เพื่อประเมินแบบจำลอง ควรจะใช้ข้อมูลชุดทดสอบ ซึ่งเป็นข้อมูลอีกชุด สำหรับการทดสอบ. การที่ต้องใช้ชุดทดสอบที่แยกออกจากนี้ เพื่อกันปัญหา ที่อาจจะเลือกแบบจำลองที่เกิดการโอเวอร์ฟิตกับชุดตรวจสอบได้. หากทำการเลือกแบบจำลองได้ดี ค่าผิดพลาดที่ประเมินกับข้อมูลชุดทดสอบ ไม่ควรห่างมากจากค่าผิดพลาดที่ประเมินกับข้อมูลชุดตรวจสอบ.



รูปที่ 3.9: พหุนามระดับขั้นเก้า กับการทำค่าน้ำหนักเสื่อม ด้วยลากกรานจ์ค่าต่าง ๆ. ภาพซ้าย แสดงการไอเวอร์ฟิต ( $\lambda = 0$ ). ภาพกลาง แสดงแบบจำลองที่เหมาะสม ( $\lambda = 0.001$ ). ภาพขวา แสดงการอันเดอร์ฟิต ( $\lambda = 1$ ). จุดวงกลม คือจุดข้อมูลฝึก และเส้น ทึบสีแดง แสดงค่าที่แบบจำลองทำนาย.

### Polynomial with regularization



รูปที่ 3.10: การทำน้ำหนักเสื่อมด้วยลากกรานจ์ค่าต่าง ๆ ประเมินด้วยข้อมูลชุดฝึกหัด (เส้นทึบสีดำ) กับ ชุดทดสอบ (เส้นประสีแดง). แกนต์ แสดงค่าผิดพลาด. แกนนอน แสดงค่าของ  $\log \lambda$ .



รูปที่ 3.11: วิธีกรอสวัลเดชั้น 5 พับ. ข้อมูลทั้งหมดจะถูกแบ่งออกเป็น 5 ส่วน และวิธีกรอสวัลเดชั้นจะทำทั้งหมด 5 ครั้ง โดยแผนภาพแสดงในเห็นว่า การทำการรังแรกใช้ข้อมูล 4 ส่วนแรกสำหรับการฝึก และส่วนสุดท้ายสำหรับการตรวจสอบ. ส่วนที่ใช้สำหรับการตรวจสอบ แสดงเป็นสีเข้ม. ครั้งที่สอง สาม สี่ และห้าก็ทำเข่นเดิม เพียงแต่เปลี่ยนส่วนที่ทำการตรวจสอบ.

**กรอสวัลเดชั้น.** ถ้าข้อมูลมีจำนวนมาก การแบ่งบางส่วนของข้อมูลมาเป็นชุดตรวจสอบนั้น ไม่ได้ดูว่ามีปัญหาอะไร แต่หากข้อมูลมีปริมาณจำกัด ควรจะจัดการสถานการณ์อย่างไร เมื่อการฝึกแบบจำลองให้ต้องการข้อมูลจำนวนมาก แต่คุณภาพของการตรวจสอบเลือกความซับซ้อน และการทดสอบ ก็ต้องการข้อมูลจำนวนมากเช่นกัน. การแบ่งส่วนข้อมูลที่มีปริมาณน้อยอยู่แล้ว ยิ่งจะทำให้แต่ละส่วนมีปริมาณน้อยลงไปอีก. วิธีหนึ่งที่ออกแบบมาเพื่อ弥补ปัญหานี้ คือ การทำกรอสวัลเดชั้น (cross-validation). แนวคิดคือ การสุ่มและใช้ผลแล้ว โดยทำการฝึกแบบจำลองและการตรวจสอบหลาย ๆ ครั้ง แต่ละครั้ง แบ่งข้อมูลต่าง ๆ กันไปแล้วเอานำผลลัพธ์ที่ได้มาเฉลี่ยกัน เพื่อสรุปหาแบบจำลองที่มีคุณสมบัติความทวีไปดีที่สุด. การดำเนินการ จะแบ่งข้อมูลออกเป็น  $K$  ส่วน แต่ละครั้งจะเลือกส่วนหนึ่งมาเป็นชุดตรวจสอบ และใช้ส่วนที่เหลือ ( $K - 1$  ส่วน) สำหรับฝึกแบบจำลอง. เนื่องจาก การแบ่งข้อมูลเป็น  $K$  ส่วน วิธีนี้ มักถูกเรียกว่า วิธีกรอสวัลเดชั้น  $K$  พับ (K-fold cross-validation).

วิธีกรอสวัลเดชั้น  $K$  พับทำการฝึกและตรวจสอบ  $K$  ครั้ง ที่แต่ละครั้งจะเลือกส่วนที่ทำการตรวจสอบแตกต่างกัน. เมื่อทำจนครบทุกส่วนแล้ว จึงนำผลประเมินจากแต่ละครั้ง รวม  $K$  ค่า มาหาค่าเฉลี่ย เป็นค่าประเมินกรอสวัลเดชั้นของแบบจำลอง (cross-validation evaluation). ค่าประเมินกรอสวัลเดชั้นนี้ สามารถใช้เปรียบเทียบกับแบบจำลองอื่น (หรือแบบจำลองเดียวกันแต่ความซับซ้อนอื่น) เพื่อหาแบบจำลอง(หรือความซับซ้อน)ที่ดีที่สุด.

รูป 3.11 แสดงแผนภาพการแบ่งข้อมูลสำหรับวิธีกรอสวัลเดชั้น 5 พับ ( $K = 5$ ) และการจัดสรรข้อมูลสำหรับการฝึก และการตรวจสอบในแต่ละครั้ง. การฝึกและตรวจสอบแต่ละครั้ง จะเรียกเป็นวาริเดชั้นรัน (validation run). ในภาพแสดง 5 วาริเดชั้นรัน ที่รันแรก (Run 1) ฝึกแบบจำลองด้วยข้อมูล 4 ส่วนแรก และ

นำแบบจำลองที่ฝึกแล้ว ไปตรวจสอบกับข้อมูลส่วนหลังสุด (แรงงานสีเข้มในรูป). รันที่สอง ฝึกแบบจำลองด้วยข้อมูลส่วนอื่นยกเว้นส่วนที่ 4 (แรงงาน) แล้วตรวจสอบกับส่วนที่ 4 ที่กันออกไว้ ทำเช่นนี้จนครบ 5 รัน และนำผลที่ได้มาเฉลี่ย.

ด้วยวิธีนี้ แต่ละรันจะฝึกแบบจำลองด้วยข้อมูลขนาด  $K - 1$  ส่วนของที่มีอยู่ทั้งหมด  $K$  ส่วน และผลค่าผิดพลาดจากการตรวจสอบ เป็นค่าเฉลี่ยของค่าผิดพลาดที่ได้จากทุกส่วนของข้อมูล. วิธีครอสвалиเดชั้นนี้ ทำให้เสมออนว่ามีข้อมูลมากขึ้น ทั้งการฝึกและการตรวจสอบ.

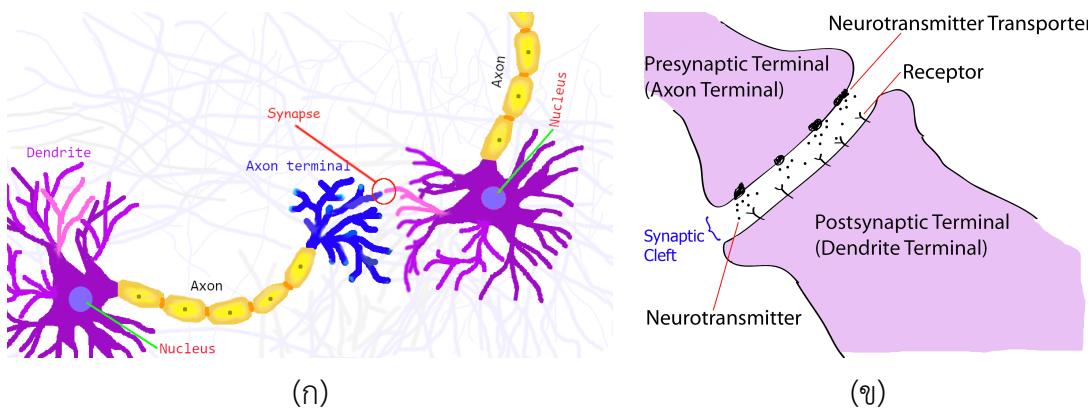
แม้วิธีครอสвалиเดชั้น จะบรรเทาปัญหาของขนาดข้อมูลที่จำกัด และเป็นวิธีที่ใช้ข้อมูลได้อย่างคุ้มค่า แต่ข้อเสียของวิธีครอสвалиเดชั้นคือ การที่ต้องทำการรันทั้งหมด  $K$  ครั้ง. ถ้าเลือกค่า  $K$  ใหญ่ ก็จะเบริ่ยบเสมือนได้ใช้ข้อมูลปริมาณมากในการฝึก แต่ข้อเสีย คือเท่ากับเพิ่มจำนวนรันด้วย โดยเฉพาะ ถ้าการรันแต่ละครั้งใช้เวลา many. กล่าวอีกนัยหนึ่ง วิธีครอสвалиเดชั้น ใช้การคำนวนที่เพิ่มขึ้น เพื่อบรรเทาปัญหาข้อมูลปริมาณน้อย. ดังนั้น หากถ้าการรันแต่ละครั้งใช้การคำนวนมากอยู่แล้ว แนวทางของวิธีครอสвалиเดชั้นอาจจะไม่เหมาะสม.

### เกร็ดความรู้เซลล์ประสาท (เรียบเรียงจาก [82] และ [217])

สมองมนุษย์ประกอบด้วยเซลล์ชนิดต่าง ๆ มากมาย เช่น เส้นเลือด เซลล์เกลีย เซลล์ประสาท. เส้นเลือดท่าน้ำที่รับส่งอากาศ น้ำ อาหาร. เซลล์เกลีย(neuroglia) ทำหน้าที่สนับสนุนต่าง ๆ รวมถึงการรักษาภาวะhomeostasis เพื่อให้ภายในสมองมี สภาวะที่เหมาะสม เช่น การควบคุมระดับความเข้มข้นของโซเดียมและแคลเซียมในอ่อน. เซลล์ประสาททำหน้าที่หลักของสมอง ได้แก่ การควบคุมระบบการทำงานต่าง ๆ ในร่างกายให้เป็นปกติ รวมไปถึง การให้ความสามารถในการจำ การเรียนรู้ การคิด การรับรู้ และ การตอบสนอง.

สมองมนุษย์มีเซลล์ประสาทอยู่ประมาณแสนล้านเซลล์ เซลล์ประสาทเองก็มีอยู่หลายประเภท แต่โครงสร้างพื้นฐานมีลักษณะ คล้าย ๆ กัน. นั่นคือ เซลล์ประสาthatแต่ละเซลล์ มีประสาทเพื่อรับสัญญาณเข้าสู่เซลล์ เรียกว่า dendrite. สัญญาณ ต่าง ๆ ทั้งสัญญาณกระตุ้นและสัญญาณยับยั้งที่เข้าสู่เซลล์ จะถูกนำมารวมกันที่นิวเคลียส และผู้รวมของสัญญาณที่รับเข้ามา จะ เป็นตัวตัดสินว่า เซลล์ประสาทนั้นจะอยู่ในสถานะถูกกระตุ้นหรือไม่. ถ้าเซลล์ประสาทอยู่ในสถานะถูกกระตุ้น มันจะส่งสัญญาณออก ไปให้กับเซลล์ประสาทอื่น ๆ ที่รับสัญญาณจากมัน โดยส่งออกผ่านประสาทนำออกสัญญาณ เรียกว่า axon. จุดต่อ ระหว่าง例外ของเซลล์ประสาทตัวหนึ่งกับเด่นได้รับของเซลล์ประสาทอีกเซลล์หนึ่ง เป็นจุดประสาทที่เรียกว่า ไซแนปส์ (synapse). แนวคิดพื้นฐานนี้เองที่ โรเชนแบลท (หัวข้อ 3.3) นำไปสร้างแบบจำลองเพอร์เซปตรอน (ดูรูป 3.13 และ 3.12 ประกอบ) เมื่อเบริ่ยบเทียบเพอร์เซปตรอน (รูป 3.14) กับเซลล์ประสาท ผลคูณของอินพุตกับค่าน้ำหนักของเพอร์เซปตรอน (เช่น  $x_1 w_1$  และ  $x_2 w_2$ ) เทียบได้กับ ความแรงของสัญญาณประสาท แต่ละสัญญาณที่รับเข้ามาผ่านไซแนปส์ แล้วเดินทางเข้าสู่นิวเคลียสของเซลล์ ประสาท เพื่อไปรวมกับความแรงของสัญญาณประสาทที่รับเข้ามาผ่านไซแนปส์ดูอีก. ความแรงของสัญญาณประสาท แต่ละ สัญญาณที่รับเข้ามาผ่านไซแนปส์ จะขึ้นอยู่กับ สัญญาณที่ส่งมา (เบริ่ยบเทียบกับ  $x_i$ ) และ ความแข็งแรงในการเชื่อมต่อสัญญาณของ ไซแนปส์ (เบริ่ยบเทียบกับ  $w_i$ ).

โดยเฉลี่ยแล้ว เซลล์ประสาทแต่ละเซลล์จะมีไซแนปส์ประมาณห้าพันจุด ซึ่งนั่นคือเมื่อร่วมแล้ว ในสมองมนุษย์หนึ่งคนจะมีการ เชื่อมต่อประสาทอยู่ราว ๆ ห้าร้อยล้านล้านไซแนปส์. การรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทมีหลายกลไก เช่น กลไก ทางเคมี (ผ่านสารสื่อประสาท) กลไกทางไฟฟ้า และ กลไกเชิงภูมิคุ้มกัน. แต่กลไกหลักของการส่งสัญญาณประสาทคือกลไกทางเคมี ซึ่งคือการรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทโดยดำเนินการผ่านสารสื่อประสาท (neurotransmitter). เซลล์ประสาทที่

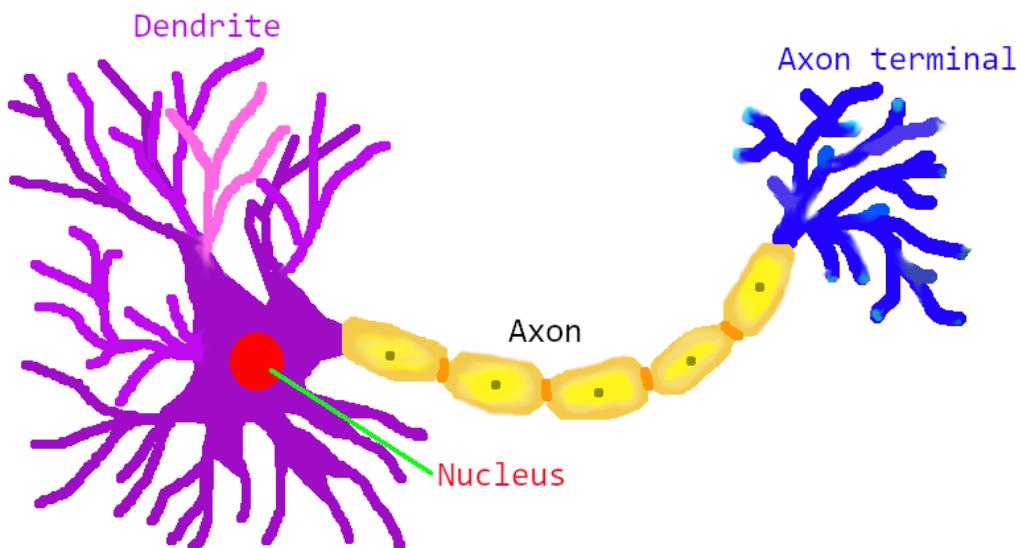


รูปที่ 3.12: ภาพแสดงเซลล์ประสาทเชื่อมต่อสัญญาณกันผ่านไซแนปซ์ โดยสัญญาณที่สื่อสารกันนั้นทำโดยผ่านกลไกของสารสื่อประสาท. ภาพ ก แสดงเซลล์ทางข่ายมีส่วนสัญญาณผ่านไซแนปซ์ ไปสู่เซลล์ทางขามีอ. ภาพ ข แสดงภาพขยายส่วนของไซแนปซ์ ซึ่งส่วนปลายของแอกซอน (presynaptic terminal) จะส่งสารสื่อประสาಥอกมา และส่วนปลายของเดนไทร์ต (postsynaptic terminal) จะรับสารสื่อประสาท.

ส่งสัญญาณจะปล่อยสารสื่อประสาಥอกมา ผ่านโปรตีนที่ทำหน้าที่ส่งสารสื่อประสาท. โปรตีนส่งสาร เรียกว่า ทรานสปอร์เตอร์ (neurotransmitter transporter). และเซลล์ประสาทที่รับสัญญาณ จะรับสารสื่อประสาทเหล่านั้น ด้วยโปรตีนที่ทำหน้าที่รับสารสื่อประสาท. โปรตีนรับสาร เรียกว่า รีเซปเตอร์ (receptor).

เมื่อรีเซปเตอร์ได้รับสารสื่อประสาท นั่นคือ โครงสร้างของสารสื่อประสาทจับกับโครงสร้างของรีเซปเตอร์ แล้วทำให้กลไกของรีเซปเตอร์เปิดทำงาน โมเลกุลที่จับกับรีเซปเตอร์ จะเรียกว่า ลิกเอนด์ (ligand). กลไกของการจับตัวระหว่างรีเซปเตอร์กับลิกเอนด์นี้ จะเป็นกลไกในลักษณะแม่กุญแจกับลูกกุญแจ (lock and key). นั่นคือ โครงสร้างของรีเซปเตอร์แต่ละชนิดจะจับตัวได้เฉพาะกับลิกเอนด์ที่มีโครงสร้างที่เข้ากันได้เท่านั้น เช่น สารสื่อประสาทอาเซ็ติล cholines (acetylcholine) ซึ่งเป็นสารสื่อประสาทที่เซลล์ประสาทใช้ติดต่อกระตุ้นเซลล์กล้ามเนื้อ จะจับกับรีเซปเตอร์สำหรับอาเซ็ติล cholines ได้เท่านั้น และ รีเซปเตอร์สำหรับสารสื่อประสาทตัวอื่น ก็ไม่อาจจับกับอาเซ็ติล cholines ได้เช่นกัน. การเข้าใจกลไกการทำงานลักษณะนี้ ช่วยให้เภสัชศาสตร์สามารถออกแบบตัวยาที่เฉพาะเจาะจงกับสารสื่อประสาทเฉพาะตัวได้ เช่น ยาต้านอาการเครียด ฟลูอูเซตทีน (Fluoxetine) ที่เฉพาะเจาะจงกับสารสื่อประสาทเซอโรโทนิน (Serotonin).

หมายเหตุ เซลล์ประสาทแต่ละชนิด จะมีลักษณะเฉพาะตัวต่างกันและ จะทำงานกับสารสื่อประสาทเฉพาะชนิด เช่น เซลล์ประสาทเซอโรโทนิน (serotonin neurons) ที่อยู่บริเวณดอร์ซอลาрапีนเคลียส (dorsal raphe nucleus) ของก้านสมอง จะทำงานกับสารสื่อประสาทเซอโรโทนิน[119], เซลล์ประสาทชีเอ1พีรามิดอล (CA1 pyramidal neurons) ที่อยู่บริเวณชีเอ1 (CA1) ของฮิปโปแคมปัส จะทำงานกับสารสื่อประสาทกลูตาเมท(Glutamate)[138], เซลล์ประสาทมิดเบรนโดพามีโนจิก (midbrain dopaminergic neurons) ที่อยู่หลาย ๆ บริเวณรวมถึง พื้นที่เวนทรอլเทกเมนทอล (ventral tegmental area) ในสมองส่วนกลาง จะทำงานกับสารสื่อประสาทโดพามีน (Dopamine)[138]. เซลล์ประสาทบางชนิดทำงานกับสารสื่อประสาทมากกว่าหนึ่งชนิด เช่น เซลล์ประสาทหนามกลาง (medium spiny neurons) ที่อยู่บริเวณ bazolateral ganglion เส่งสัญญาณออกผ่านกาบา (GABA) แต่สามารถรับสัญญาณผ่านสารสื่อประสาทหลายชนิดรวมถึงกลูตาเมทและโดพามีน.



รูปที่ 3.13: รูปแสดงโครงสร้างของเซลล์ประสาททั่ว ๆ ไป เดนไดร็ต ทำหน้าที่สืบสานอินพุตของเซลล์ และ ออกชอน ทำหน้าที่ ส่งสัญญาณกระตุ้นจากอินพุตรวมกันมากพอ เซลล์จะเข้าสู่สถานะถูกกระตุ้น และส่งการกระตุ้นออก ต่อไป ผ่านออกชอน.

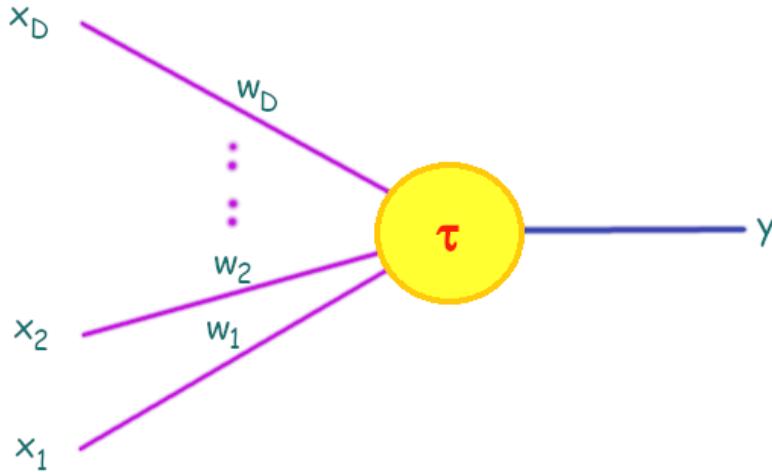
### 3.3 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial Neural Network) เป็นแบบจำลองทำงาน ที่ใช้ลักษณะการคำนวณง่าย ๆ คล้าย ๆ กัน จำนวนมาก ที่เมื่อนำมารวมกันแล้ว ให้ผลโดยรวม เป็นแบบจำลองทำงานที่มีความสามารถสูง.

โครงข่ายประสาทเทียม มีอยู่หลายชนิด หนึ่งในชนิดที่สำคัญและได้รับความนิยมอย่างมาก คือ เพอร์เซปตรอนหลายชั้น. เพอร์เซปตรอนหลายชั้น (multi-layer perceptron ค่าย่อ MLP) ที่รวมการคำนวณของ หน่วยคำนวณย่อย ที่เรียกว่า เพอร์เซปตรอน หลาย ๆ หน่วย เข้าด้วยกัน ในลักษณะเป็นชั้น ๆ.

หน่วยคำนวณย่อย เพอร์เซปตรอน (perceptron) ถูกพัฒนาโดยการเลียนแบบเซลล์ประสาทของสิ่งมี ชีวิต. เซลล์ประสาทของสิ่งมีชีวิตมีอยู่หลายชนิด แต่มีลักษณะทั่ว ๆ ไป ดังแสดงในรูป 3.13. แต่ละเซลล์รับ สัญญาณกระตุ้นจากเซลล์อื่น ๆ ผ่านเดนไดร็ต (dendrite) เมื่อผลรวมของสัญญาณกระตุ้นมากพอเซลล์จะเข้า สู่สถานะถูกกระตุ้น และส่งสัญญาณออกผ่านออกชอน (axon) ไปให้เซลล์อื่น ๆ ต่อไป. ความแรงของสัญญาณ กระตุ้นที่รับมาจากแต่ละเซลล์ก็ต่าง ๆ กันไป ขึ้นกับการเชื่อมต่อ ซึ่งความแข็งแรงของการเชื่อมต่อ ก็มีการปรับเปลี่ยนตามการใช้งาน.

แฟรงค์ โรเซนแบล็ท (Frank Rosenblatt) ออกแบบ สร้าง และได้สาธิตการทำงานของ เพอร์เซปตรอน ที่สร้างด้วยวงจรไฟฟ้า เพื่อจำลองการทำงานของเซลล์ประสาท ในปี 1957. รูป 3.14 แสดงโครงสร้างแนวคิด ของเพอร์เซปตรอน.



**รูปที่ 3.14:** แผนผังแสดงโครงสร้างของเพอร์เซปตรอน ที่เอาต์พุต  $y$  แสดงสถานะของหน่วยประสาทเทียม โดย หน่วยประสาทเทียม จะอยู่ในสถานะถูกกระตุ้น เมื่อผลรวมสัญญาณกระตุ้น ซึ่งคือ  $w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_D \cdot x_D$  มีค่ามากพอ (มากถึงหรือเกิน ค่าระดับกระตุ้น  $\tau$ ). ค่า  $x_1, \dots, x_D$  เป็นอินพุตต่าง ๆ ของเพอร์เซปตรอน. ค่า  $w_1, \dots, w_D$  เป็นค่าน้ำหนัก แสดงความแข็งแรง ของการเชื่อมต่อกับแต่ละอินพุต.

การคำนวณของเพอร์เซปตรอน ดำเนินการ โดย การนำอินพุตแต่ละตัว ไปคูณกับค่าน้ำหนักของอินพุต นั้น ๆ และนำค่าผลคูณทั้งหมดมาบวกกัน. แล้วหากผลบวกมีค่ามากพอ นั่นคือ มีค่าเท่ากับหรือมากกว่าค่า ระดับกระตุ้น เพอร์เซปตรอนจะอยู่ในสถานะถูกกระตุ้น (ให้อาต์พุตเป็น 1) แต่หากผลบวกมีค่าน้อยกว่าระดับ กระตุ้น เพอร์เซปตรอนจะอยู่ในสถานะไม่ถูกกระตุ้น (ให้อาต์พุตเป็น 0). ดังนั้น เอาต์พุตของเพอร์เซปตรอน สามารถเขียนดังสมการ 3.11.

$$y = \begin{cases} 0 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D < \tau, \\ 1 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D \geq \tau. \end{cases} \quad (3.11)$$

เมื่อ  $w_1, \dots, w_D$  เป็นค่าน้ำหนัก (weights) ของอินพุต  $x_1, \dots, x_D$  ตามลำดับ และ  $\tau$  คือ ค่าระดับกระตุ้น โดยผลลัพธ์  $y = 1$  แทนสถานะการถูกกระตุ้น และ  $y = 0$  แทนสถานะไม่ถูกกระตุ้น. บางครั้ง อินพุต  $x_1, \dots, x_D$  อาจถูกมองรวม คือมองเป็น อินพุต  $\mathbf{x} = [x_1, \dots, x_D]^T$  โดย แต่ละตัว หรือแต่ละส่วนประกอบ  $x_i$  จะเรียกเป็น มิติ (dimension) หรือคุณลักษณะ (feature) ของอินพุต.

เพื่อความสะดวก นิยาม ไบอัส (bias) เป็น  $b = -\tau$  และเพอร์เซปตรอนสามารถเขียนได้เป็น

$$y = \begin{cases} 0 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b < 0, \\ 1 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b \geq 0. \end{cases} \quad (3.12)$$

และเมื่อมองในมุมที่กว้างขึ้น สมการ 3.12 สามารถเขียนได้เป็น

$$y = h \left( \sum_{i=1}^D w_i x_i + b \right) \quad (3.13)$$

เมื่อ  $h$  เป็นฟังก์ชันกระตุ้น (activation function) ซึ่งอาจนิยามเป็น ฟังก์ชันจำกัดแข็ง (hard limit function) หรือบางครั้งอาจเรียก ฟังก์ชันขั้นบันไดหนึ่งหน่วย unit step function) ได้แก่

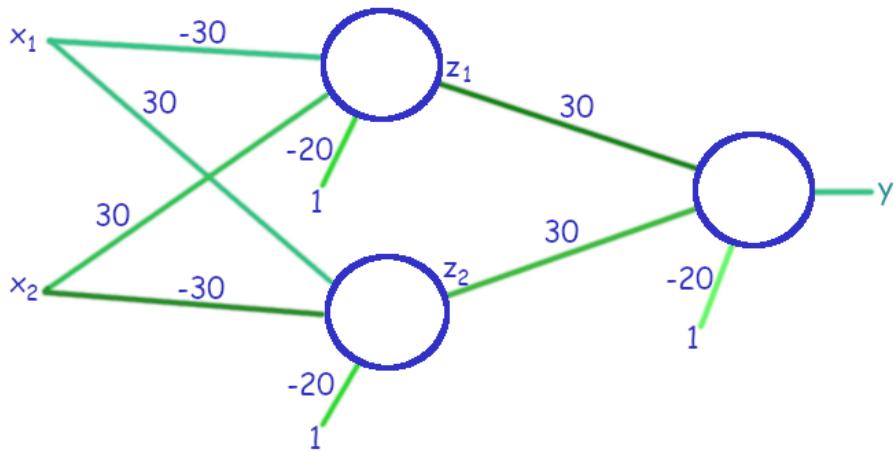
$$h(a) = \begin{cases} 0 & \text{เมื่อ } a < 0, \\ 1 & \text{เมื่อ } a \geq 0. \end{cases} \quad (3.14)$$

ในตอนนั้น งานของโรเซนแบลททำให้วิเคราะห์โดยเฉพาะอย่างยิ่งการปัญญาประดิษฐ์ตื่นเต้นมาก ที่แนวทางนี้ อาจเป็นโอกาสที่มนุษย์จะสามารถสร้างเครื่องจักร ที่สามารถเลียนแบบการทำงานของสมองมนุษย์ได้ และเป้าหมายของปัญญาประดิษฐ์ และความผันของวิทยาการคอมพิวเตอร์อาจจะสำเร็จได้ เกิดการคาดการณ์ถึงศักยภาพ ความสามารถต่าง ๆ ที่เครื่องคอมพิวเตอร์จะสามารถทำได้. แต่ความผันและความหวังกีล่อมสลายไป หลังจาก มาร์вин มินสกี้ (Marvin Minsky) และ เซมวอร์ ปาเปิต (Seymour Papert) ได้ร่วมกันเขียนหนังสือเพอร์เซปตรอนส์[128] ที่วิเคราะห์โครงสร้างและการทำงานของเพอร์เซปตรอน. ประเด็นสำคัญของหนังสือ คือ มินสกี้และปาเปิตถก และวิจารณ์ว่า เพอร์เซปตรอนนั้นสามารถทำได้แต่งานง่าย ๆ เช่น หากเป็นงานการจำแนกประเภท ก็สามารถทำงานได้กับปัญหาที่สามารถแบ่งได้ด้วยเส้นแบ่งตัดสินใจเชิงเส้นเท่านั้น ไม่สามารถทำงานที่ซับซ้อนกว่านั้นได้. พร้อมทั้งยังยกตัวอย่าง การทำงานของตระกูล เอ็กซ์-or (XOR หรือ exclusive OR) ที่เพอร์เซปตรอนไม่สามารถเลียนแบบได้. ตาราง 3.2 แสดงพฤติกรรมของตระกูลเอ็กซ์-or.

**ตารางที่ 3.2:** ตระกูลเอ็กซ์-or ที่อ้างว่าเพอร์เซปตรอน ไม่สามารถทำงานได้. ตระกูลเอ็กซ์-or เป็นตระกูลที่ผลลัพธ์ไม่สามารถใช้เส้นแบ่งตัดสินใจเชิงเส้น มาตัดสินได้.

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

ผลจากคำวิจารณ์ของมินสกี้และปาเปิต นอกจากจะจะทำให้เพอร์เซปตรอนเสื่อมความสนใจแล้ว ยังทำให้เทคนิคทางด้านโครงข่ายประสาทเทียมทั้งหมด รวมไปถึงสาขาวิชาปัญญาประดิษฐ์ เสียความนิยมและเสื่อม



**รูปที่ 3.15:** ตัวอย่างโครงข่ายเพอร์เซปตรอนสองชั้น ที่ทำตระกูลเอ็กซ์อร์. เพอร์เซปตรอนสามหน่วยต่อในลักษณะชั้นคำนวณ โดยเพอร์เซปตรอนสองหน่วย รับอินพุต  $x_1$  และ  $x_2$  และคำนวณเอาต์พุตออกมา เป็น  $z_1$  (หน่วยบน) และ  $z_2$  (หน่วยล่าง). เอาต์พุต  $z_1$  และ  $z_2$  จากสองหน่วย ถูกใช้เป็นอินพุตของเพอร์เซปตรอนตัวสุดท้าย (ตัวขวาสุด) และเอาต์พุตของเพอร์เซปตรอนตัวสุดท้าย  $y$  ถูกใช้เป็นเอาต์พุตของโครงข่าย. คำน้ำหนักระหว่างการเชื่อมต่อ เทียบกับเรณสัมแสดงการเชื่อมต่อ. ในภาพ ค่าใบอัล สูญเสดงด้วยค่าน้ำหนักของอินพุตค่าคงที่หนึ่ง.

ความสนใจไปในช่วงหลายปีต่อจากนั้น จนเรียกว่า ช่วงเวลาหนึ่งเป็น หน้าหนาวของปัญญาประดิษฐ์ (AI Winter). นอกจากคำวิจารณ์ของมินสกี้และปาเปิต เหตุผลอื่น ๆ ที่มีส่วนทำให้โครงข่ายประสาทเทียมเสียความนิยมไป ได้แก่ (1) การกำหนดโครงสร้างการเชื่อมต่อของโครงข่ายประสาทเทียม ที่ตอนนั้นยังใหม่มาก และยังไม่มีแนวทางที่ชัดเจน<sup>1</sup> และ (2) การใช้งานโครงข่ายประสาทเทียม จะต้องเลือกค่าน้ำหนัก และค่าใบอัลให้ถูกต้อง ซึ่ง ณ ช่วงเวลาหนึ่ง ยังไม่มีวิธีการที่มีประสิทธิภาพในการเลือกค่าน้ำหนัก และค่าใบอัล. (ยังไม่มีแม้แต่ หลักในการเลือกจำนวนเพอร์เซปตรอนที่เหมาะสมสมกับการใช้งาน.)

จนกระทั่งราหุลศวรรษให้หลัง งานของเวอร์โบส[214] และโดยเฉพาะอย่างยิ่งงานของกลุ่มของรูเมลาร์ต ยินตัน และวิลเลียม[170] ที่ออกแบบให้เห็นถึงประสิทธิผลของโครงข่ายประสาทเทียม พร้อมนำเสนอวิธีที่มีประสิทธิภาพ ในการหาค่าน้ำหนัก และค่าใบอัล ซึ่งงานเหล่านี้ ได้ช่วยพัฒนาความนิยมของโครงข่ายประสาทเทียมกลับมาใหม่.

เมื่อจะกล่าวไปแล้ว สิ่งที่มินสกี้กับปาเปิตวิจารณ์ว่า เพอร์เซปตรอนทำงานได้แต่ง่าย ฯ ก็ไม่ได้ผิด ซ้ำทั้งหมด. เพียงแต่ว่า มินสกี้กับปาเปิตสรุปความเห็น จากการวิเคราะห์การทำงานของเพอร์เซปตรอนที่เป็นโครงข่ายชั้นเดียว. การทำงานของโครงข่ายสมองมนุษย์ไม่ได้เป็นชั้นเดียว ในลักษณะเดียวกัน โครงข่ายประสาทเทียมที่มีประสิทธิผล จะต้องมีโครงสร้างมากกว่าหนึ่งชั้น. นั่นก็คือ ที่มาของพัฒนาการต่อมา ได้แก่ เพอร์เซปตรอนหลายชั้น ซึ่งชื่อดีเนินย้ำ ถึงการใช้โครงข่ายต่อเชื่อมกันในลักษณะหลายชั้น ของหน่วยคำนวณ

<sup>1</sup> สถาปัตยกรรมที่ต้องหน่วยคำนวณอยู่เป็นลักษณะของชั้นคำนวณ ไม่ได้มาโดยธรรมชาติ แต่มาจากการออกแบบภายหลัง.

แบบเพอร์เซปตرون.

รูป 3.15 แสดงเพอร์เซปตرونสองชั้น (two-layer perceptron) ที่สามารถเลียนแบบการทำงานของตระกูลเอ็กซ์บอร์ได้ โดยใช้เพอร์เซปตرونสามตัว ต่อ กันในลักษณะสองชั้นคำนวน. เพอร์เซปตرونแต่ละตัวอาจถูกเรียกว่า **โนนด** (node) หรือ **หน่วยคำนวน** (unit). ผลลัพธ์จากโนนดในชั้นคำนวนแรก และส่งไปเป็นอินพุตให้กับโนนดในชั้นคำนวนที่สอง. การจัดโครงสร้างเป็นลักษณะชั้นคำนวน (layer) แบบนี้ ช่วยให้โครงข่ายประสาทเทียมสามารถทำงานที่ซับซ้อนได้. เอกตพุตของเพอร์เซปตرونสองโนนดในชั้นแรก ซึ่งคือ  $z_1 = h(-30x_1 + 30x_2 - 20)$  และ  $z_2 = h(30x_1 - 30x_2 - 20)$  ทำหน้าที่เป็นอินพุตของเพอร์เซปต่อนตัวที่อยู่ชั้นที่สอง. เอกตพุตของเพอร์เซปต่อนชั้นที่สอง ซึ่งเป็นชั้นสุดท้าย ที่จะใช้เป็นเอกตพุตของทั้งโครงข่าย คำนวนด้วย  $y = h(30z_1 + 30z_2 - 20)$ .

ตาราง 3.3 แจกแจงการทำงาน โดย  $a_1^{(1)}$  และ  $a_2^{(1)}$  เป็นตัวกระตุน (ผลกระทบของสัญญาณกระตุน) ของโนนดตัวบน และของตัวล่างในชั้นที่หนึ่ง ตามลำดับ และ  $a^{(2)}$  เป็นของโนนดในชั้นที่สอง. สังเกตว่า ค่าน้ำหนักและไบอส สามารถเปลี่ยนไปใช้ค่าอื่นได้ โดยที่การทำงานยังคงเดิมได้ เช่น อาจใช้ค่า  $20, -20, -10$  แทน  $30, -30, -20$  ในรูป 3.15 ได้ โดยพฤติกรรมการทำงานนายยังคงเดิม. นี่เป็น ลักษณะอย่างหนึ่งของโครงข่ายประสาทเทียม ที่ ค่าน้ำหนักและไบอสที่ดีที่สุด มีได้หลายชุด.

ตารางที่ 3.3: การทำงานในแต่หน่วยของเพอร์เซปตرون 2 ชั้นในรูป 3.15. แต่ละแ夸 แสดงแต่ละกรณี ตั้งแต่  $(x_1, x_2)$  เป็น  $(0, 0), (0, 1), (1, 0)$  และ  $(1, 1)$ . สมมติ แสดงค่าของตัวแปรต่าง ๆ ดังระบุที่หัวสมมติ ได้แก่ อินพุต  $x_1$  และ  $x_2$ , ค่าการกระตุน  $a_1$  และ  $a_2$ , ผลลัพธ์การกระตุน  $z_1$  และ  $z_2$  และเอกตพุต  $y$ .

$x_1$	$x_2$	$a_1^{(1)}$	$z_1$	$a_2^{(1)}$	$z_2$	$a^{(2)}$	$y$
0	0	$-30(0) + 30(0) - 20 = -20$	$h(-20) = 0$	$30(0) - 30(0) - 20 = -20$	$h(-20) = 0$	$30(0) + 30(0) - 20 = -20$	$h(-20) = 0$
0	1	$-30(0) + 30(1) - 20 = 10$	$h(10) = 1$	$30(0) - 30(1) - 20 = -50$	$h(-50) = 0$	$30(1) + 30(0) - 20 = 10$	$h(10) = 1$
1	0	$-30(1) + 30(0) - 20 = -50$	$h(-50) = 0$	$30(1) - 30(0) - 20 = 10$	$h(10) = 1$	$30(1) + 30(0) - 20 = 10$	$h(10) = 1$
1	1	$-30(1) + 30(1) - 20 = -20$	$h(-20) = 0$	$30(1) - 30(1) - 20 = -20$	$h(-20) = 0$	$30(0) + 30(0) - 20 = -20$	$h(-20) = 0$

การคำนวณของโครงข่ายประสาทเทียม แบบเพอร์เซปตรอนหลายชั้น สรุปได้ดังสมการ 3.15 ถึง 3.18.

$$z_j^{(0)} = x_j \quad \text{สำหรับ } j = 1, \dots, D; \quad (3.15)$$

$$a_j^{(q)} = \sum_i z_i^{(q-1)} \cdot w_{ji}^{(q)} + b_j^{(q)} \quad \text{สำหรับ } j = 1, \dots, M_q; \quad (3.16)$$

$$z_j^{(q)} = h(a_j^{(q)}) \quad \text{สำหรับ } j = 1, \dots, M_q; \quad (3.17)$$

$$\hat{y}_k = z_k^{(L)} \quad \text{สำหรับ } k = 1, \dots, K \quad (3.18)$$

เมื่อ  $q = 1, \dots, L$  โดย  $L$  เป็นจำนวนชั้นคำนวณ และ  $\mathbf{x} = [x_1, \dots, x_D]^T$  เป็นอินพุตของแบบจำลอง และ  $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_K]^T$  เป็นเอาต์พุตของแบบจำลอง.

ตัวแปร  $a_j^{(q)}$  คือตัวกระตุ้นของโนนดที่  $j^{th}$  ในชั้นคำนวณที่  $q^{th}$ . การบวกในสมการ 3.16 ทำสำหรับทุกโนนดในชั้นก่อนหน้า นั่นคือ  $a_j^{(q)} = b_j^{(q)} + \sum_{i=1}^{M_{q-1}} z_i^{(q-1)} \cdot w_{ji}^{(q)}$  โดยกำหนด  $M_0 = D$ . ชั้นคำนวณที่  $q^{th}$  มีโนนด จำนวน  $M_q$  โนนด. พารามิเตอร์  $w_{ji}^{(q)}$  เป็นค่าน้ำหนักสำหรับการเชื่อมต่อของโนนด  $j^{th}$  ในชั้น  $q^{th}$  กับโนนด  $i^{th}$  ในชั้นก่อนหน้า. พารามิเตอร์  $b_j^{(q)}$  เป็นค่าไบอสของโนนด  $j^{th}$  ในชั้น  $q^{th}$ . ฟังก์ชัน  $h$  เป็นฟังก์ชันกระตุ้น.

ตัวแปร  $z_j^{(q)}$  เป็นเอาต์พุตของโนนด  $j^{th}$  ในชั้น  $q^{th}$  สำหรับ  $q = 1, \dots, L$  โดยสมการ 3.15 นิยามเอาต์พุตของชั้น  $0^{th}$  เป็นอินพุตของแบบจำลอง เมื่อความกระตัดรัด.

เอาต์พุตของแบบจำลองคือ เอาต์พุตของโนนดต่าง ๆ ในชั้นสุดท้าย (ดังระบุในสมการ 3.18) ดังนั้น  $M_L = K$  โดย  $K$  เป็นจำนวนมิติของเอาต์พุตที่ต้องการ. จำนวนชั้นคำนวณ และจำนวนโนนดในแต่ละชั้น เป็นอภิมานพารามิเตอร์ ของแบบจำลองเพอร์เซปตรอนหลายชั้น.

สมการ 3.15 ถึง 3.18 แสดงการคำนวณของเพอร์เซปตรอน ในรูปตัวแปรสเกลาร์. การคำนวณของเพอร์เซปตรอน สามารถเขียนได้กระชับกว่า โดยเขียนในรูปเวกเตอร์และเมทริกซ์ ดังสมการ 3.19 และ 3.20 ซึ่ง การจัดรูปในลักษณะนี้จะเรียกว่า เวคเตอร์ไซซ์ชัน (vectorization).

$$\mathbf{a}^{(q)} = \mathbf{W}^{(q)} \cdot \mathbf{z}^{(q-1)} + \mathbf{b}^{(q)} \quad (3.19)$$

$$\mathbf{z}^{(q)} = h(\mathbf{a}^{(q)}) \quad (3.20)$$

สำหรับ  $q = 1, \dots, L$  เมื่อ  $L$  คือจำนวนชั้นคำนวณ. ตัวแปร  $\mathbf{a}^{(q)} = [a_1^{(q)}, a_2^{(q)}, \dots, a_{M_q}^{(q)}]^T$  คือตัวกระตุ้นของชั้นคำนวณ  $q^{th}$  ที่มี  $M_q$  โนนด. ตัวแปร  $\mathbf{z}^{(q)} = [z_1^{(q)}, z_2^{(q)}, \dots, z_{M_q}^{(q)}]^T$  คือผลลัพธ์การกระตุ้นในชั้นคำนวณ  $q^{th}$  โดยกำหนดให้ อินพุตเสริมจากโนนดชั้นศูนย์ นั่นคือ  $\mathbf{z}^{(0)} = \mathbf{x} = [x_1, \dots, x_D]^T$

และเอาต์พุตของระบบ  $[\hat{y}_1, \dots, \hat{y}_K]^T = \hat{\mathbf{y}} \equiv \mathbf{z}^{(L)}$ . เมทริกซ์  $\mathbf{W}^{(q)} \in \mathbb{R}^{M_q \times M_{q-1}}$  และเวกเตอร์  $\mathbf{b}^{(m)} \in \mathbb{R}^{M_q}$  คือค่าน้ำหนักและค่าไบอสของชั้นคำนวน  $q^{th}$  และ  $M_0 = D$  และ  $M_L = K$ . ฟังก์ชัน  $h$  เป็นฟังก์ชันกระตุ้น ซึ่งสำหรับฟังก์ชันกระตุ้น  $h : \mathbb{R} \mapsto \mathbb{R}$  นิยามสัญกรณ์ เช่น  $h(\mathbf{a})$  เป็นปฏิบัติการที่มีลักษณะการคำนวนเชิงตัวต่อตัว (element-wise) เมื่อใช้กับเวกเตอร์ เมทริกซ์ หรือเทนเซอร์ (ยกเว้นจะระบุเป็นอย่างอื่น). นั่นคือ เช่น  $h([a_1, \dots, a_m]^T) = [h(a_1), \dots, h(a_m)]^T$ .

สังเกตว่า การคำนวนจะดำเนินการในลักษณะคล้าย ๆ กัน เป็นชั้นคำนวน อินพุตของระบบถูกป้อนให้กับชั้นคำนวนแรก เมื่อคำนวนผลจากชั้นหนึ่งเสร็จแล้ว ผลลัพธ์จะถูกใช้ป้อนเป็นอินพุตให้กับชั้นคำนวนต่อไป และดำเนินการเช่นนี้ต่อไปจนถึงชั้นคำนวนสุดท้าย ผลลัพธ์ของชั้นคำนวนสุดท้าย จะใช้เป็นเอาต์พุตของระบบ. เนื่องจากการคำนวนมีลักษณะที่คำนวนเป็นชั้น ๆ ผ่านไปทิศทางเดียว เพอร์เซปตรอนหลายชั้น บางครั้งอาจถูกเรียกว่า โครงข่ายแพร่กระจายไปข้างหน้า (feedforward network). ชั้นคำนวนทั้งหมดที่อยู่ก่อนชั้นสุดท้าย นั่นคือ ชั้น  $q = 1, \dots, L - 1$  จะเรียกว่า ชั้นซ่อน (hidden layer) เนื่องจากเอาต์พุตของชั้นคำนวนเหล่านี้ ไม่ได้ใช้เป็นเอาต์พุตสุดท้ายของแบบจำลอง และโหนดต่าง ๆ ที่อยู่ในชั้นซ่อน จะเรียกว่า โหนดซ่อน (hidden node) หรือ หน่วยซ่อน (hidden unit). การเลือกอภิมานพารามิเตอร์ของเพอร์เซปตรอนหลายชั้น บางครั้งนิยม อ้างถึงจำนวนชั้นซ่อน เช่นในตัวอย่างรูป 3.14 อาจอ้างถึงเป็น โครงข่ายประสาทเทียม หนึ่งชั้นซ่อน ที่มีสองหน่วยซ่อน. หมายเหตุ ถึงแม่โครงข่ายประสาทเทียม จะมีหลายชนิด แต่เพอร์เซปตรอนหลายชั้น เป็นชนิดแรก และเป็นชนิดที่รู้จักกันอย่างกว้างขวาง บ่อยครั้งที่ คำว่า เพอร์เซปตรอนหลายชั้น, โครงข่ายประสาทเทียม หรือโครงข่ายแพร่กระจายไปข้างหน้า ถูกใช้แทนกัน.

ปัจจุบันโครงข่ายประสาทเทียม เป็นแบบจำลองได้ถูกนำไปใช้อย่างกว้างขวาง ในงานหลาย ๆ ลักษณะ เช่น การหาค่าลดตอน การจำแนกประเภท การประมาณฟังก์ชัน. นอกจากนั้น มีการศึกษาโครงข่ายประสาทเทียมในทางทฤษฎี และพิสูจน์ว่าโครงข่ายประสาทเทียม เป็นตัวประมาณค่าสากล (universal approximator) [47, 91] ซึ่งความหมายคือ โครงข่ายประสาทเทียม สามารถแทนฟังก์ชันใด ๆ ได้ ที่ความละเอียดตามที่ต้องการ หากมีจำนวนหน่วยคำนวนมากเพียงพอ. ตามทฤษฎีแล้ว แค่โครงข่ายประสาทเทียมแบบสองชั้น ก็เป็นตัวประมาณค่าสากลได้แล้ว แต่การใช้โครงข่ายประสาทเทียมแบบลีก มีประโยชน์หลายอย่าง ดังที่จะอภิปรายในบทที่ 5.

## การฝึกโครงข่ายประสาทเทียม

กลับมาที่เรื่องการหาค่า  $\hat{y}$  หนักและค่า  $\hat{c}$  ไปอัลกอริทึม อย่างที่เห็นจากตัวอย่าง การที่โครงข่ายประสาทเทียม จะสามารถทำงานได้ตามที่ต้องการนั้น นอกจากโครงข่ายจะต้องมีโครงสร้างที่รองรับได้แล้ว (โครงสร้างเป็นลักษณะชั้น ๆ ต่อกันที่เหมาะสม และมีจำนวนหน่วยคำนวณเพียงพอ) ค่าน้ำหนักและค่าไปอัลส์ต่าง ๆ จะต้องมีค่าที่เหมาะสมด้วย.

ค่าน้ำหนักและค่าไปอัลกอริทึมของโครงข่ายประสาทเทียม ก็คือ พารามิเตอร์ต่าง ๆ ของแบบจำลอง. ดังนั้น การหาค่าน้ำหนักและค่าไปอัลกอริทึมทำได้ในลักษณะเดียวกับ การหาค่าพารามิเตอร์ของฟังก์ชันพหุนามที่ได้อภิรายในหัวข้อ 3.1. นั่นคือ การใช้วิธีของการหาค่าดีที่สุด และเช่นเดียวกัน การหาค่าน้ำหนักและค่าไปอัลกอริทึม จะเรียกว่า การฝึก.

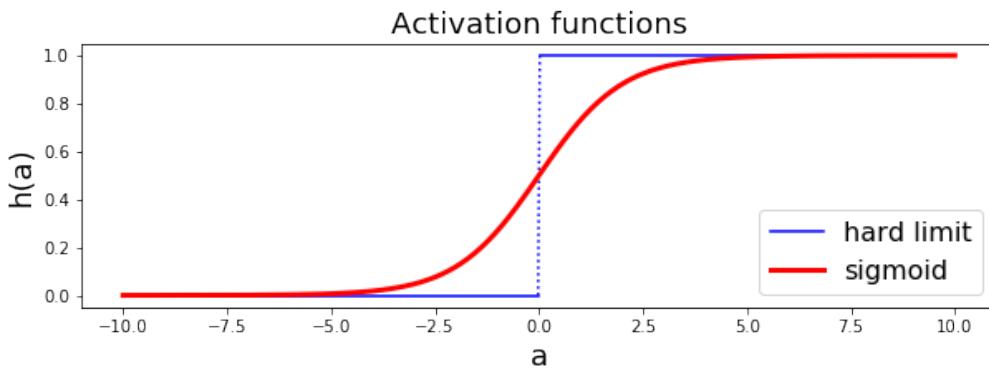
อย่างไรก็ตาม เปรียบเทียบการฝึกแบบจำลองพหุนาม กับการฝึกโครงข่ายประสาทเทียม มีประเด็นที่น่าสนใจ คือ อนุพันธ์ของฟังก์ชันจุดประสงค์ต่อค่าพารามิเตอร์. การหาอนุพันธ์ เป็นกลไกสำคัญ สำหรับวิธีของการหาค่าดีที่สุดอย่างมีประสิทธิภาพ. เพอร์เซปตรอนใช้ฟังก์ชันจำกัดแข็ง เป็นฟังก์ชันกระตุ้น แต่เนื่องจาก ฟังก์ชันจำกัดแข็ง (สมการ 3.14) เป็นฟังก์ชันที่มีค่าไม่ต่อเนื่อง (ที่  $a = 0$ ) ทำให้ไม่สามารถหาค่าอนุพันธ์ของฟังก์ชันจำกัดแข็ง และส่งผลให้การหาค่าน้ำหนักที่เหมาะสมของเพอร์เซปตรอนทำได้ยาก.

ฟังก์ชันจำกัดแข็ง แม้จะเลียนแบบการทำงานของเซลล์ประสาท แต่ลักษณะทางคณิตศาสตร์ของมัน เป็นอุปสรรคสำคัญ ต่อการฝึกโครงข่ายประสาทเทียม. การสร้างแบบจำลองคณิตศาสตร์ เพื่อทำความเข้าใจเซลล์ประสาทชีวภาพ จัดอยู่ในขอบข่ายของประสาทวิทยาเชิงคำนวณ (computational neuroscience) ซึ่งเป็นสาขาเฉพาะ และอยู่นอกเหนือจากขอบเขตของหนังสือเล่มนี้. แม้กระนั้น ฟังก์ชันจำกัดแข็ง ก็ไม่ได้อธิบายการทำงานของเซลล์ประสาทชีวภาพได้เที่ยงตรงซะที่เดียว นอกจากนั้น สิ่งที่ต้องการจริง ๆ ในมุมมองทางวิศวกรรม ก็คือเครื่องมือที่ใช้งานได้ เช่น แบบจำลองที่มีความสามารถในการทำงานที่ดี.

การฝึกโครงข่ายประสาทเทียมที่ใช้ฟังก์ชันจำกัดแข็งจะทำได้ยาก และไม่สามารถทำได้อย่างมีประสิทธิภาพ. เมื่อปัญหาอยู่ที่ฟังก์ชันจำกัดแข็ง วิธีแก้ก็แก่ที่ฟังก์ชันจำกัดแข็ง. สิ่งที่รูเมลาร์ต อินตัน และวิลเลียม[170] เสนอคือ ใช้ฟังก์ชันซิกมอยด์ (sigmoid function หรือบางครั้งเรียก logistic function หรือ logistic sigmoid function) เป็นฟังก์ชันกระตุ้น แทนฟังก์ชันจำกัดแข็ง. สมการ 3.21 แสดงการคำนวณฟังก์ชันซิกมอยด์

$$\text{sigmoid}(a) = \frac{1}{1 + \exp(-a)}. \quad (3.21)$$

ฟังก์ชันซิกมอยด์ เป็นฟังก์ชันค่าต่อเนื่อง ดังนั้นจึงสามารถหาอนุพันธ์ได้. รูป 3.16 แสดงค่าผลลัพธ์การกระ



รูปที่ 3.16: ภาพเปรียบเทียบฟังก์ชันจำกัดแข็ง (เส้นบางสีฟ้า) กับฟังก์ชันซิกมอยด์ (เส้นหนาสีแดง).

ตั้นจากฟังก์ชันจำกัดแข็งเปรียบเทียบกับฟังก์ชันซิกมอยด์.

อย่างไรก็ตามแม้ ชื่อของเพอร์เซปตรอนจะเข้มโยงกับฟังก์ชันจำกัดแข็ง แต่ในทางปฏิบัติแล้ว โดยทั่วไป ชื่อเพอร์เซปตรอนหมายชี้ คือก้าวถึงโครงข่ายประสาทเทียม ที่ใช้ฟังก์ชันซิกมอยด์ เป็นฟังก์ชันกระตุ้น. นอกจากฟังก์ชันซิกมอยด์แล้ว ฟังก์ชันไฮเปอร์บolic tangent function (hyperbolic tangent function) ซึ่งคำนวนโดย  $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$  ก็นิยมใช้เป็นฟังก์ชันกระตุ้นของโครงข่ายประสาทเทียม.

อีกประเด็น ฟังก์ชันกระตุ้น  $h$  ในสมการ 3.17 หรือ 3.20 ไม่จำเป็นต้องใช้เหมือนกันทุก ๆ ชั้นคำนวน. นั่นคือ สมการ 3.20 อาจเขียนใหม่ เพื่อเน้นประเด็นนี้ ได้เป็น  $z^{(q)} = h_q(\mathbf{a}^{(q)})$  เมื่อ  $h_q$  เป็นฟังก์ชันกระตุ้นของชั้น  $q^{th}$ . ในทางปฏิบัติแล้ว ฟังก์ชันกระตุ้นของชั้นคำนวนสุดท้าย มักจะต่างจากฟังก์ชันกระตุ้นของชั้นอื่น ๆ. ชั้นคำนวนสุดท้าย มักถูกเรียกว่า ชั้นเอ้าต์พุต (output layer) เป็นชั้นที่จะเตรียมค่าของเอ้าต์พุต ให้อยู่ในรูปแบบที่ใกล้เคียง กับรูปแบบเอ้าต์พุตที่เหมาะสมกับภารกิจมากที่สุด. ฟังก์ชันกระตุ้นของชั้นเอ้าต์พุต เรียกสั้น ๆ ว่า ฟังก์ชันกระตุ้นเอ้าต์พุต (output activation function) จะถูกเลือกใช้ตามภารกิจ และลักษณะค่าเอ้าต์พุตที่ต้องการ. ตัวอย่างเช่น การหาค่าลดถอย ซึ่งต้องการเอ้าต์พุต  $y \in \mathbb{R}$ . ฟังก์ชันกระตุ้นเอ้าต์พุต นิยมใช้ฟังก์ชันเอกลักษณ์ (identity function). นั่นคือ  $h(a) = a$ .

เมื่อเลือกใช้ฟังก์ชันกระตุ้นเป็นฟังก์ชันต่อเนื่องแล้ว การฝึกโครงข่ายประสาทเทียม จึงสามารถทำได้อย่างมีประสิทธิภาพ โดยอาศัยค่าอนุพันธ์. เช่นเดียวกับแบบจำลองทำนายอื่น ๆ การฝึกโครงข่ายประสาทเทียม คือ การหา  $\Theta^* = \arg \min_{\Theta} E$  เมื่อ  $E$  คือ ฟังก์ชันค่าผิดพลาดที่เป็นจุดประสงค์ และ  $\Theta$  คือพารามิเตอร์ของแบบจำลอง ซึ่งสำหรับโครงข่ายประสาทเทียม จากสมการ 3.16 ก็คือ  $\Theta = \{w_{ji}^{(q)}, b_j^{(q)}\}$  โดย  $q = 1, \dots, L; j = 1, \dots, M_q$ ; และ  $i = 1, \dots, M_{q-1}$ .

สำหรับงานการหาค่าลดถอย (regression) ซึ่งคือการทำนายเอ้าต์พุตที่ค่าเป็นจำนวนจริง ฟังก์ชันค่าผิด

ผลิต  $E$  สามารถนิยามด้วย ค่าเฉลี่ยค่าผิดพลาดกำลังสอง ดังแสดงในสมการ 3.22 สำหรับข้อมูลจำนวน  $N$  จุดข้อมูล.

$$E = \frac{1}{N} \sum_{n=1}^N E_n \quad (3.22)$$

$$E_n = \frac{1}{2} \|\hat{\mathbf{y}}(\mathbf{x}_n, \Theta) - \mathbf{y}(n)\|^2 \quad (3.23)$$

เมื่อ  $\mathbf{y}(n)$  คือเฉลย หรือค่าตัวแปรตามจากจุดข้อมูลที่  $n^{th}$  และ  $\hat{\mathbf{y}}(\mathbf{x}_n, \Theta)$  ที่อาจเขียนย่อเป็น  $\hat{\mathbf{y}}$  หากบริบทชัดเจน คือค่าที่แบบจำลองทำนายสำหรับจุดข้อมูลที่  $n^{th}$  เมื่อใช้ค่าพารามิเตอร์เป็น  $\Theta$ . ค่าของ  $\hat{\mathbf{y}} = \mathbf{z}^{(L)}$  คำนวณได้จากการ 3.19 และ 3.20 โดยให้  $\mathbf{z}^{(0)} = \mathbf{x}_n$  เมื่อ  $\mathbf{x}_n$  เป็นค่าตัวแปรต้นของจุดข้อมูลที่  $n^{th}$ .

การหา  $\Theta^* = \arg \min_{\Theta} E$  ที่มีประสิทธิภาพ ต้องการค่าเกรเดียนต์  $\nabla_{\Theta} E = \frac{1}{N} \sum_{n=1}^N \nabla_{\Theta} E_n$  ซึ่งสามารถพิจารณาได้จาก

$$\nabla_{\Theta} E_n = \left[ \frac{\partial E_n}{\partial w_{1,1}^{(1)}} \quad \cdots \quad \frac{\partial E_n}{\partial w_{M_1,M_0}^{(1)}} \quad \frac{\partial E_n}{\partial b_1^{(1)}} \quad \cdots \quad \frac{\partial E_n}{\partial b_{M_1}^{(1)}} \quad \cdots \quad \frac{\partial E_n}{\partial w_{M_L,M_{L-1}}^{(L)}} \quad \cdots \quad \frac{\partial E_n}{\partial b_{M_L}^{(L)}} \right]^T \quad (3.24)$$

หรืออาจเขียนย่อ ๆ เป็น  $\nabla_{\Theta} E_n = \left[ \frac{\partial E_n}{\partial w_{ji}^{(q)}}, \frac{\partial E_n}{\partial b_j^{(q)}} \right]^T$  โดย  $q = 1, \dots, L; i = 1, \dots, M_{q-1};$  และ  $j = 1, \dots, M_q$ .

พิจารณาแต่ละส่วนประกอบของเกรเดียนต์ จากกฎลูกโซ่ของการหาอนุพันธ์  $\frac{\partial E_n}{\partial w_{ji}^{(q)}} = \frac{\partial E_n}{\partial a_j^{(q)}} \cdot \frac{\partial a_j^{(q)}}{\partial w_{ji}^{(q)}}$  เพื่อความกระชับ สัญกรณ์ที่แสดง อาจลงทะเบยก  $(q)$  ในกรณีที่บีบีบทชัดเจน นั่นคือ อนุพันธ์ดังกล่าวจะเขียนเป็น

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ji}} \quad \text{และ} \quad \frac{\partial E_n}{\partial b_j} = \frac{\partial E_n}{\partial a_j} \cdot \frac{\partial a_j}{\partial b_j} \quad (3.25)$$

กำหนดให้  $\delta_j^{(q)} \equiv \frac{\partial E_n}{\partial a_j^{(q)}}$  และเมื่อบริบทชัดเจน อาจเขียนย่อเป็น

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j}. \quad (3.26)$$

จากสมการ 3.16 นั่นคือ  $a_j = \sum_i z_i^{(q-1)} \cdot w_{ji} + b_j$  ดังนั้น

$$\frac{\partial a_j}{\partial w_{ji}} = z_i^{(q-1)} \quad \text{และ} \quad \frac{\partial a_j}{\partial b_j} = 1 \quad (3.27)$$

เมื่อแทนสมการ 3.27 ลงในสมการ 3.25 จะได้

$$\frac{\partial E_n}{\partial w_{ji}^{(q)}} = \delta_j^{(q)} \cdot z_i^{(q-1)} \quad \text{และ} \quad \frac{\partial E_n}{\partial b_j^{(q)}} = \delta_j^{(q)}. \quad (3.28)$$

พิจารณา  $\delta_j^{(q)} = \frac{\partial E_n}{\partial a_j^{(q)}}$  ที่ชั้นเอาต์พุต  $q = L$  แทนค่า  $E_n$  จากสมการ 3.23 และสำหรับการหาค่า ณดถอย พังก์ชันกระตุ้นเอาต์พุตใช้พังก์ชันเอกลักษณ์ นั่นคือ  $[\hat{y}_1, \dots, \hat{y}_K]^T = [a_1^{(L)}, \dots, a_K^{(L)}]^T$  จะได้ ว่า  $\delta_k^{(L)} = \frac{\partial E_n}{\partial a_k^{(L)}} = \frac{\partial \frac{1}{2} \sum_m (\hat{y}_m - y_m)^2}{\partial a_k^{(L)}}$  และหลังจากหาอนุพันธ์จะได้

$$\delta_k^{(L)} = \hat{y}_k - y_k. \quad (3.29)$$

ที่ชั้นซ่อน  $q < L$  ตัวกระตุ้น  $a_j^{(q)}$  จะส่งผลต่อ  $E_n$  ผ่านโนนดต่าง ๆ ในชั้นถัดไป (รูป 3.17) ดังนั้น จากกฎ ลูกโซ่ของการหาอนุพันธ์

$$\delta_j^{(q)} = \frac{\partial E_n}{\partial a_j^{(q)}} = \sum_m \frac{\partial E_n}{\partial a_m^{(q+1)}} \cdot \frac{\partial a_m^{(q+1)}}{\partial a_j^{(q)}}. \quad (3.30)$$

พจน์หน้า  $\frac{\partial E_n}{\partial a_m^{(q+1)}} = \delta_m^{(q+1)}$  มาจากชั้นถัดไป. พจน์หลังคำนวณได้โดยเขียน  $a_m^{(q+1)}$  จากสมการ 3.16 และ 3.17 และหาอนุพันธ์ ซึ่งผลลัพธ์คือ  $\frac{\partial a_m^{(q+1)}}{\partial a_j^{(q)}} = w_{mj}^{(q+1)} \frac{\partial h(a_j^{(q)})}{\partial a_j^{(q)}}$ . แทนทั้งสองพจน์นี้ในสมการ 3.30 แล้วจะได้ว่า สำหรับชั้นซ่อน ( $q < L$ ) แล้ว

$$\delta_j^{(q)} = h'(a_j^{(q)}) \cdot \sum_m \delta_m^{(q+1)} \cdot w_{mj}^{(q+1)} \quad (3.31)$$

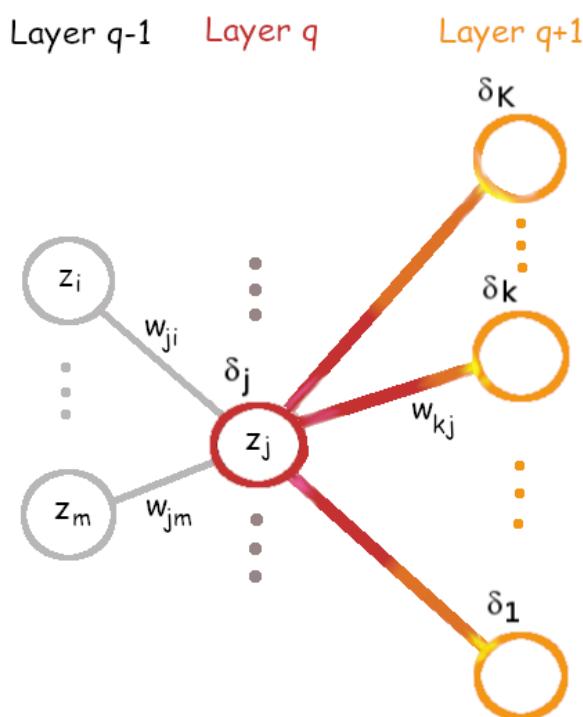
เมื่อ  $h'(a_j^{(q)}) = \frac{\partial h(a_j^{(q)})}{\partial a_j^{(q)}}$  คืออนุพันธ์ของพังก์ชันกระตุ้น ที่ค่า  $a_j^{(q)}$ . หมายเหตุ  $h'(a_j^{(q)})$  เป็นอนุพันธ์ของ พังก์ชันกระตุ้นของชั้น  $q^{th}$  และบางครั้ง อาจใช้สัญกรณ์  $h'_q(a_j^{(q)}) = \frac{\partial h_q(a_j^{(q)})}{\partial a_j^{(q)}}$  ในกรณีที่อาจสับสน.

ทั้งหมดที่อภิปรายมา สรุปได้ว่า เกรเดียนต์ของโครงข่ายประสาทเทียม  $\nabla_{\Theta} E_n$  สามารถหาได้โดยกระบวนการดังนี้

1. ทำการคำนวณไปข้างหน้า (forward propagation หรือ forward pass) โดยคำนวณสมการ 3.15 ถึง 3.18. สิ่งที่ได้คือ ค่าที่ทำนาย  $\hat{\mathbf{y}}$  กับ ค่าตัวกระตุ้นและผลลัพธ์การกระตุ้นต่าง ๆ  $\mathbf{a}^{(q)}$  และ  $\mathbf{z}^{(q)}$  สำหรับ  $q = 1, \dots, L$ .
2. คำนวณค่า  $\delta_k^{(L)}$  สำหรับทุก ๆ โนนดเอาต์พุต  $k = 1, \dots, K$  ตามสมการ 3.29. สิ่งที่ได้คือ  $\boldsymbol{\delta}^{(L)}$ .
3. คำนวณค่า  $\delta_j^{(q)}$  สำหรับทุก ๆ โนนด ในทุก ๆ ชั้นซ่อน  $j = 1, \dots, M_q; q = 1, \dots, L-1$  ตาม สมการ 3.31. สิ่งที่ได้คือ  $\boldsymbol{\delta}^{(q < L)}$ .

4. คำนวณค่าอนุพันธ์ที่ต้องการ จากสมการ 3.28. สิ่งที่ได้คือ  $\frac{\partial E_n}{\partial w_{ji}}$  และ  $\frac{\partial E_n}{\partial b_j}$  ค่าต่าง ๆ ทุกค่า ซึ่งรวมกันเป็นเกรเดียนต์  $\nabla_{\Theta} E_n$ .

เกรเดียนต์  $\nabla_{\Theta} E_n$  ที่ได้สามารถนำไปใช้กับขั้นตอนวิธีการหาค่าดีที่สุด (optimization algorithm) เช่น วิธีลิงเกรเดียนต์ที่อภิปรายในหัวข้อ 2.3 ได้. กระบวนการหาเกรเดียนต์  $\nabla_{\Theta} E_n$  ที่ได้อภิปรายมาแล้ว คือ วิธีการแพร์กระยะย้อนกลับ (error backpropagation หรือ backpropagation) คณะของ ทีรูเมลาร์ต [170] เสนอในปี ค.ศ. 1986 และเป็นการค้นพบที่พื้นฟูความสนใจในโครงข่ายประสาทเทียมกลับมา หลังจากกว่าทศวรรษของหน้าหน่าวของปัญญาประดิษฐ์.



รูปที่ 3.17: ภาพแสดงการเชื่อมต่อ และเน้นผลของโหนด  $j^{th}$  ในชั้น  $q^{th}$  ที่มีต่อค่าผิดพลาด โดยผ่านโหนดต่าง ๆ ในชั้น  $(q+1)^{th}$ . ค่า  $a_j$  (ไม่ได้แสดงในภาพ) ส่งผลผ่าน  $z_j$  ซึ่งส่งผลต่อโหนดต่าง ๆ ในชั้นถัดไปผ่านค่าน้ำหนักของการเชื่อมต่อ.

เกร็ดความรู้จิตและการเรียนรู้ (เรียบเรียงจาก [217], [1], และ [154])  
“ทุกสิ่งเริ่มที่จิต นำโดยจิต และสร้างโดยจิต” ธรรมบท[1]

จิตคือสภาวะเชิงการรับรู้และเชิงสติปัญญา ซึ่งรวมถึง สติรู้ตัว การรับรู้สัมผัส ความรู้สึก อารมณ์ การคิด การตัดสินใจ การจำ การรับประสบการณ์ การเรียนรู้ และการตอบสนองต่อสภาวะแวดล้อม. ส่วนประกอบของจิตนี้ อาจจัดออกได้เป็น 4 หมวด. หมวดหนึ่ง วิญญาณ (vijnana) ซึ่งคือสติรู้ตัว (consciousness). หมวดสอง สัญญา (samjana) ซึ่งคือการรับรู้ (perception) ผ่านสัมผัสทางการมองเห็น สัมผัสทางการได้ยิน สัมผัสทางการได้กลิ่น สัมผัสการรับรส สัมผัสทางกาย และสัมผัสที่มาจากการจิตเอง. จิตเอง

ก็มีสัมผัสเช่นกัน ดังตัวอย่างของการแขนขาลวง (phantom limbs) ที่เกิดในผู้ที่เสียแขนหรือขาไป แต่ภายในหลังยังรู้สึกคันหรือเจ็บที่แขนหรือขาที่ไม่มีอยู่ แม้ว่าสาเหตุของการนี้ยังไม่มีคำอธิบายที่ทางการแพทย์ยอมรับร่วมกันอย่างกว้างขวาง แต่นายแพทย์วิลัยานูร์ รามาชันทรัน (Vilayanur S. Ramachandran) อธิบายว่า สัมผัสที่รู้สึกนั้นมาจากการประสาทส่วนที่เคยทำงานกับแขนหรือขาส่วนนั้นขาดสัญญาณที่เคยได้รับ และอาจส่งสัญญาณอกมาทั้ง ๆ ที่ไม่ได้รับสัญญาณรับรู้จริง ๆ จากทฤษฎีนี้ นายแพทย์รามาชันทรันเสนอวิธีการบำบัดอาการแขนขาลวง โดยออกแบบกระบวนการให้ผู้มีอาการได้ฝึกประสาทรับรู้ใหม่ ซึ่งพบว่าได้ผลดีมาก หกมองจากมุมมองทางวิศวกรรม สัญญาณที่ขาดหายไปจากแขนขาที่เสียไปนั้น อาจให้ผลในลักษณะคล้ายการที่วงจรไฟฟ้ารับอินพุตมาจากขั้วปลายที่ปล่อยลายอยู่ ซึ่งขั้วปลายที่ปล่อยลายอาจรับสัญญาณรบกวนเข้ามาแทนได้ การฝึกประสาทรับรู้ใหม่ ก็อาจคล้ายการต่อขั้วปลายนั้นเข้ากับสายสัญญาณเส้นอื่น เพื่อไม่ให้มีสายลอยที่จะรับสัญญาณรบกวนเข้ามา หมวดสาม เทหนา (vedana) ซึ่งคือความรู้สึก (feeling) อารมณ์ ความชอบ ความไม่ชอบ ความวางแผน และ หมวดสี่ สังขาร (sankhara) ซึ่งคือการคิดและกระบวนการเชิงสติปัญญาอื่น ๆ (mental activity) ได้แก่ การตัดสินใจ การตอบสนองต่อสภาวะแวดล้อม การจดจำ การรับประสาร การเรียนรู้ และการเรียนรู้.

นักประสาทวิทยาด้านสติปัญญาชั้นนำ รีเบกก้า แซก [177] เชื่อว่า รูปแบบของกิจกรรมทางไฟฟ้าของเซลล์ประสาทเกี่ยวข้องโดยตรงกับจิตของเรา เพียงแต่ว่าวิทยาศาสตร์ยังไม่รู้อะไรมากเกี่ยวกับจิตและความสัมพันธ์ระหว่างรูปแบบของกิจกรรมทางไฟฟ้าและจิต

จากมุมมองของกิจกรรมทางไฟฟ้า การเรียนรู้ก็เหมือนการปรับเปลี่ยนวงจรหรือเปลี่ยนการเชื่อมต่อภายในวงจร ซึ่งส่งผลให้เกิดการปรับเปลี่ยนพฤติกรรมของกิจกรรมทางไฟฟ้า. สภาพพลาสติกของระบบประสาท (synaptic plasticity) คือ ความสามารถของระบบประสาทที่สามารถเพิ่มหรือลดความแข็งแรงของการเชื่อมต่อสัญญาณประสาทระหว่างเซลล์ได้. สภาพพลาสติกของระบบประสาทนี้เชื่อว่าเป็นคุณสมบัติที่อยู่เบื้องหลังความสามารถในการจดจำและการเรียนรู้ของสมอง. กลไกนี้เปรียบเทียบได้กับการปรับค่าน้ำหนักหรือการฝึกโครงข่ายประสาทเทียม แต่ประเด็นหนึ่งที่ต่างกันก็คือ การเปลี่ยนค่าน้ำหนักของโครงข่ายประสาทเทียมจะทำเฉพาะในขั้นตอนการฝึกโครงข่ายประสาทเทียม และค่าน้ำหนักที่ได้แล้วจะถูกตึงให้คงค่าเหล่านั้นไว้คงที่ขณะใช้งาน. แต่ระบบประสาท(ทางชีวภาพ)จะเปลี่ยนแปลงตัวเองตลอดเวลา เปเลี่ยนขณะเรียนรู้ เปเลี่ยนขณะคิด เปเลี่ยนขณะทำกิจกรรมต่าง ๆ เปเลี่ยนขณะทำงาน เปเลี่ยนขณะไม่ได้ทำงาน เปเลี่ยนขณะเล่น เปเลี่ยนขณะทำสิ่งที่มีประโยชน์ เปเลี่ยนขณะพักผ่อน เปเลี่ยนขณะนอนหลับ และที่สำคัญเปลี่ยนแม้แต่ขณะทำสิ่งที่เป็นโทษ เช่น เมื่อสิ่งที่เราคาดหวังไม่ได้ดังใจ แล้วเราไม่ชอบใจ ถ้าเราเลือกที่จะกรรสมองจะเรียนรู้การตอบสนองนั้น และ เมื่อเราทำแบบนั้นปอย ๆ เราอาจจะกลายเป็นคนที่กรรจ่าย หรือกล่าวอย่างชัดเจนก็คือ เราฝึกสมองของเราให้เก่งที่จะอยู่ในสภาวะอารมณ์กรรนั้นเอง.

กลไกเบื้องหลังการส่งสัญญาณของระบบประสาท กล่าวโดยคร่าว ๆ ก็คือ เมื่อสัญญาณจากนิวเคลียสของเซลล์ประสาทเดินทางถึงปลายแออซอนซึ่งเป็นปลายสำหรับส่งสัญญาณออก สัญญาณซึ่งถ่ายทอดมาในรูปความต่างศักดิ์จะทำให้ห้องไอออนที่ควบคุมด้วยแรงดันไฟฟ้า (voltage-gated ion channel) ของปลายแออซอนเปิดออก ทำให้แคลเซียมไอออน (calcium ion สัญลักษณ์  $Ca^{2+}$ ) ซึ่งอยู่ในของเหลวรอบ ๆ เซลล์ ไหลเข้าสู่ปลายแออซอน. แคลเซียมไอออนที่เข้าสู่ปลายแออซอนจะทำปฏิกิริยากับโปรตีนและเอนไซม์ภายในแออซอน ซึ่งส่งผลให้เกิดการปล่อยสารสื่อประสาทออกมานะ. สารสื่อประสาทที่ออกมานะมีบางโมเลกุลที่ได้จับกับรีเซปเตอร์ที่ปลายเดินไดร์ต ซึ่งปลายเดินไดร์ตคือปลายประสาทที่นำสัญญาณเข้าสู่เซลล์ประสาทด้วยที่จะรับสัญญาณ. เมื่อรีเซปเตอร์จับกับสารสื่อประสาทแล้ว การจับตัวกันทำให้รีเซปเตอร์เปลี่ยนโครงสร้าง ซึ่งจะเปิดช่องให้ออนบากไหลเข้าสู่เดินไดร์ต เมื่อไอออนบากไหลเข้าสู่เดินไดร์ตจะทำให้ความต่างศักดิ์ของเดินไดร์ตจุดนั้นเปลี่ยนไป ซึ่งความต่างศักดินี้เองเป็นสัญญาณที่จะถ่ายทอดต่อไปยังนิวเคลียสของเซลล์ประสาท.

สารสื่อประสาทเป็นกลไกหลักในการช่วยส่งสัญญาณประสาทข้ามเซลล์ประสาท แต่ตัวสารสื่อประสาทเองไม่ได้ถูกส่งเข้าไปในเดินไดร์ตของเซลล์ประสาทด้วย. มันทำหน้าที่เหมือนช่วยเปิดประตูให้ออนบากได้เข้าไปในปลายเดินไดร์ตดังที่ได้อธิบายไปข้างต้น. หลังจากสารสื่อประสาทจับกับรีเซปเตอร์ได้สักพัก มันจะหลุดออกมานะ. สารสื่อประสาททั้งที่พึงหลุดมาจากการจับกับรีเซปเตอร์และที่ยังไม่ได้จับกับรีเซปเตอร์เลยจะถูกกำจัดออกไปโดยกลไกหลาย ๆ ชนิด เช่น การทำลายทิ้ง หรือ การที่ปลายแออซอนดึงสารสื่อประสาทเหล่านี้กลับเข้าไปภายในเพื่อนำกลับไปใหม่.

การเรียนรู้หรือการปรับความแข็งแรงของการเชื่อมต่อสัญญาณระหว่างเซลล์ประสาท เกี่ยวข้องกับการปรับความสามารถใน

การรับส่งสัญญาณประสาทระหว่างเซลล์. เมื่อกล่าวถึงสัญญาณประสาทโดยละเอียดขึ้น สัญญาณประสาทจะถูกส่งในหลายรูปแบบ ขณะที่สัญญาณประสาทส่งผ่านเส้นทางจากนิวเคลียสของเซลล์ประสาทตัวหนึ่งไปสู่นิวเคลียสของเซลล์ประสาಥอีกตัวหนึ่ง มีการเปลี่ยนรูปแบบอย่างหลายครั้ง. เมื่อเซลล์ประสาทอยู่ในสถานะถูกกระตุ้น นิวเคลียสของเซลล์จะส่งสัญญาณออกมายังรูปความถี่ของพลัสร์. นั่นคือ นิวเคลียสของเซลล์ประสาทจะส่งสัญญาณในลักษณะพลัสร์ (pulse) ซึ่งมักเรียกว่าศักยะงาน (action potential) เช่น ค่าความต่างศักย์ภายนอกเซลล์ประสาทจะมีค่าประมาณ  $-70$  มิลลิโวลต์เมื่อเทียบกับจุดภายนอกเซลล์ แต่เมื่อมีศักยะงานเกิดขึ้น ค่าความต่างศักย์นี้เพิ่มขึ้นอย่างรวดเร็วจากค่าพักตัวที่ประมาณ  $-70$  มิลลิโวลต์ ไปสูงสุดที่ประมาณ  $40$  มิลลิโวลต์ หลังจากนั้นจะลดค่าลงอย่างรวดเร็วมาที่ราว ๆ  $-90$  มิลลิโวลต์ และกลับมาจบที่ค่าพักตัวที่ประมาณ  $-70$  มิลลิโวลต์ โดยที่ศักยะงานแต่ละลูก จะยานานประมาณ  $4$  มิลลิวินาที. ความถี่หรือจำนวนศักยะงานต่อวินาที จะขึ้นกับความแรงของการกระตุ้น เช่น เซลล์โอลิฟิก ตอริคอร์ทิกซ์ไฟราร์มิดอล (Olfactory Cortex pyramidal cell) ที่ทำงานเกี่ยวกับการรับรู้กลิ่น จะส่งศักยะงานออกมายังรูปความถี่ประมาณ  $0.8$  ถึง  $2.0$  ลูกต่อวินาที ขณะที่อยู่ในปกติ แต่จะส่งศักยะงานความถี่ประมาณ  $4$  ถึง  $11$  ลูกต่อวินาที ขณะที่เรากำลังดมอะไรอยู่[138]. สัญญาณในรูปความถี่นี้ได้มีการศึกษามาตั้งแต่ ค.ศ. 1926 ที่ เอเดรียนและโซตเต้มัน[2] รายงานการศึกษาการกระตุ้นและสัญญาณไฟฟ้าเซลล์ประสาท โดยใช้ตุ้มน้ำหนักดึงกล้ามเนื้อของกบและวัดสัญญาณไฟฟ้าจากเนื้อเยื่อประสาทของมัน และพบความสัมพันธ์ที่ชัดเจน ระหว่างค่าน้ำหนักที่ยืดกล้ามเนื้อออก (ตัวแทนของปริมาณความแรงของการกระตุ้น) กับความถี่ของศักยะงานที่เกิดขึ้น. ศักยะงานนี้คือสัญญาณที่ถูกส่งออกจากนิวเคลียสผ่านไปที่ปลายแอกซอน สัญญาณที่ส่งผ่านออกจากปลายแอกซอนของเซลล์ตัวส่งเข้าไปสู่ปลายเดนไดร์ตของเซลล์ตัวรับจะส่งผ่านกลไกของสารสื่อประสาท และปลายเดนไดร์ตรับสัญญาณประสาทเข้ามาในรูปดับความต่างศักย์ระหว่างภายในปลายเดนไดร์ตและภายนอก. รูปแบบที่เปลี่ยนไปของสัญญาณประสาทจากนิวเคลียสของตัวส่งไปจนถึงปลายแอกซอน (ในรูปศักยะงาน) ผ่านไซแนปซ์ (ในรูปสารสื่อประสาท) และรับเข้าสู่ปลายเดนไดร์ตไปจนถึงส่งเข้าไปสู่นิวเคลียสของตัวรับ (ในรูปดับความแรงของความต่างศักย์) รูปแบบเหล่านี้ เกี่ยวข้องสัมพันธ์กับทฤษฎีที่ใช้อิบัยกระบวนการเรียนรู้ของเซลล์ประสาท.

ทฤษฎีที่อธิบายการปรับความแข็งแรงของการเชื่อมต่อสัญญาณระหว่างเซลล์ประสาท กล่าวถึง กลไกหลัก  $2$  กลไก. นั่นคือ การเพิ่มความแข็งแรงเชิงประสาทระยะยาว (long-term synaptic potentiation คำย่อ LTP) และ การลดความแข็งแรงเชิงประสาทระยะยาว (long-term synaptic depression คำย่อ LTD). จากงานศึกษาการเชื่อมต่อของเซลล์ในอิบิโน่ปีเพลค์ โดยเฉพาะ ไซแนปซ์ที่เชื่อมต่อระหว่างเซลล์ประสาทในบริเวณซีเอ3 (CA3) ที่ส่งแอกซอน ซึ่งเรียกว่า แซฟเฟอร์คอลเลทเตอร์อล (Schaffer collaterals) ไปเชื่อมต่อกับเซลล์ประสาทในบริเวณซีเอ1 สรุปว่า เมื่อมีการกระตุ้นด้วยสัญญาณประสาทความถี่สูงผ่านไซแนปซ์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดร์ตของไซแนปซ์นั้นจะมีค่าเพิ่มขึ้นมาก และค่านั้นจะคงอยู่เป็นเวลานาน (หลายนาทีหรือหลายวัน หลังจากการกระตุ้นนั้น) โดยค่าความต่างศักย์ที่ปลายเดนไดร์ตที่เชื่อมต่อไซแนปซ์นั้นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การลดความต่างศักย์ที่ปลายเดนไดร์ตที่เชื่อมต่อไซแนปซ์อื่นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การเพิ่มความแข็งแรงเชิงประสาทระยะยาว.

ในทางกลับกัน เมื่อมีการกระตุ้นด้วยสัญญาณประสาทความถี่ต่ำผ่านไซแนปซ์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดร์ตของไซแนปซ์นั้นจะมีค่าลดลงมาก และค่านั้นก็คงอยู่เป็นเวลานาน โดยค่าความต่างศักย์ที่ปลายเดนไดร์ตที่เชื่อมต่อไซแนปซ์นั้นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การลดความแข็งแรงเชิงประสาทระยะยาว. ประเด็นสำคัญ คือ (1) มีการเปลี่ยนแปลงความแข็งแรงของไซแนปซ์ระยะยาวเกิดขึ้น (2) ผลการเปลี่ยนแปลงความแข็งแรงขึ้นกับความถี่ (3) ผลการเปลี่ยนแปลงเกิดขึ้นเฉพาะตัวของไซแนปซ์.

กลไกเบื้องหลังนั้นอธิบายว่า ไซแนปซ์ของแซฟเฟอร์คอลเลทเตอร์อล ทำงานผ่านสารสื่อประสาทกลูตามเท และปลายประสาทของเซลล์ตัวรับที่ซีเอ1 มีรีเซปเตอร์ที่ทำงานกับกลูตามทอยู่  $2$  ชนิด คือ แอมປารีเซปเตอร์ (AMPA receptor) และ เอ็นเอมดีเอรีเซปเตอร์ (NMDA receptor). เมื่อปลายประสาทของเซลล์ที่ซีเอ3 จะส่งสัญญาณกระตุ้นผ่านไซแนปซ์ มันจะปล่อยสารสื่อประสาทกลูตамเท ออกมาย. เมื่อกลูตามเทจับกับแอมປารีเซปเตอร์ แอมປารีเซปเตอร์จะเปิดออก และด้วยคุณสมบัติของแอมປารีเซปเตอร์ ไซเดียมไอโอนจะไหลเข้าสู่ปลายเดนไดร์ตตัวรับที่ซีเอ1 แคลเซียมไอโอนบางส่วนก็อาจไหลเข้าได้บ้างแต่ไม่มาก. แอมປารีเซปเตอร์ที่เปิดรับไซเดียมไอโอนแล้วซักพักก็จะปิดลง. เมื่อกลูตามเทจับกับเอนเอมดีเอรีเซปเตอร์ เอ็นเอมดีเอรีเซปเตอร์จะเปิดออกเช่นกัน แต่ เอ็นเอมดีเอรีเซปเตอร์จะมีแมกนีเซียมไอโอนปิดช่องอยู่ ทำให้ไอโอนยังไม่สามารถไหลผ่านได้. หลังจากไซเดียมไอโอนผ่านแอมປารีเซปเตอร์แล้ว สารสื่อประสาทกลูตามเทจะถูกดึงกลับเข้าไปในเซลล์ตัวรับ ทำให้ไซแนปซ์หายไป

รีเซปเตอร์เข้าสู่เซลล์ซีเอล สักพักปริมาณโซเดียมไฮอ่อนจะถูกสูบออกโดยกลไกของโซเดียม-โพแทสเซียมปั๊ม (sodium-potassium pump) ซึ่งเป็นเอนไซม์ที่ทำงานรักษาสมดุลของเซลล์.

**การเพิ่มความแข็งแรงเชิงประสาทระยะยาว** หากการกระตุนมีความถี่สูงพอที่โซเดียมไฮอ่อนที่ไหลเข้ามาจะสะสมได้ (ความถี่สูงพอที่จะสะสมโซเดียมไฮอ่อน ที่เหลือจากการสูบออกของโซเดียม-โพแทสเซียมปั๊ม) ปริมาณโซเดียมไฮอ่อนที่เพิ่มขึ้นมาก จะเพิ่มระดับแรงดันไฟฟ้าสถิตย์ขึ้น. และเมื่อไฟฟ้าสถิตย์มากพอ มันจะสร้างแรงที่จะผลักแมกนีเซียมไฮอ่อน ที่ปิดเอนเนมตีโอรีเซปเตอร์ออกไปได้. เมื่อแมกนีเซียมหลุดออกไป แคลเซียมไฮอ่อนก็สามารถผ่านเข้ามาทางเอนเนมตีโอรีเซปเตอร์ได้. แคลเซียมไฮอ่อนที่ไหลเข้าปริมาณมากจะช่วยเพิ่มค่าความต่างศักย์ที่ได้รับที่ปลาย денฯ ได้รีตขึ้นไป. นอกจากนั้นแคลเซียมไฮอ่อนปริมาณมากจะจับกับโปรตีนคีแนส (Protein kinases) ซึ่งส่งผลให้เกิดการสร้างแอมปารีเซปเตอร์และติดตั้งแอมปารีเซปเตอร์ที่สร้างใหม่ เข้าไปที่ปลายเชื่อมประสาท ทำให้จำนวน แอมปารีเซปเตอร์เพิ่มขึ้น. ผลการเพิ่มแอมปารีเซปเตอร์จากกระบวนการนี้ จะคงอยู่เพียงแค่เวลาสั้น ๆ ไม่กี่ชั่วโมงเท่านั้น แต่หากมีแคลเซียมไฮอ่อนไหลเข้ามาในปริมาณมากเป็นระยะเวลานานพอ (มีการกระตุนด้วยสัญญาณประสาทความถี่สูงเป็นระยะเวลานาน) จะส่งผลต่อเนื่องไปจนทำให้เกิดการเพิ่มบัจจุยการลอกรหัสดีเอ็นเอ (transcription factor) ซึ่งส่งผลต่อการแสดงออกของยีน (gene expression) และทำให้เกิดการสร้างโปรตีนที่ทำให้เกิดทั้งแอมปารีเซปเตอร์ใหม่เพิ่มขึ้น และ โกรทแฟคเตอร์ (growth factor) ที่จะไปทำให้เกิดการสร้างไชแนปสีใหม่เพิ่มขึ้น ซึ่งผลจากการนี้จะยาวนานและคงทนมาก.

**การลดถอยความแข็งแรงเชิงประสาทระยะยาว** สำหรับการกระตุนที่มีความถี่ต่ำ จะส่งผลให้ปริมาณของแคลเซียมไฮอ่อนอยู่ในระดับต่ำ. ปริมาณของแคลเซียมไฮอ่อนที่อยู่ในระดับต่ำจะไปกระตุนการทำงานของเอนไซม์ฟอสฟะตेज (phosphatase) ซึ่งส่งผลไปลดจำนวนแอมปารีเซปเตอร์ที่ทำงานได้ลง.

การเพิ่มความแข็งแรงเชิงประสาทระยะยาว และ การลดถอยความแข็งแรงเชิงประสาทระยะยาว ต่างก็มีปัจจัยมาจากปริมาณแคลเซียมไฮอ่อนในปลายไชแนปส์ตัวรับ โดย ปริมาณแคลเซียมไฮอ่อนในระดับสูงจะทำให้เกิดการเพิ่มความแข็งแรงเชิงประสาท ระยะยาว และ ปริมาณแคลเซียมไฮอ่อนในระดับต่ำจะทำให้เกิดการลดถอยความแข็งแรงเชิงประสาทระยะยาว. ระดับที่เป็นจุดแบ่งระหว่างการเพิ่มและการลดถอยนี้ เชื่อว่าจะเปลี่ยนแปลงตามสภาพของเซลล์ในลักษณะที่ช่วยรักษาสมดุลย์ (ทฤษฎีบีซีเอ็ม BCM theory[16]) ได้แก่ การเปลี่ยนแปลงในลักษณะการบ้อนกลับเชิงลบ (negative feedback) เช่น เมื่อไชแนปส์อยู่ในภาวะการเพิ่มความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้จะสูงขึ้น เพื่อช่วยลดความเสี่ยงของการเพิ่มมาคล่อง และ เมื่อไชแนปส์อยู่ในภาวะการลดถอยความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้จะลดลงเพื่อช่วยลดความเสี่ยงของการลดถอยจนสูญเสียการเชื่อมต่อ.

จากสภาพลักษณะของระบบประสาทและทฤษฎีบีซีเอ็ม เราอาจกล่าวได้ว่า การเรียนรู้ที่มีประสิทธิภาพคือ การเรียนรู้ต่อเนื่อง สลับกับการหยุดพักผ่อน เพื่อให้เกิดการกระตุนประสาทต่อเนื่องยาวนานเพียงพอ ที่จะทำให้เกิดการสร้างการเรียนรู้ใหม่ และ หยุดพักเพื่อให้ระดับขีดแบ่งปรับตัวลงมา ทำให้การสร้างการเรียนรู้ใหม่ทำได้ง่ายขึ้น เพราะ การเรียนรู้ต่อเนื่องเป็นเวลานานกินไป จะเพิ่มระดับขีดแบ่งซึ่งจะทำให้การเข้าสู่ภาวะการเพิ่มความแข็งแรงเชิงประสาทระยะยาวทำได้ยากขึ้น.

“Never go to excess, but let moderation be your guide.”

---Marcus Tullius Cicero

“ทำสิ่งใดอย่างมากเกินไป ให้ความพอดีเป็นตัวชี้ทาง.”

—มาร์คัส ทูลลิอุส ซิเชอร์

## การฝึกแบบหมุนกับการฝึกแบบออนไลน์

ค่าเกรเดียนต์ที่คำนวณได้จากการแพร่กระจายย้อนกลับ สามารถนำมาช่วยการฝึกโครงข่ายประสาทเทียมได้. แต่สมการเกรเดียนต์ของโครงข่ายประสาทเทียม ต่างจากสมการเกรเดียนต์ของพัฟ์กชันพหุนาม. คำตอบของ เกรเดียนต์พัฟ์กชันพหุนามเป็นศูนย์ สามารถแก้สมการ เพื่อคำนวณหาค่าพารามิเตอร์ของพัฟ์กชันพหุนาม ได้ทันที (คำตอบ สามารถเขียนอยู่ในรูปแบบบิดทางคณิตศาสตร์ได้). แต่สำหรับโครงข่ายประสาทเทียม คำตอบ ไม่สามารถหาได้โดยตรงจากการแก้สมการคณิตศาสตร์ และต้องใช้ขั้นตอนวิธีมาช่วย. ขั้นตอนวิธีการหาค่าดีที่สุด เช่น วิธีลิงเกรเดียนต์ สามารถนำมาช่วยแก้ปัญหานี้ได้.

การนำขั้นตอนวิธีการหาค่าดีที่สุดไปใช้กับการฝึกแบบจำลอง มีประเด็นที่น่าสนใจคือ จังหวะการปรับค่าพารามิเตอร์ ควรทำบ่อยขนาดไหน. ตัวอย่างเช่น วิธีลิงเกรเดียนต์ ปรับค่าพารามิเตอร์ด้วยสมการ 2.46 ที่นำมาเขียนในพจน์ของโครงข่ายประสาทเทียม

$$\Theta^{(i)} = \Theta^{(i-1)} - \alpha \cdot \nabla_{\Theta} E \left( \Theta^{(i-1)} \right) \quad (3.32)$$

เมื่อ  $\Theta^{(i)}$  คือค่าพารามิเตอร์ของแบบจำลอง ที่ได้จากการคำนวณรอบที่  $i^{th}$ . ค่าสเกลาร์  $\alpha$  คือขนาดก้าว ซึ่งสำหรับการฝึกแบบจำลอง นิยมเรียกว่า อัตราเรียนรู้ (learning rate). เวกเตอร์  $\nabla E_{\Theta} \left( \Theta^{(i-1)} \right)$  เป็นค่าเกรเดียนต์ ณ ค่าพารามิเตอร์ก่อนการคำนวณรอบที่  $i^{th}$ . การคำนวณแต่ละรอบ จะเรียกว่า สมัย (epoch).

หากตีความตรงตัว จะได้การปรับค่าน้ำหนัก ตามสมการ 3.32. การปรับค่าพารามิเตอร์ ที่ปรับทีเดียว ในแต่ละสมัย โดย การปรับ ใช้ค่าเฉลี่ยของเกรเดียนต์ที่คิดจากจุดข้อมูลผีกทุกจุด. การปรับค่าน้ำหนักในลักษณะนี้ จะเรียกว่า การฝึกแบบหมุน (batch training).

แต่จากสมการ 3.22 นั้นเท่ากับ  $\nabla_{\Theta} E = \frac{1}{N} \sum_n \nabla_{\Theta} E_n$  เมื่อ  $N$  คือจำนวนจุดข้อมูล ในชุดข้อมูลฝึก ดังนั้น

$$\Theta^{(i)} = \Theta^{(i-1)} - \alpha \cdot \frac{1}{N} \sum_{n=1}^N \nabla_{\Theta} E_n \left( \Theta^{(i-1)} \right). \quad (3.33)$$

หากตัดแปลงเล็กน้อย โดยการปรับค่าพารามิเตอร์ให้ทีขึ้น คือการปรับค่าพารามิเตอร์ แต่ละครั้งสำหรับแต่ละจุดข้อมูล ดังแสดงในสมการ 3.34 และจะปรับค่าพารามิเตอร์  $N$  ครั้งในแต่ละสมัย. การปรับค่าพารามิเตอร์ในลักษณะเช่นนี้ จะเรียกว่า การฝึกแบบออนไลน์ (online training).

$$\Theta^{(i)} = \Theta^{(i-1)} - \frac{\alpha}{N} \cdot \nabla_{\Theta} E_n \left( \Theta^{(i-1)} \right). \quad (3.34)$$

การฝึกแบบออนไลน์ จะสามารถรับข้อมูลเข้ามาฝึกเพิ่มได้ตลอด บางครั้งจึงอาจถูกเรียกว่า การฝึกในโหมดเพิ่ม (training in an incremental mode).

โดยทั่วไป การฝึกแบบหมู่จะทำได้เร็วกว่า แต่ก็ต้องการหน่วยความจำมากกว่าการฝึกแบบออนไลน์. สำหรับคุณภาพการฝึก เยย์กิน[83] ได้อภิปรายถึงข้อดีข้อเสียว่า การฝึกแบบหมู่ จะให้การประมาณค่าเกรเดียนต์ที่แม่นยำกว่า ดังนั้น เมื่อทำงานกับวิธีลงเกรเดียนต์ แล้วจะได้ค่าพารามิเตอร์ที่ดีกว่า. นอกจากนั้น การฝึกแบบหมู่ มีรูปแบบการอนุมานทางสถิติ (statistical inference) อย่างหนึ่ง ดังนั้นการฝึกแบบหมู่ จะเหมาะสมกับงานการหาค่าดัชนoy. แต่ข้อเสียของการฝึกแบบหมู่ ก็คือความต้องการใช้หน่วยความจำบริมาณมาก (เพราะว่าใช้ข้อมูลทั้งหมดที่เดียว ในการคำนวนแต่ละสมัย).

ข้อดีของการฝึกแบบออนไลน์ เมื่อทำร่วมกับการสลับลำดับข้อมูลในแต่ละสมัย จะช่วยลดความเสี่ยงในการเข้าไปติดอยู่ที่ค่าทำให้น้อยที่สุดท่องถิ่นได้ และเมื่อเทียบกับการฝึกแบบหมู่ การฝึกแบบออนไลน์ ต้องการใช้ประโยชน์จากข้อมูลที่ซ้ำซ้อนกันได้มากกว่าการฝึกแบบหมู่. การฝึกแบบออนไลน์ยังอาจช่วยให้การติดตามการเปลี่ยนแปลงเล็ก ๆ ในข้อมูล ทำได้สะดวกขึ้น โดยเฉพาะอย่างยิ่ง สำหรับข้อมูลที่มีลักษณะไม่นิ่งทางสถิติ (nonstationary). แม้ว่าการฝึกแบบออนไลน์ จะทำงานได้ช้ากว่า แต่การฝึกแบบออนไลน์ ก็มีการใช้อย่างแพร่หลาย โดยเฉพาะกับงานการจำแนกรูปแบบ ส่วนหนึ่ง เพราะว่า การฝึกแบบออนไลน์ สามารถขยายขึ้นไปทำงานกับข้อมูลขนาดใหญ่ได้更容易กว่า. อย่างไรก็ตาม พัฒนาการต่อมา นิยมใช้การสมการฝึกแบบหมู่ และการฝึกแบบออนไลน์ ที่เรียกว่า การฝึกแบบหมู่เล็ก ดังที่จะอภิปรายในบท 5.

**การกำหนดค่าเริ่มต้นในการฝึก.** ขั้นตอนวิธีในการฝึกโครงข่ายประสาทเทียม ใช้การปรับปรุงค่าพารามิเตอร์ให้ดีขึ้นเรื่อย ๆ ในแต่ละสมัยฝึก. สมัยฝึกแรกสุด ต้องการค่าเริ่มต้นของพารามิเตอร์. การกำหนดค่าเริ่มต้นของพารามิเตอร์ ที่มักเรียกว่า การกำหนดค่าน้ำหนักเริ่มต้น (weight initialization) วิธีที่นิยมในการกำหนดค่าเริ่มต้น คือการกำหนดค่าเริ่มต้นด้วยการสุ่ม (แบบฝึกหัด 3.12). นอกจากนั้น มีเทคนิคจำนวนมาก สำหรับวิธีการกำหนดค่าเริ่มต้นที่มีประสิทธิภาพ เช่น วิธีเหจี้ยนวิดโอด์ (Nguyen-Widrow weight initialization[141]) หัวข้อ 5.4 อภิปรายวิธีการกำหนดค่าเริ่มต้นที่มีประสิทธิภาพ โดยเฉพาะสำหรับโครงข่ายประสาทเทียมแบบบลีก.

## โครงข่ายประสาทเทียมสำหรับการจำแนกกลุ่ม

สมการ 3.23 แสดงค่าผิดพลาดกำลังสอง ซึ่งนิยมใช้เป็นฟังก์ชันจุดประสงค์ สำหรับงานการหาค่าคาดถอย. การรู้จักรูปแบบ ที่บ่อยครั้ง มักจะถูกติกรอบปัญหาเป็นการจำแนกกลุ่ม มีสองภาระกิจหลัก ๆ คือ การจำแนกค่าทวิภาค และการจำแนกกลุ่ม.

**การจำแนกค่าทวิภาค.** การจำแนกค่าทวิภาค เป็นภาระกิจที่ต้องการทำนายผลลัพธ์ ซึ่งมีรูปแบบที่เป็นไปได้สองแบบ. รูปแบบที่เป็นไปได้ทั้งสอง นั้นมักจะถูกเลือกให้แทนด้วย 1 และ 0 เช่น ปัญหาการทายผลการตรวจข้อมูลอีกซึ่งเต้านมของมวลเนื้อ (แบบฝึกหัด 3.15) ที่ผลท้ายมีอยู่สองอย่างคือ เนื้อร้าย หรือ ไม่ใช่น้ำนม.

การจำแนกค่าทวิภาค (binary classification) ซึ่งต้องการเอาต์พุต  $y \in \{0, 1\}$  นิยมใช้ฟังก์ชันกระตุ้น เอาต์พุตเป็นฟังก์ชันซิกโนไซด์. นั่นคือ  $h(a) = 1/(1 + e^{-a})$  ซึ่งจะทำให้ค่าที่ทำนาย  $\hat{y} \in (0, 1)$ . เพื่อให้การฝึกแบบจำลองทำได้มีประสิทธิภาพ ฟังก์ชันจุดประสงค์ สำหรับการจำแนกค่าทวิภาค นิยมใช้ ฟังก์ชันสูญเสียクロสแอนโตรปี (cross entropy loss). ฟังก์ชันสูญเสียクロสแอนโตรปี สำหรับการจำแนกค่าวิภาค นิยามดังสมการ 3.35.

$$E_n = \begin{cases} -\log(\hat{y}(\mathbf{x}_n, \Theta)) & \text{เมื่อ } y(n) = 1, \\ -\log(1 - \hat{y}(\mathbf{x}_n, \Theta)) & \text{เมื่อ } y(n) = 0. \end{cases} \quad (3.35)$$

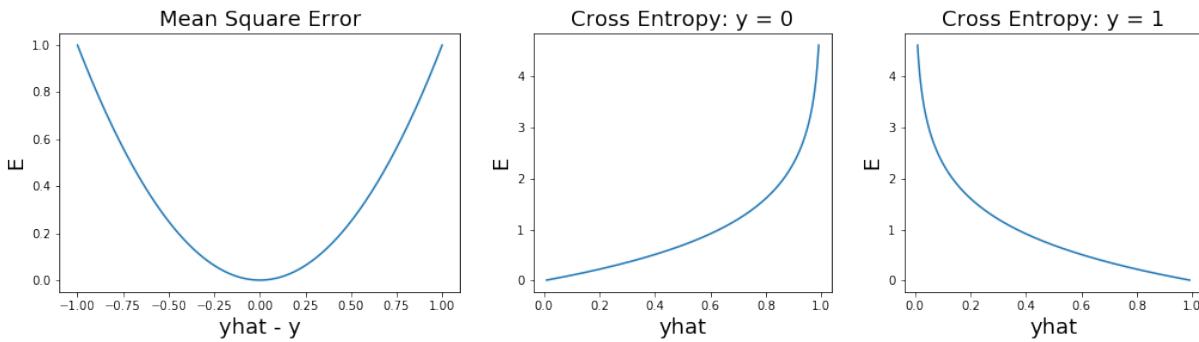
เมื่อ  $y(n)$  คือเฉลย หรือค่าตัวแปรตามของจุดข้อมูลที่  $n^{th}$  และ  $\hat{y}(\mathbf{x}_n, \Theta)$  คือผลท้าย สำหรับจุดข้อมูลที่  $n^{th}$  (ที่มีตัวแปรต้นเป็น  $\mathbf{x}_n$ ) และใช้ค่าพารามิเตอร์  $\Theta$ . สมการ 3.35 มักถูกเขียนย่อเป็น

$$E_n = -y_n \cdot \log(\hat{y}_n) - (1 - y_n) \cdot \log(1 - \hat{y}_n). \quad (3.36)$$

รูป 3.18 แสดงพัฒนารูปของฟังก์ชันสูญเสียชนิดค่าผิดพลาดกำลังสอง เปรียบเทียบกับชนิดクロสแอนโตรปี.

**การจำแนกกลุ่ม.** การจำแนกกลุ่ม เป็นภาระกิจที่ต้องการทำนายกลุ่มของรูปแบบ ตัวอย่างเช่น การรู้จำตัวเลขลายมือ (แบบฝึกหัด 3.16) ที่ต้องการระบุกลุ่มของภาพตัวเลขที่เขียนด้วยลายมือ ว่าอยู่กลุ่มตัวเลขใด ระหว่าง 0 ถึง 9.

การจำแนกกลุ่ม (classification หรือ multi-class classification) ซึ่งต้องการเอาต์พุตที่ระบุฉลากของกลุ่ม ซึ่งฉลาก นิยมกำหนดด้วยรหัสหนึ่งร้อน (one-hot coding หรือ one-of-K coding). รหัสหนึ่งร้อน จะใช้ตัวเลข  $K$  ตัว ในการระบุกลุ่ม  $K$  กลุ่ม. ตำแหน่งของเลขแต่ละตัว แทนกลุ่มที่สนใจ แต่ละกลุ่ม. หาก



รูปที่ 3.18: พฤติกรรมของฟังก์ชันสูญเสีย. ภาพซ้าย แสดงพฤติกรรมฟังก์ชันสูญเสียแบบค่าผิดพลาดกำลังสอง. แกนนอนแสดงผลต่างระหว่างค่าท่านายและค่าเฉลย แกนตั้งแสดงค่าผิดพลาดกำลังสอง. ภาพกลางแสดงค่าฟังก์ชันสูญเสียแบบครอสโอนไทรปี เมื่อเฉลยมีค่าเป็น 0. แกนนอนแสดงค่าท่านาย. หากค่าท่านายถูกต้อง ค่าสูญเสียจะเป็น 0 แต่หากท่านายผิด ค่าสูญเสียจะสูงมาก และหากท่านายผิดเต็มที่ นั่นคือ  $\hat{y} = 1$  ค่าสูญเสียจะเป็นอนันต์ (ไม่สามารถแสดงในภาพ). ภาพขวาแสดงค่าฟังก์ชันสูญเสียแบบครอสเอนไทรปี เมื่อเฉลยมีค่าเป็น 1. เช่นเดียวกับภาพกลาง ค่าสูญเสียจะสูงมาก เมื่อทายผิด.

จุดข้อมูลอยู่ในกลุ่มใด ตัวเลขที่อยู่ตำแหน่งของกลุ่มนั้นจะเป็นหนึ่ง และตัวเลขอื่น ๆ จะเป็นศูนย์. นั่นคือ เอ้าต์พุตในรหัสหนึ่งร้อน  $\mathbf{y} = [y_1, \dots, y_K]^T$  โดย  $y_k \in \{0, 1\}$  และ  $\sum_k y_k = 1$ . การจำแนกกลุ่ม นิยมใช้ฟังก์ชันกระตุ้นเอ้าต์พุต เป็นฟังก์ชันซอฟต์แมกซ์ (softmax function) ซึ่งคำนวณโดย สมการ 3.37.

$$h(\mathbf{a}) = \left[ \frac{e^{a_1}}{\sum_{k=1}^K e^{a_k}} \quad \dots \quad \frac{e^{a_K}}{\sum_{k=1}^K e^{a_k}} \right]^T \quad (3.37)$$

สมการ 3.37 มักเขียนย่อเป็น

$$h(a_i) = \frac{e^{a_i}}{\sum_{k=1}^K e^{a_k}}. \quad (3.38)$$

ฟังก์ชันซอฟต์แมกซ์ ช่วยให้ค่าที่ท่านาย  $\hat{\mathbf{y}} = h(\mathbf{a})$  สามารถเปรียบเทียบได้กับเฉลยที่อยู่ในรหัสหนึ่งร้อน. นั่นคือ  $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_K]^T$  โดย  $\sum_k \hat{y}_k = 1$  และ  $\hat{y}_k \in (0, 1)$  สำหรับทุก  $k = 1, \dots, K$ .

ในการฝึกแบบจำลอง ฟังก์ชันจุดประสงค์สำหรับการจำแนกกลุ่ม นิยมใช้ฟังก์ชันสูญเสียครอสเอนไทรปี ที่นิยามดังสมการ 3.39.

$$E_n = -\log(\hat{y}_c(\mathbf{x}_n, \Theta)) \quad (3.39)$$

เมื่อ  $c$  คือดัชนีของกลุ่มที่เฉลย นั่นคือ  $y_c(n) = 1$ . สมการ 3.39 มักถูกเขียนเป็น

$$E_n = -\sum_{k=1}^K y_k(n) \cdot \log(\hat{y}_k(\mathbf{x}_n, \Theta)) \quad (3.40)$$

เมื่อ  $K$  คือจำนวนของกลุ่มทั้งหมด.

ฟังก์ชันสูญเสียครอสเอนโทรปีในสมการ 3.35 และ 3.39 ต่างมาจากพื้นฐานเดียวกัน แต่ pragmatr ผูกต่อไปกัน. เพื่อลดความสับสน สมการ 3.35 จะเรียกว่า ฟังก์ชันสูญเสียครอสเอนโทรปีสำหรับการจำแนกค่าทวิภาค และอาจเรียกย่อเป็น ครอสเอนโทรปีทวิภาค (binary cross entropy). สมการ 3.39 เป็นครอสเอนโทรปีสำหรับการจำแนกกลุ่ม และอาจเรียกย่อเป็น ครอสเอนโทรปีพหุกลุ่ม (multi-class cross entropy หรือ categorical cross entropy).

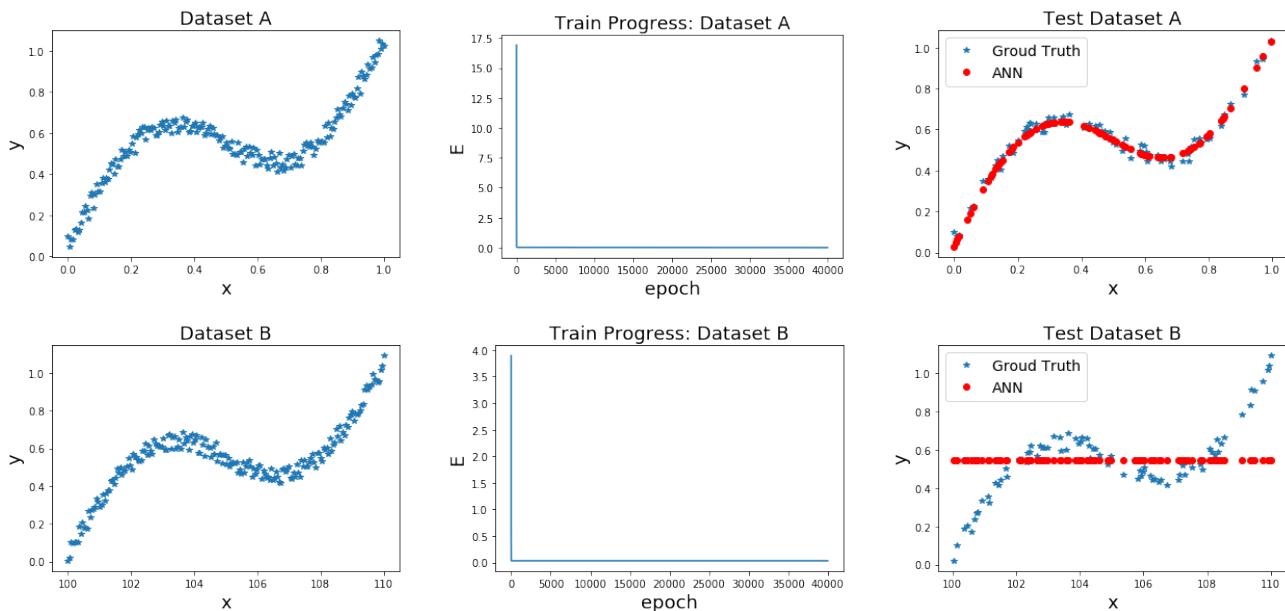
## 3.4 การประยุกต์ใช้โครงข่ายประสาทเทียม

หัวข้อ 3.3 ยกไปยกลạiที่จำเป็นสำหรับการทำงานของโครงข่ายประสาทเทียม. การประยุกต์ใช้โครงข่ายประสาทเทียม ในทางปฏิบัติ มีประเด็นอื่น ๆ ที่ต้องพิจารณาประกอบ เพื่อให้การฝึก และการใช้งานโครงข่ายประสาทเทียมทำงานได้ดี. ประเด็นหนึ่งที่สำคัญ และค่อนข้างจะท้าไป ไม่ได้เจาะจงกับภารกิจใดเฉพาะ คือ การทำนอร์มอลайซ์ (normalization) สำหรับอินพุต. การทำนอร์มอลายซ์สำหรับอินพุต คือ การปรับขนาดอินพุต เพื่อช่วยให้การฝึก และการทำงานของโครงข่ายประสาทเทียมทำได้ง่ายขึ้น.

ก่อนอภิปรายวิธีการทำนอร์มอลายซ์อินพุต พิจารณาผลการฝึก และทดสอบโครงข่ายประสาทเทียมของข้อมูลสองชุด ซึ่งลักษณะข้อมูล ความก้าวหน้าในการฝึก และผลการทำนายกับข้อมูลทดสอบ แสดงในรูป 3.19. ผลการทดสอบ (ระบุในคำบรรยายภาพ) และภาพในรูป 3.19 แสดงให้เห็นชัดเจนว่า แบบจำลองทำงานได้ดี กับข้อมูลชุดเอ (dataset A) แต่ทำงานได้แย่มากกับข้อมูลชุดบี (dataset B). ข้อมูลทั้งสองชุด มีลักษณะความสัมพันธ์ระหว่างตัวแปรต้น  $x$  และตัวแปรตาม  $y$  ในแบบเดียวกัน. ความต่างระหว่างข้อมูลชุดเอและชุดบี มีอย่างเดียวคือ ขนาดของอินพุตของข้อมูล. ขนาดของอินพุตของข้อมูลชุดเอ อยู่ในช่วง 0 ถึง 1 ในขณะที่ขนาดของอินพุตของข้อมูลชุดบี อยู่ในช่วง 100 ถึง 110.

พิจารณาใกล้ของการแพร่กระจายย้อนกลับ โดยเฉพาะสมการ 3.31 จะเห็นว่า  $h'(a_j^{(q)})$  มีผลโดยตรงกับเกรเดียนต์. ค่า  $h'(a_j^{(q)})$  คืออนุพันธ์ของฟังก์ชันกราฟต้น ซึ่งซิกมอยด์เป็นฟังก์ชันกราฟต้นที่ใช้. อนุพันธ์ของซิกมอยด์ (แสดงในรูป 3.20) จะมีค่าน้อยมาก ๆ เมื่อตัวกราฟต้นมีค่าใหญ่ ๆ (เช่นมากกว่าห้า หรือน้อยกว่าลบห้า) ซึ่งจะขับค่าของฟังก์ชันซิกมอยด์ไปที่ปลาย<sup>2</sup> (ค่าใกล้ ๆ หนึ่ง หรือค่าใกล้ ๆ สูญญ). ค่าอินพุตที่มีขนาดใหญ่จะทำให้ตัวกราฟต้นมีค่าใหญ่ และจะส่งผลให้อนุพันธ์ของซิกมอยด์มีค่าน้อย ซึ่งส่งผลต่อให้เกรเดียนต์มีค่าน้อย

<sup>2</sup>บางครั้งช่วงค่าซิกมอยด์ในย่านที่มีค่าใกล้ ๆ หนึ่ง หรือค่าใกล้ ๆ สูญญ มักถูกเรียกว่า ช่วงอิมตัว (saturation region) เนื่องจาก ในย่านเหล่านั้น ค่าฟังก์ชันซิกมอยด์มีการเปลี่ยนแปลงน้อยมาก เมื่อเทียบกับค่าตัวกราฟต้น (อนุพันธ์มีค่าใกล้สูญญ).



รูปที่ 3.19: การใช้โครงข่ายประสาทเทียมกับข้อมูลชุดเดียว (แคว้น) และข้อมูลชุดบี (ตรวจสอบ). จุดข้อมูลต่าง ๆ แสดงในภาพซ้าย. การดำเนินการกับข้อมูลทั้งสองชุดทำเหมือนกันทุกประการ ได้แก่ แบ่งข้อมูลออกเป็นชุดฝึกและชุดทดสอบ ฝึกแบบจำลองกับข้อมูลชุดฝึก (ความกว้างหน้าในการฝึก บ่งชี้จากค่าผิดพลาดต่อสมัยฝึก แสดงในภาพกลาง) และทดสอบกับข้อมูลชุดทดสอบ. ข้อมูลชุดบีได้ค่าผิดพลาดกำลังสองเป็น 0.0007 สำหรับทั้งชุดฝึกและชุดทดสอบ. ข้อมูลชุดบีได้ค่าผิดพลาดกำลังสองเป็น 0.0338 สำหรับชุดฝึก และ 0.0369 สำหรับชุดทดสอบ. ผลการคำนวณแบบจำลอง แสดงในภาพขวา.

และสุดท้าย ส่งผลให้การฝึกแบบจำลองทำได้ช้า.

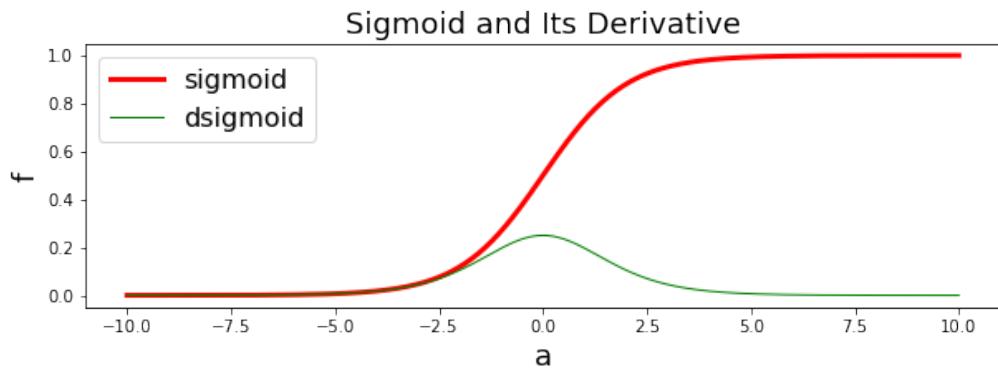
การทำอัร์มอยไลซ์อินพุต ซึ่งเป็นการปรับขนาดของอินพุตให้อยู่ในช่วงที่การฝึกแบบจำลองทำได้ง่าย จึงสำคัญอย่างมากในทางปฏิบัติ. การเรียนรู้สำหรับสิงมีชีวิตก็เช่นเดียวกัน หากสภาพแวดล้อมเหมาะสมกับการเรียนรู้ สิงมีชีวิตก็จะเรียนรู้ได้เร็วขึ้น เรียนรู้ได้ดีขึ้น.

การทำอัร์มอยไลซ์สามารถทำได้หลายวิธี สองวิธีที่ ๑ ไปที่นิยมมาก คือ วิธีการปรับสูตรช่วงที่กำหนด และวิธีการปรับสูตรค่าสถิติที่กำหนด. วิธีการปรับสูตรช่วงที่กำหนด จะปรับขนาดของอินพุตให้ค่าอยู่ในช่วงที่กำหนด เช่น  $[-1, 1]$  หรือ  $[0, 1]$ . หากช่วงที่ต้องการคือ  $[x'_{\min}, x'_{\max}]$  ค่าอินพุตที่ผ่านการทำอัร์มอยไลซ์ หรือเรียกว่า นอร์มอยไลด์อินพุต (normalized input) สัญลักษณ์  $x'$  สามารถคำนวณจาก

$$x' = (x'_{\max} - x'_{\min}) \cdot \frac{x - x_{\min}}{x_{\max} - x_{\min}} + x'_{\min} \quad (3.41)$$

เมื่อ  $x$  คือค่าอินพุตเดิม และ  $x_{\min}$  กับ  $x_{\max}$  คือค่าขั้นต่ำและขีดสูงที่สุดของอินพุตเดิม.

วิธีการปรับสูตรค่าสถิติที่กำหนด นิยมปรับค่าอินพุต เพื่อให้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน เป็น 0 กับ



รูปที่ 3.20: พังค์ชันซิกมอยด์ (เส้นหนาสีแดง) และอนุพันธ์ (เส้นบางสีเขียว).

1 ตามลำดับ. ดังนั้น นอร์มอิลเด้อินพุต  $x'$  สามารถคำนวณได้จาก

$$x' = \frac{x - \bar{x}}{\sigma_x} \quad (3.42)$$

เมื่อ  $\bar{x}$  และ  $\sigma_x$  คือค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน ของค่าอินพุตเดิม.

หมายเหตุ สำหรับอินพุตขนาดหลายมิติ  $\mathbf{x} \in \mathbb{R}^D$  เมื่อ  $D$  เป็นจำนวนมิติของบริภูมิอินพุต. การทำนอร์มอิลเด้อ ยังช่วยรักษาสมดุลของขนาดของอินพุตในมิติต่าง ๆ กันด้วย. การทำนอร์มอิลเด้อ จะทำแต่ละมิติ เช่น กรณีสองมิติ  $\mathbf{x} = [x_1, x_2]^T$  และต้องการทำนอร์มอิลเด้อให้อยู่ในช่วง  $[0, 1]$  ห้ามคุณจะดำเนินการโดย

$$\mathbf{x}' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} (x_1 - x_{1\min}) / (x_{1\max} - x_{1\min}) \\ (x_2 - x_{2\min}) / (x_{2\max} - x_{2\min}) \end{bmatrix}$$

เมื่อ  $x'_1$  และ  $x'_2$  คือค่านอร์มอิลเด้ออินพุต ของมิติที่หนึ่งและมิติที่สองตามลำดับ และ  $x_{1\min}$  กับ  $x_{1\max}$  คือค่าน้อยที่สุดกับค่ามากที่สุดของอินพุตเดิมในมิติที่หนึ่ง และ  $x_{2\min}$  กับ  $x_{2\max}$  คือค่าน้อยที่สุดกับค่ามากที่สุดของอินพุตเดิมในมิติที่สอง.

**การหยุดก่อนกำหนด.** ประเด็นของการไอเวอร์ฟิต และคุณสมบัติความทั่วไป เป็นประเด็นสำคัญสำหรับการใช้งานแบบจำลองทำนาย ที่รวมถึงโครงข่ายประสาทเทียม. การเลือกใช้แบบจำลองที่มีความยืดหยุ่นสูง ๆ เช่น การใช้โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยซ่อนมาก ๆ หรือการใช้โครงข่ายประสาทเทียมที่มีชั้นลึก ๆ ก็เสี่ยงที่จะเกิดการไอเวอร์ฟิตได้.

รูป 3.21 แสดงผลการฝึกโครงข่ายสองชั้นขนาด 100 หน่วยซ่อน กับข้อมูลฝึก 40 จุดข้อมูล. ระหว่าง 5000 ถึง 10000 สมัย ให้ผลการทำนายที่ดี และการฝึกต่อเพิ่มไป นอกจากเสียเวลาเพิ่ม แล้วยังทำให้แบบ

จำลองโอลิเวอร์พิตอิกด้วย ตั้งเห็นได้จากค่าผิดพลาดกับข้อมูลทดสอบที่เพิ่มขึ้น (ภาพล่างสุดซ้าย) และเวลาที่ใช้ยังเป็นร้าว ๆ 9 ถึง 18 เท่าอิกด้วย เวลาที่ใช้ฝึกระบุในคำบรรยายภาพ. หมายเหตุ เวลาที่ระบุ แสดงเป็นเวลา นอร์มอลайซ์ด์ (normalized time). นั่นคือ เวลาที่แสดงเป็นอัตราส่วน โดยใช้เวลาที่อ้างอิงเป็นตัวหาร. เวลาที่รายงานในรูป 3.21 ใช้เวลาที่ฝึกแบบจำลอง 5000 สมัยเป็นเวลาอ้างอิง. การรายงานเวลา ด้วยเวลา นอร์มอลายซ์ด์ แทนเวลาสมบูรณ์ ช่วยบอกความรวมของเวลาการทำงาน โดยลดความจำเป็นในการรายงานรายละเอียด ของฮาร์แวร์และระบบที่ใช้ทดสอบลง.

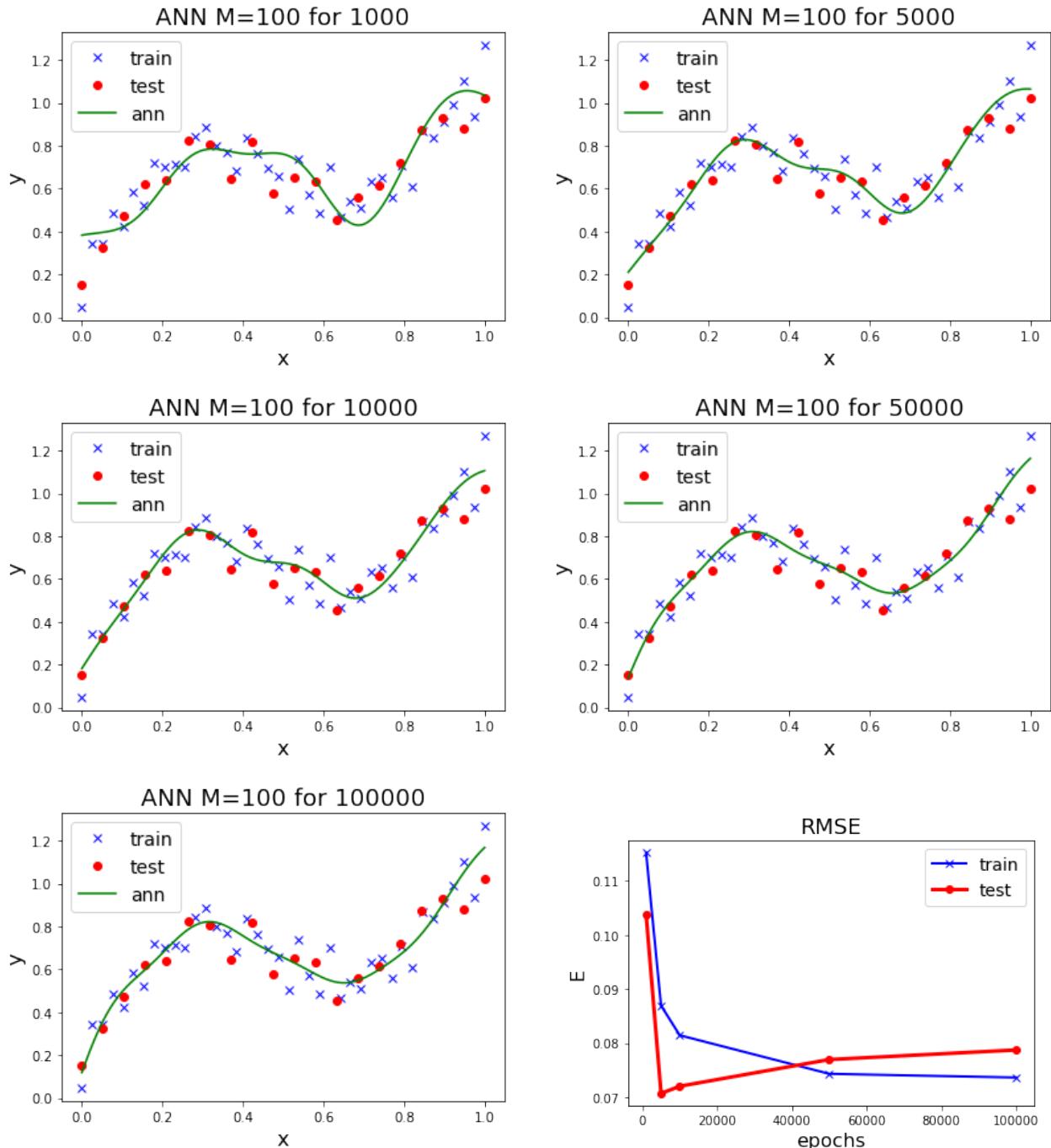
การฝึกโครงข่ายประสาทเทียมใช้เวลานาน การฝึกโดยการใช้สมัยฝึกมาก จนเกิดโอลิเวอร์พิต นอกจากจะได้แบบจำลองที่ไม่มีคุณสมบัติความทั่วไปแล้ว ยังเสียเวลาฝึกด้วย. การปฏิบัติที่นิยม ก็คือ การหยุดก่อนกำหนด (early stopping). การหยุดก่อนกำหนด คือ การใช้เงื่อนไขเพื่อยุดการฝึก โดยเงื่อนไข คือเมื่อค่าผิดพลาดของข้อมูลทดสอบสูงขึ้นกว่าเดิม ให้หยุดการฝึก. ค่าผิดพลาดของข้อมูลทดสอบสูงขึ้นกว่าเดิม เป็นสัญญาณของการโอลิเวอร์พิต และเพื่อให้ไม่มีการใช้ข้อมูลทดสอบสุดท้าย ในกระบวนการการฝึกจริง ๆ ข้อมูลทดสอบ เพื่อใช้สำหรับการหยุดก่อนกำหนดจะแบ่งออกมาจากข้อมูลชุดฝึก เรียกว่า **ข้อมูลตรวจสอบ** (validation data).

ข้อมูลตรวจสอบ เป็นข้อมูลอิกซุตที่แยกออกจาก. ข้อมูลตรวจสอบ “ไม่ใช้ในการฝึก (ไม่ใช้ในการคำนวณ เพื่อปรับค่าน้ำหนัก) และไม่ใช้ในการทดสอบสุดท้าย (ไม่ใช้ในการทดสอบ เพื่อรายงานผลการทำงานของแบบจำลอง). ข้อมูลตรวจสอบ เป็นข้อมูลที่ใช้ในกระบวนการการฝึก แต่ไม่ได้ใช้ในการฝึก ข้อมูลตรวจสอบ จะใช้เพื่อเลือกแบบจำลอง หรือใช้เพื่อเลือกความซับซ้อนของแบบจำลอง (เช่น จำนวนหน่วยช่อง) หรือใช้เพื่อการหยุดก่อนกำหนด เป็นต้น.

### 3.5 คำแนะนำสำหรับการใช้แบบจำลองทำนาย

การประเมินผล เป็นกลไกที่สำคัญมากสำหรับการใช้แบบจำลองทำนาย. แต่หากผลที่ประเมินได้ไม่น่าพอใจ และต้องการปรับปรุงให้ได้ผลการทำนายดีขึ้น มีทางเลือกต่าง ๆ มากมาย เช่น การเพิ่มจำนวนข้อมูลที่จะใช้สำหรับการฝึก หรือการคัดเลือกตัวแปรต้นของข้อมูล โดยเลือกเฉพาะคุณลักษณะที่สำคัญ นั่นคือเลือกเฉพาะมิติ บางมิติที่สำคัญ มาใช้เป็นอินพุตสำหรับแบบจำลอง หรือการเพิ่มคุณลักษณะใหม่เข้าไปในอินพุตของแบบจำลอง หรือการเพิ่มความซับซ้อนของแบบจำลองขึ้น หรือการลดความซับซ้อนของแบบจำลองลง.

ทางเลือกที่หลากหลาย อาจทำให้สับสนได้ และการปรับปรุงแบบจำลอง ควรจะลองทำอะไรก่อน อะไรหลัง ซึ่งการลองทำแต่ละอย่าง อาจใช้เวลา多く และยังอาจเพิ่มงบประมาณด้วย เช่น การอุ่นไฟฟ้าข้อมูลมาเพิ่ม (เพิ่มจำนวนจุดข้อมูล) หรือการเพิ่มลักษณะสำคัญชนิดใหม่เข้าไป (เพิ่มมิติใหม่สำหรับอินพุต). ผู้เชี่ยวชาญ



รูปที่ 3.21: ผลการฝึกโครงข่ายสองชั้นขนาดหน่วยซ่อน 100 หน่วย ที่จำนวนสมัยฝึก 1000, 5000, 10000, 50000 และ 100000 รอบ. ห้าภาพแรก แสดงจุดข้อมูลฝึก (ภาคบาทสีฟ้า) จุดข้อมูลทดสอบ (วงกลมสีแดง) ค่าทำงานจากแบบจำลอง (เส้นสีเขียว) ซึ่งภาพระบุจำนวนหน่วยซ่อน และจำนวนสมัยที่ได้ทำการฝึกไป. ภาพสุดท้าย (ล่างสุดซ้าย) แสดงค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสองที่วัดจากข้อมูลฝึก (ภาคบาทและเส้นบางสีฟ้า) และที่วัดจากข้อมูลทดสอบ (วงกลมและเส้นหนาสีแดง) ต่อจำนวนสมัยฝึก. ค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสองของชุดทดสอบ ได้แก่ 0.1038, 0.0707, 0.0720, 0.0770, และ 0.0787 หลังจากฝึกไปแล้ว 1000, 5000, 10000, 50000, และ 100000 สมัยตามลำดับ โดยใช้เวลาฝึก เป็น 0.2456, 1, 1.9123, 9.4386, และ 18.3509 ตามลำดับ.

ศาสตร์การเรียนรู้ของเครื่อง แอนดรอย อิง[139] แนะนำว่า ก่อนตัดสินใจเลือกทดลองปรับปรุงด้วยวิธีใด ควรทำการทดลองง่าย ๆ และใช้เส้นโค้งเรียนรู้ (Learning Curve) เป็นตัวชี้แนะ. การใช้เส้นโค้งเรียนรู้ มีพื้นฐานมาจาก การศึกษาเรื่องคุณภาพการทำนาย ที่สัมพันธ์กับความล้าเอียง (bias<sup>3</sup>) และความแปรปรวน (variance).

## ความล้าเอียงกับความแปรปรวน

พิจารณารูป 3.5 ที่ใช้ฟังก์ชันพหุนามทำนายข้อมูล. ภาพ x ใน รูป 3.7 แสดงค่าผิดพลาดที่ได้ต่อระดับขั้น. ระดับขั้น บอกระดับความซับซ้อนของแบบจำลองฟังก์ชันพหุนาม. ที่ความซับซ้อนของแบบจำลองที่เหมาะสม ค่าผิดพลาดของชุดข้อมูลทดสอบ จะมีค่าต่ำที่สุด.

ฟังก์ชันพหุนามระดับขั้นเก้ากีโอเวอร์พิตข้อมูลอย่างชัดเจน. ส่วนฟังก์ชันพหุนามระดับขั้นศูนย์ ระดับขั้นหนึ่ง และระดับขั้นสองอันเดอร์พิตของข้อมูลอย่างชัดเจน. การอันเดอร์พิต (underfit) หมายถึง การที่ความซับซ้อนของแบบจำลอง ไม่เพียงพอที่จะประมาณความสัมพันธ์ของข้อมูลได้. ช่วงที่ความซับซ้อนของแบบจำลองน้อยเกินไป มีจุดสังเกตสำคัญคือ ค่าผิดพลาดจะสูงกับทั้งชุดทดสอบและชุดฝึกหัด. เมื่อความซับซ้อนของแบบจำลองไม่พอ แบบจำลองจะไม่ยึดหยุ่นพอที่ปรับตัว เพื่อลดค่าผิดพลาดกับชุดทดสอบชุดฝึกหัด. กรณีที่แบบจำลองทำงานได้ไม่ดี เนื่องจากความซับซ้อนของแบบจำลองน้อยเกินไปนี้ จะเรียกว่า กรณีที่มีความล้าเอียงสูง (high bias) ซึ่งสื่อถึงการอันเดอร์พิต.

ส่วนกรณีที่ความซับซ้อนของแบบจำลองมากเกินไป ซึ่งคือกรณีการโอเวอร์พิตมีจุดสังเกตที่สำคัญคือ ค่าผิดพลาดกับชุดฝึกจะต่ำมาก แต่ค่าผิดพลาดกับชุดทดสอบจะสูง. แบบจำลองที่มีความซับซ้อนมาก จะสามารถปรับพฤติกรรมการทำนาย เพื่อลดค่าผิดพลาดกับชุดทดสอบชุดฝึกหัดลงได้ดี แต่หากความซับซ้อนมากเกินไป จะไปลดค่าผิดพลาดกับชุดทดสอบชุดฝึกหัดที่เกิดจากสัญญาณรบกวนด้วย แบบจำลองจะทำนายข้อมูลชุดทดสอบได้ไม่ดี. กรณีนี้จะเรียกว่า เป็นกรณีที่มีความแปรปรวนสูง (high variance) ซึ่งสื่อถึงการโอเวอร์พิต.

กล่าวอีกอย่างหนึ่ง ความล้าเอียงสูง หมายถึงแบบจำลองของไม่ยึดหยุ่นมากพอ ส่วนความแปรปรวนสูงหมายถึงแบบจำลองของยึดหยุ่นมากเกินไป. แบบจำลองที่ดีที่สุดที่ทำได้ คือ แบบจำลองที่ลดให้ทั้งความล้าเอียงและความแปรปรวนมีค่าต่ำ. แต่ไม่ว่าแบบจำลองไหนก็ตาม เราทำดีที่สุด ได้แค่ดีที่สุด จะฝืนข้อจำกัดของธรรมชาติไม่ได้.

ทฤษฎีที่กล่าวถึงข้อจำกัดของการทำแบบจำลองนี้คือ ทวิบถของความล้าเอียงกับความแปรปรวน (Bias/

<sup>3</sup>ในบริบทของพฤติกรรมโดยรวมของแบบจำลอง ไม่ใช่บริบทของพารามิเตอร์ของโครงข่ายประสาทเทียม.

Variance Dilemma). ทวิบตของความลำเอียงกับความแปรปรวน เสนอโดย เจมันและคณะ[70] จากการศึกษาความสามารถ และขีดจำกัดของการทำแบบจำลอง. ทวิบตของความลำเอียงกับความแปรปรวน สรุปว่า เราสามารถทำแบบจำลองให้ดีที่สุดได้ โดยลดทั้งค่าความลำเอียงและความแปรปรวนให้ต่ำ. แต่ค่าผิดพลาด ก็มีขีดจำกัดหนึ่ง (ขึ้นกับธรรมชาติของปัญหา) ที่เราไม่สามารถลดค่าผิดพลาดลงไปให้ต่ำกว่านั้นได้. การที่เราพยายามจะลดค่าผิดพลาดให้ต่ำกว่านั้น โดยการลดส่วนหนึ่ง ก็จะไปเพิ่มอีกส่วน เช่น หากพยายามลดค่าผิดพลาดจากความลำเอียงมากเกินไป ก็จะทำให้ส่วนที่เกิดจากความแปรปรวนเพิ่ม และในทางกลับกัน หากพยายามลดค่าผิดพลาดจากความแปรปรวนมากเกินไป ก็จะทำให้ส่วนที่เกิดจากความลำเอียงเพิ่ม.

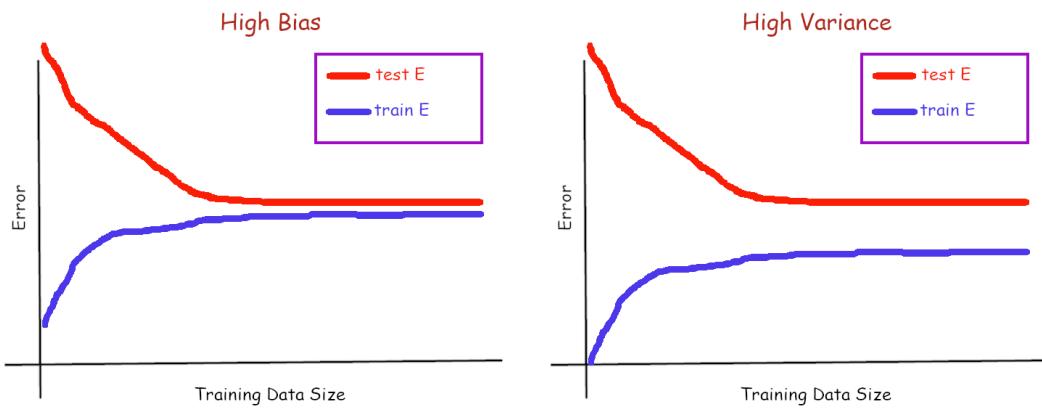
## เส้นโค้งเรียนรู้

เส้นโค้งเรียนรู้ (Learning Curve) เป็นเครื่องมือช่วยแสดงความสัมพันธ์ระหว่างความพยายามที่ใช้ไปในการฝึกแบบจำลอง กับผลการทำงานของแบบจำลอง. เส้นโค้งเรียนรู้ สามารถใช้เพื่อช่วยให้เงื่อน件ว่า แบบจำลองที่ใช้อยู่ มีความเสี่ยงที่จะมีความลำเอียงสูง หรือเสี่ยงที่จะมีความแปรปรวนสูง. เส้นโค้งเรียนรู้ สร้างโดย การตรวจสอบผลการฝึกแบบจำลองที่จำนวนจุดข้อมูลฝึกต่าง ๆ และนำค่าผิดพลาดเฉลี่ยกับชุดฝึก และค่าผิดพลาดเฉลี่ยกับชุดทดสอบ มาวาดกราฟ ดังแสดงในรูป 3.22.

การวัดค่าผิดพลาดกับชุดฝึก วัดเฉพาะกับจุดข้อมูลที่ใช้ฝึก เช่น หากฝึกแบบจำลองด้วย 10 จุดข้อมูล ก็หาค่าผิดพลาดของการทำงานค่า 10 จุดข้อมูลนี้. ดังนั้น เมื่อจำนวนจุดข้อมูลฝึกน้อย แบบจำลองมีโจทย์ที่ต้องทำน้อย ก็มีโอกาสที่จะทำค่าผิดพลาดของชุดฝึกน้อยด้วย. เมื่อเพิ่มจำนวนจุดข้อมูลฝึกขึ้น ค่าผิดพลาดของชุดฝึกก็จะเพิ่มขึ้น และถูกเข้าสู่ค่า ๆ หนึ่ง. ในขณะเดียวกัน เมื่อจำนวนจุดข้อมูลฝึกเพิ่มขึ้น ค่าผิดพลาดของชุดทดสอบจะลดลงจนถูกเข้าสู่ค่า ๆ หนึ่ง.

ในกรณีที่ แบบจำลองมีความเสี่ยงจากความลำเอียงสูง ถ้าข้อมูลทดสอบ และข้อมูลฝึกมีปริมาณเพียงพอ พฤติกรรมการทำงานของแบบจำลอง จะให้ผลในลักษณะเดียวกัน และทำให้ค่าผิดพลาดจากชุดฝึก และจากชุดทดสอบ มีค่าใกล้เคียงกัน. แต่กรณีที่แบบจำลองมีความแปรปรวนสูง นั่นคือ แบบจำลองมีความยืดหยุ่นมากเกินไป มากจนพอที่จะปรับตัวเข้ากับสัญญาณรบกวนในข้อมูลฝึกได้. ผลคือแบบจำลองสามารถลดค่าผิดพลาดของชุดฝึกได้ต่ำ และทำให้ผลต่างระหว่างค่าผิดพลาดจากชุดฝึก และจากชุดทดสอบมีค่าต่างกัน. ดังนั้น ความต่างระหว่างค่าผิดพลาดจากข้อมูลทดสอบและจากข้อมูลฝึกจึงสามารถใช้ปัจจัยสถานการณ์คร่าว ๆ ของแบบจำลองได้.

รูป 3.22 เป็นภาพวาดเพื่อให้เห็นภาพรวม แต่ ในสถานการณ์จริง เส้นโค้งเรียนรู้ที่ได้ อาจจะมีสัญญาณ



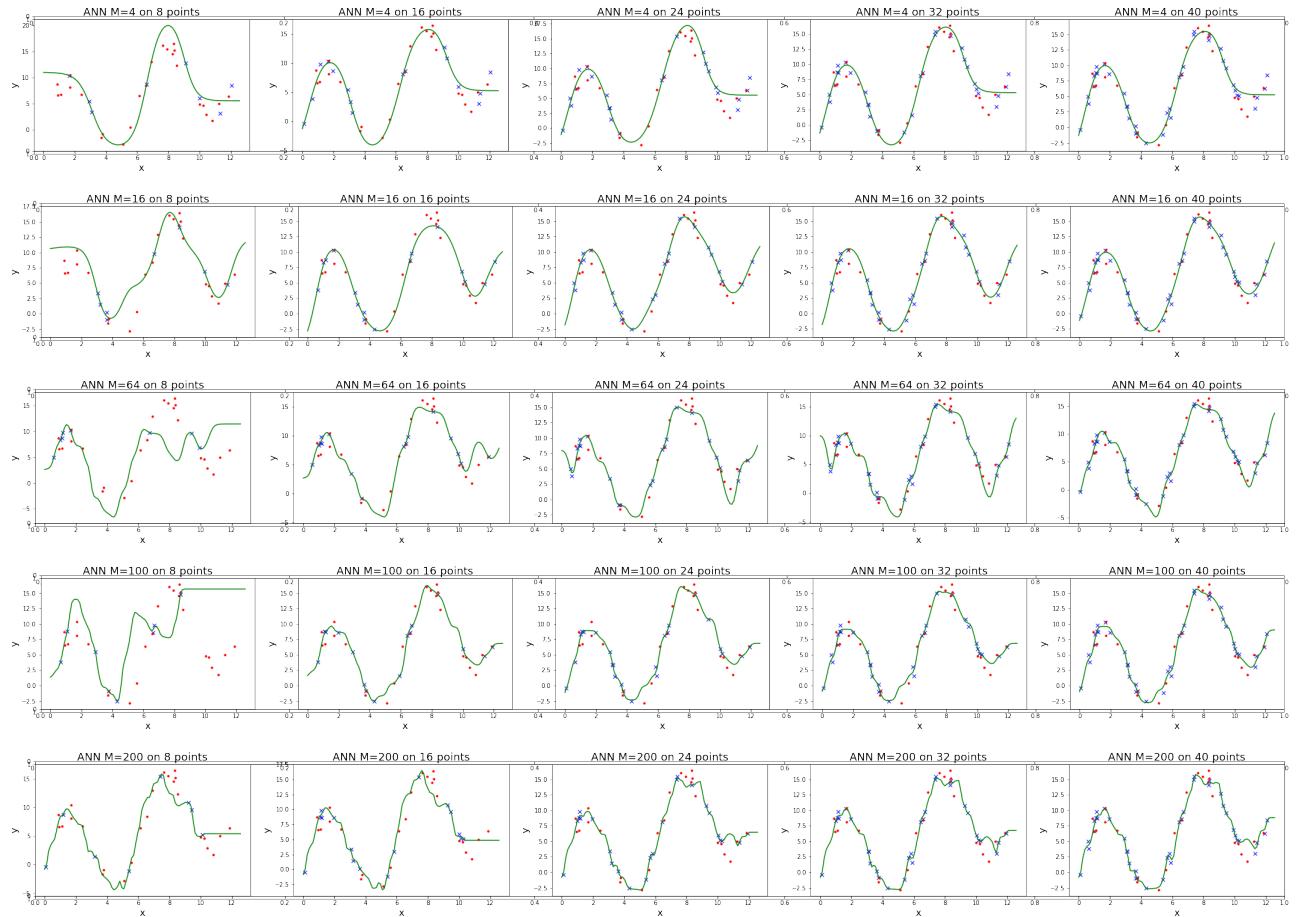
รูปที่ 3.22: ภาพวาดเส้นโค้งเรียนรู้. ภาพซ้าย แสดงกรณีแบบจำลองมีความล้าเอียงสูง. เมื่อจำนวนจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดกับชุดฝึก และค่าผิดพลาดกับชุดทดสอบ มีค่าใกล้เคียงกัน. ภาพขวา แสดงกรณีแบบจำลองมีความแปรปรวนสูง. เมื่อจำนวนของจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดกับชุดฝึก และค่าผิดพลาดกับชุดทดสอบ มีค่าต่างกัน.

รบกวนมาก หรือแม้แต่สเกลของการวาดกราฟ อาจทำให้ต้องใช้ความระมัดระวัง ในการอ่านผลจากเส้นโค้งเรียนรู้. รูป 3.23 แสดงกระบวนการทดสอบแบบจำลอง เพื่อวัดเส้นโค้งเรียนรู้. รูป 3.24 แสดงเส้นโค้งเรียนรู้ที่ได้จากแบบจำลองที่มีความซับซ้อนต่าง ๆ กัน เพื่อให้เห็นตัวอย่างลักษณะของเส้นโค้งเรียนรู้ในกรณีต่าง ๆ.

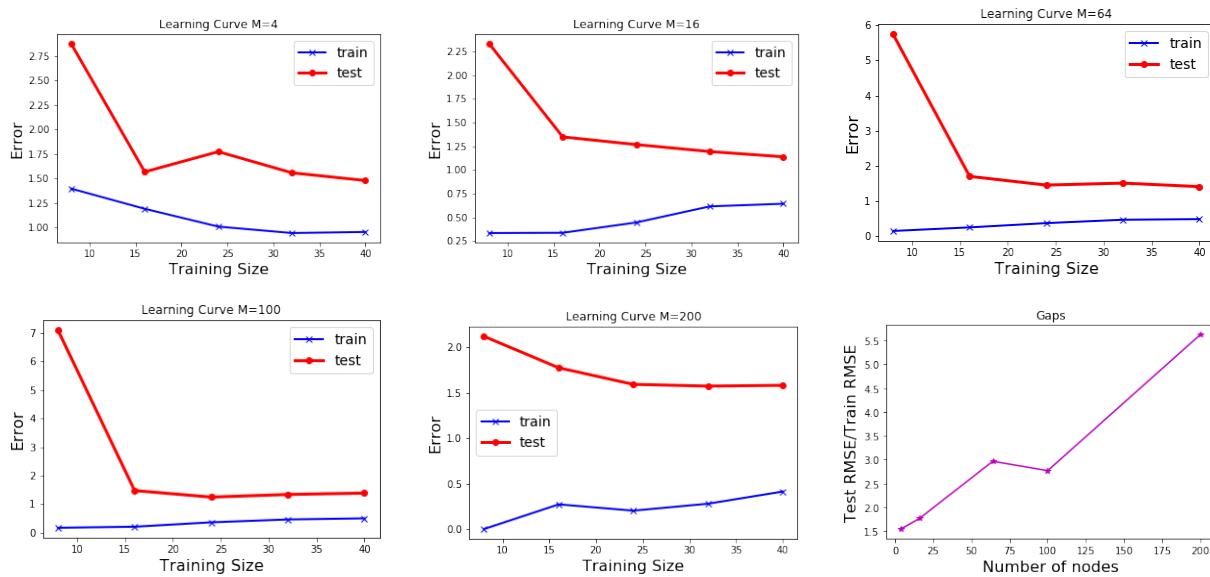
จากรูป 3.23 และคำบรรยาย แบบจำลองโครงข่ายประสาทเทียมสองชั้น ขนาด 4 หน่วยซ่อน อันเดอร์พิตข้อมูล และเส้นโค้งเรียนรู้ ก็ถูกเข้าหา กัน โดยอัตราความต่างระหว่างค่าผิดพลาดจากชุดทดสอบและชุดฝึกค่อนข้างต่ำ (ประมาณ 1.55). แบบจำลองโครงข่ายประสาทเทียมสองชั้นขนาด 200 หน่วยซ่อน โอบเวอร์พิตข้อมูลอย่างชัดเจน (รูป 3.23) และเส้นโค้งเรียนรู้ก็ถูกเข้าห่างกันชัดเจน โดยอัตราความต่างระหว่างค่าผิดพลาดจากชุดทดสอบและชุดฝึกสูง (ประมาณ 5.62 รูป 3.24).

หลังจากพอธุแล้วว่า สถานการณ์อยู่ในกรณีความล้าเอียง หรืออยู่ในกรณีความแปรปรวนสูง แอนดรูว์อิง[139] แนะนำให้พิจารณาทางเลือก ดังนี้

- เก็บข้อมูลมาเพิ่มสำหรับการฝึก. การเพิ่มจำนวนข้อมูลในการฝึก จะช่วยในกรณีความแปรปรวนสูง.
- เลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต. การเลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต. การลดมิติของอินพุตลง ก็จะช่วยในกรณีความแปรปรวนสูง โดยเฉพาะ สำหรับโครงข่ายประสาทเทียม. ตัวอย่าง เช่น โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยซ่อนเท่าเดิม การลดมิติของอินพุตลง เท่ากับลดจำนวนพารามิเตอร์ลง.
- เพิ่มลักษณะที่สำคัญใหม่เข้าไปในอินพุต. การเพิ่mlักษณะที่สำคัญใหม่เข้าไปในอินพุต การเพิ่มมิติของ



**รูปที่ 3.23:** ตัวอย่างกระบวนการทดสอบแบบจำลองเพื่อวัดเส้นโค้งเรียนรู้ สำหรับโครงข่ายประสาทเทียมสองชั้น ขนาด 4 ขนาด 16 ขนาด 64 ขนาด 100 และขนาด 200 หน่วยช่อง จากแควนลงล่าง ตามลำดับ. แต่ละແຄูແສດງ 5 ภาพ ที่แต่ละภาพเป็นการฝึกกับข้อมูลฝึกขนาด 8 จุด ขนาด 16 จุด ขนาด 24 จุด ขนาด 32 จุด และขนาด 40 จุด จากซ้ายมาขวา ตามลำดับ. แต่ละภาพใช้ภาคบทสีฟ้าแสดงจุดข้อมูลฝึก และใช้วงกลมสีแดงแสดงแสดงจุดข้อมูลทดสอบ เส้นสีเขียวแสดงค่าที่แบบจำลองที่ฝึกเสร็จแล้วทำงาน. ซึ่งแต่ละภาพระบุความซับซ้อนของแบบจำลองด้วยจำนวนหน่วยช่อง และระบุจำนวนจุดข้อมูลที่ใช้ฝึก. ตัวอย่างนี้ เพื่อแสดงในเห็นกระบวนการดำเนินการอย่างชัดเจน และเพื่อเปรียบเทียบผลของเส้นโค้งเรียนรู้ ในสถานการณ์จริง การใช้เส้นโค้งเรียนรู้ ก็เพื่อหลีกเลี่ยงการที่จะต้องทดลองโดยตรงกับหลาย ๆ แบบจำลองเช่นนี้. ค่ารากสองของค่าเฉลี่ยค่าผิดพลาดกำลังสองของโครงข่ายสองชั้น กับข้อมูลทดสอบ 40 จุด คือ 1.48, 1.14, 1.40, 1.39, และ 1.58 เมื่อใช้หน่วยช่อง 4, 16, 64, 100, และ 200 หน่วยตามลำดับ.



รูปที่ 3.24: ตัวอย่างเล่นโค้ดเรียนรู้ จากแบบจำลองขนาด 4, 16, 64, 100, และ 200 หน่วยช่อง ตามที่ระบุในชื่อแต่ละภาพ. ภาพล่างขวาแสดงความต่างระหว่างค่าผิดพลาดจากข้อมูลฝึกกับค่าผิดพลาดจากข้อมูลทดสอบ (แกนต์) และความซับซ้อนของแบบจำลอง (แกนนอน). ความต่างแสดงในรูปแบบอัตราส่วน นั่นคือ  $E_{\text{test}}/E_{\text{train}}$  เมื่อ  $E_{\text{test}}$  คือรากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากข้อมูลทดสอบ และ  $E_{\text{train}}$  คือ จากข้อมูลฝึก. ความต่างในภาพ คือ 1.55, 1.77, 2.97, 2.77, และ 5.62 สำหรับแบบจำลองขนาด 4, 16, 64, 100, และ 200 หน่วยช่อง ตามลำดับ.

อินพุตขึ้น โดยทั่วไปแล้ว จะช่วยในการณีความล้าเอียงสูง.

- เพิ่มความซับซ้อนของแบบจำลองขึ้น. การเพิ่มความซับซ้อนของแบบจำลอง เช่น การเพิ่มจำนวนหน่วยช่อง จะช่วยในการณีความล้าเอียงสูง.
- ลดความซับซ้อนของแบบจำลองลง. การลดความซับซ้อนของแบบจำลอง เช่น การลดจำนวนหน่วยช่อง การทำเรกูลาริซ์ หรือ การทำการหดก่อนกำหนด จะช่วยในการณีความแปรปรวนสูง.

ตาราง 3.4 สรุปทางเลือกที่สอนครูวิชี [139] แนะนำ สำหรับกรณีความล้าเอียงสูง และกรณีความแปรปรวนสูง. สุดท้าย หากลองวิธีทั่ว ๆ ไปดังนี้แล้ว ผลยังไม่น่าพอใจ อาจจะลองวิเคราะห์ และตรวจสอบผลที่ขึ้นตอนวิธีทำผิด ดูว่าผิดลักษณะไหน อย่างไร. เพื่อว่า อาจจะพบคุณสมบัติเฉพาะบางอย่างที่ผลมักจะผิด เช่น หากเป็นการจำแนกตัวเลขจากภาพ ขั้นตอนวิธีที่ใช้ อาจจะจำแนกเลข 2 เป็นเลข 4 บ่อย ๆ ซึ่งเมื่อดูภาพของเลขที่จำแนกผิด ก็อาจพบว่า มีรูปแบบการเขียนเลข 2 แบบหนึ่ง ที่มักจะจำแนกผิด. หากพบรูปแบบเฉพาะนั้น อาจเพิ่มการจัดการพิเศษเฉพาะสำหรับรูปแบบนั้นได้.

ตารางที่ 3.4: สรุปทางเลือกที่แนะนำในกรณีความจำเอียงสูงและความแปรปรวนสูง.

ทางเลือก	กรณี	
	ความจำเอียงสูง	ความแปรปรวนสูง
เพิ่มรอบฝึก	แนะนำ	
เพิ่มจำนวนหน่วยบอย	แนะนำ	
ทำเรกูล่าไรซ์		แนะนำ
ทำหยุดก่อนกำหนด		แนะนำ
ลดมิติของอินพุตลง		แนะนำ
เพิ่มมิติของอินพุตขึ้น	แนะนำ	แนะนำ
เพิ่มจำนวนจุดข้อมูลฝึก		แนะนำ

## 3.6 อภิรานศัพท์

**การปรับเส้นโค้ง (curve fitting):** การปรับพหุติกร姆ของแบบจำลองท่านาย โดยการเปลี่ยนค่าของพารามิเตอร์ เพื่อให้พหุติกร姆ท่านายสอดคล้องกับข้อมูลที่มี.

**จุดข้อมูล (datapoint):** ตัวอย่างข้อมูลที่มีค่าของตัวแปรต้น  $x$  และตัวแปรตาม  $y$  ที่เป็นคู่กัน.

**พังก์ชันพหุนาม (polynomial function):** พังก์ชัน  $f : \mathbb{R} \mapsto \mathbb{R}$  ที่แสดงความสัมพันธ์ระหว่างตัวแปรต้น  $x$  และตัวแปรตาม  $y$  ด้วยสมการพหุนาม  $y = f(x, w) = \sum_{m=0}^M w_m x^m$  เมื่อ พารามิเตอร์ของพังก์ชัน  $w = [w_0, w_1, \dots, w_M]^T$ .

**การฝึก (training):** การฝึก หรือการเรียนรู้ คือการปรับค่าพารามิเตอร์ของแบบจำลอง เพื่อให้แบบจำลองมีพหุติกร姆การทำนายสอดคล้องกับข้อมูล.

**ค่าเฉลย (ground truth):** ค่าของตัวแปรตาม  $y$  จากข้อมูล ที่คู่กับตัวแปรต้น  $x$  ที่สนใจ.

**คุณสมบัติความทั่วไป (generalization):** ความสามารถของแบบจำลองท่านาย ที่สามารถทำนายข้อมูลที่ไม่เคยเห็นได้ดี.

**ข้อมูลฝึก (training data):** ข้อมูลที่ใช้ในกระบวนการการฝึกแบบจำลอง

**ข้อมูลทดสอบ (test data):** ข้อมูลที่ใช้ในทดสอบแบบจำลอง.

**การโอเวอร์ฟิต (overfitting):** การที่แบบจำลองสามารถทำนายข้อมูลฝึกได้ดี แต่ทำนายข้อมูลใหม่ที่ไม่เคยเห็นได้ไม่ดี นั่นคือ การที่แบบจำลองไม่มีคุณสมบัติความทั่วไป.

**ความซับซ้อนของแบบจำลอง (model complexity):** การยึดหยุ่นของแบบจำลองอาจบ่งชี้ได้จากจำนวนพารามิเตอร์ของแบบจำลอง.

**โครงข่ายประสาทเทียม (artificial neural network):** แบบจำลองคำนวณ ที่ทำการคำนวณ โดยใช้หน่วยคำนวณย่อยหลาย ๆ หน่วย ที่แต่ละหน่วยทำการคำนวณในแบบคล้าย ๆ กัน.

**เพอร์เซปตรอนหลายชั้น (multi-layer perceptron):** โครงข่ายประสาทเทียม ที่หน่วยคำนวณต่าง ๆ ต่อ กันเป็นโครงข่ายในลักษณะชั้นคำนวณ.

**โนนด หรือหน่วยคำนวณ (node หรือ unit):** การคำนวณย่อยของโครงข่ายประสาทเทียม ที่มีลักษณะการคำนวณง่าย ๆ ไม่ซับซ้อน.

**ชั้นคำนวณ (layer):** กลุ่มของโนนด ในโครงข่ายประสาทเทียม ที่จัดโครงสร้างเป็นลักษณะชั้นคำนวณ โดยกลุ่มของโนนดในชั้นคำนวณเดียวกัน จะรับอินพุตจาก(กลุ่มของโนนดใน)ชั้นคำนวณก่อนหน้า หรือจะส่งเอ้าต์พุตออกไปให้(กลุ่มของโนนดใน)ชั้นคำนวณถัดไป หรือทั้งสองอย่าง.

**ชั้นซ่อน (hidden layer):** ชั้นคำนวณที่จะส่งเอ้าต์พุตออกไปให้(กลุ่มของโนนดใน)ชั้นคำนวณถัดไป โดยเอ้าต์พุตของชั้นซ่อนจะไม่ใช่เอ้าต์พุตสุดท้ายของโครงข่าย.

**หน่วยซ่อน (hidden node หรือ hidden unit):** โนนดในชั้นซ่อน.

**ค่าน้ำหนักและค่าไบอส (weights and biases):** ค่าพารามิเตอร์ต่าง ๆ ของโครงข่ายประสาทเทียม.

**ฟังก์ชันกระตุ้น (activation function):** ฟังก์ชันคำนวณของโนนด.

**ชั้นเอ้าต์พุต (output layer):** ชั้นคำนวณชั้นสุดท้าย ที่เอ้าต์พุตของชั้น จะเป็นเอ้าต์พุตของโครงข่าย.

**การแพร่กระจายย้อนกลับ (backpropagation หรือ error backpropagation):** ขั้นตอนวิธีการคำนวณหาค่าเกรเดียนต์ เพื่อปรับค่าน้ำหนัก สำหรับโครงข่ายประสาทเทียม.

**การฝึกแบบหมู่ (batch training):** การฝึกที่ใช้ข้อมูลฝึกทั้งหมดที่เดียว นั่นคือ การปรับค่าพารามิเตอร์ทำเดียวในแต่ละสมัยฝึก.

**การฝึกแบบออนไลน์ (online training):** การฝึกที่ใช้ข้อมูลฝึกที่ลงทะเบียนจุดข้อมูล นั่นคือ การปรับค่าพารามิเตอร์จะทำหลาย ๆ ครั้ง แต่ละครั้งสำหรับแต่ละจุดข้อมูล และจะทำงานกว่าจะครบทุกจุดข้อมูล ในแต่ละสมัยฝึก.

**สมัย (epoch):** รอบการปรับค่าพารามิเตอร์ โดยแต่ละรอบจะนับเมื่อมีการใช้ข้อมูลฝึกครบทุกจุด.

**อัตราเรียนรู้ (learning rate):** ขนาดก้าว ที่เป็นค่าสเกลาร์ เพื่อควบคุมความเร็วในการฝึกแบบจำลอง เป็นค่าที่ใช้กับขั้นตอนวิธีการหาค่าดีที่สุดที่อยู่เบื้องหลังการฝึก.

**การกำหนดค่าน้ำหนักเริ่มต้น (weight initialization):** การกำหนดค่าน้ำหนักและไบอัส ให้กับโครงข่ายประสาทเทียม ก่อนการฝึก.

**การจำแนกค่าทวิภาค (binary classification):** ภารกิจการทำนาย ที่ผลการทำนายมีได้สองแบบ.

**ฟังก์ชันสูญเสียクロสเอนโทรปี (cross entropy loss):** ฟังก์ชันจุดประสงค์ สำหรับการจำแนกค่าทวิภาค หรือการจำแนกกลุ่ม.

**รหัสหนึ่งร้อน (one-hot coding หรือ one-of-K coding):** รูปแบบแทนข้อมูลที่มีลักษณะเป็นกลุ่ม โดยรหัสจะมีจำนวนส่วนประกอบเท่ากับจำนวนกลุ่มทั้งหมด และตำแหน่งของแต่ละส่วนประกอบ แทนฉลากของกลุ่มแต่ละกลุ่ม. รหัสจะระบุฉลากของกลุ่ม โดยกำหนดให้ ส่วนประกอบที่อยู่ตำแหน่งฉลากนั้น มีค่าเป็นหนึ่ง และส่วนประกอบอื่น ๆ มีค่าเป็นศูนย์.

**ซอฟต์แมกซ์ (softmax):** ฟังก์ชันคำนวณ เพื่อควบคุมให้อาตโนมัติ ให้อยู่ในรูปแบบที่สามารถเปรียบเทียบได้กับรหัสหนึ่งร้อน.

**การทำอرمอลайเซอินพุต (input normalization):** การปรับขนาดของอินพุตทั้งหมด.

**การหยุดก่อนกำหนด (early stopping):** การทำเงื่อนไขจบการฝึก โดยใช้ข้อมูลตรวจสอบ.

**ข้อมูลตรวจสอบ (validation data):** ชุดข้อมูล เพื่อเสริมกระบวนการเตรียมแบบจำลอง อาจใช้ช่วยกระบวนการฝึก แต่ไม่ได้ใช้ฝึกแบบจำลองโดยตรง.

### 3.7 แบบฝึกหัด

“I learned that courage was not the absence of fear, but the triumph over it. The brave man is not he who does not feel afraid, but he who conquers that fear.”

---Nelson Mandela

“ผมได้เรียนรู้ว่า ความกล้าหาญไม่ใช่การปราศจากความกลัว แต่เป็นการเอาชนะความกลัว. คนกล้าหาญ ไม่ใช่คนที่ไม่มีสีสืบกลัว แต่เป็นคนที่อยู่เหนือความกลัวนั่น.”

—เนลสัน แมนเดลา

#### แบบฝึกหัด 3.1

จากตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง ในหัวข้อ 3.1 จงเขียนรูปสมการในลักษณะเดียวกับสมการ 3.9 สำหรับแบบจำลองพหุนามระดับขั้นใด ๆ  $m$ . คำให้ ลองทำสำหรับระดับขั้นสอง หรือระดับขั้นสามก่อน.

#### แบบฝึกหัด 3.2

จงแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันซิกมอยด์ คือ

$$h'(a) = z \cdot (1 - z) \quad (3.43)$$

เมื่อ  $h'(a) = \frac{dh(a)}{da}$  และ  $a$  คือผลรวมการกระตุ้น และ  $z$  คือผลลัพธ์จากการกระตุ้น นั่นคือ  $z = h(a)$ .

#### แบบฝึกหัด 3.3

จงแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันไฮเปอร์บอลิกแทนเจนต์  $\tanh(a) = (e^a - e^{-a}) / (e^a + e^{-a})$

คือ

$$\tanh'(a) = 1 - z^2 \quad (3.44)$$

เมื่อ  $\tanh'(a) = \frac{d\tanh(a)}{da}$  และ  $a$  คือผลรวมการกระตุ้น และ  $z$  คือผลลัพธ์จากการกระตุ้น นั่นคือ  $z = \tanh(a)$ .

#### แบบฝึกหัด 3.4

จงแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันเรเดียลเบชิล (radial basis function)  $r(a) = e^{-a^2}$  คือ

$$r'(a) = -2a \cdot z \quad (3.45)$$

เมื่อ  $r'(a) = \frac{dr(a)}{da}$  และ  $a$  คือผลรวมการกระตุ้น และ  $z$  คือผลลัพธ์จากการกระตุ้น นั่นคือ  $z = r(a)$ .

### แบบฝึกหัด 3.5

จะแสดงให้เห็นว่า  $\delta_k^{(L)} = \frac{\partial E}{\partial a_k^{(L)}} = \hat{y}_k - y_k$  สำหรับกรณีดังนี้

- (ก) การหาค่าคาดถอย ใช้ฟังก์ชันเอกลักษณ์ ซึ่งคือ  $\hat{y}_k = a_k^{(L)}$   
และฟังก์ชันจุดประสงค์ค่าผิดพลาดกำลังสอง คือ  $E = \frac{1}{2} \sum_k (\hat{y}_k - y_k)^2$ .
- (ข) การจำแนกค่าทวิภาค ใช้ฟังก์ชันซิกโนแอด์ ซึ่งคือ  $\hat{y}_k = \frac{1}{1 + \exp(-a_k^{(L)})}$   
และฟังก์ชันจุดประสงค์クロสเอนโทรปี  $E = - \sum_k \{y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)\}$ .
- (ค) การจำแนกกลุ่ม ใช้ฟังก์ชันซอฟต์แมกซ์ ซึ่งคือ  $\hat{y}_k = \frac{\exp(a_k^{(L)})}{\sum_{j=1}^K \exp(a_j^{(L)})}$   
และฟังก์ชันจุดประสงค์クロสเอนโทรปี  $E = - \sum_j y_j \log(\hat{y}_j)$ .

### แบบฝึกหัด 3.6

การทำเรกูลารไซซ์ กล่าวง่าย ๆ คือการควบคุมพฤติกรรมการทำนายของแบบจำลอง เพื่อช่วยลดความเสี่ยงการโอเวอร์ฟิต โดยยังคงความซับซ้อนของแบบจำลองไว้. โครงข่ายประสาทเทียมสามารถทำเรกูลารไซซ์ได้ดังเช่น ฟังก์ชันจุดประสงค์ในสมการ 3.23 สามารถถูกดัดแปลงเป็น

$$\text{loss}_n = E_n + \frac{\lambda}{2} \sum_q \sum_j \sum_i w_{ji}^2(q) \quad (3.46)$$

เมื่อ  $w_{ji}(q) \equiv w_{ji}^{(q)}$  แทนค่าน้ำหนักในชั้น  $q^{th}$  ของแบบจำลอง. หมายเหตุ สัญลักษณ์  $w_{ji}(q)$  ใช้แทน  $w_{ji}^{(q)}$  เพื่อลดความรุนแรง. จะแสดงให้เห็นว่า

$$\frac{\partial \text{loss}_n}{\partial w_{ji}^{(q)}} = \frac{\partial E_n}{\partial w_{ji}^{(q)}} + \lambda w_{ji}^{(q)}. \quad (3.47)$$

สังเกตว่า การทำเรกูลารไซซ์ไม่รวมค่าไบอส.

### แบบฝึกหัดเขียนโปรแกรม

### แบบฝึกหัด 3.7

จากตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง ในหัวข้อ 3.1 โปรแกรมในรายการ 3.3 แสดงตัวอย่างการปรับเส้นโค้งด้วยฟังก์ชันพหุนามระดับขั้นหนึ่ง. สามารถทัดเกรกเป็นการเตรียมข้อมูล. บรรทัดที่สี่ เป็นการฝึกแบบจำลอง ซึ่งเรียกใช้โปรแกรม **train\_poly1** ที่แสดงในรายการ 3.2. หลังจากฝึกแบบจำลองเรียบร้อยแบบจำลองที่ฝึกเสร็จ (แบบจำลองที่เลือก พร้อมค่าพารามิเตอร์ที่นำมาได้) จะสามารถนำไปใช้งาน ซึ่งคือการทำนายคำตอบ จากค่าที่สามารถได้ บรรทัดสุดท้าย แสดงตัวอย่างที่ทำนายค่า  $y$  สำหรับค่า  $x = 5$  ซึ่งทำโดยเรียกใช้ฟังก์ชัน **fmodel** ที่โปรแกรมแสดงในรายการ 3.1.

จะทำการทดสอบโปรแกรมเหล่านี้ ทดสอบโปรแกรม และเบรี่ยบเทียบผลกับตัวอย่างในหัวข้อ 3.1.

รายการ 3.1: ตัวอย่างฟังก์ชันพหุนาม

---

```

1 def fmodel(x, w): # e.g., fmodel(5, [0.7, -0.65, 1])
2     w = np.array(w).reshape((-1,1))
3     m = len(w)
4     y = 0
5     for i in range(m):
6         y += w[i] * x**i
7     return y

```

---

รายการ 3.2: ตัวอย่างฟังก์ชันฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง

---

```

1 def train_poly1(datax, datay):
2     N = datax.shape[0]
3     sumx, sumx2 = np.sum(datax), np.sum(datax**2)
4     sumy, sumyx = np.sum(datay), np.sum(datay*datax)
5     A = np.array([[N, sumx], [sumx, sumx2]])
6     b = np.array([[sumy], [sumyx]])
7
8     wt = np.linalg.solve(A, b)
9     return wt

```

---

รายการ 3.3: ตัวอย่างการปรับเส้นโค้งด้วยฟังก์ชันพหุนามระดับขั้นหนึ่ง

---

```

1 DX = [0.000, 0.111, 0.222, 0.333, 0.444, 0.556, 0.667, 0.778, ←
      0.889, 1]
2 DY = [0.160, 0.724, 0.931, 0.712, 0.610, -0.460, -0.684, -1.299, ←
      -1.147, -0.045]
3 X, Y = np.array(DX), np.array(DY)
4 wo = train_poly1(X, Y); print('trained w =\n', wo)
5 print('Predict y = %.3f at x = 5'%fmodel(5, wo))

```

---

### แบบฝึกหัด 3.8

จากแบบฝึกหัด 3.1 และตัวอย่างโปรแกรมในแบบฝึกหัด 3.7 จะเขียนฟังก์ชัน `train_poly` ที่รับอาร์กิวเม้นต์เป็นข้อมูล `datax` และ `datay` และระดับขั้นของฟังก์ชันพหุนาม  $M$  เพื่อฝึกแบบจำลองพหุนามระดับขั้น  $M$ .

### แบบฝึกหัด 3.9

จากแบบฝึกหัด 3.8 จะเขียนโปรแกรม เพื่อศึกษาคุณสมบัติความทวีปีของแบบจำลอง (หัวข้อ 3.2) โดยการสร้างข้อมูล  $y = \sin(2\pi x) + \epsilon$  เมื่อ  $\epsilon \sim \mathcal{N}(0, 0.3)$  โดยสร้างข้อมูลขึ้นมา 10 จุดข้อมูลสำหรับการฝึก และ 5 จุดข้อมูลสำหรับการทดสอบ. ให้  $x$  อยู่ในช่วง 0 ถึง 1. พร้อมเขียนโปรแกรม เพื่อวาดกราฟดังรูป 3.6 และ 3.7.

คำใบ้ ดูคำสั่ง `np.linspace` และ `np.random.normal`.

### แบบฝึกหัด 3.10

รายการ 3.4 แสดงโปรแกรมคำนวนโครงข่ายประสาทเทียม. โปรแกรมคำนวนตามสมการ 3.19 และ 3.20. โดยรับ จำนวนขั้นคำนวน ผ่าน `net_params['layers']`. ค่าของพารามิเตอร์ต่าง ๆ ก็รับผ่าน `net_params` เช่น ค่าไบอสขั้นที่หนึ่ง ผ่าน `net_params['bias1']` ค่าน้ำหนักขั้นที่หนึ่ง ผ่าน `net_params['weight1']` โดยตัวเลขตามหลังชื่อระบุขั้นของพารามิเตอร์. ค่าไบอส เป็นเวกเตอร์ ที่มีจำนวนส่วนประกอบเท่ากับจำนวนโนนดในขั้นคำนวน. ค่าน้ำหนัก เป็นเมทริกซ์ขนาด  $M_q \times M_{q-1}$  เมื่อ  $M_q$  คือจำนวนโนนดของขั้นคำนวน และ  $M_{q-1}$  คือจำนวนโนนดของขั้นคำนวนก่อนหน้า. เพื่อความสะดวกในการเขียนโปรแกรม อินพุตถูกกำหนด เป็นแมตทริกซ์จากขั้นคำนวนที่ศูนย์. อินพุต  $X$  ที่รับเข้าต้องอยู่ในรูปเมทริกซ์ ขนาด  $D \times N$  เมื่อ  $D$  เป็นจำนวนมิติของอินพุต และ  $N$  เป็นจำนวนจุดข้อมูล. โครงข่ายประสาทเทียมจะให้อาต์พุตออกมา ในรูปเมทริกซ์ ขนาด  $K \times N$  เมื่อ  $K$  คือจำนวนมิติของอาต์พุตที่ต้องการ.

การกำหนดจำนวนโนนดในแต่ละขั้นคำนวน ทำทางอ้อมผ่านการกำหนดขนาดของค่าไบอส และขนาดของค่าน้ำหนัก. นอกจากจำนวนขั้นคำนวน ค่าไบอส และค่าน้ำหนักแล้ว ฟังก์ชันกระตุ้นสามารถกำหนดได้ในแต่ละขั้นคำนวน เช่น หากกำหนดฟังก์ชันกระตุ้นของขั้นคำนวนที่หนึ่ง เป็นฟังก์ชันจำกัดแข็ง อาจทำโดย การกำหนดค่า `net_params['act1']` ให้เป็น `hardlimit` เมื่อ `hardlimit` จึงฟังก์ชันจำกัดแข็งที่โปรแกรมแสดงในรายการ 3.5.

ข้อสังเกต ฟังก์ชัน `hardlimit` ไม่ได้เขียนโดยใช้คำสั่ง `if`. อะไรคือข้อดีข้อเสีย ของการเขียนโปร-

แกรมในแบบรายการ 3.5 เปรียบเทียบกับการเขียนโดยใช้คำสั่ง **if**

รายการ 3.4: โปรแกรมคำนวณโครงข่ายประสาทเทียม

---

```

1 def mlp(net_params, X):
2     assert X.shape[0] == net_params['weight1'].shape[1], 'X: D,N'
3
4     num_layers = net_params['layers']
5     # Feed forward
6     Z = X
7     for i in range(1, num_layers):
8         b = net_params['bias%d'%i]
9         w = net_params['weight%d'%i]
10        act_f = net_params['act%d'%i]
11
12        A = np.dot(w, Z) + b      # A: M x N
13        Z = act_f(A)           # Z: M x N
14
15    return Z # M x N

```

---

รายการ 3.5: โปรแกรมคำนวณฟังก์ชันจำกัดแข็ง

---

```

1 def hardlimit(a):
2     return 1*(a > 0)

```

---

การใช้งานโปรแกรมคำนวณโครงข่ายประสาทเทียม สามารถทำได้ เช่น หากทำการคำนวณตระกูลเอ็กซ์ ออร์ ในรูป 3.15 การคำนวณสามารถทำได้ดังแสดงในรายการ 3.6. ตัวแปร **net** กำหนดจำนวนชั้นคำนวณ ค่าน้ำหนัก ค่าไบอส และฟังก์ชันกระตุ้น.

จะศึกษาโปรแกรมเหล่านี้ ทดลองรัน และปรับแต่งโครงข่ายประสาทเทียม โดยใช้ค่าน้ำหนักและค่าไบอส นี่ หรือปรับแต่งเป็นโครงสร้างอื่น สังเกตผล และสรุป. หมายเหตุ โครงข่ายในรูป 3.15 เป็นโครงข่ายสองชั้น แต่การเรียกใช้โปรแกรม **mlp** กำหนด '**layers**': 3 ชั้นในโปรแกรม **mlp** นับอินพุตเป็นชั้นคำนวณ ที่ศูนย์เข้าไปด้วย โดยชั้นคำนวณที่ศูนย์ไม่มีการคำนวณ (ใช้  $Z = X$  และเริ่มคำนวณลูปจากด้านหลัง ดูรายการ 3.4 ประกอบ).

รายการ 3.6: ตัวอย่างการปรับแต่งโปรแกรมคำนวณโครงข่ายประสาทสำหรับรูป 3.15

---

```

1 net = {'layers': 3, 'bias1': np.array([[ -20], [-20]]),
2        'weight1': np.array([[ -30,  30],[30,  -30]]),
3        'bias2': np.array([[ -20]]), 'weight2': np.array([[30,  30]]),
4        'act1': hardlimit, 'act2': hardlimit}

```

```

5
6 x = np.array([[0, 0, 1, 1], [0, 1, 0, 1]])
7 y = mlp(net, x)
8 print(x[0,:])
9 print(x[1,:])
10 print(y[0,:])

```

---

### แบบฝึกหัด 3.11

รายการ 3.7 แสดงโปรแกรมฝึกโครงข่ายประสาทเทียม `train_mlp` ที่ใช้วิธีแพร่กระจายย้อนกลับ คำนวณค่าเกรดเดียนต์ และใช้วิธีลงชันที่สุด เพื่อปรับค่าพารามิเตอร์. การปรับค่าพารามิเตอร์ใช้ข้อมูลฝึกทุกจุด ที่เดียว และการปรับทำครั้งเดียวในแต่ละสมัยฝึก นี้คือ การฝึกแบบหมู่. โปรแกรม `train_mlp` รับแบบ จำลอง (พร้อมค่าพารามิเตอร์เริ่มต้น) ผ่านอาร์กิวเมนต์ `net_params` ซึ่งเป็นไฟลอนดิกชันนารีที่มีคุณaje ต่าง ๆ ดังอภิรายในแบบฝึกหัด 3.10. อาร์กิวเมนต์ `trainX` และ `trainY` เป็นตัวแปรต้น และตัวแปร ตามของข้อมูลฝึก ที่อยู่ในรูป  $D \times N$  และ  $K \times N$  ตามลำดับ เมื่อ  $D, K$ , และ  $N$  เป็นจำนวนมิติของอินพุต จำนวนมิติของเอ็ตพุต และจำนวนจุดข้อมูล ตามลำดับ. อาร์กิวเมนต์ `loss` อ้างถึงฟังก์ชันลูกูเลีย ที่จะใช้ เป็นฟังก์ชันจุดประสงค์. อาร์กิวเมนต์ `lr` แทนอัตราการเรียนรู้. อาร์กิวเมนต์ `epochs` แทนจำนวนสมัยที่ จะฝึก.

โปรแกรม `train_mlp` รีเทิร์นไฟลอนดิกชันนารี `net_params` ที่แทนโครงสร้างแบบจำลอง ซึ่ง ค่าพารามิเตอร์ได้ถูกปรับเปลี่ยนไปแล้ว จากการฝึก และรีเทิร์นไฟลอลิสต์ `train_losses` ที่บันทึกค่า เคลี่ยค่าผิดพลาดกำลังสองของแบบจำลองหลังฝึกแต่ละสมัย. สังเกต ขั้นตอนที่สามของการแพร่กระจายย้อน กลับ ในโปรแกรมใช้ `delta[i - 1] = dsigmoid(Z[i - 1]) * sumdw` ซึ่งเรียกใช้ออน- พันธ์ของซิกมอยด์ ตั้งนั้น หากชั้นช่อนใช้ฟังก์ชันกราฟตันอื่น นอกจากซิกมอยด์ จะต้องแก้ไขโปรแกรมที่ส่วนนี้. ในเบื้องต้นนี้ โปรแกรมใช้คำสั่ง `assert act_f == sigmoid` เพื่อป้องกัน ความพลั้งเหลือที่อาจ เกิดขึ้น.

รายการ 3.7: โปรแกรมฝึกโครงข่ายประสาทเทียม

---

```

1 def train_mlp(net_params, trainX, trainY, loss, lr=0.1,
2                 epochs=1000):
3     #
4     #      net_params
5     #      * 'Layers': number of layers, inc. input layer.
6     #      * 'bias1': bias of layer 1.

```

```

7      # * 'weight1': weight of layer 1.
8      # * 'act1': activation function of layer 1.
9      # * ...
10     # Loss: Loss function with interface loss(yp, y).
11     # Lr: Learning rate.
12     # epochs: number of training epochs.
13     #
14 num_layers = net_params['layers']
15 last_layer = num_layers-1
16
17 out_act = 'act%d'%last_layer
18 _, N = trainX.shape
19 A = {}
20 Z = {0: trainX}
21 delta = {}
22 dEw = {}
23 dEb = {}
24 train_losses = []
25 step_size = lr/N
26
27 for nt in range(epochs):
28     # (1) Forward pass
29     for i in range(1, num_layers):
30         b = net_params['bias%d'%i]
31         w = net_params['weight%d'%i]
32         act_f = net_params['act%d'%i]
33         A[i] = np.dot(w, Z[i-1]) + b      # A: M x N
34         Z[i] = act_f(A[i])                # Z: M x N
35     # end forward pass
36     Yp = Z[i]
37
38     # (2) Calculate output dE/da
39     delta[last_layer] = Yp - trainY      # delta: M x N
40
41     # (3) Backpropagate to calculate dE/da for layer i-1
42     for i in range(last_layer, 1, -1):
43         b = net_params['bias%d'%i]          # b: Mnnext x 1
44         w = net_params['weight%d'%i]          # w: Mnnext x M
45         act_f = net_params['act%d'%(i-1)]
46
47         sumdw = np.dot(w.transpose(), delta[i])  # M x N

```

```

48         assert act_f == sigmoid
49         delta[i - 1] = dsigmoid(Z[i - 1]) * sumdw # M x N
50
51     # (4) Calculate gradient dE/dw and dE/db
52     dEw[i] = np.dot(delta[i], Z[i-1].transpose()) #Mnxt,M
53     dEb[i] = np.dot(delta[i], np.ones((N, 1)))      #Mnxt,1
54 # end backpropagate
55
56     # Calculate gradient dE/dw and dE/db: dE ~ d sum En
57     dEw[1] = np.dot(delta[1], Z[0].transpose()) # M1 x M0
58     dEb[1] = np.dot(delta[1], np.ones((N, 1))) # M1 x 1
59
60     # Update parameters w/ Gradient Descent
61     for i in range(1, num_layers):
62         b = net_params['bias%d'%i]
63         w = net_params['weight%d'%i]
64
65         b -= step_size * dEb[i]
66         w -= step_size * dEw[i]
67     # end update parameters
68
69     # Calculate loss at each epoch
70     lossn = np.sum(loss(Yp, trainY), axis=0)
71     train_losses.append(np.mean(lossn)) # Loss = MSE
72 # end epoch nt
73
74 return net_params, train_losses

```

---

รายการ 3.8 แสดงโปรแกรมคำนวณฟังก์ชันซิกมอยด์และอนุพันธ์. สังเกตว่า อนุพันธ์ของซิกมอยด์ รับอาร์กิวเม้นต์ที่เป็นผลลัพธ์จากการกระตุ้นแล้ว  $Z$  ไม่ใช่ผลรวมการกระตุ้น  $A$ . (ดูแบบฝึกหัด 3.2 และอภิปรายถึงข้อดีข้อเสียของการเขียนโปรแกรมให้รับอาร์กิวเม้นต์เป็น  $Z$  เปรียบเทียบกับการเขียนโปรแกรมให้รับอาร์กิวเม้นต์เป็น  $A$ . คำໃບ โปรแกรมที่มีประสิทธิภาพ ควรลดการคำนวณที่ซ้ำซ้อน.)

รายการ 3.9 แสดงโปรแกรมคำนวณฟังก์ชันเอกลักษณ์ ซึ่งทำหน้าที่ เป็นจุดอ้างอิง เพื่อให้โปรแกรม `mlp` และ `train_mlp` รวมถึงการปรับแต่งโครงข่ายประสาทเทียม ผ่านไฟรอนดิกชันนารี `net_params` ทำได้สะดวก และยืดหยุ่น.

---

รายการ 3.8: โปรแกรมฟังก์ชันซิกมอยด์และอนุพันธ์

---

1 `def sigmoid(a):`

```

2     return 1/(1 + np.exp(-a))
3
4 def dsigmoid(z): # Caution!: argument is z, not a!
5     return z * (1 - z)

```

---

รายการ 3.9: โปรแกรมฟังก์ชันเอกลักษณ์

```

1 def identity(a):
2     return a

```

---

รายการ 3.10 แสดงโปรแกรมกำหนดค่าหน้าหนักเริ่มต้นด้วยการสุ่ม. โปรแกรม `w_initn` สุ่มค่าไปอัลส์ และค่าหน้าหนัก จากการแจกแจงเกาส์เซียน ซึ่งค่าดีฟอลต์คือค่าเฉลี่ยเป็น 0 และค่าเบี่ยงเบนมาตรฐานเป็น 1. โปรแกรมรับอาร์กิวเมนต์ `Ms` เป็นลิสต์ของเลขที่ระบุจำนวนโหนดในแต่ละชั้น โดยเริ่มจากจำนวนโหนดในชั้นอินพุต (ซึ่งคือจำนวนมิติของอินพุต) และตามด้วยจำนวนโหนดในชั้นจำนวนที่หนึ่ง จนถึงจำนวนโหนดในชั้นเออต์พุต (ซึ่งเท่ากับจำนวนมิติของเออต์พุตสุดท้าย). โปรแกรมรีเทิร์นไฟรอนดิกชันนารี ในรูปแบบของโครงข่ายประสาทเทียม ที่จะสามารถใช้ได้กับ `mlp` (รายการ 3.4) และ `train_mlp` (รายการ 3.7) ถ้าเพิ่มค่าของฟังก์ชันกระตุนเข้าไป.

รายการ 3.10: โปรแกรมกำหนดค่าหน้าหนักเริ่มต้นด้วยการสุ่ม

```

1 def w_initn(Ms, umeansigma=(0,1)):
2     assert len(Ms) >= 2, 'Ms: list of units in each layer'
3
4     num_layers = len(Ms)
5     params = {'layers': num_layers}
6     mu = umeansigma[0]
7     sigma = umeansigma[1]
8     for i, m in enumerate(Ms[1:], start=1):
9         mprev = Ms[i-1]
10        b = np.random.randn(m,1)
11        w = np.random.randn(m, mprev)
12        params['bias%d'%i] = b*sigma + mu
13        params['weight%d'%i] = w*sigma + mu
14
15    return params

```

---

รายการ 3.11 แสดงโปรแกรมคำนวณค่าผิดพลาดกำลังสอง ซึ่งใช้ช่วยประเมินการฝึก.

รายการ 3.11: โปรแกรมคำนวณค่าผิดพลาดกำลังสอง

```

1 def sqr_error(yhat, y):

```

```
2     assert yhat.shape == y.shape
3     return (yhat - y)**2 # output: K x N
```

จากโปรแกรมต่าง ๆ ที่มี ตัวอย่างต่อไปนี้แสดง การสร้าง การฝึก และใช้แบบจำลองที่ฝึกแล้วในการทำนาย. สมมติ ข้อมูลจำนวน 200 จุดข้อมูล ทั้งตัวแปรต้นและตัวแปรตามมีนิติเดียว ได้มาดังนี้

```
x = np.linspace(0, 1, 200)
noise = np.random.rand(200)
y = x + 0.3 * np.sin(2 * np.pi * x) + 0.1 * noise
x, y = x.reshape((1, -1)), y.reshape((1, -1))
```

สมมติต้องการใช้โครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยซ่อน โครงข่ายประสาทเทียมสามารถถูกสร้าง และฝึกได้ดังนี้

```
net = w_initn([1, 8, 1])
net['act1'] = sigmoid
net['act2'] = identity
tnet, losses = train_mlp(net, x, y, sqr_error, lr=0.3, epochs=40000)
```

เมื่อ เลือกใช้อัตราเรียนรู้ 0.3 และทำการฝึก 40000 สมัย. ผลลัพธ์ที่ได้คือ แบบจำลองที่ฝึกแล้ว **tnet** และความก้าวหน้าในการฝึก **losses**. การฝึกทุกครั้งควรตรวจสอบความก้าวหน้าในการฝึก ว่าดำเนินไปได้ด้วยดี เช่น ทำ **plt.plot(losses)** เพื่อดูว่ากราฟลู่ลงจนราบดีแล้ว. หลังจากฝึกเสร็จแล้ว แบบจำลอง **tnet** สามารถนำไปใช้ทำนายได้ เช่น **Yp = mlp(tnet, x)** เมื่อ **x** เป็นตัวแปรต้นที่ถูก และผลลัพธ์การทำนายคือ **Yp**.

จะศึกษาโปรแกรมเหล่านี้ ทดลองสร้าง ทดลองฝึก และทดลองใช้แบบจำลองทำนาย รวมไปถึงประเมินผลการทำนาย เช่น ลองวัดค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง **np.sqrt(np.mean(sqr\_error(Yp, y)))** และลองวัดผลการทำนาย เปรียบเทียบกับข้อมูล เช่น **plt.plot(x[0], y[0], 'r\*', label='Ground Truth')** **plt.plot(x[0], Yp[0], 'go', label='ANN')**

อภิราย และสรุปสิ่งที่ได้เรียนรู้.

### แบบฝึกหัด 3.12

การทำหนดค่าหน้าหนักเริ่มต้น มีผลอย่างมากต่อการฝึกโครงข่ายประสาทเทียม. จะสร้างข้อมูล (ดูแบบฝึกหัด 3.11) แบ่งข้อมูลออกเป็นข้อมูลฝึก และข้อมูลทดสอบ. จากนั้น เลือกโครงข่ายประสาทเทียมที่เหมาะสม

สม แล้วทดลองฝึกโครงข่ายประสาทเทียม และวัดผลการทำงานกับข้อมูลทดสอบ. ทดลองซ้ำทั้งหมดไม่น้อยกว่า 40 ช้า ซึ่งในแต่ละช้า ทำการกำหนดค่าน้ำหนักเริ่มต้นใหม่ทุกรังส์ นอกจากนี้ให้ทำอย่างอื่นเหมือนเดิม. การกำหนดค่าน้ำหนักเริ่มต้น ให้ใช้วิธีการสุ่ม (รายการ 3.10) สังเกตผลการทำงานจากการทดลองซ้ำ อภิปรายผลและการทดลองเช่นนี้อีก แต่ใช้วิธีเหจียนวิดโดยร์ว (รายการ 3.12) ในการกำหนดค่าน้ำหนักเริ่มต้น เปรียบเทียบผลที่ได้ อภิปราย และสรุปสิ่งที่ได้เรียนรู้.

หมายเหตุ การแบ่งข้อมูล อาจทำได้ดังนี้ เมื่อ  $X$  และ  $y$  เป็นตัวแปรต้นและตัวแปรตามของข้อมูล ซึ่งมีจำนวน  $N$  จุดข้อมูล. ในตัวอย่างแบ่งข้อมูล โดยแบ่งให้ประมาณ 60% ของข้อมูลทั้งหมดใช้เป็นข้อมูลฝึก (**trainx** และ **trainy**) และที่เหลือเป็นข้อมูลทดสอบ (**testx** และ **testy**).

```
_, N = x.shape
ids = np.random.choice(N, N, replace=False)
train_size = 0.6
mark = round(train_size*N)
train_ids = ids[:mark]
test_ids = ids[mark:]
trainx, trainy = x[:, train_ids], y[:, train_ids]
testx, testy = x[:, test_ids], y[:, test_ids]
```

หากแบ่งข้อมูลดังคำสั่งข้างต้นแล้ว การฝึกแบบจำลอง การใช้แบบจำลองท่านายข้อมูลทดสอบ และวัดผลการทดสอบด้วยค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง (**rmse**) สามารถทำได้ดังเช่น

```
tnet, losses = train_mlp(net, trainx, trainy, sqr_error, 0.3, 40000)
Yp = mlp(trained_net, testx)
rmse = np.sqrt(np.mean(sqr_error(Yp, testy)))
```

รายการ 3.12: โปรแกรมกำหนดค่าน้ำหนักเริ่มต้นตามแนวทางเหจียนวิดโดยร์ว [141]

---

```
1 def w_initngw(Ms, maxoff=(1.4, 0)):
2     assert len(Ms) >= 2
3     num_layers = len(Ms)
4     params = {'layers': num_layers}
5     scale = maxoff[0]
6     offset = maxoff[1]
7     for i, m in enumerate(Ms[1:], start=1):
8         mprev = Ms[i-1]
9         wmax = scale*m**(1/mprev)
10        wi_ = np.random.rand(m, mprev) * 2 - 1
11        denomi = np.sqrt(np.sum(wi_***2, axis=1).reshape((-1,1)))
```

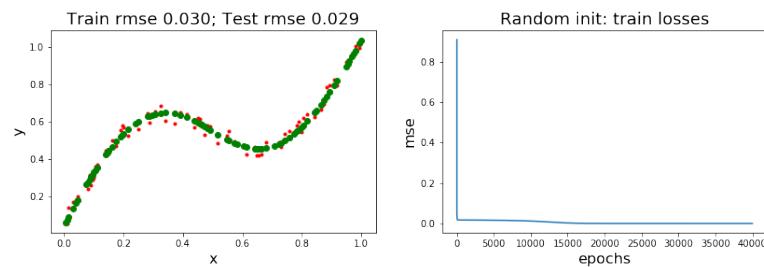
```

12     wi = wmax * wi_ /denomi
13     bi = wmax * np.linspace(-1, 1, m) \
14             * np.sign(np.random.rand(m)) + offset
15     params['bias%d'%i] = bi.reshape((m,1))
16     params['weight%d'%i] = wi
17
18     return params

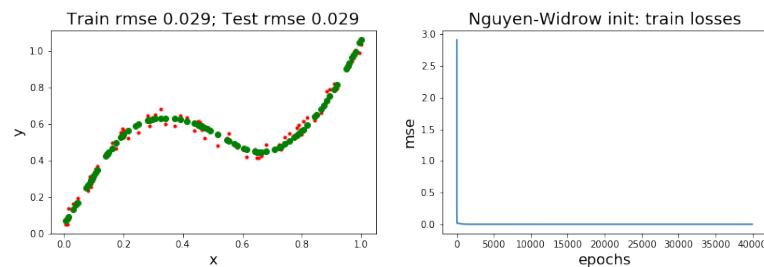
```

---

รูป 3.25 และ 3.26 แสดงตัวอย่างผลจากการทดลอง ซึ่งเลือกผลที่ดีที่สุด (ค่าผิดพลาดทดสอบต่ำสุด) จากการทดลองซ้ำ 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยวิธีเรียงวิดโดยร์ ตามที่ระบุในคำบรรยายรูป.

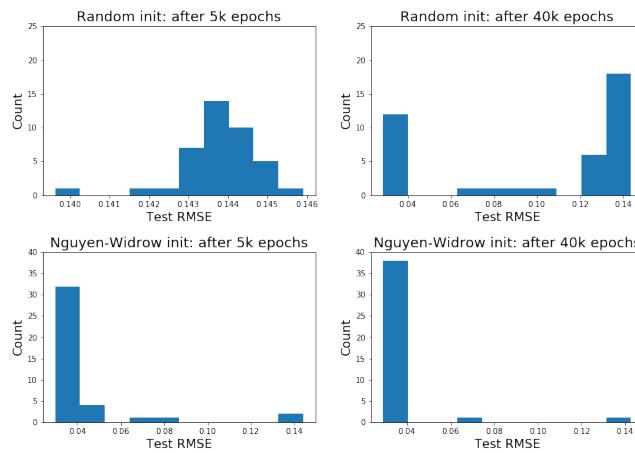


รูปที่ 3.25: ตัวอย่างผลการทำนายที่ดีที่สุด จากการทดลอง 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยการสุ่ม.



รูปที่ 3.26: ตัวอย่างผลการทำนายที่ดีที่สุด จากการทดลอง 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยวิธีเรียงวิดโดยร์.

รูป 3.27 และ 3.28 แสดงตัวอย่างวิธีการนำเสนอผลศึกษา. ตาราง 3.5 แสดงตัวอย่างค่าสถิติจากการศึกษา. สังเกตจากผลในตัวอย่าง ผลการทำงานของแบบจำลองที่ฝึกแล้ว เมื่อใช้การสุ่มกำหนดค่าเริ่มต้น จะมีความหลากหลายค่อนข้างมาก. เมื่อเปรียบเทียบกับวิธีเรียงวิดโดยร์ (1) ผลลัพธ์ที่ดีที่สุด เมื่อกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม หากฝึกนานพอ ไม่ได้ต่างจาก ผลลัพธ์ที่ดีที่สุด เมื่อกำหนดค่าเริ่มต้นด้วยวิธีเรียงวิดโดยร์. แต่ (2) การกำหนดค่าเริ่มต้นด้วยวิธีเรียงวิดโดยร์ สามารถช่วยให้ได้แบบจำลองที่ดี โดยใช้จำนวนสมัยฝึกที่น้อยกว่า การกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม. (3) ผลลัพธ์โดยเฉลี่ย ของการกำหนดค่าเริ่มต้นด้วยวิธีเรียงวิดโดยร์ ดี



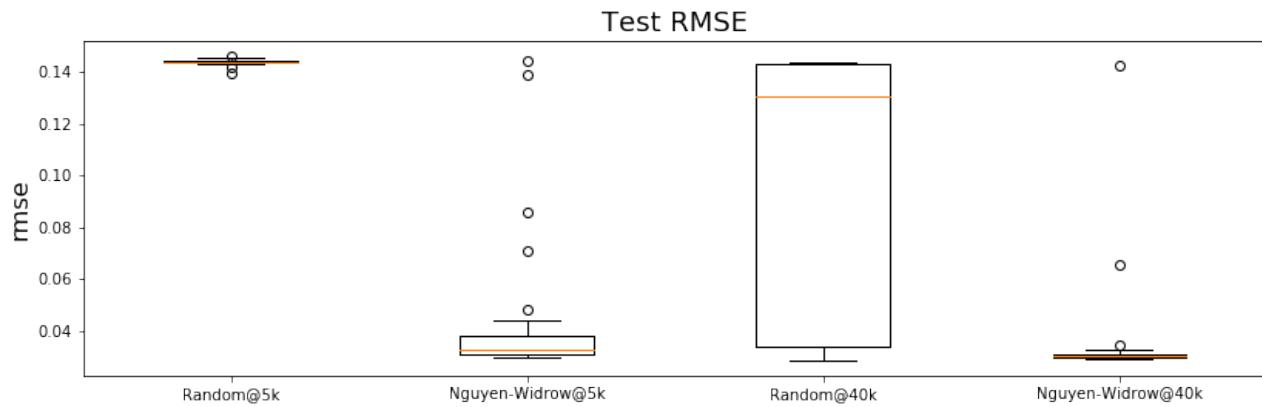
รูปที่ 3.27: ผลการทำนาย ที่ได้จากการทดลองข้า 40 ครั้ง เมื่อใช้วิธีกำหนดค่าน้ำหนักเริ่มต้นแบบต่าง ๆ. สองภาพในแถวบน แสดง อิสโทแกรมของผลที่ได้ เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยวิธีการสุ่ม. สองภาพในแถวล่าง แสดงอิสโทแกรมของผลที่ได้ เมื่อกำหนด ค่าน้ำหนักเริ่มต้น ด้วยวิธีเหจี้ยนวิดโดร์. ภาพทางซ้าย แสดงผลหลังจากการฝึกไป 5000 สมัย. ภาพทางขวา แสดงผลหลังจากการ ฝึกไป 40000 สมัย.

กว่า ผลลัพธ์โดยเฉลี่ย เมื่อกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม. (4) โอกาสที่จะได้แบบจำลองที่ดี เมื่อกำหนดค่าเริ่มต้น ด้วยวิธีเหจี้ยนวิดโดร์ จะสูงกว่าเมื่อกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม ถึงสองเท่าครึ่ง หากฝึกนานพอ.

ที่ 40000 สมัย นับค่าผิดพลาดน้อยกว่า 0.04 ได้ 12 ครั้ง จาก 40 ครั้ง หรือคิดเป็น มีโอกาสประมาณ 30% เมื่อใช้วิธีสุ่ม และฝึกนานพอ เปรียบเทียบกับ ประมาณ 95% (นับได้ 38 ครั้ง) เมื่อใช้วิธีเหจี้ยนวิดโดร์. แต่ที่ 5000 สมัย ค่าผิดพลาดที่ต่ำกว่า 0.04 ไม่มีเลย เมื่อใช้วิธีสุ่ม เปรียบเทียบกับ มีโอกาสประมาณ 80% (นับได้ 32 ครั้ง) เมื่อใช้วิธีเหจี้ยนวิดโดร์. เปรียบเทียบผลที่ได้ทำการทดลองเอง กับผลตัวอย่างที่นำเสนอใน รูป 3.27 และตาราง 3.5 อภิปรายผลที่ได้ กับข้อสังเกตที่ตั้งไว้นี้ รวมถึงอภิปรายถึงแนวทางปฏิบัติ เมื่อทำแบบ จำลองโครงข่ายประสาทเทียม.

ตารางที่ 3.5: ค่าสถิติของผลการฝึกแบบจำลอง จากการทดลองข้า 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีการสุ่ม และด้วยวิธีเหจี้ยนวิดโดร์.

การกำหนดค่าน้ำหนักเริ่มต้น	ค่าสถิติ	ผลจากแบบจำลองที่ผ่านการฝึก	
		5000 สมัย	40000 สมัย
วิธีสุ่ม	ค่าน้อยที่สุด	0.140	0.029
	ค่าเฉลี่ย	0.144	0.101
	ค่ามากที่สุด	0.146	0.144
วิธีเหจี้ยนวิดโดร์	ค่าน้อยที่สุด	0.030	0.029
	ค่าเฉลี่ย	0.041	0.034
	ค่ามากที่สุด	0.144	0.143



รูปที่ 3.28: แผนภูมิกล่อง แสดงผลการทำนาย ที่ได้จากการทดลองซ้ำ 40 ครั้ง สำหรับกรณีต่าง ๆ ดังระบุในแกนนอน ซึ่ง Random หมายถึง เมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยการสุ่ม. Nguyen-Widrow หมายถึง เมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีเหยี่ยวนิด โดยร์. คำต่อท้าย @5k หมายถึง วัดผลของแบบจำลองที่ได้ทำการฝึกไป 5000 สมัย. คำต่อท้าย @40k หมายถึง วัดผลของแบบจำลองที่ได้ทำการฝึกไป 40000 สมัย.

**ความสำคัญของการทำข้าม.** เกี่ยวกับการทำข้าม การทำแบบจำลองด้วยโครงข่ายประสาทเทียม นิยมทำข้าม หลายครั้ง (หากทรัพยากรอ่อนไหว) เนื่องจาก ผลค่อนข้างจะหลากหลาย ดังที่เห็นจากรูป 3.27 โดยเฉพาะอย่าง ยิ่งเมื่อใช้วิธีการกำหนดค่าน้ำหนักเริ่มต้นด้วยการสุ่ม. รูป 3.29 แสดงความสำคัญของการทำข้าม ซึ่งช่วยให้มี โอกาสสามารถ ที่จะพบแบบจำลองที่ดี หรืออย่างน้อย ก็ช่วยให้ได้เห็นความสามารถโดยเฉลี่ยของแบบจำลอง และการฝึกที่ใช้.

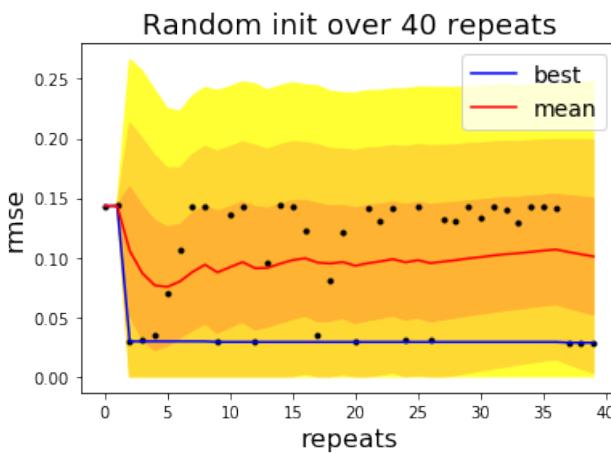
โปรแกรมข้างล่างนี้ เป็นตัวอย่างคำสั่งส่วนหนึ่ง (ไม่ใช่ทั้งหมด) ที่ใช้สร้างรูป 3.29.

```
cave = []
cstd = []
cbest = []
for i in range(40):
    cave.append(np.mean(trmses[::(i+1)]))
    cstd.append(np.std(trmses[::(i+1)]))
    cbest.append(np.min(trmses[::(i+1)]))
cave = np.array(cave)
cstd = np.array(cstd)
cbest = np.array(cbest)

ci_u = cave + cstd
ci_l = cave - cstd
ci_xs = np.hstack( (ci_u, ci_l[::-1]) )
plt.fill(ci_xs, ci_ys, color=(1, 0.7, 0.2))
plt.plot(trmses, 'k.')
plt.plot(cbest, color=(0, 0, 1), label='best')
```

```
plt.plot(cave, 'r-', label='mean')
```

เมื่อ `trmses` แทนไฟรอนลิสต์ที่เก็บค่าผิดพลาดทดสอบของแต่ละชั้นไว้ ทั้งหมด 40 ชั้น หมายเหตุ ค่าสถิติที่ใช้ทำ แถบความเชื่อมั่น (confidence intervals) ในรูป คือ  $\mu \pm \sigma$  (แสดงในโปรแกรมตัวอย่างข้างต้น) และ  $\mu \pm 2\sigma$  กับ  $\mu \pm 3\sigma$  (ไม่ได้แสดงในโปรแกรมตัวอย่าง). สังเกตว่า แนวทางปฏิบัติ ที่ทำสำหรับ ฯ ครั้ง และเลือกแบบจำลองที่ทำงานได้ดีที่สุด นั้น เป็นเสมือนการหาค่าดีที่สุดอ่อน ๆ (soft optimization) ที่ช่วยบรรเทา การติดกับสถานการณ์ที่ดีที่สุดท้องถิ่น ของขั้นตอนวิธีหากค่าดีที่สุดลงได้บ้าง.



รูปที่ 3.29: ค่าผิดพลาดทดสอบ จากการทำซ้ำ 40 ครั้ง. จุดสีดำ แสดงค่าผิดพลาดทดสอบ ของแต่ละชั้น. ตำแหน่งตามแกนนอนของจุดสีดำ คือดัชนีของการทดลองชั้น. ในขณะที่ ผลของแต่ละชั้นก็แตกต่างกันไป แต่ค่าเฉลี่ยของผลลัพธ์ที่ได้ ที่จำนวนชั้นต่าง ๆ ที่แสดงด้วยเส้นสีแดง จะค่อนข้างนิ่ง เมื่อจำนวนชั้นมากขึ้น. แกนนอน แสดงจำนวนชั้น. แถบสีส้มแก่ ส้มอ่อน และเหลืองแสดงแถบความเชื่อมั่น ของผลที่จะได้ หรือ การประเมินโอกาสของผลที่จะได้ โดยไม่จำกัดความน่าจะเป็น ตามลำดับ. แถบความเชื่อมั่น ประเมินคร่าว ๆ จากค่าสถิติ ที่ได้จำนวนจากผลลัพธ์ของแต่ละชั้น. เส้นสีน้ำเงิน แสดงค่าดีที่สุด ที่ได้จากการทำซ้ำ ที่จำนวนชั้นต่าง ๆ.

### แบบฝึกหัด 3.13

จากรูป 3.19 สร้างข้อมูลชุดบี จากความสัมพันธ์  $y = 0.1(x - 100) + 0.3 \sin(0.2\pi(x - 100)) + \epsilon$  เมื่อ  $\epsilon \sim \mathcal{U}(0, 1)$ . สัญกรณ์  $\epsilon \sim \mathcal{U}(0, 1)$  หมายถึง ค่าของ  $\epsilon$  สุ่มจากการแจกแจงเอกภูมิ. สร้างข้อมูล ขึ้นมา 500 จุดข้อมูลสำหรับการฝึก และ 250 จุดข้อมูลสำหรับการทดสอบ. ให้  $x$  อยู่ในช่วง 100 ถึง 110. ทดลองใช้โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยชั้non 8 หน่วย เลือกอภิธานพารามิเตอร์ เพื่อให้การฝึกสามารถทำงานได้ดี. เปรียบเทียบ (ก) การไม่ทำนอร์มอลайเซอร์ กับ (ข) การทำนอร์มอลายเซอร์ให้อยู่ในช่วง  $[0, 1]$  และ (ค) การทำนอร์มอลายเซอร์ให้อยู่ในช่วง  $[-1, 1]$  และ (ง) การทำนอร์มอลายเซอร์ให้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน เป็น 0 และ 1 ตามลำดับ. เปรียบเทียบ อภิประยผล และสรุป พื้นที่ที่เปลี่ยนโปรแกรมเพื่อวัดกราฟนำเสนอผลสรุป.

คำใบ้ ดูคำสั่ง `np.random.rand`. การทดลองกรณีง่ายก่อน จะช่วยให้การเลือกค่าอภิมานพารามิเตอร์สะดวกขึ้น. กรณีง่าย หมายถึง กรณีที่น่าจะช่วยให้การฝึกแบบจำลองทำได้ง่ายกว่า.

หมายเหตุ รายการ 3.13 แสดงโปรแกรมสำหรับการน้อมอิเล็กทรอนิกส์แบบกำหนดช่วง. ศึกษาโปรแกรมทดลองใช้ อภิปราย และเขียนโปรแกรมสำหรับการน้อมอิเล็กทรอนิกส์แบบค่าสถิติ ทดสอบ และนำโปรแกรมน้อมอิเล็กทรอนิกส์ทั้งสองแบบ ไปทำการทดลอง.

รายการ 3.13: โปรแกรมน้อมอิเล็กทรอนิกส์

---

```

1 def normalize1(X, params=None):
2     # X: D x N
3     xmax = np.max(X, axis=1)
4     xmin = np.min(X, axis=1)
5
6     if params is not None:
7         xmax = params['xmax']
8         xmin = params['xmin']
9     else:
10        params = {'xmax': xmax, 'xmin': xmin}
11
12    xmax = xmax.reshape((-1, 1))
13    xmin = xmin.reshape((-1, 1))
14    xn = (X - xmin)/(xmax - xmin)
15
16    return xn, params

```

---

### แบบฝึกหัด 3.14

ชุดข้อมูลเรือยอชต์ (yacht dataset) จากคลังข้อมูลยูซีไอ[8] ซึ่งดาวน์โหลดที่ <http://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics> มีภารกิจ คือการประมาณค่าแรงต้านที่เหลือค้าง (residuary resistance) ของเรือยอชต์ขณะแล่น ซึ่งเป็นปัญหาการหาค่าทดแทน. ข้อมูลชุดนี้ มี 308 ระเบียน นั่นคือ มี 308 จุดข้อมูล และแต่ละจุดข้อมูลจะมี 7 เขตข้อมูล (7 ลักษณะสำคัญ). ตามมุมมองของฐานข้อมูล (database) จุดข้อมูล จะเรียก ระเบียน (record) และคุณลักษณะสำคัญต่าง ๆ ในระเบียน จะเรียก เขตข้อมูล (field).

ชุดข้อมูลเรือยอชต์นี้ เขตข้อมูลที่หนึ่ง ตำแหน่งตามแนวยาวเรือของศูนย์กลางการลอยตัว (longitudinal position of the center of buoyancy) เป็นลักษณะที่ช่วยอธิบายการกระจายน้ำหนักของเรือตามแนวยาว ว่า น้ำหนักกระจายอยู่หน้าลำ กลางลำ หรือท้ายลำอย่างไร. เขตข้อมูลที่สอง ค่าสัมประสิทธิ์ปริซึม (prismatic

coefficient) เป็นลักษณะที่ช่วยอธิบายรูปทรงของห้องเรือ โดยวัดจากอัตราส่วนของปริมาตรที่อยู่ใต้น้ำของห้องเรือ เปรียบเทียบกับปริมาตรของรูปทรงปริซึมที่มีความยาวเท่ากัน และพื้นที่หน้าตัดเท่ากับ พื้นที่หน้าตัดที่กว้างที่สุดของห้องเรือ. เขตข้อมูลที่สาม อัตราส่วนความยาวเรือกับการกระจัด (length-displacement ratio) เป็นลักษณะที่ช่วยบ่งชี้ถึงความหนักของเรือ เมื่อเทียบกับความยาว เรือที่หนักจะมีค่าน้ำมาก เรือที่เบาจะมีค่าน้ำน้อย. เขตข้อมูลที่สี่ อัตราส่วนความกว้างเรือกับระดับจมน้ำ (beam-draught ratio) เป็นลักษณะทรงท้องเรือที่วัดจาก อัตราส่วนความกว้างเรือ กับความกว้างของส่วนที่กว้างที่สุดของเรือในแนวระดับน้ำ. เขตข้อมูลที่ห้า อัตราส่วนความยาวกับความกว้างเรือ (length-beam ratio). เขตข้อมูลที่หก ตัวเลขฟรูด (Froude number) เป็นค่าที่บอกความต้านทานของการที่วัตถุเคลื่อนที่ในน้ำ. เขตข้อมูลทั้งหกนี้ บรรยายลักษณะรูปร่าง และการกระจายน้ำหนักของห้องเรือ และเขตข้อมูลเหล่านี้จะใช้เป็นอินพุตของแบบจำลอง.

เขตข้อมูลที่เจ็ด (Residuary resistance per unit weight of displacement) เป็นค่าความต้านทานเหลือค้าง ซึ่งเป็นแรงต้านสำคัญที่เกิดกับเรือ และเกี่ยวพันกับลักษณะต่าง ๆ ของเรือ (ที่บรรยายด้วยหากเขตข้อมูลแรก). การประมาณค่าความต้านทานเหลือค้างได้แม่นยำ จะช่วยในการประเมินสมณฑะของเรือได้ดีรวมถึงช่วยเป็นข้อมูลประกอบ สำหรับการออกแบบเรือด้วย เช่น การเลือกรูปทรงและขนาดของห้องเรือ การกำหนดน้ำหนักบรรทุกของเรือ การเลือกขนาดของเครื่องยนต์ที่เหมาะสม. (ดูคำอธิบายเพิ่มเติมจากคลังข้อมูลยูชีไอ และอรทิโกสາและคณะ[144].)

ข้อมูลมี 308 ตัวอย่าง และสามารถอ่านข้อมูลเข้าได้ดังเช่นไฟล์ข้อมูลทั่วไป ดังแสดงในโปรแกรมตัวอย่างข้างล่าง เมื่อไฟล์ข้อมูลถูกดาวน์โหลดมาในชื่อ `yacht_hydrodynamics.data`.

```
with open('yacht_hydrodynamics.data', 'r') as f:
    yacht = f.read()
```

ข้อมูลที่นำมาจะอยู่ในรูปข้อความ เพื่อประสิทธิภาพในการประมวลผล ควรจัดข้อมูลเข้าในรูปนัมไฟล์เรียก ก่อน ดังแสดงด้วยคำสั่งข้างล่าง

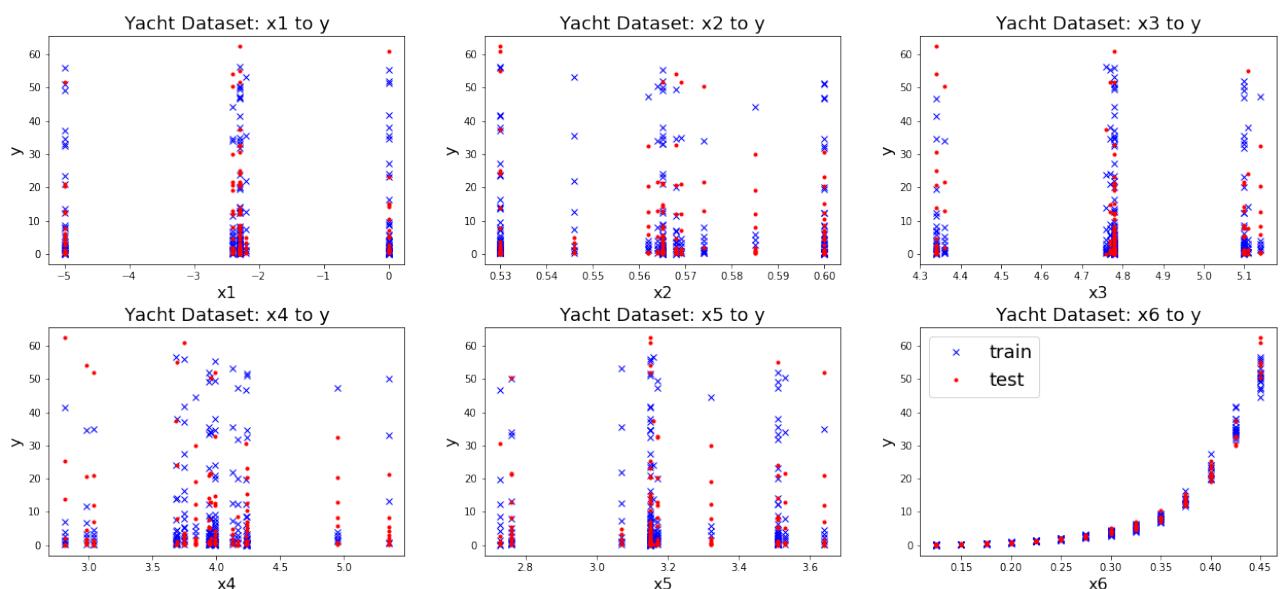
```
dataxy = []
lines = yacht.split('\n')
i = 0
for line in lines:
    i += 1
    row = []
    j = 0
    flag = False
    for d in line.split(' '):
```

```

j += 1
try:
    c = float(d)
    row.append(c)
except:
    flag = True
# end for d
if flag: print(i, ',', j, ';', d, ':', row)
if len(row) > 0: dataxy.append(row)
# end for line
Dataxy = np.array(dataxy)

```

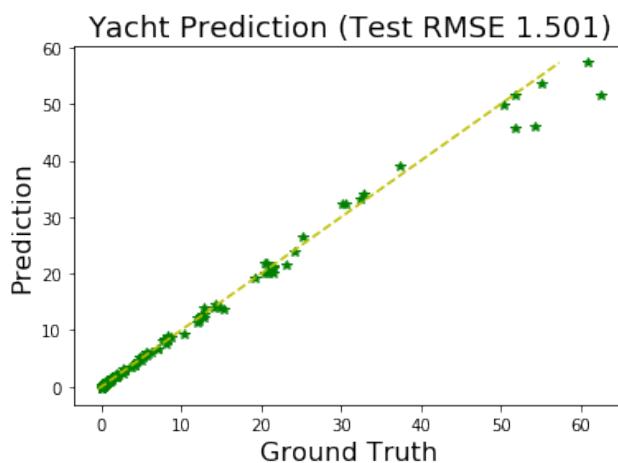
เมื่อ `Dataxy` คือข้อมูลในรูปแบบ `np.array` ขนาด  $308 \times 7$ . รูป 3.30 แสดงความสัมพันธ์ระหว่างเขตข้อมูลที่หนึ่งถึงหก (ที่ละเอียดข้อมูล) กับเขตข้อมูลที่เจ็ด (ที่เป็นตัวแปรตาม) หลังจากแบ่งข้อมูลแล้ว. ในตัวอย่างนี้ ข้อมูล 185 จุดข้อมูล (ประมาณ 60%) ถูกแบ่งเป็นชุดฝึก (`trainx` และ `trainy`) และที่เหลือเป็นชุดทดสอบ (`testx` และ `testy`).



รูปที่ 3.30: ชุดข้อมูลเรือยอร์ช หลังจากแบ่งข้อมูลแล้ว. แต่ละภาพแสดงความสัมพันธ์ของแต่ละเขตข้อมูลของตัวแปรต้น กับตัวแปรตาม. ภาพบทสینเนอร์เงิน แทนจุดข้อมูลที่ถูกแบ่งไปชุดฝึก. จุดสีแดง แทนจุดข้อมูลที่ถูกแบ่งไปชุดทดสอบ.

จากรูป 3.30 สังเกตว่า อินพุตมิติต่าง ๆ มีขนาดแตกต่างกันพอสมควร เช่น  $x_1$  (ภาพซ้ายบน) มีขนาดตั้งแต่  $-5$  ถึง  $0$  ในขณะที่  $x_6$  (ภาพขวาล่าง) มีขนาดไม่เกิน  $0.5$ . เพื่อช่วยให้การฝึกแบบจำลองทำได้ง่ายขึ้น สถานการณ์เช่นนี้ ควรนอร์มอลайเซอินพุต (หัวข้อ 3.4).

จงสร้างแบบจำลองโครงข่ายประสาทเทียม เพื่อประมาณค่าความต้านทานเหลือค้าง จากคุณลักษณะทั้งหมดของเรือ โดยใช้ชุดข้อมูลเรือยacht ในกรณีก และการทดสอบ โดยแบ่งข้อมูลให้เรียบร้อย ทำนอร์มอลайซ์อนพุต ภูมิป่าฯว่า กรณีนี้การนอร์มอลายซ์นิดๆ (ระหว่างนอร์มอลายซ์เข้าสู่ช่วงที่จำกัด กับนอร์มอลายซ์เข้าสู่ค่าสถิติที่กำหนด) เหมาะสมมากกว่ากัน พร้อมเหตุผลประกอบ. หลังจากฝึกและทดสอบเสร็จ นำเสนอผลให้ดูเจน. รูป 3.31 แสดงตัวอย่างการนำเสนอผล.



```

7      delta = {}
8      dEw = {}
9      dEb = {}
10     train_losses = []
11     step_size = lr/N
12     for nt in range(epochs):
13         for n in range(N):      # ONLINE! through each datapoint
14             Z = {0: trainX[:,[n]]}                      # ONLINE!
15             # (1) Forward pass
16             for i in range(1, num_layers):
17                 b = net_params['bias%d'%i]
18                 w = net_params['weight%d'%i]
19                 act_f = net_params['act%d'%i]
20                 A[i] = np.dot(w, Z[i-1]) + b
21                 Z[i] = act_f(A[i])
22             # end forward pass
23             Yp = Z[i]
24             # (2) Calculate output dE/da
25             delta[last_layer] = Yp - trainY[:,[n]]    # ONLINE!
26             # (3) Backpropagate
27             for i in range(last_layer, 1, -1):
28                 b = net_params['bias%d'%i]
29                 w = net_params['weight%d'%i]
30                 act_f = net_params['act%d'%(i-1)]
31                 sumdw = np.dot(w.transpose(), delta[i])
32                 assert act_f == sigmoid
33                 delta[i - 1] = dsigmoid(Z[i - 1]) * sumdw
34             # (4) Calculate gradient dE/dw and dE/db
35             dEw[i] = np.dot(delta[i], Z[i-1].transpose())
36             dEb[i] = delta[i]                           # ONLINE!
37             # end backpropagate
38             # Calculate gradient
39             dEw[1] = np.dot(delta[1], Z[0].transpose())
40             dEb[1] = delta[1]                           # ONLINE!
41             # Update parameters
42             for i in range(1, num_layers):
43                 b = net_params['bias%d'%i]
44                 w = net_params['weight%d'%i]
45                 b -= step_size * dEb[i]
46                 w -= step_size * dEw[i]
47             # end update parameters

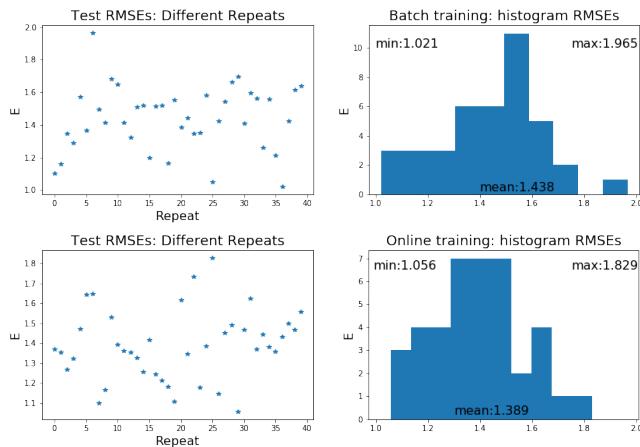
```

```

48 # end ONLINE through each datapoint
49 # Calculate loss at each epoch
50 Yp = mlp(net_params, trainX)
51 lossn = np.sum(loss(Yp, trainY), axis=0)
52 train_losses.append(np.mean(lossn))
53 # end epoch nt
54 return net_params, train_losses

```

---



รูปที่ 3.32: ผลความแม่นยำ ที่ได้จากการฝึกแบบหมุน 比べยับเทียบกับแบบออนไลน์ จากการทดลองซ้ำ 40 ครั้ง. ภาพในแถวบนแสดงผลเมื่อฝึกแบบหมุน. ภาพในแถวล่าง แสดงผลเมื่อฝึกแบบออนไลน์

ในกรณีนี้ ดังผลที่แสดงในรูป 3.32 ความแม่นยำที่ได้ ไม่ได้แตกต่างกันเท่าไร แต่เวลาในการฝึกต่างกันมาก เช่นในการทดลองตัวอย่าง การฝึกแบบออนไลน์ใช้เวลาเป็น 36 เท่าของเวลาฝึกแบบหมุน.

### แบบฝึกหัด 3.15

ข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ (Mammographic Mass dataset) จากคลังข้อมูลยูซีไอที และคณะ[62] ใช้ศึกษาการทำนายผลการตรวจภาพเอ็กซเรย์เต้านม ของร้องรอยมวลเนื้อ (mammographic mass lesion) ว่าเป็นเนื้อดี (benign) หรือเนื้อร้าย (malignant) จากค่าลักษณะสำคัญต่าง ๆ ของของภาพ ประกอบกับอายุของผู้ป่วย. วิธีตรวจภาพเอ็กซเรย์เต้านม (Mammography) เป็นวิธีที่มีประสิทธิผลมากในการตรวจมะเร็งทรวงอก [62]. ข้อมูลชุดนี้ประกอบด้วย ค่าการประเมินไบแรตส์ (BI-RADS assessment เป็นค่าเชิงเลขลำดับ ordinal values), อายุของผู้ป่วย (เลขจำนวนเต็ม), รูปทรงของมวลเนื้อ (mass shape ซึ่งถูกแทนด้วย 1 สำหรับทรงกลม round, 2 สำหรับทรงรี oval, 3 สำหรับทรงกลืนย่อย lobular, 4 สำหรับทรงที่ผิดแปลง irregular), ลักษณะขอบของมวลเนื้อ (mass margin ซึ่งถูกแทนด้วย 1 สำหรับเขต

รอบซัดเจน circumscribed, 2 สำหรับขอบเขตเป็นกลีบย่อย ๆ microlobulated, 3 สำหรับขอบเขตคลุมเครือ obscured, 4 สำหรับขอบเขตยากจะระบุ ill-defined, 5 สำหรับขอบเขตเป็นลักษณะหนามหรือปุ่ม spiculated), ความหนาแน่นของมวลเนื้อ (mass density ซึ่งถูกแทนด้วย 1 สำหรับความหนาแน่นสูง high, 2 สำหรับความหนาแน่นกลาง medium, 3 สำหรับความหนาแน่นต่ำ low, 4 สำหรับมวลเนื้อมีไขมันอยู่ fat-containing) และความร้ายแรง (severity ซึ่งมีสองค่า 0 สำหรับเนื้อดี หรือ 1 สำหรับเนื้อร้าย). ค่าความร้ายแรง คือค่าเป้าหมาย ที่ต้องการทำนาย. การสามารถทำนายค่าความร้ายแรงได้อย่างแม่นยำจะช่วยให้แพทย์และผู้ป่วยสามารถตัดสินใจได้ดีขึ้นว่า ควรจะทำการตัดเนื้อจากบริเวณที่สงสัยออกตรวจเพื่อยืนยันผลหรือไม่.

ข้อมูลชุดนี้มี 961 ระเบียน เฉลยหรือผลการตรวจจริง ระบุ 516 ระเบียนที่ผลเป็นเนื้อดี (ค่าความร้ายแรง เป็นศูนย์) และ 445 ระเบียนที่ผลเป็นเนื้อร้าย (ค่าความร้ายแรง เป็นหนึ่ง). ข้อมูลชุดนี้มีค่าบางค่าของเขตข้อมูลที่ไม่ครบ (missing attribute values) ได้แก่ ค่าการประเมินไป-retscore ไป 2 ค่า, อายุชาดไป 5 ค่า, รูปทรงของมวลเนื้อชาดไป 31 ค่า, ลักษณะขอบของมวลเนื้อชาดไป 48 ค่า และความหนาแน่นของมวลเนื้อชาดไป 76 ค่า. ส่วนความร้ายแรงมีค่าครบทุกรอบเรียบ.

จงทำแบบจำลองโครงข่ายประสาทเทียมเพื่อทำนายค่าความร้ายแรง จากลักษณะสำคัญ ด้วยข้อมูลชุดภาพเอ็กซเรย์เต้านม เลือกแบบจำลอง ฝึก ทดสอบ วัดผล รายงานผลที่ได้ อภิปราย และสรุป. ภาระกิจนี้ เป็นการจำแนกค่าทวิภาค ที่เอาต์พุต คือความร้ายแรง ซึ่งมีค่าเป็นหนึ่งหรือศูนย์. การจัดการกับข้อมูลขาดหายสามารถดำเนินการได้หลายแนวทาง. แนวทางหนึ่ง ซึ่งเป็นแนวทางที่ง่ายที่สุด คือตัดระเบียนที่มีข้อมูลขาดหายทิ้งทั้งระเบียน.

คำสั่งข้างล่าง แสดงตัวอย่างการอ่านไฟล์ข้อมูล `mammographic_masses.data` และเตรียมข้อมูล

```
with open('mammographic_masses.data', 'r') as f:
    mammo_text = f.read()

mammo1 = []
for line in mammo_text.split('\n'):
    row = []
    complete = True
    for field in line.split(','):
        try:
            val = int(field)
        except:
            complete = False
    if complete:
        ammo1.append(row)
```

```

    val = None
    row.append(val)
if complete:
    mammo1.append(row)
else:
    mammo_miss.append(row)
# end for line
mammo1 = np.array(mammo1)

```

ซึ่งผลลัพธ์จะคือ `mammo1` ซึ่งเป็นนิมไพลอาร์เรย์ขนาด  $830 \times 6$  ซึ่งทุกแกร์ (ทุกระเบียนข้อมูล) ครบถ้วนสมบูรณ์ ไม่มีข้อมูลขาดหาย. ในขณะที่ `mammo_miss` เป็นลิสต์ของระเบียนข้อมูลที่เหลือ และทุกระเบียนใน `mammo_miss` มีข้อมูลที่ขาดหายไป (ดูย่อหน้า การจัดการกับข้อมูลขาดหาย).

ข้อมูลของ `mammo1` แสดงในรูป 3.33, 3.34 และ 3.35. สังเกตว่า ข้อมูลชุดภาพเอ็กซเรย์เต้านมนี้ อินพุตมิติแรก (รูป 3.33 และคำอธิบายข้อมูล ที่สามารถดาวน์โหลดได้พร้อมไฟล์ข้อมูล) เป็นค่าเชิงเลขลำดับซึ่งสามารถคิดเห็นว่าเป็นค่าตัวเลขได้. อินพุตมิติที่สอง เป็นจำนวนเต็มแสดงอายุ ซึ่งเป็นตัวเลขจริง ๆ. แต่ อินพุตมิติที่สามถึงห้า (รูป 3.35) เป็นค่าแทนชื่อ (nominal values) ซึ่งไม่ได้มีความหมายเป็นปริมาณตามขนาดใหญ่เล็กของตัวเลขจริง ๆ ตัวเลขท่าน้ำที่ แค่เป็นตัวชี้ที่อ้างถึงชื่อเท่านั้น.

รูป 3.36 แสดงผลลัพธ์ของการทำนาย ด้วยแผนภูมิกล่อง จากตัวอย่างผลการทำนายของ แบบจำลองที่ทำโดยไม่สนใจความหมายของอินพุต (กล่องซ้าย ระบุด้วยชื่อ **Naive**) และแบบจำลองที่ทำการเข้ารหัสอินพุต ให้เหมาะสมกับความหมาย (กล่องขวา ระบุด้วยชื่อ **Coding**). แบบจำลองที่ทำโดยไม่สนใจความหมายของ อินพุต นั่นคือ ใช้ค่าตัวเลขของข้อมูลใส่เป็นอินพุตให้กับแบบจำลองโดยตรง เช่น ในตัวอย่างข้างล่าง ที่เพียงทำ นอร์มอลайซ์อินพุต

```

trainx = mammo1[train_ids,:5].transpose()      # 5 x N
trainy = mammo1[train_ids,5].reshape((1,-1))    # 1 x N
testx = mammo1[test_ids,:5].transpose()         # 5 x N
testy = mammo1[test_ids,5].reshape((1,-1))       # 1 x N
trainxn, normpars = normalize1(trainx)
# Configure binary classification net
net = w_initn([5, 10, 1])
net['act1'] = sigmoid
net['act2'] = sigmoid
# Train net
trained_net, train_losses = train_mlp(net, trainxn, trainy,
                                         binaries_cross_entropy, lr=0.1, epochs=5000)

```

เมื่อ `train_ids` และ `test_ids` เป็นตัวชี้ที่สุ่มเลือก เพื่อใช้เป็นข้อมูลฝึก และข้อมูลทดสอบ ตามลำดับ. ตัวอย่างผลที่แสดงในแบบฝึกหัดนี้ ได้จากการใช้ข้อมูลจำนวน 300 จุดข้อมูลเป็นข้อมูลทดสอบ และใช้ที่ข้อมูลที่เหลือเป็นข้อมูลฝึก (ดูตัวอย่างวิธีแบ่งข้อมูล จากแบบฝึกหัด 3.12). โปรแกรม `normalize1`, `w_initn`, `sigmoid` และ `train_mlp` ได้อภิรายไปแล้ว ดังรายการ 3.13, 3.10, 3.8 และ 3.7 ตามลำดับ. ส่วนโปรแกรม `binaries_cross_entropy` เป็นฟังก์ชันสูญเสียครอสโอนโกรปแบบทวิภาคที่นิยมใช้กับงานจำแนกค่าทวิภาค แสดงในรายการ 3.15.

รายการ 3.15: ฟังก์ชันสูญเสียครอสโอนโกรปทวิภาค

---

```

1 def binaries_cross_entropy(yhat, y):
2     assert yhat.shape == y.shape
3
4     loss = yhat.copy()
5     zero_ids = np.where(y == 0)
6     loss[zero_ids] = 1 - loss[zero_ids]
7     loss = -np.log(loss)
8     return loss # output: K x N

```

---

หลังจากการฝึกสมบูรณ์แล้ว แบบจำลองถูกทดสอบดังคำสั่งตัวอย่างข้างล่างนี้

```

testxn, _ = normalize1(testx, normpars)
Yp = mlp(trained_net, testxn)
Yc = cutoff(Yp)
accuracy = np.mean(Yc == testy)

```

เมื่อ `Yp` เป็นค่าที่ทำนายจากแบบจำลอง ซึ่งมีค่าอยู่ในช่วง  $[0, 1]$  และเพื่อบังคับให้ผลตัดสินใจเป็น 0 หรือ 1 จึงใช้โปรแกรม `cutoff` ช่วย. ผลประเมินความแม่นยำของแบบจำลองจำแนกทวิภาค อาจวัดด้วยค่าความแม่นยำ `accuracy` ที่มีค่าระหว่างศูนย์ถึงหนึ่ง และค่าใกล้หนึ่งหมายถึงความแม่นยำสูง. โปรแกรม `cutoff` เผยนัดวยคำสั่งดังนี้

```

def cutoff(a, tau=0.5):
    return (a > tau)*1

```

ผลลัพธ์ที่แสดงในรูป 3.36 ได้มาจากการทดลองซ้ำ 40 ครั้ง. แบบจำลอง `naive` สร้างโดยไม่สนใจความหมายของข้อมูล ดังได้อภิรายไป. ส่วนแบบจำลอง `coding` สร้างโดยแปลงอินพุตเป็นรหัสตามความหมายของลักษณะสำคัญ แต่ละอย่าง โดยเฉพาะลักษณะสำคัญที่สามถึงห้า ซึ่งเป็นค่าแทนชื่อ ได้แก่ รูปทรงของมวลเนื้อ ลักษณะขอบของมวลเนื้อ และความหนาแน่นของมวลเนื้อ. แบบจำลอง `coding` เข้ารหัสลักษณะ

ทั้งสาม ด้วยรหัสหนึ่งร้อน โดยคำสั่ง `trainxc, cparams = mammo_coding(trainx)`.  
นอกจากลักษณะสำคัญทั้งสาม ลักษณะสำคัญที่หนึ่งถูกเข้ารหัสเป็นระดับค่า และลักษณะสำคัญที่สอง ซึ่งเป็นอายุ ถูกน้อมอุ่นโดยใช้ค่าสถิติ. โปรแกรม `mammo_coding` เจียนด้วยคำสั่งดังนี้

```
def mammo_coding(xin, cpars=None):
    d0 = xin[[0],:].copy()
    d0[d0 > 5] = 5 # make it {0,...,5}
    d1 = xin[[1],:].copy()
    d2 = xin[[2],:].copy()
    d3 = xin[[3],:].copy()
    d4 = xin[[4],:].copy()
    code0 = coding(d0, level_cbook(6))      # ordinal
    code1, cpars = normalize2(d1, cpars)    # age normalized with stats
    z4code = np.vstack( (np.zeros((1,4)), onehot_cbook(4)) )
    z5code = np.vstack( (np.zeros((1,5)), onehot_cbook(5)) )
    code2 = coding(d2, z4code) # nominal
    code3 = coding(d3, z5code) # nominal
    code4 = coding(d4, z4code) # nominal
    return np.vstack((code0, code1, code2, code3, code4)), cpars
```

เมื่อ `normalize2` เป็นโปรแกรมน้อมอุ่นโดยใช้อินพุต (ดูแบบฝึกหัด 3.13) และ `coding, onehot_cbook`  
และ `level_cbook` เจียนดังนี้

```
def coding(xin, code_book):
    ...
    xin: 1 x N
    code_book: np.array in K x C, K: key of x, C: code
    ...
    return code_book[xin][0].transpose()

def onehot_cbook(K):
    return np.diag(np.ones((K,)))

def level_cbook(K):
    return np.tril(np.ones((K,K)))
```

การเข้ารหัสอินพุตเช่นนี้ ทำให้อินพุตที่เข้าแบบจำลองกล้ายเป็น 20 มิติ แต่ช่วยให้ผลการทำงานดีขึ้น  
อย่างชัดเจน ดังตัวอย่างผลลัพธ์ที่แสดงในรูป 3.36. รูป 3.36 รายงานผลด้วยค่าความแม่นยำ ในช่วง 0 ถึง

1. บางครั้ง ค่าความแม่นยำนิยมรายงานเป็นเปอร์เซ็นต์ เช่น ค่าความแม่นยำเฉลี่ยเป็น 80.9% และ 82.3% สำหรับแบบจำลอง **naive** และแบบจำลอง **coding**.

นอกจากการรายงานผลด้วยค่าความแม่นยำแล้ว เมทริกซ์ความสับสน (confusion matrix) เป็นการแจกแจงผลการทำงานของอุปกรณ์ คือ **จำนวนบวกจริง** (true positive คำย่อ TP), **จำนวนบวกเท็จ** (false positive คำย่อ FP), **จำนวนลบจริง** (true negative คำย่อ TN) และ **จำนวนลบเท็จ** (false negative คำย่อ FN). จำนวนบวกจริง คือจำนวนจุดข้อมูลทดสอบที่ถูกทายเป็น 1 และเฉลยเป็น 1. จำนวนบวกเท็จ คือจำนวนจุดข้อมูลทดสอบที่ถูกทายเป็น 1 แต่เฉลยเป็น 0. จำนวนลบจริง คือจำนวนจุดข้อมูลทดสอบที่ถูกทายเป็น 0 และเฉลยเป็น 0. จำนวนลบเท็จ คือจำนวนจุดข้อมูลทดสอบที่ถูกทายเป็น 0 แต่เฉลยเป็น 1.

ตัวอย่างเมทริกซ์ความสับสน แสดงข้างล่าง

		ผลจริง		
		1	0	
ผลทำนาย	1	True Positive 130	False Positive 26	Precision = 0.833
	0	False Negative 25	True Negative 119	
		Recall = 0.839		

นอกจากเมทริกซ์ความสับสนแล้ว ตัวอย่างยังรายงานค่าความเที่ยงตรง (precision) และค่าการระลึกกลับ (recall) ที่ด้านข้าง และด้านล่างของเมทริกซ์ความสับสนด้วย.

ค่าความเที่ยงตรงและค่าการระลึกกลับ ออกแบบ โดยคำนึงถึง ความสมดุลของการแจกแจงของกลุ่มข้อมูล. ข้อมูลชุดนี้มีการแจกแจงข้อมูลพอ ๆ กัน. ข้อมูลกลุ่ม 1 (ผลเป็นเนื้อร้าย) และข้อมูลกลุ่ม 0 (ผลเป็นเนื้อดี) มีจำนวนระเบียนใกล้เคียงกัน คือ 445 ระเบียนและ 516 ระเบียน ตามลำดับ. ลักษณะการแจกแจง เช่นนี้ ทำให้ค่าความแม่นยำ สามารถสะท้อนคุณภาพการทำนายจริงของแบบจำลองได้ดี. แต่หากการแจกแจงไม่สมดุลอย่างมาก เช่น สมมติอัตราส่วนคนเป็นมะเร็งตับอ่อนต่อประชากรมีค่าน้อยกว่า 1% เพียงแต่แบบจำลองทำนายผลเป็น 0 คือไม่เป็นมะเร็ง สำหรับทุก ๆ ตัวอย่างที่เข้ามาทดสอบ แบบจำลองนั้นก็มีโอกาสถูกถูกสูงมาก ๆ แต่มันไม่ได้มีประโยชน์ต่อการช่วยระบุกลุ่มเสี่ยงเลย.

ตั้งนั้นแทนที่จะใช้แค่ค่าความแม่นยำ ค่าความเที่ยงตรง (สมการ 3.48) และค่าการระลึกกลับ (สมการ 3.49) จะช่วยสะท้อนคุณภาพการทำงานได้ดีกว่า โดยเฉพาะสำหรับข้อมูลที่มีการแจกแจงข้อมูลไม่สมดุล การตรวจสอบค่าความเที่ยงตรง และค่าการระลึกกลับ จะช่วยบอกได้ว่าแบบจำลองทำงานได้ดีอย่างสมดุลกับผลที่นาย. สำหรับงานจำแนกค่าทวิภาค ค่าความเที่ยงตรง  $P$  สามารถคำนวณได้จาก

$$P = \frac{TP}{TP + FP} \quad (3.48)$$

และค่าการระลึกกลับ  $R$  สามารถคำนวณได้จาก

$$R = \frac{TP}{TP + FN} \quad (3.49)$$

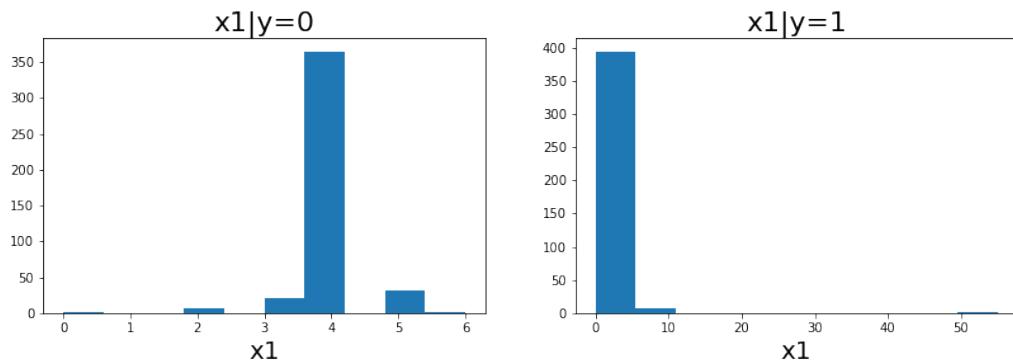
หมายเหตุ เมื่อใช้ค่าความเที่ยงตรงและค่าการระลึกกลับในการวัดผล กับข้อมูลที่มีการแจกแจงไม่สมดุล จะกำหนดให้กลุ่ม 1 (กลุ่มบวก) เป็นกลุ่มที่มีสัดส่วนน้อย (เช่น กลุ่มของมะเร็งตับอ่อน) และกลุ่ม 0 (กลุ่มลบ) เป็นกลุ่มใหญ่ (เช่น กลุ่มที่ไม่ได้เป็น).

ค่าความเที่ยงตรงและค่าการระลึกกลับ ช่วยสะท้อนความสามารถของแบบจำลองได้ดีขึ้น โดยเฉพาะในกรณีที่ข้อมูลมีการแจกแจงระหว่างกลุ่มไม่สมดุล. แต่การรายงานผล ด้วยตัวเลขสองตัวนี้ อาจทำให้ลำบากในการเปรียบเทียบผล เช่น ผลของแบบจำลองหนึ่ง อาจจะได้ค่าความเที่ยงตรงสูง แต่ค่าการระลึกกลับต่ำ แต่ อีกแบบจำลองหนึ่ง มีค่าความเที่ยงตรงต่ำ แต่ค่าการระลึกกลับสูง. เพื่อความสะดวก ค่าคะแนนเอฟ (F Score หรือ บางครั้งเรียก F1 Score) ซึ่งเป็นค่าเฉลี่ยเชิงเรขาคณิต มักใช้ในการสรุป ค่าความเที่ยงตรง และค่าการระลึกกลับ ให้เป็นตัวเลขตัวเดียว. ค่าคะแนนเอฟ สัญกรณ์  $F$  สามารถคำนวณได้จาก

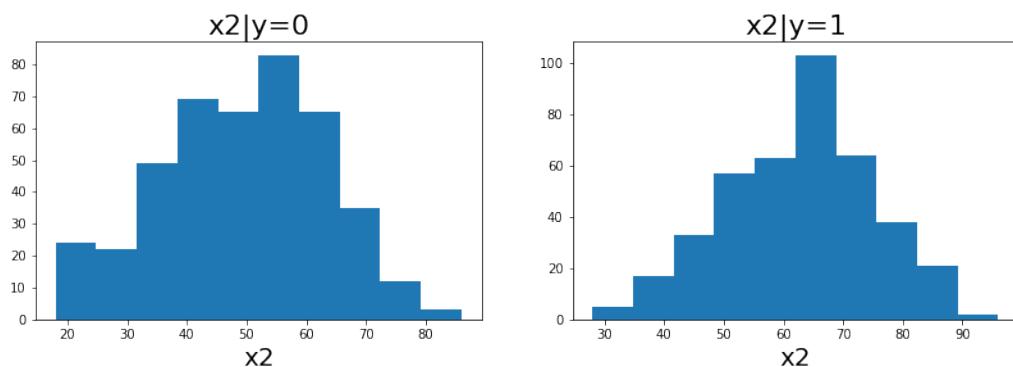
$$F = 2 \cdot \frac{P \cdot R}{P + R}. \quad (3.50)$$

จากเมทริกซ์ความสับสนในตัวอย่าง ค่าคะแนนเอฟ จะเท่ากับ 0.836.

สุดท้าย ย้อนกลับมาทบทวนเรื่องผลลัพธ์สุดท้าย จากหัวข้อ 2.2 ตัวอย่าง ปัญหาการตรวจเต้านมด้วยภาพเอ็กซเรย์. ตัวอย่างนั้น อภิปรายการคำนวณหา โอกาสที่จะเป็นมะเร็ง เมื่อผลการตรวจน้ำนมบุ่ว เป็น. นั่นคือ หา  $\Pr(C = 1 | M = 1)$  เมื่อ  $M = 1$  หมายถึงผลการตรวจระบุว่าเป็นมะเร็ง และ  $C = 1$  หมายถึงการเป็นมะเร็งจริง ๆ.



รูปที่ 3.33: ข้อมูลชุดภาพเอ็กซ์เรย์เต้านม ลักษณะสำคัญมิตร ค่าประเมินไบเบตส์ สำหรับกรณีก้อนเนื้อเป็นเนื้อดี (ภาพซ้าย) และเนื้อร้าย (ภาพขวา).

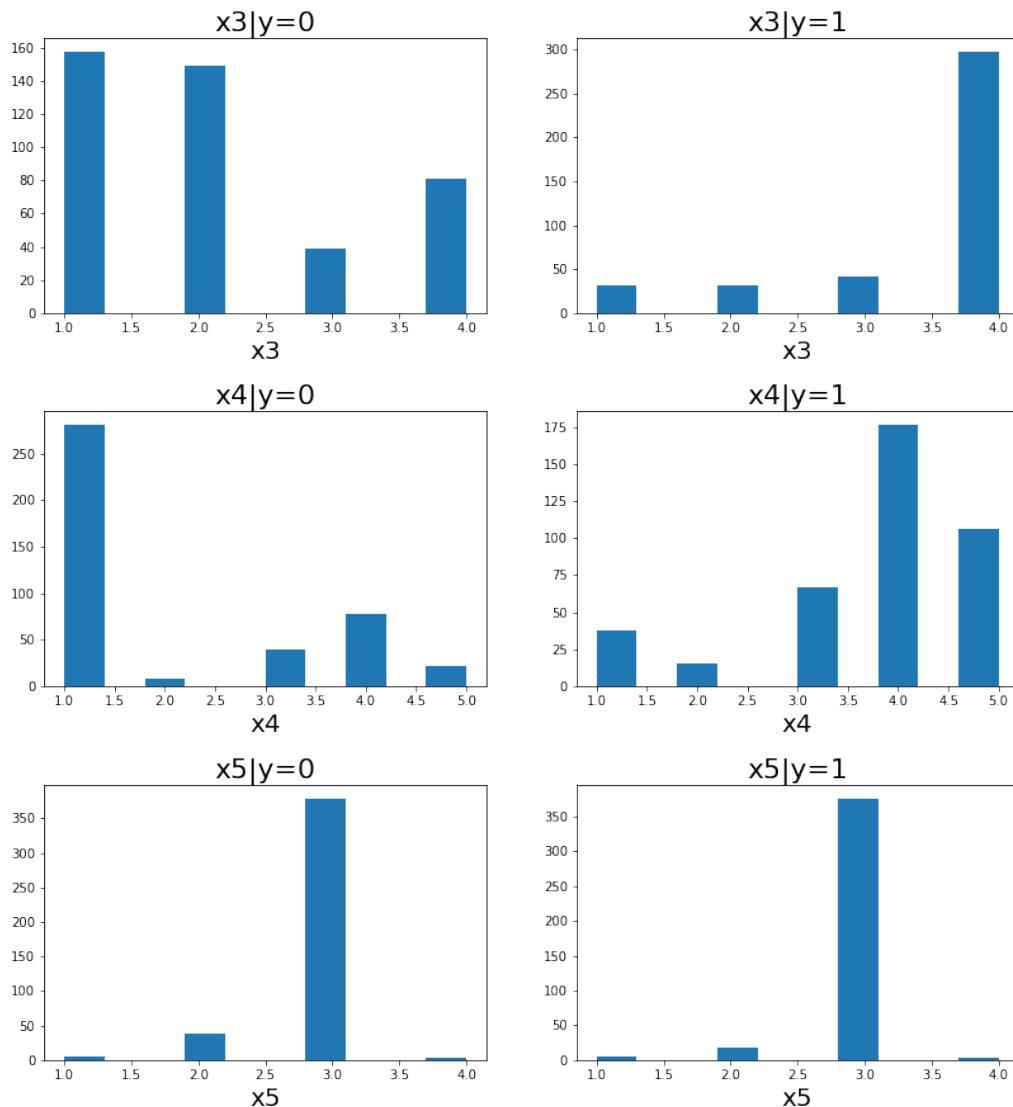


รูปที่ 3.34: ข้อมูลชุดภาพเอ็กซ์เรย์เต้านม ลักษณะสำคัญมิตรที่สอง อายุ สำหรับกรณีก้อนเนื้อเป็นเนื้อดี (ภาพซ้าย) และเนื้อร้าย (ภาพขวา).

ข้อมูลประกอบ คือ 17% ของผู้หญิงอายุเกิน 40 ปี เป็นมะเร็งเต้านม นั่นคือ  $\Pr(C = 1) = 0.17$ . ค่า  $\Pr(M = 1|C = 1)$  สามารถประมาณได้จากค่าในเมทริกซ์ความสัปสน  $\Pr(M = 1|C = 1) \approx R$  เมื่อ  $R$  คือค่าระลึกกลับ ซึ่งในตัวอย่างนี้เป็น 0.839 และค่า  $\Pr(M = 1|C = 0)$  ประเมินได้จาก  $\Pr(M = 1|C = 0) \approx \frac{FP}{FP+TN}$  ซึ่งในตัวอย่างนี้เป็น  $\frac{26}{26+119} = 0.179$ .  
เมื่อร่วมหลักฐานทุกอย่างเข้าด้วยกัน โดยใช้กฎของเบย์ล์ จะได้ว่า

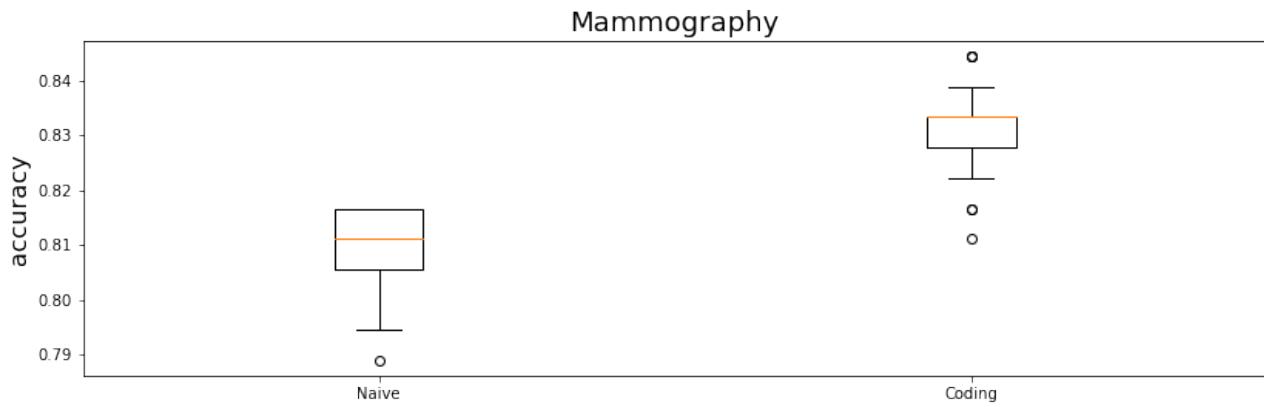
$$\begin{aligned}\Pr(C = 1|M = 1) &= \frac{\Pr(M = 1|C = 1)\Pr(C = 1)}{\Pr(M = 1|C = 0)\Pr(C = 0) + \Pr(M = 1|C = 1)\Pr(C = 1)} \\ &= \frac{0.839 \cdot 0.17}{0.179 \cdot 0.83 + 0.839 \cdot 0.17} = 0.4898.\end{aligned}$$

ดังนั้น ผลตรวจภาพเอ็กซ์เรย์เต้านม จึงเป็นเสมือนแค่การตรวจเบื้องต้น เพื่อประกอบการตัดสินใจ ตัดซื้อน้ำหนึ่งไปตรวจ (biopsy).



**รูปที่ 3.35:** ข้อมูลชุดภาพເອົກະເຮົາຕ່ານມ ລັກຜະສຳຄູນມືດີທີ່ສາມັງເຫຼົ້າ ຮູປທຽບມາລັກຜະສຳມາລັກຜະສຳນີ້ ແລະຄວາມໜາກແນ່ນຂອງມາລັກຜະສຳຄູນເຫຼົ້ານີ້ເປັນຄ່າແທນໜີ້. ພາພ້ອຍ ແສດຄ່າກຣັນກ້ອນນີ້ເປັນນີ້ເວົ້າດີ. ພາພ້ອຍ ແສດຄ່າກຣັນກ້ອນນີ້ເປັນນີ້ເວົ້າດີ. ພາພ້ອຍ ແສດຄ່າກຣັນກ້ອນນີ້ເປັນນີ້ເວົ້າດີ.

**การຈັດການກັບຂໍ້ມູນຂາດໝາຍ.** ແບບຝຶກທັດ 3.15 ແນະນຳວິທີຕໍ່ຮະເບີນທີ່ມີຂໍ້ມູນຂາດໝາຍທີ່ໄປ ຈຶ່ງເປັນໜີ້ໃນແນວທາງທີ່ນີ້ຢູ່ມ ແລະ ດຳເນີນການໄດ້ຈ່າຍ. ນອກຈາກແນວທາງຕໍ່ຮະເບີນທີ່ ຍັງມີແນວທາງອື່ນ ຈຶ່ງ ອີກ ເຊັ່ນ ວິທີກາຮແທນຄ່າຂໍ້ມູນທີ່ຂາດໝາຍໄປ ດ້ວຍຄ່າເນັລີ່ຍ ສໍາຮັບມືດີຫຸ້ນໂຫຼດຂອງເຂົດຂໍ້ມູນທີ່ເປັນຄ່າຕ່ອນເນື່ອງ (continuous-value field) ຫຼື ແທນດ້ວຍຄ່າທີ່ພບບ່ອຍທີ່ສຸດ ໃນກຣັນທີ່ເຂົດຂໍ້ມູນເປັນຄ່າແທນໜີ້ ຂລາກ ຫຼື ມາດໜູ້ (categorical field). ອີກວິທີທີ່ນີ້ຢູ່ມ ດື່ມ ວິທີກາຮແທນຄ່າທີ່ໝາຍໄປ ດ້ວຍຄ່າທຸກຄ່າທີ່ເປັນໄປໄດ້. ວິທີນີ້ ຮະເບີນທີ່ຂໍ້ມູນຂາດໝາຍໄປ ຈະ ຖຸກແທນດ້ວຍຮະເບີນໃໝ່ໜ່າຍ ຈຶ່ງ ຮະເບີນ ໂດຍຄ່າຂໍ້ມູນຕ່າງ ຈຶ່ງ ຂອງຮະເບີນໃໝ່ ຈະເໜືອນຮະເບີນໃໝ່ ຍກເວັ້ນຂໍ້ມູນທີ່ໝາຍໄປ ຈະ ຖຸກແທນດ້ວຍຄ່າໜີ້ໃນກລຸ່ມຄ່າທີ່ເປັນໄປໄດ້. ຈຳນວນຮະເບີນໃໝ່ນີ້ຈະເຖິງກັບຈຳນວນຄ່າທີ່ເປັນໄປໄດ້ຂອງເຂົດຂໍ້ມູນທີ່ຂໍ້ມູນຂາດໝາຍໄປ. ດູກຮີ້ສີມາລາບຸສເສແລະສູ[81]ເພີ່ມເຕີມ ສໍາຮັບວິທີຕ່າງ ຈຶ່ງ ໃນການຈັດການ



**รูปที่ 3.36:** ผลการทำนายข้อมูลภาพเอ็กซเรย์เต้านม ของแบบจำลองที่ฝึกด้วยข้อมูลโดยไม่สนใจความหมาย **naive** (กล่องซ้าย) เปรียบเทียบกับผลของแบบจำลองที่ฝึกด้วยข้อมูลที่เข้ารหัสตามความหมาย **coding** (กล่องขวา). แกนนอน แสดงค่าความแม่นยำ.

ข้อมูลที่ขาดหายไป. นอกจากนั้น การเรียนรู้แบบกึ่งมีผู้ช่วยสอน[19] ยังเสนอแนวทางที่น่าสนใจ ในจัดการกับข้อมูลขาดหายไป โดยเฉพาะกับฉลากที่ขาดหายไป.

วิธีแทนค่าขาดหายด้วยทุกค่าที่เป็นไปได้ การดำเนินการจะยุ่งยากกว่าวิธีอื่น ๆ คำสั่งข้างล่างแสดงตัวอย่างวิธีทำ

```
mammo3 = []
for row in mammo_miss[:-1]:
    mammo3.extend(fix_row(row, field_vals))
```

เมื่อ **mammo\_miss** เป็นลิสต์ของระเบียนที่มีข้อมูลขาดหาย. โปรแกรม **fix\_row** แสดงในรายการ 3.16 และการทดลองตัวอย่าง กำหนด **field\_vals** ด้วย

```
field_vals = {0: [0, 2, 3, 4, 5], 1: list(range(18, 97, 6)),
              2: [1, 2, 3, 4],
              3: [1, 2, 3, 4, 5], 4: [1, 2, 3, 4]}
```

สังเกต การทดลองตัวอย่างเลือกแทนค่าอยุ่เป็นช่วง ๆ ช่วงละหกปี แทนการแทนทุกค่าที่เป็นไปได้ ซึ่งหากทำทุกค่า ในกรณีนี้ อาจเพิ่มจำนวนระเบียนขึ้นมาศาลโดยไม่จำเป็น.

รายการ 3.16: โปรแกรมช่วยวิธีแทนค่าขาดหายด้วยทุกค่าที่เป็นไปได้.

---

```
1 def fix_row(row, fix_vals):
2     rows = [row]
3     i = 0
4     while i < len(rows):
5         flag_next = True
```

```

6   for j, c in enumerate(rows[i]):
7       if c is None:
8           for v in fix_vals[j]:
9               newrow = rows[i].copy()
10              newrow[j] = v
11              rows.append(newrow)
12          del rows[i]
13          flag_next = False
14          break
15      # end for j, c
16  if flag_next:
17      i += 1
18 # end while i
19 good_rows = rows
20 return good_rows

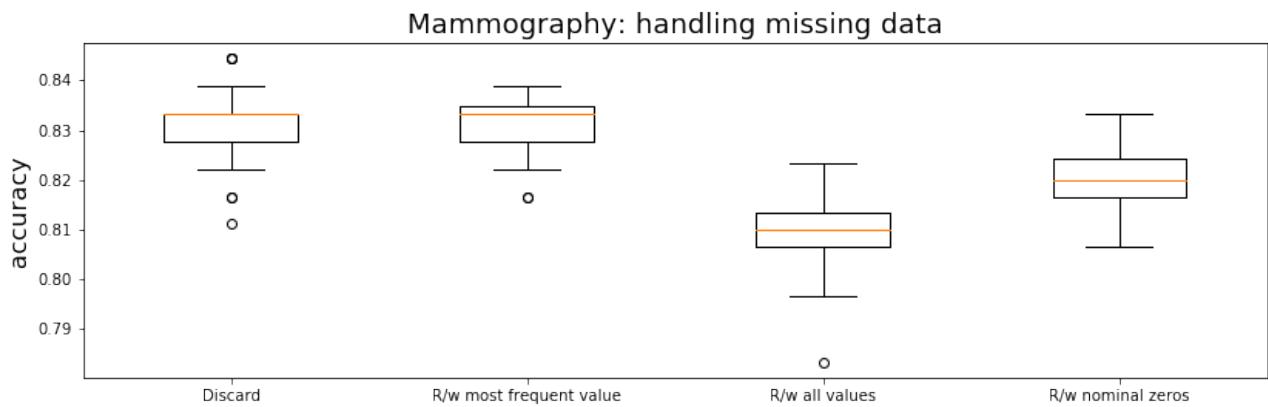
```

---

ตารางที่ 3.6: ตัวอย่างผลการทำนายของชุดข้อมูลภาพอีกชุดเด้านม เมื่อใช้วิธีจัดการกับข้อมูลขาดหายแบบต่าง ๆ.

วิธี	ค่าความแม่นยำ	
	ค่าเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน
ตัดระเบียนไม่สมบูรณ์ออก	0.8314	0.0073
แทนด้วยค่าเฉลี่ย หรือค่าพบร้อยที่สุด	0.8308	0.0063
แทนด้วยทุกค่าที่เป็นไปได้	0.8093	0.0070
แทนด้วยรหัสศูนย์	0.8203	0.0061

รูป 3.37 และตาราง 3.6 แสดงตัวอย่างผลเปรียบเทียบ วิธีจัดการข้อมูลขาดหายแบบต่าง ๆ. นอกจากวิธีทั่วไป ดังอภิปรายข้างต้น ในรูป ยังเสนอผลจากวิธีแทนข้อมูลค่าแทนชื่อที่ขาดหาย ด้วยรหัสศูนย์. เนื่องจากข้อมูลภาพอีกชุดเด้านม มีเขตข้อมูลหลายเขต ที่เป็นชนิดค่าแทนชื่อ ซึ่งในตัวอย่างของแบบฝึกหัด 3.15 ใช้การเข้ารหัสหนึ่งร้อน (ดูโปรแกรม **coding** และ **onehot\_cbook**). รหัสหนึ่งร้อน จะใช้ค่าทวินภาพหลายค่า แทนชื่อ โดยมีแค่หนึ่งค่าที่ดำเนินการสำหรับชื่อนั้น ๆ ที่จะมีค่าเป็นหนึ่ง เพื่อรับรู้ว่า ออกนั้น และค่าอื่น ๆ ในรหัสจะเป็นศูนย์ เช่น ฉลากหนึ่ง แทนด้วย  $[1, 0, 0, 0]^T$  และฉลากสอง แทนด้วย  $[0, 1, 0, 0]^T$  สำหรับกรณีที่มีสี่ฉลาก. ข้อมูลที่ขาดหาย อาจสามารถแทนเป็นรหัสศูนย์ โดยไม่มีหนึ่งที่ดำเนินการได้ในรหัสเลย เช่น  $[0, 0, 0, 0]^T$ . นี่คือ แนวคิดของวิธีแทนข้อมูลชนิดค่าแทนชื่อที่ขาดหายด้วยรหัสศูนย์ (ระบุด้วย R/w nominal zeros ในภาพ).



**รูปที่ 3.37:** แผนภูมิกล่อง แสดงตัวอย่างผลเบรียบเทียบ วิธีการจัดการกับข้อมูลขาดหายแบบต่าง ๆ. ผลได้จากการทดลองซ้ำ 40 ครั้ง. แกนตั้งแน่นค่าความแม่นยำ. กล่องช้ายสุด **Discard** แสดงผลที่ได้ เมื่อใช้วิธีตัดระเบียนที่ข้อมูลไม่สมบูรณ์ทั้งหมด. กล่องที่สองจากช้าย **R/w most frequent value** แสดงผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ด้วยค่าเฉลี่ย หรือค่าที่พบบ่อยที่สุด. กล่องที่สาม **R/w all values** แสดงผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ด้วยทุกค่าที่เป็นไปได้. กล่องที่สี่ **R/w nominal zeros** แสดงผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ที่เป็นค่าแทนชื่อ ด้วยรหัสศูนย์.

### แบบฝึกหัด 3.16

ข้อมูลเอมนิสต์ [116] เป็นชุดข้อมูลขนาดใหญ่ของภาพตัวเลขลายมือเขียน พร้อมเดลย์. ชุดข้อมูลประกอบด้วย 60000 ตัวอย่าง สำหรับข้อมูลฝึก และ 10000 ตัวอย่าง สำหรับข้อมูลทดสอบ. แต่ละตัวอย่าง ตัวเลขในภาพถูกปรับขนาด และปรับจุดศูนย์กลาง และอยู่ในภาพสเกลเทา (gray scale) ขนาด  $28 \times 28$  พิกเซล. คำอธิบาย และข้อมูลเอมนิสต์ สามารถหาได้ที่ <http://yann.lecun.com/exdb/mnist/>

จะทำแบบจำลองโครงข่ายประสาทเทียม เพื่อรู้จำภาพตัวเลขลายมือเขียน. นั่นคือ สร้างแบบจำลองที่ tally ฉลากตัวเลข จากภาพตัวเลขลายมือเขียน เลือกแบบจำลอง ฝึก ทดสอบ วัดผล รายงานผล ภาระราย และสรุป.

ข้อมูลเอมนิสต์อยู่ในไฟล์ชนิดไบนาเรีย (binary file) และเก็บด้วยรูปแบบเอนเดียนใหญ่ (big endian). ไฟล์ฉลาก เช่น **train-labels.idx1-ubyte** มีส่วนหัว 8 ไบต์ ซึ่งเป็นตัวเลขขนาดสามสิบสองบิตสองตัว แล้วจึงตามด้วยข้อมูลขนาด 60000 ไบต์ แต่ละไบต์เป็นตัวเลขแทนฉลากของข้อมูลแต่ละระเบียน. ไฟล์ข้อมูลภาพ เช่น **train-images.idx3-ubyte** มีส่วนหัว 16 ไบต์ ซึ่งเป็นตัวเลขขนาดสามสิบสองบิตสี่ตัว แล้วจึงตามด้วยข้อมูลที่แต่ละไบต์เป็นตัวเลขแทนค่าความเข้มของพิกเซล (0 ถึง 255) เรียงกันต่อ กันไป 784 ไบต์ต่อภาพ จำนวน 60000 ภาพ รวมเป็นข้อมูลภาพทั้งหมด 47040000 ไบต์.

ตัวอย่างคำสั่งข้างล่างนำเข้าข้อมูลฉลากของชุดฝึก

```
import struct
with open('train-labels.idx1-ubyte', 'rb') as f:
    # Read header
```

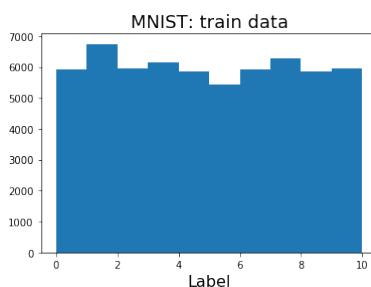
```

for i in range(2):
    try:
        bi = f.read(4)
        print('bi:', bi)
        print('* little endian:', struct.unpack_from("<I", bi)[0])
        print('* big endian:', struct.unpack_from(">I", bi)[0])
    except struct.error:
        print('error')

trainy = []
for i in range(60000):
    try:
        bi = f.read(1)
        trainy.append(struct.unpack_from(">B", bi)[0])
    except struct.error:
        print('error')
trainy = np.array(trainy).reshape((1, -1))

```

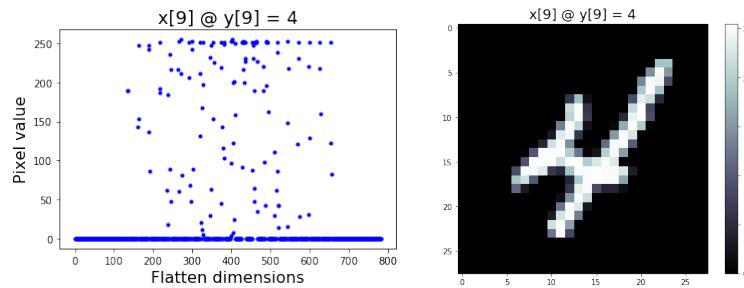
คำสั่งข้างต้นใช้คำสั่ง `struct.unpack_from(">I", bi)` เพื่ออ่านตัวเลข (32 บิต) ออกมารับตรวสอบความถูกต้อง และใช้คำสั่ง `struct.unpack_from(">B", bi)` เพื่ออ่านข้อมูลตัวเลข (8 บิต) ออกมาระบบรวมใน `trainy`. รูป 3.38 แสดงการแจกแจงของข้อมูลเอมนิสต์.



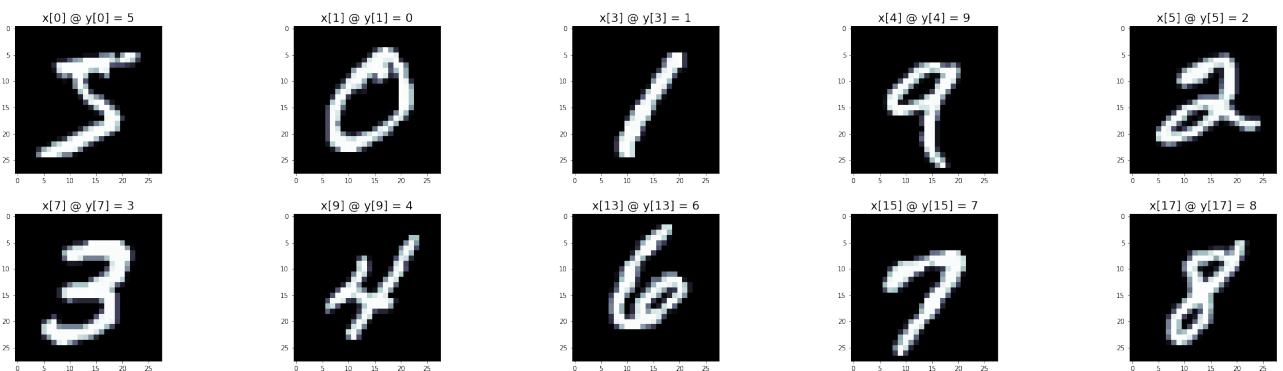
รูปที่ 3.38: การแจกแจงของข้อมูลเอมนิสต์ชุดฝึก. จำนวนจุดข้อมูลของแต่ละฉลากแสดงในแกนตั้ง. ฉลาก (0 ถึง 9) แสดงในแกนนอน.

ในลักษณะเดียวกัน ข้อมูลภาพ เช่น `train-images.idx3-ubyte` ก็สามารถนำเข้าเป็นตัวแปร `trainx` (นามไฟล์เรียก สัดส่วน `(784, 60000)`). ข้อมูลอินพุตแต่ละระเบียน จะมี 784 ค่าลักษณะ. แต่ละค่าลักษณะ เป็นเลขจำนวนเต็มตั้งแต่ 0 ถึง 255. ตัวอย่างของอินพุตแต่ละระเบียน แสดงในรูป 3.39 ภาพช้าย. เมื่อนำค่าลักษณะ จะจัดเรียงใหม่เป็นสองลำดับชั้นมิติ และนำไปวัดลงในภาพสองมิติ จะได้ดังแสดงในภาพขวา ซึ่งวาดด้วยคำสั่ง `plt.imshow(trainx[:, 9].reshape((28,`

28)), cmap=plt.cm.bone) ซึ่งเลือกข้อมูลภาพลำดับที่เก็บมาแสดง. ตัวอย่างภาพตัวเลขต่าง ๆ ของข้อมูลเอมนิสต์ แสดงดังรูป 3.40.



รูปที่ 3.39: อินพุตของเอมนิสต์. ภาพช้าย แสดงโดยไม่มีโครงสร้างมิติ. ค่าของอินพุตแสดงเรียงไปทั้ง 784 ค่า. ด้านซ้ายของค่าแสดงตามแกนนอน และแกนตั้งแสดงค่าความเข้มของพิกเซล. ภาพขวา แสดงโดยจัดโครงสร้างมิติเป็นสองลำดับชั้น (อินพุต 784 ค่าถูกจัดเรียงเป็น  $28 \times 28$ ) และค่าพิกเซลแทนด้วยระดับสีเทาต่าง ๆ ดังแสดงในแบบแผนที่สีทางขวาของภาพ. บนแต่ละภาพ แสดงลำดับของภาพและฉลากเฉลย. ตัวอย่างนี้ ได้จากข้อมูลลำดับที่เก็บ ซึ่งฉลากเฉลยระบุว่าเป็นเลขสี่.



รูปที่ 3.40: ตัวอย่างภาพตัวเลขข้อมูลเอมนิสต์. บนแต่ละภาพ แสดงตัวเลข แล้วฉลากเฉลย.

ภาระกิจการรู้จำตัวเลขลายมือ เป็นงานการจำแนกประเภท ที่เอาร์พุตมีค่าที่เป็นไปได้ 10 ฉบับ. คำสั่งข้างล่าง แสดงตัวอย่างการสร้างโครงข่ายประสาทเทียมเป็นแบบจำลองสำหรับงานรู้จำตัวเลขลายมือ โดยใช้ข้อมูลเอมนิสต์ในการฝึก.

```
trainxn = trainx/255    # normalize pixel [0,255] to [0,1]
trainyc = coding(trainy, onehot_cbook(10))

# Configure net
net = w_initngw([784, 8, 10])
net['act1'] = sigmoid
net['act2'] = softmax
```

```
# Train net
```

```
trained_net, train_losses = train_mlp(net, trainxn, trainyc,
                                         cross_entropy, lr=1, epochs=300)
```

เมื่อ `trainx` และ `trainy` เป็นข้อมูลเอมนิสต์สำหรับฝึกที่นำเข้ามา. คำสั่ง `trainxn = trainx/255` ปรับขนาดของอินพุตให้อยู่ในช่วงศูนย์ถึงหนึ่ง. ส่วนคำสั่งถัดมา ปรับเอาต์พุตให้อยู่ในรหัสหนึ่งร้อน (โปรแกรม `coding` และ `onehot_cbook` ดูแบบฝึกหัด 3.15). ตัวอย่างนี้ใช้โครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยซ่อน โดยโครงข่ายรับอินพุตขนาด 784 มิติ และให้อาต์พุตออกขนาด 10 มิติ. ค่าน้ำหนักเริ่มต้น กำหนดด้วยวิธีเหنجี่นวิดโดยร์ (รายการ 3.12). พังก์ชันซอฟต์แมกซ์ และฟังก์ชันสูญเสียクロสเซอนโทรปี ถูกเลือกใช้สำหรับภารกิจการจำแนกกลุ่ม (รายการ 3.17 และ 3.18). ตัวอย่างนี้ฝึก 300 สมัย โดยใช้อัตราเรียนรู้เป็น 1.

สังเกต โปรแกรมซอฟต์แมกซ์ เขียนโดยอาศัยคณสมบัติคณิตศาสตร์

$$\frac{e^{a_k}}{\sum_i e^{a_i}} = \frac{e^{a_k - a_{\max}}}{\sum_i e^{a_i - a_{\max}}}$$

เมื่อ  $a_{\max}$  คือค่าส่วนประกอบของเวกเตอร์ที่มีค่ามากที่สุด. (ทดลองรันฟังก์ชันซอฟต์แมกซ์ โดยใช้ค่า `va` ต่าง ๆ. ทดลองค่าใหม่ ๆ ด้วย เช่น `va = np.array([[800], [500], [100]])`) สังเกตผลและเปรียบเทียบกับผลจากโปรแกรมที่แสดงในแบบฝึกหัด 2.32 และอภิราย.) ฟังก์ชันสูญเสียクロสเซอนโทรปี (รายการ 3.18) ซึ่งคือ  $-\log \hat{y}_k$  สำหรับ ทุก ๆ ค่า  $k$  ที่ทำให้  $y_k = 1$  ก็เขียนด้วยการคำนวณ

$$-\log \left( \sum_k y_k \cdot \hat{y}_k \right)$$

เมื่อ  $\hat{y}_k$  คืออาต์พุตจากแบบจำลองที่ผ่านซอฟต์แมกซ์ออกมามา สำหรับกลุ่มที่  $k^{th}$  และอาต์พุตเฉลี่ย  $y_k$  เป็นส่วนประกอบของฉลากในรหัสหนึ่งร้อน  $\mathbf{y} = [y_1, \dots, y_K]$  เมื่อ  $K$  เป็นจำนวนกลุ่ม. นั่นคือ  $y_k \in \{0, 1\}$  และ  $\sum_{k=1}^K y_k = 1$ . (อภิราย การเขียนโปรแกรมฟังก์ชันสูญเสียクロสเซอนโทรปี ดังรายการ 3.18 เปรียบเทียบกับการเขียนโปรแกรมตาม  $-\sum_k y_k \cdot \log \hat{y}_k$ .) หมายเหตุ ตัวแปร `yhat` และ `y` เป็นเมตริกซ์ขนาด  $K \times N$  เมื่อ  $N$  เป็นจำนวนจุดข้อมูล และผลลัพธ์ของ `cross_entropy` ซึ่งคือค่าสูญเสียของจุดข้อมูลต่าง ๆ เป็นเมตริกซ์ขนาด  $1 \times N$ . ตั้งนั้นการคำนวณค่าสูญเสียไม่สามารถเขียนในรูปเวกตอร์เช่นได้. โปรแกรมในรายการ 3.18 จึงเขียนด้วย `-np.log(np.sum(y*yhat, axis=0))`.

```

1 def softmax(va):
2     assert va.shape[0] > 1, 'va must be in K x N.'
3
4     amax = np.max(va, axis=0)
5     ap = va - amax
6     expa = np.exp(ap)
7     denom = np.sum(expa, axis=0)
8     return expa/denom

```

---

รายการ 3.18: พัฟ์ชันสูญเสียครอสแอนโตรปี

```

1 def cross_entropy(yhat, y):
2     assert yhat.shape == y.shape
3     eps = 1e-323
4     return -np.log(np.sum(y*yhat, axis=0) + eps).reshape((1, -1))

```

---

หลังจากฝึกเสร็จ แบบจำลองสามารถทำไปใช้งานได้. คำสั่งข้างล่าง แสดงตัวอย่างการทดสอบแบบจำลองที่ฝึกมา

```

testxn = testx/255
Yp = mlp(trained_net, testxn)
Yc = np.argmax(Yp, axis=0)
accuracy = np.mean(Yc == testy[0, :])
print('Accuracy = ', accuracy)

```

เมื่อ **testx** และ **testy** เป็นข้อมูลทดสอบ. ตัวแปร **Yp** เป็นเอาต์พุตของแบบจำลอง ที่อยู่ในรูปประมวลผลหัศหนึ่งร้อน ส่วน **Yc** คือฉลากที่ทาย โดย เลือกฉลากที่มีส่วนประกอบในรหัสหนึ่งร้อน มีค่าสูงสุด เป็นฉลากที่ทาย.

ตาราง 3.7 แสดง เมทริกซ์สับสน ของผลทดสอบตัวอย่างที่ได้. ตัวเลขตามแนวทะแยงมุม คือจำนวนที่ทายถูก ในแต่ละประเภท. จากเมทริกซ์สับสนในตัวอย่าง ช่วยให้การวิเคราะห์ความผิดพลาด ทำได้สะดวกขึ้น เช่น จากเมทริกซ์ ภาพตัวเลขที่ทายผิดมากที่สุด คือภาพเลขเก้า ที่ถูกทายเป็นเลขสี่ตีน 81 ครั้ง และในทางกลับกัน ก็มีผิดไป 48 ครั้ง. รูป 3.41 แสดงตัวอย่างรูปที่สับสนระหว่างภาพเลขสี่ และภาพเลขเก้า.

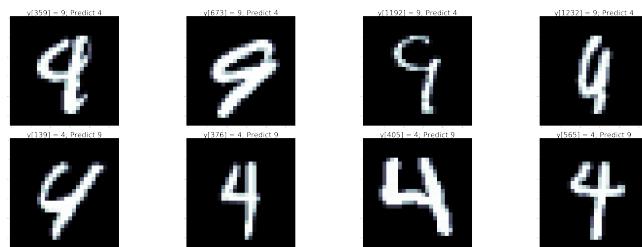
### แบบฝึกหัด 3.17

การทำนายการจับตัวกันระหว่างโปรตีนและโมเลกุลขนาดเล็ก (protein-ligand binding prediction) เป็นภารกิจที่สำคัญในกระบวนการค้นหายา โดยเฉพาะการออกแบบยา (ดูเกร็ดความรู้การค้นหายา). แบบฝึกหัดนี้ ได้แรงบรรดาลใจจากการศึกษาของชานเชซและคณะ[175]. คณะของชานเชซ ใช้ข้อมูลจากฐาน

ตารางที่ 3.7: เมทริกซ์สับสน แสดงผลการทำนาย โดยแยกตามประเภท ทั้งประเภทที่ไทย (แสดงตามแกรม) และประเภทจริง ที่ระบุด้วยฉลากเฉลย (แสดงตามสมมติ).

ทำนาย	ฉลากเฉลย									
	0	1	2	3	4	5	6	7	8	9
0	947	0	17	5	3	12	17	4	8	8
1	1	1100	24	1	1	5	4	18	12	6
2	1	4	900	30	3	3	5	28	5	1
3	4	5	17	872	0	68	0	8	21	15
4	0	1	11	0	896	14	14	11	13	81
5	15	2	4	55	3	737	20	0	44	11
6	6	4	10	2	14	19	894	0	21	1
7	4	2	12	16	1	9	1	921	8	34
8	2	17	31	22	13	18	3	2	831	8
9	0	0	6	7	48	7	0	36	11	844

ค่าความแม่นยำ 89.4%.



รูปที่ 3.41: ตัวอย่างภาพที่สับสนของเอมนิสต์. ภาพในแควรบ์ เลขเก้าที่ถูกไทยเป็นเลขสี่ และภาพในแควร์ล่าง เลขสี่ที่ถูกไทยเป็นเลขเก้า.

ช้อมูลดียูดี (DUD: A Directory of Useful Decoys <http://dud.docking.org/>) ที่รวบรวม ข้อมูลของลิแกนต์ และตัวหลอก ของโปรตีนต่าง ๆ ซึ่งลิแกนต์ (ligand) คือโมเลกุลที่จับตัวกับโปรตีนที่สนใจ ส่วนตัวหลอก (decoy) คือโมเลกุลที่ไม่จับตัวกับโปรตีนที่สนใจ.

จะเลือกโปรตีนเป้าหมายจากฐานข้อมูลดียูดี และสร้างแบบจำลองการทำนายการจับตัวกันของโปรตีนเป้าหมาย กับโมเลกุลขนาดเล็ก โดยจะสร้างเป็นแบบจำลองเฉพาะสำหรับโปรตีนนั้น และใช้คุณลักษณะต่าง ๆ ของโมเลกุลขนาดเล็ก เพื่อทำนายว่าโมเลกุลจะสามารถจับกับโปรตีนเป้าหมายได้หรือไม่. ทดสอบ วิเคราะห์ ภัยประยุต แล้วสรุป. หมายเหตุ การสร้างแบบจำลองทั่วไปที่สามารถทำนายการจับตัวระหว่างโมเลกุลกับโปรตีนได้ มีความท้าทายมาก และคู่ควรกับโครงการวิจัยระยะยาว (งานวิจัยของคณะของชานเชซ[175] เอง ก็เป็นการสร้างแบบจำลองเฉพาะกับแต่ละโปรตีน). ดังนั้น เพื่อให้เหมาะสมกับเนื้อหา ระดับความยาก และเวลา แบบฝึกหัดนี้จำกัดปัญหาเป็นการสร้างแบบจำลองเฉพาะโปรตีนก่อน.

การกิจนี้ มีผลทำนายเป็นสองสถานะ คือ จับตัวกัน หรือไม่จับตัวกัน. ดังนั้น ภาระกิจนี้ควรวางแผนเป็นงานการจำแนกค่าทวิภาค. อินพุตของแบบจำลองเป็นคุณลักษณะของโมเลกุล ซึ่งคณะของชานเชช[175] ใช้ไลบรารี่เคมอย[27] <http://code.google.com/p/pychem/downloads/list>) ช่วยในการจัดเตรียมคุณลักษณะของโมเลกุล จากข้อมูลรูปแบบโมลสอง (Tripos's mol-2 format) ที่ได้จากฐานข้อมูลดียูดี. แต่ตัวอย่างที่จะแสดงต่อไปนี้ เลือกที่จะจัดเตรียมคุณลักษณะต่าง ๆ ของโมเลกุลเอง โดยเลือกทำเฉพาะคุณลักษณะง่าย ๆ ที่ไม่ซับซ้อนจนเกินไป. ผู้อ่านอาจทดลองไลบรารี่เคมอย หรือไลบรารี่ที่เกี่ยวข้องอื่น ๆ เช่น อาร์ดีคิต (RDKit <http://www.rdkit.org>) หากสนใจ.

ฐานข้อมูลดียูดี มีข้อมูลของโปรตีนสำคัญ ๆ อยู่หลายตัว ([dud.docking.org/r2/](http://dud.docking.org/r2/)) เช่น แอนจิโอลีนซินคอนเวอร์ตติง เอนไซม์ (Angiotensin-converting enzyme), อะเซติลโคเลิน เอสเตอเรส (Acetylcholine esterase) รวมถึง เอชเอมจี โคเอ ริดักเตส (Hydroxymethylglutaryl-CoA reductase) และไทโรซีนคิเนส ชาร์ค (Tyrosine kinase SRC). ตัวอย่างต่อไปนี้ เลือกเป้าหมายเป็นโปรตีนไทโรซีนคิเนสชาร์ค ซึ่งเป็นเอนไซม์ที่เกี่ยวข้องกับมะเร็งเนื้อเยื่อกีบพัน (sarcoma). รายการ 3.19 แสดงโปรแกรมสำหรับนำเข้าข้อมูล โดยโปรแกรมรับชื่อไฟล์ข้อมูล (พาร์มเมต้าฟิลด์) ด้วยอาร์กิวเมนต์ `cpath` และรีเทิร์นลิสต์ของดิกชันนารีอ กมา. ลิแกนต์และตัวหลอก ถูกโหลดได้ด้วยคำสั่ง เช่น

```
ligands = load_compounds('databases/dud_decoys2006/src_decoys.mol2')
decoys = load_compounds('databases/dud_ligands2006/src_ligands.mol2')
```

สำหรับโปรตีนไทโรซีนคิเนสชาร์คนี้ ข้อมูลลิแกนต์มีอยู่ 159 โมเลกุล และข้อมูลตัวหลอกมีอยู่ 6319 โมเลกุล. โมเลกุลแต่ละตัว จะมีข้อมูลอยู่สามชนิดคือ ข้อมูลทั่วไปของโมเลกุล ข้อมูลของอะตอมต่าง ๆ ในโมเลกุล และข้อมูลของพันธะที่เชื่อมอะตอมต่าง ๆ ซึ่งสามารถเข้าถึงได้ด้วยคำสั่ง เช่น `ligands[0]['MOLECULE']` หรือ `ligands[0]['ATOM']` หรือ `ligands[0]['BOND']` สำหรับข้อมูลต่าง ๆ ของลิแกนต์โมเลกุลแรก (ลำดับที่ศูนย์). รูปแบบไฟล์ข้อมูลโมลสอง และคำอธิบาย สามารถศึกษาเพิ่มเติมได้จากเอกสารประกอบ Tripos Mol2 SYBYL 7.1 (Mid-2005) ที่สามารถค้นหาได้จากอินเตอร์เนต.

รายการ 3.19: โปรแกรมโหลดข้อมูลสารประกอบ

---

```
1 def load_compounds(cpath):
2     with open(cpath, 'r') as f:
3         ctxt = f.read()
4
5     field_name = 'dummy key'
6     v = 'dummy value'
```



```

48                         *atom_info[5:]) # type and others
49                         v.append(row)
50             elif field_name == 'BOND':
51                 if field_count == 0:
52                     v = []
53                 bond_info = line.split()
54                 if len(bond_info) > 3:
55                     row = [int(bond_info[0]), # id
56                            int(bond_info[1]), # atom1
57                            int(bond_info[2]), # atom2
58                            bond_info[3]] # bond_type
59                     v.append(row)
60                 field_count += 1
61             # end for Line
62         d[field_name] = v
63         compounds.append(d)
64     return compounds

```

---

จากข้อมูลที่ได้นำเข้ามา ตัวอย่างนี้เลือกแปลงข้อมูลของโมเลกุลเป็นลักษณะสำคัญเชิงเลข โดยใช้จำนวนอะตอม จำนวนพันธะ จำนวนอะตอมคาร์บอน จำนวนอะตอมไฮโดรเจน จำนวนอะตอมออกซิเจน จำนวนอะตอมไนโตรเจน จำนวนอะตอมกำมะถัน จำนวนพันธะเดี่ยว จำนวนพันธะคู่ จำนวนพันธะสาม จำนวนพันธะเอนไซด์ และ จำนวนพันธะอะโรมาติก ดังโปรแกรม **compound\_feat1** ในรายการ 3.20. ข้อมูลลิแกนต์และตัวหลอก (ตัวแปร **xlig** และ **xdec** ตามลำดับ) เตรียมได้ดังตัวอย่างคำสั่ง

```

xlig = np.zeros((12,0))
for c in ligands:
    xi = compound_feat1(c)
    xlig = np.hstack((xlig, xi))
xdec = np.zeros((12,0))
for c in decoys:
    xi = compound_feat1(c)
    xdec = np.hstack((xdec, xi))

```

โปรแกรม **compound\_feat1** เรียกใช้ **count\_elements** และ **count\_bonds** ซึ่งแสดงในรายการ 3.21.

รายการ 3.20: ตัวอย่างโปรแกรมเลือกลักษณะสำคัญของโมเลกุล

---

```

1 def compound_feat1(c):
2     feat = np.zeros((12, 1))

```

```

3     feat[0] = c['MOLECULE']['num_atom']
4     feat[1] = c['MOLECULE']['num_bonds']
5
6     celements = count_elements(c)
7     feat[2] = celements['C']
8     feat[3] = celements['H']
9     if 'O' in celements.keys():
10        feat[4] = celements['O']
11    if 'N' in celements.keys():
12        feat[5] = celements['N']
13    if 'S' in celements.keys():
14        feat[6] = celements['S']
15
16    cbonds = count_bonds(c)
17    feat[7] = cbonds['1']
18    feat[8] = cbonds['2']
19    feat[9] = cbonds['3']
20    feat[10] = cbonds['am']
21    feat[11] = cbonds['ar']
22    return feat

```

---

รายการ 3.21: ตัวอย่างโปรแกรมนับอะตอมและนับพันธะ

```

1 import re
2 def count_elements(c):
3     elements = {}
4     for r in c['ATOM']:
5         mr = re.match('[A-Za-z]+', r[1])
6         if not mr:
7             print('No element!', r[1])
8             continue
9         else:
10            e = mr.group(0)
11            if e not in elements.keys():
12                elements[e] = 1
13            else:
14                elements[e] += 1
15    return elements
16 def count_bonds(c):
17     bonds = {'1': 0, '2': 0, '3': 0, 'am': 0, 'ar': 0}
18     for b in c['BOND']:
19         bond_type = b[3]

```

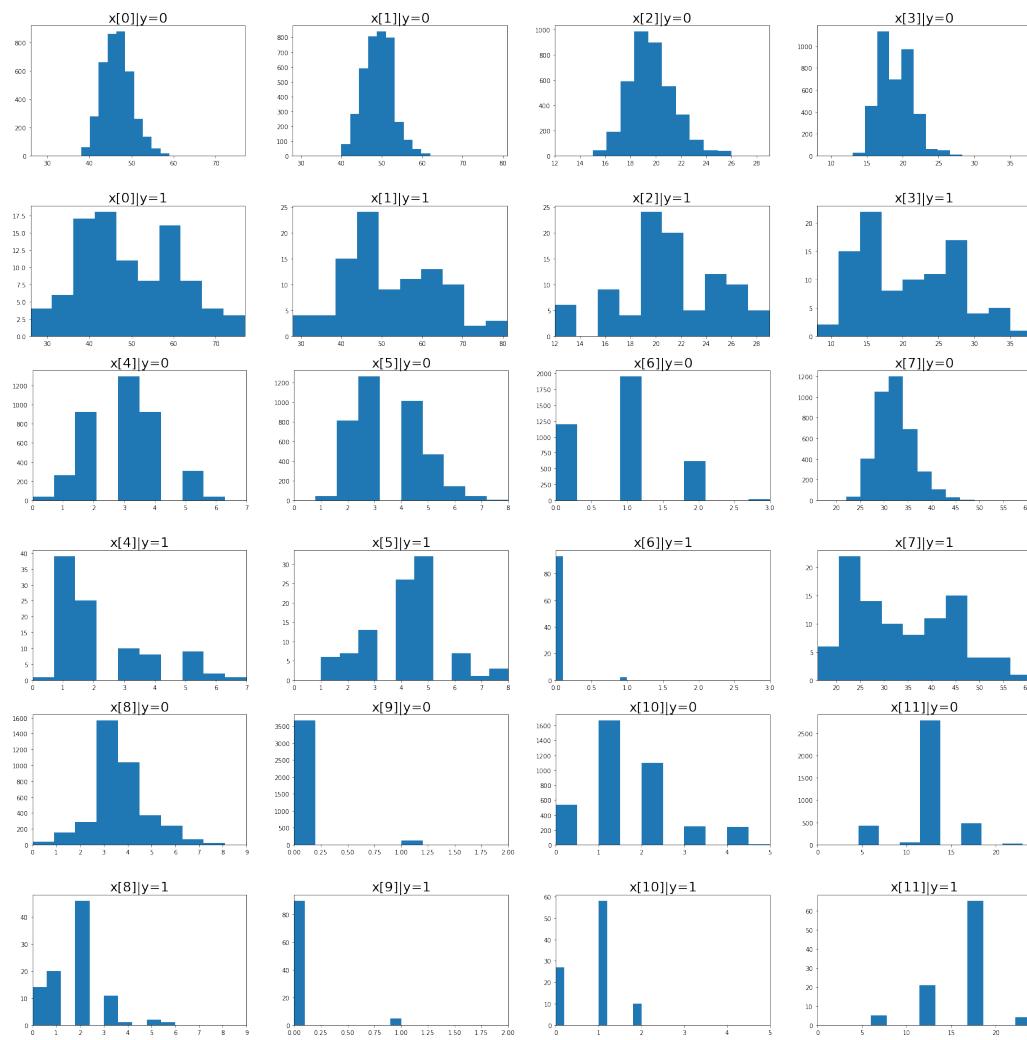
```
20     if bond_type in bonds.keys():
21         bonds[bond_type] += 1
22     else:
23         print('undefined bond:', bond_type)
24 return bonds
```

เนื่องจากสัดส่วนจำนวนข้อมูลลิแกนต์ ต่างจากจำนวนข้อมูลตัวหลอกมาก ตัวอย่างคำสั่งข้างล่าง แบ่งข้อมูลประมาณ 60% สำหรับการฝึก และที่เหลือสำหรับการทดสอบ และรวมข้อมูลลิแกนต์และตัวหลอกเข้าด้วยกัน

```
_, Nlig = xlig.shape
_, Ndec = xdec.shape
ids_lig = np.random.choice(Nlig, Nlig, replace=False)
ids_dec = np.random.choice(Ndec, Ndec, replace=False)
mark_lig = round(Nlig * 0.6)
trainx_lig = xlig[:, ids_lig[:mark_lig]]
testx_lig = xlig[:, ids_lig[mark_lig:]]
mark_dec = round(Ndec * 0.6)
trainx_dec = xdec[:, ids_dec[:mark_dec]]
testx_dec = xdec[:, ids_dec[mark_dec:]]

# Combine ligands and decoys
_, N1 = trainx_lig.shape
_, N0 = trainx_dec.shape
trainx = np.hstack((trainx_lig, trainx_dec))
trainy = np.hstack((np.ones((1, N1)), np.zeros((1, N0))))

_, N1 = testx_lig.shape
_, N0 = testx_dec.shape
testx = np.hstack((testx_lig, testx_dec))
testy = np.hstack((np.ones((1, N1)), np.zeros((1, N0))))
```



รูปที่ 3.42: การแจกแจงของลักษณะสำคัญเชิงเลข ทั้ง 12 ลักษณะสำคัญ ( $x[0]$  ถึง  $x[11]$ ) ของข้อมูลไม่เลกุลสารประกอบทั้งตัวแปร ( $y=1$ ) และตัวหลอก ( $y=0$ ).

รูป 3.42 แสดงการแจกแจงของข้อมูลฝึก. ค่าอินพุตมีช่วงค่อนข้างกว้าง คำสั่งข้างล่างแสดงตัวอย่างการทำอิร์โมลีซินพุต เตรียมแบบจำลองโครงข่ายประสาทเทียมสองชั้นขนาด 8 หน่วยซ่อน และฝึก 500 สมัยด้วยอัตราเรียนรู้ 0.1.

```
trainxn, normpars = normalize2(trainx)
num_epochs = 500
learn_rate = 0.1
net = w_initn([12, 8, 1])
net['act1'] = sigmoid
net['act2'] = sigmoid
trained_net, train_losses = train_mlp(net, trainxn, trainy,
                                       binaries_cross_entropy, lr=learn_rate, epochs=num_epochs)
```

ตัวอย่างคำสั่งข้างล่างทำการทดสอบผลการทำงาน

```
testxn, _ = normalize2(testx, normpars)
Yp = mlp(trained_net, testxn)
Yc = cutoff(Yp)
accuracy = np.mean(Yc == testy)
```

ผลลัพธ์ของตัวอย่าง แสดงค่าความแม่นยำออกมาที่ 97.5%. หมายเหตุ ผลลัพธ์ที่ทดลองแต่ละครั้งอาจแสดงค่าที่ต่างกันไปเนื่องจากผลของการสุ่ม ซึ่งอยู่ในกระบวนการแบ่งข้อมูล และการกำหนดค่าน้ำหนักเริ่มต้น. ดังนั้น หากต้องการศึกษาปัจจัยที่เกี่ยวข้องอย่างสมบูรณ์ ควรทำการทดลองซ้ำ โดยให้จำนวนทำซ้ำมากพอ<sup>4</sup> เพื่อ印ยันผลว่าความต่างของผลลัพธ์เป็นผลมาจากการปัจจัยที่ต่างกันจริง ๆ ไม่ใช่มาจากความแปรปรวนของข้อมูล หรือความแปรปรวนจากกระบวนการสุ่ม. แต่เพื่อความกระชับ ตัวอย่างนี้ไม่ได้ทำซ้ำ.

ค่าความแม่นยำที่ได้ แม้จะดูดีมาก แต่เมื่อพิจารณาเมทริกซ์ความสับสนที่ได้ (ดังแสดงข้างล่าง) แล้วจะพบว่าแบบจำลองนี้ล้มเหลว เพราะมันไม่ระบุสารประกอบใดที่อาจจับตัวกับเป้าหมายเลย.

		ผลจริง	
		1	0
ผลทำนาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	0	0
		จำนวนลบเท็จ	จำนวนลบจริง
	0	64	2528

สังเกตว่า เมื่อสัดส่วนจำนวนข้อมูลต่างกันมาก แบบจำลองเพียงทำนายว่า ไม่จับตัวกับเป้าหมาย กับทุก ๆ สารประกอบ ก็สามารถจะได้ค่าความแม่นยำที่สูงมากได้. แต่เมื่อพิจารณาค่าความเที่ยงตรง และค่าการระลึกกลับ ซึ่งเป็น 0/0 และ 0 ตามลำดับ จะพบว่า ความเที่ยงตรง และการระลึกกลับ สะท้อนความล้มเหลวของแบบจำลองทำนายได้ชัดเจนมาก.

<sup>4</sup> ประดิ่นเรื่องจำนวนข้าม นิหลักการอยู่ว่า จำนวนข้ามต้องมากพอ ที่หลักการทางสถิติ เช่น การทดสอบนัยสำคัญ (significance test) สามารถยืนยันความต่างได้ หากความต่างมีจริง. แต่หากการทดสอบนัยสำคัญ ไม่สามารถยืนยันความต่างได้ อาจหมายความได้ว่า (1) ผลที่เปรียบเทียบกันนั้นไม่ได้ต่างกันจริง ๆ ความต่างที่สังเกตเป็นเพียงความแปรปรวนของข้อมูล หรือ (2) ผลที่เปรียบเทียบอาจต่างกันจริง ๆ เพียงแต่ ด้วยจำนวนข้อมูลหรือจำนวนข้ามที่มี ไม่สามารถยืนยันได้. นั่นหมายความว่า หากเลือกจำนวนข้ามแล้ว การทดสอบนัยสำคัญสามารถยืนยันความต่างได้ แปลว่าจำนวนข้ามที่เลือกนั้นเพียงพอ. แต่หากเลือกจำนวนข้ามแล้ว การทดสอบนัยสำคัญไม่สามารถยืนยันความต่างได้ อาจแปลว่า (1) จำนวนข้ามที่เลือกนั้นไม่เพียงพอ ควรเพิ่มจำนวนข้าม หรืออาจแปลว่า (2) ผลที่เปรียบเทียบไม่ได้ต่างกัน. ดังนั้น ในทางปฏิบัติ หากการทดสอบนัยสำคัญ ยังไม่สามารถยืนยันความต่างได้ ผู้ทดลองอาจเลือกเพิ่มจำนวนข้าม หากผู้ทดลองเชื่อว่าเป็นกรณีแรก หรือผู้ทดลองอาจเลือกจบการทดลอง และสรุปว่าการทดสอบนัยสำคัญไม่สามารถยืนยันความต่างได้ ที่ความมั่นใจที่ระบุ เมื่อใช้จำนวนข้ามที่เลือก. สังเกตว่า การทดสอบนัยสำคัญ จะสามารถยืนยันความต่างได้ แต่ไม่สามารถยืนยันความเหมือน (หรือความไม่ต่าง). นั่นคือ หากการทดสอบนัยสำคัญยืนยันว่าผลต่างกันจริง หมายถึงผลต่างกันจริง ๆ. แต่หากการทดสอบนัยสำคัญไม่สามารถยืนยันความต่าง อาจแปลว่าหลักฐานไม่พอ หรืออาจแปลว่าผลไม่ต่างกัน.

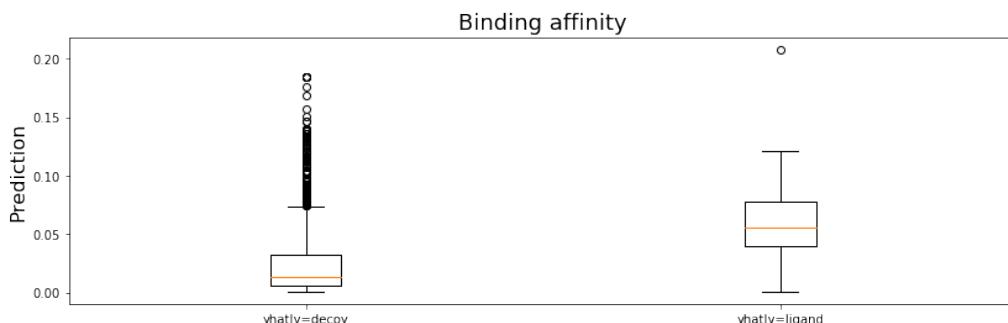
ก่อนจะอภิปรายเรื่องวิธีจัดการกับปัญหาลักษณะส่วนจำนวนข้อมูลไม่สมดุล พิจารณาค่าเออต์พูตที่ได้จากแบบจำลอง สำหรับกรณีของลิแกนต์และตัวหลอก. รูป 3.43 แสดงให้เห็นว่า ค่าเออต์พูตที่มากที่สุด มีค่าอยู่แค่ประมาณ 0.2. ค่าเออต์พูตที่ได้จะถูกตัดสินใจสุดท้ายด้วย โปรแกรม **cutoff** (ดูแบบฝึกหัด 3.15) ที่ค่าดีฟอลต์คือตัดทายหนึ่งที่ 0.5. ดังนั้น ค่าใด ๆ ที่น้อยกว่า 0.5 จะถูกตัดสินใจเป็นศูนย์ และทำให้ทุก ๆ สารประกอบถูกทายเป็นศูนย์ (หรือทายว่าไม่จับตัวกับเป้าหมาย). แต่เมื่อพิจารณารูป 3.43 โดยเฉพาะค่าความต่างระหว่างเออต์พูตที่ได้สำหรับลิแกนต์ เปรียบเทียบกับตัวหลอก จะพบว่า แม้ทั้งคู่จะมีค่าต่ำ แต่ค่าเออต์พูตสำหรับลิแกนต์ส่วนใหญ่ ก็มากกว่าค่าเออต์พูตสำหรับตัวหลอก ค่อนข้างชัดเจน. ดังนั้น ความล้มเหลวของการทำนายนี้ อาจบรรเทาได้เพียงแค่การปรับระดับค่าขีดแบ่ง (threshold) ลง.

ผลติกรรมการทำนายของแบบจำลองจำแนกค่าทวิภาค สามารถถูกปรับแต่งได้จากการปรับระดับค่าขีดแบ่ง. รูป 3.44 แสดงค่าคะแนนเอฟ เมื่อใช้ระดับค่าขีดแบ่งต่าง ๆ. หมายเหตุ เพื่อลดความยุ่งยากจากการณ์ 0/0 ตัวหารของค่าความเที่ยงตรง และค่าคะแนนเอฟ คำนวณด้วยคำสั่งดังต่อไปนี้

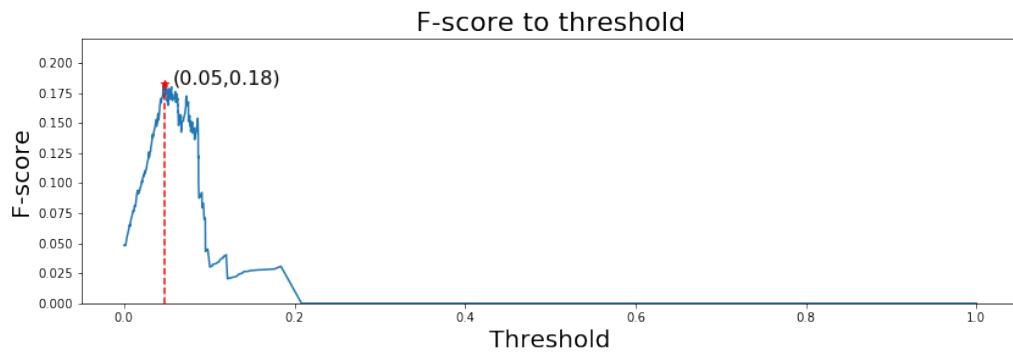
$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP} + 1e-12)$$

$$\text{fscore} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall} + 1e-12)$$

เมื่อ **TP**, **FP**, และ **Recall** เป็นจำนวนบวกจริง, จำนวนบวกเท็จ, และค่าการระลีกกลับ ตามลำดับ. ค่า **1e-12** เป็นค่าน้อย ๆ ที่เพิ่มเข้าไป ซึ่งจะเปลี่ยนกรณี 0/0 เป็น 0 และไม่รบกวนกรณีอื่นๆมาก.



รูปที่ 3.43: แผนภูมิกล่องแสดงตัวอย่างผลจากแบบจำลองที่ทำนายการจับตัวกับโปรตีน สำหรับลิแกนต์และตัวหลอก.



รูปที่ 3.44: ค่าคะแนนเอฟของการทำนายการจับตัว เมื่อใช้ระดับค่าขีดแบ่งต่าง ๆ.

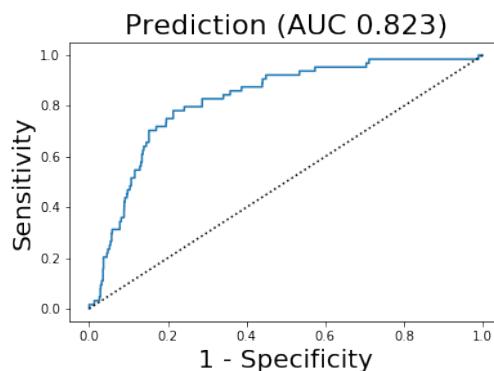
จากรูป 3.44 ค่าคะแนนเอฟจะสูงสุด เมื่อเลือกใช้ระดับค่าขีดแบ่งประมาณ 0.05 และเมื่อเลือกระดับค่าขีดแบ่งที่ประมาณ 0.05 แล้วจะได้ผลทดสอบดังเมทริกซ์ความสับสน

		ผลจริง	
		1	0
ผลทำนาย	1	จำนวนบวกจริง 45	จำนวนบวกเท็จ 384
	0	จำนวนลบเท็จ 19	จำนวนลบจริง 2144

และได้ค่าความเที่ยงตรง 0.105 ค่าการระลึกกลับ 0.703 และค่าคะแนนเอฟ 0.183. ผลลัพธ์ที่ได้ แม้จะยังแย่ แต่ก็ดีขึ้นกว่าเดิมมาก. นอกจากนี้ ระดับค่าขีดแบ่ง ที่สามารถเลือกปรับให้เหมาะสมกับความชอบส่วนบุคคล หรือให้เหมาะสมกับสถานการณ์ได้ เช่น บางภาระกิจ อาจเลือก บวกเท็จดีกว่าลบเท็จ (เช่น หากทรัพยากรเพียงพอ ได้ตัวหลอกเกินมา ดีกว่าขาดลิแกนต์ไป) ในขณะที่บางภาระกิจ อาจเลือก ลบเท็จดีกว่าบวกเท็จ (เช่น เมื่อทรัพยากรจำกัดมาก ๆ ตกลิแกนต์ไปบ้าง ดีกว่าได้ตัวหลอกมา และเปลี่ยนค่าใช้จ่ายในขั้นตอนการพัฒนาต่อไปเป็นล่า ๆ). เนื่องจากการเลือกระดับค่าขีดแบ่ง มีผลต่อการทำนายมาก และยังอาจถูกปรับให้เหมาะสมกับความชอบส่วนบุคคลได้ การประเมินแบบจำลอง บางครั้งจึงนิยมใช้กราฟอร์โวชี (Receiver Operating Characteristic คำย่อ ROC) และพื้นที่ใต้เส้นโค้ง (Area Under Curve คำย่อ AUC). กราฟอร์โวชี หมายถึง กราฟแสดงผลการทำนาย โดยอาจเลือกใช้ต้นนิ้วด้วยหลายแบบ เช่น อาจใช้กราฟระหว่างค่าความเที่ยงตรงกับค่าการระลึกกลับ (precision-recall plot) หรืออาจใช้กราฟระหว่างค่าอัตราการตรวจจับได้ อาจเรียก

ว่าค่าความไว (sensitivity หรือ true positive rate)  $S_1 = \text{Recall} = TP / (TP + FN)$  เมื่อ  $TP$  และ  $FN$  คือจำนวนบวกจริง และจำนวนลบเท็จ ตามลำดับ. อัตราสัญญาณหลอก (false alarm rate)  $FAR = FP / (TN + FP)$  หรือ  $FAR = 1 - S_2$  เมื่อ  $S_2$  คือค่าความจำเพาะ (specificity หรือ true negative rate) ซึ่ง  $S_2 = TN / (TN + FP)$  โดย  $TN$  และ  $FP$  คือจำนวนบวกจริง และจำนวนบวกเท็จ ตามลำดับ.

รูป 3.45 แสดงกราฟระหว่างค่าความไวกับอัตราสัญญาณหลอก จากผลตัวอย่าง. จุดต่าง ๆ บนเส้นกราฟคำนวณโดยการปรับระดับค่าขีดแบ่งจากน้อยที่สุดไปมากที่สุด และประเมินผลการทำนายสำหรับแต่ละระดับค่าขีดแบ่ง. พื้นที่ใต้เส้นโค้ง ก็คือพื้นที่ใต้กราฟอาร์โอดีที่เลือกใช้. รูป 3.45 แสดงค่าพื้นที่ใต้เส้นโค้ง กำกับไว้หน้าภาพ. ค่าพื้นที่ใต้เส้นโค้งที่ใกล้หนึ่ง แสดงถึงคุณภาพการทำนายที่ดีของแบบจำลอง.



รูปที่ 3.45: กราฟระหว่างค่าความไว (Sensitivity) กับอัตราสัญญาณหลอก (1 - Specificity) ของตัวอย่างผลการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีนไทรอีโนไซด์. ค่าพื้นที่ใต้เส้นโค้ง (AUC) แสดงหนึ่งภาพ.

สำหรับการกิจกรรมจำแนกค่าทวิภาค หรือการจำแนกกลุ่ม เมื่อจำนวนจุดข้อมูลของแต่ละกลุ่มข้อมูลต่างกันมาก จะเกิดปัญหาสัดส่วนจำนวนข้อมูลไม่สมดุล (unbalanced data) ขึ้น. วิธีจัดการปัญหาสัดส่วนจำนวนข้อมูลไม่สมดุลในชุดข้อมูลฝึก สามารถทำได้หลายวิธี<sup>[34]</sup> เช่น วิธีการสุมเกิน (over sampling), วิธีการสุมขาด (under sampling), วิธีปรับฟังก์ชันจุดประสงค์. วิธีการสุมเกิน ใช้การสุมแบบคืนที่ (sampling with replacement) เพื่อเพิ่มจุดข้อมูลของกลุ่มน้อยขึ้นมาให้ใกล้เคียงกับกลุ่มใหญ่. วิธีการสุมขาด ใช้การสุมเลือกบางส่วนของข้อมูลจากกลุ่มใหญ่ เพื่อให้ข้อมูลของที่ใช้ฝึกของกลุ่มใหญ่มีจำนวนใกล้เคียงกับจำนวนของกลุ่มน้อย. วิธีปรับฟังก์ชันจุดประสงค์ ปรับการคำนวณค่าฟังก์ชันจุดประสงค์ โดยให้น้ำหนักความสำคัญกับการทำนายกลุ่มน้อยมากขึ้น (หรือลดน้ำหนักความสำคัญของการทำนายกลุ่มใหญ่ลง หรือทำทั้งสองทาง) เพื่อชดเชยกับจำนวนข้อมูลที่ต่างกัน.

ตัวอย่างที่จะแสดงต่อไปนี้ ใช้วิธีการสุมเกิน ซึ่งสัดส่วนความต่างกันของจำนวนข้อมูลทั้งสองกลุ่ม คือ

$3791/95 \approx 40$  เท่า เมื่อ 3791 และ 95 คือจำนวนจุดข้อมูลในชุดฝึกของกลุ่มใหญ่ (ตัวหลอก) และของกลุ่มน้อย (ลิแกนต์) ตามลำดับ. ตัวอย่างนี้ เลือกเพิ่มจำนวนในกลุ่มน้อยขึ้นมาเป็นประมาณ 80% ของจำนวนในกลุ่มใหญ่ ดังแสดงในคำสั่งข้างล่าง

```
over_factor = int(np.floor(N0/N1 * 0.8))
ids = np.random.choice(N1, over_factor * N1, replace=True)
trainx = np.hstack((trainx_lig[:, ids], trainx_dec))
trainy = np.hstack((np.ones((1, over_factor*N1)), np.zeros((1, N0))))
```

เมื่อ **N0** คือจำนวนข้อมูลฝึกในกลุ่มใหญ่ และ **N1** คือจำนวนข้อมูลฝึกในกลุ่มน้อย. ตัวแปร **trainx\_lig** และ **trainx\_dec** คือตัวแปรค่าอินพุตของข้อมูลกลุ่มน้อย และของข้อมูลกลุ่มใหญ่ ตามลำดับ. ฉลากเฉลยของข้อมูลลิแกนต์ กำหนดให้มีค่าเป็นหนึ่ง (ลิแกนต์ คือสารประกอบที่จับตัวกับโปรตีนเป้าหมาย) และฉลากเฉลยของข้อมูลตัวหลอก กำหนดให้มีค่าเป็นศูนย์ (ตัวหลอก คือสารประกอบที่ไม่จับตัวกับโปรตีนเป้าหมาย). ข้อมูลฝึกหลังทำการสุ่มเกินเพื่อปรับเพิ่มจำนวนข้อมูลกลุ่มน้อย คือ **trainx** (อินพุต) และ **trainy** (เอาต์พุต เฉลย).

หลังจากทำการน้อมอไลซ์อินพุต ฝึกแบบจำลอง<sup>5</sup> และทดสอบแล้ว ผลที่ได้แสดงดังเมทริกซ์ความสับสน

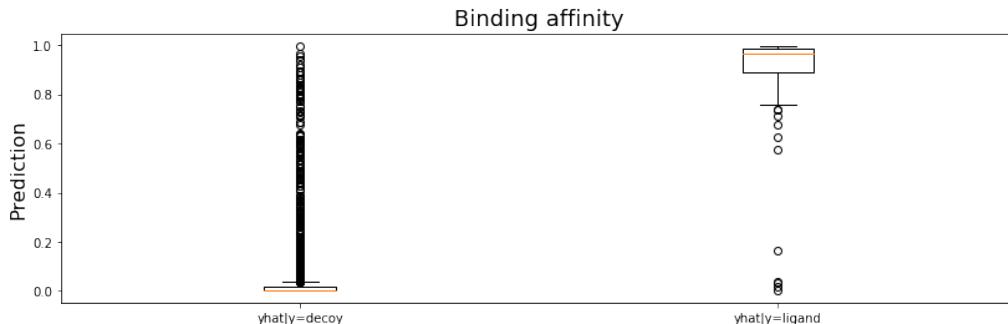
ผลจริง

		1	0
		จำนวนบวกจริง	จำนวนบวกเท็จ
ผลทำนาย	1	59	87
	0	5	2441

ค่าความเที่ยงตรง 0.404 ค่าการระลึกกลับ 0.922 และค่าคะแนนเอฟ 0.562 ซึ่งปรับปรุงขึ้นมาก. ผลประเมินนี้ได้จากการตัดสินผลทำนายด้วยระดับค่าขีดแบ่ง 0.5. ในลักษณะเดียวกัน รูป 3.46 แสดงแผนภูมิกล่องของค่าเอาต์พุตจากแบบจำลอง สำหรับข้อมูลกลุ่มน้อย และกลุ่มใหญ่. สังเกตว่า เอาต์พุตจากแบบจำลอง

<sup>5</sup> ตัวอย่างนี้ ฝึกโครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยชั้น บน 10000 สมัย ด้วยอัตราเรียนรู้ 0.1. การฝึกครั้งนี้ใช้จำนวนสมัยฝึกมากกว่า จำนวนสมัยของการฝึกกับข้อมูลที่ไม่มีการจัดการข้อมูลไม่สมดุล เนื่องจาก การฝึกควรทำงานการฝึกสมบูรณ์ หรือค่อนข้างสมบูรณ์ โดยพิจารณาจากความก้าวหน้าของการฝึก (**train\_losses**). จากความก้าวหน้าของการฝึกที่ได้ กรณีการสุ่มเกิน ไม่สามารถฝึกแค่ 500 สมัย ได้ (เพราะการฝึกดูยังห่างความสมบูรณ์อยู่มาก) แต่กรณีการไม่ทำอะไร สามารถฝึก 10000 สมัยได. อย่างไรก็ตาม ผู้เขียนเห็นว่า การทดลองดังผลที่นำเสนอในกระชับ และเปิดโอกาสให้เห็นความเสี่ยงจากการพึงค่าความแม่นยำเพียงอย่างเดียว รวมถึงชี้ความสำคัญของการตรวจสอบผลที่ได้ ซึ่งน่าจะเป็นประโยชน์มากกว่า. การทดลองโดยใช้จำนวนสมัยฝึกพอ ๆ กันสามารถทำได้ และผู้เขียนพบว่า ได้ผลลัพธ์ในทิศทางเดียวกัน เพียงแต่ผลต่างอาจไม่เด่นชัดเท่าที่นำเสนอในตัวอย่างนี้.

มีช่วงค่าแยกกันชัดเจนมากระหว่างข้อมูลกลุ่มน้อย (ค่าใกล้หนึ่ง) และข้อมูลกลุ่มใหญ่ (ค่าใกล้ศูนย์) แม้จะมีค่าผิดปกติบ้าง. ในทางสถิติ ค่าผิดปกติ (outliers) หมายถึง ค่าของจุดข้อมูลจำนวนน้อย ที่มีค่าต่างจากค่าของจุดข้อมูลอื่น ๆ ในกลุ่มอย่างมาก.

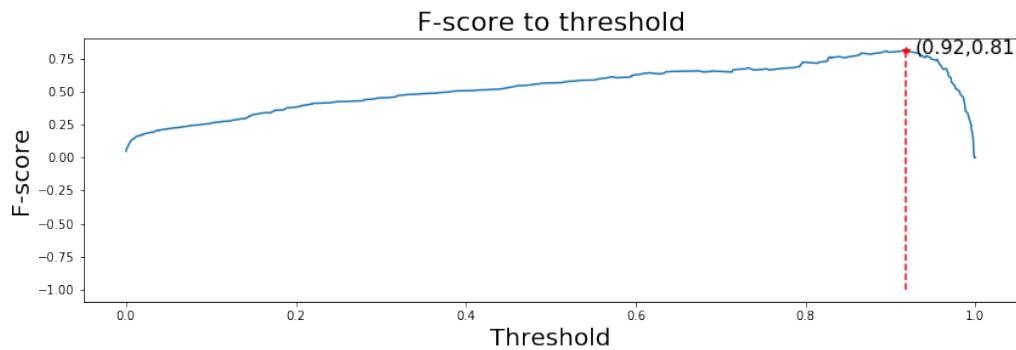


รูปที่ 3.46: แผนภูมิกล่องของค่าเออตพุตจากแบบจำลอง สำหรับข้อมูลกลุ่มใหญ่ (*decoy*) และข้อมูลกลุ่มน้อย (*ligand*) เมื่อใช้วิธีการสุ่มเกิน เพื่อจัดการปัญหาจำนวนข้อมูลไม่สมดุล.

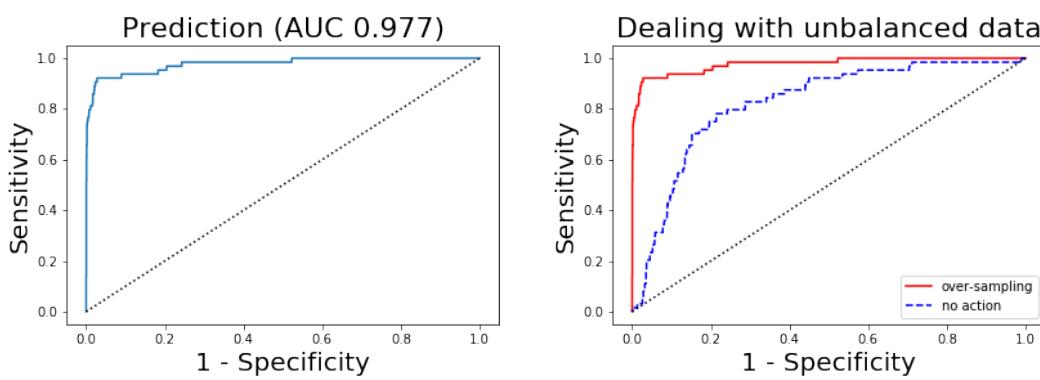
จากรูป 3.46 การตัดสินผลทำนาย อาจสามารถปรับปรุงได้ง่าย ๆ ด้วยการเปลี่ยนระดับค่าขีดแบ่ง. พิจารณาที่รูป 3.47 ซึ่งแสดงค่าคะແນນເອີມ ที่ระดับค่าขีดแบ่งต่าง ๆ และเมื่อเปลี่ยนระดับค่าขีดแบ่งเป็นประมาณ 0.92 จะได้เมทริกซ์ความสัมสุน

		ผลจริง	
		1	0
ผลทำนาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	47	5
		จำนวนลบเท็จ	จำนวนลบจริง
		17	2523

ค่าความเที่ยงตรง 0.904 ค่าการระลึกกลับ 0.734 และค่าคະແນນເອີມ 0.810 ซึ่งโดยทั่วไปแล้ว ค่าคະແນນເອີມขนาดนี้ ถือว่าแบบจำลองสามารถทำงานได้ดีพอสมควร. รูป 3.48 แสดงแสดงกราฟระหว่างค่าความไวกับอัตราสัญญาณหลอก พื้นที่พื้นที่ได้เส้นโค้ง เมื่อใช้วิธีสุ่มเกิน (ກາພ້າຍ) และບວລິຍບເທີຍບກັບການໄມ່ທຳອະໄຮເລຍ (ກາພ້າວ). จากຕ້ວຍຢ່າງຂ້າງຕົ້ນ ວິຊີສຸ່ມເກີນສາມາດช่วยປັບປຸງຄຸນກາພຂອງການເທີຍມແບບจำลองทำนาย ในกรณีจำนวนข้อมูลไม่สมดุลได้อย่างชัดเจน.



รูปที่ 3.47: ค่าคะแนนเอฟ ที่ระดับค่าขีดเบ่งต่าง ๆ เมื่อใช้วิธีการสุ่มเกิน เพื่อจัดการปัญหาจำนวนข้อมูลไม่สมดุล.



รูปที่ 3.48: ภาพซ้าย แสดงกราฟระหว่างค่าความไวกับอัตราสัญญาณหลอก ของตัวอย่างการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีนไทรอีนคิโนเซาร์ค หลังปรับปรุงข้อมูลไม่สมดุลด้วยวิธีสุ่มเกิน และภาพขวา แสดงกราฟเปรียบเทียบระหว่างการไม่ทำอะไรเลยกับปัญหาข้อมูลไม่สมดุล (**no action**) กับการใช้วิธีสุ่มเกิน (**over-sampling**).

**เกร็ดความรู้การค้นหายา** (เรียบเรียงจาก [29] และ [66] และ [217]) ยา โดยทั่วไปคือ โมเลกุลที่กระตุ้นหรือยับยั้งการทำงานของชีวโมเลกุล เช่น โปรตีน ซึ่งส่งผลทางการรักษาโรคกับผู้ป่วย. แนวทางในการค้นหายาแบบดั้งเดิม อาจจะเริ่มด้วยการหาส่วนผสมออกฤทธิ์ (active ingredient) จากตัวรับยาดั้งเดิม เช่น ยาเรเซอร์พิน. รีเซอร์พิน (Reserpine) เป็นยาสำหรับบำบัดอาการความดันสูง ที่สะกัดจากรากของต้นระยองมน้อย (Rauvolfia serpentina หรือชื่อสามัญ Indian snakeroot) ที่อยู่ในตัวรับยาอายุรเวทของอินเดียแต่โบราณ. หรืออาจจะเริ่มด้วยการค้นหารายประกอบต่าง ๆ ที่ส่งผลที่ต้องการ จากการทดลองกับสัตว์ที่ป่วยเป็นโรค หรือจากการทดลองในหลอดทดลองกับเซลล์ที่เป็นโรค. สารประกอบที่พบจากการค้นหาเป็นต้น จะเรียกว่า สารประกอบหลัก. สารประกอบหลัก (lead compounds) คือ สารประกอบที่จากการทดลองแล้วพบว่าจะช่วยรักษาโรคได้ แต่โครงสร้างทางเคมีอาจจะยังไม่ดีเท่าไร. จากนั้น สารประกอบหลักต่าง ๆ ที่ได้ จะถูกดัดแปลงทางเคมี เพื่อปรับปรุง การออกฤทธิ์ (potency) และสมรรถนะการเลือก (selectivity) รวมถึงปรับปรุงคุณสมบัติทางเภสัชจลนศาสตร์อื่น ๆ ให้เหมาะสมที่จะเป็นยา และสามารถดำเนินการทำทดสอบกับสัตว์ทดลอง และทดสอบทางคลินิกได้ต่อไป. แนวทางในการค้นหายาแบบดั้งเดิมนี้ เริ่มจากการค้นหาสารประกอบหลักโดยสังเกตผลที่ได้โดยตรง และเมื่อพบสารประกอบหลักต่าง ๆ แล้วจึงค่อยศึกษาถกไกการทำงาน และชีวโมเลกุลต่าง ๆ ที่เกี่ยวข้องกับการทำงานของสารประกอบเหล่านั้น และนำความรู้ความเข้าใจที่ได้ กลับมาปรับปรุงโครงสร้างของสารประกอบหลัก เพื่อให้ได้สารประกอบที่มีคุณสมบัติทางยาที่ดีมากขึ้น. แต่แนวทางการพัฒนาฯ เช่น กลีเวค ที่อภิปรายไปในเกร็ดความรู้ รูปแบบ

ของลูคีเมียและยารักษา ดำเนินการกลับกัน คือ เริ่มจากการเข้าใจกลไกของโรค และวิถีที่เกี่ยวข้อง (biological pathway). จากนั้นเลือกชีวโมเลกุลในวิถีที่เกี่ยวข้องกับโรค เป็นเป้าหมาย. แล้วจึงออกแบบบลิแกนต์หรือโมเลกุลของสารประกอบที่จะเข้าไปจับกับชีวโมเลกุลเป้าหมาย เพื่อปรับการทำงานของโมเลกุลเป้าหมายในทางรักษาบรรเทาโรค. แนวทางหลังนี้ อาจเรียกว่า การค้นหายาแบบย้อนกลับ (reverse drug discovery) หรือ การค้นหายาโดยกำหนดเป้าหมาย (target-based drug discovery).

**การค้นหายาโดยกำหนดเป้าหมาย โรคหัวใจ คอเลสเตรอรอล และกลุ่มยาสแตติน.** แนวทางการค้นหายาโดยกำหนดเป้าหมาย เริ่มจากการเข้าใจกลไกของการทำงานของร่างกายและกลไกของโรค หรือเข้าใจเหตุก่อน. จากนั้นเลือกเป้าหมายที่อยู่ในกลไกซึ่งอาจเป็นโปรตีน หรือดีเอ็นเอ หรืออาร์เอ็นเอ แล้วจึงหาบลิแกนต์ ซึ่งคือโมเลกุลจะเข้าไปจับกับเป้าหมาย และเปลี่ยนการทำงานของเป้าหมาย ในทางที่จะช่วยแก้ไขกลไกที่เป็นเหตุของโรค. การค้นพบกลุ่มยาสแตติน (Statins) เป็นตัวอย่างหนึ่งของการค้นพบยาโดยกำหนดเป้าหมาย.

หลังสงครามโลกครั้งที่สองสงบ ยุโรปหลังสงครามล้างบาดาลมาก ขาดแคลนแพทย์ทุกอย่าง แต่ อันเชล คีส (Ancel Keys) นักผจญภัยและนักวิทยาศาสตร์ จากมินนิโซตา สหรัฐอเมริกา พบรัฐิติที่น่าสนใจ คือ สัตติการตายด้วยโรคหัวใจในยุโรปหลังสงครามลดลงอย่างมาก ขณะที่สัตติในอเมริกาสูงมาก. คีสสังสัย และศึกษาว่าอะไรเป็นปัจจัยต่อการตายด้วยโรคหัวใจ นอกจากนั้น ระหว่างท่องเที่ยว คีสพบว่า ชาวประมงในเนเปิล อิตาลี มีระดับคอเลสเตรอรอลในกระแสเลือดต่ำกว่าระดับคอเลสเตรอรอลของนักธุรกิจอเมริกันมาก ๆ. จากข้อมูลที่เห็น คีสเขียนไว้ว่า คนรวยกินอาหารที่อุดมด้วยไขมัน และก็หัวใจวายมากกว่า. แต่ตอนนั้น ส่วนใหญ่ไม่ได้เขียนแบบคีส.

ถึงแม้ว่าก่อนหน้านั้น มีงานศึกษาที่พบว่า หลอดเลือดแดงใหญ่จากเนื้อเยื่อผู้ป่วยโรคหัวใจเลือดแดงและหลอดเลือดแดงแข็ง มีคอเลสเตรอรอลมากกว่าที่เนื้อเยื่อปกติมี ถึงกว่าสี่สิบเท่า และถ้าให้สัตว์กินคอเลสเตรอรอลมาก ๆ แล้วมันจะป่วยเป็นโรคภาวะไขมันในเลือดสูง และโรคหัวใจเลือดแดงและหลอดเลือดแดงแข็ง แต่คนส่วนใหญ่ก็ยังไม่ค่อยมั่นใจเท่าไรว่า อาหาร ระดับคอเลสเตรอรอลในกระแสเลือด และโรคหัวใจ มันเกี่ยวข้องกัน. ดังนั้น คีสและเพื่อนักวิจัย ได้ร่วมกันทำโครงการวิจัยระดับนานาชาติ เพื่อศึกษาปัจจัยความเสี่ยงต่ออาการหัวใจวาย โดยครอบคลุมกลุ่มตัวอย่างมากกว่า 12,000 คน จากที่ต่าง ๆ ของโลก ยูโกรัสลาเวีย อิตาลี กรีก ฟินแลนด์ เนเธอร์แลนด์ ญี่ปุ่น และสหราชอาณาจักร ซึ่งแต่ละที่มีวัฒนธรรมอาหารการกินที่แตกต่างกันมาก. ผลการศึกษาพบ ความสัมพันธ์ระหว่างอาหารที่กินกับระดับคอเลสเตรอรอลในกระแสเลือด และยืนยันว่า ระดับคอเลสเตรอรอลในกระแสเลือดเป็นปัจจัยเสี่ยงหลักต่อการเป็นโรคหัวใจ คนที่มีระดับคอเลสเตรอรอลในกระแสเลือดสูงกว่า 260 มิลลิกรัมต่อเดซิลิตร จะมีโอกาสที่จะหัวใจวายเป็นห้าเท่าของคนที่มีระดับคอเลสเตรอรอลในกระแสเลือดต่ำกว่า 200 มิลลิกรัมต่อเดซิลิตร.

สิ่งหนึ่งที่ควรระลึก คือ เช่นเดียวกับ华丽 ๆ อย่างในธรรมชาติและชีวิต ไม่มีอะไรที่ดีหรือชั่วโดยสมบูรณ์. คอเลสเตรอรอลไม่ใช่สิ่งชั่วร้าย น่ารังเกียจ ที่ต้องกำจัดออกไปให้สิ้นเชิง ถอนหากถอนโคน. คอเลสเตรอรอลเป็นสิ่งที่จำเป็นกับชีวิต ร่างกายเราต้องการคอเลสเตรอรอล คอเลสเตรอรอลเป็นส่วนประกอบสำคัญของเยื่อหุ้มเซลล์ในเซลล์ของสัตว์ทุกชนิด (รวมถึงเซลล์ของเราด้วย). คอเลสเตรอรอลไม่ได้ชั่วร้าย เพียงแต่ ปริมาณของมันที่เกินระดับ จะสร้างปัญหา.

“All things are poison, and nothing is without poison,  
the dosage alone makes it so a thing is not a poison.”

“ทุกสิ่งล้วนเป็นพิษ ไม่มีสิ่งใดปราศจากพิษ  
ปริมาณเท่านั้นที่จะทำให้มันเป็นพิษ.”

—Paracelsus

—พาราเซลซัส

ช่วงปี ค.ศ. 1969-1970 ระหว่างที่นายแพทย์โจเซฟ โกลด์สไตน์ (Joe Goldstein) ทำงานที่โรงพยาบาลของสถาบันหัวใจแห่งชาติ (National Heart Institute) ในเมืองเบресด้า รัฐแมรี่แลนด์ สหรัฐอเมริกา โกลด์สไตน์ได้ดูแลคนไข้เด็กสองคนที่ป่วยเป็นโรคภาวะไขมันในเลือดสูงพันธุกรรม (familial hypercholesterolemia คำย่อ FH). เด็กทั้งสองเป็นพี่น้องกัน อายุแค่หกขวบกับแปดขวบเท่านั้น แต่มีคอเลสเตรอรอลในเลือดอยู่ในระดับสูงมาก คืออยู่ในช่วง 800 มิลลิกรัมต่อเดซิลิตร.

โกลด์สไตน์สนใจรีโนเวต และศึกษากรณีร่วมกับマイคลิ บราร์น (Michael Brown). ตอนนั้นในวงการแพทย์รู้อยู่แล้วว่า ร่างกายมีการสังเคราะห์คอเลสเตรอรอล และการสังเคราะห์คอเลสเตรอรอลเป็นกลไกการควบคุมแบบป้อนกลับ นั่นคือ ถ้าให้อาหารที่มีคอเลสเตรอรอลสูงกับสุนัข ร่างกายของสุนัขนั้นจะหยุดการสังเคราะห์คอเลสเตรอรอลลง. ความรู้นี้ ทำให้โกลด์สไตน์และบราร์น

สงสัยว่า เด็กทั้งสองอาจจะมีการผิดปกติในกลไกการควบคุมแบบป้อนกลับนี้.

ขณะที่เพื่อน ๆ ของโกลเด้นและบราร์น ส่วนใหญ่ศึกษาเรื่องมะเร็ง หรือประสาทวิทยา หรือเรื่องอื่น ๆ ที่อยู่ในกระแส แต่ทั้งโกลเด้นและบราร์น ตัดสินใจที่จะศึกษาเรื่องกลไกควบคุมคอเลสเตอรอลอย่างจริงจัง ถึงแม้เพื่อน ๆ ของเขายังชอบล้อเลียนว่า “มันก็แค่ก้อนหยุ่น ๆ ไวรัสปร่าง” โกลเด้นและบราร์น ได้ทำงานร่วมกันอย่างเป็นทางการ หลังจากทั้งคู่พยายามไปศูนย์การแพทย์ตัววัน ตกเลี้ยงใต้เมืองมหาวิทยาลัยเทกซัส. ระหว่างสองปีที่ทั้งคู่มุ่งมั่นทำงานหนัก บริษัทกลไกควบคุมคอเลสเตอรอลก็เฉลย.

โกลเด้นและบราร์น เริ่มสืบจากวิถีการสังเคราะห์คอเลสเตอรอลที่วงการแพทย์ตอนนั้นรู้ดีอยู่แล้ว. ทั้งคู่มุ่งความสนใจที่อัตราการสังเคราะห์คอเลสเตอรอล ซึ่งจะขึ้นกับเอนไซม์ในขั้นแรกของวิตามินซี ชื่อ เอชเอมจี โคเอ รีดักเตส (HMG-CoA reductase หรือ 3-hydroxy-3-methyl-glutaryl-coenzyme A reductase) ซึ่งจะเรียกว่า รีดักเตส. ถ้ารีดักเตสทำงานมาก คอเลสเตอรอล จะถูกสังเคราะห์ออกมาก.

การทำงานของรีดักเตส จะอยู่ที่ตับ เพราะฉะนั้น โกลเด้นและบราร์นไม่สามารถศึกษาการทำงานของรีดักเตสโดยตรงได้. ทั้งคู่ตัดสินใจ ศึกษาการทำงานของรีดักเตสจากเซลล์ที่ตัดและนำมาเพาะเลี้ยงไว้แทน. เซลล์ที่เพาะเลี้ยงในหลอดทดลอง ต้องการสารอาหารที่จะป้อนให้ในรูปซีรัม (serum ซึ่งเป็นน้ำเลือดที่ไม่มีเนื้อเลือด). โกลเด้นและบราร์น สังเกตว่าการทำงานของรีดักเตสสูงควบคุมจากอะไรอย่างในซีรัม คือ พ่อให้ซีรัม การทำงานของรีดักเตสลดลง แต่พ่ออาซีรัมออก การทำงานของรีดักเตสเพิ่มขึ้นเป็นสิบเท่า. โกลเด้นและบราร์นสงสัย และค้นหาว่าอะไรในซีรัมที่ควบคุมการทำงานของรีดักเตส จนพบว่า ไขมันโปรตีนเบา (low-density lipoprotein คำย่อ LDL) เป็นตัวบั้นยั้ง (inhibitor) การทำงานของรีดักเตส.

โกลเด้นและบราร์นมีสมมติฐานว่า ผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม ที่ร่างกายสร้างคอเลสเตอรอลมากเกินไป อาจเพราษีการกลایพันธุ์ของยีนของรีดักเตส ที่ทำให้มีการสร้างรีดักเตสที่ผิดปกติและไม่ตอบสนองต่อไขมันโปรตีนเบา. ทั้งคู่ทำการทดลอง และพบว่า เซลล์จากผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม มีการทำงานของรีดักเตสมากกว่าเซลล์ปกติ สีสันถึงหากสิบเท่า และไขมันโปรตีนเบาไม่มีผลต่อการทำงานของรีดักเตส. แต่การทดลองต่อมากของทั้งคู่ กลับไม่พบความผิดปกติในตัวเอนไซม์รีดักเตสของผู้ป่วย ซึ่งชี้ว่า สมมติฐานรีดักเตสผิดปกติไม่ถูกต้อง.

ไขมันโปรตีนเบาบัญชาระบบที่ทำงานของรีดักเตสในเซลล์ปกติ แต่ไม่ทำให้เซลล์ผู้ป่วย. รีดักเตสของเซลล์ผู้ป่วยไม่ได้ผิดปกติ. ดังนั้น น่าจะต้องมีอะไรระหว่างกลาง ที่เป็นปัจจัย. ไขมันโปรตีนเบา จะประกอบไปด้วยโปรตีน ที่เรียกว่า ลิโปโปรตีน และไขมัน ซึ่งรวมถึงคอเลสเตอรอล. โกลเด้นและบราร์น ทดลองป้อนเฉพาะคอเลสเตอรอล โดยไม่มีลิปอโปรตีน และพบว่า คอเลสเตอรอลบัญชาระบบที่ทำงานของรีดักเตสอย่างชัดเจน ทั้งในเซลล์ปกติและเซลล์ผู้ป่วย. นั่นคือ รีดักเตสของผู้ป่วยทำงานได้ปกติ ถูกควบคุมด้วยคอเลสเตอรอลได้เหมือนกับรีดักเตสปกติ แต่ถูกควบคุมไม่ได้ถ้าคอเลสเตอรอลอยู่ในรูปไขมันโปรตีนเบา.

ไขมันโปรตีนเบาจับตัวได้ดีกับเซลล์ปกติ แต่ไม่จับกับเซลล์ของผู้ป่วย. เซลล์ปกติมีรีเซปเตอร์สำหรับจับตัวกับไขมันโปรตีนเบา แต่เซลล์ของผู้ป่วยไม่มี. โกลเด้นและบราร์น ศึกษากลไกนี้ และพบว่า ลิปอโปรตีนของไขมันโปรตีนเบา นำคอเลสเตอรอลไปให้เซลล์ โดยตัวลิปอโปรตีนจะจับตัวกับรีเซปเตอร์ไขมันโปรตีนเบา (LDL receptors) และคอเลสเตอรอลจะถูกแยกออกจากโปรตีนตอนที่เข้าไปอยู่ในเซลล์ ซึ่งคอเลสเตอรอลจะสามารถเข้าควบคุมการทำงานของรีดักเตสได้.

นั่นคือ ในเซลล์ปกติ ไขมันโปรตีนเบา (ซึ่งมีคอเลสเตอรอลอยู่) จับกับรีเซปเตอร์ไขมันโปรตีนเบา และส่งผลบัญชาระบบที่ทำงานของรีดักเตส. แต่ในเซลล์ของผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม ไขมันโปรตีนเบา (ซึ่งมีคอเลสเตอรอลอยู่) ไม่สามารถส่งคอเลสเตอรอลเข้าไปในเซลล์ได้ การทำงานของรีดักเตสไม่ถูกบัญชาระบบที่ทำงานของรีดักเตสได้.

ช่วงเวลาใกล้เคียงกัน อา基ระ เอนโด (Akira Endo) ที่ขณะนั้นทำงานกับบริษัทยาชันเคียว ในโตเกียว ญี่ปุ่น พยายามค้นหาสารประกอบเพื่อยับยั้งการทำงานของรีดักเตส. เอนโด้มีประสบการณ์จากการก่อนหน้า ที่เขาค้นพบเอนไซม์จากรา เพื่อย่อยเนื้อผลไม้ที่ป่นมาในไวน์และเหล้าผลไม้. เอนโด้มีรูเรื่องของราบานชินิด ที่มีไม้เลกูลอโภสเทอรอล (ergosterol) เป็นส่วนประกอบสำคัญของเยื่อหุ้มเซลล์ แทนที่จะเป็นคอเลสเตอรอล เขาเลยคิดว่า ราบานชินิดน่าจะมีสารประกอบที่บัญชาระบบที่ทำงานของรีดักเตสได้.

เอนโด้มีงานค้นหาสารประกอบที่อย่างได้ โดยค้นหาจากราประมาณ 6000 ชนิด และทดสอบดูว่า น้ำจากการแต่ละชนิด

จะยับยั้งการทำงานของรีดักเตสได้หรือไม่ จากการค้นหาอยู่สองปี เอโน่ได้กับทีมงาน พบรารออกฤทธิ์สกัดจากรากของชนิดที่ยับยั้งการทำงานของรีดักเตสได้ ตัวหนึ่งได้จากรา ไฟเซียม อัลติมัม (*Pythium ultimum*) ซึ่งถูกทราบว่าเป็นยาปฏิชีวนะที่รู้จักกันอยู่แล้ว ชื่อ ซิตรินิน (*citrinin*). ซิตรินินยับยั้งการทำงานของรีดักเตสได้ แต่เป็นพิษมาก. อีกตัวหนึ่งได้จากรา เพนนิชิเลียม ซิตรินัม (*Penicillium citrinum*) ซึ่งมาจากการสกัดจากราที่ใช้สกัดยาเพนนิชิลิน.

สำหรับศึกษาและพัฒนาความเป็นยาต่อ เอโน่ได้กับทีมงานต้องเพาะเจี้ยงเพนนิชิเลียมซิตรินัมมากถึง 600 ลิตร เพื่อที่จะสกัดสารประกอบมาได้ปริมาณ 23 มิลลิกรัม และพบว่า โมเลกุล ML-236B ซึ่งภายหลังคือ คอมแพคติน (*Compactin* หรือชื่ออื่น เมวาสแตติน *Mevastatin*) เป็นสารออกฤทธิ์. เอโน่ได้กับทีมงานเผยแพร่การค้นพบนี้<sup>[63]</sup> และพัฒนาคอมแพคตินต่อเพื่อเป็นยา ซึ่งคือการทดลองในสัตว์. การทดลองในหนู แม้ว่าไม่เพ็บผลเป็นพิษ แต่คอมแพคตินไม่ช่วยลดคอเลสเตอรอลในกระแสเลือดของหนูเลย ไม่ว่าจะให้ยาอยู่เจ็ดวัน หรือใช้ขนาดยาสูงอยู่ถึงห้าสัปดาห์.

ผิดหวัง แต่เอโน่ได้ยังไม่ยอมแพ้. จากการทดลองที่ผ่าน ๆ มา เอโน่ได้สังสัยว่า ที่คอมแพคตินไม่เป็นผลกับหนู อาจเป็นเพราะร่างกายของหนูมีกลไกควบคุมคอเลสเตอรอลที่ต่างไป และเอโน่จึงเริ่มการทดลองใหม่ในไก่ ซึ่งได้ผลดีมาก และผลในลิงและผลในสุนัข ก็พบการลดลงของคอเลสเตอรอลอย่างเด่นชัด. โดยการของคอมแพคตินเริ่มสุดใส และชันเคียวให้การสนับสนุนอย่างเด่นที่. แต่ นักพิชวิทยาเห็นความผิดปกติในเซลล์ตับของหนูที่ให้คอมแพคตินที่ขนาดยาสูงมาก สุดท้ายหลังจากไตร่ตรองอยู่หลายเดือน ชันเคียวที่ตัดสินใจจะดำเนินการทดสอบทางคลินิก. แต่แล้ว ชันเคียว ก็สั่งหยุดการพัฒนาคอมแพคตินทันที หลังจาก นักพิชวิทยาของบริษัทสังสัยว่า สุนัขที่ห้ามคอมแพคตินที่ขนาดยาสูงติดต่อกันสองปี จะมีเนื้องอกในลำไส้.

ในช่วงนั้น บริษัทฯต่าง ๆ รู้เรื่องการพัฒนาคอมแพคตินของชันเคียว. รอย วาเจโลส (Roy Vagelos) หัวหน้าฝ่ายวิจัยของบริษัทเมอร์ค อย่างจะเปลี่ยนวิธีการค้นหายา จากเดิมที่การค้นหาสารประกอบทำด้วยการทดลองกับเซลล์หรือจุลทรรศ์ วาเจโลสอย่างจะเปลี่ยนเป็นการทดลองกับโมเลกุลเป้าหมาย.

จากการของโกลเดิลส์ไตน์และบราน์ และการค้นพบคอมแพคตินของเอโน่ วาเจโลสเห็นโอกาสที่จะได้ลองวิธีใหม่. วาเจโลสและทีมงานที่เมอร์ค ค้นหายาแบบคอมแพคตินจากรากชนิดอื่น ๆ และสุดท้าย พบรารประกอบจากรา อัสเพอร์จิลลัส เทโรเรียส (*Aspergillus terreus*) ซึ่งภายหลังคือ โลวาสแตติน (*Lovastatin*). แต่หลังจากที่เมอร์ครู้ข่าวชันเคียวยกเลิกการพัฒนาคอมแพคติน เมอร์คก็ตัดสินใจยกเลิกการพัฒนาโล瓦สแตตินด้วย.

โกลเดิลส์ไตน์และบราน์ เองก็รู้เรื่องงานของเอโน่ ทั้งคู่สนใจ ติดต่อกับเอโน่ แล้วได้ตัวอย่างคอมแพคตินมาทดลอง ซึ่งผลการทดลอง นอกจากแสดงในเห็นว่า เมื่อใช้คอมแพคติน การทำงานของรีดักเตสลดลงชัดเจนแล้ว. สิ่งที่โกลเดิลส์ไตน์และบราน์พบใหม่ก็คือ เซลล์สร้างรีดักเตสเพิ่มขึ้น.

ในขณะที่การสังเคราะห์คอเลสเตอรอล ถูกควบคุมด้วยการทำงานของรีดักเตส การสังเคราะห์รีดักเตสเองก็ถูกควบคุมยับยั้งด้วยปริมาณคอเลสเตอรอล. ผลการทดลองที่โกลเดิลส์ไตน์และบราน์พบ แสดงให้เห็นถึง อิทธิพลภาคเสริม (*double-negative effect*) ในกระบวนการควบคุมปริมาณคอเลสเตอรอล. นั่นคือ การยับยั้งการทำงานของรีดักเตส ส่งผลให้ไม่มีคอเลสเตอรอลผลิต เมื่อไม่มีคอเลสเตอรอล ก็ไม่มีอะไรยับยั้งการสังเคราะห์รีดักเตส ดังนั้นปริมาณรีดักเตสจึงเพิ่มขึ้น. หมายเหตุ แม้ปริมาณของรีดักเตสเพิ่มขึ้น แต่รีดักเตสไม่ได้ทำงาน.

โกลเดิลส์ไตน์และบราน์ ตีโจมากกับการค้นพบนี้ เพราะว่า งานวิจัยก่อนหน้านี้ทำให้ทั้งคู่รู้ว่า การสังเคราะห์รีดักเตสและรีเชบ/เตอร์ไขมันโปรตีนเบาถูกควบคุมไปพร้อม ๆ กัน ดังนั้น การเห็นการสังเคราะห์รีดักเตสเพิ่ม ก็อาจหมายถึงการสังเคราะห์รีเชบ/เตอร์ไขมันโปรตีนเบาเพิ่มด้วย. การเพิ่มรีเชบ/เตอร์ไขมันโปรตีนเบา ก็น่าจะทำให้เซลล์สามารถดึงไขมันโปรตีนเบาจากกระแสเลือดเข้าเซลล์ได้มากขึ้น และลดระดับคอเลสเตอรอลในกระแสเลือด ที่เป็นสาเหตุของการหัวใจวาย.

นั่นคือ โกลเดิลส์ไตน์และบราน์ วางแผนติดตามว่า สำหรับผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพัณฑุกรรม รีเชบ/เตอร์ไขมันโปรตีนเบามีจำนวนน้อย ทำให้คอเลสเตอรอลในกระแสเลือดมีปริมาณมาก. แต่เมื่อใช้คอมแพคตินแล้ว การทำงานของรีดักเตสลด การสังเคราะห์คอเลสเตอรอลในเซลล์ลด การสังเคราะห์รีดักเตสและรีเชบ/เตอร์ไขมันโปรตีนเบาเพิ่ม รีเชบ/เตอร์ไขมันโปรตีนเบามีจำนวนเพิ่มขึ้น สามารถรับไขมันโปรตีนเบาจากกระแสเลือดเข้ามาในเซลล์ได้ ช่วยให้ภายในเซลล์มีคอเลสเตอรอลใช้ และทำให้คอเลสเตอรอลในกระแสเลือดมีปริมาณลดลง.

ทั้งคู่ทดสอบสมมติฐาน โดยขอตัวอย่างลาวสแตนมาจากเมอร์ค และทดลองกับสุนัข. ผลคือ ทั้งรีเซปเตอร์ไขมันปรตีนบางมีจำนวนเพิ่มขึ้น และคงเหลือไว้ในกระแสเลือดมีปริมาณลดลง. ทั้งคู่มั่นใจกับผลการทำงาน แต่จะพยายามให้ผู้ป่วยจากใน เมื่อทั้งชั้นเคียและเมอร์คก์รับการพัฒนา เนื่องจากกลัวความเสี่ยงของการเกิดเนื้องอกในลำไส้. โกลเด้นไซด์และบราวน์ตัดสินใจไปปฏิญญาเพื่อปรึกษา กับเอนดोซี. ตอนนั้นเอนดोซีไม่ได้ทำงานให้ชั้นเคียแล้ว เอ็นดูซีย้ายไปทำงานที่มหาวิทยาลัยเกษตรและเทคโนโลยีโดยเกี้ยว. เอ็นดูซี เห็นว่า นักพิชิตยาจารจะตีความผลที่เห็นในสุนัขผิด และคิดว่า สิ่งที่นักพิชิตยาเห็นในลำไส้ อาจจะไม่ใช่นেืองอก อาจจะเป็นยาที่ไม่ย่อยมากกว่า เพราะว่า การทดลองใช้ขนาดยาที่สูงมาก ซึ่งมากกว่าที่จะใช้ในคนถึงร้อยเท่า.

ค่อนข้างมั่นใจกับยา และด้วยโลภสแตตินที่ได้มา โกลเดิลสไตน์และบราร์น์ร่วมกับเพื่อนอีกสองคน ทดสอบยา กับผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรมจำนวนหกคน และผลที่ได้คือรีเชปเตอร์ไขมันโปรดีนเบมีจำนวนเพิ่มขึ้น และколоเลสเตอรอลในกระเพาะเลือดลดลงประมาณ 27% ผลที่ได้นี้ช่วยให้เมอร์คตัดสินใจกลับมาพัฒนาโลภสแตตินต่อ แต่ผู้บริหารของเมอร์ค ก็ยังกังวลกับความเสี่ยงจากเนื้องอกอยู่ เพื่อทำประดีนเรื่องเนื้องอกให้ชัดเจน และโอกาสในการใช้ยา กับผู้ป่วยภาวะไขมันในเลือดสูงทั่วไป เอดเวิร์ด สโคลนิก (Edward Skolnick) หัวหน้าฝ่ายวิจัยพื้นฐานของเมอร์ค ตั้งทีมงานเฉพาะชั้นนำ เพื่อศึกษาผลทางพิชิตยาให้สมบูรณ์ สโคลนิกปรึกษากับโกลเดิลสไตน์และบราร์น์ และโกลเดิลสไตน์และบราร์น์ได้แนะนำวิธีการทดสอบ เพื่อระบุว่า สิ่งที่เห็นในสัตว์ทดลองว่าเป็นผลจากยาจริง ๆ หรือว่าแค่จากการทดสอบด้วยขนาดยาสูงมาก ซึ่งสามารถป้องกันได้ง่าย ๆ ทีมงานนักวิจัยของสโคลนิกทดลอง และไม่พบผลร้ายจากยา สโคลนิกลองก และเมอร์คที่ได้รับความประทับใจของยา.

ผลจากการทดสอบอยู่ส่องปี ยืนยันว่า โล瓦สแตตินช่วยลดคอเลสเตอรอลในกระแสเลือดได้กว่า 20% เมอร์คยืนใจที่เป็นยา และได้เริ่มขายโล瓦สแตตินในปี ค.ศ. 1987. ทั้งโลวาสแตติน และคอมแพคติน รวมไปถึงยาที่พัฒนาขึ้นมาภายหลังตัวอื่นๆ ในกลุ่มนี้ จะเรียกว่า กลุ่มยาสแตติน (Statins). เพื่อการติดตามผลการใช้ยา เมอร์คสนับสนุนการศึกษาห้าปี กับผู้ป่วยระดับคอเลสเตอรอลในกระแสเลือดสูง จำนวน 4,444 คน ที่ใช้ยาซิมัวสแตติน (ซึ่งเป็นยาในกลุ่มสแตติน ที่พัฒนาขึ้นมาภายหลัง) และพบว่า ยาช่วยลดอัตราการตายจากหัวใจวายของผู้ป่วยลง 42%. ปัจจุบัน มีผู้ใช้ยาในกลุ่มสแตตินมากกว่า 6 ล้านคนทั่วโลก และอัตราการตายจากหัวใจวายของชาวอเมริกันลดลงเกือบทกสิบเปอร์เซ็นต์ (นับจากที่ อันเชล คีส พอบันตรายจากคอเลสเตอรอล).

**อุตสาหกรรมยา การค้นหาและพัฒนายา.** โรนัล คริสโตเฟอร์ (Ronald Christopher) จากบริษัทยาอาเรينا บรรยายเรื่องการเลือกสารประกอบและการศึกษาภัยคุกคาม[66] ว่า การค้นหายาเป็นกิจกรรมที่อัตราการล้มเหลวสูงมาก ประมาณหนึ่งในพันนั้นคือ จากขั้นตอนแรก ๆ อาจมีสารประกอบที่สนใจอยู่ประมาณห้าพันถึงหนึ่งหมื่นตัว สุดท้ายจะเหลือแค่ประมาณสิบตัวที่ผ่านกระบวนการประเมินถึงการทดสอบทางคลินิกกับมนุษย์ได้. ระยะเวลาในการค้นหายา โดยเฉลี่ย จะประมาณสิบสองปี และค่าใช้จ่ายในการพัฒนายาแต่ละตัว ประมาณ 1.3 พันล้านดอลลาร์หรือประมาณสี่แสนล้านบาท ต่อการค้นหาและพัฒนายาที่จะได้รับการอนุมัติ เป็นไปในอัตราต่อเนื่องที่ 2.87 พันล้านดอลลาร์ แต่ประสิทธิภาพและรายได้ต่อหน่วยต่ำกว่าค่าใช้จ่ายอยู่ที่ 648 ล้านดอลลาร์ซึ่งต่ำกว่ามาก. อาย่างไรก็ตาม แมทธิว เออร์เปอร์ ได้เขียนบทความ “The Cost Of Developing Drugs Is Insane. That Paper That Says Otherwise Is Insanely Bad” Oct 16, 2017, 10:58am EST ในเวปไซต์ <http://www.forbes.com> ซึ่งวิจารณ์วิธีประเมินของประสิทธิภาพและรายได้โดยเฉพาะเรื่องที่ประสิทธิภาพและรายได้ไม่ได้รวมค่าใช้จ่ายของความล้มเหลวในกระบวนการค้นหายาเข้าไปด้วย. เออร์เปอร์วิจารณ์ว่า ผลสรุปของประสิทธิภาพและรายได้เป็นลักษณะของความลำเอียงไปทางคนที่รอด survivorship bias. ความลำเอียงไปทางคนที่รอด หมายถึง การวิเคราะห์ที่ใช้ผลสรุปแทนภาพรวมทั้งหมด แต่ใช้ข้อมูลเฉพาะจากกลุ่มที่หลุดรอดเท่านั้น ไม่ว่าจะอย่างไร กิจกรรมการค้นหาพัฒนายาเป็นกิจกรรมที่ลงทุนมหาศาล อาศัยเครื่องมือขั้นสูงและทักษะกับความทุ่มเทอย่างยิ่งคาดของบุคคลากรที่เกี่ยวข้อง.)

رونัล คริสโตเฟอร์ อธิบายการเลือกสารประกอบมาเป็นยาว่า มีเกณฑ์ในการพิจารณาอย่าง ๆ หลาย เช่น คุณสมบัติทางเคมีชีวภาพ ได้แก่ การออกฤทธิ์ที่ดี สมรรถนะ การเลือกที่สูง (สารประกอบจับตัวกับเป้าหมายดีกว่าจับตัวกับชีวะโมเลกุลอื่นในร่างกาย มากกว่าพันเท่า) ประสิทธิผลที่ดีในการทดลองกับสัตว์ (แสดงให้เห็นว่ามันได้ผล). นอกจากนั้น ก็ยังพิจารณาเรื่อง เมแทบอโลซีมของยา และคุณสมบัติทางเคมีชีวศาสตร์ (ร้ายกาจตอบสนองต่อยาอย่างไร) รวมถึง การปฏิสัมพันธ์ระหว่างยา (drug-drug interactions) ว่ายาตัวใหม่นี้จะไม่ไป愧ળยาอื่นที่ผู้ป่วยใช้อยู่ และปัจจัยด้านความปลอดภัย เช่น ผลการศึกษาด้านความปลอดภัยในทางที่ดีทั้งการศึกษาในหลอดทดลอง และในสัตว์ทดลอง.

คริสโตเฟอร์ยกตัวอย่างประสบการณ์การพัฒนายาโลกลิบติน สำหรับบำบัดโรคเบาหวาน. โรคเบาหวาน เป็นภาวะที่ร่างกาย มีน้ำตาลในเลือดสูง. ระดับน้ำตาลในเลือด (blood glucose) ถูกควบคุมด้วยอินซูลิน (insulin). การปล่อยอินซูลินถูกควบคุมด้วย ออร์โมนอินคริตินส์ (incretins[109]) การควบคุมอินคริตินส์ถูกควบคุมด้วยดีพีพีสี (Dipeptidyl peptidase-4 คำย่อ DPP-4). การควบคุมในร่ายกายมีอยู่สองแบบหลัก ๆ ได้แก่ การควบคุมเชิงบวก คือการสนับสนุนหรือกระตุ้น และควบคุมเชิงลบ คือการลดหรือ ยับยั้ง. อินซูลินควบคุมระดับน้ำตาลในเลือดในเชิงลบ อินคริตินส์ควบคุมอินซูลินในเชิงบวก ดีพีพีส์ควบคุมอินคริตินส์ในเชิงลบ. นั่น คือ หากอินซูลินเพิ่มขึ้น ระดับน้ำตาลในเลือดจะลดลง. หากอินคริตินส์เพิ่มขึ้น อินซูลินจะเพิ่มขึ้น. แต่หากดีพีพีส์ทำงาน อินคริตินส์ จะลดลง ส่งผลให้อินซูลินลดลง ส่งผลให้ระดับน้ำตาลในเลือดเพิ่มขึ้น. ยานบางตัว เช่น เอ็กซีนาไทด์ (Exenatide) เลือกเป้าหมายเป็น รีเซปเตอร์ของอินคริตินส์. ตัวยาจะไปจับกับรีเซปเตอร์ของอินคริตินส์ เพื่อส่งผลเหมือนการเพิ่มของอินคริตินส์. เอ็กซีนาไทด์ เป็นยา ฉีดและมีผลข้างเคียงค่อนข้างมาก. ทีมงานของคริสโตเฟอร์ เลือกเป้าหมายเป็นดีพีพีส์ และต้องการหาโมเลกุลที่ยับยั้งดีพีพีส์ (DPP4 inhibitor). การยับยั้งดีพีพีส์ เท่ากับเพิ่มการทำงานของอินคริตินส์ อินคริตินส์ทำงานมากขึ้นจะไปเพิ่มอินซูลิน อินซูลินเพิ่มขึ้นจะไป ลดระดับน้ำตาลในเลือด.

ตอนนี้ มียาที่ยับยั้งดีพีพีส์ในตลาดอยู่หลายตัวแล้ว เช่น วิลดาเกลิปทิน. ทีมงานของคริสโตเฟอร์ ศึกษาโครงสร้างทางเคมี ของดีพีพีส์ ซึ่งเป็นงานที่มีขั้นตอนที่ซับซ้อนมาก ตั้งแต่การโคลนดีพีพีส์ และทำกระบวนการต่าง ๆ ที่จะทำให้ดีพีพีส์ติดผิว แลบถ่าย ภาพดีพีพีส์ที่ติดผิวด้วยอีกซ์เรย์ ซึ่งต้องใช้เครื่องชินโคตรอน. ภาพถ่ายที่ได้จะเป็นภาพของรูปแบบการกระเจิงของอีกซ์เรย์ ซึ่ง ต้องใช้นักชีววิทยาโครงสร้างอ่าน ตีความ และแปลงออกมายเป็นแบบจำลองคอมพิวเตอร์ของโครงสร้างเคมีสามมิติ ที่นักเคมีสามารถ ใช้วิเคราะห์ได้ต่อไป. นักเคมีในทีมงานของคริสโตเฟอร์ ดูโครงสร้างดีพีพีส์ และการเข้าอุจจับตัวกับวิลดาเกลิปทิน แล้วพบว่า ใน การจับตัวกันของวิลดาเกลิปทินและดีพีพีส์ มีการจับด้วยพันธะโควาเลนท์อยู่. พันธะโควาเลนท์ทำให้โมเลกุลยาจับตัวกับดีพีพีส์แน่น. การจับดีพีพีส์แน่นเกินไป อาจก่อให้เกิดผลข้างเคียงต่อระบบภูมิคุ้มกัน. ทีมงานของคริสโตเฟอร์ ต้องการจะพัฒนาใหม่ที่จับดีพีพี สี โดยไม่มีพันธะโควาเลนท์.

การค้นหาออกแบบและพัฒนาของทีมของคริสโตเฟอร์ ทำโดยอาศัยโครงสร้างโมเลกุล. ในการค้นยาออกแบบเบบยา โดยที่นำไป จะมีสองแนวทางหลัก ๆ คือ อาศัยลิกแคนต์ (ligand-based) หรืออาศัยโครงสร้าง (structure-based). วิธีอาศัยลิกแคนต์ ไม่จำเป็น ต้องรู้โครงสร้างทางเคมีสามมิติของเป้าหมาย แต่ต้องรู้จักบางลิกแคนต์ของเป้าหมาย แล้วสร้างแบบจำลองทำงานการจับตัว และ ค้นหาโมเลกุลที่อาจเป็นยาได้จากแบบจำลองทำงาน (ที่สร้างโดยอาศัยข้อมูลลิกแคนต์เหล่านั้น). วิธีอาศัยโครงสร้าง ต้องรู้โครงสร้าง ของโมเลกุลเป้าหมาย. ยา มักเป็นโมเลกุลขนาดเล็ก แต่เป้าหมาย เช่น โปรตีน เป็นโมเลกุลขนาดใหญ่. การหาโครงสร้างสามมิติของ โมเลกุลขนาดใหญ่ เป็นเรื่องซับซ้อนและใช้ทักษะสูง แต่หากได้โครงสร้างสามมิติของโมเลกุลเป้าหมายมาแล้ว นักเคมีจะดูโครงสร้าง ของตำแหน่งจับตัวในโปรตีนเป้าหมาย แล้วจึงพิจารณาหาลิกแคนต์ โดยอาจเริ่มจากส่วนเล็ก ๆ ของโครงสร้างทางเคมีที่ต้องการ และ ค่อยค้นหาโมเลกุลของลิกแคนต์ตามนั้น หรืออาจจะค้นหาจากสารประกอบที่มีอยู่ในฐานข้อมูล ด้วยวิธีการกลั่นกรองเสมอ หรืออาจ จะออกแบบโครงสร้างของลิกแคนต์ขึ้นมาใหม่เลยก็ได้.

วิธีการกลั่นกรองเสมอ (virtual screening[98] คำย่อ vs) เป็นการใช้คอมพิวเตอร์เข้ามาช่วยค้นหาโมเลกุลต่าง ๆ จากฐาน ข้อมูล เพื่อหาโมเลกุลที่มีโอกาสสูงในการนำมาพัฒนาต่อเป็นยา. โมเลกุลต่าง ๆ ที่อาจเป็นยาได้ มีจำนวนมหาศาล. การทดสอบ แต่ละโมเลกุลกับเป้าหมายในหลอดทดลองมีค่าใช้จ่ายสูง. การใช้คอมพิวเตอร์ช่วยกลั่นกรองเลือกโมเลกุลต่าง ๆ ก่อน แล้วค่อยเลือก ทดสอบโมเลกุลที่ผ่านการกลั่นกรองขึ้นกับเป้าหมายในหลอดทดลอง จะช่วยลดค่าใช้จ่าย เวลา และทรัพยากรบุคคลในการพัฒนา ยาลงได้มาก. ในทางปฏิบัติ วิธีการกลั่นกรองเสมอ ก็ไม่ได้ค้นหากับทุกโมเลกุลที่เป็นไปได้ แต่อาจจะเลือกจากฐานข้อมูลของยาที่ ได้มีการทดสอบแล้ว อาจเลือกจากรายการของสารประกอบที่มีอยู่ในคลังของบริษัทแล้ว อาจเลือกจากฐานข้อมูลจากผู้ขาย เป็นต้น โดยอาจลำดับความสำคัญของการค้นหา จากยาที่มีการทดสอบแล้ว ต่อด้วยสารต่าง ๆ ที่มีอยู่คลัง แล้วไปสารต่าง ๆ ที่สามารถจัด ชื่อได้ จนสุดท้ายถึงค้นหาสารต่าง ๆ ที่จะต้องสังเคราะห์ขึ้นใหม่. หากพบโมเลกุลจากฐานข้อมูลยาที่มีการทดสอบแล้ว จะช่วยลดค่า ใช้จ่ายในการศึกษาหลาย ๆ อย่างที่มีผลการศึกษาอยู่แล้ว. สารที่มีอยู่แล้วในคลังก็จัดหาได้ง่ายกว่า และสารที่สามารถหาชื่อได้ ก็ สะดวกและมักเสียค่าใช้จ่ายน้อยกว่าการสั่งเคราะห์สารขึ้นมาเองใหม่.

วิธีการกลั่นกรองเสมอ อาจใช้การทำนายการเข้าอุจจับ ( docking ) ซึ่ง จะทำนายรูปร่างและทิศทางการวางตัวของโมเลกุล เมื่อ โมเลกุลจับตัวกับเป้าหมาย ซึ่งผลการทำนายนี้อาจใช้ประกอบ เพื่อทำนายอัตราการจับตัวกัน (binding affinity). อัตราการจับตัว

กัน เป็นโอกาสของการจับตัวกันระหว่างลิแกนต์กับเป้าหมาย. แบบจำลองที่ทำนายอัตราการจับตัวกัน จะเรียกว่า พังก์ชันคะแนน (scoring function). พังก์ชันคะแนน อาจทำนายโอกาสของการจับตัว ด้วยพลังงานรวมของการจับตัวกัน โดยคำนวณจากทฤษฎี สมมติฐาน ซึ่งอาศัยรูปร่างและทิศทางการวางตัวของโมเลกุลที่จับตัวกัน. ค่าพลังงานที่ต่ำกว่า หมายถึงผลการจับตัวที่มีเสถียรภาพ มากกว่า และโอกาสที่มากกว่าของการจับตัวกัน. การใช้แบบจำลองการเรียนรู้ของเครื่องเป็นอีกแนวทางหนึ่งที่สามารถนำมาใช้สร้าง พังก์ชันคะแนนได้. หมายเหตุ ตัวอย่างในแบบฝึกหัด 3.17 เป็นแค่การทำนายการจับตัวกัน ไม่ได้มีการประเมินโอกาสจับตัวกันของ มาเป็นตัวเลข ซึ่งฐานข้อมูลดีดีไม่มีข้อมูลน้อย. แต่หากมีข้อมูลอัตราการจับตัวกัน ซึ่งมักวัดเป็นเปอร์เซ็นต์การจับตัวต่อความเข้ม ข้นของสารละลายของลิแกนต์ (%binding per molar concentration) ก็สามารถนำมาสร้างเป็นแบบจำลองได้ โดยการทำนาย ลักษณะนี้เป็นการทำนายค่าต่อเนื่อง และหมายความว่าการครอบเป็นแบบจำลองการหาค่าต่อตอย.

การค้นหาและพัฒนาฯ มักดำเนินการในลักษณะการวนทวนกลั่นกรอง นั่นคือ เป็นลักษณะวนค้นหา ปรับปรุง และสลับกันไป จนกว่าจะได้ลิแกนต์ที่มีลักษณะความเป็นยาสูงอ่อนมา.

หลังจากได้ลิแกนต์ที่ผ่านรอบแรกมา ทีมพัฒนาจะปรับปรุงโมเลกุล โดยอาจจะค้นหาโมเลกุลอื่น ๆ ที่ใกล้เคียงกับลิแกนต์ เหล่านั้น หรืออาจจะปรับโครงสร้างบางส่วนของลิแกนต์เหล่านั้น เพื่อให้มีคุณสมบัติทางยาต่าง ๆ ดีขึ้น. คุณสมบัติทางยาต่าง ๆ ที่ปรับปรุง ได้แก่ อัตราการจับตัวกับเป้าหมาย ( $IC_{50} \leq 100$  นาโนโมลาร์ ซึ่งค่า  $IC_{50}$  วัดจากความเข้มข้นของลิแกนต์ที่สามารถจับ กับเป้าหมายได้ครึ่งหนึ่ง), สมรรถนะการเลือก (ลิแกนต์มีการจับตัวกับเป้าหมายได้ต่ำกว่าจับตัวกับโมเลกุลอื่นที่คล้ายเป้าหมายหนึ่ง พันเท่า), รวมถึงการออกฤทธิ์ของยา คุณสมบัติเมแทบอลิซึมของยา คุณสมบัติทางเภสัชจานศึกษาและเภสัชพลศึกษา เป็นต้น. ใน ขั้นตอนการพัฒนาฯ อาจมีการใช้แบบจำลองทำนาย เช่น ความสัมพันธ์เชิงปริมาณระหว่างโครงสร้างและกิจกรรม (Quantitative Structure-Activity Relationship[25, 130] คำย่อ QSAR) เข้ามาช่วย. ความสัมพันธ์เชิงปริมาณระหว่างโครงสร้างและกิจกรรม เป็นการทำนายกิจกรรมหรือคุณสมบัติของโมเลกุล จากโครงสร้างของโมเลกุล ซึ่งกิจกรรมที่ทำนาย อาจเป็น ผลกระทบเป้าหมายหลัง การจับตัว (ว่าเป็น ผลทำการ agonism ที่ทำให้เป้าหมายทำงานมากขึ้น หรือผลต่อต้าน antagonism ที่ลดการทำงานของเป้าหมาย ลง), ชีวปริมาณออกฤทธิ์ (bioavailability), การละลายน้ำ (solubility), สมรรถนะการเลือก, การออกฤทธิ์ เป็นต้น. ความสัมพันธ์ เชิงปริมาณระหว่างโครงสร้างและกิจกรรม อาจสร้างจากพื้นฐานทางฟิสิกส์และเคมี หรืออาจสร้างตามแนวทางการเรียนรู้ของ เครื่องโดยอาศัยข้อมูลกีด้วย คล้ายกับแบบฝึกหัด 3.17 โครงสร้างทางเคมีจะถูกแปลงเป็นลักษณะสำคัญเชิงเลข ที่อาจเรียกว่า ตัว บอค (descriptor) เพื่อให้สามารถใช้คำนวณในแบบจำลองได้.

หลังจากทีมงานของคริสโตเฟอร์ทำงานอย่างหนัก กระบวนการค้นหาออกแบบและพัฒนาโมเลกุลเสร็จสิ้น ทีมงานได้ออกลิบ ติน (Aloglibitin). อโลกลิบตินเข้าอยู่จับตัวกับดีพีพีสีได้ และไม่เมพันธะที่เป็นพันธะโควาเลนต์. อโลกลิบตินจับตัวกับดีพีพีสีสักพักแล้ว ก็หลุด และกลับไปจับตัวใหม่ สลับกันไป เปิดโอกาสให้ดีพีพีสีเป็นอิสระเป็นพัก ๆ ลดความเสี่ยงของผลข้างเคียงต่อระบบภูมิคุ้มกัน.

ทีมงานทดสอบอโลกลิบตินในทดลอง และพบว่าอโลกลิบตินมีการออกฤทธิ์ที่ดี ( $IC_{50} = 6.9$  ซึ่ง  $IC_{50}$  สำหรับการออกฤทธิ์ของตัวยับยั้ง วัดจากความเข้มข้นของสารละลายยา ที่เพียงพอที่จะยับยั้งการทำงานของเป้าหมายได้ 50%. ตั้งนั้นตัวเลขน้อย กว่า หมายถึงการออกฤทธิ์ที่ต่ำกว่า). ยาตัวอื่นในกลุ่มเดียวกันที่อยู่ในตลาด มีการออกฤทธิ์วัดด้วย  $IC_{50}$  เป็น 23.8 และ 12.1 ซึ่ง หักหมดจัดว่ามีการออกฤทธิ์ที่ดี.

ผลการทดสอบสมรรถนะ การเลือกของอโลกลิบตินก็ออกมาดี. อโลกลิบติน มีอัตราการจับตัวกับดีพีพีสี ต่ำกว่าการจับตัว กับโปรตีนที่ใกล้เคียงมากกว่าหนึ่งแสนเท่า สำหรับโปรตีนที่ใกล้เคียงแต่ละตัว ซึ่งได้แก่ DPP-2, DPP-8, DPP-9, FAP, PREP, และ Tryptase.

การทดสอบกับสัตว์ทดลอง แสดงผลที่ดีเช่นกัน. ผลในลิ๊ง แสดง (1) ความเข้มข้นของยาในกระแสเลือดหลังจากรับยาเข้าไป โดยความเข้มข้นเพิ่มจนไปถึงจุดสูงสุด ใช้เวลาประมาณหนึ่งชั่วโมง หลังจากนั้นค่อย ๆ ลดลง และความเข้มข้นของยาในกระแสเลือด เป็นไปตามขนาดยาที่ได้ ซึ่งนี้เป็นการถูกการตอบสนองของร่างกายต่อยา ซึ่งผลเป็นไปตามที่ทีมงานคาด และ (2) เปอร์เซ็นต์การยับยั้ง การทำงานของดีพีพีสี หลังรับยาเข้าไป ซึ่งเปอร์เซ็นต์ยับยั้งสูงสุดอยู่ที่ประมาณ 90% หลังจากรับยาไปราوا ๆ สองถึงสามชั่วโมง แต่ โดยรวมเปอร์เซ็นต์ยับยั้งค่อนข้างคงที่ต่อเวลาในรอบยี่สิบสี่ชั่วโมง และยังค่อนข้างคงที่ต่อขนาดยาที่ทดสอบด้วย (2mg/kg, 10mg/kg, และ 30 mg/kg) ซึ่งทีมงานก็พูดใจ.

ผลทดสอบกับหนูทดลอง ที่มีกลุ่มควบคุม (ไม่เป็นโรค) และกลุ่มที่หนูเป็นโรคเบาหวาน (Neonatally streptozotocin-

induced diabetic rats คำย่อ N-STZ-1.5 rats) ที่ได้ผลที่ดี. ผลกระทบการทำงานของดีพีพีสี แสดงการลดลงของเปอร์เซ็นต์การทำงานของดีพีพีสีตามขนาดยาที่ใช้. ผลกระทบปริมาณของอินซิตรินส์ แสดงการเพิ่มขึ้นของระดับอินซิตรินส์ตามขนาดยาที่ใช้. ผลกระทบอินซูลินในกระแสเลือดต่อเวลา แสดงการเพิ่มขึ้นของอินซูลินต่อเวลา ตามขนาดยาที่ใช้. ผลกระทบกลูโคสในกระแสเลือดต่อเวลา แสดงการลดลงของกลูโคสต่อเวลา ตามขนาดยาที่ใช้.

นอกจากการทดสอบการลดระดับน้ำตาลในเลือดแล้ว สำหรับโรคเบาหวาน ยังต้องทดสอบด้วยว่าจะไม่ทำให้เกิดอาการน้ำตาลต่ำ (hypoglycemia). นั่นคือ ต้องทดสอบว่า ยาจะไม่ปลดน้ำตาลในเลือดมากเกินไป. จากการทดลองกับหนูที่อดอาหารและกินอลอกลิบตินเข้าไปที่ขนาดยาสูง (30mg/kg และ 100mg/kg) เปรียบเทียบกับกลุ่มควบคุม และเปรียบเทียบกับกลุ่มที่กินน้ำที่ไกไลน์ด์ ยานี้ที่มีงานรู้ว่าให้ผลด้านนี้เมื่อตี ที่มีงานไม่พบว่า อลอกลิบตินทำให้ระดับน้ำตาลดต่ำเกินไปหรือระดับอินซูลินสูงเกินไป.

การทดลองต่าง ๆ ข้างต้นเป็นการให้ยาครั้งเดียว การศึกษาสภาพทนทาน (robustness) ของยา จะทดสอบโดยให้ยาติดต่อกันเป็นเวลานาน ซึ่งผลที่ได้ แสดงการลดลงของการทำงานของดีพีพีสีตามขนาดยา การเพิ่มขึ้นของอินซิตรินส์ตามขนาดยา เมื่อใช้ยาติดต่อกัน สำหรับหนูทดลองหลาย ๆ ชนิด. ผลสภาพทนทานเป็นไปได้ด้วยดี.

การศึกษาเมแทบอลิซึมของยา พบร่วยว่าถูกย่อยสลายด้วยเอนไซม์ CYP-2D6 และ CYP-3A4 และยามีผลกระแทบต่อการยับยั้งหรือการตุ้นเอนไซม์ทั้งสองน้อยมาก. นอกจากนั้น ผลศึกษาแสดงการจับของยา กับโปรตีนในน้ำเลือดต่ำมาก และไม่พบปฏิกิริยาน้ำพันธุ์ กับยาตัวอื่น เมื่อทดสอบกับยาเบาหวานตัวอื่น ๆ. ที่มีงานโล่งใจกับผลทดสอบนี้.

การทดสอบเพื่อวัดข้อมูลเกี่ยวกับความสามารถในการตัดสินใจ (F เป็น 87% ในลิงที่ให้อลอกลิบตินขนาด 10mg/kg หนึ่งครั้ง ซึ่ง F เป็นสัดส่วนของยาที่ดูดซึมเข้าไปในกระแสเลือดต่อยาที่กินเข้าไป). ค่าครึ่งอายุที่ 5.7 ชั่วโมง (ในลิง และค่าครึ่งอายุ เป็นดัชนีที่บอกรเวลาที่ยาลดความเข้มข้นเหลือครึ่งหนึ่งจากความเข้มข้นสูงสุด) ซึ่งคริสโตเฟอร์รูสิกว่าันอยู่ในปนิตและอย่างได้ที่เบ็ดข่าวโมง เพื่อที่ผู้ป่วยจะสามารถกินยาครั้งเดียวต่อวันได้ แต่ก็หวังว่าค่าครึ่งอายุในมนุษย์จะมากขึ้นกว่านี้. เส้นทางการขับถ่ายยาออกจากร่างกายคือผ่านปัสสาวะและอุจจาระ.

การทดลองด้านความปลอดภัยของยา ซึ่งที่มีงานของคริสโตเฟอร์ต้องทำการศึกษามากกว่าสิบการทดสอบ และใช้งบประมาณไปหลักสิบล้านдолลาร์ ผลที่ได้สรุปว่า ไม่พบความเป็นพิษต่อระบบประสาทกลาง ไม่พบความเป็นพิษต่อหัวใจและหลอดเลือด ไม่พบความเป็นพิษกับระบบปอดและทางเดินหายใจ ไม่พบความพิษทางพันธุกรรม ไม่พบความเป็นพิษเมื่อใช้ต่อเนื่อง (กับยาขนาด 200mg/kg ในสุนัข โดยให้ติดต่อกันทุกวันเป็นเวลา 9 เดือน) ซึ่งจากผลที่ได้ที่มีงานสามารถนำไปคำนวณความปลอดภัยในมนุษย์ ซึ่งผลที่ได้ถือว่าปลอดภัยมาก.

หลังจากได้ผลที่น่าพอใจกับสัตว์ทดลองแล้ว ยาจึงจะมีโอกาสได้ทดสอบทางคลินิกกับมนุษย์ได้ต่อไป ก่อนที่จะได้รับการพิจารณาเพื่อการรับรองขึ้นทะเบียนยา.

“Whatever you do may seem insignificant to you,  
but it is most important that you do it.”

--Mohandas Karamchand Gandhi

“อะไรก็ตามที่เราทำ มันอาจดูเล็กน้อยไม่สำคัญ  
แต่umnสำคัญที่เราทำมัน.”

—มหาตมา คานธี

### แบบฝึกหัด 3.18

จากเรื่องการหยุดก่อนกำหนด ในหัวข้อ 3.4 จงเขียนโปรแกรม เพื่อสร้างข้อมูล และฝึกโครงข่ายประสาทเทียมสองชั้น ขนาด 100 หน่วยอย่าง และทดลอง และเปรียบเทียบผลการทดลอง กับผลที่แสดงในรูป 3.21. สำหรับข้อมูล ให้สร้างจากความสัมพันธ์  $y = x + 0.3 \sin(2\pi x) + 0.3\varepsilon$  เมื่อ  $\varepsilon \sim \mathcal{U}(0, 1)$ . สร้าง 40 จุดข้อมูลสำหรับการฝึก และ 20 จุดข้อมูลสำหรับการทดสอบ. หลังจากนั้น ให้เพิ่มเงื่อนไขการหยุดก่อนกำหนด

พร้อมแบ่งข้อมูลส่วนหนึ่งออกมาราบจากข้อมูลฝึก มาเป็นข้อมูลตรวจสอบ ทดลองฝึกแบบจำลอง ที่มีการหยุดก่อนกำหนด เปรียบเทียบผล สรุป และอภิปราย.

หมายเหตุ ตัวอย่างในรูป 3.21 ใช้การกำหนดค่าน้ำหนักเริ่มต้น ที่ดัดแปลงมาจากวิธีเชิงนิยมโดยร์ เพื่อให้เห็นภาพชัดเจนขึ้น และลดเวลาในการฝึก. การใช้การกำหนดค่าน้ำหนักเริ่มต้นจากการสุมค่า อาจต้องทำการฝึกนานกว่าจำนวนสมัยฝึกที่เห็นในรูป 3.21.

### แบบฝึกหัด 3.19

จากเรื่องเส้นโค้งเรียนรู้ ในหัวข้อ 3.5 จะเขียนโปรแกรม เพื่อทดลองสร้างเส้นโค้งเรียนรู้. จะสร้างข้อมูลขึ้นมา 40 จุดสำหรับฝึก และ 25 จุดสำหรับทดสอบ จากความสัมพันธ์  $y = x + 8 \sin(x) + \varepsilon$  เมื่อ  $x \in [0, 4\pi]$  และ  $\varepsilon \sim \mathcal{N}(0, 1)$ . แล้ว สร้างเส้นโค้งเรียนรู้สำหรับ (ก) โครงข่ายประสาทเทียมสองชั้น ที่มี 2 หน่วยช่อน, (ข) โครงข่ายประสาทเทียมสองชั้น ที่มี 8 หน่วยช่อน และ (ค) โครงข่ายประสาทเทียมสองชั้น ที่มี 64 หน่วยช่อน โดยดำเนินการฝึกและทดสอบแบบจำลอง 5 ครั้ง ครั้งแรกใช้ข้อมูลฝึก 8 จุด ต่อมาใช้ 16, 24, 32, และ 40 จุดตามลำดับ. วัดค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสองของทั้งข้อมูลทดสอบ และข้อมูลฝึก. นำเสนอ (ดูตัวอย่างรูป 3.23 และ 3.24) อภิปรายผลที่ได้ และสรุป.

หมายเหตุ ให้ฝึกโครงข่ายประสาทเทียมให้สมบูรณ์ดีทุกครั้ง อาจใช้วิธีเชิงนิยมโดยร์ ช่วยในการกำหนดค่าน้ำหนักเริ่มต้น เพื่อช่วยลดเวลาในการฝึกลงได้.

### แบบฝึกหัด 3.20

จากวิธีคำนวนค่าเกรเดียนต์เชิงเลข ด้วย

$$\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial E}{\partial \theta_1} \\ \frac{\partial E}{\partial \theta_2} \\ \vdots \\ \frac{\partial E}{\partial \theta_M} \end{bmatrix} \text{ และ } \frac{\partial E}{\partial \theta_i} \approx \frac{E\left(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i + \varepsilon \\ \theta_{i+1} \\ \vdots \end{bmatrix}\right) - E\left(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i \\ \theta_{i+1} \\ \vdots \end{bmatrix}\right)}{\varepsilon} \quad (3.51)$$

เมื่อ  $E(\boldsymbol{\theta})$  คือค่าฟังก์ชันจุดประสงค์ และ  $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_M\}$  คือพารามิเตอร์ของแบบจำลอง ที่มีจำนวนพารามิเตอร์ทั้งหมด  $M$  ตัว.

จะตรวจสอบค่าเกรเดียนต์ที่คำนวณจากวิธีแพร์กราจายย้อนกลับ เปรียบเทียบกับค่าเกรเดียนต์ที่คำนวณจากวิธีการเชิงเลข โดยให้ใช้ค่า  $\epsilon$  เป็น 1, 0.01 และ 0.0001 ตามลำดับ. งสูปและอภิปรายผล.

รายการ 3.22 แสดงตัวอย่างโปรแกรมคำนวณเกรเดียนต์เชิงเลขแบบทางเดียว (one-sided numerical gradient calculation) ตามสมการ 3.51. ตัวอย่างการใช้งาน คือ

```
dE = nGrad_oneside(cross_entropy, x, y, net, epsilon=1e-4)
```

เมื่อ `cross_entropy` คือฟังก์ชันจุดประสงค์. ตัวแปร `x` และ `y` เป็นข้อมูล. ตัวแปร `net` เป็นแบบจำลองของโครงข่ายประสาทเทียม นิยามดังที่ได้อธิบาย. ผลลัพธ์ `dE` เป็นไฟรอนดิกชันนารี ที่เก็บค่าเกรเดียนต์ของค่าน้ำหนักและไบอส ที่เรียกว่า `weight1` และ `bias1` คือ เกรเดียนต์ของค่าน้ำหนักและไบอสของชั้นคำนวณที่หนึ่ง.

รายการ 3.22: โปรแกรมหาเกรเดียนต์เชิงเลข

---

```

1 def nGrad_oneside(lossf, datX, datY, net, epsilon=1e-4):
2     num_layers = net['layers']
3     ngrad = {'layers': num_layers}
4
5     yc = mlp(net, datX)
6     lossc = lossf(yc, datY)
7
8     for i in range(1, num_layers):
9         # weight
10        w = net['weight%d'%i]
11        gradw = np.zeros(w.shape)
12
13        nr, nc = gradw.shape
14        for r in range(nr):
15            for c in range(nc):
16                wu = w.copy()
17                wu[r, c] += epsilon
18
19                net_ = net.copy()
20                net_[ 'weight%d'%i] = wu
21                yu = mlp(net_, datX)
22                lossu = lossf(yu, datY)
23                gradw[r,c] = (lossu - lossc)
24
25                ngrad[ 'weight%d'%i] = gradw/epsilon

```

```
26      # bias
27      b = net['bias%d'%i]
28      gradb = np.zeros(b.shape)
29
30      nr, _ = gradb.shape
31      for r in range(nr):
32          bu = b.copy()
33          bu[r] += epsilon
34
35          net_ = net.copy()
36          net_[ 'bias%d'%i] = bu
37          yu = mlp(net_, datX)
38          lossu = lossf(yu, datY)
39          gradb[r,0] = (lossu - lossc)
40          ngrad[ 'bias%d'%i] = gradb/epsilon
41      return ngrad
```

---



## บทที่ 4

### การเรียนรู้ของเครื่องในโลกกว้าง

``Failure is the key to success;  
each mistake teaches us something."

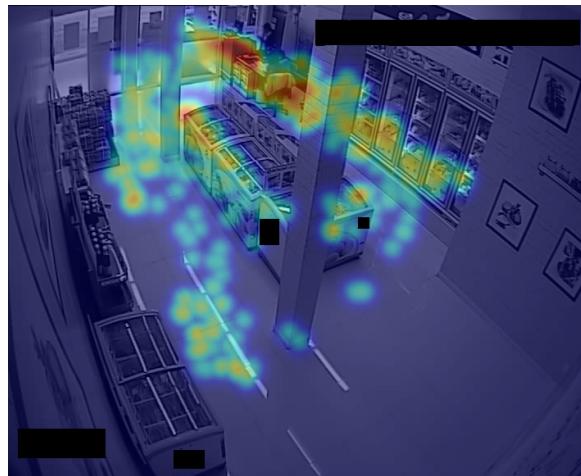
---Morihei Ueshiba

“ความล้มเหลวเป็นกุญแจสู่ความสำเร็จ  
แต่ละความผิดพลาดสอนเราบางอย่าง.”  
—โมเรไฮ อุเชิบะ

บทนี้ นำเสนอตัวอย่างการนำวิธีการเรียนรู้ของเครื่องไปประยุกต์ใช้กับงานการรู้จำรูปแบบ รวมถึง วิธีการเรียนรู้ของเครื่องแบบต่าง ๆ ที่หลากหลาย. หัวข้อ 4.1 นำเสนอตัวอย่างระบบบิเคราะห์พุติกรรมลูกค้า ที่ใช้แบบจำลองการเรียนรู้ของเครื่องประกอบอยู่ในระบบ ทั้งในส่วนการตรวจจับตำแหน่งลูกค้า และการแสดงผล. หัวข้อ 4.1 อภิปราย เทคนิคหลาย ๆ อย่างที่ใช้ประกอบกับวิธีการเรียนรู้ของเครื่อง เพื่อสามารถใช้กับงานการรู้จำรูปแบบได้อย่างมีประสิทธิผล. หัวข้อ 4.2 อภิปราย ชัพพอร์ตเวกเตอร์แมชชีน ซึ่งเป็นหนึ่งในวิธีการเรียนรู้ของเครื่องอีกที่ได้รับความนิยมอย่างมาก. ชัพพอร์ตเวกเตอร์แมชชีน ออกแบบและพัฒนาโดยอาศัยศาสตร์การหาค่าดีที่สุด มีทฤษฎีรองรับมั่นคง และมีเบื้องหลังการออกแบบที่ сложสลวย.

#### 4.1 การวิเคราะห์พุติกรรมลูกค้า

เนื้อหาในหัวข้อนี้ ได้รับอิทธิพลหลัก ๆ จากโครงการวิจัยการวิเคราะห์พุติกรรมลูกค้าผ่านข้อมูลวิดีโอจากกล้องวงจรปิด[106].



รูปที่ 4.1: ตัวอย่างภาพสรุป ซึ่งเป็นภาพช้อนของภาพถ่ายแสดงผังบริเวณร้านจากกล้องวงจรปิด และห้องทับด้วยแผนภาพความร้อนที่มีลักษณะกึ่งโปรงใส. แผนภาพความร้อนแสดงบริเวณที่พบลูกค้าบ่อยที่สุดด้วยสีแดง และสีโภนที่เย็นลงสื่อถึงความถี่ที่พบลูกค้าลดลง และความถี่ต่ำที่สุดแทนด้วยสีที่เย็นที่สุดคือสีน้ำเงิน (ภาพนี้เป็นภาพสี และในภาพได้ทำการปิดบังตราสัญลักษณ์ของร้านค้าไว้)

## การวิเคราะห์พฤติกรรมลูกค้าจากภาพวิดีโอ

ข้อมูลวิดีโอด้วยกล้องวงจรปิดในร้านค้าปลีก นอกจากใช้เพื่อเหตุผลด้านความปลอดภัยแล้ว ข้อมูลวิดีโอยังสามารถนำมาใช้ เพื่อการวิเคราะห์พฤติกรรมการจับจ่ายของลูกค้าที่เข้ามาภายในร้านได้. การเข้าใจพฤติกรรมของลูกค้าสามารถนำมาช่วยเพิ่มโอกาสทางธุรกิจได้ เช่น ความเข้าใจบริเวณและเวลาที่ลูกค้าใช้ขณะเข้ามาภายในร้าน สามารถใช้ประกอบการตัดสินใจ สำหรับการจัดวางสินค้า หรือการจัดกิจกรรมส่งเสริมการตลาดได้.

ระบบวิเคราะห์พฤติกรรมอัตโนมัติจะช่วยอำนวยความสะดวกในการวิเคราะห์เบื้องต้น ถึงพฤติกรรมของลูกค้าที่เข้ามาในร้าน โดยใช้ข้อมูลวิดีโอด้วยกล้องวงจรปิด. ระบบตัวอย่างนี้ วิเคราะห์พฤติกรรม ซึ่งคือ การตรวจหาตำแหน่งของลูกค้า เมื่อลูกค้าเข้ามาใช้บริการภายในร้านค้า และนำเสนอผลสรุป เป็นแผ่นภาพสรุป. แผ่นภาพสรุปที่ได้ สามารถนำไปช่วยประกอบการตัดสินใจด้านการตลาด และอาจช่วยให้เข้าใจรูปแบบการเข้าใช้พื้นที่ในบริเวณร้านได้ดีขึ้น.

ข้อมูลจากการกล้องวงจรปิดภายในร้านค้าปลีก ถูกนำมาใช้เพื่อประมวลผล และตรวจหาตำแหน่งของลูกค้า แล้วจัดทำแผนภาพสรุป. รูป 4.1 แสดงตัวอย่าง ของรูปแบบภาพสรุป ที่จัดทำเป็นรูปแบบแผนที่ความร้อน (heat map). ภาพสรุปแสดงบริเวณที่พบลูกค้าบ่อยที่สุดด้วยสีแดง และสีโภนที่เย็นลงสื่อถึงความถี่ที่พบลูกค้าลดลง (ความถี่ต่ำที่สุด แทนด้วยสีที่เย็นที่สุด คือ สีน้ำเงิน).

## ขั้นตอนการทำงานของระบบวิเคราะห์พฤติกรรมลูกค้าจากภาพวิดีโอ

ตัวอย่างระบบวิเคราะห์พฤติกรรมอัตโนมัตินี้ มีส่วนประกอบหลักสองส่วน ได้แก่ (1) ส่วนการตรวจหาและระบุตำแหน่งของลูกค้าจากข้อมูลวิดีโอ และ (2) ส่วนนำเสนอผล ที่นำตำแหน่งของลูกค้าที่ตรวจหาได้มาสรุปและแสดงผลเป็นแผนที่ความร้อน.

### ส่วนของการตรวจหาและระบุตำแหน่งของลูกค้า จากข้อมูลวิดีโอ

ลักษณะงานวิเคราะห์พฤติกรรมลูกค้า เป็นงานลักษณะออฟไลน์ (offline mode) หรือลักษณะการประมวลผลเป็นชุด (batch processing ที่ไม่ใช่ระบบเวลาจริง real-time processing). เช่นเดียวกับงานของเบนเนนสันและคณะ[11] ที่ศึกษาการตรวจหาและระบุตำแหน่งคนเดินเท้า ซึ่งเป็นงานในลักษณะใกล้เคียงกัน งานการตรวจหาและระบุตำแหน่งลูกค้านี้ ดำเนินการโดย การแปลงจากวิดีโอมาเป็นชุดลำดับของภาพ แล้วจึงใช้วิธีการประมวลผลภาพ เพื่อตรวจหาและระบุตำแหน่งของลูกค้า.

การระบุตำแหน่งลูกค้าอาจทำได้หลายวิธี เช่น (ก) แนวทางวิธีลบฉากพื้นหลัง และ (ข) แนวทางวิธีตรวจหาภาพวัตถุ.

#### แนวทางวิธีลบฉากพื้นหลัง

แนวทางวิธีลบฉากพื้นหลัง ไม่ได้ใช้เทคนิคใดจากศาสตร์การเรียนรู้ของเครื่อง แต่ใช้อัศย์เทคนิคการประมวลผลภาพ และอัศย์ลักษณะเฉพาะของข้อมูล. จากการสังเกตลักษณะเฉพาะของข้อมูล พื้นหลังของภาพวิดีโอด้วยตัวเอง ค่อนข้างคงที่ ส่วนใหญ่ สิ่งที่เคลื่อนที่ในภาพมักจะเป็นคน ที่รวมทั้งลูกค้าและพนักงาน. แนวทางวิธีลบฉากพื้นหลัง อัศย์ลักษณะเฉพาะของข้อมูลนี้เข้ามาช่วย.

แนวทางวิธีลบฉากพื้นหลัง (Background Subtraction) แสดงดังรูปที่ 4.2 ซึ่งเป็นแผนภูมิลำดับกระบวนการ. ข้อมูลวิดีโอ (ภาพซ้ายล่าง) จะถูกแปลงเป็นเฟรมภาพ (ขั้นตอน 1.1). หลังจากนั้น นำแต่ละภาพที่ได้ไปลบออกจากภาพพื้นหลัง (background หรือภาพในบริเวณร้านที่ไม่มีลูกค้าอยู่) รวมถึงการปรับปรุงเพื่อขยายความต่างให้ชัดเจนขึ้น (ขั้นตอนที่ 1.2a – 1.2c). สุดท้าย ทำการค้นหาและระบุตำแหน่งลูกค้าในภาพ (ขั้นตอนที่ 1.3a – 1.3b).

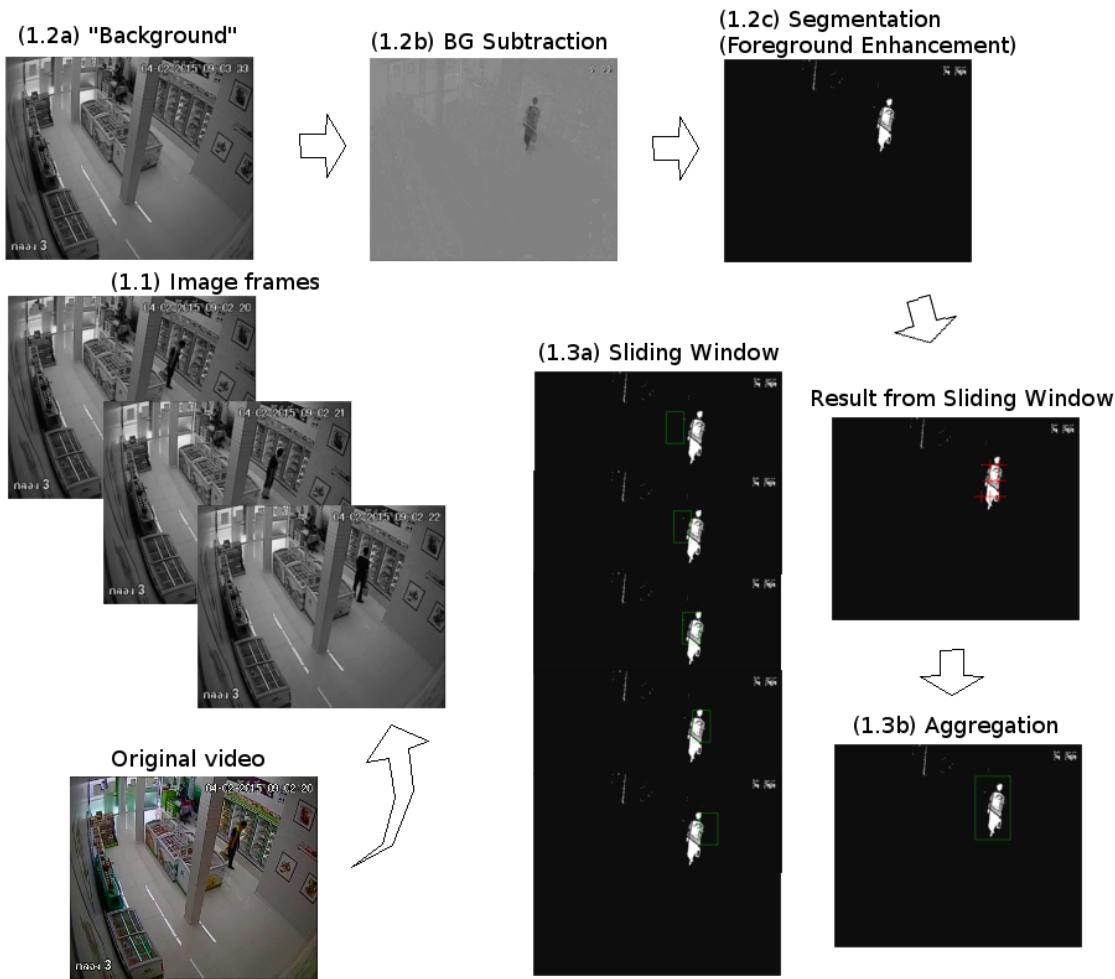
ขั้นตอนที่ 1.1. การแปลงจากวิดีโอไฟล์เป็นภาพ มีกระบวนการทางเทคนิคที่มีรายละเอียดมาก รวมไปถึงข้อกำหนดต่าง ๆ ตามมาตรฐานของชนิดข้อมูลวิดีโอ แต่ในทางปฏิบัติมีเครื่องมือที่ช่วยทำงานเหล่านี้ได้มากมาย

หนึ่งในเครื่องมือที่นิยมใช้ ก็เช่น โอเพ่นซีวี (OpenCV<sup>1</sup>) ซึ่งเป็นไลบรารีหัสรีปิด (open-source library) สำหรับงานด้านหัศนศาสตร์พิวเตอร์ (Computer Vision). ขั้นตอนที่ 1.2 การปรับปรุงภาพ เพื่อให้ตำแหน่งของลูกค้าเด่นชัดขึ้น เพื่อช่วยให้การตรวจหาตำแหน่งของลูกค้าในขั้นตอนต่อไปทำได้ง่าย. ขั้นตอนย่อยในการปรับปรุงภาพนี้ ได้แก่ เทคนิคไวรีลิฟพื้นหลัง ที่นำภาพพื้นหลังไปลบออกจากภาพที่ต้องการตรวจหาตำแหน่งลูกค้า. สิ่งที่ต่างจากพื้นหลังก็จะปรากฏเด่นขึ้น (1.2a และ 1.2b) และ เทคนิคไวรีแยกส่วนโดยการกำหนดระดับค่าขีดแบ่ง (segmentation by thresholding) ก็สามารถนำมาใช้ เพื่อแยกจากหน้า (ที่อาจเป็นลูกค้า) ออกจากฉากหลัง. เทคนิคไวรีแยกส่วนโดยการกำหนดระดับค่าขีดแบ่ง คือการตั้งค่าระดับขีดแบ่งขึ้นมาหนึ่งค่า โดย หากค่าความเข้มของพิกเซลมากกว่าระดับนั้น ก็จะปรับเพิ่มค่าความเข้มของพิกเซลนั้นไปจนมีค่ามากที่สุด (สว่างเต็มที่) ไม่ เช่นนั้นก็ปรับลดค่าลงเป็นศูนย์ (มืดเต็มที่). ดังนั้น ภาพจะถูกแยกเป็นส่วนสว่างและมืดอย่างชัดเจน. ขั้นตอน 1.3 เป็นการนำภาพที่ปรับปรุงความต่างระหว่างฉากหน้าและฉากหลัง ไปตรวจหาตำแหน่งของลูกค้า ซึ่งใช้เทคนิคการตรวจหาแบบหน้าต่างเลื่อน (sliding window detection). เทคนิคการตรวจหาแบบหน้าต่างเลื่อน เป็นหนึ่งในวิธีที่นิยมใช้สำหรับงานการตรวจหาภาพวัตถุ. เทคนิคการตรวจหาแบบหน้าต่างเลื่อน เป็นเทคนิคการเลือกส่วนภาพ โดยส่วนภาพ(ในขนาดที่พอเหมาะสมแก่การตรวจสอบว่าเป็นวัตถุที่ต้องการ หรือไม่) จะถูกเลือกขึ้นมาจากการที่ส่อง. การเลือกจะเริ่มที่มุมด้านหนึ่งของภาพ และขยายไปเรื่อย ๆ จนครอบคลุมทั้งภาพ. ส่วนภาพขนาดเล็กที่ถูกเลือกออกอกรามันจะถูกตรวจสอบว่ามีวัตถุที่ค้นหาอยู่ในส่วนภาพนั้นหรือไม่ (ขั้นตอน 1.3a). เมื่อได้ผลการตรวจสอบส่วนภาพขนาดเล็กจากตำแหน่งต่าง ๆ แล้ว ส่วนภาพต่าง ๆ ที่ได้ผลการตรวจเป็นบวกที่มีตำแหน่งใกล้ ๆ กัน จะถูกรวบกันตามเกณฑ์การรวมที่กำหนด และตำแหน่งของการรวมนั้นจะถูกบันทึกเป็นตำแหน่งของลูกค้าทั้งตัว (ขั้นตอน 1.3b). เมื่อถึงขั้นตอนนี้ ระบบก็จะสามารถระบุได้แล้วว่า ลูกค้าอยู่ที่ตำแหน่งใดในภาพ.

### แนวทางของวิธีตรวจหาวัตถุ

กลไกจริง ๆ ของวิธีการลบฉากหลัง คือ การตรวจหาตำแหน่งของวัตถุที่ต่างจากฉากหลัง ซึ่งสมมติฐานก็คือ น่าจะเป็นลูกค้า. วิธีการลบฉากหลังมีข้อดีคือ สามารถทำได้ง่ายและรวดเร็ว (ในกรณีที่มีภาพฉากหลังแล้ว). แต่ ข้อเสีย คือความสามารถดังกล่าวไม่สามารถขยายไปใช้ในการแยกวัตถุอื่นได้ เช่น ยกที่จะแยกคนกับรถเข็นออกจากกันได้ และไม่สามารถนำไปใช้ในกรณีที่ฉากหลังมีการเปลี่ยนแปลงสูงได้ เช่น ไม่สามารถนำไปใช้ตรวจจับภาพคนเดินถนน ที่จากหลังเป็นสภาพการจราจรได้ และก็ไม่สามารถใช้ในกรณีคนมีการขับน้อย เช่น ในสวนที่มีคนนั่งอ่านหนังสือ ฝึกสมาธิ หรือนอนหลับอยู่.

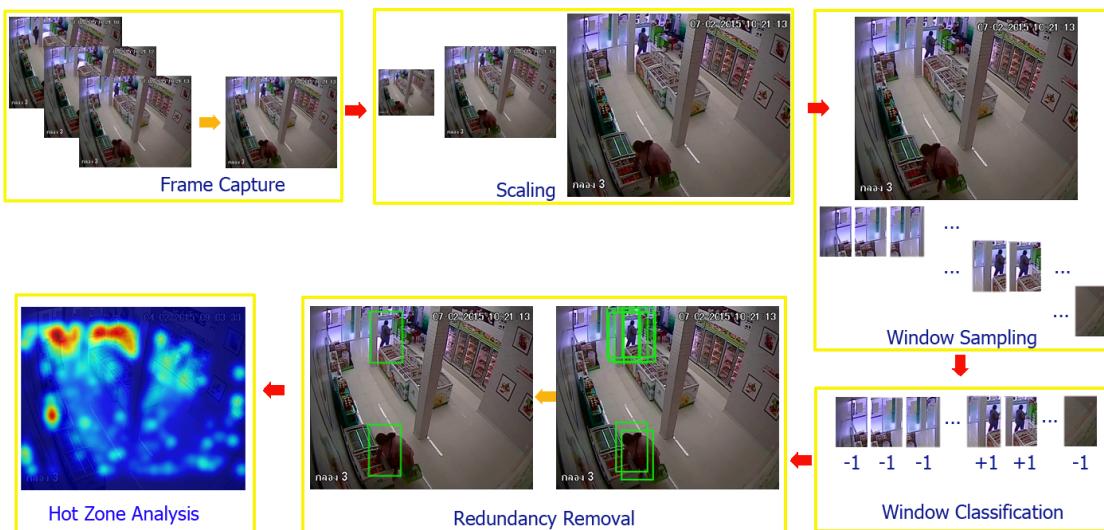
<sup>1</sup><http://opencv.org/>



รูปที่ 4.2: แผนภาพแสดงขั้นตอนต่าง ๆ ในการตรวจหาตำแหน่งลูกค้าจากข้อมูลวิดีโอด้วยแนววิธีการลบทกหลัง

อีกแนวทางหนึ่งที่สามารถทำได้ คือ แนวทางวิธีตรวจหาวัตถุ. การตรวจหาวัตถุ (object detection) เป็นภารกิจการหาตำแหน่งของวัตถุในภาพ หากในภาพมีวัตถุปรากฏอยู่. แนวทางของวิธีตรวจหาวัตถุมีพื้นฐานมาจากสมมติฐานทางสถิติ เช่น รูปทรงของคนแม้จะมีความหลากหลาย แต่ก็มีรูปแบบและลักษณะร่วมกันอยู่มาก และวิธีการคือ การหาลักษณะร่วมของเหล่านั้นออกมานี้ และใช้มันช่วยในการจำแนกระหว่างคนกับสิ่งอื่น ๆ ที่ไม่ใช่คน. รูปที่ 4.3 แสดงขั้นตอนย่อย ๆ ในการตรวจหาตำแหน่งของลูกค้าด้วยวิธีการตรวจหาวัตถุ โดยเริ่มตั้งแต่

1. การแปลงข้อมูลวิดีโอดอกมาเป็นข้อมูลภาพหลาย ๆ ภาพ (frame capture) เพื่อเท่าขั้นตอน 1.1 ของรูปที่ 4.2 ของแนวทางการลบทกหลัง. นั่นคือ การปรับปัญหาจากการทำงานกับข้อมูลวิดีโอด้วยปัญหาเดียวกันที่ทำงานกับภาพแทน.
2. การย่อและขยายภาพให้อยู่ในหลาย ๆ ขนาด (scaling) เพื่อให้วัตถุที่อยู่ห่างกล้องในระยะต่าง ๆ มีโอกาสที่จะถูกตรวจพบใกล้เคียงกัน.

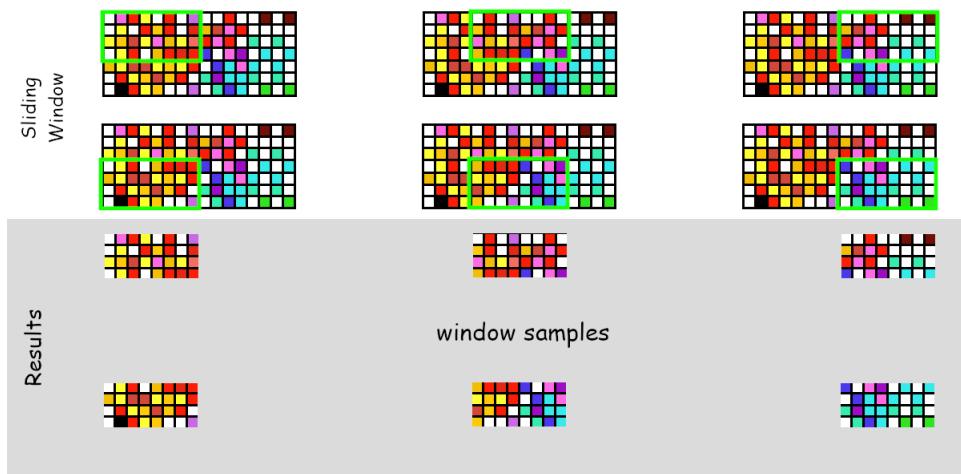


รูปที่ 4.3: แผนภาพแสดงขั้นตอนต่าง ๆ ในการตรวจหาตำแหน่งลูกค้าจากข้อมูลวิดีโอด้วยแนววิธีการตรวจจับภาพวัตถุ

3. การเลือกส่วนภาพ (window sampling). แนวทางนี้ ก็ยังคงเลือกใช้เทคนิคหน้าต่างเลื่อน. นั่นคือ ตีกรอบปัญหาการตรวจจับภาพวัตถุ เป็นปัญหาการจำแนกค่าทวิภาค โดยใช้การเลือกส่วนภาพอุ กมาเพื่อเป็นอินพุตของแบบจำลองจำแนก (ในขั้นตอนต่อไป).
4. การจำแนกส่วนภาพว่ามีภาพคนอยู่หรือไม่ (window classification). ภาพขนาดเท่า ๆ กัน (ได้จาก การสุ่มภาพด้วยวิธีหน้าต่างเลื่อน) จะถูกผ่านเข้าแบบจำลองจำแนกข้อมูล โดยแบบจำลองจะทำหน้าที่ ทำนายว่าภาพอินพุตนั้น มีภาพของวัตถุที่ต้องการ (ในที่นี้คือ ภาพคน) หรือไม่.
5. การลดผลการตรวจหาที่ซ้ำซ้อน (redundancy removal). การสุ่มด้วยวิธีหน้าต่างเลื่อน มักทำใน ลักษณะที่มีการซ้อนทับกัน เพื่อป้องกันการตัดกรอบที่คลางแคลง แต่การสุ่มในลักษณะนี้ก็อาจทำให้ การสุ่มในตำแหน่งใกล้เคียงกัน ตรวจพบวัตถุเดียวกันได้. ดังนั้น หลังการตรวจพบวัตถุแล้ว จึงต้องมี การทำการลดการซ้ำซ้อนลง. หลังจากขั้นตอนการลดการซ้ำซ้อน เราจะสามารถระบุตำแหน่งของวัตถุ ที่ตรวจพบได้.

จากนั้น ผลการตรวจหาที่ได้จะนำไปสรุป และแสดงผลด้วยวิธีแผนที่ความร้อน (hot zone analysis ซึ่ง เป็นขั้นตอนที่ 6 ในรูป 4.3). ขั้นตอนที่ 2 ถึง 5 คือขั้นตอนที่ต่างจากแนวทางวิธีลับจากหลัง (เปรียบเทียบกับ 1.2a ถึง 1.3b ในรูปที่ 4.2) ถึงแม้ทั้งสองแนวทางจะใช้กลไกของการเลือกส่วนภาพด้วยวิธีหน้าต่างเลื่อนก็ตาม.

การทำเทคนิคหน้าต่างเลื่อน. วิธีหน้าต่างเลื่อน[206] เป็นวิธีการเลือกส่วนภาพขนาดที่กำหนดจากข้อมูล ภาพใหญ่ โดย การเลือกส่วนภาพจะเลือกทั่วถึงจากทุกบริเวณในภาพใหญ่ โดยอาจเริ่มจากมุมซ้ายบนของภาพ



**รูปที่ 4.4:** ภาพแสดงตัวอย่างการเลือกส่วนภาพด้วยวิธีหน้าต่างเลื่อน. ภาพใหญ่และกรอบหน้าต่างเลือก แสดงในส่วนบน (พื้นหลัง สีขาว). กรอบหน้าต่างเลือก แสดงด้วยกรอบสีเขียว. ส่วนภาพที่เลือกมาจากการขับหน้าต่างแต่ละครั้ง แสดงในส่วนล่าง (พื้นหลังสีเข้ม). ในตัวอย่าง ภาพใหญ่ขนาด  $16 \times 7$  (กว้าง คูณ สูง). ขนาดกรอบหน้าต่างเลือกเป็น  $8 \times 4$  และขนาดขับเลื่อนเป็น  $4 \times 3$ .

ใหญ่ เลือกส่วนภาพอ กมา แล้วขับไปทางขวา และทำเช่นนี้ไปจนสุดปลายด้านขวา แล้วจึงขับลงล่างและไปเริ่มจากซ้ายสุด และทำลักษณะเช่นนี้อีก จนครอบคลุมบริเวณทั้งภาพใหญ่. ลำดับของภาพที่เลือกอ กมา จะคล้ายกับลำดับของภาพที่มองจากหน้าต่างที่เลื่อนไปตามแน่นอนๆ ของภาพใหญ่ ดังนั้นเทคนิคนี้จึงเรียกว่า เทคนิคหน้าต่างเลื่อน. ขนาดของหน้าต่าง (window size) ซึ่งคือขนาดของส่วนภาพที่เลือก และขนาดของการขับหน้าต่าง ที่มักเรียกว่า **ขนาดขับเลื่อน** (stride) ซึ่งเป็นจำนวนพิกเซลของการขับการเลือกส่วนภาพแต่ละครั้ง เป็นอภิธานพารามิเตอร์ของวิธีหน้าต่างเลื่อน.

รูป 4.4 แสดงตัวอย่างการทำงานของวิธีหน้าต่างเลื่อน. ในตัวอย่าง ภาพใหญ่ขนาด 16 พิกเซล กรอบหน้าต่างเลือกกว้าง 8 พิกเซล และขนาดขับเลื่อนแนวนอนเป็น 4 พิกเซล ดังนั้นจึงสามารถขับได้ 3 ตำแหน่งในแนวนอน (ได้แก่ เริ่มต้นที่พิกเซล 0, ขับไปพิกเซล 4, และขับไปพิกเซล 8). จำนวนตำแหน่งของหน้าต่างในแนวนอนและแนวตั้งสามารถเขียนเป็นการคำนวณทั่วไปได้ดังนี้ กำหนดให้เมตริกซ์  $\mathbf{F} = [f_{m,n}]$ ,  $m = 0, \dots, C - 1$  และ  $n = 0, \dots, R - 1$  แทนภาพใหญ่ขนาด  $C \times R$  และ  $f_{m,n} \in \mathbb{I}$  เป็นค่าความเข้มของพิกเซลที่ตำแหน่ง  $(m, n)$  และให้  $\mathbf{W}_{ij} \in \mathbb{I}^{A \times B}$  แทนส่วนภาพขนาด  $A \times B$  ของด้านหน้าต่าง  $(i, j)$ . หากขนาดขับเลื่อนตามแนวนอนและตั้งเป็น  $(a, b)$  แล้ว วิธีหน้าต่างเลื่อน เป็นสมือนฟังก์ชันแปลง  $S : \mathbf{F} \mapsto \{\mathbf{W}_{ij}\}$  สำหรับด้าน  $i = 0, \dots, \lfloor \frac{C-A}{a} \rfloor$  และด้าน  $j = 0, \dots, \lfloor \frac{R-B}{b} \rfloor$ . แต่ละส่วนภาพ  $\mathbf{W}_{ij} = [w_{p,q}(i, j)]$  เมื่อ  $p = 0, \dots, A - 1$  และ  $q = 0, \dots, B - 1$  เป็นเมตริกซ์ป้อม  $w_{p,q}(i, j) = f_{a \cdot i + p, b \cdot j + q}$ .

**รูป 4.5** แสดงตัวอย่างการใช้วิธีหน้าต่างเลื่อนกับงานการตรวจจับภาพเป้าหมาย. ผลลัพธ์ที่ได้คือส่วนภาพ

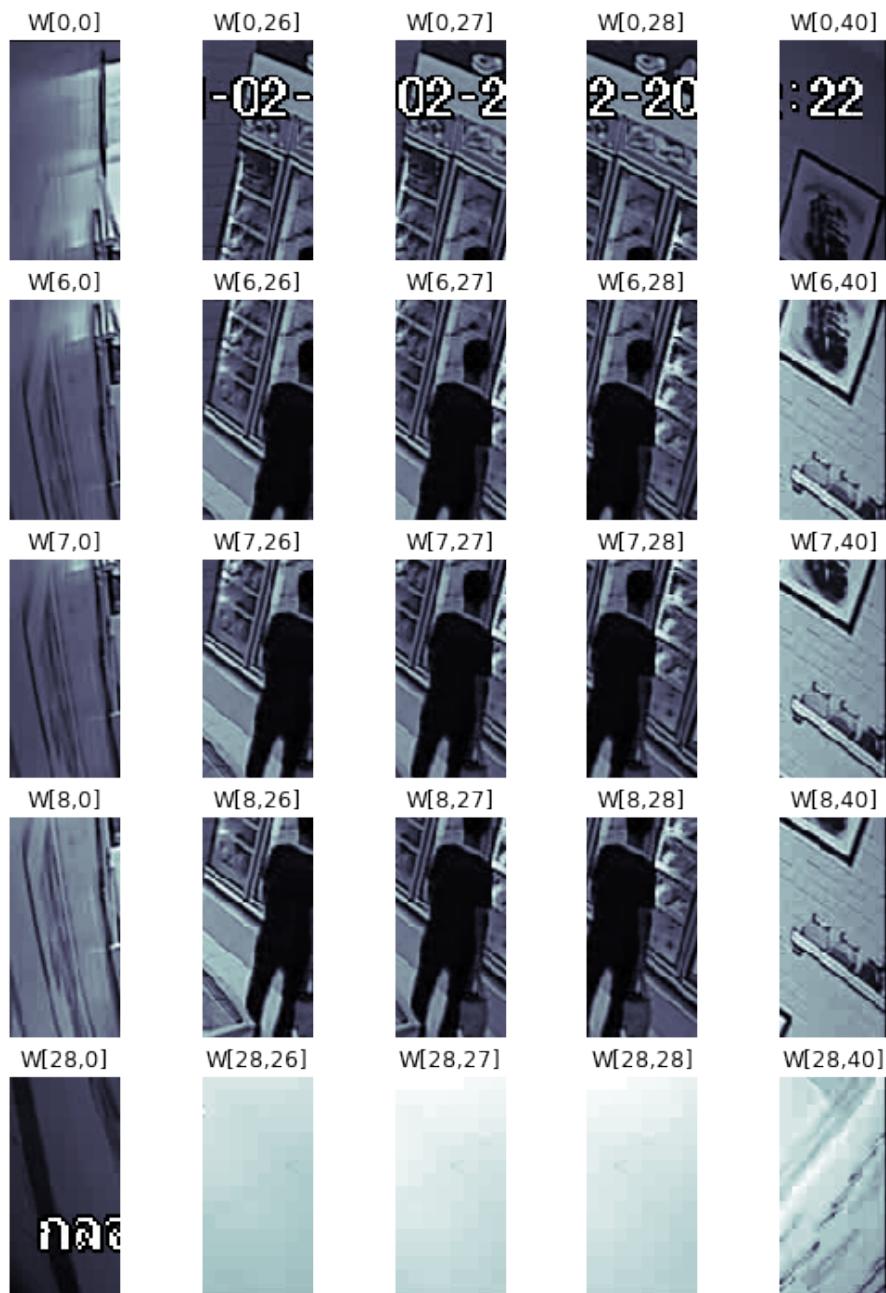
ต่าง ๆ ส่วนภาพต่าง ๆ จะถูกนำมาไปตรวจสอบ ด้ัชนีของส่วนภาพที่พบว่ามีเป้าหมายอยู่ จะถูกนำไปใช้ระบุตำแหน่งของเป้าหมายในภาพใหญ่ (หากพบเป้าหมายในภาพ). จากอภิมานพารามิตเตอร์ที่ใช้ (หน้าต่างขนาด  $128 \times 64$  ใช้ขนาดขับเลื่อนเป็น 16) และขนาดของภาพใหญ่ ( $576 \times 704$ ) หลังดำเนินการวิธีหน้าต่างเลื่อน จะได้ส่วนภาพ  $\mathbf{W}_{0,0}, \dots, \mathbf{W}_{28,40}$  (จาก  $\lfloor \frac{576-128}{16} \rfloor = 28$  และ  $\lfloor \frac{704-64}{16} \rfloor = 40$ ). แต่ละส่วนภาพตัดมาจากภาพใหญ่ เช่น

$$\mathbf{W}_{7,27} = \begin{bmatrix} f_{112,432} & \dots & f_{112,495} \\ \vdots & \ddots & \vdots \\ f_{239,432} & \dots & f_{239,495} \end{bmatrix}$$

เมื่อ  $f_{m,n}$  คือค่าความเข้มพิกเซลของภาพใหญ่ที่ตำแหน่งตามแนวตั้ง  $m$  และแนวนอน  $n$ .

**การจำแนกและระบุตำแหน่งวัตถุ.** สำหรับแต่ละส่วนภาพที่เลือกมา  $\mathbf{W}_{ij}$  แบบจำลองจำแนกค่าทวิภาคสามารถใช้เพื่อท่านายว่าในส่วนภาพมีวัตถุเป้าหมายอยู่หรือไม่. นั่นคือ แบบจำลองจำแนกค่าทวิภาค เป็นสมือนฟังก์ชันแปลง  $f : \mathbf{W}_{ij} \mapsto y_{ij}$  เมื่อ  $y_{ij}$  คือค่าทวิภาคที่ท่านาย ซึ่งในกรณีตัวอย่างนี้นิยามเป็น +1 (มีเป้าหมายที่ค้นหาอยู่) หรือ -1 (ไม่มีเป้าหมายที่ค้นหาอยู่). หาก  $y_{ij} = 1$  นั้นหมายถึง ส่วนภาพ  $\mathbf{W}_{ij}$  มีเป้าหมายอยู่ และตำแหน่งของเป้าหมาย ก็คือตำแหน่งต่าง ๆ ที่  $\mathbf{W}_{ij}$  ครอบคลุม.

การแปลงจากส่วนภาพเป็นค่าทวิภาค ในตัวอย่างนี้ จะดำเนินเป็นสองขั้นตอน ได้แก่ (1) การแปลงส่วนภาพ ที่มักมีจำนวนมิติสูงมาก เป็นลักษณะสำคัญ ที่มีจำนวนมิติน้อยลงและเกี่ยวข้องการจำแนกเป้าหมายจากสิ่งอื่น ๆ และ (2) การแปลงลักษณะสำคัญที่ได้จากขั้นตอนแรกไปเป็นค่าทวิภาคที่ต้องการทำนาย. ลักษณะสำคัญ เป็นตัวแทนของอินพุตตันฉบับในแบบที่ช่วยให้ภาระกิจเป้าหมายดำเนินการได้ง่ายขึ้น. การเทคนิคที่สำคัญต่าง ๆ ในงานการตรวจจับภาพวัตถุ ล้วนเกี่ยวข้องโดยตรงกับการทำลักษณะสำคัญแทนของอินพุตตันฉบับ ไม่ว่าจะเป็น ลักษณะhaar (Haar features[206]) หรือ แผนภูมิแท่งของทิศทางเกรเดียนต์ (Histogram of Oriented Gradient) หรือ ถุงของทศนะถ้อยคำ (Bag of Visual Words[67]) เป็นต้น. ลักษณะสำคัญที่พัฒนาขึ้นมาในยุคหลัง ๆ อาจสร้างขึ้นจากลักษณะสำคัญที่พัฒนามาก่อนแล้ว เช่น แบบจำลองแปลงรูป (deformable model[68]) ที่ใช้แผนภูมิแท่งของทิศทางเกรเดียนต์ เป็นพื้นฐาน. แม้แต่แนวทางการเรียนรู้เชิงลึก (บท 5) ที่โดยทั่วไปแล้ว ไม่จำเป็นต้องดำเนินการแปลงเป็นสองขั้นตอน นั่นคือ ไม่ต้องเตรียมลักษณะสำคัญให้แบบจำลอง และสามารถรับอินพุตเป็นภาพตันฉบับได้โดยตรง ก็มีการใช้ลักษณะสำคัญ เพียงแต่แบบจำลองทำการสร้างลักษณะสำคัญขึ้นได้เองจากข้อมูลจำนวนมากที่ใช้ฝึก.



**รูปที่ 4.5:** ตัวอย่างการใช้วิธีหน้าต่างเลื่อนกับงานการตรวจจับภาพเป้าหมาย. ภาพซ้ายແກวนสุด แสดงส่วนภาพแรก ( $W[0,0]$ ) ที่วิธีหน้าต่างเลื่อนเริ่มต้น. และภาพขวาແກวนล่างสุด แสดงส่วนภาพสุดท้าย ( $W[28,40]$ ) ที่วิธีหน้าต่างเลื่อนเลือกออกอกรมา เมื่อภาพใหญ่มีขนาด  $576 \times 704$  และหน้าต่างขนาด  $128 \times 64$  ใช้ขนาดขับเลื่อนเป็น 16.

ตัวอย่างนี้ดำเนินการจำแนกค่าทิวภาคของส่วนภาพโดย (1) ส่วนภาพ  $\mathbf{W}_{ij}$  จะถูกแปลงเป็นเวกเตอร์ลักษณะสำคัญ  $\mathbf{X}_{ij}$  และจากนั้น (2) ลักษณะสำคัญ  $\mathbf{X}_{ij}$  จะถูกแปลงเป็นฉลากทำนาย  $y_{ij} \in \{-1, +1\}$ . ตัวอย่างนี้ แสดงการใช้แผนภูมิแห่งของทิศทางเกรเดียนต์[49] เป็นลักษณะสำคัญ และใช้ชัพพอร์ตเวกเตอร์แมชีน[43] เป็นแบบจำลองทำนาย.

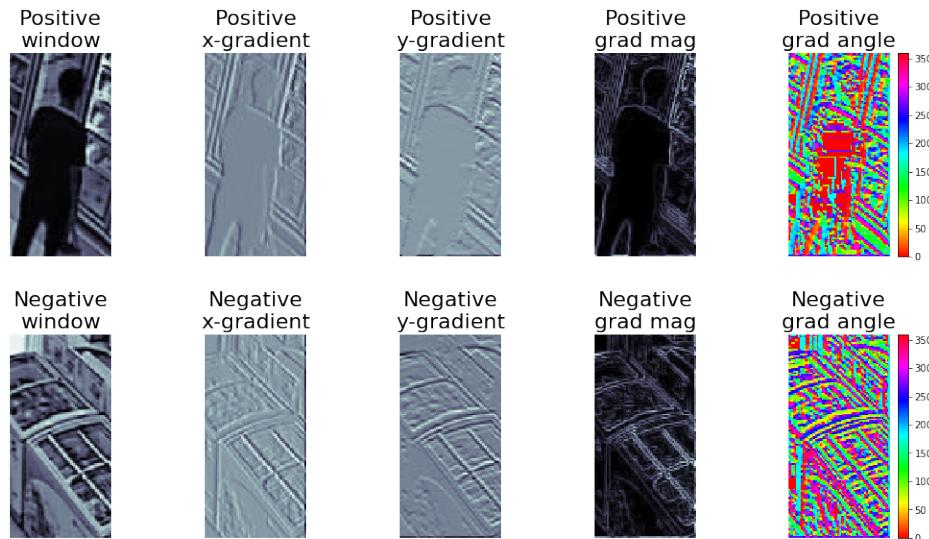
หมายเหตุ แนวทางนี้ เพียงตรวจจับตำแหน่งของบุคคลในภาพ และไม่ได้มีการจำแนกแยกแยะมโนคงติดับสูง ว่าบุคคลในภาพเป็นลูกค้าจริง ๆ หรือเป็นพนักงานร้าน. เพื่อจะวิเคราะห์ในมโนคงติดับสูงดังกล่าว รูปแบบของเครื่องแบบ และเส้นทางการเคลื่อนที่ อาจนำมาใช้ประกอบได้.

**การสกัดลักษณะสำคัญ.** แผนภูมิแห่งของทิศทางเกรเดียนต์ (Histogram of Oriented Gradients[49]) ที่มักย่อเป็น เอชโอจี (HOG) เป็นลักษณะสำคัญที่นิยมใช้ในงานคอมพิวเตอร์วิทัศน์. เอชโอจี เป็นฟังก์ชันแปลง  $H : \mathbf{W} \mapsto \mathbf{x}$  เมื่อ  $\mathbf{W} \in \mathbb{I}^{A \times B}$  เป็นเมตริกซ์ของค่าความเข้มพิกเซล และ  $\mathbf{x} \in \mathbb{R}^D$  เป็นเวกเตอร์ค่าลักษณะสำคัญเอชโอจี. โดยทั่วไปแล้ว ขนาดของเวกเตอร์  $\mathbf{x}$  จะเล็กกว่าขนาดของ  $\mathbf{W}$ มาก (นั่นคือ  $D \ll A \times B$ ). สมมติฐานของเอชโอจี คือการแจกแจงพิกเซลเกรเดียนต์ของภาพสามารถเป็นนัยที่บ่งชี้รูปร่างและสามารถใช้ระบุเป้าหมายได้.

เอชโอจีเริ่มด้วยการคำนวนพิกเซลเกรเดียนต์  $\mathbf{G}$ . พิกเซลเกรเดียนต์ที่ตำแหน่ง  $(r, c)$  เขียนด้วยสัญกรณ์  $\mathbf{g}_{rc} = [g_x(r, c), g_y(r, c)]^T$  เมื่อ  $g_x(r, c) = f_{r, c+1} - f_{r, c}$  และ  $g_y(r, c) = f_{r+1, c} - f_{r, c}$  และ  $f_{rc}$  คือค่าความเข้มพิกเซลของภาพที่ตำแหน่งแนวตั้ง  $r$  และแนวอน  $c$  โดย กำหนดให้  $f_{rc} \equiv 0$  เมื่อตั้งแต่  $r$  หรือ  $c$  เกินขอบเขตภาพ ( $r > R$  หรือ  $c > C$ ). พิกเซลเกรเดียนต์ สามารถเขียนในรูปขนาดและมุมได้ ด้วยสัญกรณ์  $\mathbf{g}_{rc} = m_{rc} \angle \theta_{rc}$  โดยขนาด  $m_{rc} = \sqrt{g_x^2(r, c) + g_y^2(r, c)}$  และมุม<sup>2</sup>  $\theta_{rc} = \arctan \frac{g_y(r, c)}{g_x(r, c)}$ . รูป 4.6 แสดงค่าพิกเซลเกรเดียนต์ของส่วนภาพตัวอย่างที่มีเป้าหมาย และที่ไม่มีเป้าหมายอยู่.

จากนั้น เอชโอจีดำเนินการโดยแบ่งส่วนภาพ  $\mathbf{W}$  เป็นส่วนย่อย ๆ และเรียกแต่ละส่วนย่อยว่า เชลล์ (cell). ในแต่ละเชลล์ เอชโอจีจัดทำข้อมูล โดยจินตนาการเป็นสมือนการทำแผนภูมิแห่ง โดย แต่ละแห่ง แทนค่าขนาดของเกรเดียนต์ในแต่ละทิศทาง (ทิศทางที่ใกล้เคียงกันจะถูกรวบอยู่ในแห่งเดียวกัน) และความสูงของแต่ละแห่งเรียกว่า โหวต (vote) คำนวนจากผลรวมขนาดของเกรเดียนต์ในทิศทางของแห่งนั้น ๆ. ขึ้นตอนสุดท้าย เชลล์ต่าง ๆ ที่จะถูกรวบกันเป็นบล็อก (block) ในลักษณะซ้อนทับกัน และค่าโหวตจากเชลล์ในบล็อกจะถูกรวบ

<sup>2</sup>หากเขียนให้สมบูรณ์ขึ้น คือ มุม  $\theta_{rc} = \arctan \frac{g_y(r, c)}{g_x(r, c)}$  เมื่อ  $g_x(r, c) > 0$ . มุม  $\theta_{rc} = \pi + \arctan \frac{g_y(r, c)}{g_x(r, c)}$  เมื่อ  $g_x(r, c) < 0$ . มุม  $\theta_{rc} = \pi/2$  เมื่อ  $g_x(r, c) = 0$  และ  $g_y(r, c) > 0$ . มุม  $\theta_{rc} = -\pi/2$  เมื่อ  $g_x(r, c) = 0$  และ  $g_y(r, c) < 0$ . มุม  $\theta_{rc}$  จะไม่มีความหมายถ้า  $g_x(r, c) = 0$  และ  $g_y(r, c) = 0$ .

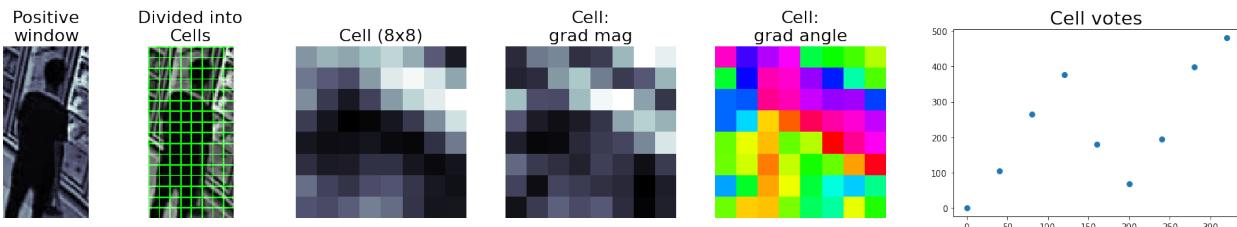


**รูปที่ 4.6:** ตัวอย่างพิกเซลเกรเดียโนต์ของภาพ. แถวบน แสดงตัวอย่างเมื่อส่วนภาพมีเป้าหมายอยู่. แถวล่าง แสดงตัวอย่างเมื่อส่วนภาพไม่มีเป้าหมายอยู่. ภาพช้ายสุดแสดงส่วนภาพที่ในสเกลเท่า. ภาพที่สองและสามจากช้าย แสดงพิกเซลเกรเดียโนต์ในแนวนอนและแนวตั้งตามลำดับ. ถัดอีกสองภาพ แสดงขนาด (grad mag สำหรับ gradient magnitude) และมุม (grad angle สำหรับ gradient angle) ของพิกเซลเกรเดียโนต์ ตามลำดับ. สำหรับภาพแสดงขนาดเกรเดียโนต์ สีขาวแทนขนาดที่มีค่ามาก และสีดำแทนขนาดที่มีค่าน้อย. สำหรับภาพแสดงมุม สีแทนองศาของมุม ตามที่ระบุด้วยແບสีด้านข้าง. สีที่ใช้สำหรับแสดงมุม ใช้ระบบสีลักษณะวัฏจักร เนื่องจาก 360 องศา เป็นทิศทางเดียวกับ 0 องศา.

อร์ไลซ์ภายในบล็อก. ค่าลักษณะสำคัญของเซ็ตโอจิ คือค่าให้ที่ถูกน้อมอร์ไลซ์แล้วจากบล็อกต่าง ๆ. นั่นคือ ขั้นตอนแรก คำนวณให้ที่ของเซลล์จากส่วนภาพ  $\mathbf{W}$  ที่มีขนาด  $A \times B$  โดยดำเนินการแปลง  $\mathbf{W} \mapsto \{\mathbf{v}_{ij}\}$  สำหรับดัชนีแนวตั้ง  $i = 0, \dots, \lfloor \frac{A}{h_v} \rfloor - 1$  และดัชนีแนวอน  $j = 0, \dots, \lfloor \frac{B}{w_v} \rfloor - 1$  เมื่อ  $\mathbf{v}_{ij}$  เป็นเวกเตอร์ ของเซลล์ให้ที่ ที่เซลล์มีขนาด  $h_v \times w_v$ .

หากเลือกจำนวนทิศทางของแผนภูมิที่จะเป็น  $K$  ทิศทาง เซลล์  $\mathbf{v}_{ij} \in \mathbb{R}^K$  จะมีส่วนประกอบที่  $k^{th}$  ของ เซลล์ ที่เขียนเป็นสัญกรณ์  $v_k(i, j)$  และสามารถคำนวณค่าได้จากผลรวมของขนาดของเกรเดียโนต์ในทิศทาง ที่  $k^{th}$  รวมถึงทิศทางใกล้เคียง ของพิกเซลที่อยู่ในขอบเขตของเซลล์. นั่นคือ หากส่วนภาพ  $\mathbf{W} = [w_{r,c}]$  โดย  $r = 0, \dots, A - 1$  และ  $c = 0, \dots, B - 1$  มีขนาดพิกเซลเกรเดียโนต์  $m_{rc}$  และมุมพิกเซลเกรเดียโนต์  $\theta_{rc}$  และเซลล์ให้ที่  $v_k(i, j) = \sum_{r,c \in \Omega_{cell}} m_{rc}$  สำหรับ  $k = 0, \dots, K - 1$  เมื่อเซต  $\Omega_{cell}$  แทนเรื่องไขพิกเซล ในขอบเขตของเซลล์ ได้แก่  $i \cdot h_v \leq r < (i + 1) \cdot h_v$  และ  $j \cdot w_v \leq c < (j + 1) \cdot w_v$  และเงื่อนไขทิศทาง ได้แก่  $\frac{k \cdot 360}{K} \leq \theta_{rc} < \frac{(k + 1) \cdot 360}{K}$ .

รูป 4.7 แสดงตัวอย่างการแบ่งส่วนภาพออกเป็นเซลล์. จากอภิมานพารามิเตอร์ของตัวอย่าง ส่วนภาพถูก แบ่งออกเป็นเซลล์  $\mathbf{v}_{0,0}, \dots, \mathbf{v}_{15,7}$  รวมทั้งหมด 128 เซลล์. ค่าขนาดของพิกเซลเกรเดียโนต์ในเซลล์จะถูก นำมาร่วมกันตามทิศทาง. ในภาพแสดงตัวอย่างเมื่อเลือกทำ 9 ทิศทาง ดังนั้นผลลัพธ์คือหนึ่งเซลล์จะมีส่วน



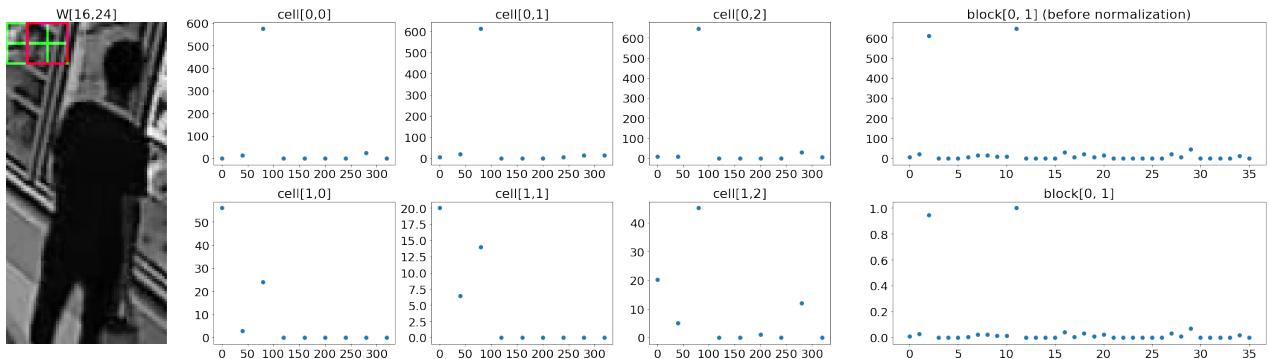
รูปที่ 4.7: ตัวอย่างแสดงการทำเอชโอลีเซลล์. ภาพข่ายสุดแสดงตัวอย่างส่วนภาพ. ภาพที่สองจากข่ายแสดงตัวอย่างส่วนภาพ พร้อมขอบเขตของแต่ละเซลล์ ซึ่งแสดงด้วยเส้นสีเขียว. เส้นสีเขียวในภาพทำเพื่อการแสดงผลให้เห็นขอบเขตของแต่ละเซลล์เท่านั้น. เส้นสีเขียวไม่ได้เกี่ยวข้องกับการทำลักษณะสำคัญเอชโอลี. ส่วนภาพขนาด  $128 \times 64$  ถูกแบ่งเป็นเซลล์ต่าง ๆ ที่แต่ละเซลล์ขนาด  $8 \times 8$ . ภาพที่สามแสดงเซลล์  $v_{0,0}$  ซึ่งเป็นเซลล์แรกอยู่มุมข้างบนของส่วนภาพ. ภาพที่สี่และห้าแสดงขนาดและมุมของพิกเซลเกรเดียนต์ของเซลล์  $v_{0,0}$ . ภาพสุดท้าย (ขวาสุด) แสดงค่าเซลล์ Howard เมื่อเลือกจำนวนทิศทาง  $K = 9$ .

ประกอบ 9 ตัวสำหรับทิศทาง  $0, 40, 80, 120, 160, 200, 240, 280, 320$  องศา. แต่ละทิศทางครอบคลุมทิศทางใกล้เคียง เช่น  $0$  องศา ครอบคลุม  $0 \leq \theta_{rc} < 40$ . หมายเหตุ ภาพในรูป 4.7 มีการใช้ค่าชาดเซย (offset) เพื่อใช้ทิศทางตัวแทนอยู่ตรงกลาง. นั่นคือใช้เงื่อนไขทิศทาง  $\frac{k \cdot 360}{K} + \delta \leq \theta_{rc} < \frac{(k+1) \cdot 360}{K} + \delta$  และใช้ค่าชาดเซย  $\delta = -\frac{360}{2K}$  ซึ่งในกรณีนี้คือ  $-20$ . นั่นทำให้  $0$  องศา ครอบคลุม  $-20 \leq \theta_{rc} < 20$ .

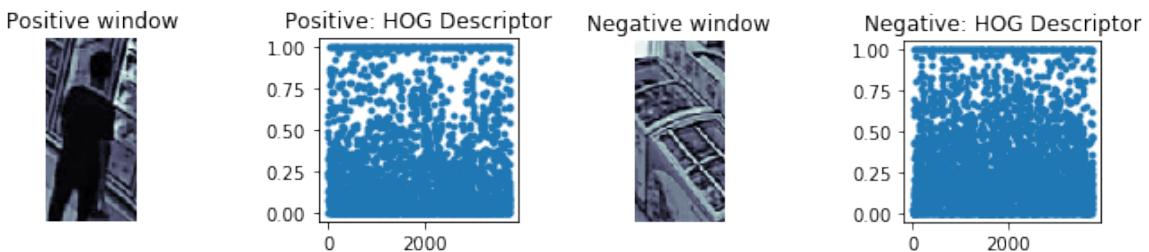
จากเซลล์ Howard ที่ได้ เพื่อลดผลกระทบของแสงและเงาในบริเวณต่าง ๆ เซลล์จะถูกรวบเป็นบล็อก. นั่นคือหากบล็อกมีขนาด  $n_y \times n_x$  เซลล์ และมีขนาดขัยบล็อกเป็น  $m_y \times m_x$  เซลล์ ส่วนภาพ  $\mathbf{W}$  ที่มีจำนวนเซลล์เป็น  $N_y \times N_x$  จะมีบล็อก  $\mathbf{b}_{pq}$  สำหรับ  $p = 0, \dots, \lfloor \frac{N_y - n_y}{m_y} \rfloor$  และ  $q = 0, \dots, \lfloor \frac{N_x - n_x}{m_x} \rfloor$ . บล็อก  $\mathbf{b}_{pq}$  จะมีส่วนประกอบเป็นค่าเซลล์ Howard ของเซลล์ในขอบเขตของบล็อก. นั่นคือ  $\mathbf{b}_{pq} = \{\hat{v}_{ij}\}_{i,j \in \Omega_{block}}$  เมื่อเขต  $\Omega_{block}$  แทนเงื่อนไข  $p \cdot m_y \leq i < p \cdot m_y + n_y$  และ  $q \cdot m_x \leq j < q \cdot m_x + n_x$ . เวගเตอร์  $\hat{v}_{ij}$  คือค่าเซลล์ Howard หลังการทำอรวมอวาร์ลีซ นั่นคือ ส่วนประกอบที่  $k^{th}$  ของมัน  $\hat{v}_k(i, j) = (v_k(i, j) - v_{\min}(p, q)) / (v_{\max}(p, q) - v_{\min}(p, q))$  โดย  $v_k(i, j)$  คือเซลล์ Howard ที่  $k^{th}$  ของเซลล์  $(i, j)$  และ  $v_{\max}(p, q)$  กับ  $v_{\min}(p, q)$  คือค่าเซลล์ Howard ที่มากที่สุดกับน้อยที่สุดในบล็อก ตามลำดับ.

รูป 4.8 แสดงตัวอย่างขั้นตอนการทำบล็อก. ในตัวอย่าง บล็อกขนาด  $2 \times 2$  เซลล์ รวมผล Howard ของ 4 เซลล์ หรือเท่ากับ 36 ค่า Howard. หากส่วนภาพมีจำนวนเซลล์เป็น  $16 \times 8$  เซลล์ จะมีจำนวนบล็อกทั้งหมดเป็น  $(\lfloor \frac{16-2}{1} \rfloor + 1) \times (\lfloor \frac{8-2}{1} \rfloor + 1) = 105$  บล็อก. ดังนั้น สำหรับส่วนภาพขนาด  $128 \times 64$  ใช้เซลล์ขนาด  $8 \times 8$  ทำ Howard 9 ทิศทาง ใช้บล็อกขนาด  $2 \times 2$  และขนาดขัยบล็อก  $1 \times 1$  ลักษณะสำคัญของเอชโอลี จะมี  $105 \times 2 \times 2 \times 9 = 3780$  ค่า.

รูป 4.9 แสดงตัวอย่างของลักษณะสำคัญของเอชโอลี สำหรับส่วนภาพที่มีเป้าหมาย และส่วนภาพที่ไม่มีเป้าหมาย. ลักษณะสำคัญที่แปลงมาอาจจะดูยากด้วยตาเปล่า. การวัดผลที่เหมาะสมจึงมีความสำคัญมาก.



รูปที่ 4.8: ขั้นตอนการทำเอชโอลีอิก. บล็อกขนาด  $2 \times 2$  เซลล์ และใช้ขนาดขับเลื่อน  $1 \times 1$  เซลล์. แต่ละเซลล์ทำให้หาต 9 ทิศทาง. ภาพซ้าย แสดงส่วนภาพ โดยเส้นสีเขียวแสดงขอบเขตแบ่งเซลล์  $v_{0,0}$  ถึง  $v_{1,2}$  และเส้นสีแดงแสดงขอบเขตของบล็อก  $b_{0,1}$ . ภาพด้านมา (ซึ่งภาพ Cell[0,0] ถึง Cell[1,2]) แสดงค่าเซลล์หาตของเซลล์  $v_{0,0}$  ถึง  $v_{1,2}$ . ภาพขวาบน แสดงเซลล์ให้ตغاายในบล็อกก่อนที่จะคำนวณมอร์แลร์. ภาพขวาล่าง ค่าของบล็อก.



รูปที่ 4.9: ตัวอย่างลักษณะสำคัญเอชโอลี. สองภาพทางซ้าย แสดงตัวอย่างสำหรับส่วนภาพที่มีเป้าหมายอยู่. สองภาพทางขวา แสดงตัวอย่างสำหรับส่วนภาพที่ไม่มีเป้าหมายอยู่. ภาพแรกและสามจากซ้าย แสดงส่วนภาพ  $W$  ขนาด  $128 \times 64$  (เท่ากับ 8192 มิติ). ภาพสองและสี่จากซ้าย แสดงลักษณะสำคัญเอชโอลี  $x$  ขนาด 3780. มิติของลักษณะสำคัญเอชโอลีน้อยกว่ามิติของส่วนภาพมาก.

ข้อสังเกต การตรวจจับภาพวัตถุ ที่ใช้วิธีการหน้าต่างเลื่อนกับลักษณะสำคัญเอชโอลี มีการทำงานในลักษณะพื้นที่ย่อย. นั่นคือ หน้าต่างและขนาดขับเลื่อน ในวิธีหน้าต่างเลื่อน แบ่งจากการใหญ่เป็นส่วนภาพ. เซลล์ ในเอชโอลี แบ่งจากส่วนภาพเป็นเซลล์ แล้วใช้บล็อกกับขนาดขับเลื่อนบล็อก แบ่งจากส่วนภาพเป็นบล็อก โดยอาศัยค่าที่ได้จากเซลล์. ทั้งสามารถดับมีการทำงานในลักษณะคล้าย ๆ กัน ขั้นตอนหนึ่งต่อจากอีกขั้นตอนหนึ่ง. บทที่ 5 อภิปรายแนวคิดของการเรียนรู้เชิงลึก และโครงสร้างคอนโวลูชัน ที่ทำแนวคิดในลักษณะนี้ แต่ทำในลักษณะที่ทั่วไปและยืดหยุ่นขึ้น. ปัจจุบัน การเรียนรู้เชิงลึกและโครงสร้างคอนโวลูชัน เป็นศาสตร์และศิลป์ของการตรวจจับภาพวัตถุ และสามารถให้ผลการทำงานที่แม่นยำมาก.

**การจำแนกค่าทวิภาค.** จากภาพ  $F$  เลือกส่วนภาพต่าง ๆ  $W_{ij}$  ออกมาร่วมกับวิธีหน้าต่างเลื่อน. แต่ละส่วนภาพ  $W_{ij}$  จะถูกสกัดเป็นลักษณะสำคัญ  $x_{ij}$ . ตอนนี้จากลักษณะสำคัญ  $x_{ij}$  แบบจำลองจำแนกค่าทวิภาค สามารถนำมาใช้เพื่อทำนายผลว่าที่ตำแหน่ง  $(i, j)$  มีเป้าหมายอยู่หรือไม่. แบบจำลองจำแนกค่าทวิภาค มี

หลายชนิด. บทที่ 3 อภิปรายโครงข่ายประสาทเทียม. โครงข่ายประสาทเทียม ก็สามารถนำมาใช้ได้ แต่ ตัวอย่างนี้ นำเสนอแบบจำลองจำแนกค่าทวิภาค อีกชนิดที่ได้รับความนิยมมาก คือ ชัพพอร์ตเวกเตอร์แมชีน.

แบบจำลองจำแนกค่า (ทั้งการจำแนกค่าทวิภาค และการจำแนกกลุ่ม) มีหลายชนิด และอาจแบ่งเป็นแนวทางใหญ่ ๆ ได้สามแนวทาง. แนวทางแรก เรียกว่า แนวทางแบบจำลองแบ่งแยก (discriminative model). แนวทางนี้ เริ่มจากการสร้างแบบจำลองแบ่งแยก ที่ทำนายความน่าจะเป็นแบบมีเงื่อนไข ที่เอาร์พุตจะเป็น หนึ่ง สำหรับอินพุตที่ถูก นั่นคือ  $\Pr(y = 1 | \mathbf{x})$  หรือสำหรับการจำแนกกลุ่ม ความน่าจะเป็นแบบมีเงื่อนไข ของเอาร์พุตกลุ่มที่  $k^{th}$  สำหรับอินพุตที่ถูก นั่นคือ  $\Pr(y = k | \mathbf{x})$ . หลังจากนั้น ใช้ทฤษฎีการตัดสินใจ เช่น วิธีระดับค่าขีดแบ่ง เพื่อเลือกค่าทวิภาค หรือลักษณะของกลุ่ม สำหรับกรณีการจำแนกกลุ่ม. โครงข่ายประสาท เทียม (บท 3) สำหรับการจำแนกค่าทวิภาค หรือสำหรับการจำแนกกลุ่ม ก็จัดเป็นแบบจำลองแบ่งแยก.

อย่างไรก็ตาม การตีความเอาร์พุตของโครงข่ายประสาทเทียมในเชิงความน่าจะเป็น โดยเฉพาะกรณีจำแนก กลุ่มว่า  $\hat{y}_k \approx \Pr(y = k | \mathbf{x})$  มีข้อสงสัย ข้อสงสัย และประเด็นที่กำลังสำรวจและศึกษาไว้จัดอยู่[136].

แนวทางที่สอง เรียกว่า แนวทางแบบจำลองสร้างกำเนิด (generative model). แนวทางนี้ อาศัยความ น่าจะเป็นแบบมีเงื่อนไขของอินพุต สำหรับเอาร์พุตแต่ละแบบ นั่นคือ  $\Pr(\mathbf{x} | y)$  และความน่าจะเป็นก่อนของ เอาร์พุตแต่ละแบบ นั่นคือ  $\Pr(y)$  เพื่ออนุมานความน่าจะเป็นภายหลัง จากกฎของเบส. นั่นคือ

$$\Pr(y = k | \mathbf{x}) = \frac{\Pr(\mathbf{x} | y = k) \cdot \Pr(y = k)}{\Pr(\mathbf{x})} \quad (4.1)$$

โดย  $\Pr(\mathbf{x}) = \sum_k \Pr(\mathbf{x} | y = k) \cdot \Pr(y = k)$ . หมายเหตุ บางแบบจำลอง แม้อาศัยการอนุมานการแจกแจง ของอินพุต แต่อาจไม่ได้ประมาณ  $\Pr(\mathbf{x} | y)$  ออกมาโดยตรง เช่น โครงข่ายปรัปักษ์เชิงสร้างกำเนิด (Generative Adversarial Network[77] คำย่อ GAN) หรือ ตัวเข้าอัตรหัส (Autoencoder[111]).

แนวทางที่สาม เป็นการสร้างฟังก์ชันที่แปลงอินพุตไปเป็นเอาร์พุตโดยตรง นั่นคือ  $f : \mathbf{x} \mapsto y$  โดยไม่ได้ อาศัยความน่าจะเป็น หรือไม่สามารถตีในเชิงความเป็นความน่าจะเป็น. แนวทางนี้ เรียกว่า แนวทางฟังก์ชัน แบ่งแยก (discriminant function). ชัพพอร์ตเวกเตอร์แมชีน ก็จัดอยู่ในแนวทางนี้. หัวข้อ 4.2 อภิปราย ชัพพอร์ตเวกเตอร์แมชีน และทฤษฎีเบื้องหลัง.

## การจำจัดการระบุช้าช้อน

หลังจากขั้นตอนการจำแนกส่วนภาพที่มีเป้าหมายแล้ว ตำแหน่งของส่วนภาพที่ถูกจำแนกว่ามีเป้าหมาย จะ ถูกบันทึกเป็นค่าตำแหน่งที่ตราชพบ. ตำแหน่ง อาจบันทึกเป็นพิกัดของกล่องขอบเขต (bounding box) เช่น พิกัด  $(x, y)$  มุมขวาบนของกล่องขอบเขต กับพิกัดมุมล่างซ้าย หรืออาจจะเป็น พิกัดมุมขวาบนกับความกว้าง

และความสูง. กล่องขอบเขตสามารถนิยามเป็นขอบเขตของหน้าต่างที่เลือกส่วนภาพอุอกมา. ส่วนภาพในบริเวณใกล้เคียงกัน อาจถูกระบุว่ามีเป้าหมาย โดยที่เป้าหมายที่ส่วนภาพเหล่านั้นมี เป็นเป้าหมายเดียวกัน ซึ่งเป็นการระบุช้าช้อน. รูป 4.5 แสดงตัวอย่างส่วนภาพ ที่ได้จากการวิเคราะห์หน้าต่างเลื่อน. สังเกตว่า มีหลายส่วนภาพที่ครอบคลุมเป้าหมายเดียวกัน และอาจมีการระบุช้าช้อนเกิดขึ้น.

จากตำแหน่งของส่วนภาพต่าง ๆ ที่ช้าช้อน จะมีแค่ตำแหน่งเดียวที่จะเป็นตัวแทนของตำแหน่งของเป้าหมาย และที่เหลือจะถูกลบทิ้งไป. ขั้นตอนการกำจัดการระบุช้าช้อนนี้ จะเรียกว่า การกำจัดความช้าช้อน (redundancy removal). การกำจัดความช้าช้อน ดำเนินการตั้งแต่ตรวจสอบความช้าช้อน และกำจัดความช้าช้อนที่พบ. ผลลัพธ์คือ ตำแหน่งต่าง ๆ ของการตรวจจับ ที่ไม่ช้าช้อนกัน.

เพื่อตรวจสอบความช้าช้อน แนวทางที่นิยม คือ กำหนดค่าระดับขีด贲ง  $\tau$  และกล่องขอบเขตสองกล่อง จะถือว่าช้าช้อนกัน เมื่อ บริเวณช้อนทับกันมีค่าการซ้อนทับมากกว่า ค่าระดับขีด贲ง  $\tau$ . ค่าการซ้อนทับ มักถูกวัดด้วย ไอโอยู (IoU ซึ่งย่อมาจาก Intersect over Union) ซึ่งเป็นสัดส่วนพื้นที่ช้อนทับกันต่อพื้นที่รวม. นั่นคือ

$$\text{IoU} = \frac{A_1 \cap A_2}{A_1 \cup A_2} \quad (4.2)$$

เมื่อ  $A_1$  และ  $A_2$  คือพื้นที่ของกล่องขอบเขตสองกล่องที่พิจารณา.

สำหรับกล่องขอบเขตต่าง ๆ ที่ช้าช้อนกัน การกำจัดความช้าช้อน อาจทำโดยสูญเสียกล่องขอบเขตไว้กล่องหนึ่ง และตัดกล่องที่เหลือทิ้งก็ได้ แต่อาจทำให้คุณภาพโดยรวมของการตรวจจับด้อยลง. วิธีการระบุช้าช้อนค่าไม่มากสุดท้องถิ่น (non-local-maximum suppression[126]) จะระงับหรือตัดทิ้งกล่องขอบเขต ที่มีค่าความเหมะสมไม่มากที่สุด เมื่อเปรียบเทียบกับกล่องขอบเขตอื่น ๆ ที่อยู่รอบ ๆ กล่องนั้น. ค่าความเหมะสมของกล่องขอบเขต อาจได้มาจากแบบจำลองจำแนก เช่น กรณีโครงข่ายประสาทเทียม ค่าเออร์พุตของโครงข่าย (ก่อนผ่านการตัดสินใจด้วยวิธีระดับค่าขีด贲ง) ถูกตีความเป็นค่าความน่าจะเป็น และสามารถนำมาใช้เป็นค่าความเหมะสมได้. สำหรับกรณีเซ็พพอร์ตเวกเตอร์แมชชีน ค่าคะแนนตัดสินใจ (decision score สมการ 4.19) สามารถนำมาใช้ได้. สามารถรอบกล่องที่พิจารณา โดยที่ว่าไป มักหมายถึงกล่องที่อยู่ติดกัน แต่อย่างไรก็ตาม ความหวังของอาณาเขตนี้ สามารถกำหนดเป็นอภิมานพารามิเตอร์ได้.

## การนำเสนอผลด้วยแผนที่ความร้อน

แผนที่ความร้อน เป็นแผนภาพสี ที่ให้ข้อมูลความถี่เชิงพื้นที่. ความถี่ของตำแหน่งที่พบลูกค้าบ่อย ๆ อนุมานมาจากพิกัดตำแหน่งต่าง ๆ ที่ตรวจพบลูกค้า จากชุดลำดับภาพของวิดีโอ. เวลาที่ลูกค้าใช้ในแต่ละตำแหน่ง จะ

สะท้อนอุปกรณ์ความถี่ที่แสดงนี้.

จากพิกัดตำแหน่งที่ตรวจพบลูกค้า ซึ่งอาจเป็นพิกัดของกล่องขอบเขต จะถูกแปลงเป็นพิกัดตัวแทน ซึ่งอาจใช้จุดศูนย์กลางของกล่องขอบเขต. จากนั้น พิกัดตำแหน่งที่ตรวจพบลูกค้าในแต่ละภาพจะถูกนำมารวมกัน ซึ่งเปลี่ยนเป็น  $\mathbf{D} = \{\mathbf{d}_i\}$  สำหรับ  $i = 1, \dots, N$  เมื่อ  $\mathbf{d}_i = [x_i, y_i]^T$  เป็นพิกัดตำแหน่งที่พบลูกค้า ในแนวโน้มและแนวตั้งของภาพ ตามลำดับ. ด้วย  $i$  เป็นตัวชี้ของพิกัด และ  $N$  คือจำนวนพิกัดทั้งหมดที่ต้องการนำมาสรุปเป็นความถี่เชิงพื้นที่. หมายเหตุ ในทางปฏิบัติ การเลือกพิกัดตรวจพบมาสรุปนั้น อาจเลือกตามระยะเวลา เช่นภายในหนึ่งเดือนที่ผ่านมา หรือ อาจเลือกตามช่วงเวลาที่สนใจได้ เช่นแยกสรุประวัติวันธรรมดากับวันเสาร์อาทิตย์. แต่ ณ ที่นี่ แสดงตัวอย่าง  $i$  เป็นตัวชี้สำหรับพิกัดที่คัดเลือกมาแล้ว และ  $N$  เป็นจำนวนทั้งหมดที่ต้องการนำมาสรุปรวมกัน.

การสร้างแผนที่ความร้อน ก็คือ การแปลงข้อมูล  $\{\mathbf{d}_i\}$  ไปเป็นภาพสีขนาด  $H \times W$ . วิธีหนึ่งที่นิยมคือวิธีการประมาณความหนาแน่นแก่น (Kernel Density Estimation คำย่อ KDE). วิธีการประมาณความหนาแน่นแก่น เป็นการคำนวณการแจกแจงความน่าจะเป็นของข้อมูล และจัดเป็นแบบจำลองสร้างกำเนิดชนิดหนึ่ง อย่างไรก็ตาม วิธีการประมาณความหนาแน่นแก่นใช้การคำนวณมาก ดังนั้น วิธีการประมาณความหนาแน่นแก่นจึงมีการใช้งานค่อนข้างจำกัด ในทางปฏิบัติ. และข้อจำกัดนี้ จะเห็นได้ชัดมากขึ้น เมื่อจำนวนข้อมูลมีมากขึ้น หรือข้อมูลมีมิติมากขึ้น.

วิธีการประมาณความหนาแน่นแก่น ประมาณการแจกแจงความน่าจะเป็น ที่  $\mathbf{v} \in \mathbb{R}^M$  จากข้อมูล  $\mathbf{d}_i \in \mathbb{R}^M$  สำหรับ  $i = 1, \dots, N$  โดยคำนวณ

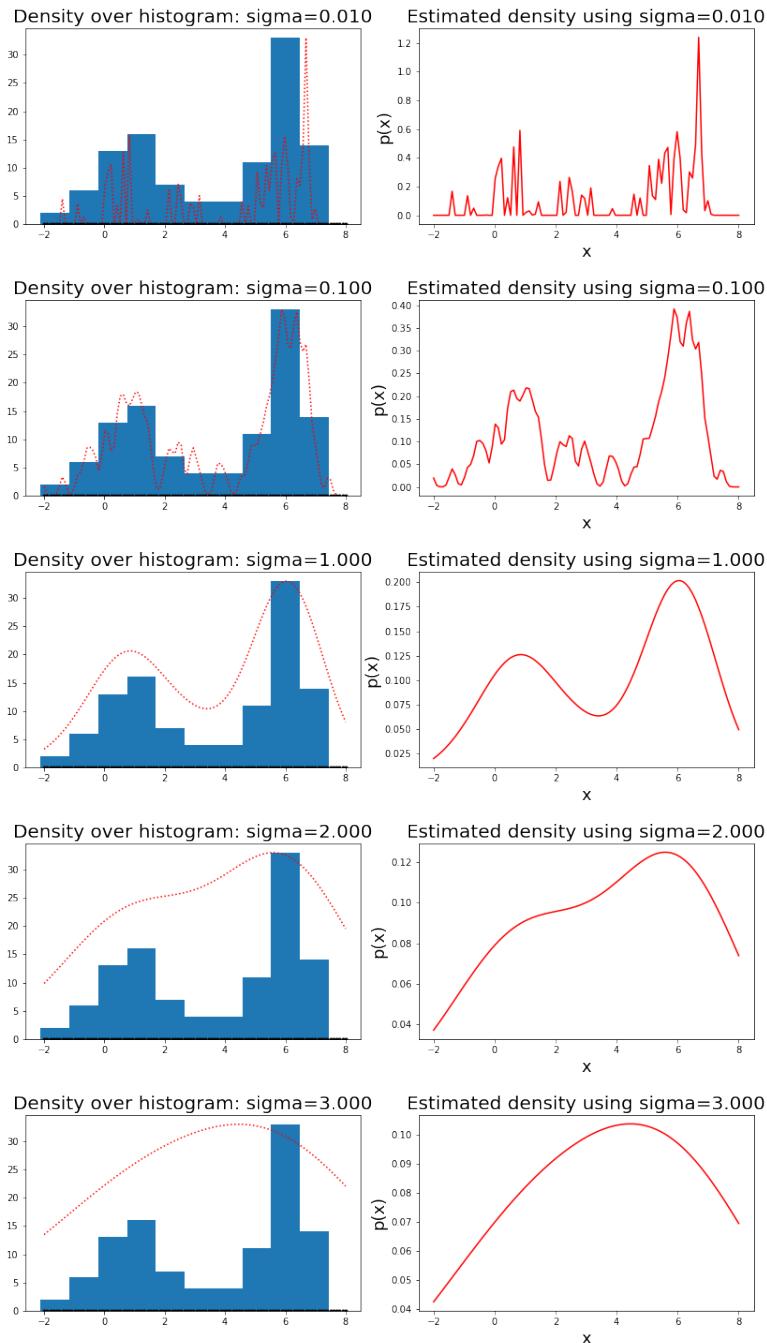
$$\hat{p}(\mathbf{v}) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot z(\mathbf{v}) \quad (4.3)$$

เมื่อ

$$z(\mathbf{v}) = \frac{1}{N} \sum_{i=1}^N \exp \left( -\frac{\|\mathbf{v} - \mathbf{d}_i\|^2}{2\sigma^2} \right) \quad (4.4)$$

และ  $\sigma$  เป็นอภิมานพารามิเตอร์ ซึ่งควบคุมความราบรื่นความต่อเนื่องของผลลัพธ์.

ตัวหารในสมการ 4.3 ทำเพื่อให้  $\hat{p}(\mathbf{v})$  มีคุณสมบัติความหนาแน่นความน่าจะเป็นที่ถูกต้อง. เพื่อการสร้างแผนที่ความร้อน การคำนวณเฉพาะค่า  $z$  ที่ตำแหน่งต่าง ๆ ก็เพียงพอ. นั่นคือ สำหรับภาพขนาด  $H \times W$  แผนที่ความร้อน  $\mathbf{Z} = [z([c, r]^T)]$  สำหรับ  $c = 0, \dots, W - 1$  และ  $r = 0, \dots, H - 1$ . จากนั้น  $\mathbf{Z}$  จะถูกนำไปคาดบนภาพสี โดยการแปลงค่า  $z$  ที่แต่ละพิกเซลเป็นสีตามแต่ระบบสีที่จะเลือกใช้.



รูปที่ 4.10: ผลการประมาณความหนาแน่นความน่าจะเป็น ด้วยวิธีการประมาณความหนาแน่นแก่น เมื่อใช้ค่า  $\sigma$  ต่าง ๆ ได้แก่ 0.01, 0.1, 1, 2, และ 3. ค่า  $\sigma$  ระบุไว้เหนือรูป. ภาพซ้ายแสดงความหนาแน่นที่ประมาณ ข้อมูลอยู่บนอิสโทแกรมของจุดข้อมูล โดยความหนาแน่นที่ประมาณถูกปรับขนาด เพื่อเปรียบเทียบกับอิสโทแกรมได้ชัดเจน. ภาพขวาแสดงความหนาแน่นที่ประมาณโดยแกนนอนแสดงค่าข้อมูล  $x$  และแกนตั้งแสดงค่าความหนาแน่นความน่าจะเป็น  $p(x)$  ที่ได้จากการประมาณ.

รูป 4.10 แสดงผลการประมาณความหนาแน่นความน่าจะเป็น ด้วยวิธีการประมาณความหนาแน่นแก่น เมื่อใช้ค่า  $\sigma$  ต่าง ๆ. ค่า  $\sigma$  อาจมองเหมือนเป็นการปรับความเรียบของความหนาแน่นที่จะประมาณ หรืออาจเปรียบเสมือนสมมติฐานเบื้องต้น เกี่ยวกับการความหนาแน่นของข้อมูล ว่าข้อมูลมีความหนาแน่นที่มีลักษณะเป็นการแจกแจงฐานนิยมเดียว (unimodal distribution) หรือแบบพหุฐาน (multimodal) และการแจกแจง มีความหลากหลาย มีความซับซ้อนมากขนาดไหน. ภาพในรูป แสดง ค่า  $\sigma$  ขนาดเล็กให้ผลการประมาณความหนาแน่นที่มีความซับซ้อนมาก มีจำนวนฐานมาก ฐานแคบ. ค่า  $\sigma$  ขนาดใหญ่ให้ผลการประมาณความหนาแน่นที่ซับซ้อนน้อย มีจำนวนฐานน้อยลง และความหนาแน่นมีการกระจายตัวออกไปมากขึ้น ฐานกว้าง.

## การประเมินผลการตรวจจับ

การประเมินผลเป็นหัวใจของงานการเรียนรู้ของเครื่อง เป็นหัวใจของงานวิศวกรรมและวิทยาศาสตร์ และเป็นหัวใจของ น่าจะเรียกได้ว่า ทุกภาระกิจ. การประเมินผลการตรวจจับวัตถุ อาจทำได้หลายวิธี. หนึ่งในวิธีที่นิยมคือ การประเมินด้วยค่าเฉลี่ยค่าประมาณความเที่ยงตรง.

ค่าเฉลี่ยค่าประมาณความเที่ยงตรง (mean Average Precision คำย่อ mAP) เป็นตัวชี้วัดที่นิยมใช้ประเมินคุณภาพระบบตรวจจับภาพวัตถุ. ค่าเฉลี่ยค่าประมาณความเที่ยงตรงสามารถใช้ประเมินความแม่นยำในการตรวจจับตำแหน่ง พร้อมกับประเมินความแม่นยำในการทายชนิดของวัตถุ โดยเป็นการประเมินความสามารถของระบบโดยรวม ไม่เฉพาะเจาะจงกับการเลือกระดับค่าชีดแบ่ง. (ดูแบบฝึกหัด 3.17 สำหรับตัวอย่างผลกระทบจากการเลือกระดับค่าชีดแบ่งที่ต่างกัน.) ค่าเฉลี่ยค่าประมาณความเที่ยงตรง สามารถคำนวณได้จาก

$$\text{mAP} = \frac{1}{K} \sum_{k \in \text{Classes}} \text{AP}_k \quad (4.5)$$

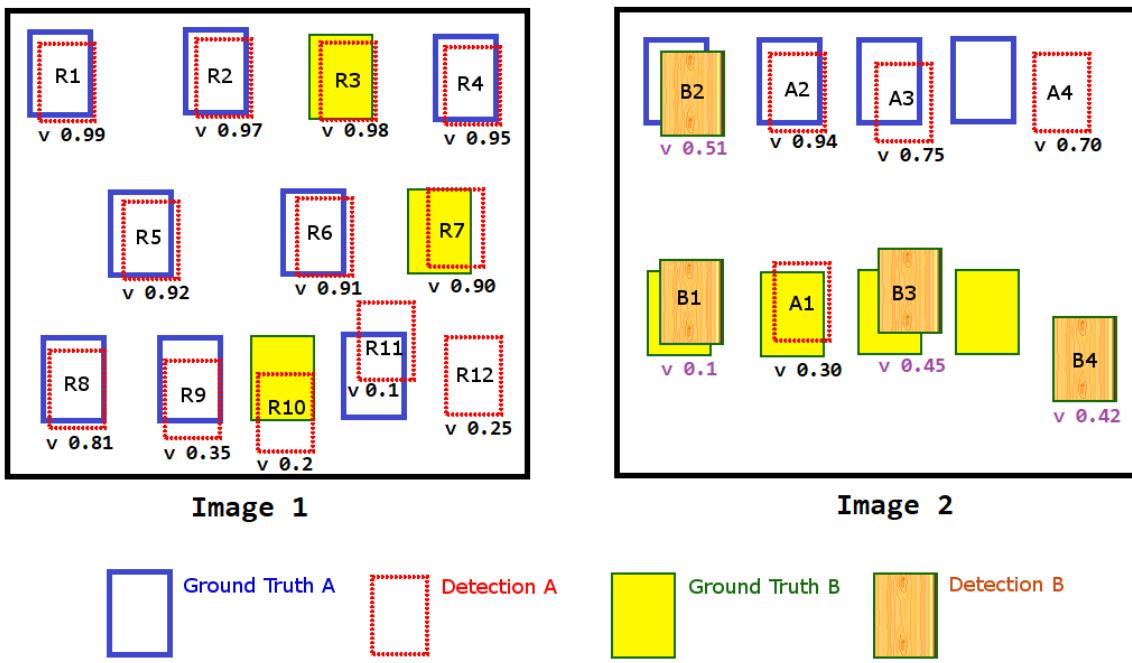
เมื่อ **Classes** คือเซตของกลุ่มต่าง ๆ ที่เกี่ยวข้อง. ค่า  $K$  คือจำนวนของชนิดกลุ่มที่เกี่ยวข้อง (ขนาดของเซต **Classes**). ค่า  $\text{AP}_k$  คือค่าประมาณความเที่ยงตรงของสำหรับการตรวจจับภาพของวัตถุชนิด  $k$ .

ค่าประมาณความเที่ยงตรงของวัตถุแต่ละชนิด  $\text{AP}_k$  สามารถประเมินได้จาก

$$\text{AP}_k = \sum_{j \in \text{Ranks}} p_{kj} \cdot \Delta r_{kj} \quad (4.6)$$

เมื่อ **Ranks** คือเซตลำดับของผลลัพธ์การตรวจพบวัตถุชนิด  $k$ . ค่า  $p_{kj}$  เป็นค่าความเที่ยงตรงสำหรับชนิด  $k$  ที่ลำดับ  $j$ . และ  $\Delta r_{kj} = r_{kj} - r_{k,j-1}$  โดย  $r_{kj}$  เป็นค่าการเรียงกลับสำหรับวัตถุชนิด  $k$  ที่ลำดับ  $j$ .

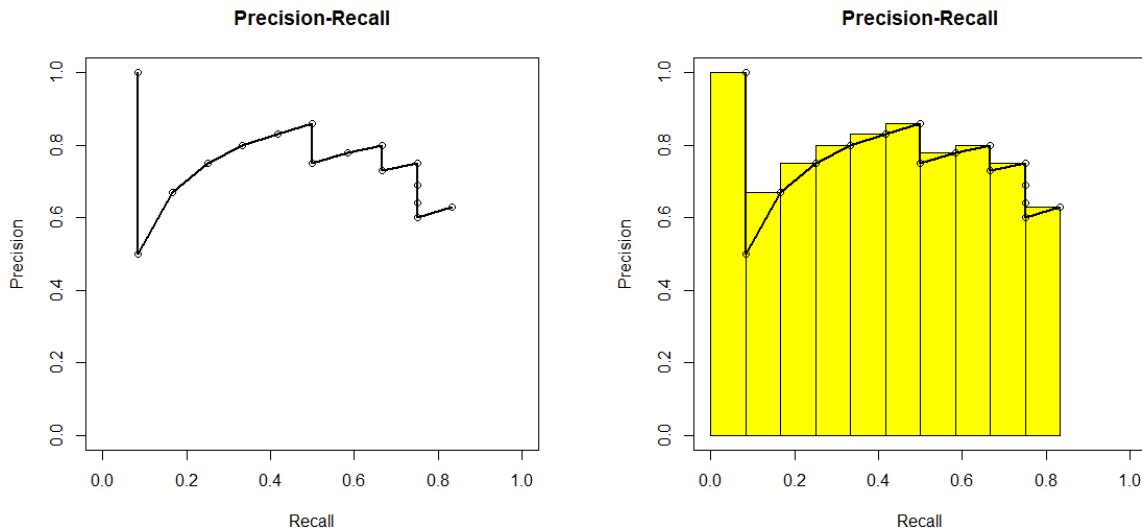
รูป 4.11 แสดงเฉลย และผลการตรวจจับวัตถุสองชนิด จากภาพสองภาพ. แต่ละการตรวจจับจะมีค่าการตรวจจับระบุอยู่ ( $\nu$  ในภาพ). มีหลายวิธีในการนำความแม่นยำในการตรวจจับตำแหน่ง เข้ามารวมด้วย. วิธีง่าย



**รูปที่ 4.11:** ตัวอย่างแสดงผลลัพธ์การตรวจจับและผลเฉลยของวัตถุนิด A และชนิด B สำหรับภาพ 2 ภาพ. ทุกกล่องขอบเขตของ การตรวจจับ จะมีค่าการตรวจจับ  $v$  และตัวเลขกำกับ. ค่าการตรวจจับ  $v$  จะนำไปใช้จัดลำดับการตรวจจับได้. ภาพ 1 (ทางซ้าย) มี วัตถุนิด A (กรอบเส้นทึบสีน้ำเงิน) อยู่ 8 วัตถุ. วัตถุนิด B (กรอบเส้นทึบสีเขียวภายในสีเหลือง) อยู่ 3 วัตถุ การตรวจจับให้ผลลัพธ์ ออกมาเป็น กล่องขอบเขต 12 ตำแหน่งสำหรับชนิด A (กรอบเส้นประสีแดง) และตรวจไม่พบวัตถุนิด B เลย (ไม่กล่องขอบเขตของ B ที่แทนด้วยกรอบลายไม้). ภาพ 2 (ทางขวา) มีวัตถุนิด A อยู่ 4 วัตถุ. วัตถุนิด B มีอยู่ 4 วัตถุ. การตรวจจับให้ผลลัพธ์ ออกมา เป็น กล่องขอบเขต 4 ตำแหน่งสำหรับชนิด A และ 4 ตำแหน่งสำหรับชนิด B. ผลลัพธ์จากการตรวจจับมีทั้ง (1) กรณีที่ตรวจจับได้ ถูกต้องทั้งตำแหน่งและชนิด ได้แก่ ชนิด A คือ R1, R2, R4, R5, R6, R8, R9, R11, A2, A3 และชนิด B คือ B1, B3. (2) กรณีตรวจ จับตำแหน่งได้ถูกต้องแต่ผิดชนิด ได้แก่ หาย A ให้ B คือ R3, R7, R10, A1 และหาย B ให้ A คือ B2. (3) กรณีผิดทั้งตำแหน่งและ ชนิด ได้แก่ R12, A4, B4. และ (4) กรณีตรวจไม่พบวัตถุ ทั้ง ฯ ที่มีวัตถุอยู่ ได้แก่ ภาพขวา (Image 2) วัตถุ A และบนตำแหน่งที่สอง จากขวา และวัตถุ B และล่างตำแหน่งสองจากขวา. นอกจากนี้ ตัวอย่างนี้ยังแสดงความแม่นยำในการระบุตำแหน่งที่แตกต่างกัน อีกด้วย.

๗ อาจใช้วิธีระดับค่าซีดแบง กับการวัดไอโอyu เพื่อตัดการตรวจจับที่ตำแหน่งคลาดเคลื่อนมากทั้ง. ตัวอย่าง เช่น หากกล่องขอบเขตของการตรวจจับ มีค่าไอโอyu กับกล่องขอบเขตเฉลย ต่ำกว่า 0.5 ถือว่าผิด นั่นคือเท่ากับ ตรวจจับเกินหนึ่ง สำหรับการตรวจจับโดย แล้วตรวจไม่พบหนึ่ง สำหรับเฉลยที่ไม่มีกล่องขอบเขตตรวจพบ. หรืออาจใช้ค่าไอโอyu เข้าไปคำนวนประกอบกับค่าความมั่นใจอื่น เพื่อสรุปอุปกรณ์เป็นค่าการตรวจจับก็ได้ ซึ่ง จะทำให้กล่องขอบเขตของการตรวจจับ ที่มีตำแหน่งคลาดเคลื่อนไปมาก จะได้คะแนนค่าการตรวจจับน้อย.

ตัวอย่าง การประเมินค่าเฉลี่ยค่าประมาณความเที่ยงตรง จากรูป 4.11 ค่าเฉลี่ยค่าประมาณความเที่ยง ตรงของการตรวจจับ สามารถหาได้จาก (1) เรียงลำดับการตรวจจับตามชนิด (2) ตรวจสอบผลลัพธ์จากการ ตรวจจับ เปรียบเทียบกับผลเฉลย (3) คำนวนหาค่าความเที่ยงตรง  $p_{kj}$  และค่าการเรียกกลับ  $r_{kj}$  (4) คำนวน พื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับ (Area under P-R curve) และ (5) คำนวนพื้นที่ใต้กราฟเฉลี่ย



รูปที่ 4.12: ภาพซ้าย กราฟความเที่ยงตรงและการเรียกกลับ ของการตรวจจับวัตถุนิด A (ดูตาราง 4.1 ประกอบ). ภาพขวา ค่า ประมาณพื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับ ของการตรวจจับวัตถุนิด A

ของทุก ๆ ชนิด ซึ่งคือค่าเฉลี่ยค่าประมาณความเที่ยงตรง mAP. ตาราง 4.1 แสดงตัวอย่างการคำนวณค่าเฉลี่ยค่าประมาณความเที่ยงตรง.

ผลลัพธ์การตรวจหาวัตถุกันสำหรับลำดับตามค่าการตรวจจับ. ค่าการตรวจจับนี้อาจเป็นค่าความน่าจะเป็นที่ระบบตรวจจับวัตถุให้ออกมา หรืออาจจะเป็นค่าอื่นในลักษณะคล้ายกัน เช่น ค่าความมั่นใจ[160] หรือค่าไอโอดิย (สมการ 4.2) หรือค่าความน่าจะเป็นคุณกับค่าไอโอดิย[161] ก็ได้. ในตัวอย่างนี้ ค่าการตรวจจับที่สูงหมายถึงการตรวจจับได้รับลำดับความสำคัญเป็นลำดับต้น ๆ. สังเกตว่า การจัดลำดับ ทำตามชนิดวัตถุที่ทำนาย เช่น ทำการตรวจจับวัตถุที่ถูกทำนายเป็นชนิด A จะถูกนิยามว่าจัดลำดับด้วยกัน ไม่ว่าจะเป็นการทำนายที่ภาพใด (หรือผลทำนายถูกหรือไม่ หรือว่าเฉลยจริงเป็นชนิดใด).

แต่ละผลลัพธ์จากการตรวจจับ จะถูกเปรียบเทียบกับผลเฉลย. ในตาราง 4.1 แสดงค่าความถูกต้อง จะระบุเป็น 1 หากมีผลเฉลยชนิดนั้นในตำแหน่งบริเวณนั้น และค่าความถูกต้อง จะระบุเป็น 0 หากไม่ใช่. ตัวอย่างเช่น ในรูป 4.11 ภาพซ้าย กล่องของเขต R3, R7, และ R10 ที่ tally ตำแหน่งของวัตถุนิด A แต่บริเวณนั้นไม่มีวัตถุชนิด A อยู่ (มีแต่วัตถุนิด B) หรือ กล่องของเขต R12 ที่ tally ตำแหน่งของวัตถุนิด A แต่บริเวณนั้นไม่มีวัตถุใดอยู่เลย ค่าความถูกต้องของกล่องของเขตเหล่านี้ จะเป็นศูนย์.

หากผลลัพธ์จากการตรวจจับถูกต้อง ค่าบวกจริง  $TP$  จะเพิ่มขึ้นหนึ่ง (เริ่มจากลำดับบนสุด) แต่หากผลลัพธ์จากการตรวจจับไม่ถูกต้อง ค่าบวกเท็จ  $FP$  จะเพิ่มขึ้นหนึ่ง (เริ่มจากลำดับบนสุด). ค่าความเที่ยงตรง  $p_{kj}$  ซึ่งเป็นอัตราส่วนการหายถูกต่อการหายทั้งหมด จะสามารถคำนวณได้จาก  $p_{kj} = TP / (TP + FP)$

ตารางที่ 4.1: ตัวอย่างการคำนวณค่าเฉลี่ยค่าประมาณความเที่ยงตรง

ผลลัพธ์	ค่าการตรวจจับ	ความถูกต้อง	ชนิด	<i>TP</i>	<i>FP</i>	$p_{kj}$	$r_{kj}$	$\Delta r_{kj}$	$p_{kj} \cdot \Delta r_{kj}$
R1	0.99	1	A	1	0	1.00	0.08	0.08	0.08
R3	0.98	0	A	1	1	0.50	0.08	0.00	0.00
R2	0.97	1	A	2	1	0.67	0.17	0.08	0.06
R4	0.95	1	A	3	1	0.75	0.25	0.08	0.06
A2	0.94	1	A	4	1	0.80	0.33	0.08	0.07
R5	0.92	1	A	5	1	0.83	0.42	0.08	0.07
R6	0.91	1	A	6	1	0.86	0.50	0.08	0.07
R7	0.90	0	A	6	2	0.75	0.50	0.00	0.00
R8	0.81	1	A	7	2	0.78	0.58	0.08	0.06
A3	0.75	1	A	8	2	0.80	0.67	0.08	0.07
A4	0.70	0	A	8	3	0.73	0.67	0.00	0.00
R9	0.35	1	A	9	3	0.75	0.75	0.08	0.06
A1	0.30	0	A	9	4	0.69	0.75	0.00	0.00
R12	0.25	0	A	9	5	0.64	0.75	0.00	0.00
R10	0.20	0	A	9	6	0.60	0.75	0.00	0.00
R11	0.10	1	A	10	6	0.63	0.83	0.08	0.05
$AP_A =$									<b>0.65</b>
B2	0.51	0	B	0	1	0	0	0	0
B3	0.45	1	B	1	1	0.50	0.14	0.14	0.07
B4	0.42	0	B	1	2	0.33	0.14	0.00	0.00
B1	0.10	1	B	2	2	0.50	0.29	0.14	0.07
$AP_B =$									<b>0.14</b>
ค่าเฉลี่ยค่าประมาณความเที่ยงตรง $mAP = \text{mean}_k AP_k = \frac{AP_A + AP_B}{2} \approx$									<b>0.40</b>

เมื่อ *TP* และ *FP* เป็นค่าบวกจริง และค่าบวกเท็จ สำหรับชนิดวัตถุ  $k$  ที่ลำดับ  $j$  เช่น สำหรับการทายวัตถุชนิด A ที่ลำดับแรกสุด นั่นคือ การทายกล่องของเขต R1 ทำให้ได้  $TP = 1, FP = 0$  และ  $p_{A1} = 1$ . แต่ ที่ลำดับที่สอง (เปรียบเสมือนการตั้งค่าขีดแบ่งอ่อนลง) นั่นคือ การทายกล่องของเขต R1 กับกล่องของเขต R3 ทำให้ได้  $TP = 1, FP = 1$  และ  $p_{A2} = 0.5$ . ค่าการเรียกกลับ  $r_{kj}$  เป็นอัตราส่วนการทายถูกต่อผลเฉลยทั้งหมด สำหรับชนิดวัตถุ  $k$  ที่ลำดับ  $j$  ซึ่งคำนวณได้จาก  $r_{kj} = TP/N_k$  เมื่อ  $N_k$  คือจำนวนผลเฉลยทั้งหมดของวัตถุชนิด  $k$ . ในที่นี้ (ดูรูป 4.11 ประกอบ) จำนวนผลเฉลยทั้งหมดชนิด A หรือ  $N_A = 12$  และ จำนวนผลเฉลยทั้งหมดชนิด B หรือ  $N_B = 7$ . ดังนั้น ที่ลำดับล่าง ๆ (เทียบเท่ากับ เมื่อทายมากขึ้น) ค่าการเรียกกลับ  $r_{kj}$  จึงมีแนวโน้มเพิ่มขึ้น เช่น สำหรับการทายวัตถุชนิด A ที่ลำดับที่หนึ่ง (เทียบเท่า การทาย R1 อันเดียว) ค่าการ

เรียกกลับ  $r_{A1} = 1/12 \approx 0.08$ . แต่ที่ลำดับที่สิบหก (ลำดับสุดท้าย เทียบเท่า การทายผลลัพธ์ทั้ง 16 อัน ออกไป) ซึ่งทายถูก 10 อัน ( $TP = 10$ ) ทำให้ได้ค่าการเรียกกลับ  $r_{A16} = 10/12 \approx 0.83$ .

ค่า  $p_{kj}$  และ  $r_{kj}$  ที่ได้สามารถนำมารวบรวมเป็นกราฟความเที่ยงตรงและการเรียกกลับ (Area under P-R curve) ได้. รูป 4.12 (ภาพซ้าย) แสดงกราฟความเที่ยงตรงและการเรียกกลับของการตรวจจับวัตถุชนิด A. ค่าประมาณความเที่ยงตรงเป็นการประมาณพื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับ. หากพื้นที่ใต้กราฟ มีขนาดใหญ่ หมายถึงคุณภาพที่ดีของตรวจจับภาพวัตถุ. การประมาณพื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับของการตรวจจับวัตถุชนิด A แสดงในรูป 4.12 (ภาพขวา).

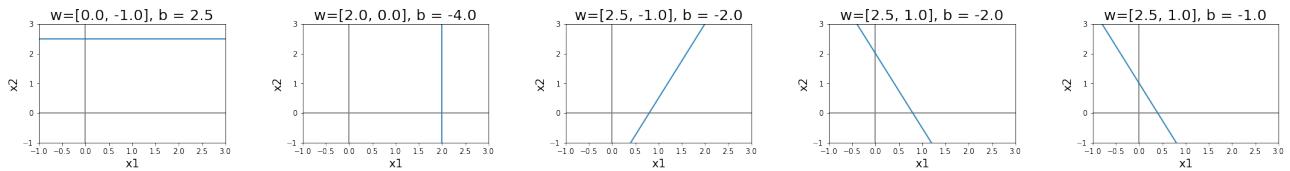
การประมาณพื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับ อาจใช้วิธีการคำนวณสี่เหลี่ยมคางหมูที่ช่วย ให้ได้พื้นที่ที่แม่นยำกว่าได้ แต่โดยทั่วไปแล้ว การประมาณคร่าว ๆ ด้วยสี่เหลี่ยมก็เพียงพอ. พื้นที่ใต้กราฟ หรือค่าประมาณความเที่ยงตรงของวัตถุแต่ละชนิด จะถูกนำมาเฉลี่ยกัน เพื่อคำนวณเป็นค่าเฉลี่ยค่าประมาณ ความเที่ยงตรง เช่นในตัวอย่างนี้  $mAP = 0.40$ .

## 4.2 ชั้พพอร์ตเวกเตอร์แมชชีน

ชัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine[43] 俗名 SVM) เป็นแบบจำลองจำแนกค่าทวิภาค ในแนวทางฟังก์ชันแบ่งแยก ซึ่งแปลงค่าอินพุต  $\mathbf{x} \in \mathbb{R}^D$  ไปเป็นเอาต์พุต  $\hat{y} \in \{-1, +1\}$  โดยตรง และ ไม่มีความเชื่อมโยงกับค่าความน่าจะเป็น. กลไกการทำงานของชัพพอร์ตเวกเตอร์แมชชีน อาศัยการแปลง ข้อมูลจากปริภูมิของข้อมูลต้นฉบับไปสู่ปริภูมิใหม่ ซึ่งอาจเรียกว่า **ปริภูมิลักษณะสำคัญ** (feature space) โดยปริภูมิลักษณะสำคัญนี้จะช่วยให้การแบ่งแยกข้อมูลออกเป็นกลุ่มได้ง่ายขึ้น และอาศัยอภิรະนาบในปริภูมิ ลักษณะสำคัญ เพื่อตัดแบ่งแยกข้อมูลออกเป็นสองกลุ่ม.

อภิรະนาบ (hyperplane) หมายถึง ระนาบในปริภูมิหลายมิติ. ในปริภูมิสองมิติ อภิรະนาบ จะหมายถึง เส้นตรง. นั่นคือ อภิรະนาบ จะสามารถระบุได้ด้วยสมการ  $w_1x_1 + w_2x_2 + b = 0$  เมื่อ  $\mathbf{x} = [x_1, x_2]^T$  เป็นตัวแปรในปริภูมิ และ  $\mathbf{w} = [w_1, w_2]^T$  กับ  $b$  เป็นค่าสัมประสิทธิ์. ในปริภูมิสามมิติ อภิรະนาบ จะหมายถึง ระนาบ (แผ่นตรงเรียบ ในสามมิติ). นั่นคือ อภิรະนาบ จะสามารถระบุได้ด้วยสมการ  $w_1x_1 + w_2x_2 + w_3x_3 + b = 0$  เมื่อ  $\mathbf{x} = [x_1, x_2, x_3]^T$  เป็นตัวแปรในปริภูมิ. ในปริภูมิ  $D$  มิติ อภิรະนาบ อาจจะยากที่จะ จินตนาการ แต่อภิรະนาบ ก็จะสามารถระบุได้ด้วยสมการ  $\mathbf{w}^T \mathbf{x} + b = 0$  เมื่อ  $\mathbf{w}, \mathbf{x} \in \mathbb{R}^D$ .

รูป 4.13 แสดงอภิรະนาบในสองมิติ. สังเกตความสัมพันธ์ระหว่างทิศทางและตำแหน่งของอภิรະนาบ กับ ค่าของ  $\mathbf{w}$  และ  $b$ . หากกำหนดให้  $\mathbf{x}_a$  และ  $\mathbf{x}_b$  เป็นจุดใด ๆ บนระนาบ. นั่นคือ  $\mathbf{w}^T \mathbf{x}_a + b = 0$  และ

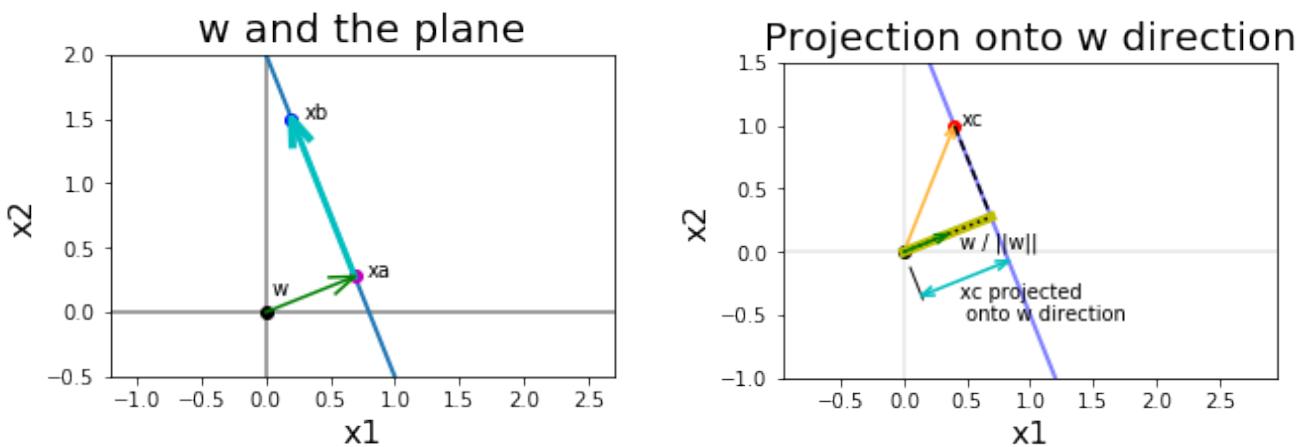


รูปที่ 4.13: อภิรนาบในสองมิติ เทียบเท่าเส้นตรง. แต่ละภาพแสดงอภิรนาบในสองมิติ เมื่อใช้ค่า  $w$  และ  $b$  ต่าง ๆ ซึ่งค่าระบุอยู่ด้านบนของภาพ. อภิรนาบแสดงด้วยเส้นทึบสีฟ้า.

$w^T \mathbf{x}_b + b = 0$  เวกเตอร์จาก  $\mathbf{x}_a$  ไป  $\mathbf{x}_b$  ซึ่งคือ  $\mathbf{x}_b - \mathbf{x}_a$  เป็นเวกเตอร์ในแนวของระนาบ. ดูรูป 4.14 ภาพซ้ายประกอบ. พิจารณาผลคุณเวกเตอร์ระหว่าง  $w$  กับเวกเตอร์ในแนวระนาบ และจากคุณสมบติของจุดบนระนาบ ทำให้พบว่า  $w^T \cdot (\mathbf{x}_b - \mathbf{x}_a) = w^T \mathbf{x}_b - w^T \mathbf{x}_a = w^T \mathbf{x}_b + b - w^T \mathbf{x}_a - b = 0$ . ผลคุณของ  $w$  กับเวกเตอร์ในแนวระนาบ จะเป็นศูนย์เสมอ. นั่นแปลว่า เวกเตอร์  $w$  ตั้งฉากกับแนวระนาบ. ดังนั้น ทิศทางของระนาบกำหนดด้วยค่าของเวกเตอร์  $w$ .

แต่ระนาบจะห่างจากจุดกำเนิดเท่าไร พิจารณารูป 4.14 ภาพขวา. จุดกำเนิด (origin) คือจุด  $[0, 0]^T$  ในปริภูมิสองมิติ แสดงด้วยจุดกลมสีดำในภาพ (ภาพขวา จุดนี้อาจถูกบังจากเวกเตอร์ต่าง ๆ). ระยะห่างระหว่างจุดกำเนิดกับระนาบ คือระยะจากจุดกำเนิดไประนาบในทิศทางที่ตั้งฉากกับระนาบ. นั่นคือ หากกำหนดให้  $\mathbf{x}_c$  เป็นจุดใด ๆ ในระนาบ ขนาดของภาพฉายของ  $\mathbf{x}_c$  ลงบนเวกเตอร์หนึ่งหน่วยในแนว  $w$  จะเป็นระยะทางจากจุดกำเนิดไปถึงระนาบ. ดังนั้น ระยะทางจากจุดกำเนิดไประนาบ สามารถคำนวณได้จาก  $d = \frac{w^T}{\|w\|} \cdot \mathbf{x}_c = \frac{w^T \mathbf{x}_c}{\|w\|}$  เมื่อ  $d$  คือระยะทางจากจุดกำเนิดไประนาบ. จากคุณสมบติของระนาบ  $w^T \mathbf{x}_c + b = 0$  ทำให้พบว่า  $d = \frac{-b}{\|w\|}$ . หมายเหตุ ขนาดของการฉายภาพเป็นลบ หมายถึงทิศทางของภาพที่ฉาย จะกลับทิศกับเวกเตอร์หนึ่งหน่วยที่เป็นเสมือนกระจก. ถ้า  $b < 0$  ทำให้ ระยะ  $d > 0$  ระนาบจะห่างจุดกำเนิดออกไปขนาด  $|d|$  ทางทิศ  $w$  ถ้า  $b > 0$  ทำให้ ระยะ  $d < 0$  ระนาบจะห่างจุดกำเนิดออกไปขนาด  $|d|$  ทางทิศตรงข้ามกับ  $w$  และถ้า  $b = 0$  หมายถึง ระนาบจะผ่านจุดกำเนิด.

การวางแผนปัญหาของชั้พพร์ตเวกเตอร์แมชชีน. ชั้พพร์ตเวกเตอร์แมชชีน ในปัจจุบันถูกประยุกต์ในงานหลากหลายทั้งงานการหาค่าคาดถอย และการจำแนกกลุ่ม แต่ตั้งเดิมเริ่มต้น ชั้พพร์ตเวกเตอร์แมชชีน ถูกออกแบบสำหรับการจำแนกค่าทิวภาค. ข้อมูลจะถูกแบ่งออกเป็น กลุ่มบวก และกลุ่มลบ. แนวคิดของชั้พพร์ตเวกเตอร์แมชชีน คือ การใช้อภิรนาบเป็นเส้น直เส้นแบ่งการตัดสินใจ และข้อมูลฝึกจะถูกนำมาใช้ เพื่อเลือกอภิรนาบในปริภูมิลักษณะสำคัญ ที่ทำให้ช่องว่างที่แบ่งระหว่างจุดข้อมูลกลุ่มบวกกับกลุ่มลบห่างกันมากที่สุด.

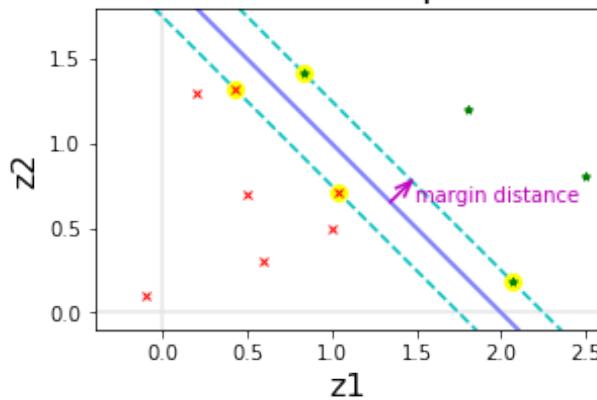


รูปที่ 4.14: ความสัมพันธ์ระหว่างพารามิเตอร์  $w$  และ  $b$  กับคุณลักษณะของอภิรนาบ. ภาพช้าย  $x_a$  และ  $x_b$  เป็นจุดใด ๆ บนรนาบ. รนาบแสดงด้วยเส้นสีฟ้า. เวกเตอร์จาก  $x_a$  ไป  $x_b$  แสดงด้วยลูกศรสีฟ้าเขียว. เวกเตอร์จากจุดกำเนิดไป  $w$  แสดงด้วยลูกศรสีเขียว. ภาพช้า  $x_c$  เป็นจุดใด ๆ บนรนาบ ซึ่งรนาบแสดงด้วยเส้นสีฟ้า. เวกเตอร์จากจุดกำเนิดไป  $w$  แสดงด้วยลูกศรสีฟ้าเขียว. ขนาดของเวกเตอร์  $x_c$  ที่ฉายลงบน  $w/\|w\|$  แสดงด้วยเส้นสีเหลือง. แต่บริเวณนั้นมีการซ้อนทับกันมาก อาจมองไม่ชัด เส้นสีฟ้าเขียวที่มีลูกศรส่องทาง ขยับออกมาระยะห่างของการฉายให้ชัดเจนขึ้น.

ซัพพอร์ตเวกเตอร์แมชีน อาศัยกลไกที่สำคัญสองอย่าง. กลไกสำคัญแรก คือ การแปลงจุดข้อมูลไปสู่ปริภูมิลักษณะสำคัญ. เนื่องจากอภิรนาบเป็นฟังก์ชันเชิงเส้น บทบาทของการแปลงข้อมูล จะช่วยในการแปลงข้อมูลไปสู่ปริภูมิที่ข้อมูลจะสามารถถูกแบ่งได้ด้วยอภิรนาบ. กำหนดให้ลักษณะสำคัญ  $z = \phi(x)$  เมื่อ  $x$  เป็นจุดข้อมูลในปริภูมิข้อมูลดังเดิม และ  $\phi : \mathbb{R}^D \mapsto \mathbb{R}^M$  เป็นฟังก์ชันที่ใช้แปลงข้อมูลไปสู่ปริภูมิลักษณะสำคัญ โดย  $D$  และ  $M$  คือจำนวนมิติของปริภูมิข้อมูลดังเดิมและของปริภูมิลักษณะสำคัญตามลำดับ. ดังนั้น จุดข้อมูล  $x$  (ในปริภูมิข้อมูลดังเดิม) จะถูกแทนด้วย  $z$  ในปริภูมิลักษณะสำคัญ. การใช้งานซัพพอร์ตเวกเตอร์แมชีนให้มีประสิทธิผล เกี่ยวพันโดยตรงกับการเลือกฟังก์ชันลักษณะสำคัญให้เหมาะสม ซึ่งจะได้อภิปรายรายละเอียดในหัวข้อ 4.2. หมายเหตุ การเลือกที่จะทำการแบ่งข้อมูลในปริภูมิดังเดิม สามารถทำได้ และในหลาย ๆ กรณี ก็เป็นทางเลือกที่เหมาะสม. การเลือกที่จะทำการแบ่งข้อมูลในปริภูมิดังเดิม กับเปรียบเสมือนการเลือกใช้ฟังก์ชันเอกลักษณ์เป็นฟังก์ชันลักษณะสำคัญ นั่นคือ  $\phi(x) = x$  และ  $z = x$ .

กลไกสำคัญที่สอง คือ การหาอภิรนาบที่ทำให้ขอบเขตของการแบ่งกว้างที่สุด. รูป 4.15 แสดงตัวอย่างความสัมพันธ์ระหว่าง จุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสำคัญ อภิรนาบทัดสินใจ และขอบเขตของการแบ่ง. ออกแบบเพื่อการแบ่งกลุ่มสองกลุ่มโดยเฉพาะ จุดประสงค์ของซัพพอร์ตเวกเตอร์แมชีน ไม่ใช่แค่หาอภิรนาบที่ดีที่แบ่งข้อมูลได้ แต่ต้องการหาอภิรนาบที่แบ่งข้อมูลได้ และแบ่งได้โดยมีขอบเขตของการแบ่ง (margin of separation) ที่กว้างที่สุดด้วย. สังเกตว่า ในรูป 4.15 หากเอียงอภิรนาบทัดสินใจเพิ่มขึ้นหรือลดลงเล็ก

### Datapoints and hyperplane in feature space



**รูปที่ 4.15:** จุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสำคัญ (จุดกาบาทสีแดงกลุ่มลบ และจุดดาวสีเขียวกลุ่มบวก) และอภิรະนาบทัดสินใจ (เส้นทึบสีน้ำเงิน). เส้นประสีฟ้าเจี้ยว แสดงขอบเขตของการแบ่ง. ระยะจากอภิรະนาบทึงขอบเขตของการแบ่ง แสดงในรูปด้วยลูกศร สีม่วง. จุดข้อมูลที่อยู่บนแนวขอบเขตของการแบ่ง เน้นด้วยสีเหลืองรอบ ๆ จุด.

น้อย ผลที่ได้ก็จะยังคงสามารถแบ่งข้อมูลได้สมบูรณ์ แต่ขอบเขตของการแบ่งจะแคบลง.

จุดข้อมูลที่อยู่บนแนวขอบเขตของการแบ่ง ซึ่งเป็นจุดข้อมูลที่อยู่ใกล้กับจุดข้อมูลจากต่างกลุ่มมากที่สุด เป็นจุดที่แบ่งยากที่สุด และอภิรະนาจะลูกกำหนดด้วยจุดข้อมูลเหล่านี้. จุดข้อมูลเหล่านี้จะเรียกว่า ชัพพร์ต เวกเตอร์ (support vectors) ซึ่งเป็นที่มาของชื่อ ชัพพร์ตเวกเตอร์แมชชีน. จุดข้อมูลอื่น ๆ ที่อยู่ลึกลงไปในเขตของกลุ่ม เป็นจุดข้อมูลที่แบ่งแยกได้ง่ายกว่า จะไม่มีบทบาทในการกำหนดอภิรະนา.

ปัญหาการจำแนกค่าทิวภาคในทางปฏิบัติ อาจจะมีขอบเขตของการแบ่ง ที่ซับซ้อนกว่าสถานการณ์ในรูป 4.15 ซึ่งสามารถแบ่งกลุ่มได้อย่างสมบูรณ์ด้วยอภิรະนา. ในที่นี่ พิจารณาการพัฒนาชัพพร์ตเวกเตอร์แมชชีน สำหรับกรณีที่ข้อมูลสามารถแบ่งแยกได้สมบูรณ์ก่อน และหัวข้อ 4.2 ยกไปรายการพัฒนาขยายความสามารถสำหรับกรณีทั่วไป (ซึ่งรวมถึงสถานการณ์ที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์).

**การหาอภิรະนา.** หากอภิรະนาบทัดสินใจ บรรยายด้วย  $\mathbf{w}^T \mathbf{z} + b = 0$  และมีข้อมูลฝึก  $\{\mathbf{x}_i, y_i\}_{i=1,\dots,N}$  ซึ่งเทียบเท่า  $\{\mathbf{z}_i, y_i\}$  โดย  $\mathbf{z}_i = \phi(\mathbf{x}_i)$  แล้ว อภิรະนาที่แบ่งแยกข้อมูลได้อย่างสมบูรณ์ คือ อภิรະนาที่มีค่าพารามิเตอร์  $\mathbf{w}$  กับ  $b$  ที่ทำให้

$$\begin{aligned} \mathbf{w}^T \mathbf{z}_i + b &> 0 & \text{เมื่อ } y_i = 1 \\ \mathbf{w}^T \mathbf{z}_i + b &< 0 & \text{เมื่อ } y_i = -1 \end{aligned} \quad (4.7)$$

สำหรับ  $i = 1, \dots, N$  โดย  $N$  เป็นจำนวนข้อมูลฝึก.

เพื่อความสะดวก กำหนดฟังก์ชันแบ่งแยก  $f$  เป็น

$$f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} + b. \quad (4.8)$$

ผลทายกลุ่ม หรือผลตัดสินใจ  $\hat{y}$  สามารถคำนวณได้จาก

$$\hat{y} = \begin{cases} 1 & \text{เมื่อ } f(\mathbf{z}) > 0, \\ -1 & \text{เมื่อ } f(\mathbf{z}) < 0. \end{cases} \quad (4.9)$$

นอกจากแบ่งแยกข้อมูลได้สมบูรณ์แล้ว เรายังต้องการให้ขอบเขตของการแบ่งกว้างที่สุดด้วย. พิจารณา  
ระยะจากอภิรนาบไปสู่จุดข้อมูลใด ๆ ดูรูป 4.16 ประกอบ. เมื่อแทนจุดใด ๆ ในปริภูมิด้วยเวกเตอร์จากจุด  
กำหนดไปจุดนั้น เวกเตอร์  $\mathbf{z}_i$  สามารถเขียนในรูปส่วนประกอบได้ว่า

$$\mathbf{z}_i = \mathbf{z}_p + r\vec{u}$$

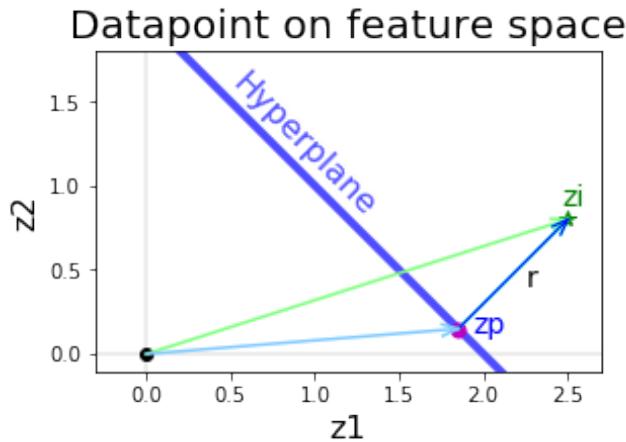
เมื่อ  $\mathbf{z}_p$  คือจุดภาพฉายเชิงตั้งจากจุด  $\mathbf{z}_i$  ลงบนอภิรนาบ และ  $\vec{u} = \mathbf{w}/\|\mathbf{w}\|$  คือเวกเตอร์หนึ่งหน่วย  
ในทิศทางตั้งจากกับอภิรนาบ และ  $r$  คือระยะห่างระหว่างจุด  $\mathbf{z}_i$  กับอภิรนาบ. ดังนั้น จุดใด ๆ  $\mathbf{z}_i =$   
 $\mathbf{z}_p + r\mathbf{w}/\|\mathbf{w}\|$  และค่าฟังก์ชันแบ่งแยก

$$\begin{aligned} f(\mathbf{z}_i) &= f(\mathbf{z}_p + r\mathbf{w}/\|\mathbf{w}\|) = \mathbf{w}^T \cdot (\mathbf{z}_p + r\mathbf{w}/\|\mathbf{w}\|) + b \\ &= \mathbf{w}^T \mathbf{z}_p + b + r\|\mathbf{w}\|^2/\|\mathbf{w}\| = r\|\mathbf{w}\| \\ r &= \frac{f(\mathbf{z}_i)}{\|\mathbf{w}\|}. \end{aligned} \quad (4.10)$$

นั่นคือ ระยะห่างระหว่างจุดใด ๆ  $\mathbf{z}_i$  กับอภิรนาบจะเท่ากับค่าฟังก์ชันแบ่งแยกหารด้วยขนาดของ  $\mathbf{w}$ . ถ้า  
 $f(\mathbf{z}_i) = 0$  ก็คือระยะห่าง  $r = 0$  จุดอยู่บนรูนาบ. ถ้า  $f(\mathbf{z}_i) > 0$  และจุดนั้นอยู่ห่างรูนาบทามคำนวณ  
ไปทางฝั่งกลุ่มบวก. ถ้า  $f(\mathbf{z}_i) < 0$  และจุดนั้นอยู่ห่างรูนาบไปทางฝั่งกลุ่มลบ. หมายเหตุ จุดกำหนด  $\mathbf{0}$  จะอยู่  
ห่างรูนาบ  $r = f(\mathbf{0})/\|\mathbf{w}\| = b/\|\mathbf{w}\|$ . สังเกตว่า ระยะ  $r$  คือระยะจากอภิรนาบไปจุดใด ๆ ซึ่งเมื่อพิจารณา  
ที่จุดกำหนด ระยะ  $r$  กับระยะจากจุดกำหนดไปรูนาบ  $d$  (ที่อภิปรายตอนต้นหัวข้อ) จะกลับทิศทางกัน.

พารามิเตอร์ของอภิรนาบที่ทำให้ขอบเขตของการแบ่งกว้างที่สุด อาจเขียนเป็นเงื่อนไขที่ว่าไปได้ว่า

$$\begin{aligned} \mathbf{w}^T \mathbf{z}_i + b &\geq +1 \quad \text{สำหรับ } y_i = +1 \\ \mathbf{w}^T \mathbf{z}_i + b &\leq -1 \quad \text{สำหรับ } y_i = -1 \end{aligned} \quad (4.11)$$



**รูปที่ 4.16:** ระยะไปสู่จุดข้อมูลใด ๆ จากอภิรานาบทันด้วย  $r$  ซึ่งคือขนาดของเวกเตอร์จากจุด  $\mathbf{z}_p$  ไป  $\mathbf{z}_i$ . จุดข้อมูลใด ๆ  $\mathbf{z}_i$  แสดงด้วยดาวสีเขียวเข้ม (บางส่วนถูกบัง). อภิรานาบท แสดงด้วยเส้นหนาสีน้ำเงิน. จุด  $\mathbf{z}_p$  (จุดถอนสีม่วง บางส่วนถูกบัง) คือจุดที่ถูกฉายจาก  $\mathbf{z}_i$  ลงบนระนาบ. เวกเตอร์จากจุดกำหนดไป  $\mathbf{z}_i$  (เวกเตอร์สีเขียว) เท่ากับเวกเตอร์จากจุดกำหนดไป  $\mathbf{z}_p$  (เวกเตอร์สีฟ้าอ่อน) บวกกับเวกเตอร์จาก  $\mathbf{z}_p$  ไป  $\mathbf{z}_i$  (เวกเตอร์สีฟ้าเข้ม).

ในสถานการณ์ที่ข้อมูลสามารถแบ่งแยกได้โดยสมบูรณ์ ค่าพารามิเตอร์  $\mathbf{w}$  และ  $b$  ที่ได้ สามารถนำไปปรับขนาดโดยคูณค่าคงที่เข้าไป เพื่อให้เงื่อนไขในสมการ 4.11 เป็นจริงได้. จุดข้อมูล  $i^{th}$  ที่ทำให้เงื่อนไขในสมการ 4.11 ทำงาน<sup>3</sup> จะอยู่บนแนวขอบเขตของการแบ่ง และจุดเหล่านี้จะเรียกว่า ชัพพร์ตเวกเตอร์. (ดูรูป 4.15 ประกอบ). ค่าพังก์ชันแบ่งแยกของชัพพร์ตเวกเตอร์ เป็น

$$f(\mathbf{z}'_i) = \mathbf{w}^T \mathbf{z}'_i + b = \begin{cases} +1 & \text{เมื่อ } y'_i = +1, \\ -1 & \text{เมื่อ } y'_i = -1. \end{cases}$$

โดย  $\mathbf{z}'_i$  และ  $y'_i$  คือค่าลักษณะสำคัญและเฉลยของชัพพร์ตเวกเตอร์ดั้งนี้  $i^{th}$ .

ระยะจากอภิรานาบทไปชัพพร์ตเวกเตอร์  $\mathbf{z}'_i$  จะเป็น

$$r = \frac{f(\mathbf{z}'_i)}{\|\mathbf{w}\|} = \begin{cases} \frac{+1}{\|\mathbf{w}\|} & \text{เมื่อ } y'_i = +1, \\ \frac{-1}{\|\mathbf{w}\|} & \text{เมื่อ } y'_i = -1. \end{cases}$$

ดังนั้นความกว้างของขอบเขตของการแบ่ง  $\rho = 2r = 2/\|\mathbf{w}\|$  และปัญหาค่ามากที่สุด  $\max_{\mathbf{w}, b} \rho = \frac{2}{\|\mathbf{w}\|}$  ก็เทียบเท่าปัญหาค่าน้อยที่สุด  $\min_{\mathbf{w}, b} \|\mathbf{w}\|$ . นอกจากนั้น สมการ 4.11 สามารถเขียนให้กระชับขึ้นได้เป็น

$$y_i \cdot (\mathbf{w}^T \mathbf{z}_i + b) \geq 1. \quad (4.12)$$

<sup>3</sup> ในทฤษฎีการหาค่าตี่ที่สุด เงื่อนไขของสมการหรือข้อจำกัดของสมการ (inequality constraint) เมื่อนำมาเขียนในรูป  $g(x) \geq 0$  จะเรียกว่า ทำงาน (active) ที่ค่า  $x_0$  ถ้า  $g(x_0) = 0$ . ตัวอย่างเช่น  $\mathbf{w}^T \mathbf{z}_i + b \geq 1$  ซึ่งเทียบเท่า  $\mathbf{w}^T \mathbf{z}_i + b - 1 \geq 0$  จะเรียกว่า ทำงานที่  $\mathbf{z}_0$  เมื่อ  $\mathbf{w}^T \mathbf{z}_i + b - 1 = 0$ .

นั่นคือ กรอบปัญหาการฝึกชั้พพอร์ตเวกเตอร์แมชีน สามารถเขียนได้เป็น

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{s.t. } & y_i(\mathbf{w}^T \mathbf{z}_i + b) \geq 1 \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.13)$$

ฟังก์ชันจุดประสงค์ของชั้พพอร์ตเวกเตอร์แมชีน เป็นค่อนเวกซ์ (convex function) และข้อจำกัดเป็นฟังก์ชันเชิงเส้น ซึ่งเหล่านี้ล้วนเป็นคุณสมบัติที่ดี จากมุ่งมองของการแก้ปัญหาค่าตัวที่สุด (หัวข้อ 2.3) เพราะว่า ลักษณะเหล่านี้ ทำให้ เมื่อแก้ปัญหาและพบค่าทำให้น้อยที่สุดท้องถิ่นแล้ว ค่าทำให้น้อยที่สุดท้องถิ่นจะเป็น ค่าทำให้น้อยที่สุดทั่วหมดด้วย.

การใช้งานชั้พพอร์ตเวกเตอร์แมชีนในทางปฏิบัติจะไม่แก้ปัญหานี้โดยตรง แต่จะใช้คุณสมบัติของภาวะคู่กัน (ดูแบบฝึกหัด 2.28 เพิ่มเติม) เพื่อแปลงปัญหานิพจน์ 4.13 ซึ่งเป็นปัญหาปัจมุข ไปอยู่ในรูปปัญหาคู่ ซึ่งจะสามารถใช้งานได้มีประสิทธิภาพกว่า.

**ปัญหาคู่.** จากวิธีการนับ [39] จุดประสงค์และข้อจำกัดที่ระบุด้วยนิพจน์ 4.13 จะแทนด้วยลากرانจ์ฟังก์ชัน

$$J(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i \cdot (y_i \cdot (\mathbf{w}^T \mathbf{z}_i + b) - 1) \quad (4.14)$$

เมื่อ  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T$  เป็นลากرانจ์พารามิเตอร์ โดย  $\alpha_i \geq 0$  สำหรับ  $i = 1, \dots, N$ .

จากทฤษฎีบทการคุณทักษะ (ดูแบบฝึกหัด 2.26) ที่กล่าวว่า หากกำหนดให้  $\mathbf{w}_o$  และ  $b_o$  แทนชุดค่าพารามิเตอร์ที่ดีที่สุด (ค่าทำให้น้อยที่สุด) และ ณ จุดที่ดีที่สุด เงื่อนไขต่อไปนี้จะต้องเป็นจริง. เงื่อนไขที่หนึ่ง คือ  $\alpha_i \geq 0$  สำหรับทุก ๆ ค่าของ  $i$ . เงื่อนไขที่สอง คือ

$$\nabla_{\mathbf{w}} J(\mathbf{w}_o, b_o, \boldsymbol{\alpha}) = 0$$

$$\nabla_b J(\mathbf{w}_o, b_o, \boldsymbol{\alpha}) = 0$$

ซึ่งเมื่อหาอนุพันธ์และแก้สมการแล้วจะได้ว่า

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_i y_i \mathbf{z}_i \quad (4.15)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (4.16)$$

และเงื่อนไขที่สาม คือ

$$\sum_{i=1}^N \alpha_i \cdot (y_i(\mathbf{w}_o^T \mathbf{z}_i + b_o) - 1) = 0.$$

ดังนั้น เมื่อพิจารณาเงื่อนไขที่หนึ่งกับเงื่อนไขที่สามแล้วจะพบว่า ณ จุดที่ดีที่สุด ถ้า  $\alpha_i > 0$  และ เราไว้แน่ ๆ เลยกว่า  $y_i(\mathbf{w}_o^T \mathbf{z}_i + b_o) - 1 = 0$ . นั่นคือข้อจำกัดทำงาน ซึ่งหมายถึง จุดข้อมูลที่  $i^{th}$  เป็นชัพพร์ตเวกเตอร์. กำหนดให้  $J'$  แทนลักษณะฟังก์ชัน เมื่อใช้ชุดค่าพารามิเตอร์ที่ดีที่สุด (ใช้  $\mathbf{w}_o$  และ  $b_o$ ) พร้อมแทนค่า จากสมการ 4.15 และ 4.16 จะได้

$$J'(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{z}_i^T \mathbf{z}_j$$

เมื่อ  $\alpha \geq 0$  และ  $\sum_{i=1}^N \alpha_i y_i = 0$ . กำหนดให้ ฟังก์ชันเครอร์เนล  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x})_i^T \phi(\mathbf{x})_j = \mathbf{z}_i^T \mathbf{z}_j$ .

ดังนั้นปัญหาคู่สามารถระบุได้เป็น

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \\ & \alpha \geq 0 \quad \text{for } i = 1, \dots, N. \end{aligned} \tag{4.17}$$

สังเกตว่า (1) ปัญหาปัญมเป็นปัญหาค่าน้อยที่สุด แต่ปัญหาคู่เป็นปัญหาค่ามากที่สุด<sup>4</sup>. (2) ปัญหาคู่อยู่ในรูปของตัวแปร  $\boldsymbol{\alpha}$  เท่านั้น ไม่มี  $\mathbf{w}$  ไม่มี  $b$ .

หาก  $\boldsymbol{\alpha}^*$  เป็นลักษณะพารามิเตอร์ที่ดีที่สุดที่หาได้มา แล้วการทำนายกลุ่มของจุดข้อมูล  $\mathbf{x}$  สามารถคำนวณได้โดยค่าฟังก์ชันแบ่งแยก  $f(\phi(\mathbf{x})) = \mathbf{w}_o^T \phi(\mathbf{x}) + b_o$ . เพื่อความสะดวกนิยาม  $g(\mathbf{x}) = \mathbf{w}_o^T \phi(\mathbf{x}) + b_o$ . เมื่อแทนค่าสมการ 4.15 เข้าไปแล้วจะได้

$$\begin{aligned} g(\mathbf{x}) &= \sum_{i=1}^N \alpha_i^* y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b_o \\ &= \sum_{i=1}^N \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}) + b_o \end{aligned} \tag{4.18}$$

เมื่อ  $\mathbf{x}_i, y_i$  คือจุดข้อมูลฝึก.

<sup>4</sup>พิจารณาฟังก์ชันจุดประสงค์ของปัญหาคู่ จะเห็นว่าฟังก์ชันจุดประสงค์ของปัญหาคู่ ได้รับอิทธิพลส่วนหนึ่งมาจากข้อจำกัดในปัญหาปัญม. ปัญหาปัญม ต้องการหาค่าทำน้อยที่สุด ภายใต้ข้อจำกัด. ปัญหาคู่ ต้องการหาค่าทำมากที่สุด เพื่อจะรักษาข้อจำกัดปัญมไว้โดยไม่ทำร้ายจุดประสงค์ปัญม.

เนื่องจากตัวแปร  $\alpha_i^*$  กำหนดให้ของลักษณะพารามิเตอร์ ดังนั้น สำหรับข้อจำกัดที่ไม่ได้ทำงาน ซึ่งสัมพันธ์กับจุดข้อมูลที่อยู่ลึกลงไปในกลุ่ม ไม่ได้อยู่บริเวณขอบเขตของการแบ่ง ไม่ใช่ชัพพอร์ตเวกเตอร์ ค่า  $\alpha_i^*$  ของจุดข้อมูลเหล่านั้นจะเป็นศูนย์ (เงื่อนไขที่สามและที่หนึ่งของครูชคุนทั้กเกอร์). สำหรับ  $\alpha_i^* = 0$  ไม่ได้ส่งผลต่อการคำนวณสมการ 4.18 เลย. ดังนั้น การใช้ชัพพอร์ตเวกเตอร์แมชีนอนุมานกลุ่ม จึงไม่จำเป็นต้องใช้ข้อมูลทุกตัว ใช้เฉพาะชัพพอร์ตเวกเตอร์ก็พอ. นั่นคือ ค่าการอนุมานกลุ่ม คำนวณจาก

$$g(\mathbf{x}) = \sum_{i \in S} \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}) + b_o \quad (4.19)$$

เมื่อ  $S$  คือ เซตของตัวนี้ของชัพพอร์ตเวกเตอร์ นั่นคือ  $S = \{i : \alpha_i^* > 0\}$ .

สำหรับค่าของพารามิเตอร์  $b_o$  พิจารณาจากชัพพอร์ตเวกเตอร์ ( $i \in S$ ) ที่มี  $\alpha_i^* > 0$ . จากทฤษฎีบทของครูชคุนทั้กเกอร์ทำให้รู้ว่า ถ้า  $\alpha_i^* > 0$  หมายถึง เงื่อนไข  $y_i(\mathbf{w}_o^T \mathbf{z}_i + b_o) \geq 0$  ทำงาน. นั่นคือ สำหรับ  $i \in S$  และ

$$y_i \cdot g(\mathbf{x}_i) = 1 \quad (4.20)$$

$$y_i \cdot \left( \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) + b_o \right) = 1. \quad (4.21)$$

เมื่อคุณ  $y_i$  เข้าไปทั้งสองข้าง (ซึ่ง  $y_i^2 = 1$ ) และจัดรูปใหม่จะได้

$$b_o = y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i)$$

การคำนวณ อาจสุ่มเลือกดัชนี้  $i$  ของชัพพอร์ตเวกเตอร์ขึ้นมาหนึ่งตัว หรือที่นิยม[83] คือการใช้ค่าเฉลี่ย

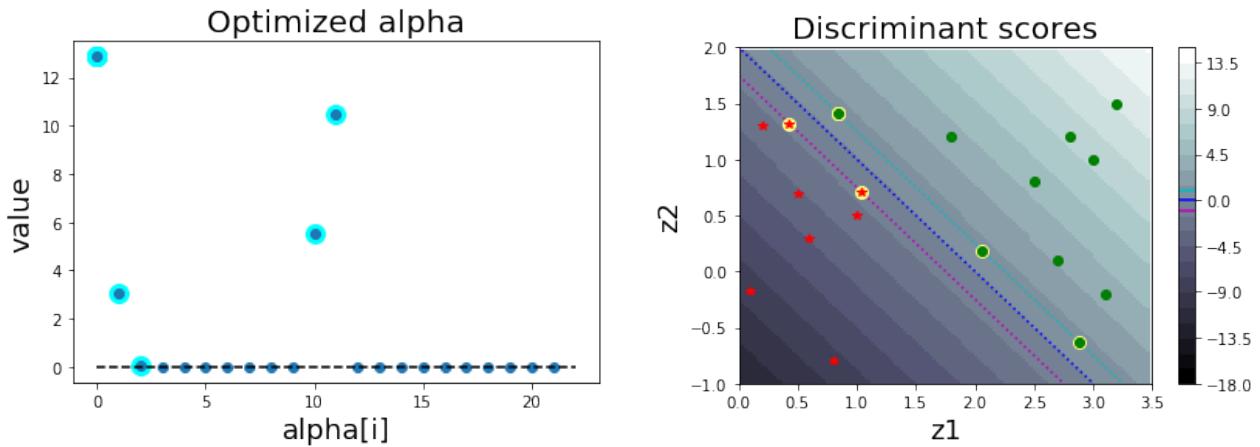
$$b_o = \frac{1}{|S|} \sum_{i \in S} \left( y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) \right) \quad (4.22)$$

เมื่อ  $|S|$  คือจำนวนของชัพพอร์ตเวกเตอร์.

ตัวอย่างผลลัพธ์จากการฝึกชัพพอร์ตเวกเตอร์แมชีน แสดงในรูป 4.17. ภาพซ้ายเป็นค่า  $\alpha$  ที่ได้จากการฝึก. ภาพขวาแสดงค่าฟังก์ชันแบ่งแยก ที่คำนวณจากสมการ 4.19. ชัพพอร์ตเวกเตอร์ คือจุดข้อมูลที่เน้น ดังแสดงในภาพขวา ซึ่งระบุได้จากค่า  $\alpha_i$  ที่สัมพันธ์กับมันมีค่ามากกว่าศูนย์.

## สถานการณ์ที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์

ในทางปฏิบัติ การแยกของกลุ่มข้อมูลอาจทำให้บริเวณของข้อมูลมีการซ้อนทับกันได้. สมมติฐานการแบ่งแยกได้อย่างสมบูรณ์ อาจจะทำให้ได้แบบจำลองที่การโอเวอร์พิท ขาดคุณสมบัติความทั่วไป. ดังนั้น เพื่อ



รูปที่ 4.17: ผลลัพธ์ของชัพพร์ตเวกเตอร์แมชชีน. ภาพซ้าย แสดงค่า  $\alpha_i$  ที่ได้จากการฝึก.  $\alpha_i > 0$  เน้นด้วยสีฟ้าเขียวรอบ ๆ. เส้นประสีดำ แสดงแนวของค่าศูนย์. ภาพขวา แสดงค่าฟังก์ชันแบ่งแยกของชัพพร์ตเวกเตอร์แมชชีนในปริภูมิลักษณะสามตัวแปร. ค่าฟังก์ชันแบ่งแยกเป็นค่าต่อเนื่อง แต่ในภาพค่าฟังก์ชันแบ่งแยกแสดงด้วยระดับสีเทา 18 ระดับ ซึ่งค่าระดับด้วยและสีด้านข้าง. เส้นประสีขาว สีฟ้าเขียว และสีฟ้าเขียว แสดงแนวที่ค่าฟังก์ชันแบ่งแยกเป็น  $-1, 0$ , และ  $1$  ตามลำดับ. จุดข้อมูลฝึก แสดงด้วยวงกลมสีเขียว (กลุ่มขวา) และดาวสีแดง (กลุ่มลับ). จุดข้อมูลฝึกที่ถูกเลือกเป็นชัพพร์ตเวกเตอร์ เน้นด้วยสีเหลืองรอบ ๆ.

ผ่อนสมมติฐานการแบ่งแยกได้อย่างสมบูรณ์ ควรจะยอมให้มีบางจุดข้อมูลที่อาจล้าเข้าไปในขอบเขตของการแบ่งบ้าง หรือแม้แต่ยอมให้มีบางจุดข้อมูลที่ถูกจำแนกกลุ่มผิดบ้าง.

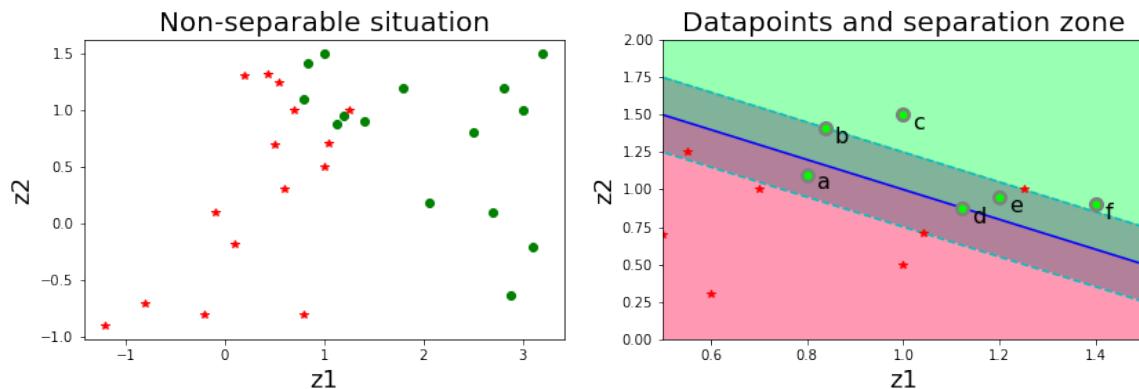
ชัพพร์ตเวกเตอร์แมชชีนผ่อนปรนสมมติฐานการแบ่งแยกสมบูรณ์ลง ด้วยการผ่อนปรนข้อจำกัดของขอบเขตของการแบ่ง (อสมการ 4.11) ผ่านกลไกของตัวแปรช่วย  $\xi_i$  เป็น

$$\begin{aligned} \mathbf{w}^T \mathbf{z}_i + b &\geq +1 - \xi_i && \text{สำหรับ } y_i = +1 \\ \mathbf{w}^T \mathbf{z}_i + b &\leq -1 + \xi_i && \text{สำหรับ } y_i = -1 \end{aligned} \quad (4.23)$$

โดย  $\xi_i \geq 0$  สำหรับ  $i = 1, \dots, N$  เมื่อ  $N$  เป็นจำนวนข้อมูลฝึก.

ตั้งนั้น ถ้า  $\xi_i = 0$  หมายถึง จุดข้อมูลที่  $i^{th}$  จะอยู่ห่างอภิรนาบอกมาทางกลุ่มที่ถูกต้อง และอยู่นอกขอบเขตของการแบ่ง. รูป 4.18 แสดงตัวอย่างต่าง ๆ เมื่อผ่อนปรนเงื่อนไขแบ่งแยกสมบูรณ์ลง. โดยในรูป จุด b, จุด c, และจุด f จะมี  $\xi_b, \xi_c, \xi_f = 0$ . ถ้า  $\xi_i > 0$  หมายถึง จุดข้อมูลที่  $i^{th}$  อยู่ล้ำแนวของขอบเขตของการแบ่งออกไป. ในรูป จุด a, จุด d, และจุด e จะมี  $\xi_a, \xi_d, \xi_e > 0$ .

พิจารณากรณีที่  $0 < \xi_i < 1$  นั่นคือ จุดข้อมูลอยู่ล้ำแนวของขอบเขตของการแบ่งออกไป แต่ยังไม่ถึงอภิรนาบ เช่น จุด e ในรูป จะมี  $0 < \xi_e < 1$ . และเนื่องจากจุดข้อมูลยังอยู่ผ่านของกลุ่มอยู่ จุดข้อมูลที่มี  $0 < \xi_i < 1$  จะยังถูกจำแนกได้ถูกต้อง. กรณีที่  $\xi_i = 1$  นั่นคือ จุดข้อมูลอยู่ล้ำแนวของขอบเขตของการแบ่งออกไป และไปอยู่บนอภิรนาบพอดี เช่น ในรูป  $\xi_d = 1$ . กรณีที่  $\xi_i > 1$  นั่นคือ จุดข้อมูลอยู่ล้ำแนว



รูปที่ 4.18: ข้อมูลที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์. ภาพซ้าย แสดงจุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสามัญ. จุดข้อมูลไม่สามารถถูกแบ่งแยกกลุ่มได้อย่างสมบูรณ์ด้วยอัตราภิรະนาบ. วงกลมสีเขียว แทนจุดข้อมูลของกลุ่มบวก. ดาวสีแดง แทนจุดข้อมูลของกลุ่มลบ. ภาพขวา แสดงจุดข้อมูลกับบริเวณของการแบ่งต่าง ๆ. เส้นทึบสีน้ำเงิน แทนอัตราภิรະนาบ. เส้นประสีฟ้าเขียว แทนแนวของขอบเขตของการแบ่ง. บริเวณพื้นหลังสีเขียวอ่อน แทนบริเวณของกลุ่มบวก. บริเวณพื้นหลังสีชมพูอ่อน แทนบริเวณของกลุ่มลบ. บริเวณพื้นหลังสีม่วง แทนบริเวณที่อยู่ภายในขอบเขตของการแบ่งกลุ่มบวก. บริเวณพื้นหลังสีฟ้าอ่อน แทนบริเวณของกลุ่มลบ. จุดข้อมูล a เป็นจุดข้อมูลกลุ่มบวก ที่ตัดแนวไปอยู่ในฝั่งของกลุ่มลบ จุดนี้จะถูกจำแนกผิดเป็นกลุ่มลบ. จุดข้อมูล b อยู่พอดีบนแนวของขอบเขตของการแบ่งฝั่งกลุ่มบวก. จุดข้อมูล c และ f อยู่ลึกลงไปในบริเวณของกลุ่มบวก. จุดข้อมูล d อยู่พอดีบนอัตราภิรະนาบแบ่ง. จุดข้อมูล e อยู่ล้ำแนวของกາມຈາກເຂົ້າໄປອູ່ໃນຂອບເຂດຂອງການແບ່ງ ແຕ່ຍຸ້ນຢູ່ໃນຝ່າຍຂອງການແບ່ງ.

ของขอบเขตของการแบ่งออกໄປมาก มากจนเลยแนวของอัตราภิรະนาบ ข้ามໄປອູ່ອຶກຝ່າຍຂອງການຈຳແນກ ດັ່ງນັ້ນ ຈຸດข้อมูลจะถูกຈຳແນກຜິດ ເຊັ່ນໃນຮູບ  $\xi_a > 1$ .

ເຈື່ອນໄວໃນສາມາດ [4.23](#) ເຂົ້າມີເກຣະບັນຫຼັບເປັນ

$$y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq 1 - \xi_i \quad \text{ສໍາທັບ } i = 1, \dots, N \quad (4.24)$$

ເມື່ອ  $\xi_i \geq 0$ . ການປັບເຈື່ອນໄວນີ້ ເປີຍບເສີມວິນການປັບຈາກຂ້ອຈຳກັດທີ່ເຂັ້ມຈວດ ຜ່ອນປຽນລົງມາເປັນຂ້ອຈຳກັດທີ່ອ່ອນລົງ. ກຽບປໍ່າພາຫາ ຈຶ່ງຖືກວາງໃໝ່ເປັນ

$$\begin{aligned} \underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \text{for } i = 1, \dots, N \end{aligned} \quad (4.25)$$

ເມື່ອ  $C > 0$ . ອົກມານພາຣາມີເຕອ້ວ  $C$  ເປັນເໜືອນຄ່າທີ່ໃຊ້ກວບຄຸມຄວາມເຂັ້ມຈວດຂອງຂ້ອຈຳກັດ. ຄ່າ  $C$  ທີ່ເລີກຈະຍອມໃຫ້ການຈຳແນກຜິດໄດ້ມາກຈີ້ນ ໃນຂະນະທີ່ຄ່າ  $C$  ໃຫຍ່ຈະບັງຄັບໃຫ້ແບບຈຳລອງຈຳແນກຜິດໃຫ້ນ້ອຍລົງ.

**ປໍ່າພາຫາ.** ໃນທຳນອງເດືອກກັນ ລາກຮານຈົ່າກົງຂັ້ນຂອງປໍ່າພາຫາປັບປຸງ (ນິພຈນ [4.25](#)) ສື່ບັນດາ

$$J(\mathbf{w}, b, \xi, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \cdot (y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i$$

โดย  $\alpha_i, \beta_i \geq 0$ . ทั้ง  $\alpha_i$  และ  $\beta_i$  เป็นลากรานจ์พารามิเตอร์.

จากทฤษฎีบทكارูซคุนท์เกอร์ ณ จุดที่ดีที่สุด ค่าพารามิเตอร์ที่ดีที่สุด  $w_o, b, \xi_o$  จะทำให้เงื่อนไขดังนี้เป็นจริง. เงื่อนไขที่หนึ่ง  $\alpha_i \geq 0$  และ  $\beta_i \geq 0$ . เงื่อนไขที่สอง

$$\nabla_w J(w_o, b, \xi_o) = 0$$

$$\nabla_b J(w_o, b, \xi_o) = 0$$

$$\nabla_{\xi} J(w_o, b, \xi_o) = 0.$$

หลังจากหาอนุพันธ์และแก้สมการแล้ว สรุปได้ว่า

$$w_o = \sum_{i=1}^N \alpha_i y_i \phi(x_i) \quad (4.26)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (4.27)$$

$$C - \alpha_i - \beta_i = 0 \text{ for } i = 1, \dots, N. \quad (4.28)$$

สมการ 4.28 คือ  $\beta_i = C - \alpha_i$ . แทนค่าเหล่านี้ เข้าไปในลากรานจ์ฟังก์ชันแล้ว ลากรานจ์ฟังก์ชัน ณ จุดที่ดีที่สุด  $J'$  สามารถเขียนได้ว่า

$$J'(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (4.29)$$

เมื่อ  $\sum_{i=1}^N \alpha_i y_i = 0$  และ  $\alpha_i \geq 0$  กับ  $\beta_i \geq 0$ . แต่  $\beta_i \geq 0$  เทียบเท่ากับ  $\alpha_i \leq C$ . ดังนั้นปัญหาคู่สามารถระบุได้เป็น

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

s.t.

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad (4.30)$$

$$0 \leq \alpha_i \leq C \quad \text{for } i = 1, \dots, N.$$

สังเกตว่า ปัญหาคู่สำหรับกรณีทั่วไป แทบจะเหมือนกับปัญหาคู่กรณีข้อมูลแบ่งแยกได้โดยสมบูรณ์เลย ต่างกันเพียงแต่เงื่อนไขของค่า  $\alpha_i$  ที่เปลี่ยนมาเป็น  $0 \leq \alpha_i \leq C$ .

การอนุมานค่าฟังก์ชันแบ่งแยกก็ทำได้โดยการคำนวณสมการ 4.18 เช่นเดิม. และค่าพารามิเตอร์  $b_o$  ในกรณีที่ข้อมูลไม่สามารถแบ่งแยกได้สมบูรณ์ สามารถพิจารณาจาก เงื่อนไข  $y_i (w^T z_i + b) \geq 1 - \xi_i$

(อสมการ 4.24) ที่เมื่อ  $\alpha_i > 0$  แล้ว เงื่อนไขจะทำงาน. นั่นคือ  $y_i (\mathbf{w}^T \mathbf{z}_i + b) = 1 - \xi_i$ . แต่เรา秧ไม่สามารถแก้สมการนี้ได้ เพราะเรา秧ไม่มีรู้ค่า  $\xi_i$ . อย่างไรก็ตาม จากการที่  $\xi_i \geq 0$  เป็นเงื่อนไข ที่ควบคุมด้วย ลักษณะพารามิเตอร์  $\beta_i$ . นั่นคือ เรา秧ว่า เมื่อ  $\beta_i > 0$  (เทียบเท่า  $\alpha_i < C$ ) แล้ว เงื่อนไขจะทำงาน ซึ่งคือ  $\xi_i = 0$ . ดังนั้น จุดข้อมูลที่  $0 < \alpha_i < C$  จะบอกได้ว่า  $y_i (\mathbf{w}^T \mathbf{z}_i + b) = 1$  ซึ่งเราสามารถใช้จุดข้อมูลเหล่านี้ แก้สมการหาค่า  $b_o$  ได้. นั่นคือ

$$y_i \cdot g(\mathbf{x}_i) = 1 \text{ เมื่อ } i \in \{j : 0 < \alpha_j < C\} \quad (4.31)$$

$$\begin{aligned} \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) + b_o &= y_i \\ b_o &= y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i). \end{aligned} \quad (4.32)$$

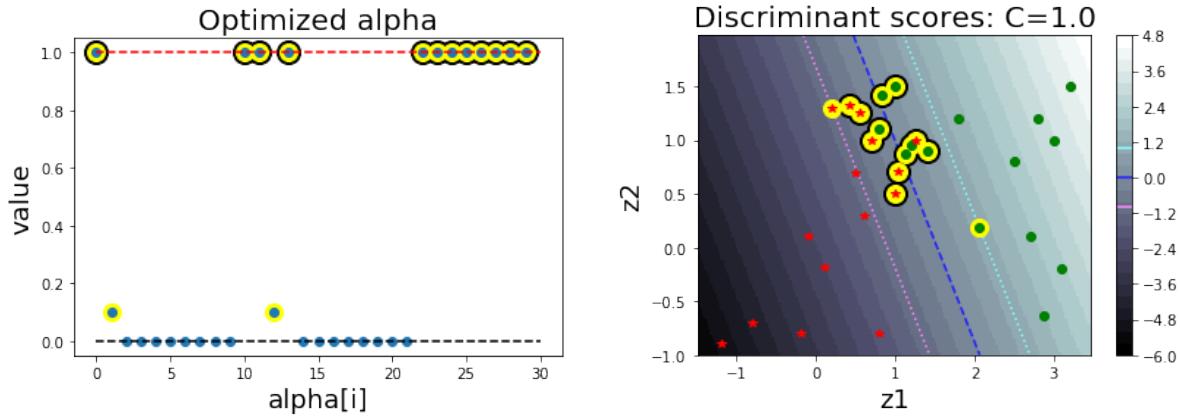
เช่นเดียวกัน  $i$  อาจเลือกจากดัชนีหนึ่ง ซึ่งทำให้  $0 < \alpha_i < C$  หรือ อาจใช้ค่าเฉลี่ย ซึ่งคือ

$$b_o = \frac{1}{|S'|} \sum_{i \in S'} \left( y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) \right) \quad (4.33)$$

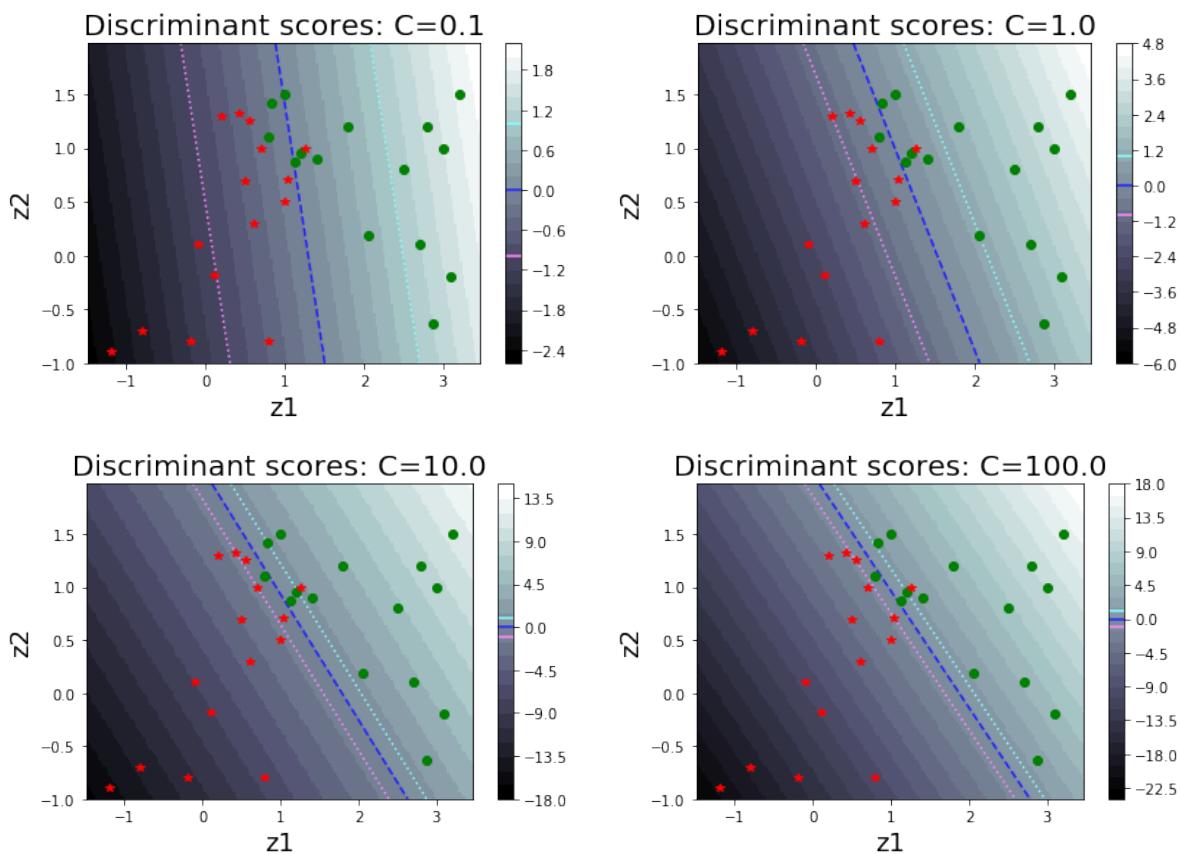
เมื่อ  $S = \{i : \alpha_i > 0\}$  และ  $S' = \{i : 0 < \alpha_i < C\}$  เป็นเซตดัชนีของชัพพร์ตเวกเตอร์ และชัพพร์ตเวกเตอร์ที่แนวขอบเขตของการแบ่ง ตามลำดับ. หมายเหตุ  $S$  มาจากสมการ 4.26 ที่คำนวนทุกตัว แต่  $\alpha_i = 0$  ไม่มีผล ดังนั้นจึงเลือกเฉพาะที่  $\alpha_i > 0$  มาคำนวน เพื่อลดการคำนวนที่ไม่จำเป็นและลดข้อมูล  $(\mathbf{x}_i, y_i)$  ที่ต้องเก็บรักษาไว้. ส่วน  $S'$  มาจากทฤษฎีบพศุชคุณทักษิณที่ทำให้อสมการ 4.24 เปลี่ยนมาอยู่ในรูปสมการ 4.31 เพื่อทำให้สามารถคำนวนค่า  $b_o$  ได้.

รูป 4.19 แสดงค่า  $\alpha_i$  ต่าง ๆ ที่ฝึกเสร็จ (ภาพซ้าย) และค่าฟังก์ชันแบ่งแยก (ภาพขวา). รูป 4.20 แสดงตัวอย่างของพฤติกรรมการจำแนกของแบบจำลอง เมื่อเลือกค่า  $C$  ต่าง ๆ. สังเกตว่า ที่ค่า  $C$  ขนาดเล็ก จะเห็นขอบเขตของการแบ่งกว้าง และมีจุดข้อมูลล้ำแนวขอบเขตของการแบ่งจำนวนมาก. ที่ค่า  $C$  ขนาดใหญ่ ฟังก์ชันแบ่งแยกจะปรับการคำนวน เพื่อให้จุดข้อมูลล้ำแนวขอบเขตของการแบ่งออกไปน้อยลง แต่ก็ทำให้ขอบเขตของการแบ่งแคบลง.

นอกจาก ชัพพร์ตเวกเตอร์แมชชีนในรูปแบบดังเดิม ที่ระบุด้วยนิพจน์ 4.30 ยังมีรูปแบบที่พัฒนาขึ้นมา



รูปที่ 4.19: ผลการฝึกชัพพร์ตเวกเตอร์แมชชีนกรณีที่  $C$  ต่าง ๆ. ภาพซ้าย แสดงค่า  $\alpha_i$  ที่ดัชนี  $i$  ต่าง ๆ. เส้นประสีดា และเส้นประสีแดง แสดงแนวศูนย์ และแนวค่า  $C$  ซึ่งเป็นขอบของช่วงค่าที่อนุญาตสำหรับ  $\alpha_i$ . ค่า  $\alpha_i > 0$  เน้นด้วยสีเหลือง และค่า  $\alpha_i = C$  เน้นด้วยขอบสีดำอีกที. ภาพขวา แสดงจุดข้อมูลและค่าฟังก์ชันแบ่งแยกในปริภูมิลักษณะสำคัญ. ชัพพร์ตเวกเตอร์ ( $\alpha_i > 0$ ) เน้นด้วย สีเหลือง และชัพพร์ตเวกเตอร์ที่มี  $\alpha_i = C$  เน้นด้วยขอบสีดำอีกที.



รูปที่ 4.20: พฤติกรรมของชัพพร์ตเวกเตอร์แมชชีนที่ค่า  $C$  ต่าง ๆ. แต่ละภาพ แสดง จุดข้อมูล (วงกลมสีเขียว แทนจุดข้อมูลกลุ่ม บวก. ดาวสีแดง แทนจุดข้อมูลกลุ่มลบ) ค่าฟังก์ชันแบ่งแยก (สีฟ้า ที่แสดงในระบบระดับสีเทา โดยค่าของสีแสดงด้วยแบบสีต้านข้าง) อกิริรนา (เส้นประสีน้ำเงิน) และแนวขอบเขตของการแบ่ง (เส้นประสีม่วงอ่อน สำหรับแนวผังลบ และเส้นประสีฟ้าเขียว สำหรับ แนวผังบวก). ค่าพารามิเตอร์  $C$  แสดงอยู่เหนือภาพ.

ใหม่อีก เช่น นิวตันพอร์ตเวกเตอร์แมชชีน ( $\nu$ -SVM[181]) ที่ wang กรอบปัญหาเป็น

$$\begin{aligned} \text{maximize}_{\alpha} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \\ & \sum_{i=1}^N \alpha_i \geq v, \\ & 0 \leq \alpha_i \leq 1/N \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.34)$$

โดย  $v$  เป็นอภิมานพารามิเตอร์ แทน  $C$  ในนิพจน์ 4.30.

## ฟังก์ชันเครอร์เนล

ซัพพอร์ตเวกเตอร์แมชชีน จัดการการคำนวณอย่าง сложสลายที่ในการฝึก (นิพจน์ 4.30 และสมการ 4.33) และการอนุมาน (สมการ 4.19) สามารถทำงานโดยตรงกับฟังก์ชันเครอร์เนล (kernel function)  $k(\mathbf{x}, \mathbf{x}')$  โดยไม่จำเป็นต้องอาศัยฟังก์ชันลักษณะสำคัญ  $\phi(\mathbf{x})$ . โอกาสเช่นนี้ ทำให้การใช้งานซัพพอร์ตเวกเตอร์แมชชีนสามารถกำหนดฟังก์ชันเครอร์เนล ที่เทียบเท่าการทำงานในปริภูมิลักษณะสำคัญที่มีจำนวนมิติมาก ๆ ได้ โดยไม่จำเป็นต้องเข้าไปทำงานในปริภูมิที่มีมิติสูงนั้นโดยตรง. การใช้ประโยชน์แบบมุนี มักถูกเรียกว่า ลูกเล่นเครอร์เนล (kernel tricks).

ฟังก์ชันเครอร์เนล อาจนิยมตรง ๆ จากฟังก์ชันลักษณะสำคัญด้วย

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') = \sum_{m=1}^M \phi_m(\mathbf{x}) \phi_m(\mathbf{x}') \quad (4.35)$$

เมื่อ  $\mathbf{x}$  และ  $\mathbf{x}'$  เป็นจุดข้อมูลสองจุด. ฟังก์ชัน  $\phi(\mathbf{x})$  เป็นฟังก์ชันลักษณะสำคัญ และ  $\phi_m(\mathbf{x})$  เป็นส่วนประกอบที่  $m^{th}$  ของค่าฟังก์ชันลักษณะสำคัญ. ฟังก์ชันเครอร์เนล สามารถถูกออกแบบได้หลายวิธี. วิธีหนึ่ง (1) อาจกำหนดผ่านฟังก์ชันลักษณะสำคัญ และสมการ 4.35 อีกวิธีหนึ่งในการสร้างเครอร์เนล (2) อาจกำหนดฟังก์ชันเครอร์เนลโดยตรง โดยไม่ต้องอาศัยฟังก์ชันลักษณะสำคัญ แต่ต้องตรวจสอบว่าฟังก์ชันที่กำหนดนั้น มีคุณสมบัติเป็นฟังก์ชันเครอร์เนลได้. การตรวจสอบนั้น อาจจะใช้ทฤษฎีบทของเมอร์เซอร์ (Mercer's theorem ดู [83] สำหรับรายละเอียด) หรือ ใช้การตรวจแกรมเมทริกซ์ (Gram matrix)  $\mathbf{K} = [k_{ij}]$  สำหรับ  $i, j = 1, \dots, N$  เมื่อ  $N$  เป็นจำนวนจุดข้อมูลฝึก (หรือจำนวนซัพพอร์ตเวกเตอร์) และ  $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ . ฟังก์ชันจะมีคุณสมบัติ

เป็นฟังก์ชันเครื่องเนลได้ หากแกรมเมทริกซ์เป็นเมทริกซ์บวกแน่นอน<sup>5</sup> (positive definite matrix) สำหรับทุก ๆ ค่าที่เป็นไปได้ของ  $\mathbf{x}$  และ  $\mathbf{x}'$ .

แต่ (3) วิธีที่สะดวกกว่าในการสร้างฟังก์ชันเครื่องเนล คือสร้างจากฟังก์ชันเครื่องเนลที่ถูกตรวจสอบมาแล้วด้วยคุณสมบัติดังนี้ (จาก [15]). หาก  $k_1(\mathbf{x}, \mathbf{x}')$  และ  $k_2(\mathbf{x}, \mathbf{x}')$  มีคุณสมบัติเป็นฟังก์ชันเครื่องเนลได้แล้ว ฟังก์ชันต่อไปนี้ก็จะมีคุณสมบัติเป็นเครื่องเนลได้เช่นกัน

$$k(\mathbf{x}, \mathbf{x}') = ck_1(\mathbf{x}, \mathbf{x}') \quad (4.36)$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}') \quad (4.37)$$

$$k(\mathbf{x}, \mathbf{x}') = \text{polynomial}^+(k_1(\mathbf{x}, \mathbf{x}')) \quad (4.38)$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}')) \quad (4.39)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \quad (4.40)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \quad (4.41)$$

$$k(\mathbf{x}, \mathbf{x}') = k_3(g(\mathbf{x}), g(\mathbf{x}')) \quad (4.42)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}' \quad (4.43)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (4.44)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) \cdot k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (4.45)$$

เมื่อ  $c > 0$  เป็นค่าคงที่. ฟังก์ชัน  $f : \mathbb{R}^D \mapsto \mathbb{R}$  เป็นฟังก์ชันใด ๆ. ฟังก์ชัน  $\text{polynomial}^+$  เป็นฟังก์ชันพหุนามที่สัมประสิทธิ์ไม่มีค่าลบ. ฟังก์ชัน  $g : \mathbb{R}^D \mapsto \mathbb{R}^M$  และ  $k_3$  มีคุณสมบัติเป็นฟังก์ชันเครื่องเนลสำหรับ  $\mathbb{R}^M$ . เมทริกซ์  $\mathbf{A}$  สมมาตร และเป็นบวกกึ่งแน่นอน<sup>6</sup> เวกเตอร์  $\mathbf{x}_a, \mathbf{x}'_a$  เป็นส่วนหน้าของ  $\mathbf{x}, \mathbf{x}'$  และเวกเตอร์  $\mathbf{x}_b, \mathbf{x}'_b$  เป็นส่วนหน้าของ  $\mathbf{x}, \mathbf{x}'$  นั่นคือ  $\mathbf{x} = [\mathbf{x}_a, \mathbf{x}_b]^T$  และ  $k_a$  กับ  $k_b$  เป็นคุณสมบัติเป็นฟังก์ชันเครื่องเนลสำหรับปริภูมิอยู่ดังกล่าว.

ฟังก์ชันเครื่องเนลที่นิยม และแนะนำสำหรับการเริ่มต้นใช้งานชั้พพร์ตเวกเตอร์แมชชีน[32] ได้แก่ ฟังก์ชันเครื่องเนลเชิงเส้น และฟังก์ชันเครื่องเนลเก้าล์เชียน.

<sup>5</sup> เมทริกซ์บวกแน่นอน ไม่ได้หมายถึง ทุกส่วนประกอบเป็นบวก. แต่มีความหมาย ดังนิยามว่า เมทริกซ์สมมาตร  $\mathbf{Q}$  จะเรียกว่า บวกแน่นอน (positive definite) ก็ต่อเมื่อทุก ๆ ค่าลักษณะเฉพาะ (eigenvalues) ของ  $\mathbf{Q}$  เป็นบวก. เมทริกซ์สมมาตร  $\mathbf{Q}$  จะเรียกว่า บวกกึ่งแน่นอน (positive semidefinite) ก็ต่อเมื่อทุก ๆ ค่าลักษณะเฉพาะของ  $\mathbf{Q}$  เป็นบวกหรือศูนย์.

<sup>6</sup> ดูนิยาม บวกกึ่งแน่นอน (positive semidefinite).

ฟังก์ชันคอร์เนลเชิงเส้น (linear kernel) ที่นิยามเป็น

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' \quad (4.46)$$

ซึ่งคือ ฟังก์ชันลักษณะสำคัญเป็นฟังก์ชันเอกลักษณ์  $\phi(\mathbf{x}) = \mathbf{x}$ . ฟังก์ชันคอร์เนลเชิงเส้น สร้างจากนิยามของ เคอร์เนลในสมการ 4.35.

ฟังก์ชันคอร์เนลเกาส์เชียน (Gaussian kernel) นิยามเป็น

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (4.47)$$

ฟังก์ชันคอร์เนลเกาส์เชียน อาจจะมองว่าสร้างมาจากคุณสมบัติของคอร์เนล. พิจารณา

$$\|\mathbf{x} - \mathbf{x}'\|^2 = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{x}' + (\mathbf{x}')^T \mathbf{x}'$$

ซึ่งเท่ากับว่า

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\sigma^2} \mathbf{x}^T \mathbf{x}\right) \cdot \exp\left(\frac{1}{\sigma^2} \mathbf{x}^T \mathbf{x}'\right) \cdot \exp\left(-\frac{1}{2\sigma^2} (\mathbf{x}')^T \mathbf{x}'\right).$$

นั่นคือ ใช้ฟังก์ชันคอร์เนลเชิงเส้น  $\mathbf{x}^T \mathbf{x}'$  เป็นพื้นฐาน และใช้คุณสมบัติในสมการ 4.37, 4.39, และ 4.36 ประกอบ.

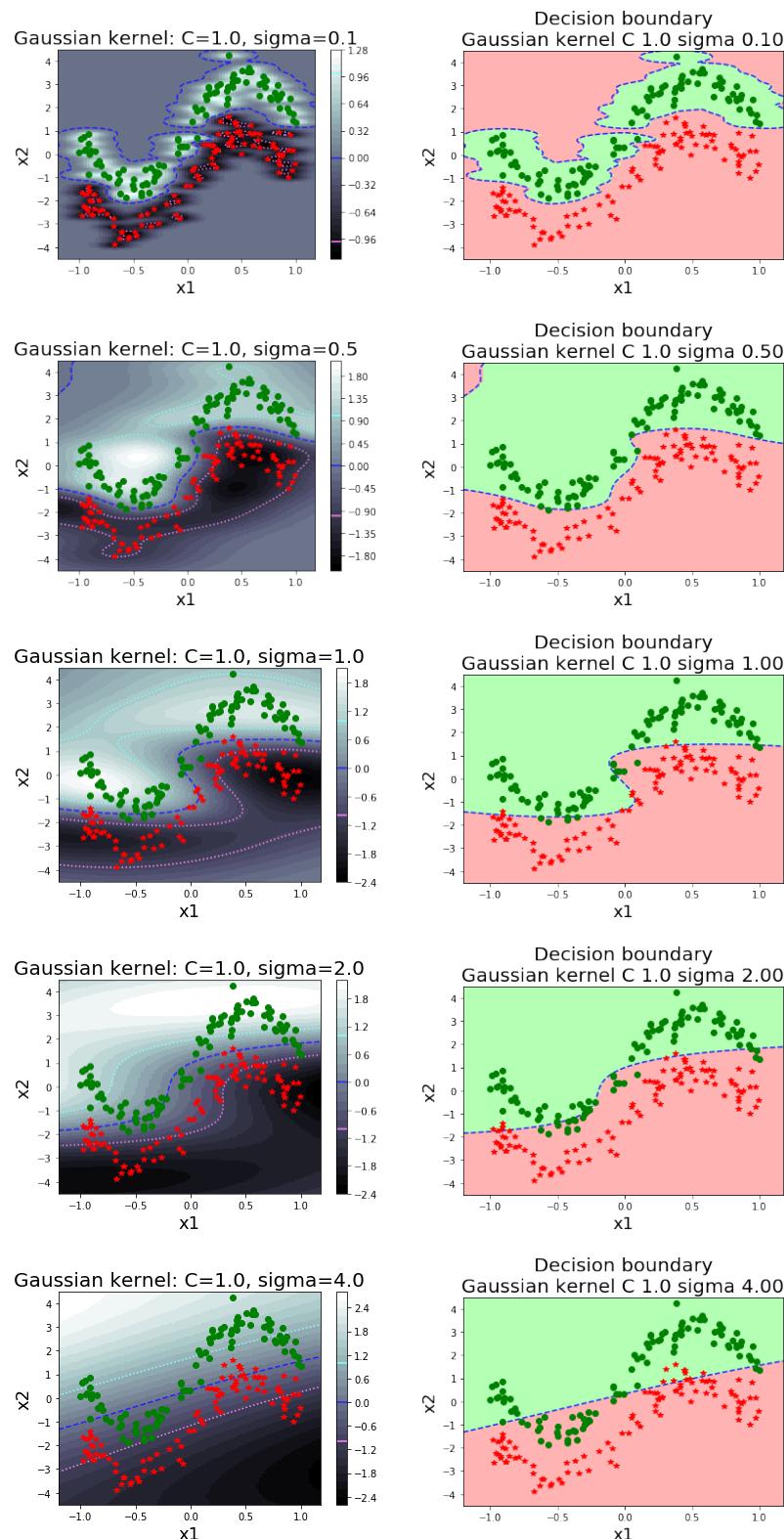
รูป 4.21 แสดงค่าฟังก์ชันแบ่งแยก เมื่อใช้ฟังก์ชันคอร์เนลเกาส์เชียน ที่ค่า  $\sigma$  ต่าง ๆ. สังเกตว่า ในภาพ เส้นค่าค่าฟังก์ชันแบ่งแยกเป็นศูนย์ (ซึ่งสะท้อนถึงอภิรະนาบในปริภูมิลักษณะสำคัญ) สามารถโค้งเลี้ยวไปตาม ข้อมูลได้ในปริภูมิข้อมูล. ภาพต่างทางขวา แสดงขอบเขตตัดสินใจ (decision boundary) ซึ่งเป็นส่วนในปริภูมิ ข้อมูล ที่ข้อมูลที่อยู่ภายใต้บริเวณจะถูกตัดสินตามชนิดของขอบเขตตัดสินใจ.

รูป 4.22 แสดงการใช้ฟังก์ชันคอร์เนลเชิงเส้น เพื่อเปรียบเทียบ.

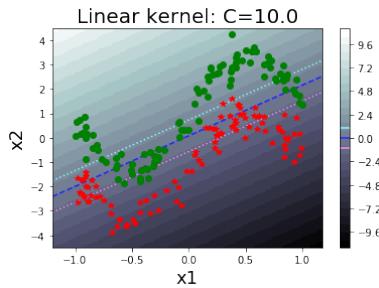
## 4.3 อภิรานศัพท์

**การตรวจหาวัตถุ (object detection):** การกิจกรรมหาตำแหน่งของวัตถุในภาพ หากภาพมีวัตถุอยู่.

**วิธีหน้าต่างเลื่อน (sliding window):** วิธีการเลือกส่วนภาพขนาดที่กำหนดจากข้อมูลภาพใหญ่ โดยการเลือก ส่วนภาพ จะเลือกทั่วถึงจากทุกบริเวณในภาพใหญ่ ซึ่งอาจเริ่มจากมุมซ้ายบนของภาพใหญ่ เลือกส่วน ภาพอ กมา แล้วขยับไปทางขวา และทำเช่นนี้ไปจนสุดปลายด้านขวา แล้วจึงขยับลงล่างและไปเริ่ม จากซ้ายสุด และทำลักษณะเช่นนี้อีก จนครอบคลุมบริเวณทั้งภาพใหญ่.



**รูปที่ 4.21:** การทำงานของชั้ฟฟอร์ตเกกเตอร์แมชชีน ด้วยเกาส์เซียนคอร์เนลที่ค่า  $\sigma$  ต่าง ๆ. ค่า  $C$  และ  $\sigma$  ระบุไว้เหนือภาพ. ภาพช้าย แสดงค่าฟังก์ชันแบ่งแยกของชัฟฟอร์ตเกกเตอร์แมชชีนในปริภูมิข้อมูล. ค่าฟังก์ชันแบ่งแยกแสดงด้วยระดับสีเทา ซึ่งค่าระบุด้วยແບບสีด้านข้าง. เส้นประสีม่วง สีน้ำเงิน และสีฟ้าเขียว แสดงแนวที่ค่าฟังก์ชันแบ่งแยกเป็น  $-1, 0$ , และ  $1$  ตามลำดับ. จุดข้อมูลຟີກ แสดงด้วยวงกลมສีເຊີຍ (ກລຸ່ມບວກ) และดาวສີແຕງ (ກລຸ່ມລບ). ລາພວ່າ แสดงຈຸດຂອ້ມລົບຟີກ ກັບຂອນເຫດຕັດສິນໃຈ. ຂອບເຂດຕັດລືນໃຈສໍາຮັບກລຸ່ມບວກ ແລະ ດ້ວຍສີເຊີຍອ່ອນ. ຂອບເຂດຕັດລືນໃຈສໍາຮັບກລຸ່ມລບ ແລະ ດ້ວຍສີໜົມພູ.



รูปที่ 4.22: การทำงานของชั้พพอร์ตเวกเตอร์แมชชีน ด้วยคอร์เนลเชิงเส้น เพื่อเปรียบเทียบกับเกาส์เชียนในรูป 4.21.

**ขนาดขั้บเลื่อน (stride):** ขนาดของการเลื่อนหน้าต่างแต่ละครั้ง.

**การสกัดลักษณะสำคัญ (feature extraction):** การแปลงอินพุตดั้งเดิม ให้อยู่ในรูปแบบใหม่ โดยที่รูปแบบใหม่นี้จะช่วยให้ภาระกิจที่ต้องการดำเนินการได้สะดวกขึ้น.

**แบบจำลองแบ่งแยก (discriminative model):** แบบจำลองการจำแนกกลุ่ม ที่อาศัยหรือตีความได้ว่าใช้ความน่าจะเป็นภายหลัง  $\Pr(y|x)$  เมื่อ  $y$  เป็นค่ากลุ่มที่ต้องการทำนาย และ  $x$  เป็นอินพุตหรือตัวแปรต้น. ตัวอย่างเช่น โครงข่ายประสาทเทียม.

**แบบจำลองสร้างกำเนิด (generative model):** แบบจำลองการจำแนกกลุ่ม ที่อาศัยความน่าจะเป็น  $\Pr(x|y)$  ทางตรงหรือทางอ้อม เมื่อ  $x$  เป็นอินพุตหรือตัวแปรต้น และ  $y$  เป็นค่ากลุ่ม. โดยที่ว่าเป็น  $x$  จะอยู่ในปริภูมิที่มีขนาดใหญ่กว่า  $y$  มาก ๆ เช่น ปัญหาการจำแนกภาพคน โดยเป็นภาพสเกลเทาขนาด  $H \times W$  ตัวแปร  $\mathbf{x}$  อยู่ในปริภูมิ  $\mathbb{R}^{H \times W}$  ในขณะที่ตัวแปร  $\mathbf{y}$  อยู่ในปริภูมิ  $\{+1, -1\}$ .

**ฟังก์ชันแบ่งแยก (discriminant function):** แบบจำลองการจำแนกกลุ่ม ทำการคำนวณค่าเพื่อจำแนกกลุ่ม โดยตรง ไม่อาศัยและไม่สามารถตีความในเชิงความน่าจะเป็น. ตัวอย่างเช่น ชัพพอร์ตเวกเตอร์แมชชีน.

**กล่องขอบเขต (bounding box):** บริเวณสี่เหลี่ยมที่เป็นขอบเขตภายในภาพ ซึ่งแต่ละกล่องขอบเขตสามารถระบุได้ด้วยสี่ค่า เช่น พิกัด  $(x, y)$  มุมซ้ายบน และขนาดความกว้างกับความสูงของกล่อง  $(w, h)$ .

**การกำจัดการระบุช้ำซ้อน (redundancy removal):** กลไกที่สำคัญสำหรับการตรวจสอบภาพวัตถุ เพื่อกำจัดการระบุการตรวจพบร่องแต่สองอันขึ้นไป ที่จริง ๆ แล้วระบุถึงวัตถุเดียวกัน.

**วิธีระงับค่าไม่มากสุดท้องถิ่น (non-local-maximum suppression):** วิธีหนึ่งในการกำจัดการระบุช้าซ้อนที่ดำเนินการด้วยการตัดทิ้งกล่องขอบเขตที่มีค่าความหมายสมไม่มากที่สุด เมื่อเปรียบเทียบกับกล่องขอบเขตอื่น ๆ ที่อยู่รอบ ๆ กล่องนั้น โดย ค่าความหมายสม คือค่าที่ใช้วัดความมั่นใจว่ากล่องขอบเขตนั้นมีรัฐุที่ค้นหาอยู่.

**ไอโอยู (IoU หรือ intersection of union):** ปริมาณวัด ที่วัดจากสัดส่วนพื้นที่ซ้อนทับกันของกล่องขอบเขตสองกล่อง ต่อพื้นที่รวม เพื่อบอกความใกล้เคียงของตำแหน่งการตรวจจับ อาจใช้ในกลไกของการตรวจจับ หรือใช้ในกระบวนการประมวลผล.

**วิธีการประมาณความหนาแน่นแก่น (kernel density estimation):** วิธีหนึ่งในการประมาณความหนาแน่นความน่าจะเป็นของข้อมูล.

**ค่าเฉลี่ยค่าประมาณความเที่ยงตรง (mean Average Precision หรือ mAP):** วิธีการวัดความสามารถของระบบตรวจจับภาพวัตถุ ที่คำนวณจากค่าประมาณพื้นที่ใต้กราฟของค่าความเที่ยงตรงและค่าระลึกกลับของวัตถุชนิดต่าง ๆ แล้วนำมาเฉลี่ยกัน.

**ปริภูมิลักษณะสำคัญ (feature space):** ปริภูมิหรือเซตของของค่าต่าง ๆ ที่เป็นไปได้ทั้งหมดของข้อมูล ในรูปแบบที่น่าจะช่วยให้ภาระกิจที่ต้องการทำได้ง่ายขึ้น.

**ซัพพอร์ตเวกเตอร์แมชชีน (support vector machine):** แบบจำลองจำแนกค่าทวิภาค ที่จำแนกข้อมูล ด้วยการใช้อภิรานาบในปริภูมิลักษณะสำคัญ. อภิรานาบที่ใช้ ถูกเลือกมาจากการอภิรานาบที่สามารถแบ่งข้อมูลตัวอย่างได้ของเขตของการแบ่งกว้างที่สุด.

**อภิรานาบ (hyperplane):** ระนาบในปริภูมิหลายมิติ. สำหรับปริภูมิสองมิติ อภิรานาบคือเส้นตรง. สำหรับปริภูมิสามมิติ อภิรานาบคือแผ่นตรงเรียบ. ปริภูมิหลายมิติได ๆ (กี่มิติก็ตาม) อภิรานาบสามารถบรรยายได้ด้วยสมการ  $\mathbf{w}^T \mathbf{x} + b = 0$  เมื่อ  $\mathbf{x}$  เป็นจุดใด ๆ ในปริภูมิ และ  $\mathbf{w}$  กับ  $b$  เป็นพารามิเตอร์ของอภิรานาบ.

**ขอบเขตของการแบ่ง (margin of separation):** ความห่างที่แบ่งกลุ่มข้อมูลสองกลุ่มออกจากกัน (ซึ่งอาจแบ่งได้สมบูรณ์ หรือไม่สมบูรณ์ก็ตาม).

**ขอบเขตตัดสินใจ (decision boundary):** บริเวณในปริภูมิข้อมูล ที่จุดข้อมูลต่าง ๆ หากอยู่ภายใต้ในบริเวณจะถูกจำแนกชนิดตามชนิดของขอบเขตตัดสินใจ.

**ชัพพร์ตเวกเตอร์ (support vectors):** จุดข้อมูลที่สำคัญต่อการกำหนดอภิรະนาบ.

**ลูกเล่นเครอร์เนล (kernel tricks):** การอาศัยรูปแบบการคำนวณของชัพพร์ตเวกเตอร์แมชีน ที่สามารถกำหนดฟังก์ชันเครอร์เนลได้โดยตรง และไม่ต้องกำหนดฟังก์ชันลักษณะสำคัญ.

**ฟังก์ชันเครอร์เนล (kernel function):** ฟังก์ชันคำนวณค่าสเกลาร์ ที่บรรยายความสัมพันธ์ระหว่างจุดข้อมูลสองจุด. ในปริภูมิลักษณะสำคัญ.

**ฟังก์ชันเครอร์เนลเชิงเส้น (linear kernel):** ฟังก์ชันเครอร์เนล  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ .

**ฟังก์ชันเครอร์เนลเกาส์เซียน (gaussian kernel):** ฟังก์ชันเครอร์เนล  $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$ .

## 4.4 แบบฝึกหัด

``Try not to become a man of success,  
but rather try to become a man of value.''

---Albert Einstein

“อย่าพยายามเป็นคนประสบความสำเร็จ  
แต่ให้พยายามเป็นคนที่มีคุณค่า.”

—อัลเบิร์ต ไอน์สไตน์

### แบบฝึกหัด 4.1

จะสร้างข้อมูลหนึ่งมิติขึ้นมา (อาจใช้คำสั่ง เช่น `np.random.normal`) และใช้วิธีการประมาณความหนาแน่นแก่น (สมการ 4.3 และ 4.4) เพื่อประมาณความหนาแน่นความน่าจะเป็น จากข้อมูลนั้น โดยทดลองค่า  $\sigma$  หลาย ๆ ค่า. สังเกตผล อภิปราย และสรุป.

รายการ 4.1 แสดงโปรแกรมวิธีการประมาณความหนาแน่นแก่น. ตัวอย่างการเรียกใช้ เช่น

```
datax = np.random.normal(3, 2, 100).reshape((1,-1))
xs = np.linspace(-2, 8, 50).reshape((1,-1))
pdx = kde(xs, datax, sigma=1)
plt.plot(xs[0,:], pdx[0,:], 'k')
```

บรรทัดแรกเป็นคำสั่งเพื่อสร้างข้อมูล `datax` ขึ้นมา จากการแจกแจงแบบเกาส์เชียน จำนวน 100 จุดข้อมูล โดยมีค่าเฉลี่ยเป็น 3 และค่าเบี่ยงเบนมาตรฐานเป็น 2. บรรทัดที่สอง สร้างค่า  $x$  ที่ต้องการสาม และบรรทัดที่สาม คือการเรียกโปรแกรม `kde` เพื่อประมาณความหนาแน่นความน่าจะเป็นของข้อมูล `datax`. บรรทัดสุดท้ายเป็นการวาดกราฟแสดงความหนาแน่นที่ค่าอินพุตต่าง ๆ. ดูตัวอย่างจากรูป 4.10.

รายการ 4.1: โปรแกรมวิธีการประมาณความหนาแน่นแก่น

---

```
1 def kde(x, kx, sigma=1):
2     ...
3     x: D x Nx
4     kx: D x Nk; D - # dimensions, Nk - # datapoints
5     ...
6     N = x.shape[1]
7     px = np.zeros((1, N))
8     norm = 1/N * 1/np.sqrt(2*np.pi*sigma**2)
9
10    for n in range(N):
11        distn = np.sum((x[:,[n]] - kx)**2, axis=0) # (Nk, )
12        texn = np.exp(-distn/(2*sigma**2)) # (Nk, )
```

```

13     px[0,n] = norm * np.sum(texn)
14
15     return px

```

## แบบฝึกหัด 4.2

จงสร้างข้อมูลสองมิติขึ้นมา โดยอาจใช้คำสั่ง เช่น

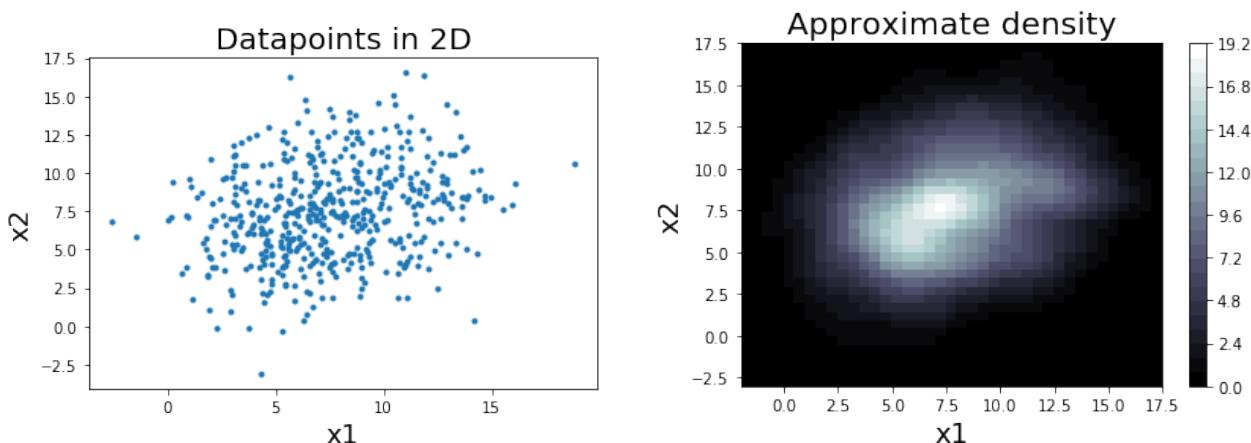
```

mu = [6, 9]
cov = [[8, -5], [-5, 9]]
x = np.random.multivariate_normal(mu, cov, 500).reshape((2,-1))

```

เมื่อ `mu` และ `cov` แทนค่าเฉลี่ยและค่าความแปรปรวนร่วมเกี่ยว ตามลำดับ และสร้างเป็นข้อมูลสองมิติ จำนวน 500 จุดข้อมูล. และใช้วิธีการประมาณความหนาแน่นแก่น (สมการ 4.3 และ 4.4) เพื่อประมาณความหนาแน่นความน่าจะเป็น จากข้อมูลนั้น โดยทดลองค่า  $\sigma$  หลาย ๆ ค่า. สังเกตผล ภาระ ราย และสรุป.

รูป 4.23 แสดงตัวอย่างข้อมูลสองมิติที่สร้างขึ้น และค่าความหนาแน่นที่ประมาณออกมา.



รูปที่ 4.23: ตัวอย่างการประมาณค่าความหนาแน่นความน่าจะเป็นสำหรับข้อมูลสองมิติ. ภาพซ้าย แสดงจุดข้อมูลตัวอย่าง. ภาพขวา แสดงค่าประมาณความหนาแน่นความน่าจะเป็นของข้อมูล. ระดับสีแทนค่าความหนาแน่นความน่าจะเป็น ซึ่งค่าแสดงด้วยแถบสีด้านข้าง.

## แบบฝึกหัด 4.3

แบบฝึกหัดนี้ เราจะศึกษารูปปั๊มน้ำของชัพพอร์ตเวกเตอร์แมชชีน. จงสร้างข้อมูลขึ้นมาจำนวน 200 จุด ข้อมูล โดยเป็นกลุ่มบวกและกลุ่มลบอย่างละครึ่ง จุดข้อมูลอยู่ในปริภูมิสองมิติ และข้อมูลสามารถแบ่งแยกได้อย่างสมบูรณ์เชิงเส้น เช่น ข้อมูลที่แสดงในรูป 4.24. และแก้ปัญหาในรูปปั๊มน้ำของชัพพอร์ตเวกเตอร์แมชชีน เพื่อหาอภิรานาบแบ่งแยก นั่นคือ หาค่าพารามิเตอร์  $w$  และ  $b$  ดังแสดงในรูป 4.25.

ทบทวนปัญหาปัญม สำหรับข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น คือ

$$\begin{aligned} \text{minimize}_{\mathbf{w}, b} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, \dots, N. \end{aligned}$$

หมายเหตุ ปัญหานี้ เราจะใช้ฟังก์ชันเอกลักษณ์เป็นลักษณะสำคัญ นั่นคือ  $\mathbf{z} = \phi(\mathbf{x}) = \mathbf{x}$  (ซึ่งเทียบเท่าการใช้เครื่องเรนาลเชิงเส้น  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ ).

เราอาจสามารถแก้ปัญหาได้ด้วยวิธีการลงโทษ เช่น อาจกำหนดฟังก์ชันลงโทษ เป็น

$$P(\mathbf{w}, b) = \sum_i \text{relu}(1 - y_i \cdot (\mathbf{w}^T \mathbf{x}_i + b))$$

เมื่อ

$$\text{relu}(a) = \begin{cases} a & \text{เมื่อ } a \geq 0, \\ 0 & \text{หากเป็นกรณีอื่น.} \end{cases}$$

จากปัญหาปัญมของชั้พพอร์ตเวกเตอร์แมชชีน ตัวอย่างคำสั่งข้างล่าง

```
la = 100
loss_adaptor = lambda wb: loss(wb[:2], wb[2], la)[0,0]
dloss_adaptor = lambda wb: dloss(wb[:2], wb[2], la)
wb0 = np.zeros((3,1))
wbo, gd_losses, wbs = gd(dloss_adaptor, wb0, loss_adaptor,
    step_size = 0.0001, Nmax = 5000)
```

ใช้คันหาอภิรานาบ (ระบุด้วย  $\mathbf{w}$  และ  $b$  ซึ่งคือ  $wbo[:2]$  และ  $wbo[2]$  ตามลำดับ). โปรแกรม **loss** และ **dloss** รวมถึงโปรแกรมอื่นที่เกี่ยวข้องกำหนดดังแสดงในรายการ 4.2. โปรแกรม **gd** (แสดงในรายการ 4.3) คำนวณวิธีลงเกรเดียนต์.

**รายการ 4.2:** ตัวอย่างโปรแกรมการค้นหาอภิรานาบ จากปัญหาปัญมของชั้พพอร์ตเวกเตอร์แมชชีน ด้วยวิธีลงเกรเดียนต์. ตัวแปร **datax** และ **datay** แทนข้อมูลอินพุตและผลลัพธ์ตามลำดับ โดยทั้งคู่เป็น **np.array** สัดส่วน  $(2, N)$  และ  $(1, N)$  เมื่อ  $N$  เป็นจำนวนจุดข้อมูล. หมายเหตุ ตัวแปร **datax** และ **datay** เป็นตัวแปรส่วนกลาง (global variables). ดูแบบฝึกหัด 4.4 สำหรับตัวอย่างการทำเป็นโปรแกรมเชิงวัตถุ ซึ่งมีการจัดการข้อมูลที่เป็นสัดเป็นส่วนมากกว่า.

---

```
1 relu = lambda a: (a >= 0)*a
2 drelu = lambda a: (a >= 0)*1
3
4 def loss(w, b, la):
```

```

5     term1 = 0.5 * np.dot(w.transpose(), w)
6     term2 = relu(1 - datay * (np.dot(w.transpose(), datax) + b))
7     return term1 + la * np.sum(term2)
8
9 def dloss(w, b, la):
10    N = datax.shape[1]
11    term1 = np.vstack((w, 0))
12    dc = - datay * np.vstack((datax, np.ones((1, N))))
13    term2 = drelu(1 - datay*(np.dot(w.transpose(), datax) + b))*dc
14    sum_dpenalty = np.sum(term2, axis=1).reshape((-1, 1))
15
16    return term1 + la * sum_dpenalty

```

---

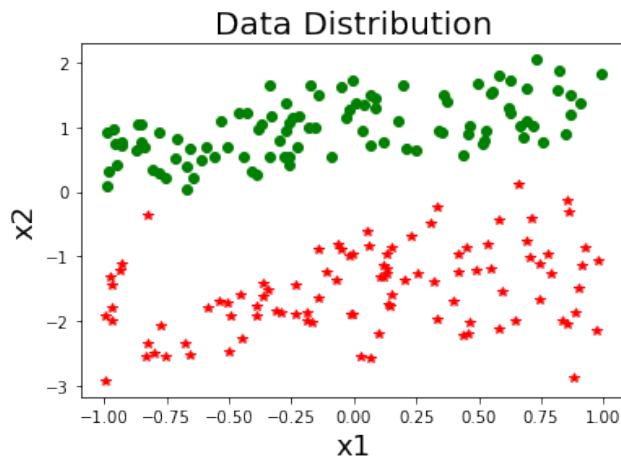
## รายการ 4.3: โปรแกรมวิธีลงเกรเดียนต์

```

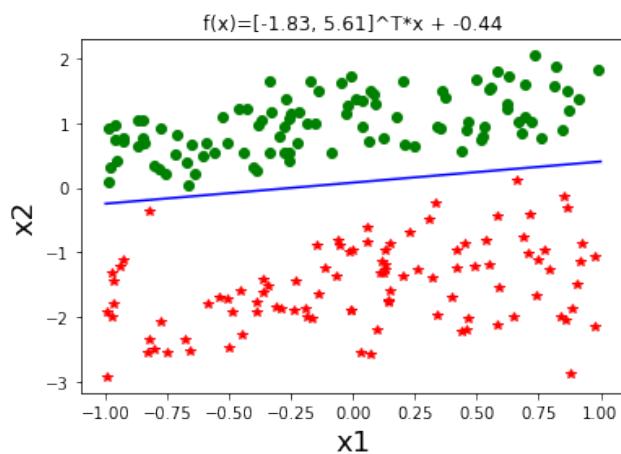
1 def gd(grad, v0, g, step_size=0.01, Nmax=100, tol=1e-6):
2     """
3         grad: gradient function
4         v0: initial value
5         g: objective function
6         """
7     losses = []
8     vs = np.zeros(v0.shape)
9     v = v0
10    gradv = grad(v)
11
12    for i in range(Nmax):
13        v = v - step_size * gradv
14        gradv = grad(v)
15
16        loss = g(v)
17        losses.append(loss)
18        vs = np.hstack((vs, v))
19
20        eps = np.linalg.norm(gradv)
21        if eps <= tol:
22            print('Reach termination criteria')
23            break
24
25    return v, losses, vs

```

---



รูปที่ 4.24: ตัวอย่างข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น.



รูปที่ 4.25: ตัวอย่างอภิรະนาบที่หาได้จากรูปปฐม สำหรับข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น. เส้นทึบสีนำเงิน แสดงอภิรະนาบที่หาได้.

**มอดูลไซไฟ.** หัวข้อนี้ แนะนำมอดูลไซไฟ (Scipy) ซึ่งมีเครื่องมือหลายอย่างสำหรับงานคำนวณทางวิทยาศาสตร์ รวมถึงมอดูล **optimize**. มอดูล **optimize** มีเครื่องมือการหาค่าดีที่สุดอยู่หลายวิธี. แม้วิธีลงเกรเดียนต์ (หัวข้อ 2.3) เป็นวิธีการที่ใช้งานได้ แต่ในทางปฏิบัติ มีวิธีที่มีประสิทธิภาพมากกว่าวิธีลงเกรเดียนต์อยุ่มากมาย และด้วยแบบจำลอง **optimize** เราสามารถนำไปใช้วิธีเหล่านั้นได้ โดยไม่ต้องใช้เวลาในการศึกษารายละเอียดของวิธีเหล่านั้นมากนั้น. มอดูล **optimize** สามารถนำเข้าได้ด้วยคำสั่ง

```
from scipy import optimize
```

แบบฝึกหัด 4.4 จะแก้ปัญหาเดียวกับแบบฝึกหัด 4.3 เพียงแต่จะใช้เครื่องมือจากแบบจำลอง **optimize** แทนวิธีลงเกรเดียนต์.

## แบบฝึกหัด 4.4

เช่นเดียวกับแบบฝึกหัด 4.3 จะสร้างข้อมูลขึ้นมาจำนวน 200 จุดข้อมูล โดยเป็นกลุ่มบวกและกลุ่มลบอย่างลงตัว จุดข้อมูลอยู่ในปริภูมิสองมิติ และข้อมูลสามารถแบ่งแยกได้อย่างสมบูรณ์เชิงเส้น แล้วแก้ปัญหาในรูปปัจจุบันของซัพพอร์ตเวกเตอร์แมชีน เพื่อหาอภิรักษานาบแบ่งแยก.

ศึกษาการเขียนโปรแกรมด้วยการใช้มอดูล **optimize** (ดังแสดงในตัวอย่างของแบบฝึกหัดนี้) เปรียบเทียบกับด้วยวิธีลงเกรเดียนต์ (แบบฝึกหัด 4.3) ทดลองใช้งาน สังเกตผล อภิปราย และสรุป.

ตัวอย่างคำสั่งข้างล่าง ฝึกแบบจำลอง และรายงานผลอภิรักษานาบที่พบ

```
svm = primal_SVM()
svm.phi = lambda xq: xq # identity projection
res = svm.train(datax, datay)
print('w=', svm.wopt.T)
print('b=', svm.bopt)
```

โดย บรรทัดแรก เป็นการสร้างตัวแปรตัว **svm** จากคลาส **primal\_SVM** ซึ่งแสดงในรายการ 4.4. บรรทัดที่สอง กำหนดฟังก์ชันลักษณะสำคัญเป็นฟังก์ชันเอกลักษณ์. บรรทัดที่สาม เป็นการฝึกซัพพอร์ตเวกเตอร์แมชีน ด้วยข้อมูล **datax** และ **datay**. ผลลัพธ์ที่ได้จากการฝึกจะสรุปอภิรักษานาบเป็นค่าของพารามิเตอร์ **svm.wopt** และ **svm.bopt** ที่ใช้บรรยายอภิรักษานาบ. สังเกตว่า เมท็อด **train** มีการเรียกใช้

```
res = optimize.minimize(minf, wb0, method='SLSQP', jac=gradf,
constraints=ineq_cons, options=options)
```

ซึ่งเป็นเครื่องมือการหาค่าดีที่สุด โดยระบุวิธีเป็น '**SLSQP**' (ศึกษารายละเอียดเพิ่มเติมจากเวปไซต์ของไซไฟหากสนใจ) โดยภายใน เมท็อด **train** ได้กำหนดค่า **minf** และ **gradf** ซึ่งคือ ฟังก์ชันจุดประสงค์ และฟังก์ชันเกรเดียนต์ ตามลำดับ พร้อมทั้งข้อจำกัดแบบสมการ **ineq\_cons**.

หมายเหตุ การใช้งานจริง สิ่งที่ต้องการ คือการทำนายกลุ่มของข้อมูล. เมท็อด **decision\_score** ใช้คำนวณค่าฟังก์ชันแบ่งแยก ซึ่งจะนำไปใช้ตัดสินกลุ่ม. ส่วนค่าของ **w** และ **b** นั้น เป็นรายละเอียดภายในไม่จำเป็นต้องรายงาน และการใช้งานจริงของซัพพอร์ตเวกเตอร์แมชีน ก็จะไม่มีการคำนวณค่า **w** อภิรักษานาบ เพราะว่า รูปแบบการคำนวณถูกแปลงไป เพื่อใช้ประโยชน์จากฟังก์ชันเครอร์เนล. ดูแบบฝึกหัด 4.5 สำหรับโปรแกรมซัพพอร์ตเวกเตอร์แมชีนที่ใช้งานจริง.

---

รายการ 4.4: โปรแกรมซัพพอร์ตเวกเตอร์แมชีน จากปัญหาปัจจุบัน สำหรับกรณีแบ่งแยกได้โดยสมบูรณ์

1 **class primal\_SVM:**

```

2     def __init__(self):
3         self.bopt = None
4         self.wopt = None
5         self.phi = None
6
7     def train(self, x, y, wb0,
8             options={'ftol': 1e-9, 'disp': True}):
9         assert (x.shape[1] == y.shape[1]) and (y.shape[0] == 1)
10
11        N = x.shape[1]
12        # projected x to z
13        z = self.phi(x)
14        assert z.shape[1] == x.shape[1]
15        D = z.shape[0] # number of projected dimensions
16
17        if wb0 is None: # w: (D,1), b: scalar
18            wb0 = np.random.normal(0, 1, D+1)
19
20    def primal_ineq(w, b):
21        w = w.reshape((-1,1))
22        b = np.asscalar(b)
23        cineq = y.reshape((1,-1)) * \
24            (np.dot(w.T,z.reshape((D,-1))) + b) -1
25        return cineq.reshape((-1,)) # (M,)
26
27    def primal_dc(w, b):
28        gradw = y.reshape((1,-1)) * z.reshape((D,-1))
29        gradb = y.reshape((1,-1))
30        dc = np.hstack( (gradw.T, gradb.T) )
31        return dc.reshape((-1,D+1)) # (M,D+1)
32
33    # inequality constraints: y[i] * ( w.T x[i] + b)-1 >= 0
34    ineq_cons = {'type': 'ineq',
35                 'fun' : lambda wb: primal_ineq(wb[:-1], wb[-1]),
36                 'jac' : lambda wb: primal_dc(wb[:-1], wb[-1])}
37
38    # Objective
39    def primal_f(wb): # primal loss: minimization form
40        w = wb[:-1].reshape((-1,1)) # D x 1
41        L = 0.5 * np.dot(w.T, w) # scalar
42        return np.asscalar(L)

```

```

43
44     def primal_grad(wb): # gradient of primal loss
45         w = wb[:-1].reshape((-1,1)) # D x 1
46         pgrad = np.vstack((w, 0))
47         return pgrad.reshape((-1,)) # (D,)
48
49     minf = lambda wb: primal_f(wb)
50     gradf = lambda wb: primal_grad(wb)
51     res = optimize.minimize(minf, wb0, method='SLSQP',
52                             jac=gradf, constraints=ineq_cons, options=options)
53
54     if not res.success:
55         return res
56
57     # Train succeeds.
58     self.wopt = res.x[:-1].reshape((-1,1))
59     self.bopt = res.x[-1]
60     return res
61
62     def decision_score(self, x):
63         # Project to feature space
64         z = self.phi(x)
65         assert z.shape[0] == self.wopt.shape[0]
66
67         # Compute the score
68         yhat = np.dot(self.wopt.T, z) + self.bopt
69         return yhat      # 1 x N

```

---

## แบบฝึกหัด 4.5

ปัญหาเดียวกับแบบฝึกหัด 4.4 แต่แก้ปัญหาจากปัญหาคู่'.

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

s.t.

$$\sum_{i=1}^N \alpha_i y_i = 0,$$

$$0 \leq \alpha_i \leq C \quad \text{สำหรับ } i = 1, \dots, N.$$

เนื่องจาก ปัญหาคู่' สำหรับกรณีแบ่งแยกได้โดยสมบูรณ์ กับกรณีที่ไม่ได้โดยสมบูรณ์ ตัวอย่างข้างล่างนี้ จึงใช้รูปแบบที่มีเลือกค่า  $C$  ขนาดใหญ่ก็จะให้ผลแบบเดียวกับกรณีแบ่งแยกได้โดยสมบูรณ์ ตัวอย่างข้างล่างนี้ จึงใช้รูปแบบที่มี

อภิมานพารามิเตอร์  $C$ .

โปรแกรมในรายการ 4.5 แสดงตัวอย่างโปรแกรมชั้พพอร์ตเวกเตอร์แมชชีน. สังเกต วิธีการเขียนโปรแกรมจะใช้การทำเวคตอร์ไซน์มากที่สุดเท่าที่จะทำได้ เนื่องจากประสิทธิภาพการคำนวณและความยืดหยุ่น. ตัวอย่าง เช่น หากการคำนวณค่าพารามิเตอร์  $b$  คือ  $b_o = y_i - \sum_{j \in S} \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}_i)$  สำหรับ  $i \in \{j : 0 < \alpha_j < C\}$ . นั่นคือ เทียบเท่า  $b_o = y_i - (\boldsymbol{\alpha}^T \odot \mathbf{y}) \cdot \mathbf{K}_{:,i}$  สำหรับ  $i \in \{j : 0 < \alpha_j < C\}$ . หมายเหตุ โปรแกรม 4.5 คำนวณค่า  $b$  ด้วยค่าเฉลี่ย ซึ่งจะซับซ้อนกว่าตัวอย่างการทำเวคตอร์ไซน์ที่อธิบายข้างต้นเล็กน้อย.

คล้ายกับตัวอย่างในแบบฝึกหัด 4.4 คำสั่งข้างล่าง แสดงตัวอย่างการฝึก และการอนุमानด้วยชัพพอร์ตเวกเตอร์แมชชีน

```
svm = cSVM()
svm.kernel = lambda xq, xp: np.dot(xq.T, xp) # linear kernel
C = 1
res = svm.train(datax, datay, C=C)
svm.decision_score(testx)
```

ในตัวอย่าง ใช้อภิมานพารามิเตอร์  $C = 1$  ฝึกด้วยข้อมูล **datax** และ **datay** ที่ต้องเป็นชนิด **np.array** สัดส่วน ( $D, N$ ) และ ( $1, N$ ) ตามลำดับ เมื่อ  $D$  และ  $N$  แทนจำนวนมิติของอินพุต และจำนวนจุดข้อมูลตามลำดับ. คำสั่งสุดท้าย ใช้คำนวณค่าฟังก์ชันแบ่งแยกสำหรับ ข้อมูลทดสอบ **testx**.

ลองศึกษาวิธีการเขียนโปรแกรม ทดลองใช้งาน สังเกตผล อภิปราย และสรุป. ทดลองค่า  $C$  ต่าง ๆ และรายงานดังตัวอย่างในรูป 4.20.

รายการ 4.5: ชัพพอร์ตเวกเตอร์แมชชีน

---

```
1 class cSVM:
2     def __init__(self):
3         self.epsilon = 0.001
4         self.bopt = None
5         self.sv = None
6         self.sy = None
7         self.salpha = None
8         self.kernel = None
9
10    def train(self, x, y, C=1, a0=None,
11               options={'ftol': 1e-9, 'disp': True}):
12        assert (x.shape[1] == y.shape[1]) and (y.shape[0] == 1)
```

```

13
14     N = x.shape[1]
15     if a0 is None:
16         a0 = np.random.normal(0, 1, N)
17
18     # parameter bounds: 0 <= alpha_i <= C for all i's
19     bounds = optimize.Bounds([0 for i in range(N)],
20                             [C for i in range(N)])
21
22     # inequality constraints: dummy
23     ineq_cons = {'type': 'ineq',
24                   'fun' : lambda a: np.array([0]),    # (1,)
25                   'jac' : lambda a: np.zeros((1,N))}   # (M,N)
26
27     # equality constraint: sum_i y_i alpha_i = 0
28     eq_cons = {'type': 'eq', 'fun' : lambda a: np.dot(y,
29                                         a.reshape((-1, 1))).reshape((-1,)),
30             'jac' : lambda a: y.reshape((1,N))}
31
32     # Objective
33     K = self.kernel(x, x)      # N x N
34     H = np.dot(y.T, y) * K     # N x N
35
36     def dual_minf(a, H): # dual Loss in minimization form
37         a = a.reshape((-1,1))
38         Q = 0.5 * np.dot(a.T, np.dot(H, a)) - np.sum(a)
39         return np.asscalar(Q)
40
41     def dual_grad(a, H, N): # gradient of dual loss
42         dQ = np.dot(H, a.reshape((-1,1))) - np.ones((N,1))
43         return dQ.reshape((-1,))    # (N,)
44
45     minf = lambda a: dual_minf(a, H)
46     gradf = lambda a: dual_grad(a, H, N)
47     res = optimize.minimize(minf, a0, method='SLSQP',
48                             jac=gradf, constraints=[ineq_cons, eq_cons],
49                             bounds=bounds, options=options)
50
51     if not res.success:
52         return res
53

```

```

54     alpha = res.x.reshape((-1,1))
55     sv_ids = np.where(alpha > self.epsilon)[0]
56     svp_ids = np.where( np.logical_and(alpha > self.epsilon,
57                                         alpha < C - self.epsilon) )[0]
58     Ns = len(sv_ids)
59     Np = len(svp_ids)
60
61     if Np == 0: # No support vector on the edge
62         print('# No support vector on the edge.')
63         svp_ids = sv_ids
64         Np = Ns
65
66     # support vectors
67     self.sv = x[:, sv_ids] # D x Ns
68     self.sy = y[0, sv_ids].reshape((1,-1)) # 1 x Ns
69     spy = y[0, svp_ids].reshape((1,-1)) # 1 x Np
70
71     # support alphas
72     self.salpha = alpha[sv_ids]
73
74     # optimal b
75     sK = K[np.repeat(sv_ids, Np),
76             np.tile(svp_ids, Ns)].reshape((Ns, Np))
77     self.bopt = np.mean(spy - \
78                         np.dot(self.salpha.T * self.sy, sK))
79     return res
80
81 def decision_score(self, x):
82     assert x.shape[0] == self.sv.shape[0] # x in D x N
83     # Compute support kernels
84     sK = self.kernel(x, self.sv) # N x Ns
85     assert (sK.shape[0] == x.shape[1])
86                 and (sK.shape[1] == self.sv.shape[1])
87
88     yhat = np.dot(sK, self.salpha*self.sy.reshape((-1,1))) \
89             + self.bopt
90     return yhat.T      # 1 x N

```

---

### แบบฝึกหัด 4.6

จงสร้างข้อมูลที่ไม่สามารถแบ่งแยกสมบูรณ์ได้เชิงเส้น โดยจะสร้างให้มีลักษณะไดกีได้ แต่ควรทำให้สา-

มารถตรวจสอบได้สะดวก เช่น อินพุตควรจะเป็นสองมิติ. ตัวอย่างอาจจะเช่นที่แสดงในรูป 4.26. ทดลองใช้ชั้พพอร์ตเวกเตอร์แมชชีน กับเครื่องเนลเชิงเส้น และเครื่องเนลเก้าส์เชิงที่ค่า  $\sigma$  ต่าง ๆ. ดูรูป 4.22 กับ 4.21 สำหรับตัวอย่าง. ออกแบบการทดลอง เพื่อศึกษาผลของ  $C$  และ  $\sigma$ . ทดลอง สังเกตผล อภิปราย และสรุป.

ตัวอย่าง คำสั่งข้างล่างแสดงการกำหนดค่าเครื่องเนลของตัวแปรต่อ (จาก SVM) ให้เป็นเครื่องเนลเก้าส์เชิง (ใช้  $\sigma = 2$ )

```
svm.kernel = lambda xq, xp: gaussian(xq, xp, sigma=2)
```

โดยโปรแกรม **gaussian** แสดงในรายการ 4.6.

รายการ 4.6: โปรแกรมเครื่องเนลเก้าส์เชิง สำหรับชัพพอร์ตเวกเตอร์แมชชีน

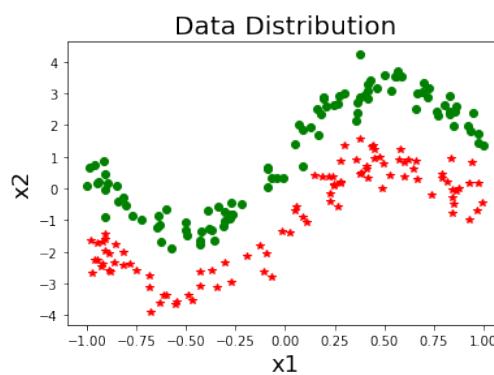
---

```

1 def gaussian(xq, xp, sigma=1):
2     assert xq.shape[0] == xp.shape[0]      # xq: D x Nx, xp: D x Ns
3     Nx = xq.shape[1]
4     Ns = xp.shape[1]
5     K = np.zeros((Nx, Ns))
6
7     c = -1/(2*sigma**2)
8     for i in range(Nx):
9         vi = xq[:,[i]] - xp # (D,1)-(D,Ns): broadcast to (D,Ns)
10        K[i,:] = np.exp(c*np.sum(vi**2, axis=0)) # (Ns,)
11
12    return K # Nx x Ns

```

---



รูปที่ 4.26: ตัวอย่างข้อมูลที่ไม่สามารถแบ่งแยกสมบูรณ์ได้เชิงเส้น.

ภาค ii

## การเรียนรู้เชิงลึก



## บทที่ 5

### การเรียนรู้เชิงลึก

``What you get by achieving your goals is not as important as what you become by achieving your goals."

---Johann Wolfgang von Goethe

“สิ่งที่คุณได้จากการบรรลุจุดมุ่งหมายไม่สำคัญเท่า สิ่งที่คุณเป็นจากการบรรลุจุดหมาย。”

—约翰·沃尔夫冈·歌德

โครงข่ายประสาทเทียมความลึกสองชั้นนั้น แม้จะสามารถทำงานหลาย ๆ อย่างได้ดี ดังที่แสดงในตัวอย่าง บท 3 และในทางทฤษฎีนั้น โครงข่ายประสาทเทียมความลึกสองชั้นนั้น จะสามารถฝึกให้ประมวลฟังก์ชันอะไรก็ได้[47, 91] แต่ในทางปฏิบัติแล้ว สำหรับงานที่ซับซ้อนมาก ๆ ทั้งจำนวนหน่วยช่องที่ต้องเพิ่มจำนวนมาศาก และการฝึกที่ใช้ทรัพยากรการคำนวณมาก รวมถึงข้อมูลพร้อมฉลากที่ต้องมีจำนวนมากพอ ทำให้ การประยุกต์ กับงานการรู้จำรูปแบบของโครงข่ายประสาทเทียมสองชั้น จำกัดอยู่มาก โดยเฉพาะกับงานที่ซับซ้อนมาก ๆ เช่น งานรู้จำภาพ เสียงพูด หรือภาษาธรรมชาติ.

จนกระทั่ง ความก้าวหน้าล่าสุด ก็คือ แนวทางของการเรียนรู้เชิงลึก (Deep Learning) ที่ได้ขยายความสามารถของการประยุกต์ใช้โครงข่ายประสาทเทียมไปแบบก้าวกระโดด การเรียนรู้เชิงลึก แม้จะมีโครงสร้างพื้นฐานเป็นโครงข่ายประสาทเทียม แต่มีกลไกสำคัญหลายอย่างที่ช่วยขยายความสามารถ ซึ่งหนึ่งในนั้นคือ การใช้โครงสร้างเชิงลึก หรือการใช้โครงข่ายประสาทเทียมที่มีจำนวนชั้นคำนวณมาก<sup>1</sup>.

<sup>1</sup> กรณับจำนวนชั้นของโครงข่าย ไม่ได้มีข้อตกลงสากล และอาจมีวิธีนับที่แตกต่างกันไป. ตัวอย่าง เช่น โครงข่ายสองชั้น  $\mathbf{y} = h^{(2)}(\mathbf{W}^{(2)}\mathbf{z}^{(1)} + \mathbf{b}^{(2)})$  โดย  $\mathbf{z}^{(1)} = h^{(1)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$  เมื่อ  $\mathbf{y}$  คืออัตราพุทธิที่ทำงาน และ  $\mathbf{x}$  คืออินพุตที่ถูกนำ. โครงข่ายลักษณะนี้ อาจนับเป็นสองชั้น ตามจำนวนชุดของค่าน้ำหนัก  $\mathbf{W}^{(2)}$  และ  $\mathbf{W}^{(1)}$  ซึ่ง ณ ที่นี้ ใช้แบบแผนนี้ในการนับ. แต่ผู้อ่านอาจพบ บางแห่งนับเป็นสามชั้น โดยนับชุดค่าของหน่วยอยู่ ได้แก่  $\mathbf{y}, \mathbf{z}^{(1)},$  และ  $\mathbf{x}$  โดยมองเสมือนว่า อินพุตที่เป็นชุดค่าของหน่วยอยู่. บางแห่ง อาจเลือกวิธีนับโครงข่ายนี้ ว่าเป็นโครงข่ายจำนวนหนึ่งชั้นซ่อน ได้แก่ ชั้นซ่อน  $\mathbf{z}^{(1)}$  เนื่องจากหากมองอินพุตกับเอาต์พุตเป็นชั้นของโครงข่าย อินพุตกับเอาต์พุตก็จะเป็นชั้นที่มีกันทุก ๆ โครงข่าย ฉะนั้น รายงานเฉพาะส่วนที่ต่าง ซึ่งก็คือจำนวนชั้นซ่อนก็พอ. วิธีการนับจำนวนชั้นของโครงข่าย จะมีความสำคัญอย่างเมื่อพิจารณาโครงข่ายที่มีความซับซ้อนมากขึ้น เช่น โครงข่ายอเล็กซ์เน็ต (หัวข้อ 6.5) ที่มีการแยกเส้นทางการคำนวณ และนำผลการคำนวณจากแต่ละเส้นทางกลับมารวมกันภายหลัง.

หากจะเริ่มต้นกล่าวถึงการเรียนรู้แบบลึก แม้แนวคิดจะมีนานานมากพอ ๆ กับจุดกำเนิดของโครงข่ายประสาทเทียมเอง และวิธีการแพร่กระจายย้อนกลับ (หัวข้อ 3.3) ก็มีความทั่วไปมากพอ ที่จะใช้ในกระบวนการฝึกของโครงข่ายประสาทเทียมกี่ชั้นก็ได้. แต่การประยุกต์ใช้โครงข่ายประสาทเทียมแบบลึกในช่วงก่อนศตวรรษที่ยังสืบอีกด้วยนั้นจำกัดอยู่มาก. อุปสรรคที่ที่สำคัญสำหรับการประยุกต์ใช้โครงข่ายประสาทเทียมแบบลึก ก็คือ ปัญหาการเลื่อนหายของเกรเดียนต์.

จนกระทั่งงานศึกษาที่สำคัญของยินตันและชาลาคูทินอฟ[87] ที่พัฒนาการฝึกโครงข่ายประสาทเทียมแบบลึกได้อย่างมีประสิทธิภาพ. หลังจากนั้น ก็มีการศึกษาการเรียนรู้เชิงลึกอย่างกว้างขวาง และการเรียนรู้เชิงลึกก็กลายเป็นศาสตร์และศิลป์ที่สำคัญสำหรับศาสตร์หลาย ๆ แขนง [114, 97, 178, 115, 48, 52, 64, 26, 157, 35, 123, 220] เช่น คอมพิวเตอร์วิทัคน์ (Computer Vision), การรู้จำคำพูด (Speech Recognition), การประมวลผลภาษาธรรมชาติ (Natural Language Processing), การค้นหายา (Drug Discovery), และเจโนมิกส์ (Genomics). ความสนใจในการเรียนรู้เชิงลึกและโครงข่ายประสาทเทียมมีสูงมาก จนทำให้เกิดการศึกษาและพัฒนาอุปกรณ์คำนวณ สำหรับโครงข่ายประสาทเทียมโดยเฉพาะ ได้แก่ หน่วยประมวลผลเชิงประสาทแปลง (Neuromorphic Processing Unit[127] คำย่อ NPU).

ปัจจัยของความสำเร็จของการเรียนรู้เชิงลึก หรือโครงข่ายประสาทเทียมแบบลึก มีอยู่หลายประการ ตั้งแต่ฮาร์ดแวร์ที่เร็วขึ้น, ข้อมูลที่มากและหลากหลายขึ้น, วิธีการเตรียมข้อมูลที่ดีขึ้น รวมถึงการทำการฝึกก่อน[65], ขั้นตอนวิธีการหาค่าดีที่สุดที่มีประสิทธิภาพมากขึ้น[93, 110], การฝึกที่ละเอียด[13], การใช้แบบจำลองที่จับลักษณะสำคัญของข้อมูลได้ดีขึ้น เช่น โครงข่ายคอนโวลูชัน[114] และโครงข่ายประสาทเวียนกลับ[88, 89, 180], การใช้กลไกการตกอ้อก[189], การใช้กลไกความลaise[54, 203], การใช้กลไกโครงข่ายปรับปรุงเชิงสร้าง[77, 145] ไปจนถึงการเปลี่ยนพัฟฟ์ชันกระตุ้นจากซิกมอยด์ไปเป็นพัฟฟ์ชันกระตุ้นที่ลดช่วงค่าอิมตัว เช่น เรลู[87].

บทที่ 5 นี้ อภิปรายการแก้ปัญหาการเลื่อนหายของเกรเดียนต์ ด้วยพัฟฟ์ชันกระตุ้นเรลู (หัวข้อ 5.1), เทคนิคการจัดการฝึกกับข้อมูลขนาดใหญ่ด้วยการฝึกที่ละเอียด (หัวข้อ 5.2), เทคนิคการตกอ้อก (หัวข้อ 5.3), วิธีการกำหนดค่าน้ำหนักเริ่มต้น (หัวข้อ 5.4), และขั้นตอนวิธีการฝึกที่มีประสิทธิภาพมากขึ้น (หัวข้อ 5.5). นอกจากนั้น ปัจจัยหนึ่งที่มีส่วนอย่างมาก ในพัฒนาการ และความสนใจ ไปจนถึงการประยุกต์ใช้ที่กว้างขวาง ก็คือเครื่องมือที่ช่วยให้การใช้งานเทคนิคต่าง ๆ เหล่านี้ทำได้สะดวกมากขึ้น หัวข้อ 5.7 อภิปรายตัวอย่างเครื่องมือที่ได้รับความนิยมอย่างสูง สำหรับการประยุกต์ใช้การเรียนรู้เชิงลึก.

การใช้โครงข่ายคอนโวลูชัน ที่หมายความว่า กับข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง เช่น ข้อมูลภาพ รวมไปจนถึง

เทคนิคการฝึกก่อน และตัวอย่างโครงสร้างของโครงข่ายคอนโวโลชันที่รู้จักกันอย่างกว้างขวาง อภิปรายในบทที่ 6. การนำโครงข่ายคอนโวโลชันไปประยุกต์ใช้กับงานการรู้จำทัศนรูปแบบ เป็นเนื้อหาหลักที่อภิปรายในบทที่ 7.

บทที่ 8 อภิปรายโครงข่ายประสาทเวียนกลับ ที่หมายความว่ามีลักษณะเชิงลำดับ. ตัวอย่างการประยุกต์ใช้โครงข่ายประสาทเวียนกลับ กับงานการรู้จำรูปแบบเชิงลำดับ เช่น การประมวลผลภาษาธรรมชาติ รวมไปถึงกลไกความลึก เนื่องจากเป็นเนื้อหาหลักที่อภิปรายในบทที่ 9.

## 5.1 ปัญหาการเลือนหายของเกรเดียนต์

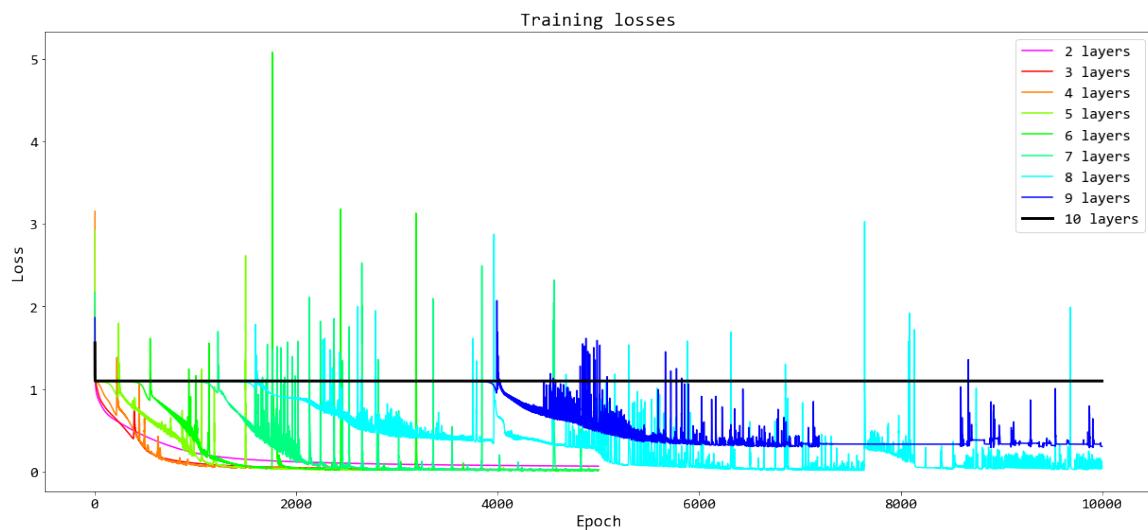
ปัญหาสำคัญที่ทำให้โครงข่ายลึกไม่ได้รับความนิยมในยุคต้นของพัฒนาการ คือ การฝึกโครงข่ายประสาทเทียมที่ลึกมากนั้นทำได้ยากมาก.

รูป 5.1 แสดงความก้าวหน้าของการฝึก เมื่อใช้ความลึกต่าง ๆ. แนวโน้มรวม ก็คือ ยิ่งความลึกมาก ดูเหมือนจะต้องการจำนวนสมัยฝึกที่มากขึ้น ค่าฟังก์ชันสูญเสียต่อสมัยของความลึกที่มากขึ้น ลุ่งที่จำนวนสมัยมากขึ้น. . สังเกต ความก้าวหน้าของการฝึกเมื่อใช้ความลึกสิบชั้น (สิบหนาสีดำ) ซึ่งมีลักษณะลุ่งราบรื่นร้อย แสดงถึงการฝึกที่สมบูรณ์ แต่ผลการฝึกได้คุณภาพการทำนายแย่มาก. นั่นคือ การฝึกเสร็จสิ้น แต่ฝึกไม่สำเร็จ ซึ่งยืนยันอย่างชัดเจนจากรูป 5.2 ที่สรุปค่าฟังก์ชันสูญเสียของการฝึก เมื่อใช้ความลึกต่าง ๆ.

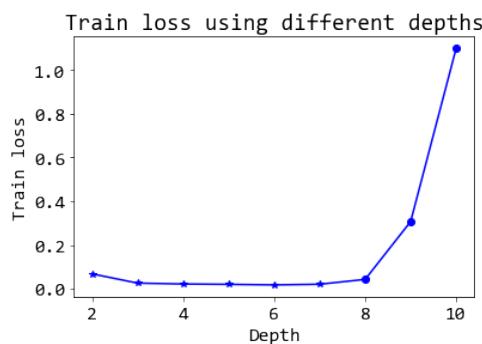
ปัญหาของการฝึกโครงข่ายลึกแบบนี้ ภายหลังพบว่า สาเหตุคือ ขนาดเกรเดียนต์ในชั้นต้น ๆ ของโครงข่ายที่ลึกมากเกินกว่าจะที่ปรับค่าน้ำหนักและใบอัสถีอย่างมีประสิทธิภาพ. รูป 5.3 แสดงตัวอย่างขนาดเฉลี่ยของเกรเดียนต์ที่ชั้นต่าง ๆ ของโครงข่ายประสาทเทียมสิบชั้น. รูป 5.4 สรุปค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ในชั้นต่าง ๆ. สังเกตว่า ขนาดเฉลี่ยเกรเดียนต์แต่ละชั้นแตกต่างกันอย่างมาก. (แบบฝึกหัด 5.1).

การฝึกโครงข่ายประสาทเทียม ก็คือการหาค่าน้ำหนักที่เหมาะสม ซึ่งแนวทางที่ใช้ก็คือใช้เกรเดียนต์ของฟังก์ชันจุดประสงค์ต่อค่าน้ำหนัก. แต่หากเกรเดียนต์มีขนาดเล็กมาก การหาค่าน้ำหนักที่เหมาะสมก็ทำได้ยาก และในหลายสถานการณ์ก็คือความล้มเหลวของการฝึกโครงข่าย. ปัญหาการฝึกโครงข่ายลึก ดังที่อภิปรายนี้ รู้จักกันในชื่อปัญหาการเลือนหายของเกรเดียนต์ (vanishing gradient problem). นั่นคือ ค่าเกรเดียนต์ที่คำนวณจากวิธีแพร์กระกระจายอิอยกกลับสำหรับค่าน้ำหนักในชั้นต้น ๆ (ใกล้อินพุต) จะมีขนาดเล็กลงมากจนแทบไม่สามารถปรับค่าน้ำหนักได้ ซึ่งส่งผลให้ การฝึกโครงข่ายลึกล้มเหลว.

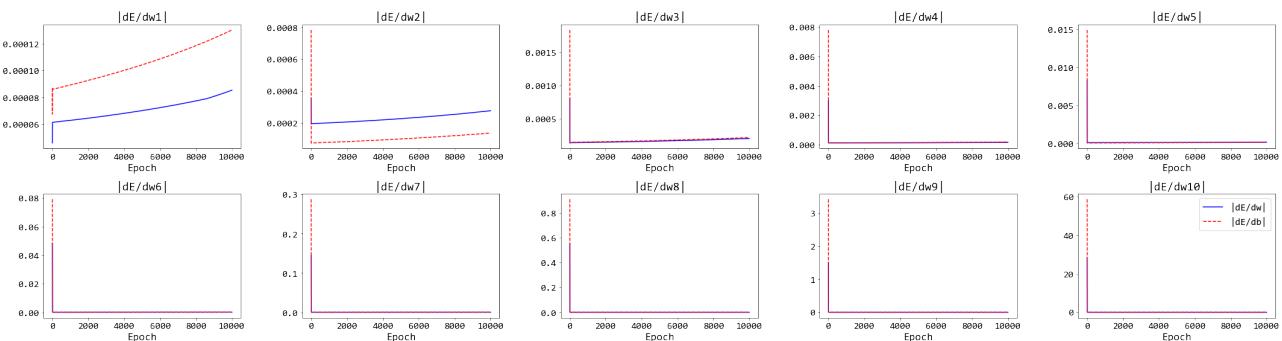
การเลือนหายของเกรเดียนต์เอง ก็พบว่าสาเหตุหลักมาจากการใช้ฟังก์ชันกระตันซิกมอยด์ ที่มีช่วงพลวัตรแคบ. เพื่อแก้ปัญหาช่วงพลวัตรของฟังก์ชันกระตันซิกมอยด์ ฟังก์ชันกระตันเรคติไฟฟ์ลีเนียร์ (rectified linear



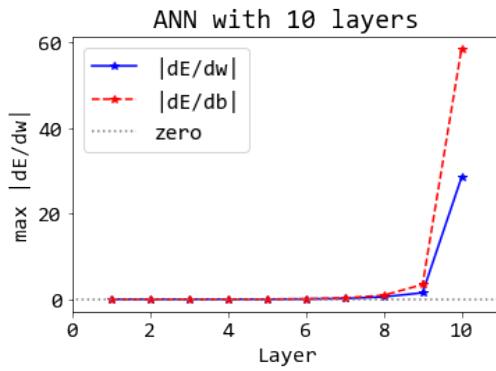
รูปที่ 5.1: ค่าฟังก์ชันสูญเสียต่อสมัยฝึก เมื่อใช้โครงข่ายประสาทเทียมที่ความลึกต่าง ๆ โดย ความลึกสองขั้นถึงเจ็ดขั้น โครงข่ายสามารถถูกฝึกได้ภายใน 5000 สมัย แต่ความลึกแปดขั้นและเก้าขั้น ต้องทำการฝึกถึง 10000 สมัย และที่ความลึกสิบขั้น (เส้นทึบหนาสีดำ) แม้ทำการฝึกไป 10000 สมัยแล้ว ซึ่งความก้าวหน้าของการฝึกก็ดูคล้ายการฝึกสมบูรณ์ แต่ได้ผลการฝึกที่แย่มาก.



รูปที่ 5.2: การฝึกโครงข่ายลึกล้มเหลว. ภาพ แสดงการสรุปค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึกที่ความลึกต่าง ๆ. สรุปค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึก หลังจากฝึกเสร็จ (5000 สมัยสำหรับโครงข่าย 2 ถึง 7 ขั้น และ 10000 สมัยสำหรับโครงข่าย 8 ถึง 10 ขั้น). ในขณะที่โครงข่ายที่ตื้นกว่าสามารถฝึกได้ดี แต่การฝึกโครงข่ายลึกกลับล้มเหลว.



รูปที่ 5.3: ปัญหาการเลื่อนหายของเกรเดียนต์. ตัวอย่างความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบขั้น. แต่ละภาพแสดงค่าเฉลี่ยเกรเดียนต์ของค่าน้ำหนัก (เส้นทึบสีฟ้า) และไบอส (เส้นประสีแดง) ของแต่ละขั้นคำนวน (ระบุเหนือภาพ) โดยเห็นอนเป็นสมัยฝึก.



รูปที่ 5.4: การเลือนหายของเกรเดียนต์ในโครงข่ายสิบชั้น. ภาพแสดงขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้นเปรียบเทียบกัน โดย แกน t แสดงค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้น และแกนบนระบุชั้นคำนวณ. เกรเดียนต์ของค่าน้ำหนัก แสดงด้วยเส้นทึบสีฟ้า. เกรเดียนต์ของค่าไบอัส แสดงด้วยเส้นประสีแดง. เส้นไปปลาสีเทา แสดงแนวค่าศูนย์. ขนาดเกรเดียนต์ตั้งแต่ชั้นที่หกขึ้นไปถึงชั้นที่หนึ่งมีค่าน้อยมาก ๆ (ใกล้ศูนย์) ซึ่งเป็นสาเหตุทำให้การฝึกล้มเหลว.

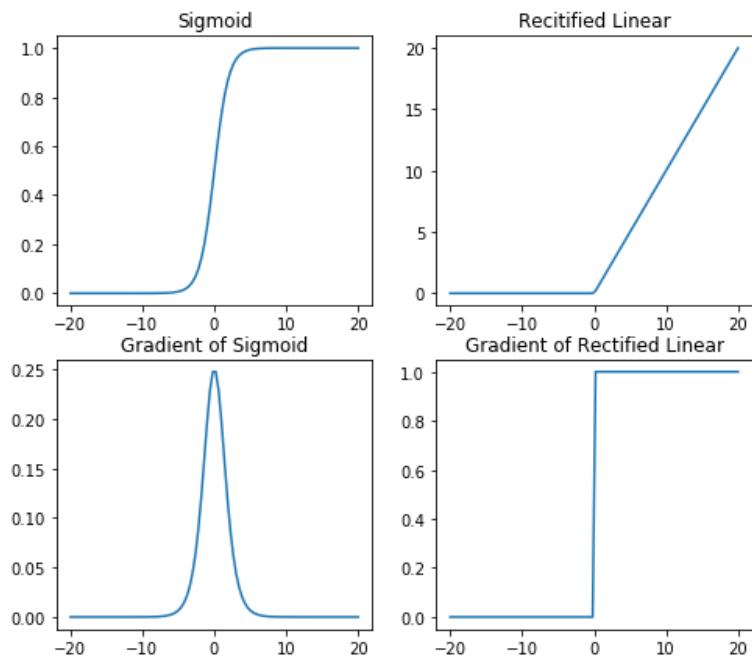
ซึ่งมักย่อว่า เรลู relu สำหรับ rectified linear unit[87] ถูกเสนอขึ้นมา. การเปลี่ยนไปใช้ฟังก์ชันกราฟตุน ที่ เป็นฟังก์ชันเรคติไฟฟ์ลิเนียร์ หรือเรลู เป็นปัจจัยที่สำคัญ ที่ช่วยให้การฝึกโครงข่ายประสาทเทียมใช้งานได้ ง่ายขึ้น และช่วยลดปัญหาการเลือนหายของเกรเดียนต์. สมการ 5.1 แสดงการคำนวณของฟังก์ชันกราฟตุนเรลู

$$\text{relu}(a) = \begin{cases} a & \text{เมื่อ } a \geq 0, \\ 0 & \text{เมื่อ } a < 0. \end{cases} \quad (5.1)$$

ซึ่งอนุพันธ์สามารถคำนวณได้จาก

$$\frac{d\text{relu}}{da} = \begin{cases} 1 & \text{เมื่อ } a \geq 0, \\ 0 & \text{เมื่อ } a < 0. \end{cases} \quad (5.2)$$

รูป 5.5 ภาพบนซ้ายแสดงการกราฟตุนของฟังก์ชันซิกมอยด์ เมื่อเปรียบเทียบกับการกราฟตุนของฟังก์ชันเรลู (ภาพบนขวา) และค่าอนุพันธ์ของฟังก์ชันซิกมอยด์ (ภาพล่างซ้าย) และค่าอนุพันธ์ของฟังก์ชันเรลู (ภาพล่างขวา). สังเกตว่า ค่าอนุพันธ์ของฟังก์ชันซิกมอยด์ จะมีช่วงพลวัตรอยู่ในบริเวณแคบ ๆ ใกล้ ๆ ศูนย์ ในขณะที่ ค่าอนุพันธ์ของฟังก์ชันเรลู จะมีช่วงพลวัตรครอบคลุมบริเวณที่มีค่าเป็นบวกทั้งหมด. การที่ค่าอนุพันธ์ของฟังก์ชันซิกมอยด์มีช่วงพลวัตรแคบ ทำให้การฝึกโครงข่ายประสาทเทียมที่ใช้ฟังก์ชันกราฟตุนเป็นซิกมอยด์ทำได้ยาก. ในที่นี้ ช่วงพลวัตร หมายถึง ช่วงบริเวณของอินพุตที่ค่าอนุพันธ์มีขนาดใหญ่. จากรูป 5.5 เราจะเห็นว่าค่าอนุพันธ์ของฟังก์ชันเรลู มีขนาดเป็นหนึ่ง ตลอดช่วงอินพุตที่มีค่าเป็นบวก เปรียบเทียบกับอนุพันธ์ของฟังก์ชันซิกมอยด์ ที่มีค่ามากกว่าศูนย์อย่างชัดเจน อยู่แค่บริเวณที่อินพุตมีค่าใกล้ ๆ ศูนย์เท่านั้น.



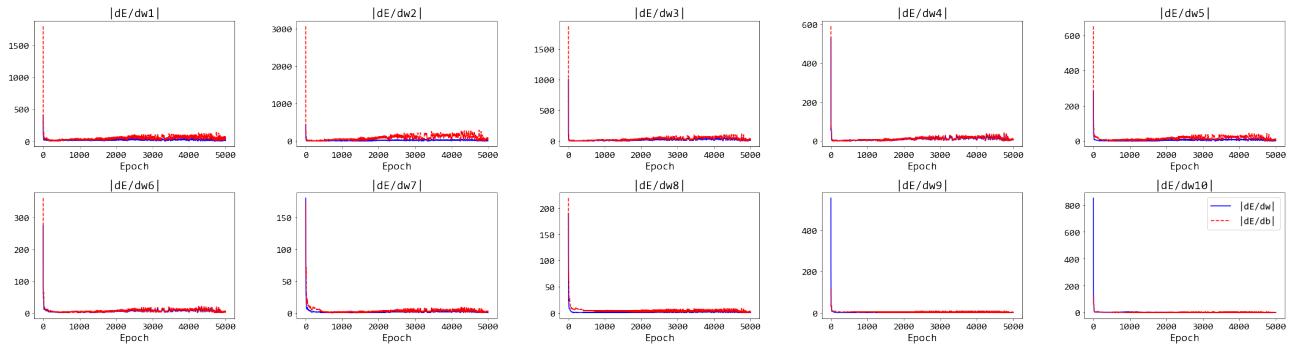
รูปที่ 5.5: พังก์ชันกราฟตุนซิกมอยด์ (ภาพบนขวา) พังก์ชันกราฟตุนเรลู (ภาพบนซ้าย) เกรเดียนต์ของพังก์ชันซิกมอยด์ (ภาพล่างขวา) เกรเดียนต์ของพังก์ชันเรลู (ภาพล่างซ้าย).

รูป 5.6 และ 5.7 แสดงให้เห็นว่า การเปลี่ยนพังก์ชันกราฟตุนจากซิกมอยด์มาเป็นเรลู ช่วยแก้ปัญหาการเลือนหายของเกรเดียนต์ได้อย่างชัดเจน. เปรียบเทียบรูป 5.4 (ใช้พังก์ชันกราฟตุนซิกมอยด์) กับรูป 5.7 (ใช้พังก์ชันกราฟตุนเรลู) ซึ่งทั้งคู่ เป็นโครงข่ายประสาทเทียมสิบชั้นเหมือนกัน เพียงแต่ใช้พังก์ชันกราฟตุนต่างกัน. จะเห็นว่า เมื่อใช้พังก์ชันกราฟตุนซิกมอยด์ ขนาดของเกรเดียนต์จะลดลงเรื่อยๆ จากชั้นสุดท้ายไปสู่ชั้นต้น. นอกจากขนาดเกรเดียนต์ลดลงเรื่อยๆ แล้ว ขนาดเกรเดียนต์ยังลดลงไปอยู่ในระดับใกล้ศูนย์ด้วย (เส้นประสีเทา แสดงแนวค่าศูนย์). ขณะที่ เมื่อใช้พังก์ชันกราฟตุนเรลู (รูป 5.7) นอกจาก จะไม่เห็นแนวโน้มการลดลงของเกรเดียนต์จากชั้นสุดท้ายไปชั้นต้นแล้ว (1) ขนาดเกรเดียนต์มีค่าใหญ่ขึ้นมาก และ (2) ขนาดเกรเดียนต์มีค่าอยู่ในระดับมากกว่าศูนย์อย่างเห็นได้ชัด (อยู่เหนือเส้นประสีเทา).

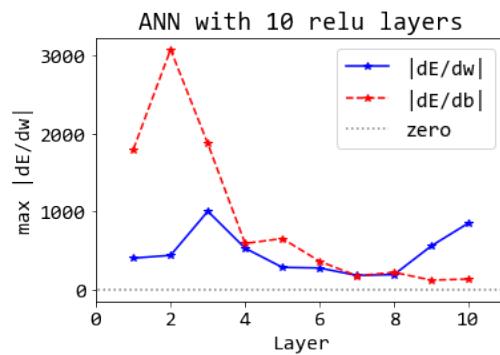
รูป 5.8 แสดงให้เห็นว่า เมื่อแก้ปัญหาการเลือนหายของเกรเดียนต์ได้ การฝึกโครงข่ายลึกสามารถทำได้ดีขึ้นมาก.

## 5.2 การฝึกทีละหมู่เล็ก

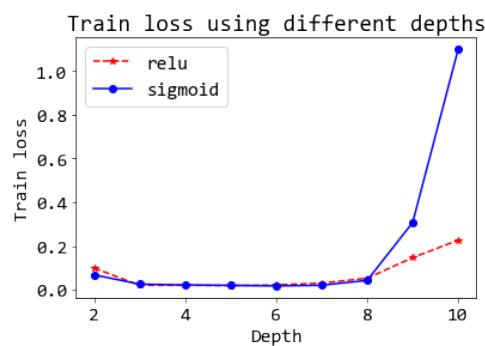
การมีข้อมูลจำนวนมาก แม้เป็นโอกาสที่ดี แต่ก็ต้องการกลไกที่ช่วยในการจัดการข้อมูลมหาศาล เพื่อการใช้งานหน่วยประมวลผลและหน่วยความจำได้อย่างมีประสิทธิภาพ. หนึ่งในกลไกที่สำคัญนั้น คือ การฝึกทีละหมู่เล็ก



รูปที่ 5.6: ตัวอย่างความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบชั้น เมื่อใช้ฟังก์ชันกราฟตุ้นrelu. แต่ละภาพแสดงค่าเฉลี่ยเกรเดียนต์ของค่าน้ำหนัก (เส้นทึบสีฟ้า) และไบอัส (เส้นประสีแดง) ของแต่ละชั้นคำนวน (ระบุเหนือภาพ) โดยแกนนอนเป็นสมัยฝึก.



รูปที่ 5.7: ขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้นเบรียบเทียบกัน โดย แกนตั้ง แสดงค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้น และแกนนอนระบุชั้นคำนวน. เกรเดียนต์ของค่าน้ำหนัก แสดงด้วยเส้นทึบสีฟ้า. เกรเดียนต์ของค่าไบอัส แสดงด้วยเส้นประสีแดง. การใช้ฟังก์ชันกราฟตุ้นrelu ช่วยให้ค่าเกรเดียนต์ของทั้งสิบชั้นมีขนาดค่าใหญ่พอสำหรับการปรับค่าน้ำหนักและเบ้อสอย่างมีประสิทธิภาพ.



รูปที่ 5.8: ค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึกที่ความลึกต่าง ๆ เบรียบเทียบเมื่อใช้ฟังก์ชันกราฟตุ้นซิกมอยด์กับเมื่อใช้ฟังก์ชันกราฟตุ้นrelu. ฟังก์ชันกราฟตุ้นreluช่วยให้การฝึกโครงข่ายลึกทำได้ดีขึ้นมาก.

(minibatch training).

บท 3 ได้อภิปรายถึง ทางเลือกในการฝึก โดย การฝึกที่ใช้ข้อมูลทั้งหมดในการปรับปรุงค่าน้ำหนักพร้อม ๆ กันที่เดียว ซึ่งเรียกว่า การฝึกแบบออฟไลน์ (offline training) หรือ การฝึกแบบหมุน และ การฝึกที่ใช้ข้อมูลที่จะจุดข้อมูลและปรับปรุงค่าน้ำหนักทีละครั้งสำหรับแต่ละจุดข้อมูล ซึ่งเรียกว่า การฝึกแบบออนไลน์ (online training) หรือ การฝึกแบบส่วนเพิ่ม (incremental mode). นอกจากนั้น ยังได้อภิปรายถึงข้อดีและข้อเสีย ต่าง ๆ ของทางเลือกทั้งสองนี้ นั่นคือ การฝึกแบบออฟไลน์ สามารถทำการคำนวณได้อย่างรวดเร็ว และยัง สามารถใช้การประมวลผลแบบขนาน (parallel processing) มาช่วยทำให้การคำนวณมีประสิทธิภาพมาก ขึ้นได้ แต่ข้อเสียคือ ต้องการหน่วยความจำมาก. ในขณะที่ ข้อดีของการฝึกแบบออนไลน์ นอกจგต้องการ ใช้หน่วยความจำปริมาณน้อยกว่า คือ เมื่อทำการคำนวณแล้ว ก็จะไม่ต้องเก็บข้อมูลทั้งหมดไว้ในเครื่อง จึงช่วยลดความ เสี่ยงในการเข้าไปติดอยู่ในค่าตัวที่ทำต่ำสุดท้องถิ่นได้ หรือ กล่าวอีกอย่างอาจช่วยปรับปรุงคุณภาพของการฝึก ได้. ประเด็นเรื่องการฝึกแบบออนไลน์ช่วยคุณภาพของการฝึกนี้ ถูกเพโลและคณะ[76] เสริมว่า การฝึกโดยใช้ ข้อมูลทีล่นน้อย ๆ อาจจะช่วยให้ผลคล้ายการทำเรกูลาราizer[219] ซึ่งอาจจะเพรำสัญญาณรบกวนที่ปนเข้ามา ในกระบวนการฝึก.

การฝึกทีละหมู่เล็ก (minibatch) เป็นเทคนิคที่ประณีประนอม ระหว่างแนวทางการฝึกแบบหมุน และการ ฝึกแบบออนไลน์ เพื่อใช้เวลาในการฝึกไม่นานเกินไป ใช้หน่วยความจำไม่มากเกินไป และได้คุณภาพการฝึกที่ดี. นั่นคือ การฝึกทีละหมู่เล็ก จะแบ่งข้อมูลออกเป็นกลุ่มเล็ก ๆ โดยที่ แต่ละกลุ่มนี้มีจำนวนข้อมูลมากกว่าหนึ่งจุด ข้อมูล แต่น้อยกว่าจำนวนข้อมูลฝึกทั้งหมด. การฝึกแต่ละหมู่ จะปรับค่าน้ำหนักและใบอัสสำหรับการคำนวณ กับแต่ละหมู่เล็กนี้ จนครบทุกหมู่. สมการ 5.3 แสดงการคำนวณของค่าน้ำหนักสำหรับแต่ละหมู่เล็ก (การคำ นวณใบอัสก์ทำได้ในลักษณะเดียวกัน)

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \alpha \frac{1}{|B_m|} \sum_{n \in B_m} \frac{\partial E_n}{\partial w_{ji}^{(l)}}, \quad (5.3)$$

สำหรับ  $m = 1, 2, \dots, \lceil \frac{N}{|B|} \rceil$  เมื่อ  $w_{ji}^{(l)}$  เป็นค่าน้ำหนักระหว่างหน่วย  $j$  ของชั้น  $l$  และหน่วย  $i$  ของชั้นก่อน หน้า. พจน์  $\frac{\partial E_n}{\partial w_{ji}^{(l)}}$  คือ ค่าอนุพันธ์ของฟังก์ชันจุดประสงค์ คำนวณจากจุดข้อมูลที่  $n^{th}$  จากจำนวนทั้งหมด  $N$  จุดข้อมูล. ตัวแปร  $m$  แทนดัชนีของหมู่เล็ก. สัญกรณ์  $|B_m|$  และ  $|B|$  แทน ขนาดของหมู่เล็กที่  $m^{th}$  และขนาด ของหมู่เล็กส่วนใหญ่ ตามลำดับ. ขนาดของหมู่เล็ก (batch size) คือ จำนวนจุดข้อมูลในหมู่เล็ก. เชต  $B_m$  เป็นเซตของดัชนีของจุดข้อมูลที่ถูกจัดอยู่ในหมู่เล็กที่  $m^{th}$  โดยดัชนีของจุดข้อมูล จะถูกสุ่มจัดเข้าหมู่เล็ก และ

หากจำนวนข้อมูลทั้งหมดไม่อาจแบ่งได้เท่า ๆ กันทุกหมู่เล็ก จะมีหมู่เล็กหนึ่งหมู่ที่มีจำนวนต่างจากหมู่อื่น ๆ (หรือหมู่เศษนี้อาจถูกตัดทิ้ง เพื่อประสิทธิภาพของการคำนวณ).

การทำลักษณะเช่นนี้ จะคล้ายกับการฝึกแบบหมู่ ในแต่ละการคำนวณกับหมู่เล็กจะเป็นการคำนวณกับจุดข้อมูลหลาย ๆ จุดพร้อม ๆ กัน และก็จะคล้ายกับการฝึกแบบออนไลน์ ในแต่ละว่า ค่าน้ำหนักจะถูกปรับหลาย ๆ ครั้งในหนึ่งสมัย (แต่ละครั้ง ปรับสำหรับแต่ละหมู่เล็ก).

กูดเพโลและคณะ<sup>[76]</sup> อภิปรายว่า ขนาดของหมู่เล็กที่เหมาะสม ขึ้นกับทั้งฮาร์ดแวร์และขั้นตอนวิธีการหาค่าดีที่สุดที่เลือกใช้ เช่น หากใช้การคำนวณด้วยหน่วยประมวลผลกราฟิกส์ (Graphics Processing Unit หรือ GPU) การเลือกขนาดหมู่เล็กเป็นจำนวนของสองยกกำลัง เช่น 16, 32, 64, 128, 256 จะช่วยให้ได้เวลาประมวลผลที่เร็ว. ขั้นตอนวิธีการหาค่าดีที่สุดที่ใช้แค่ค่าเกรเดียนต์ เช่น วิธีลงเกรเดียนต์ โดยทั่วไปแล้ว มักจะทบทานและสามารถได้งานได้กับขนาดต่าง ๆ ของหมู่เล็กได้.

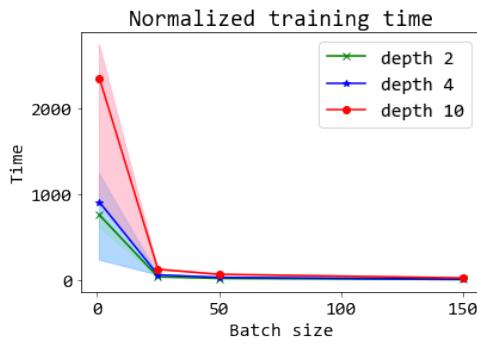
หมายเหตุ แต่ขั้นตอนวิธีการหาค่าดีที่สุดที่ใช้ทั้งค่าเกรเดียนต์และไฮเซียน<sup>2</sup> ต้องการขนาดหมู่เล็กที่ใหญ่พอที่จะสามารถประมาณค่าไฮเซียนได้อย่างมีประสิทธิภาพ เช่น[76] 10000.

รูป 5.9 แสดงเวลาที่ใช้ในการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ. หมายเหตุ การใช้ขนาดหมู่เล็กเป็นหนึ่ง เทียบเท่าการฝึกแบบออนไลน์ และการใช้ขนาดหมู่เล็กเท่ากับหรือมากกว่าจำนวนข้อมูล (ซึ่งในตัวอย่างแสดงในรูปคือ 150) เทียบเท่าการฝึกแบบหมู่. สังเกตว่า ขนาดหมู่ที่เล็กลง จะใช้เวลาในการฝึกนานขึ้น. แม้ว่า การใช้ขนาดหมู่ที่เล็กลง จะทำให้เวลาในการฝึกนานขึ้น แต่ขนาดหมู่ที่เล็กลง ทำให้ในการคำนวณทำงานกับเมทริกซ์ขนาดเล็กลงด้วย. การเลือกขนาดหมู่ เป็นเสมือนการหาสมดุลระหว่างเวลาฝึกและขนาดหน่วยความจำ.

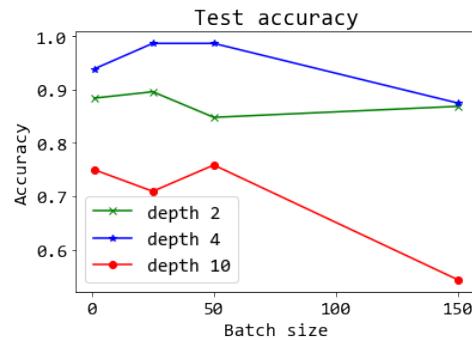
นอกจากนั้น ผลทางอ้อมของการฝึกหมู่เล็ก ยังช่วยให้คุณภาพการฝึกดีขึ้นได้ด้วย (ถ้าทำอย่างเหมาะสม) ดังแสดงในรูป 5.10. กูดเพโลและคณะ<sup>[76]</sup> อภิปรายว่า ขนาดหมู่เล็ก อาจจะช่วยให้ผลในเชิงการเรกูลาริซ หรือคุณความซับซ้อนของแบบจำลอง ช่วยลดโอกาสการโอเวอร์พิทข้อมูลลง ซึ่งอาจจะเป็น เพราะผลจากสัญญาณรบกวนจากการฝึกหมู่เล็ก ในกระบวนการหาค่าดีที่สุด. ในทางปฏิบัติ การใช้ขนาดหมู่ที่เล็กลง มักจะทำให้ต้องการจำนวนสมัยฝึกน้อยลง (ถึงแม้แต่สมัย อาจจะใช้เวลาฝึกนานขึ้น).

กลไกที่สำคัญในการฝึกหมู่เล็ก คือการสุ่มลำดับของข้อมูล. การสุ่มนี้อาจจะสุ่มครั้งเดียว และใช้ลำดับนั้นตลอด หรือจะสุ่มทุกสมัยฝึกก็ได้. ในทางปฏิบัติ กูดเพโลและคณะ<sup>[76]</sup> อภิปรายว่า ผลจากการสุ่มแค่ครั้งเดียว กับการสุ่มทุกสมัยฝึก ไม่ได้ต่างกันมาก แต่สำคัญมาก ๆ ที่ต้องทำการสุ่ม. รูป 5.11 ภาพชี้แจง แสดงเวลาในการ

<sup>2</sup>ไฮเซียน (Hessian) คือ เมทริกซ์ของอนุพันธ์อันดับที่สอง. ขั้นตอนวิธีการหาค่าดีที่สุดหลายอย่าง อาศัยไฮเซียน เพื่อเพิ่มประสิทธิภาพในการทำงาน. ตัวอย่างเช่น วิธีนิวตัน (Newton method) และ วิธีบีโเฟลีส (BFGS method) ที่สามารถทำงานได้อย่างรวดเร็ว แต่ต้องการเชิงลึก. ดู ซองและเชค<sup>[39]</sup> เพิ่มเติมสำหรับรายละเอียดของขั้นตอนวิธีการหาค่าดีที่สุดที่อาศัยไฮเซียน.



รูปที่ 5.9: ตัวอย่างเวลาการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ โครงข่ายขนาดสองชั้น สี่ชั้น และสิบชั้น แสดงด้วยสัญลักษณ์ ดังระบุในภาพ. พื้นที่สีเขียวอ่อน ฟ้าอ่อน และชมพู แสดงช่วงระหว่างค่ามากที่สุดและน้อยที่สุดของเวลาฝึกโครงข่ายสองชั้น สี่ชั้น และสิบชั้น ตามลำดับ. หมายเหตุ พื้นที่สีเขียวอ่อน อาจสังเกตได้ยากในภาพ เนื่องจากเวลาที่ทดสอบพบว่ามีความผันผวนต่ำ.

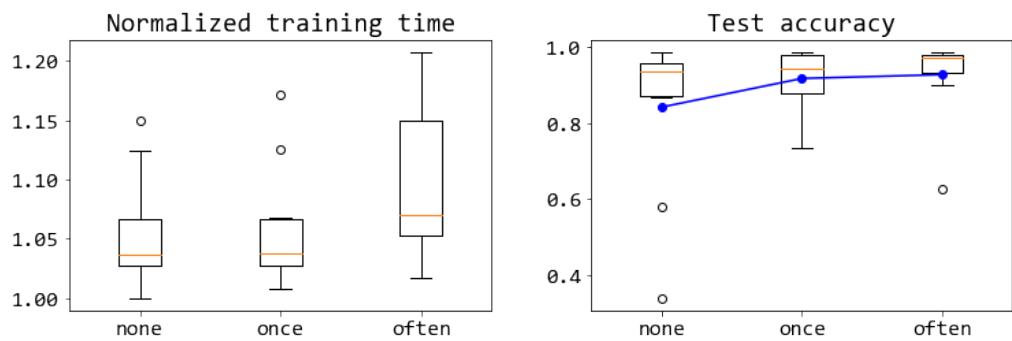


รูปที่ 5.10: ตัวอย่างคุณภาพการทำนายเมื่อใช้ขนาดหมู่เล็กต่าง ๆ ในด้านคุณภาพการทำนาย ผลอาจจะไม่ได้เป็นแนวทางได้อย่างชัดเจนมาก นอกจากการใช้ขนาดหมู่ที่เล็กลง โดยรวมแล้ว ช่วยเพิ่มคุณภาพการทำนายจากการฝึกแบบหมู่ (Batch size 150 ในภาพ). โครงข่ายขนาดสองชั้น สี่ชั้น และสิบชั้น แสดงด้วยสัญลักษณ์ ดังระบุในภาพ.

ฝึก เมื่อฝึกหมู่เล็กด้วยลำดับข้อมูลแบบต่าง ๆ ได้แก่ ไม่มีการสุ่มลำดับ, สุ่มลำดับครั้งเดียว, และสุ่มลำดับทุกสมัยฝึก. ภาพขวา แสดงผลความแม่นยำของการทำนาย. จากภาพ เวลาในการฝึกเมื่อทำการสุ่มครั้งเดียว ไม่ได้ต่างจากการไม่สุ่มมาก แต่การสุ่มทุกรอบฝึกมีผลในการเพิ่มเวลาฝึกอย่างชัดเจน ในขณะที่ คุณภาพของการฝึก (ความแม่นยำในการทำนาย) การสุ่มครั้งเดียว และการสุ่มทุกสมัย ไม่ได้ต่างกันมาก แต่มีผลตีกว่าการไม่สุ่มอย่างชัดเจน.

### 5.3 เทคนิคการตกอອก

เทคนิคการตกอອก (drop out[189]) เป็นกลไกสำหรับการทำเรกูลารีซ์สำหรับโครงข่ายประสาทเทียม โดยได้รับแรงบันดาลใจ จากการทำงานของโครงข่ายประสาททางชีววิทยา ที่ผลการทำงานเชือก็อ้อได้สูง ในขณะที่ เชลล์ประสาทต่าง ๆ ที่เป็นส่วนประกอบของโครงข่าย แต่ละเชลล์มีการทำงานที่เชือก็อ้อไม่ค่อยได้. นั่นคือ ใน



รูปที่ 5.11: ตัวอย่างผลจากการใช้วิธีจัดหมู่เล็กแบบต่าง ๆ ภาพซ้าย แผนภูมิกล่องแสดงเวลาที่ใช้ ภาพขวา แผนภูมิกล่องแสดงผลการทดสอบของแบบจำลองที่ฝึกแบบหมู่เล็ก โดยจัดหมู่เล็ก (1) ตามลำดับข้อมูล (ไม่มีการเปลี่ยนลำดับ **none**) (2) ตามการสุ่มลำดับโดยสุ่มครั้งเดียวและใช้ลำดับที่สุ่มน้ำาตลอดทุกสมัย (**once**) และ (3) ตามการสุ่มลำดับโดยสุ่นใหม่สำหรับแต่ละสมัย (**often**). จุดสีน้ำเงิน แสดงค่าเฉลี่ยความแม่นยำ.

ขณะที่ แต่ละเซลล์ บางครั้งอาจจะทำงาน บางครั้งอาจจะไม่ทำงาน แต่ด้วยการที่โครงข่ายมีเซลล์จำนวนมาก และการเชื่อมต่อได้เตรียมสำหรับความไม่แน่นอนนี้ไว้ ทำให้ผลโดยรวม ยังคงรักษาการทำงานที่เชื่อถือได้สูง.

สำหรับโครงข่ายประสาทเทียม การตอกออก สามารถทำได้โดยการสุ่มเลือกหน่วยคำนวณ ที่จะปิดการทำงาน ซึ่ง ทางการคำนวณ สามารถทำได้ง่ายๆ โดยคุณด้วยค่าศูนย์. ดังนั้น อาจมองได้ว่า การตอกออก เป็นเสมือน การใช้น้ำากา หรือค่าสัมประสิทธิ์ของการตอกออก  $m$  ไปคูณกับค่าหน่วยคำนวณ  $z$  โดย ค่าของ  $m$  สุ่มมาจากค่าศูนย์หรือหนึ่ง.

นั่นคือ ค่าหน่วยคำนวณหลังทำการตอกออก  $\tilde{z}$  คำนวณได้จาก  $\tilde{z} = m \cdot z$  เมื่อ  $m \sim \text{Bernoulli}(p)$  โดย  $\text{Bernoulli}(p)$  หมายถึง การแจกแจงแบบเบรนูลลี (Bernoulli distribution) ที่โอกาสที่ค่า  $m = 1$  คือ  $p$  นอกจากนั้น (โอกาส  $1 - p$ )  $m = 0$ . การคำนวณการตอกออก เขียนในรูปเวกเตอร์ได้เป็น

$$\tilde{\mathbf{z}} = \mathbf{m} \odot \mathbf{z} \quad (5.4)$$

เมื่อ หน้ากาก  $\mathbf{m}$  มีส่วนประกอบแต่ละตัวเป็นค่าที่สุ่มมาจากการหันหนึ่งหรือศูนย์ (การแจกแจงแบบเบรนูลลี).

การตอกออก จะให้ผลในลักษณะคล้ายกับการทำเรกูลาริซ์ นั่นคือ ช่วยคุณสมบัติความทั่วไปของแบบจำลอง. นอกจากนั้น ยังเชื่อว่า การตอกออก ยังช่วยเพิ่มความยืดหยุ่น ความทนทานในการเชื่อมต่อ ในลักษณะที่ช่วยลดการพึงพาคุณลักษณะที่สำคัญไม่เกี่ยวย่างลง และเพิ่มโอกาสที่ทำให้แบบจำลองได้เรียนรู้คุณลักษณะที่สำคัญต่าง ๆ ของรูปแบบได้ครบถ้วนมากขึ้น. การสุ่มปิดการทำงานของหน่วยคำนวณ เชื่อว่า น่าจะช่วยหยุดการปรับตัวร่วมกัน (break co-adaptation) และน่าจะส่งผลให้เกิดความหลากหลายในการเชื่อมต่อมากขึ้น. และเพิ่มความหลากหลายในการเชื่อมต่อ สัมพันธ์โดยตรงกับความยืดหยุ่นกับความทนทานของระบบโดย

รวม จึงเชื่อว่า การตอกออก จะช่วยให้แบบจำลองมีความยืดหยุ่น และทนทาน(ต่ออินพุตที่หลากหลาย)ได้ดีขึ้น.

**การใช้งานการตอกออก.** การสุมปิดการทำงาน มักจะใช้เฉพาะตอนฝึกเท่านั้น. ตอนใช้งานอนุมาน จะเปิดการทำงานของทุกส่วน. เนื่องจาก หากฝึกได้ดีพอ ส่วนย่อยต่าง ๆ ในโครงข่ายจะทำงานได้ดีพอสมควร ดังนั้น หากเปิดทุกส่วนหมดพร้อมกัน จะต้องทำการซัดเซย เพื่อไม่ให้อาร์พุตที่ได้มีค่ามากเกินไป. การซัดเซยนี้ มักถูกเรียกว่า การปรับส่วนค่าน้ำหนัก (weight scaling). นั่นคือ หากสูงด้วยความน่าจะเป็น 0.5 หมายถึง โดยประมาณ หน่วยคำนวณต่าง ๆ ในโครงข่ายจะทำงานแค่ครึ่งเดียว. ถ้าหากเปิดทุกหน่วยคำนวณพร้อม ๆ กัน จะต้องซัดเซยด้วยการลดความแรงของค่าหน่วยคำนวณลงครึ่งหนึ่ง. และในกรณีทั่ว ๆ ไป สำหรับความน่าจะเป็นของการคงอยู่  $p$  และ ค่าหน่วยคำนวณ

$$\mathbf{z}' = p \cdot \mathbf{z} \quad (5.5)$$

เมื่อ  $\mathbf{z}'$  คือค่าหน่วยคำนวณหลังการปรับส่วนค่าน้ำหนัก และ  $p$  เป็นค่าความน่าจะเป็นของการคงอยู่ และ  $\mathbf{z}$  คือค่าหน่วยคำนวณ (ก่อนการปรับส่วนค่าน้ำหนัก).

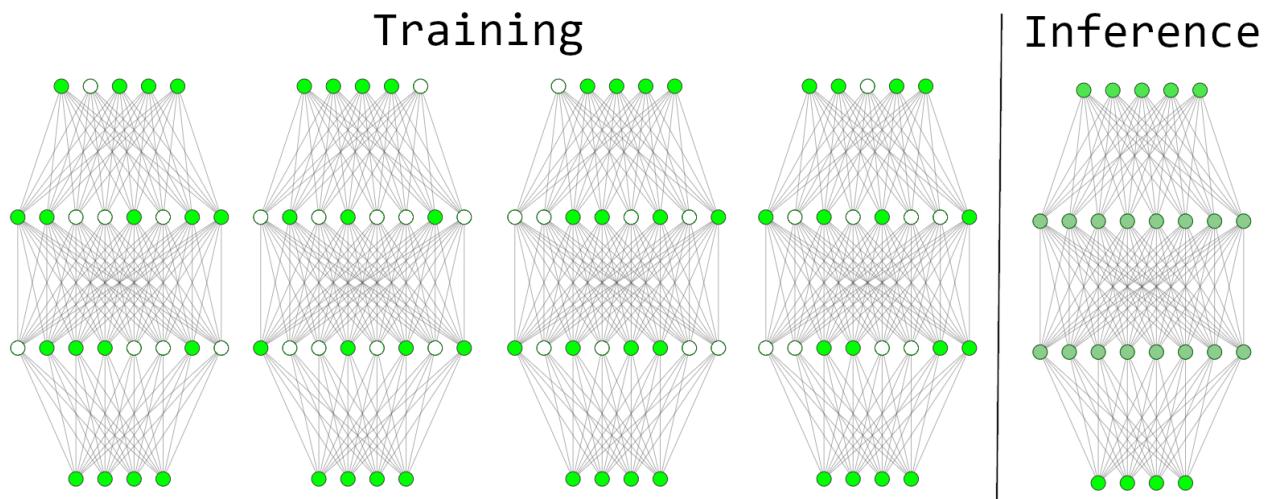
รูป 5.12 แสดงภาพประกอบแนวคิดของการตอกออก. ขณะฝึก จะมีหน่วยคำนวณบางส่วนถูกปิดไป และ ส่งผลเสมือนว่า กำลังใช้งานโครงข่ายอยู่โดยอุบัติ โดยที่โครงข่ายย่อยจะเปลี่ยนไปแบบสุ่มในการคำนวณเกรเดียนต์ แต่ละครั้ง. ขณะใช้งาน ทุกหน่วยคำนวณจะทำงานพร้อมกัน ดังนั้น เพื่อไม่ให้อาร์พุตมีค่ามากเกินไป ค่าหน่วยคำนวนจะถูกปรับขนาดลงอย่างเหมาะสม. สังเกตว่า การตอกออก จะไม่ทำกับชั้นเอาร์พุต.

ในทางปฏิบัติ การคำนวนค่าอนุมาน โดยซัดเซยการตอกออกที่ทำในขณะฝึก ค่อนข้างเทอะทะ และทำให้ การใช้งานแบบจำลองต้องระมัดระวังมากในการนำค่าน้ำหนักที่ฝึกแล้วมาใช้. นั่นคือ หากระหว่างการฝึก ไม่ได้ใช้การตอกออก ก็ต้องไม่ทำการปรับส่วนค่าน้ำหนัก แต่หากระหว่างการฝึก ทำการตอกออก ก็ต้องทำการปรับส่วนค่าน้ำหนักและปรับส่วนค่าน้ำหนักด้วยค่า  $p$  ที่ใช้ (และแต่ละชั้นคำนวนสามารถทำการตอกออก ด้วยค่า  $p$  ที่ต่างกันได้). ดังนั้น การนำค่าน้ำหนักที่ฝึกจากหลาย ๆ วิธีมาใช้ จะค่อนข้างบุกยากและมีความเสี่ยงมาก รวมถึงยังจำกัดการฝึกด้วยว่า หากฝึกด้วยการตอกออก และต้องทำการตอกออก แล้วต้องทำการตอกออกทุกสมัยฝึก และใช้ค่า  $p$  เท่าเดิมตลอด.

เพื่อลดปัญหาดังกล่าว การทำการตอกออก จึงอาจเลือกทำ

$$\tilde{\mathbf{z}} = \frac{1}{p} \cdot \mathbf{m} \odot \mathbf{z} \quad (5.6)$$

ขณะฝึก และไม่ต้องทำการปรับส่วนค่าน้ำหนัก ขณะใช้งานอนุมาน (นั่นคือ  $\mathbf{z}' = \mathbf{z}$ ). แนวทางนี้ ยืดหยุ่น สะดวก และลดความเสี่ยงของการซัดเซยผิดลง. การใช้  $\frac{1}{p}$  (ซึ่งมากกว่าหนึ่ง) เทียบเท่าการขยาย



**รูปที่ 5.12:** ภาพประกอบแสดงแนวคิดของการตกลอก. ภาพแสดงโครงข่ายประสาทเทียมสามชั้น อินพุต (ห้ามบิต) อยู่ด้านบน และเอาต์พุต (สีมิติ) อยู่ด้านล่าง ชั้นช่องทั้งสองชั้น แต่ละชั้นมีจำนวนหน่วยช่องแปดหน่วย. การฝึกโครงข่าย ใช้การตกลอกโดย ความนำ จะเป็นของการคงอยู่เป็น 0.8 สำหรับชั้นอินพุต และ 0.5 สำหรับชั้นช่องทั้งสอง. สีโครงข่ายทางซ้าย แสดงตัวอย่างของการทำงาน ของโครงข่าย ขณะฝึก ซึ่งแต่ละโครงข่าย สำหรับการคำนวณเกรเดียนต์แต่ละครั้ง. แต่ละครั้ง การตกลอกจะสูม และให้ผลเป็นหน่วย คำนวณที่คงอยู่ต่าง ๆ กันไป. วงกลมสีเขียว แสดงหน่วยที่ทำงาน และวงกลมสีขาว แสดงหน่วยที่ถูกปิด. โครงข่ายทางขวา แสดง การทำงานของโครงข่าย ขณะใช้งานอนุมาน. เมื่อใช้งาน ทุกหน่วยคำนวณจะทำงานพร้อม ๆ กันทั้งหมด แต่ค่าของหน่วยคำนวณจะ ถูกลดความแรงลง (ทำการปรับส่วนค่าน้ำหนัก) เพื่อไม่ให้อาต์พุตมีค่ามากเกินไป. สีของวงกลมที่ต่างไป สะท้อนการปรับค่าลงของ หน่วยคำนวณ โดยชั้นอินพุตปรับลงเป็น 0.8 เท่า และชั้นช่องปรับลงเป็น 0.5 เท่า.

ขนาดของหน่วยคำนวณ ขณะทำการตกลอก ซึ่งจะบังคับให้แบบจำลองเรียนรู้ค่าน้ำหนัก ที่จะไม่ทำให้อาต์พุต มีค่ามากเกินไป ในอัตราส่วนที่สัมพันธ์กับโอกาสการคงอยู่. ดังนั้น ค่าน้ำหนักที่ได้จึงสามารถนำไปใช้งานได้เลย โดยไม่ต้องทำการปรับส่วนค่าน้ำหนักอีก.

ประโยชน์ของการตกลอก ยังถูกมองว่า เป็นเพิ่มความน่าเชื่อถือได้ของการอนุมาน ในลักษณะคล้าย แนวทางการจัดถุง (bagging). แนวทางการจัดถุง เป็นหนึ่งในแนวทางหลัก ของการประสานการเรียนรู้ (ensemble learning). การประสานการเรียนรู้ เป็นเทคนิคของการเรียนรู้ของเครื่อง เพื่อปรับปรุงคุณภาพ การอนุมาน โดยใช้ค่าท่านายจากหลาย ๆ แบบจำลอง และนำค่าท่านายต่าง ๆ เหล่านี้มาสรุปรวมเป็น ค่า ท่านายของการประสานการเรียนรู้. วิธีการสรุปอาจทำได้หลายแบบ ขึ้นกับภารกิจการทำนาย เช่น หากเป็น การทำนายค่าตัดตอน (อาต์พุต  $y \in \mathbb{R}$ ) จะใช้ค่าเฉลี่ยจากค่าท่านายของแบบจำลองต่าง ๆ. แต่หากเป็น การจำแนกกลุ่ม (อาต์พุต  $y \in \{1, \dots, K\}$  เมื่อ  $K$  เป็นจำนวนกลุ่ม) จะสรุปโดยการลงคะแนนเสียง (vote) นั่น คือ การใช้ค่าฐานนิยม หรือสรุปเป็นกลุ่มที่ถูกจำแนกมากที่สุด (ซึ่งอาจต้องการกลยุทธ์ในการจัดการกับกรณี เสมอกัน).

ในขณะที่ การประสานการเรียนรู้ เป็นเทคนิคแนวทางกว้างๆ ที่เน้นการนำผลทำนายจากหลาย ๆ แบบ

จำลอง มาสรุปร่วมกัน. แนวทางการจัดถุง เป็นแนวทางการเตรียมแบบจำลองต่างๆ สำหรับใช้ในการประมาณการเรียนรู้.

แบบจำลองต่างๆ ที่กล่าวถึงในการประมาณการเรียนรู้ หมายถึง ฟังก์ชันการทำงานใดๆ ที่สร้างมาต่างกัน อาจจะโดยมีโครงสร้างทางคณิตศาสตร์ที่ต่างกัน (เช่น โครงข่ายประสาทเทียมสามชั้น กับโครงข่ายประสาทเทียมห้าชั้น หรือโครงข่ายประสาทเทียม กับชั้พพอร์ตเวกเตอร์แมชชีน) หรืออาจจะโดยมีโครงสร้างทางคณิตศาสตร์ที่เหมือนกัน แต่ผ่านกระบวนการฝึกที่ต่างกัน เช่น ใช้ข้อมูลในการฝึกที่ต่างกัน หรือต่างกันทั้งโครงสร้างทางคณิตศาสตร์และกระบวนการฝึก.

แนวทางการจัดถุง เน้นการเตรียมแบบจำลองที่ต่างกัน ด้วยการใช้ข้อมูลฝึกที่ต่างกัน นั่นคือ หากต้องการเตรียม  $M$  แบบจำลองสำหรับใช้ในการประมาณการเรียนรู้ จากชุดข้อมูลฝึกที่มีจำนวนจุดข้อมูลเป็น  $N$  แนวทางการจัดถุง จะสร้างข้อมูลสำหรับฝึกขึ้นมา  $M$  ชุด โดยแต่ละชุด จะสุมจุดข้อมูลจากชุดฝึกมาแบบหยิบคืน (sample with replacement) โดยจำนวนข้อมูลในแต่ละชุด  $N'$  เป็นอภิมานพารามิเตอร์ของการจัดถุง. จากนั้น แบบจำลองแต่ละตัว จะถูกฝึกกับข้อมูลที่สร้างขึ้นแต่ละชุด และแบบจำลองทั้งหมดที่ฝึกเสร็จ ก็จะสามารถนำไปใช้ในการประมาณการเรียนรู้ได้.

การตกลอก ถูกมองว่า เป็นกลไกในลักษณะคล้ายแนวทางการจัดถุง จากการที่ ขณะฝึก การปรับค่าน้ำหนักแต่ละครั้ง จะมีเฉพาะบางส่วนของโครงข่ายเท่านั้นที่จะถูกปรับค่า ส่วนที่ถูกปิดการทำงาน จะไม่ได้ถูกปรับค่าน้ำหนัก. ดังนั้น เมื่อใช้งาน และเปิดการทำงานของทุกส่วน จึงคล้ายการประสาทการเรียนรู้ของส่วนย่อยต่างๆ ภายในโครงข่าย. แต่การตกลอก ก็ไม่ได้ทำให้การใช้งานโครงข่ายประสาทเทียมเหมือนการประมาณการเรียนรู้แบบดั้งเดิม. เพราะว่า โดยทั่วไปแล้ว ส่วนต่างๆ ของโครงข่ายที่เปิดและปิดขณะฝึกจากกลไกของการตกลอก จะมีการซ้อนทับกันอยู่มาก ซึ่งต่างจาก การประมาณการเรียนรู้แบบดั้งเดิม ที่แต่ละแบบจำลองมีความเป็นอิสระต่อกันสูงกว่ามาก. อย่างไรก็ตาม ด้วยเหตุผลดังอภิรายนี้ ในมุมมองหนึ่ง การตกลอกถูกตีความว่า นำจะช่วยปรับปรุงคุณภาพของแบบจำลอง ได้จากการที่ให้ผลในลักษณะของการประมาณการเรียนรู้.

โดยทั่วไป การตกลอก นิยมใช้ความน่าจะเป็นของการคงอยู่  $p$  เป็น 0.8 สำหรับชั้นอินพุต และ 0.5 สำหรับชั้นช่อง และการตกลอก จะใช้งานได้กับแบบจำลองที่มีขนาดใหญ่พอ (ความซับซ้อนมากเพียงพอ). แต่การตกลอก อาจทำให้การฝึกทำได้ช้าลง. การศึกษาของศรีวิสาทาวาและคณะ[189] รายงานว่า การตกลอก ให้ผลช่วยการฝึกได้ดีกว่าการทำเรกูลาริซ์舍弃 ฯ วิธี รวมถึง วิธีค่าน้ำหนักเลื่อน. ข้อเสียของการใช้การตกลอกที่สำคัญ ก็เช่น[76] อาจทำให้การฝึกทำได้ช้าลง และอาจทำให้ต้องการแบบจำลองที่ใหญ่ขึ้น. ลักษณะ

เดียวกับการทำReLU การตกลอก อาจไม่ได้ช่วยมาก หากข้อมูลที่ฝึกมีปริมาณมาก ซึ่งประโยชน์ที่ได้จากการทำการตกลอก อาจจะน้อยกว่าข้อเสียที่จะทำให้ต้องการแบบจำลองใหญ่ขึ้น และทำให้การฝึกช้าลง.

นอกจากเทคนิคการตกลอกแล้ว เทคนิคการตกลอก ยังเป็นแรงบันดาลใจให้เกิดการพัฒนาเทคนิคอื่น ๆ ที่คล้าย ๆ กันจำนวนมาก เช่น การตกลอกเร็ว (fast drop out[209]), การส่งเสริมการตกลอก (dropout boosting[212]), การเชื่อมตกลอก (DropConnect[207]). อย่างไรก็ตาม ด้วยผลลัพธ์การทำงาน ประสิทธิภาพ และความสะดวกของการใช้งาน การตกลอก เป็นแนวทางที่ได้รับความนิยมสูงกว่าริทึ่พัฒนาต่อ ๆ ขึ้นมา เหล่านี้. ภูดเฟโลและคณะ[76] อภิปรายสรุปว่า กลไกสำคัญที่เทคนิคการตกลอกเป็นตัวแทน คือ การใส่ความไม่แน่นอนเข้าไปในการฝึกโครงข่าย และทำการอนุமานโดยสรุปจากผลต่าง ๆ ที่ผ่านความไม่แน่นอน ซึ่งในผลในลักษณะการการจัดดู โดยมีการใช้พารามิเตอร์ร่วมกัน. การใส่ความไม่แน่นอน เข้าไปจะช่วยให้แบบจำลองเรียนรู้ที่จะยึดหยุ่นขึ้น และครอบคลุมขึ้น ซึ่งอาจจะคล้ายกับคน ที่เรียนรู้ที่จะยึดหยุ่นขึ้นและรอบคอบขึ้น เมื่อคำนึงความไม่แน่นอนที่อาจเกิดขึ้น. ศรีวิสาทาวาและคณะ[189] ได้ทดลองใช้หน้ากากค่าจริง  $\mathbf{m} \sim \mathcal{N}(\mathbf{1}, \mathbf{I})$  (การแจกแจงปกติ ที่มีค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานเป็นหนึ่ง ไม่มีสหลัมพันธ์ระหว่างตัวแปร) แทนการแจกแจงเบรนูลลี่ (สมการ 5.4) และพบว่า ได้ผลการทำงานที่ดีเช่นกัน นอกจากนั้น หน้ากากค่าจริงนี้ มีค่าคาดหมาย  $E[\mathbf{m}] = \mathbf{1}$  จึงไม่ต้องทำการปรับส่วนค่า'n้ำหนัก.

## 5.4 การกำหนดค่า'n้ำหนักเริ่มต้น

การฝึกโครงข่ายประสาทเทียมแบบลึก ใช้ขั้นตอนวิธีการหาค่า'dีที่สุดที่อาศัยเกรเดียนต์. ค่าเริ่มต้นของตัวแปร มีผลอย่างมากต่อการทำงานการหาค่า'dีที่สุด เช่นการเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์พอดี จะช่วยทำให้การฝึกโครงข่ายทำได้ง่ายและเร็วขึ้น. ในขณะที่การเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์เปลี่ยนแปลงอย่างรุนแรง อาจนำไปสู่ปัญหาเสถียรภาพของการฝึก หรือหากเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์มีค่าน้อยมาก (มักอ้างถึง ด้วยคำว่า “ทีราบ” หรือ plateau) อาจทำให้การฝึกไม่ก้าวหน้า หรือหยุดชะงักได้.

เนื่องจากความเข้าใจในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมยังไม่กระจ่างชัดสมบูรณ์ ปัจจัยต่าง ๆ ของการฝึก รวมถึงการกำหนดค่า'n้ำหนักเริ่มต้น จึงยังไม่มีข้อสรุปที่แน่นชัด. นอกจาก สิ่งหนึ่งที่จำเป็น คือ ค่า'n้ำหนักเริ่มต้น ต้องช่วยลดการปรับตัวไปเหมือน ๆ กัน (มักอ้างถึงเป็น break symmetry).

การปรับตัวไปเหมือน ๆ กัน มาจากการที่โครงข่ายประสาทเทียมมีวิธีการคำนวณแต่ละหน่วยคำนวณ เหมือน ๆ กัน. ดังนั้นการที่แต่ละหน่วยคำนวณเริ่มต้นด้วยค่าเดียวกัน จะทำให้มันมีค่าเกรเดียนต์เท่ากัน และถูกปรับค่าไปเท่า ๆ กัน จนสุดท้าย แต่ละหน่วยคำนวณจะทำงานเหมือนกัน ตอบสนองกับรูปแบบอย่างเดียวกัน

ไม่ได้แยกกันรับผิดชอบแต่ละรูปแบบย่อย ๆ ทำให้ความสามารถโดยรวมของโครงข่าย ที่แม้จะมีจำนวนหน่วยคำนวณมาก แต่ให้ประสิทธิผลการทำงานเหมือนโครงข่ายที่มีหน่วยคำนวณน้อย (หรือ ในกรณีสุดโต่ง อาจทำงานเหมือนมีหน่วยคำนวณเดียว).

นอกจากลดการปรับตัวไปเหมือนกัน อีกปัจจัยหนึ่งที่สำคัญในการกำหนดค่าเริ่มต้น คือ ขนาดของค่าน้ำหนักไม่ควรจะมากเกินไป จนผลต่อเนื่อง ให้เกรเดียนต์มีค่าน้อย (ฝึกยาก) หรือมากเกินไป (การคำนวณขาดเสียรูปภาพ). แนวปฏิบัติคือ การใช้การสุ่มค่า เพื่อกำหนดค่าน้ำหนักเริ่มต้น. การกำหนดค่าน้ำหนักเริ่มต้นด้วยการสุ่มจากการแจกแจงที่มี exon โทรปีสูง ( เช่น การแจกแจงเอกรูป ) สามารถทำได้ง่ายๆ และไม่มีโอกาสสูงมาก ที่ค่าน้ำหนักจะไปเริ่มต้นที่เดียวกัน แม้ว่าจะมีพารามิเตอร์ค่าน้ำหนักจำนวนมาก.

กูดเฟโลและคณะ[76] อภิปรายว่า แทนที่การสุ่ม เราอาจจะคำนวณหาชุดค่าน้ำหนัก ที่แต่ละชุดแตกต่างกันมาก ๆ ได้ เช่น ในกรณีที่เหมาะสม อาจใช้ชั้นตอนวิธีแกรมช์มิตด์<sup>3</sup> แต่แนวทางนี้ มักจะเพิ่มภาระการคำนวณก่อนการฝึกขึ้นมาก และภาระการคำนวณก่อนการฝึกที่เพิ่มขึ้นมากนี้ อาจไม่คุ้มกับผลประโยชน์ที่ช่วยลดภาระการคำนวณระหว่างการฝึกลง เมื่อเปรียบเทียบกับการใช้แนวทางการสุ่ม.

ค่าพารามิเตอร์น้ำหนัก  $\mathbf{w}$  นิยมกำหนดค่าเริ่มต้นด้วยการสุ่ม ส่วนค่าพารามิเตอร์ใบอัส  $\mathbf{b}$  จะกำหนดค่าเริ่มต้นเป็นค่าคงที่ หรืออาจจะสุ่มค่าเซ็นเดียวกันก็ได้[76]. การสุ่มค่าน้ำหนัก มักนิยมสุ่มจากการแจกแจงเอกรูป หรือการแจกแจงปกติ. ค่าการแจกแจงที่นิยม[74] คือ กำหนดค่าเริ่มต้นน้ำหนักระหว่างการแจกแจงเอกรูป  $\mathcal{U}\left(-\frac{1}{\sqrt{m_i}}, \frac{1}{\sqrt{m_i}}\right)$  เมื่อ  $m_i$  เป็นจำนวนอินพุตของชั้นคำนวณ (อาจอ้างถึงว่าเป็น “จำนวนแผ่เข้า” หรือ a number of fan-in units). ค่า  $\frac{1}{\sqrt{m_i}}$  เพื่อป้องกันไม่ให้ผลคำนวณมีค่าใหญ่เกินไป จนผลเสียต่อเสียรูปของ การฝึก สำหรับโครงข่ายขนาดใหญ่.

ตัวอย่างเช่น หากชั้นคำนวณ ทำ  $\mathbf{a} = \mathbf{w}^T \mathbf{x} + \mathbf{b}$  กับ  $\mathbf{z} = h(\mathbf{a})$  เมื่อ  $h$  เป็นฟังก์ชันกระตุ้น และอินพุตของชั้น  $\mathbf{x} \in \mathbb{R}^{m_i}$ . ค่าน้ำหนักของชั้น  $\mathbf{w} \in \mathbb{R}^{m_o \times m_i}$ . และ ค่าเริ่มต้นของ  $\mathbf{w}$  กำหนดโดย

$$w_{kj} \sim \mathcal{U}\left(-\frac{1}{\sqrt{m_i}}, \frac{1}{\sqrt{m_i}}\right) \quad (5.7)$$

เมื่อ  $w_{kj}$  คือค่าน้ำหนักแต่ละค่า โดย  $k = 1, \dots, m_o$  และ  $j = 1, \dots, m_i$ .

อย่างไรก็ตาม เชเวียร์ โกลโลร์ต และโยชัว เบนจิโจ[74] ศึกษาความยากของการฝึกโครงข่ายประสาทเทียม ตีความผลที่ได้ และเมื่อประกอบกับผลงานศึกษาของแบรดลีย์[22] ที่พบว่าความแปรปรวน (variance) ของค่าเกรเดียนต์ที่แพร์เซอร์จายย้อนกลับ ลดลงเรื่อย ๆ ตามชั้นที่ย้อนกลับ ทั้งคุ้นเคยนิยฐานว่า หากความแปร-

<sup>3</sup>ชั้นตอนวิธีแกรมช์มิตด์ (Gram-Schmidt algorithm) เป็นวิธีคำนวณหาเวกเตอร์ที่ตั้งฉากกับเวกเตอร์ที่กำหนด. ดู [39] เพิ่มเติมสำหรับรายละเอียด.

ปรวน ของผลการกระตุ้น  $Z$  และความแปรปรวนค่าเกรเดียโนต์ ของแต่ละชั้นคำนวณมีค่าพอ ๆ กัน จะช่วยให้สารสนเทศไหลผ่านได้ดีขึ้น และจะช่วยให้การฝึกโครงข่ายทำได้สะดวกขึ้น. จากข้อสันนิษฐานดังกล่าว ทั้งคู่-เคราะห์ความแปรปรวนของของแต่ละชั้นคำนวณโดยประมาณ (อาศัยสมมติฐานหลายอย่าง รวมถึงสมมติฐานเชิงเส้น) และเสนอว่า กำหนดค่าเริ่มต้นสำหรับค่าน้ำหนัก โดยให้

$$\forall l, \text{var}[\mathbf{w}^{(l)}] = \frac{2}{m_i^{(l)} + m_o^{(l)}} \quad (5.8)$$

เมื่อ  $\text{var}[\mathbf{w}^{(l)}]$  คือความแปรปรวนของค่าน้ำหนักชั้นคำนวณที่  $l^{th}$  และ  $m_i^{(l)}$  คือจำนวนแผ่นเข้า และ  $m_o^{(l)}$  คือจำนวนหน่วยคำนวณในชั้น (หรือจำนวนเอาร์พุตของชั้นคำนวณ ที่อาจอ้างถึงเป็น “จำนวนแผ่ออก” หรือ a number of fan-out units). เมื่อนำเงื่อนไขนี้ไปใช้กับการแจกแจงเอกรูป จะได้ว่า

$$w_{kj} \sim \mathcal{U}\left(-\sqrt{\frac{6}{m_i + m_o}}, \sqrt{\frac{6}{m_i + m_o}}\right). \quad (5.9)$$

การกำหนดค่าน้ำหนักด้วยนิพจน์ 5.9 นิยม เรียกว่า การกำหนดค่าน้ำหนักด้วยวิธีเซเวียร์ (Xavier weight initialization).

การวิเคราะห์ของโกลโลร์และเบนจิโอ[74] คิดจากฟังก์ชันกระตุ้นไฮเปอร์บอลิกแทนเจนต์ ( $\tanh$ ) และฟังก์ชันกระตุ้นเครื่องหมายอ่อน (softsign,  $h(a) = \frac{a}{1+|a|}$ ). ทั้งคู่เป็นฟังก์ชันที่สามารถที่ศูนย์<sup>4</sup> คอมิง เห้อ และคณะ[84] พบร่วมกัน ที่ไม่สามารถใช้กับโครงข่ายลึก. ตามแนวทางของโกลโลร์และเบนจิโอ คณะของไอคอมิง เห้อ[84] ทำการวิเคราะห์เงื่อนไขของค่าน้ำหนัก โดยพิจารณาฟังก์ชันกระตุ้นเรลู และฟังก์ชันอื่นในลักษณะคล้ายกัน. ฟังชั่นตระกูลเรลู ที่คณะของเห้อพิจารณา อาจเขียนเป็นรูปทั่วไปได้ดังสมการ 5.10.

$$h(a) = \begin{cases} a, & \text{เมื่อ } a > 0, \\ \alpha \cdot a, & \text{เมื่อ } a \leq 0. \end{cases} \quad (5.10)$$

เมื่อ  $\alpha$  คือ พารามิเตอร์ของฟังก์ชัน. หาก  $\alpha = 0$  จะทำให้  $h(a)$  เป็นฟังก์ชันเรลู. หาก  $\alpha > 0$  เป็นค่าคงที่ โดยเป็นอภิมานพารามิเตอร์ที่กำหนดโดยผู้ใช้ จะทำให้  $h(a)$  เป็นฟังก์ชันเรลูรั่ว (leaky relu). นอกจากนั้น คณะของเห้อ ได้เสนอฟังก์ชันกระตุ้นที่สามารถปรับตัวได้ โดยให้  $\alpha > 0$  เป็นค่าพารามิเตอร์ที่ถูกฝึกไปพร้อม ๆ กับค่าน้ำหนักและไบอัส และคณะของเห้อ เรียกฟังก์ชันกระตุ้นนี้ว่า ฟังก์ชันพีเรลู (PReLU).

<sup>4</sup>ในแห่งที่ว่า ค่าห่างจากศูนย์ไปทางซ้ายและขวาเท่า ๆ กัน จะหักล้างกันได้.

เงื่อนไขที่คณะของเห้อเสนอ แสดงในสมการ 5.11.

$$\text{var}[\mathbf{w}^{(l)}] = \frac{2}{(1 + \alpha^2) \cdot m_i^{(l)}} \quad (5.11)$$

เมื่อ  $\alpha$  เป็นพารามิเตอร์ของฟังก์ชันตระกูลเรลู.

เมื่อนำเงื่อนไขในสมการ 5.11 ไปใช้กับการแจกแจงปกติ (ที่มักนิยมใช้กับเงื่อนไขของคณะของเห้อ) จะได้ว่า

$$w_{kj} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{(1 + \alpha^2) \cdot m_i}}\right). \quad (5.12)$$

การกำหนดค่าน้ำหนักเริ่มต้น ด้วยนิพจน์ 5.12 รู้จักกันทั่วไปในชื่อ การกำหนดค่าน้ำหนักด้วยวิธีไคเมิง (Kaiming weight initialization). สังเกต หากพิจารณากรณีฟังก์ชันกราฟตุ้นเรลู ( $\alpha = 0$ ) ความแปรปรวนของค่าน้ำหนัก จากเงื่อนไขไคเมิง คือ  $\frac{2}{m_i}$ . ในขณะที่นิพจน์ 5.7 ส่งผลให้ ความแปรปรวนของค่าน้ำหนัก คือ  $\frac{1}{3 \cdot m_i}$  (โดยนิพจน์ 5.7 ไม่คำนึงถึงฟังก์ชันกราฟตุ้น).

นอกจาก การกำหนดค่าน้ำหนักเริ่มต้น ด้วยการสุ่มดังอภิปรายนี้แล้ว การทำการฝึกก่อน (หัวข้อ 5.5) เพื่อได้ค่าน้ำหนักที่ดี ก่อนที่จะทำการฝึกแบบจำลองสำหรับภารกิจที่ต้องการจริงๆ เป็นแนวทางหนึ่งที่ให้ผลดีมากในทางปฏิบัติ.

## 5.5 กลไกช่วยการฝึก

ขณะที่วิธีลงเกรเดียนต์ หรือมักนิยมเรียกวิธีลงเกรเดียนต์สโตแคสติก (stochastic gradient descent) ที่เน้นถึงการสุ่มลำดับของการฝึกทีละหมู่เล็ก เป็นวิธีที่นิยมใช้ในการฝึกโครงข่ายประสาทเทียม.

แต่บ่อยครั้งที่อาจพบว่า วิธีลงเกรเดียนต์สโตแคสติก ทำให้การฝึกทำได้ช้า. หลาย ๆ เทคนิคจากศาสตร์การหาค่าดีที่สุด ได้ถูกนำมาใช้ เพื่อปรับปรุงประสิทธิภาพการฝึก. นอกจากนั้น ยังมีเทคนิคจำนวนมากที่พัฒนาขึ้นมาโดยเฉพาะสำหรับการเรียนรู้ของเครื่อง โดยเฉพาะการเรียนรู้เชิงลึก. หัวนี้ อภิรายเทคนิคต่าง ๆ บางส่วน โดยเฉพาะ เทคนิคเด่น ๆ ที่มีการใช้อย่างกว้างขวางกับการเรียนรู้เชิงลึก.

## กลไกโมเมนตัม

ในสถานะการณ์ที่ค่าฟังก์ชันจุดประสงค์ต่อค่าตัวแปรตัดสินใจต่าง ๆ (เช่นหมายถึง ฟังก์ชันสูญเสียและค่าน้ำหนักและไบอัสทั้งหลาย ในกรณีโครงข่ายประสาทเทียม) มีลักษณะความสัมพันธ์ที่มีการเปลี่ยนแปลงเร็ว หรือ

อาจจะมีสัญญาณรบกวนมาก กลไกของโมเมนตัม (momentum[153]) นิยมถูกนำมาใช้เพื่อช่วยเพิ่มประสิทธิภาพการทำงานของขั้นตอนวิธีการหาค่าดีที่สุด.

รูป 5.13 แสดงผลติกรรมการทำงานของวิธีลิงเกรเดียนต์ เมื่อ ความสัมพันธ์ของฟังก์ชันสัญเสียงกับค่าน้ำหนัก ที่มีลักษณะโค้ง แต่ความโคงแตกต่างกันมากระหว่างน้ำหนักแต่ละตัว. พฤติกรรมการทำงานปรับค่าน้ำหนัก ของวิธีลิงเกรเดียนต์ จะแสดงออกในลักษณะส่ายเข้าหากำตอบ.

แทนที่จะใช้ค่าเกรเดียนต์เพียงอย่างเดียว กลไกของโมเมนตัม เสนอที่จะใช้ ทิศทางเดิม ประกอบกับทิศทางใหม่ เพื่อลดการส่ายเข้าหากำตอบ เพื่อปรับค่าตัวแปร. นั่นคือ

$$\boldsymbol{v}^{(i+1)} = \beta \boldsymbol{v}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)}) \quad (5.13)$$

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \boldsymbol{v}^{(i+1)} \quad (5.14)$$

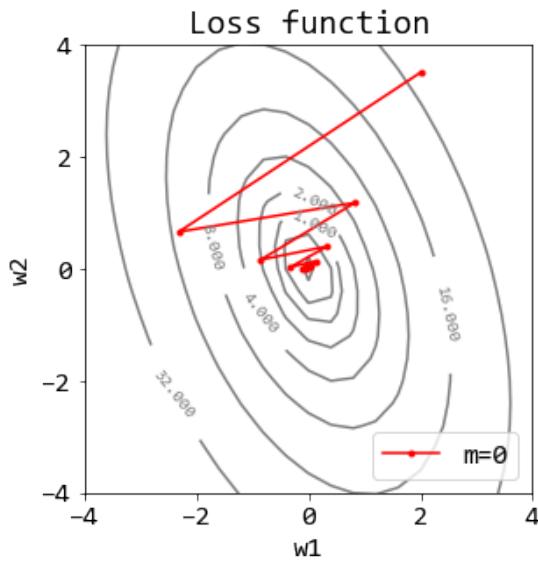
เมื่อ  $\boldsymbol{v}$  เป็นเวกเตอร์สำหรับปรับค่าตัวแปร และ  $\beta$  เป็นค่าโมเมนตัม. ส่วน  $\nabla L(\boldsymbol{\theta}^{(i)})$  คือ เกรเดียนต์ต่อตัวแปรตัดสินใจ และ  $\alpha$  คืออัตราการเรียนรู้. และ  $\boldsymbol{\theta}$  เป็นตัวแปรตัดสินใจ เช่น ค่าน้ำหนักและไบอส. ตัวยกระบุสมัยฝึก. ค่าเริ่มต้นของ  $\boldsymbol{v}$  อาจกำหนดเป็น  $\mathbf{0}$ .

เปรียบเทียบกับ  $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)})$  ซึ่งเป็นวิธีลิงเกรเดียนต์ที่ไม่มีกลไกโมเมนตัม จะเห็นว่า หากให้  $\beta = 0$  นั่นเท่ากับปิดกลไกโมเมนตัม และการทำงานของสมการ 5.13 และ 5.14 จะลดรูปมาเป็นวิธีลิงเกรเดียนต์ดั้งเดิม.

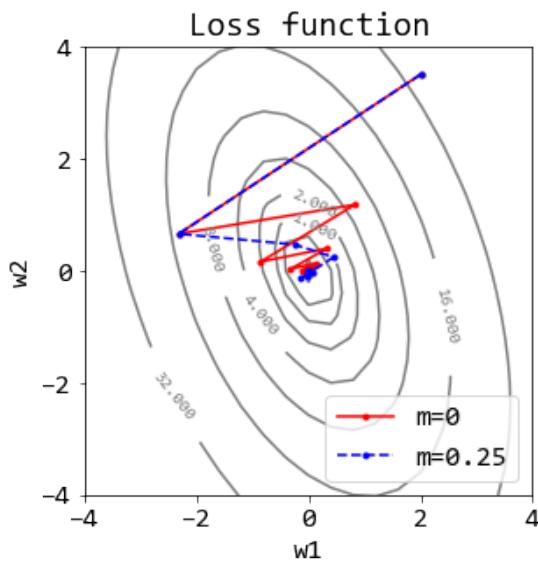
รูป 5.14 แสดงตัวอย่างที่กลไกโมเมนตัม ช่วยปรับปรุงประสิทธิภาพการฝึก โดยลดการส่ายเข้าหากำตอบระหว่างการฝึกลง. นอกจาก กลไกของโมเมนตัม ซึ่งเป็นเทคนิคที่รู้จักกันดีในวงการการหาค่าดีที่สุดอยู่แล้ว อีเลีย ซุตสเกเวอร์ (Ilya Sutskever) นักวิจัยการเรียนรู้ของเครื่องชั้นนำ ได้เสนอ เนสเตอรอฟโมเมนตัม (Nesterov momentum[192]) ซึ่งคำนวณสมการ 5.15 แทนสมการ 5.13

$$\boldsymbol{v}^{(i+1)} = \beta \boldsymbol{v}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)} + \beta \boldsymbol{v}^{(i)}). \quad (5.15)$$

เปรียบเทียบกับสมการ 5.13 เนสเตอรอฟโมเมนตัม ใช้ค่าเกรเดียนต์ ที่คำนวณ ณ ตำแหน่งค่าตัวแปรที่ขยับต่ออ กามาตามโมเมนตัม แทนตำแหน่งค่าตัวแปรปัจจุบัน. ดูแบบฝึกหัด 5.20 เพิ่มเติมสำหรับการใช้งานกลไกโมเมนตัม.



รูปที่ 5.13: ภาพคอนทัวร์ของฟังก์ชันสูญเสียต่อตัวแปรค่าน้ำหนักสองตัว  $w_1$  และ  $w_2$  พร้อมเส้นทางการปรับค่าน้ำหนัก. เส้นสีเทา แสดงระดับค่าของฟังก์ชันสูญเสีย. เส้นทึบสีแดง แสดงเส้นทางการปรับค่าน้ำหนัก ด้วยวิธีลงเกรเดียนต์. ในภาพ ค่าเริ่มต้นจาก  $(w_1, w_2) = (2, 3.5)$  (บริเวณด้านบนทางขวา). จุดที่ค่าฟังก์ชันสูญเสียต่ำสุดอยู่ที่  $(0, 0)$  (กลางภาพ). สังเกตเส้นทางการปรับค่าตัวแปร และเป็นลักษณะซิกแซก.



รูปที่ 5.14: ภาพแสดงการทำงานของกลไกโมเมนตัม (เส้นประสีน้ำเงิน) เปรียบเทียบกับ การไม่ใช้โมเมนตัม (เส้นทึบสีแดง). พื้นหลัง แสดงคอนทัวร์ของฟังก์ชันสูญเสียต่อค่าตัวแปร. ในภาพ ทั้งสองวิธีเริ่มต้นจาก  $(w_1, w_2) = (2, 3.5)$  (บริเวณด้านบนทางขวา). จุดที่ค่าฟังก์ชันสูญเสียต่ำสุดอยู่ที่  $(0, 0)$  (กลางภาพ). สังเกต โมเมนตัมช่วยลดการส่ายของเส้นทางการปรับค่าตัวแปรลง.

## ขั้นตอนวิธีที่ปรับค่าอัตราเรียนรู้

**อดาแกรต.** อดาแกรต (AdaGrad) ปรับอัตราเรียนรู้สำหรับพารามิเตอร์แต่ละตัว โดยลดขนาดอัตราเรียนรู้ลง ตามขนาด rak ที่สองของผลรวมกำลังสองของเกรเดียนต์ที่ผ่านมา. นั่นคือ พารามิเตอร์  $\theta$  จะถูกปรับค่าโดย

$$\theta^{(i+1)} = \theta^{(i)} - \frac{\alpha}{\sqrt{r^{(i+1)}} + \epsilon} \odot g \quad (5.16)$$

เมื่อ ผลรวมกำลังสองของเกรเดียนต์ที่ผ่านมา  $r^{(i+1)} = r^{(i)} + g \odot g$  และเกรเดียนต์  $g = \nabla L(\theta^{(i)})$  โดย  $\alpha$  คืออัตราเรียนรู้ (ฐาน) ที่ผู้ใช้กำหนด และ  $\epsilon$  คือค่าคงที่ขนาดเล็ก เช่น  $10^{-7}$  สำหรับเสียงรบกวน. ค่า  $r^{(0)}$  อาจกำหนดเป็น  $\mathbf{0}$ . สังเกตว่า อดาแกรต ปรับอัตราการเรียนรู้แยกกันสำหรับพารามิเตอร์แต่ละตัว. ในทางปฏิบัติพบว่า อดาแกรต ใช้งานได้ดีบางครั้ง และอาจลดอัตราเรียนรู้มากเกินไปในบางครั้ง [76].

**อาร์เอมเอสพรอป.** อาร์เอมเอสพรอป (RMSProp) ปรับปรุงอดาแกรต ด้วยการใช้ค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลัง (exponentially weighted moving average). นั่นคือ พารามิเตอร์  $\theta$  จะถูกปรับค่าโดย

$$\theta^{(i+1)} = \theta^{(i)} - \frac{\alpha}{\sqrt{r^{(i+1)}} + \epsilon} \odot g \quad (5.17)$$

เมื่อค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลัง  $r^{(i+1)} = \rho \cdot r^{(i)} + (1 - \rho) \cdot g \odot g$  โดยอัตราการสื่อมน้ำหนัก  $\rho$  เป็นอภิมานพารามิเตอร์ที่เพิ่มขึ้นมา และค่าคงที่  $\epsilon$  มักถูกเลือกเป็น  $10^{-6}$ .

แม้ว่า การเสนออาร์เอมเอสพรอปครั้งแรกไม่ได้ถูกเผยแพร่ด้วยช่องทางปกติสำหรับงานวิชาการ (การตีพิมพ์ในวารสารหรือการประชุมวิชาการ) แต่เป็นส่วนหนึ่งของการบรรยายในการสอนออนไลน์ [86] ในทางปฏิบัติ อาร์เอมเอสพรอปเป็นหนึ่งในขั้นตอนวิธีที่ใช้งานได้ดี และมีการใช้งานอย่างแพร่หลายสำหรับการฝึกแบบจำลองเชิงลึก [76].

**อดัม.** อดัม (Adam [110] ย่อจาก adaptive moments) รวมอาร์เอมเอสพรอป เข้ากับโมเมนต์ โดยเพิ่มกลไกค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลังกับการคำนวณโมเมนต์ และการปรับแก้ขนาดตามสมัยฝึก. นั่น

คือ สำหรับสมัยฝึก  $i$  และเกรเดียนต์  $\mathbf{g}$  การปรับค่าพารามิเตอร์สามารถทำได้ดังสมการ 5.22.

$$\mathbf{v}^{(i+1)} = \rho_1 \cdot \mathbf{v}^{(i)} + (1 - \rho_1) \cdot \mathbf{g} \quad (5.18)$$

$$\mathbf{r}^{(i+1)} = \rho_2 \cdot \mathbf{r}^{(i)} + (1 - \rho_2) \cdot \mathbf{g} \odot \mathbf{g} \quad (5.19)$$

$$\hat{\mathbf{v}} = \frac{\mathbf{v}^{(i+1)}}{1 - \rho_1^i} \quad (5.20)$$

$$\hat{\mathbf{r}} = \frac{\mathbf{r}^{(i+1)}}{1 - \rho_2^i} \quad (5.21)$$

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \frac{\alpha}{\sqrt{\hat{\mathbf{r}} + \epsilon}} \odot \hat{\mathbf{v}} \quad (5.22)$$

เมื่อ  $\boldsymbol{\theta}^{(i)}$  คือพารามิเตอร์หลังปรับค่าในสมัยฝึก  $i^{th}$ . ค่าเริ่มต้นของ  $\mathbf{v}^{(0)}$  และ  $\mathbf{r}^{(0)}$  อาจกำหนดเป็น  $\mathbf{0}$ . อภิมานพารามิเตอร์  $\alpha$  คืออัตราเรียนรู้ (ฐาน),  $\epsilon$  คือค่าคงที่ขนาดเล็ก (ซึ่งอาจใช้  $10^{-8}$ ), ค่า  $\rho_1 \in [0, 1]$  และ  $\rho_2 \in [0, 1]$  โดย ค่าที่แนะนำคือ  $\rho_1 = 0.9$  และ  $\rho_2 = 0.999$ .

อุดม เป็นอีกขั้นตอนวิธีที่นิยมใช้กับแบบจำลองเชิงลึก และพบว่าค่อนข้างทันทันต่อค่าอภิมานพารามิเตอร์ที่เลือก แต่อาจจะต้องปรับค่าอัตราเรียนรู้  $\alpha$  บ้างเท่านั้น.

ปัจจุบันยังไม่มีข้อสรุปถึงขั้นตอนวิธีที่ดีที่สุดโดยทั่วไป แต่ขั้นตอนวิธีที่นิยมใช้คือ[76] วิธีลงเกรเดียนต์, วิธีลงเกรเดียนต์กับโมเมนตัม, อาร์เอมแอดพรอป, อาร์เอมแอดพรอปกับโมเมนตัม, และอุดม.

## แบบนอร์มอย่างเช่น

แบบนอร์มอย่างเช่น (batch normalization) หรือเรียกว่า แบบนอร์ม (batch norm) จริง ๆ แล้ว ไม่ใช่ขั้นตอนวิธีการหาค่าดีที่สุด แต่เป็นกลไกเพื่อช่วยให้การฝึกทำได้ง่ายขึ้น.

ไอโอพีกับเซเจดี[93] ตั้งข้อสังเกตว่า ความยากของการฝึกโครงข่ายเชิงลึก ส่วนหนึ่งมาจากการเปลี่ยนแปลงอยู่ตลอดของการแยกแจงของอินพุตสำหรับแต่ชั้นคำนวน ซึ่งการเปลี่ยนแปลงนี้ เกิดจากการเปลี่ยนแปลงของค่าพารามิเตอร์ในชั้นคำนวนก่อนหน้า. ดังนั้น การฝึกจึงทำได้ช้า เพราะไม่สามารถเลือกค่าอัตราเรียนรู้ที่สูงได้ และยังต้องระวังอย่างมากในการกำหนดค่าพารามิเตอร์เริ่มต้น และยังสร้างปัญหาอย่างมากกับการใช้ฟังก์ชันกราฟตันที่มีช่วงอิมตัว (เช่น ซิกมอยด์). ไอโอพีกับเซเจดี เรียก การเปลี่ยนแปลงของการแยกแจงของอินพุตสำหรับแต่ชั้นคำนวน จากการเปลี่ยนแปลงของค่าพารามิเตอร์ในชั้นคำนวนก่อนหน้า ว่า การเลื่อนของความแปรปรวนร่วมเกี่ยวกายใน (internal covariance shift) และเสนอกลไก แบบนอร์ม เพื่อลดการเลื่อนของความแปรปรวนร่วมเกี่ยวกายใน.

หากกำหนดให้  $\mathbf{X} = [x_{ij}]$  เป็นอินพุตของชั้นคำนวน โดย  $i = 1, \dots, D; j \in B$  และ  $D$  เป็นจำนวนมิติ และ  $B$  เป็นเซตของดัชนีจุดข้อมูลในหมู่เล็ก แล้ว แบบนอร์ม เสนอแปลงอินพุตของชั้นคำนวนนี้ ดังสมการ 5.23 และ 5.24. ค่าคงที่ขนาดเล็ก  $\epsilon$  มีเพื่อรักษาเสถียรภาพของการคำนวน (อาจกำหนดให้  $\epsilon = 10^{-8}$ ).

$$x'_{ij} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad (5.23)$$

$$\hat{x}_{ij} = \gamma_i \cdot x'_{ij} + \beta_i. \quad (5.24)$$

ค่า  $\gamma_i$  กับ  $\beta_i$  เป็นพารามิเตอร์ของแบบนอร์ม ที่เรียนรู้ระหว่างการฝึก. ส่วน  $\mu_i$  และ  $\sigma_i^2$  คือค่าเฉลี่ยและความแปรปรวน นั่นคือ ในระหว่างการฝึก  $\mu_i = \frac{1}{|B|} \sum_j x_{ij}$  กับ  $\sigma_i^2 = \frac{1}{|B|} \sum_j (x_{ij} - \mu_i)^2$ . สำหรับ การใช้งานหลังฝึกเสร็จ ค่า  $\mu_i$  และ  $\sigma_i^2$  สามารถใช้ค่าที่ประมาณเตรียมไว้ระหว่างการฝึกได้.

ค่าประมาณ  $\hat{\mu}_i$  และ  $\hat{\sigma}_i^2$  นิยมประมาณด้วยค่าเฉลี่ยเคลื่อนที่อย่างน้ำหนักแบบชี้กำลัง เช่น  $\hat{\mu}_i^{(new)} = \rho \cdot \mu_i + (1 - \rho) \cdot \hat{\mu}_i^{(old)}$  และ  $\hat{\sigma}_i^{(new)} = \rho \cdot \sigma_i^2 + (1 - \rho) \cdot \hat{\sigma}_i^{(old)}$  เมื่อ ค่าสื่อมน้ำหนัก  $\rho$  เป็นอภิมานพารามิเตอร์ และ  $\mu_i$  กับ  $\sigma_i^2$  เป็นค่าที่คำนวนจากหมู่เล็กที่กำลังฝึก.

**การใช้งานแบบนอร์ม.** ชั้นคำนวน โดยทั่วไป ทำการคำนวน  $h(\mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)})$  เมื่อ  $h$  เป็นฟังก์ชันกระตุน และ  $\mathbf{W}^{(q)}$  กับ  $\mathbf{b}^{(q)}$  คือพารามิเตอร์ของชั้น. การทำแบบนอร์ม อาจทำกับค่า  $\mathbf{Z}^{(q-1)}$  โดยตรง หรืออาจทำกับ  $\mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)}$  ก็ได. ไอโอพีและเซเจดี[93] แนะนำให้ทำกับตัวกระตุน  $\mathbf{A}^{(q)} = \mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)}$ . การทำแบบนอร์มกับตัวกระตุนของชั้น ช่วยปรับค่าตัวกระตุนให้อยู่ในย่านที่ฟังก์ชันกระตุนทำงานง่ายขึ้นด้วย. นอกจากนั้น เมื่อร่วมผลลัพธ์จาก การคำนวนผลคูณค่าน้ำหนัก กับสมการ 5.23 และ 5.24 แล้วจะเห็นว่า พารามิเตอร์  $\gamma_i$  และ  $\beta_i$  ช่วยให้อิสระและความหลากหลายในการเลือกปรับค่าความแปรปรวนและค่าเฉลี่ยได. อีกเรื่องที่ควรกล่าวถึงคือ เมื่อร่วมการคำนวนแบบนอร์มเข้าไปด้วยแล้ว จะเห็นว่า ใบอัส  $\mathbf{b}^{(q)}$  ซึ่งช้อนและเกินความจำเป็น สามารถตัดออกได้.

กลไกของแบบนอร์ม ช่วยให้การฝึกของโครงข่ายประสาทเทียมทำได้ง่ายขึ้น ไอโอพีและเซเจดี[93] พบว่า แบบนอร์ม อาจช่วยให้การฝึกทำได้เร็วขึ้นถึงสิบสี่เท่า และช่วยให้การกำหนดค่าเริ่มต้นและการเลือกค่าอัตราเรียนรู้ทำได้ง่ายขึ้น (สามารถเลือกค่าได้ช่วงกว้างขึ้น โดยที่ผลลัพธ์ไม่แย่ลงมาก เมื่อเปรียบเทียบกับการเลือกค่าที่ดี).

หมายเหตุ การทำแบบนอร์ม ควรดำเนินการอย่างระมัดระวัง เพื่อไม่ใช้สัญเสียสารสนเทศที่สำคัญไป (ดูแบบฝึกหัด 5.21 ประกอบ). ดังเช่นที่ไอโอพีและเซเจดี[93] ได้แนะนำการประยุกต์ใช้กลไกของแบบนอร์ม กับโครงข่ายโครงข่ายคอนโวลูชันไว้เฉพาะ. โครงข่ายคอนโวลูชัน (บทที่ 6) นิยมใช้กับงานคอมพิวเตอร์วิทัศน์ ซึ่งข้อมูลมีลักษณะเชิงโครงสร้างของพิกเซล. นั่นคือ แต่ละจุดข้อมูลประกอบด้วยค่าพิกเซลหลาย ๆ ค่าที่จัดเรียงกันในโครงสร้าง โดยความสัมพันธ์ของค่าพิกเซลกับตำแหน่งในโครงสร้างมีสารสนเทศที่สำคัญอยู่. หากกำหนดให้ อินพุต (ค่าการกระตุน)  $\mathbf{X} = [x_{ijc}(n)] \in \mathbb{R}^{H \times W \times C \times N}$  เป็นเทนเซอร์ลำดับชั้นสี แทนรูปภาพจำนวน  $N$  รูป แต่ละรูปขนาด  $H \times W$  และมีช่องสี  $C$  ช่อง (ภาพสเกลเทา  $C = 1$ . ภาพสี  $C = 3$ . ภาพหลายสเปกตรัม multi-spectral image หรือ multi-band image ซึ่งคือภาพถ่ายของจากเหตุการณ์เดียวกันแต่ใช้อุปกรณ์รับสัญญาณหลายตัว และแต่ละตัวทำงานกับช่วงความถี่สัญญาณคลื่นแม่เหล็กไฟฟ้าต่าง ๆ กัน  $C > 1$ ). ในกรณีที่ว่าไปจำนวนช่องสีอาจมองเป็นจำนวนลักษณะสำคัญ) แล้วการทำแบบนอร์มอาจทำได้ดังสมการ 5.25 และ 5.26 สำหรับ  $c = 1, \dots, C$ .

$$x'_{ijc}(n) = \frac{x_{ijc}(n) - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \quad (5.25)$$

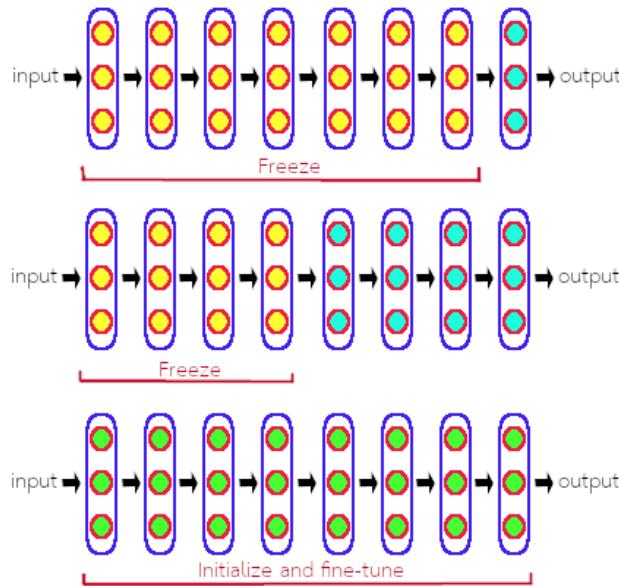
$$\hat{x}_{ijc}(n) = \gamma_c \cdot x'_{ijc}(n) + \beta_c \quad (5.26)$$

เมื่อ  $\mu_c = \frac{1}{|B| \cdot H \cdot W} \sum_{n \in B} \sum_i \sum_j x_{ijc}(n)$  และ  $\sigma_c^2 = \frac{1}{|B| \cdot H \cdot W} \sum_{n \in B} \sum_i \sum_j (x_{ijc}(n) - \mu_c)^2$ . สังเกต แบบนอร์มสำหรับชั้นคำนวนคอนโวลูชัน จะใช้พารามิเตอร์  $\gamma_c$  และ  $\beta_c$  หนึ่งคู่ต่อหนึ่งช่องสี.

## กลไกช่วยการฝึกอื่น ๆ

กลไกการฝึกก่อน (pre-training) ที่ใช้การฝึกแบบจำลองกับปัญหาที่ง่ายขึ้น หรือปัญหาที่ใกล้เคียง ก่อนจะนำค่าน้ำหนักที่ได้จากการฝึก(เบื้องต้น) มาฝึกต่อ (หรือบางครั้งนิยมอ้างถึงว่าเป็น การปรับละเอียด fine tuning) กับปัญหาที่ต้องการจริง ๆ ที่มักเรียกว่า ปัญหาเป้าหมาย.

การฝึกก่อน อาจทำโดยใช้แบบจำลองแบบเดียวกับแบบจำลองสุดท้ายที่ต้องการ แต่ฝึกกับข้อมูลอีกชุด หรือเป้าหมายอีกแบบ หรือ อาจจะฝึกแบบจำลองที่เล็กกว่า แล้วค่อยเพิ่มขยายเป็นแบบจำลองที่ต้องการ เมื่อนำมาใช้กับภารกิจเป้าหมาย เช่น ในงานคอมพิวเตอร์วิทัศน์ การทำแบบจำลองอาจจะเลือกแบบจำลองที่นิยมอยู่แล้ว เช่น อเล็กซ์เน็ต (หัวข้อ 6.5) พร้อมการเริ่มต้นด้วยค่าน้ำหนักของอเล็กซ์เน็ตที่ผ่านการฝึกมาแล้ว แทนที่จะเริ่มจากค่าน้ำหนักสุ่ม. แล้วจึงค่อยดำเนินการฝึกต่อ กับข้อมูลและฟังก์ชันสัญเสียของภารกิจเป้าหมาย. บาง



รูปที่ 5.15: แนวทางที่นิยมดำเนินการกับค่า�้าหนักจากการฝึกก่อน. ภาพบนสุด แสดงกรณีที่ข้อมูลเป้าหมายค่อนข้างน้อย วิธีที่นิยม คือ กำหนดค่า�้าหนักของชั้นคำนวนต้น ๆ ด้วยค่า�้าหนักจากการฝึกก่อน และตรึงค่าเหล่านี้ไว้ และดำเนินการฝึกโครงข่าย ด้วยการปรับค่า�้าหนักเฉพาะชั้นคำนวนหลัง ๆ. ภาพกลาง กรณีที่มีข้อมูลเป้าหมายมากพอสมควร อาจลดจำนวนชั้นที่ตรึงค่า�้าหนักลง และฝึกจำนวนชั้นคำนวนมากขึ้น. ภาพล่างสุด แสดงกรณีที่ใช้ค่า�้าหนักจากการฝึกก่อน เป็นเพียงค่า�้าหนักเริ่มต้น และทำการฝึกทั้งโครงข่ายใหม่. การเริ่มต้นฝึกจากค่า�้าหนักที่ได้จากการฝึกก่อน จะช่วยให้การฝึกต่อทำได้ง่ายและเร็วขึ้น.

ครั้ง เพื่อให้แบบจำลองที่นำมา เหมาะกับภารกิจเป้าหมาย อาจมีการปรับแต่งแบบจำลองบ้าง ได้แก่ เปลี่ยน ชั้นคำนวนท้าย ๆ เช่น เปลี่ยนชั้นสุดท้ายให้มีจำนวนเอาร์พุตสุดท้ายตามที่ต้องการ.

การนำค่า�้าหนักที่ฝึกแล้วมาใช้ในการฝึกต่อ หากมีข้อมูลของการกิจเป้าหมายมีปริมาณไม่มาก การดำเนินการ นิยมตรึงค่า�้าหนักที่ฝึกมาก่อนไว้ (ไม่มีการปรับค่า�้าหนักเหล่านี้) แต่ปรับค่า�้าหนักเฉพาะกับชั้นคำนวนหลัง ๆ ซึ่งเชื่อว่าเกี่ยวข้องกับภารกิจเป้าหมายมากกว่า. แต่หากมีข้อมูลของการกิจเป้าหมายมีปริมาณมาก อาจลดจำนวนชั้นคำนวนต้น ๆ ที่ตรึงค่า�้าหนักที่ฝึกก่อนลง และเพิ่มจำนวนชั้นคำนวนหลัง ๆ ที่ปรับค่า�้าหนักให้มากขึ้น หรือ อาจจะเพิ่มใช้ค่า�้าหนักที่ฝึกก่อนมาแทนค่า�้าหนักเริ่มต้น และฝึกค่า�้าหนักทั้งหมดในโครงข่ายเลย. รูป 5.15 แสดงแนวทางที่นิยมดำเนินการกับค่า�้าหนักจากการฝึกก่อน.

ค่า�้าหนักของการฝึกก่อน อาจได้มาโดยการเรียนรู้แบบมีผู้สอน ดังที่ได้อธิบายไป หรืออาจได้มาโดยการเรียนรู้แบบไม่มีผู้สอน (unsupervised pre-training ดูอ่วนและคณะ[64] เพิ่มเติม). การฝึกก่อน จะช่วยทั้งในแง่ของลดเวลาในฝึกลง เพิ่มคุณภาพ รวมถึงคุณสมบัติความทั่วไป และยังมองได้ว่า เป็นความสามารถในการถ่ายโอนการเรียนรู้ (transfer learning[227, 120, 146]) อีกด้วย. การถ่ายโอนการเรียนรู้ อ้างถึง สถานการณ์ที่เราสามารถใช้ประโยชน์ จากการเรียนรู้ในภารกิจหนึ่ง เพื่อช่วยการเรียนรู้ในอีกภารกิจได้ โดยการเรียนรู้ในภารกิจใหม่ ที่ได้รับการถ่ายโอนการเรียนรู้มา จะทำได้ดีหรือเร็วกว่า การเรียนรู้ในภารกิจใหม่ ที่

ไม่มีการถ่ายโอนการเรียนรู้ และต้องเริ่มเรียนทุกอย่างจากศูนย์. ความสามารถในการถ่ายโอนการเรียนรู้ ในโครงข่ายลึก เป็นอีกปัจจัยที่ช่วยให้การประยุกต์ใช้การเรียนรู้เชิงลึกทำได้ง่ายขึ้น กว้างขวางขึ้น และมีส่วนอย่างมากที่ช่วยเร่งการพัฒนาของศาสตร์อย่างมาก.

แนวทางหรือกลไก ที่อาจมองว่าคล้ายการฝึกก่อน เช่น พິຕເນີຕ (FitNets[168]) ที่ใช้แบบจำลองครู (teacher model) กับแบบจำลองนักเรียน (student model) โดยแบบจำลองครูเป็นแบบจำลองที่ดีนั้นแต่กว้าง (นั่นคือ มีจำนวนชั้นคำนวนน้อย แต่ว่าแต่ละชั้นมีจำนวนหน่วยคำนวนมาก) ซึ่งฝึกได้ง่ายกว่า. ส่วนแบบจำลองนักเรียน จะลึกแต่แคบ ทำให้มีประสิทธิภาพในการคำนวนมากกว่า แต่ฝึกยากกว่า. กลไกการฝึกแบบครูนักเรียนนี้ คือ ในการฝึกแบบจำลองนักเรียน นอกจากจะฝึกแบบจำลองนักเรียนสำหรับจุดประสงค์หลัก แล้วยังฝึกให้แบบจำลองนักเรียน โดยเฉพาะในชั้นคำนวนต้นๆ ทำนายค่าผลการกระตุ้นของชั้นคำนวนซ่อนในแบบจำลองครู ด้วย. การทำดังนี้ คือการใช้ค่าผลการกระตุ้นของชั้นคำนวนซ่อนในแบบจำลองครู เป็นเสมือนตัวช่วยนำทางสำหรับการฝึกชั้นคำนวนซ่อนของแบบจำลองนักเรียน.

การเรียนหลักสูตร (curriculum learning[12]) เป็นอีกแนวทางหนึ่งของกลไกฝึกระดับสูง. กล่าวโดยทั่วไป การเรียนหลักสูตร จะจัดการฝึกเป็นหลาย ๆ ยก โดยเริ่มจากยกแรก ๆ ที่ทำการฝึกที่ง่าย แล้วเพิ่มความยากในการฝึกขึ้นในแต่ละยก จนสุดท้าย คือการฝึกกับปัญหาที่ต้องการ. เบนจิโอ[12] ทดลองเปรียบเทียบ การเรียนหลักสูตร ซึ่งทำการฝึกข้อมูลที่ง่ายก่อน ที่จะฝึกข้อมูลที่ยาก กับการฝึกปกติ ที่ใช้ข้อมูลที่ยาก ที่เป็นเป้าหมายตั้งแต่แรก ผลที่ได้พบว่า แบบจำลองสามารถเรียนรู้ได้ดีขึ้นอย่างชัดเจน.

นอกจากขั้นตอนวิธีและกลไกต่าง ๆ ที่ช่วยการฝึกแล้ว แนวทางที่ประสบความสำเร็จอย่างมากเลย คือ การออกแบบโครงสร้างแบบจำลอง เพื่อช่วยให้การฝึกทำได้ง่ายขึ้น. จริง ๆ แล้ว การเปลี่ยนฟังก์ชันกระตุ้น ก็เป็นการเปลี่ยนโครงสร้างของแบบจำลอง เพื่อช่วยให้การฝึกทำได้ง่ายขึ้น. กูดเฟโลและคณะ[76] ตั้งข้อสังเกตว่า โครงสร้างที่มีลักษณะเชิงเส้นมากขึ้น จะช่วยให้การฝึกทำได้ง่ายขึ้น. กลไกของแบบจำลอง อาร์ม ที่มีลักษณะเป็นการเปลี่ยนโครงสร้างของแบบจำลอง. แบบจำลองหลายชนิด ถูกออกแบบมาให้มีเส้นทางการเชื่อมต่อระหว่างชั้นคำนวน โดยอาจมีการเชื่อมต่อข้ามชั้นคำนวนได้ เพื่อช่วยในการฝึก เช่น เรสเน็ต ResNet[85] ที่มีการเชื่อมต่อข้ามชั้นคำนวน เพื่อช่วยให้การแพร่กระจายของเกรเดียนต์กลับไปหาชั้นคำนวนต้น ๆ ทำได้มีประสิทธิภาพขึ้น. โครงข่ายคอนโวลูชัน (บทที่ 6) ก็เป็นลักษณะของการเปลี่ยนโครงสร้าง ซึ่งโครงสร้างของโครงข่ายคอนโวลูชัน ช่วยลดจำนวนพารามิเตอร์ที่ต้องการลง โดยอาศัยคุณสมบัติที่เหมาะสมกับข้อมูลที่มีลักษณะเชิงท้องถิ่น เช่น ภาพ. การลดจำนวนพารามิเตอร์ที่ไม่จำเป็นลง ช่วยโดยตรงต่อกระบวนการฝึก. แบบจำลองความจำระยะสั้นที่ยาว (บทที่ 8) ก็ถูกออกแบบมา สำหรับข้อมูลเชิงลำดับ เพื่อช่วยให้การฝึก เรียนรู้

ความสัมพันธ์เชิงลำดับระยะยาวทำได้มีประสิทธิภาพมากขึ้น.

## 5.6 อภิรานศัพท์

**การเรียนรู้เชิงลึก (deep learning):** การเรียนรู้ของเครื่องที่ใช้โครงข่ายประสาทเทียมจำนวนชั้นจำนวนมาก รวมไปจนถึงเทคนิคและกลไกอื่นๆ ที่เกี่ยวข้อง

**ปัญหาการเลื่อนหายของเกรเดียนต์ (vanishing gradient problem):** ปัญหา หรือปراภภารณ์ ที่ขนาดเฉลี่ยของเกรเดียนต์ลดลงอย่างมากที่ชั้นคำนวณต้น ๆ เมื่อเปรียบเทียบกับค่าเฉลี่ยชั้นคำนวณปลาย ๆ

**ฟังก์ชันกระตุ้น rectified linear function (relu):** หรือ เรลู (relu): ฟังก์ชันกระตุ้น  $\text{relu}(a) = \max(a, 0)$

**หมู่เล็ก (minibatch):** ส่วนของข้อมูลที่ถูกแบ่งเป็นกลุ่มเล็ก ๆ สำหรับการฝึก โดยในหนึ่งสมัยฝึก จะต้องทำการปรับค่าน้ำหนักหลายครั้ง แต่ละครั้งสำหรับแต่ละหมู่เล็ก และการปรับแต่ละครั้ง คำนวณจากจุดข้อมูลต่างๆ ในหมู่เล็ก และจะใช้ค่าเฉลี่ยของเกรเดียนต์全局ในหมู่ในการปรับค่าน้ำหนัก เปรียบเทียบกับการฝึกแบบออนไลน์ ที่การปรับค่าน้ำหนักแต่ละครั้ง คำนวณจากหนึ่งจุดข้อมูล และเปรียบเทียบกับการฝึกแบบออฟไลน์ ที่การปรับค่าน้ำหนัก คำนวณจากข้อมูลทั้งหมดที่เดียว และปรับค่าแค่ครั้งเดียว ต่อสมัยฝึก

**การตกออก (drop out):** กลไกการทำ regularization สำหรับโครงข่ายประสาทเทียม โดยการสุ่มปิดผลการคำนวณ ตัวนั้น ของหน่วยคำนวณย่อยต่าง ๆ

**แบนนอร์ม (batch norm) หรือ แบนนอร์มอไลเซชัน (batch normalization):** กลไก เพื่อช่วยการฝึกโครงข่ายประสาทเทียม โดยปรับค่าเฉลี่ยและความแปรปรวนของตัวกระตุ้นในชั้นคำนวณ

## 5.7 แบบฝึกหัด

“For any scientist, the real challenge is not to stay within the secure garden of the known but to venture out into the wilds of the unknown.”

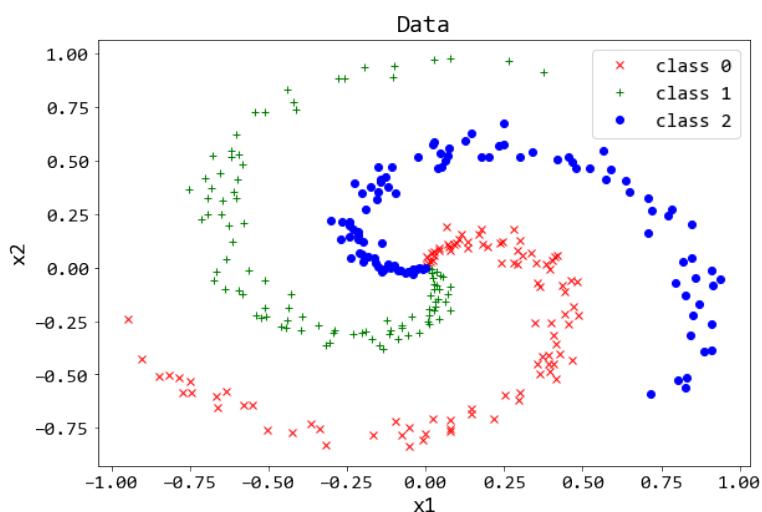
---Marcus Du Sautoy

“สำหรับนักวิทยาศาสตร์ ความท้าทายจริง ๆ ไม่ใช่การพักอยู่ภายในสวนที่ปลอดภัยของสิ่งที่รู้ แต่เป็นการท่องออกไปในป่าของความไม่รู้.”

—マークス・ダウソト

### แบบฝึกหัด 5.1

จงศึกษาตัวอย่างและแสดงปัญหาการเลือนหายของเกรเดียนต์ พร้อมเปรียบเทียบผลลัพธ์จากการบรรเทา โดยเปลี่ยนมาใช้ฟังก์ชันกระตุ้นเร็ว.



รูปที่ 5.16: ตัวอย่างข้อมูลงานจำแนกประเภทเพื่อแสดงปัญหาการเลือนหายของเกรเดียนต์. ข้อมูลสร้างจาก จุดข้อมูลที่  $i^{th}$  ของกลุ่ม  $c$  นั่นคือ  $\mathbf{x}_c(i) = [r_c(i) \cdot \sin \theta_c(i), r_c(i) \cdot \cos \theta_c(i)]^T$  โดย  $c$  เป็นดัชนีของกลุ่ม และทุก ๆ กลุ่มมี  $r_c(i) = (i-1)/N$  กับ  $\theta_c(i) = (i-1) \cdot \frac{4\pi}{3N} + c \cdot \frac{2\pi}{3} + \varepsilon$  สำหรับ  $i \in \{1, \dots, N\}$  และ  $N$  คือจำนวนจุดข้อมูลของแต่ละกลุ่ม. ส่วนสัญญาณรบกวน  $\varepsilon \sim \mathcal{N}(0, 0.2)$ .

ตัวอย่างเช่น (1) เขียนโปรแกรมเพื่อสร้างข้อมูล. รูป 5.16 แสดงตัวอย่างข้อมูล<sup>5</sup> ที่เป็นปัญหาการจำแนกกลุ่ม โดยอินพุตมี 2 มิติ และเอาต์พุตเป็นชนิดมี 3 ชนิด ซึ่งสร้างจากตัวอย่างคำสั่งข้างล่าง

$N = 100$

$X = np.zeros((2, N*3))$  # Initialize dummy input

<sup>5</sup> ตัดแปลงจาก [https://cs224d.stanford.edu/notebooks/vanishing\\_grad\\_example.html](https://cs224d.stanford.edu/notebooks/vanishing_grad_example.html) (ข้อมูลเมื่อ 24 พ.ค. 2560).

```

y = np.zeros((1, N*3), dtype='uint8') # Initialize dummy output
sec = 2*np.pi/3

for k in range(3):
    ix = range(N*k,N*(k+1)) ## Indices of class k
    r = np.linspace(0.0,1,N) ## Radius
    t = np.linspace(k*sec,(k+2)*sec, N) + np.random.randn(N)*0.2
    X[:, ix] = np.c_[r*np.sin(t), r*np.cos(t)].T
    y[0, ix] = k

```

หมายเหตุ ไม่จำเป็นต้องสร้างข้อมูลตามตัวอย่างในรูป.

จากนั้น (2) ทดลองสร้าง ฝึก และทดสอบโครงข่ายประสาทเทียมความลึกต่าง ๆ โดยเพิ่มความลึกขึ้นเรื่อยๆ และสังเกตความยากของการฝึก. ดูหัวข้อ 5.1 ประกอบ. (ตัวอย่างโปรแกรม ศึกษาได้จากหัวข้อ 3.7.) สุดท้าย (3) ทดลองเปลี่ยนฟังก์ชันกระตุนเป็นReLU (ตัวอย่างโปรแกรมการคำนวณReLU แสดงในรายการ 4.2.) สังเกตผล เปรียบเทียบ และอภิปราย.

## แบบฝึกหัด 5.2

จากแบบฝึกหัด 5.1 ตั้งสมมติฐานถึงสาเหตุของปัญหาการฝึกโครงข่ายประสาทเทียมลึก ออกแบบการทดลอง เพื่อพิสูจน์และศึกษาสมมติฐานนั้น ดำเนินการทดลอง สังเกตผล วิเคราะห์ สรุป วิจารณ์และอภิปราย. ศึกษาและทดลองทั้งฟังก์ชันกระตุนซิกมอยด์ และReLU พร้อมสังเกตขนาดเกรเดียนต์ที่ซึ้งต่าง ๆ ขณะฝึก. อภิปรายถึงสาเหตุอื่นที่อาจเป็นไปได้ นอกจกขนาดของเกรเดียนต์. ดูรูป 5.17 และผลในหัวข้อ 5.1 ประกอบ.

รายการ 5.1 แสดงโปรแกรมโครงข่ายประสาทเทียมที่ปรับปรุงใหม่ โดยเขียนอยู่ในรูปแบบโปรแกรมเชิงวัตถุ และที่เมท็อด `train` มีอาร์กิวเมนต์ `track_grad` ที่สามารถสั่งให้เก็บขนาดของเกรเดียนต์ไว้เพื่อตรวจสอบภายหลังได้. ตัวอย่างคำสั่งข้างล่าง ฝึกและทดสอบโครงข่ายสามชั้น (จำนวนหน่วยชั้non เป็น 4 และ 8 ชั้นตามลำดับ) สำหรับข้อมูล `datax` และ `y_onehot` ที่อินพุตมีขนาดสองมิติและเอาต์พุตอยู่ในรูปแบบรหัสหนึ่งร้อน สำหรับงานจำแนกกลุ่มที่มีสามกลุ่ม โดยมีจำนวนข้อมูลฝึกเป็น 300 จุดข้อมูล

```

net = w_initn([2, 4, 8, 3])
net['act1'] = sigmoid
net['act2'] = sigmoid
net['act3'] = softmax
ann = ANN(net, NB=300, shuffle='once')

# Train net
train_losses, maggrads = ann.train(datax, y_onehot, cross_entropy,

```

```
lr=0.3/300, epochs=500, track_grad=True)
```

```
yp = ann.predict(testx)
yc = np.argmax(yp, axis=0)
accuracy = np.mean(yc == testy[0,:])
print('Test accuracy: ', accuracy)
```

เมื่อ **testx** และ **testy** เป็นอินพุตและเอาต์พุตของข้อมูลทดสอบ และเฉลย **testy** ระบุฉลากที่ถูกต้องของจุดข้อมูล. โปรแกรม **w\_initn**, **sigmoid**, **softmax**, และ **cross\_entropy** แสดงในรายการ 3.10, 3.8, 3.17 และ 3.18 ตามลำดับ. โปรแกรม **cross\_entropy** ในรายการ 3.18 คำนวณผลรวมของค่าฟังก์ชันสูญเสียต่อจุดข้อมูลอกรอบ การกำหนดค่าอัตราการเรียนรู้ **lr=0.3/300** ให้ผลในการฝึก เสมือนว่าค่าน้ำหนักถูกปรับจากค่าเฉลี่ยของค่าฟังก์ชันสูญเสียต่อจุดข้อมูล ด้วยอัตราการเรียนรู้ 0.3. นั่นคือ  $w - (\alpha/N) \cdot \sum_n \nabla E \equiv w - \alpha \cdot \frac{1}{N} \sum_n \nabla E$ . แม้ว่าผลจริงไม่ได้แตกต่างกัน แต่การใช้ค่าเฉลี่ย (ในวิธีที่แสดงนี้) ช่วยให้การเลือกอัตราเรียนรู้ทำได้สะดวกขึ้น. ค่าอัตราเรียนรู้ สามารถเลือกได้โดยไม่ต้องคำนึงถึงจำนวนจุดข้อมูลฝึก.

หมายเหตุ นอกจากการเขียนในรูปโปรแกรมเชิงวัตถุ และเพิ่ม **track\_grad** แล้ว ส่วนหนึ่งที่สำคัญคือ โปรแกรมในรายการ 5.1 ได้เตรียมความสามารถในการฝึกหมู่เล็ก (หัวข้อ 5.2) ซึ่งการฝึกหมู่เล็ก ไม่ใช่จุดประสงค์ของแบบฝึกหัดนี้ และ ดังเช่นที่แสดงในตัวอย่างคำสั่งข้างต้น สามารถกำหนดให้ทำการฝึกแบบหมู่ได้โดยการกำหนดจำนวนหมู่เล็ก เท่ากับ(หรือมากกว่า) จำนวนของจุดข้อมูลฝึก ดังเช่น

```
ann = ANN(net, NB=300, shuffle='once')
```

เมื่อ 300 คือจำนวนจุดข้อมูลฝึก.

รายการ 5.1: คลาส สำหรับคำนวณการฝึกและการคำนวณของโครงข่ายประสาทเทียน

---

```
1 class ANN:
2     def __init__(self, net_params, NB=16, shuffle='once'):
3         ...
4         NB: minibatch size
5         shuffle: 'none'=no shuffle, 'once', 'often'=every epoch
6         net_params: weights, biases, and activation functions
7         ...
8         self.NB = NB
9         self.shuffle = shuffle
10        self.net_params = net_params
11        self.NB_ids = None
```

```
12         self.NMB = None
13
14     def prepare_minibatches(self, N):
15         if self.NB > N:
16             self.NB = N
17
18         self.NMB = int(N/self.NB)    # a number of minibatches
19         self.NB_ids = np.arange(N)
20         if self.shuffle != 'none':
21             np.random.shuffle(self.NB_ids)
22
23     def getbatch(self, i, X, Y):
24         if i == 0 and self.shuffle == 'often':
25             np.random.shuffle(self.NB_ids)
26         bids = i * self.NB
27         eids = bids + self.NB
28         ids = self.NB_ids[bids:eids]
29         return X[:, ids], Y[:, ids]
30
31     def train(self, trainX, trainY, loss, lr=0.1, epochs=1000,
32               track_grad=False, term=1e-8, term_count_max=5):
33         num_layers = self.net_params['layers']
34         last_layer = num_layers-1
35
36         out_act = 'act%d'%last_layer
37         _, N = trainX.shape
38         A = {}
39         Z = {}
40         delta = {}
41         dEw = {}
42         dEb = {}
43         train_losses = []
44         term_count = 0
45
46         # Minibatch
47         self.prepare_minibatches(N)
48
49         step_size = lr
50         if track_grad:
51             magGrad = {}
52             for i in range(1, num_layers):
```

```

53         magGrad['dEw%d'%i] = []
54         magGrad['dEb%d'%i] = []
55
56     for nt in range(epochs):
57         for ib in range(self.NMB):
58             Z[0], batchY = self.getbatch(ib, trainX, trainY)
59             # (1) Forward pass
60             for i in range(1, num_layers):
61                 b = self.net_params['bias%d'%i]
62                 w = self.net_params['weight%d'%i]
63                 act_f = self.net_params['act%d'%i]
64                 A[i] = np.dot(w, Z[i-1]) + b      # A: M x N
65                 Z[i] = act_f(A[i])              # Z: M x N
66             # end forward pass
67             Yp = Z[i]
68
69             # (2) Calculate output dE/da
70             delta[last_layer] = Yp - batchY # delta: M x N
71
72             # (3) Backpropagate: calc. dE/da for Layer i-1
73             for i in range(last_layer, 1, -1):
74                 b = self.net_params['bias%d'%i]    # Mnxt,1
75                 w = self.net_params['weight%d'%i] # Mnxt,M
76                 act_f = self.net_params['act%d'%(i-1)]
77
78                 sumdw = np.dot(w.transpose(), delta[i]) #M,N
79                 if act_f == sigmoid:
80                     delta[i - 1] = dsigmoid(Z[i - 1]) * sumdw
81                 elif act_f == relu:
82                     delta[i - 1] = drelu(A[i - 1]) * sumdw
83                 else:
84                     assert act_f == sigmoid or act_f == relu
85
86             # (4) Calculate gradient dE/dw and dE/db
87             dEw[i] = np.dot(delta[i], Z[i-1].transpose())
88             dEb[i] = np.dot(delta[i], np.ones((self.NB,1)))
89             if track_grad:
90                 magE = np.mean(np.abs(dEw[i]))
91                 magB = np.mean(np.abs(dEb[i]))
92                 magGrad['dEw%d'%i].append(magE)
93                 magGrad['dEb%d'%i].append(magB)

```

```

# end backpropagate

# Calculate gradient dE/dw and dE/db
dEw[1] = np.dot(delta[1], Z[0].transpose())
dEb[1] = np.dot(delta[1], np.ones((self.NB, 1)))

if track_grad:
    magE = np.mean(np.abs(dEw[1]))
    magB = np.mean(np.abs(dEb[1]))
    magGrad['dEw1'].append(magE)
    magGrad['dEb1'].append(magB)

# Update parameters w/ Gradient Descent
gnorm = 0
for i in range(1, num_layers):
    b = self.net_params['bias%d'%i]
    w = self.net_params['weight%d'%i]
    b -= step_size * dEb[i]
    w -= step_size * dEw[i]

    gnorm += np.linalg.norm(dEb[i])
    gnorm += np.linalg.norm(dEw[i])
# end update parameters

# Calculate loss at each batch
lossn = np.sum(loss(Yp, batchY), axis=0)
train_losses.append(np.mean(lossn))

# Check termination condition
if gnorm < term:
    term_count += 1

    if term_count > term_count_max:
        print('Reach term. at %d(%d)'%(nt, ib))
        if track_grad:
            return train_losses, magGrad
        return train_losses # Losses per batches
else: # reset term_count
    term_count = 0
# end if term_count
# end ib

```

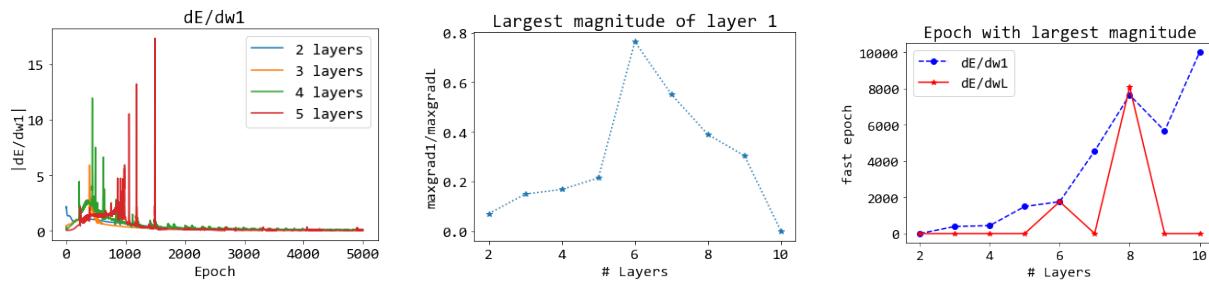
```

135 # end epoch nt
136
137 if track_grad:
138     return train_losses, magGrad
139 return train_losses # Losses per batches
140
141 def predict(self, X):
142     num_layers = self.net_params['layers']
143     Z = X
144     for i in range(1, num_layers):
145         b = self.net_params['bias%d'%i]
146         w = self.net_params['weight%d'%i]
147         act_f = self.net_params['act%d'%i]
148         A = np.dot(w, Z) + b      # A: M x N
149         Z = act_f(A)            # Z: M x N
150     return Z # M x N

```

---

รูป 5.17 แสดงตัวอย่างการนำเสนอผล. จากผลที่แสดงในรูป 5.17 อาจอภิปราย ได้ดังนี้ (1) ภาพกลางแสดงในเห็นชัดเจนว่า ส่วนใหญ่ขนาดของเกรเดียนต์ในชั้นแรก น้อยกว่าชั้นสุดท้าย และน้อยกว่ามากๆ โดยส่วนใหญ่ แต่แนวโน้ม ไม่ได้เป็นไปในทางเดียว นั่นคือ พบรขนาดเกรเดียนต์ในชั้นแรกที่ใหญ่ที่สุด เมื่อใช้ความลึก 6 ชั้น (ซึ่งมีขนาดถึงเกือบ 0.8 หรือเกือบ 80% ของขนาดเกรเดียนต์ชั้นสุดท้าย) และผลลัพธ์แสดงการลดลงในทั้งสองทิศทาง โดยที่ความลึกสิบชั้น ขนาดเกรเดียนต์ในชั้นแรกมีค่าต่ำมากเมื่อเทียบกับชั้นสุดท้าย. (2) ภาพซ้าย และภาพขวา แสดงสาเหตุในเห็นอกมุมหนึ่ง คือไม่ใช่แค่ขนาดที่น้อยอย่างเดียว แต่เป็น เมื่อไรที่ชั้นคำนวนต้น ๆ จะได้เกรเดียนต์ขนาดใหญ่. ภาพซ้าย แสดงให้เห็นว่า เกรเดียนต์ชั้นแรกที่มีขนาดใหญ่จะมาชั่ลง ในโครงข่ายที่ลึกขึ้น. ภาพขวา ยืนยันเรื่องที่เกรเดียนต์ขนาดใหญ่มาช้า ในโครงข่ายลึก. สังเกตว่า ชั้นสุดท้าย (เส้นหนาสีแดง) จะเห็นเกรเดียนต์ขนาดใหญ่ที่สุด ในสมัยฝึกต้นๆ (เห็นเร็ว) แทบจะทุกระดับความลึก (ยกเว้นความลึก 8). แต่ชั้นแรก (เส้นประสีน้ำเงิน) จะเห็นเกรเดียนต์ขนาดใหญ่ที่สุด ชั่ลงเรื่อยๆ (สมัยฝึกสูง) เมื่อความลึกเพิ่มขึ้นเรื่อยๆ โดยแนวโน้มแทบจะเป็นลำดับทางเดียว (monotonic). การที่เห็นเกรเดียนต์ขนาดใหญ่ช้า อาจหมายถึง การปรับค่าน้ำหนักของชั้นได้ช้าด้วย ซึ่งตีความได้ว่า การใช้โครงข่ายที่ลึกนั้น ต้องการการฝึกที่ยาวนานขึ้น และการฝึกที่ยาวนานขึ้น โดยทั่วไปแล้ว หมายถึง การฝึกที่ยาก.



รูปที่ 5.17: ปัญหาการเลื่อนหายของเกรเดียนต์. ภาพซ้าย แสดงขนาดเฉลี่ยของเกรเดียนต์ชั้นที่หนึ่ง ต่อสมัยฝึก ของโครงข่าย ประสิทธิ์ความลึกต่าง ๆ. ภาพกลาง แสดงอัตราส่วนระหว่างขนาดที่ใหญ่ที่สุดจากเกรเดียนต์ชั้นที่หนึ่ง กับขนาดที่ใหญ่ที่สุดจาก เกรเดียนต์ชั้นสุดท้าย เมื่อใช้ความลึกต่าง ๆ. ภาพขวา แสดงสมัยฝึกที่เกรเดียนต์มีขนาดใหญ่ที่สุด ของชั้นแรก และชั้นสุดท้าย เมื่อใช้ ความลึกต่าง ๆ.

### แบบฝึกหัด 5.3

จากแบบฝึกหัด 5.1 และ 5.2 ออกแบบการทดลอง เพื่อวัดผลการแก้ปัญหาการฝึกโครงข่ายลึก และผล การบรรเทาปัญหาการเลื่อนหายของเกรเดียนต์ เมื่อใช้ฟังก์ชันกระตุนเรลู เปรียบเทียบกับซิกมอยด์ ดำเนินการ ทดลอง สังเกต วัดผล สรุปและนำเสนอผลให้ชัดเจน ทั้งประเด็นใหญ่ (การฝึกโครงข่ายลึก) และประเด็นย่อย (การเลื่อนหายของเกรเดียนต์).

### แบบฝึกหัด 5.4

จากหัวข้อ 5.2 ออกแบบการทดลอง เพื่อศึกษาผลของขนาดหมู่ลึก ต่อเวลาในการฝึก ความยากง่ายใน การฝึก และคุณภาพการฝึก โดยมีปัจจัยประกอบคือ (1) ความลึกของโครงข่ายประสิทธิ์เทียม และ (2) จำนวน ข้อมูลฝึก. เลือก (หรือสร้าง) ข้อมูลขึ้นมา ดำเนินการทดลอง สังเกตและบันทึกผล สรุปและอภิปราย.

ด้วยข้อมูลที่มีเพิ่มมากขึ้น ชุดข้อมูลที่มีขนาดใหญ่มากๆ อาจพบการฝึกแบบหมู่ลึกที่ทำเพียงสมัยเดียว หรือแม้แต่บางครั้งอาจจะไม่สามารถฝึกได้ครบทุกหมู่ลึก (ไม่ครบสมัย และไม่ได้เห็นข้อมูลครบทั้งหมด). สำหรับชุดข้อมูลที่มีขนาดใหญ่มากๆ อาจพบปัญหาประสิทธิ์ภาพของการคำนวณ และหากเลือกใช้ข้อมูลเพียง บางส่วน อาจเกิดปัญหาการอันเดอร์พิทต์ได้. อภิปราย ประเด็นการทำงานกับข้อมูลขนาดใหญ่มาก และศึกษา เพิ่มเติมจากบทความวิจัยต่าง ๆ.

ไฟทอร์ช. โปรแกรมการเรียนรู้เชิงลึก สามารถเขียนด้วยนัมเบอร์ไฟ์ได้ แต่เนื่องจากการประยุกต์ใช้ที่เด่นๆ ของ เกี่ยวกับข้อมูลที่มีมิติและจำนวนมหาศาล การคำนวณด้วยจีพียู จะช่วยการทำงานกับข้อมูลเหล่านี้ให้เสร็จได้ เร็วขึ้นมาก. หัวข้อนี้ แนะนำมอดูลไฟทอร์ช (PyTorch) ซึ่งเป็นหนึ่งในเครื่องมือที่นิยมใช้กับการเรียนรู้เชิงลึก.

มอดูลไฟثورช ช่วยอำนวยความสะดวก ตั้งแต่การย้ายการคำนวณไปทำที่จีพียุ การหาค่าเกรเดียนต์อัตโนมัติ ไปจนถึงโปรแกรมสำเร็จรูปสำหรับกลไกการเรียนรู้เชิงลึกเด่น ๆ ซึ่งจะช่วยให้การใช้งาน และการเรียนรู้การเรียนรู้เชิงลึกทำได้สะดวกมากยิ่งขึ้น.

อย่างไรก็ตาม ถึงแม้ไฟثورช จะได้เตรียมโปรแกรมสำเร็จต่าง ๆ ไว้ให้ แต่การได้เขียนโปรแกรมจากปฏิบัติ การพื้นฐานขึ้นเอง ก็ยังเป็นกระบวนการเรียนรู้ที่สำคัญ ที่ช่วยให้เข้าใจอย่างแท้จริง. ดังนั้น การดำเนินเนื้อหา จะเป็นลักษณะเช่นเดิม นั่นคือ เริ่มจากการเขียนโปรแกรมกลไกต่าง ๆ ขึ้นเอง จากปฏิบัติการพื้นฐาน แล้วค่อยทดลองใช้เครื่องมือสำเร็จที่มี ในลักษณะค่อย ๆ ขยับทีละขั้น เพื่อสร้างทั้งความเข้าใจ ความคุ้นเคย และสำคัญ ไม่แพ้กันคือ ความมั่นใจ.

การติดตั้งไฟثورช แนะนำให้ศึกษาจากเวป <https://pytorch.org/> โดยหากระบบมีจีพียุ และยังไม่ได้เตรียมการใช้งาน แนะนำให้ติดตั้งและเตรียมการใช้งานจีพียุ ก่อนติดตั้งไฟثورช. หลังติดตั้งเรียบร้อย เช่นเดียวกับการใช้งานโมเดลเพิ่มเติมอื่น ๆ เราต้องนำเข้า มอดูลไฟثورชก่อน ด้วยคำสั่ง เช่น `import torch` เมื่อนำเข้าสมบูรณ์ สามารถทดสอบง่าย ๆ ได้โดยการตรวจสอบเวอร์ชันของไฟثورช เช่น

```
>>> print(torch.__version__)
1.0.0
```

ซึ่ง **1.0.0** คือเวอร์ชันที่ใช้<sup>6</sup> หากไฟثورชที่ติดตั้งเป็นเวอร์ชันอื่นก็จะได้ค่าอื่นอกมา.

รายการ [5.2](#) แสดงโปรแกรมฟังก์ชันกระตุ้นเร็ว ซอฟต์แวร์ และครอสเอนโทรปี พร้อมฟังก์ชันกำหนดค่าเริ่มต้น ซึ่งทั้งหมดเปลี่ยนเครื่องมือจากนัมไฟมาเป็นไฟثورช. หมายเหตุ ฟังก์ชันครอสเอนโทรปี ใช้ `eps` เป็นกลไกในการลดปัญหาการคำนวณเชิงเลข. นั่นคือ กรณีที่ค่าที่หายเป็นศูนย์ สำหรับเฉลยเป็นหนึ่ง (หายผิดมากๆ อาจเกิดตอนเริ่มต้น) จะทำให้เกิด  $-\log(0) \rightarrow \infty$ . กรณีเช่นนี้ จะทำให้การคำนวณพัง และไม่สามารถคำนวณต่อไปได้. กลไกในการแก้คือใช้ค่าเล็กๆ เติมเข้าไป  $-\log(0 + \epsilon) \rightarrow v_{\max}$  ซึ่ง  $v_{\max}$  คือ ค่ามากที่สุด ( $\approx 103$ ) เท่าที่ `-torch.log` จะสามารถคำนวณได้ก่อนจะให้ค่าอกมาเป็น `inf`. ค่า **1e-45** ที่เลือกใช้ มาจากค่าบวกที่เล็กที่สุด ที่เลขทศนิยมขนาดสามสิบสองบิตจะแทนได้ ซึ่งตัวเลขนี้จะต่างจาก **1e-323** ในรายการ [3.18](#) ที่สำหรับเลขทศนิยมขนาดหกสิบสี่บิตจะแทนได้.

รายการ [5.4](#) แสดงโปรแกรมคำนวณโครงข่ายประสาทเทียม ด้วยไฟثورช. สังเกตว่า การสร้างเทนเซอร์ใหม่

<sup>6</sup>ตัวอย่างคำสั่งและโปรแกรมต่าง ๆ ที่จะแสดงนี้ ทดสอบกับไฟثورช เวอร์ชัน 1.0.0.

จะมีการกำหนด **device** ด้วย ซึ่ง การกำหนดนี้จะช่วยให้เราสามารถเปลี่ยนการคำนวณระหว่าง ชิปปี้ และ จีพียูได้สะดวกขึ้น. ดูแบบฝึกหัด 5.6 สำหรับการคำนวณด้วยจีพียู.

รายการ 5.2: พัฒนากระตุน เขียนด้วยไฟทอร์ช

```

1 def trelu(a):
2     return a.clamp(min=0)
3
4 def tdrelu(a):
5     g = torch.ones(a.shape, device=a.device)
6     g[a < 0] = 0
7     return g
8
9 def tcross_entropy(yhat, y):
10    assert yhat.shape == y.shape
11    eps = 1e-45
12    v = -torch.log(torch.sum(y * yhat, dim=0) + eps)
13    return v.reshape((1, -1))
14
15 def tsoftmax(va):
16    assert va.shape[0] > 1, 'va must be in K x N.'
17    amax = torch.max(va, dim=0)[0]
18    expa = torch.exp(va - amax)
19    denom = torch.sum(expa, dim=0)
20    return expa/denom
21
22 def tw_initn1(Ms, umeansigma=(0,1), dev=torch.device('cpu')):
23     assert len(Ms) >= 2, 'Ms: #units, e.g., M = [2, 8, 3]'
24     num_layers = len(Ms)
25     params = {'layers': num_layers}
26     mu = umeansigma[0]
27     sigma = umeansigma[1]
28     for i, m in enumerate(Ms[1:], start=1):
29         mprev = Ms[i-1]
30         b = torch.randn((m,1), device=dev)
31         w = torch.randn((m,mprev), device=dev)
32         params['bias%d'%i] = b*sigma + mu
33         params['weight%d'%i] = w*sigma + mu
34
35     return params

```

รายการ 5.4 แสดงโปรแกรมโครงข่ายประสาทเทียม ที่เขียนด้วยไฟทอร์ช. เมื่อเปรียบเทียบโปรแกรม

ในรายการ 5.4 กับโปรแกรมในรายการ 5.1 จะพบว่า (1) คลาส **tANN1** รับมารดก<sup>7</sup> มาจากคลาส **ANN** (รายการ 5.1) เพื่อลดความซ้ำซ้อน และ (2) เมท็อด **train** และ **predict** เพียงเปลี่ยนมาใช้คำสั่งของไฟฟอร์ชเท่านั้น<sup>8</sup>. นอกจากนั้น เพื่อความกระชับ เมท็อด **train** ได้ตัด **track\_grad** ออก (ไม่มี **track\_grad** ในเมท็อด **train** เช่นในรายการ 5.4. หมายเหตุ **track\_grad** ใช้ประกอบการศึกษาปัญหาการเลือนหายของเกรเดียนต์ ดูแบบฝึกหัด 5.2 เพิ่มเติม). ข้อควรระวังคือ เมื่อใช้ไฟฟอร์ช ข้อมูลเห็นเชอร์ที่ประมวลผลทุกตัว ต้องอยู่ในรูปแบบเทนเซอร์ของไฟฟอร์ช.

ตัวอย่างคำสั่งต่อไปนี้ฝึก และทดสอบโครงข่ายประสาทเทียมที่เขียนด้วยไฟฟอร์ช

รายการ 5.3: ตัวอย่างโปรแกรมรันโครงข่ายประสาทเทียมที่เขียนด้วยไฟฟอร์ช

---

```

1 dev = torch.device('cpu')
2 net = tw_initn1([2, 8, 8, 3], dev=dev)
3 net['act1'] = trelu
4 net['act2'] = trelu
5 net['act3'] = tsoftmax
6 ann = tANN1(net, NB=50, shuffle='once')
7
8 t_losses = ann.train(x, y_onehot, tcross_entropy,
9                      lr=0.0017, epochs=500)
10 yp = ann.predict(ttestx)
11 ypn = yp.to(torch.device('cpu')).data.numpy()
12 yc = np.argmax(ypn, axis=0)
13 accuracy = np.mean(yc == testy[0,:])
14 print('**Test accuracy: ', accuracy)

```

---

เมื่อ **x**, **y\_onehot**, และ **ttestx** เป็นอินพุตของข้อมูลฝึก, เอ้าต์พุตของข้อมูลฝึก, และอินพุตของข้อมูลทดสอบ ในรูปแบบของไฟฟอร์ช. ส่วน **testy** เป็นเอ้าต์พุตของข้อมูลทดสอบในรูปแบบนัมมไป.

ข้อมูลสามารถแปลงไปมาระหว่างรูปแบบของนัมมไปและไฟฟอร์ช ได้เช่น คำสั่ง

**ypn = yp.to(torch.device('cpu')).data.numpy()**

แปลง **yp** จากรูปแบบไฟฟอร์ช ออกมามาเป็นข้อมูลในรูปแบบนัมมไปอาร์เรย์. การแปลงจากข้อมูลนัมมไปอาร์เรย์ ก็สามารถแปลงเป็นไฟฟอร์ช ได้เช่น

**x = torch.from\_numpy(trainx).float().to(dev)**

<sup>7</sup>การรับมารดก (inheritance) เป็นกลไกการเขียนโปรแกรมเชิงวัตถุ (object-oriented programming) ที่สำคัญ ช่วยให้เราสามารถใช้โปรแกรมเดิมซ้ำได้โดยเปลี่ยนเฉพาะส่วนที่จำเป็น.

<sup>8</sup>เพื่อลดความซับซ้อนของโปรแกรม สำหรับขั้นตอน คลาส **tANN1** รับฟังก์ชันกระตุ้น **trelu** ได้เท่านั้น.

เป็นการแปลงข้อมูลนั้นไปอยู่ในรูปแบบไฟฟอร์ช.

รายการ 5.4: คลาส สำหรับคำนวณการฝึกและการทำนายของโครงข่ายประสาทเทียม ด้วยไฟฟอร์ช

```

1 class tANN1(ANN):
2     def train(self, trainX, trainY, loss, lr=0.1, epochs=1000,
3             term=1e-8, term_count_max=5):
4         num_layers = self.net_params['layers']
5         last_layer = num_layers-1
6         out_act = 'act%d'%last_layer
7         _, N = trainX.shape
8         A = {}
9         Z = {}
10        delta = {}
11        dEw = {}
12        dB = {}
13        train_losses = []
14        term_count = 0
15        step_size = lr
16        self.prepare_minibatches(N)
17        for nt in range(epochs):
18            for ib in range(self.NMB):
19                Z[0], batchY = self.getbatch(ib, trainX, trainY)
20                # (1) Forward pass
21                for i in range(1, num_layers):
22                    b = self.net_params['bias%d'%i]
23                    w = self.net_params['weight%d'%i]
24                    act_f = self.net_params['act%d'%i]
25
26                    A[i] = w.mm(Z[i-1]) + b      # A: M x N
27                    Z[i] = act_f(A[i])          # Z: M x N
28                # end forward pass
29                Yp = Z[i]
30                # (2) Calculate output dE/da
31                delta[last_layer] = Yp - batchY # delta: M x N
32                # (3) Backpropagate. Calc. dE/da for Layer i-1
33                for i in range(last_layer, 1, -1):
34                    b = self.net_params['bias%d'%i]    # Mnnext,1
35                    w = self.net_params['weight%d'%i] # Mnnext,M
36                    act_f = self.net_params['act%d'%(i-1)]
37
38                    sumdw = w.transpose(0, 1).mm(delta[i]) # M,N

```

```

39         if act_f == trelu:
40             delta[i - 1] = tdrelu(A[i - 1]) * sumdw
41         else:
42             assert act_f == trelu
43
44         # (4) Calculate gradient dE/dw and dE/db
45         dEw[i] = delta[i].mm(Z[i-1].transpose(0, 1))
46         dEb[i] = delta[i].mm(torch.ones(self.NB, 1,
47                                         device=delta[i].device))
48     # end backpropagate
49
50     # Calculate gradient dE/dw and dE/db
51     dEw[1] = delta[1].mm(Z[0].transpose(0, 1))
52     dEb[1] = delta[1].mm(torch.ones(self.NB, 1,
53                                     device=delta[1].device))
54
55     # Update parameters w/ Gradient Descent
56     gnorm = 0
57     for i in range(1, num_layers):
58         b = self.net_params['bias%d'%i]
59         w = self.net_params['weight%d'%i]
60         b -= step_size * dEb[i]
61         w -= step_size * dEw[i]
62
63         gnorm += torch.norm(dEb[i])
64         gnorm += torch.norm(dEw[i])
65     # end update parameters
66
67     # Calculate loss at each batch
68     lossn = torch.sum(loss(Yp, batchY), dim=0)
69     train_losses.append(torch.mean(lossn))
70
71     # Check termination condition
72     if gnorm < term:
73         term_count += 1
74         if term_count > term_count_max:
75             print('Reach term. at %d(%d)'%(nt, ib))
76             return train_losses
77     else: # reset term_count
78         term_count = 0
79     # end if term_count

```

```

80         # end ib
81     # end epoch nt
82     return train_losses # Losses per batches
83
84 def predict(self, X):
85     num_layers = self.net_params['layers']
86     Z = X
87     for i in range(1, num_layers):
88         b = self.net_params['bias%d'%i]
89         w = self.net_params['weight%d'%i]
90         act_f = self.net_params['act%d'%i]
91         A = w.mm(Z) + b    # A: M x N
92         Z = act_f(A)       # Z: M x N
93     return Z # M x N

```

---

## แบบฝึกหัด 5.5

ศึกษาโปรแกรมในรายการ 5.4 เปรียบเทียบกับโปรแกรมในรายการ 5.1. จะออกแบบการทดลองเพื่อทดสอบเปรียบเทียบโปรแกรมทั้งสองแบบ ทั้งในเชิงเวลาในการฝึก เวลาในการอนุมาน คุณภาพการฝึก โดยคำนึงถึงปัจจัยประกอบคือ ความลึกและความซับซ้อนของโครงข่ายประสาทเทียมที่เลือกใช้ และจำนวนจุดข้อมูลกับจำนวนมิติของอินพุต. ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

**การคำนวณด้วยจีพีyu.** จุดประสงค์หลักของการใช้ไฟฟอร์ช คือ การที่ไฟฟอร์ชสามารถส่งการคำนวณไปทำในจีพีyuได้ โดยไม่ต้องยุ่งเกี่ยวกับรายละเอียดปลีกย่อยด้วยตัวเอง จึงต้องกำหนดอุปกรณ์จีพีyu และเพื่อจะคำนวณด้วยจีพีyu ตัวแปรแทนเซอร์ทุกตัว จะต้องกำหนดอุปกรณ์เป็นจีพีyu ดังตัวอย่าง เช่น `x = torch.randn(D, N, device='cuda:0', dtype=torch.float)` เมื่อ D และ N เป็นจำนวนส่วนประกอบในลำดับมิติที่หนึ่งและสองตามลำดับ. หรือแม้แต่การแปลงตัวแปรจากนัมไพาร์เรย์ ตัวอย่างเช่น `torchx = torch.from_numpy(datax).float().to('cuda:0')` เมื่อ datax เป็นข้อมูลในรูปแบบนัมไพาร์เรย์ ที่ต้องการ. สังเกตว่า นอกจากการกำหนดอุปกรณ์คำนวณแล้ว ชนิดของข้อมูลก็ต้องกำหนดเป็นเลขทศนิยมขนาดสามสิบสองบิต (32-bit floating point number).

คำสั่ง `torch.cuda.device_count()` ตรวจสอบจำนวนจีพีyuที่สามารถใช้งานได้. คำสั่ง `dev = torch.device('cuda:0')` เตรียมตัวแปรวัตถุ dev สำหรับการอ้างถึงอุปกรณ์จีพีyu และเพื่อจะคำนวณด้วยจีพีyu ตัวแปรแทนเซอร์ทุกตัว จะต้องกำหนดอุปกรณ์เป็นจีพีyu ดังตัวอย่าง เช่น `x = torch.randn(D, N, device='cuda:0', dtype=torch.float)` เมื่อ D และ N เป็นจำนวนส่วนประกอบในลำดับมิติที่หนึ่งและสองตามลำดับ. หรือแม้แต่การแปลงตัวแปรจากนัมไพาร์เรย์ ตัวอย่างเช่น `torchx = torch.from_numpy(datax).float().to('cuda:0')` เมื่อ datax เป็นข้อมูลในรูปแบบนัมไพาร์เรย์ ที่ต้องการ. สังเกตว่า นอกจากการกำหนดอุปกรณ์คำนวณแล้ว ชนิดของข้อมูลก็ต้องกำหนดเป็นเลขทศนิยมขนาดสามสิบสองบิต (32-bit floating point number).

## แบบฝึกหัด 5.6

คล้ายกับแบบฝึกหัด 5.5 จะออกแบบการทดลองเพื่อทดสอบเปรียบเทียบโปรแกรมในรายการ 5.4 เมื่อทำการคำนวณด้วยจีพีyu เปรียบเทียบกับ เมื่อทำการคำนวณด้วยจีพีyu ทั้งในเชิงเวลาในการฝึก เวลาในการอนุมาน คุณภาพการฝึก โดยคำนึงถึงปัจจัยประกอบคือ ความลึกและความซับซ้อนของโครงข่ายประสาทเทียมที่เลือกใช้ และจำนวนจุดข้อมูลกับจำนวนมิติของอินพุต. ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

หมายเหตุ ดังที่ได้อภิปราย การเปลี่ยนอุปกรณ์คำนวณ สามารถทำได้โดยการระบุอุปกรณ์ที่แทนเซอร์ทุกตัว ตัวอย่างเช่น คำสั่งในรายการ 5.3 สามารถเปลี่ยนอุปกรณ์คำนวณเป็นจีพีyu ได้โดยแก้ไขคำสั่งกำหนดอุปกรณ์ในบรรทัดที่หนึ่งเป็น `dev = torch.device('cuda')` และเพิ่มคำสั่ง

```
x = x.to(dev)
y_onehot = y_onehot.to(dev)
ttestx = ttestx.to(dev)
```

เพื่อระบุอุปกรณ์ให้กับแทนเซอร์ของข้อมูลที่จะนำไปคำนวณ.

การหาเกรเดียนต์อัตโนมัติ. นอกจากความสามารถในการเปลี่ยนอุปกรณ์การคำนวณเป็นจีพีyuแล้ว ความสามารถที่สามารถมากอย่างหนึ่งของไฟฟอร์ช คือ การหาค่าเกรเดียนต์ได้โดยอัตโนมัติ (ผ่านกลไกของมอดูล>y อย่าง `torch.autograd`). นั่นหมายถึง เราไม่จำเป็นต้องคำนวณและเตรียมโปรแกรมเพื่อคำนวณเกรเดียนต์เอง ดังเช่น โปรแกรมที่เขียนสำหรับเมธอด `train` ในรายการ 5.4.

การหาเกรเดียนต์อัตโนมัติด้วยไฟฟอร์ช (1) จะต้องระบุในตัวแปรที่ต้องการคำนวณเกรเดียนต์ โดยกำหนด `requires_grad` ของแทนเซอร์ให้ค่าเป็น `True` ตัวอย่างเช่น หากต้องการคำนวณเกรเดียนต์  $\nabla_w E$  ซึ่งเป็นเกรเดียนต์ของค่า  $E$  ต่อตัวแปร  $w$  จะจะระบุที่ตัวแปร  $w$  โดยตรงด้วย `w.requires_grad = True` หรืออาจจะระบุไปพร้อมการกำหนดค่าเริ่มต้น ด้วย

```
w = torch.randn(M, D, requires_grad=True)
```

ก็ได้. การกำหนด `requires_grad` เป็น `True` จะบอกให้ไฟฟอร์ชติดตามการคำนวณที่เกี่ยวข้องกับตัวแปร เพื่อนำมาคำนวณค่าค่าเกรเดียนต์ได้ถูกต้อง.

จากนั้นหลังการคำนวณค่าเป้าหมาย  $E$  เสร็จสิ้น (2) ต้องระบุให้ไฟฟอร์ชคำนวณเกรเดียนต์ ด้วยคำสั่ง เช่น `E.backward()` เมื่อ  $E$  เป็นตัวแปรแทนเนื้อหาค่าเป้าหมาย  $E$ . ค่าเกรเดียนต์  $\nabla_w E$  ที่คำนวณได้ จะเก็บไว้ที่ลักษณะประจำ (attribute) `grad` ของตัวแปร เช่น ในตัวอย่างนี้ คือ `w.grad`. แต่การปรับ

ค่าพารามิเตอร์ ต้องทำการคำนวณเกรเดียโนต์อัตโนมัติ และหลังการปรับค่า ต้องล้างค่า เกรเดียโนต์ออกสำหรับการคำนวณครั้งต่อไป. ตัวอย่างเช่น เมื่อต้องการปรับค่าพารามิเตอร์ อาจทำโดย

```
with torch.no_grad():
    w -= learning_rate * w.grad
    w.grad.zero_()
```

เมื่อ `learning_rate` เป็นค่าอัตราการเรียนรู้.

แบบฝึกหัด 5.7 แสดงตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่เขียนโดยใช้ภาษาเกรเดียโนต์อัตโนมัติ และการเรียกใช้.

### แบบฝึกหัด 5.7

รายการ 5.5 แสดงโปรแกรมโครงข่ายประสาทเทียมที่เขียนด้วยไพทอร์ชและใช้ภาษาเกรเดียโนต์อัตโนมัติ โดยเพื่อลดความซ้ำซ้อน คลาส `tANN2` รับมรดก จากคลาส `tANN1` (รายการ 5.4).

นอกจาก คลาส `tANN2` สังเกตว่า พังก์ชันต่างๆ ในเส้นทางของการแพร่กระจายย้อนกลับ ต้องถูกเขียนใหม่ และการเขียนเมื่อต้อง `backward` ต้องเขียนการคำนวณอนุพันธ์ย้อน เช่น  $\frac{\partial E}{\partial a}$  ซึ่ง  $\frac{\partial E}{\partial a} = \frac{\partial z}{\partial a} \cdot \frac{\partial E}{\partial z} = h'(a) \cdot \frac{\partial E}{\partial z}$ . กลไกของการหาเกรเดียโนต์อัตโนมัติ จะคำนวณส่วน  $\frac{\partial E}{\partial z}$  มาให้. (เปรียบเทียบกับ `drelu` ในรายการ 5.2 ซึ่งคำนวณ  $h'(a)$ . ดูสมการ 3.31 ประกอบ.) ตัวอย่างนี้ แสดงพังก์ชันReLU และพังก์ชันเอกลักษณ์ พังก์ชันอื่น ๆ ที่สามารถทำได้ในลักษณะเดียวกัน.

ตั้งที่ได้อธิบาย ตัวแปรที่ต้องการคำนวณเกรเดียโนต์ต้องถูกระบุอย่างชัดเจน ซึ่งดำเนินการในโปรแกรม `tw_initn2` (เปรียบเทียบกับ `tw_initn1` จากรายการ 5.2).

การใช้งานสามารถทำได้ในลักษณะเดิม ตัวอย่างเช่น

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

net = tw_initn2([1, 16, 1], dev=device)
net['act1'] = auto_relu.apply
net['act2'] = auto_identity.apply

ann = tANN2(net, NB=50, shuffle='once')
train_losses = ann.train(tx, ty, sse, lr=0.2/50, epochs=500)

yp = ann.predict(torch.from_numpy(testx).float().to(device))
yn = yp.to(torch.device('cpu')).data.numpy()
print('test rmse', np.sqrt(np.mean((yn - testy)**2)))
```

เมื่อ **tx** กับ **ty** เป็นอินพุตและเอาต์พุตของข้อมูลฝึกในรูปแบบไฟฟอร์ช และ **testx** กับ **testy** เป็นอินพุตและเอาต์พุตของข้อมูลทดสอบในรูปแบบนัมเบอร์ไพร์.

รายการ 5.5: คลาสโค้ดข่ายประสานที่ยอม สำหรับการหาค่าลดตอน ที่เขียนด้วยไฟฟอร์ชและการหากรเดียวนต์อัตโนมัติ.

```

1 class tANN2(tANN1):
2     def train(self, trainX, trainY, lossf, lr=0.1, epochs=1000,
3               term=1e-8, term_count_max=5):
4         num_layers = self.net_params['layers']
5         last_layer = num_layers - 1
6         out_act = 'act%d' % last_layer
7         _, N = trainX.shape
8         A = {}
9         Z = {}
10        delta = {}
11        dEw = {}
12        dEb = {}
13        train_losses = []
14        term_count = 0
15        step_size = lr
16        self.prepare_minibatches(N)
17
18        for nt in range(epochs):
19            for ib in range(self.NMB):
20                Z[0], batchY = self.getbatch(ib, trainX, trainY)
21                # (1) Forward pass
22                for i in range(1, num_layers):
23                    b = self.net_params['bias%d' % i]
24                    w = self.net_params['weight%d' % i]
25                    act_f = self.net_params['act%d' % i]
26                    A[i] = w.mm(Z[i-1]) + b      # A: M x N
27                    Z[i] = act_f(A[i])          # Z: M x N
28                # end forward pass
29                Yp = Z[i]
30
31                # (2) Calculate loss
32                loss = lossf(Yp, batchY)
33
34                # (3) Calculate gradients with autograd
35                loss.backward()
36
37                # (4) Update parameters w/ Gradient Descent

```

```
38         gnorm = 0
39         for i in range(last_layer, 0, -1):
40             b = self.net_params['bias%d'%i]    # Mnnext,1
41             w = self.net_params['weight%d'%i]   # Mnnext,M
42             with torch.no_grad():
43                 b -= step_size * b.grad
44                 w -= step_size * w.grad
45
46                 gnorm += torch.norm(b.grad)
47                 gnorm += torch.norm(w.grad)
48
49                 b.grad.zero_()
50                 w.grad.zero_()
51             # end update parameters
52             train_losses.append(loss.item())
53
54             # Check termination condition
55             if gnorm < term:
56                 term_count += 1
57                 if term_count > term_count_max:
58                     print('Reach term. at %d(%d)'%(nt, ib))
59                     return train_losses # Losses per batches
60             else: # reset term_count
61                 term_count = 0
62             # end if term_count
63             # end ib
64         # end epoch nt
65         return train_losses # Losses per batches
66
67     def sse(yhat, y):
68         return (yhat - y).pow(2).sum()
69
70     class auto_relu(torch.autograd.Function):
71         @staticmethod
72         def forward(ctx, a):
73             ctx.save_for_backward(a)
74             return a.clamp(min=0)
75
76         @staticmethod
77         def backward(ctx, dEz):
78             a, = ctx.saved_tensors
```

```

79     dEa = dEz.clone()
80     dEa[a < 0] = 0
81     return dEa
82
83 class auto_identity(torch.autograd.Function):
84     @staticmethod
85     def forward(ctx, a):
86         return a
87
88     @staticmethod
89     def backward(ctx, dEz):
90         dEa = dEz.clone()
91         return dEa
92
93 def tw_initn2(Ms, umeansigma=(0,1), dev=torch.device('cpu')):
94     assert len(Ms) >= 2, 'Ms: #units, e.g., M = [2, 8, 3]'
95     num_layers = len(Ms)
96     params = {'layers': num_layers}
97     mu = umeansigma[0]
98     sigma = umeansigma[1]
99     for i, m in enumerate(Ms[1:], start=1):
100         mprev = Ms[i-1]
101         b = torch.randn((m,1), device=dev)
102         w = torch.randn((m,mprev), device=dev)
103         params['bias%d'%i] = b*sigma + mu
104         params['weight%d'%i] = w*sigma + mu
105
106         params['bias%d'%i].requires_grad = True
107         params['weight%d'%i].requires_grad = True
108     return params

```

---

จากโปรแกรมตัวอย่างข้างต้น จะทดสอบโปรแกรม เปรียบเทียบกับการคำนวณเกรเดียนต์ด้วยมือ (รายการ 5.4) โดย ออกแบบการทดลอง เลือกหรือสร้างข้อมูล ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย. หมายเหตุ ตัวอย่างโปรแกรมในรายการ 5.5 มีฟังก์ชัน **identity** และ **sse**. ดังนั้นงานการหาค่าผลโดย สามารถทำได้ทันที แต่งานอื่นๆ เช่น การจำแนกค่าทวิภาค (ต้องการฟังก์ชันซิกมอยด์และครอสເອນໂທຣີ) ซึ่งสามารถทำได้ เช่นเดียวกัน แต่ต้องเตรียมฟังก์ชันที่เกี่ยวข้องให้พร้อมก่อน.

**มอดูลอย่าง nn.** การหาเกรเดียนต์อัตโนมัติ ช่วยลดภาระทั้งการวิเคราะห์เกรเดียนต์ และการเขียนโปรแกรมลงไปมาก. แม้จะลดภาระลงไปมาก แต่การโปรแกรมโครงข่ายประสาทเทียม จากปฏิบัติการพื้นฐาน (ดัง เช่นที่ทำตัวอย่างในรายการ 5.5) ถือเป็นการเขียนโปรแกรมในระดับล่าง ซึ่งเป็นภาระเชิงปัญญา (cognitive burden). เพื่อช่วยลดภาระนี้ รวมถึงช่วยในเรื่องลำดับชั้นของความคิด<sup>9</sup> (hierarchy of abstraction) การประยุกต์ใช้งานโครงข่ายประสาทเทียมลึก จะทำได้มีประสิทธิภาพกว่า เมื่อใช้มอดูลสำเร็จ เช่น มอดูล nn. มอดูล nn มีโครงสร้างและฟังก์ชันสำเร็จต่างๆ สำหรับกลไกที่มีการใช้อย่างแพร่หลาย. ตัวอย่างคำสั่งกำหนดโครงข่ายด้วยไฟฟอร์ช nn แสดงในรายการ 5.6 โดยตัวอย่างคำสั่ง สำหรับการฝึกและทดสอบ แสดงในรายการ 5.7. หมายเหตุ ในรายการ 5.7 โปรแกรมทำ `model.zero_grad()` ในช่วงปลายสมัย (หลังจากทำอย่างอื่นเสร็จ) เพื่อให้เปรียบเทียบได้ตรงมาตรงไปกับการฝึกที่แสดงในรายการ 5.5. อย่างไรก็ตาม ความนิยม คือทำการล้างค่าเกรเดียนต์ช่วงต้นสมัยฝึก (ทำก่อนที่จะทำอย่างอื่น).

รายการ 5.6: ตัวอย่างการใช้ไฟฟอร์ช nn เพื่อกำหนดโครงข่ายประสาทเทียมสำหรับการจำแนกกลุ่ม โดยอินพุตมีสองมิติและเอาต์พุตมีสามมิติ ชั้นชั้นมี 8 และ 16 หน่วยตามลำดับ. ฟังก์ชันกระตุ้นเป็นrelu. ฟังก์ชันกระตุ้นเอาต์พุตเป็นซอฟต์แมกซ์.

---

```

1 device = torch.device('cuda')
2 Ms = [2, 8, 16, 3]
3 model = torch.nn.Sequential(
4     torch.nn.Linear(Ms[0], Ms[1]), torch.nn.ReLU(),
5     torch.nn.Linear(Ms[1], Ms[2]), torch.nn.ReLU(),
6     torch.nn.Linear(Ms[2], Ms[3]),
7     torch.nn.Softmax(dim = 1)).to(device)

```

---

รายการ 5.7: ตัวอย่างคำสั่งการฝึกและทดสอบโครงข่ายประสาทเทียมของรายการ 5.6 โดยใช้ฟังก์ชันสูญเสียเป็นครอสเซอนไฟฟอร์ช. ข้อมูลฝึก อินพุต tdatax เป็นไฟฟอร์ชแทนเซอร์สัดส่วน  $D \times N$  เมื่อ  $D$  เป็นจำนวนมิติ และ  $N$  เป็นจำนวนข้อมูล. ฉลากเฉลยของเอาต์พุต tdatay เป็นไฟฟอร์ชแทนเซอร์สัดส่วน  $1 \times N$ . เนื่องจาก มอดูล nn รับค่าอินพุตในสัดส่วน  $N \times D$  และ torch.nn.NLLLoss รับฉลากเฉลยของเอาต์พุต ในรูปเลขจำนวนเต็ม ในสัดส่วน  $N$  ดังนั้น คำสั่งตัวอย่าง จึงใช้ `tdatax.transpose(0,1)` และ `tdatay[0].long()` สำหรับการป้อนข้อมูลฝึก. ข้อมูลทดสอบ อินพุต ttestx เป็นไฟฟอร์ชแทนเซอร์สัดส่วน  $D \times N'$  เมื่อ  $N'$  คือจำนวนข้อมูลทดสอบ. ส่วนฉลากเฉลยของข้อมูลทดสอบ ttesty เป็นไฟฟอร์ชแทนเซอร์สัดส่วน  $1 \times N'$ . ตัวแปร nepochs และ lr คือจำนวนสมัยฝึก และค่าอัตราเรียนรู้.

---

```

1 loss_fn = torch.nn.NLLLoss()
2 for t in range(nepochs):
3     # (1) Forward pass
4     yhat = model(tdatax.transpose(0,1))
5     loss = loss_fn(torch.log(yhat), tdatay[0].long())

```

---

<sup>9</sup>ลำดับชั้นของความคิด เป็นแนวคิดทางวิศวกรรมคอมพิวเตอร์ และวิศวกรรมทั่วไป (และจริง ๆ แล้วก็รวมถึงกิจกรรมต่าง ๆ ไปจนถึง ภาระธรรมของมนุษยชาติ) ที่จะมองหรือแก้ปัญหาในหลาย ๆ ระดับของรายละเอียด โดยการคิดในระดับบน จะทำการพิจารณารายละเอียดของ ระดับล่างที่เกินความจำเป็นออก. การลดรายละเอียดระดับล่างออก ช่วยลดภาระเชิงปัญญา ทำให้สามารถมองปัญหาไปได้ไกลขึ้น กว้างขึ้น และ เป็นองค์รวมมากขึ้น.

```

6   # (2) Backward pass
7   loss.backward()
8   # (3) Update parameters
9   with torch.no_grad():
10      for param in model.parameters():
11          param -= lr * param.grad
12      # Zero the gradients.
13      model.zero_grad()
14
15 # Test the model
16 yp = model(ttestx.transpose(0,1))
17 _, yc = torch.max(yp, 1)
18 print('Accuracy', torch.mean((yc == ttesty[0].long()).float()))

```

---

โปรแกรมในรายการ 5.6 คำนวณซอฟต์แมกซ์ด้วย `torch.nn.Softmax(dim = 1)` และคำนวณครอสแอนโตรปีด้วย `loss = loss_fn(torch.log(yhat), tdatay[0].long())` โดย `loss_fn = torch.nn.NLLLoss()`. การจัดการคำนวณเช่นนี้ เพื่อให้โปรแกรมในรายการ 5.6 สามารถเปรียบเทียบกับโปรแกรมที่เขียนจากปฏิบัติการพื้นฐานได้สะดวกขึ้น. แต่ในทางปฏิบัติ การคำนวณจะมีประสิทธิภาพมากกว่า หากทำโดยใช้ `nn.LogSoftmax` คู่กับ `nn.NLLLoss` หรือสะดวกกว่า โดยใช้ `nn.CrossEntropyLoss` ซึ่งคำนวณซอฟต์แมกซ์และครอสแอนโตรปีรวมกันเลย.

หมายเหตุ โดยดีฟอลต์ ทั้ง `nn.NLLLoss` และ `nn.CrossEntropyLoss` คำนวณค่าสูญเสียเฉลี่ยต่อของหมู่เล็กอกราก (ดีฟอลต์ เป็น `reduction='mean'`. ดูรายละเอียดการทำงานของแต่ละฟังก์ชันได้จาก <https://pytorch.org/docs/stable/nn.html>) ในเชิงตรรกะการทำางานแล้ว การใช้ผลรวมหรือค่าเฉลี่ย ต่างกันเพียงค่าคงที่ที่นำไปหารค่าฟังก์ชันสูญเสียเท่านั้น. แต่ในทางปฏิบัติ ความต่างนี้มีผลโดยตรง คือ (1) หากเขียนโปรแกรมเอง ผลรวม อาจทำได้อย่างมีประสิทธิภาพมาก ผ่านการจัดการคุณเมทริกซ์ แต่ค่าเฉลี่ยต้องเพิ่มการหารเข้ามา ซึ่งการหานี้ อาจทำได้อย่างมีประสิทธิภาพมาก โดยทำที่อัตราการเรียนรู้. (2) ไม่ว่าจะเขียนโปรแกรมเอง หรือใช้โปรแกรมสำเร็จ การใช้ค่าเฉลี่ย จะให้ผลคำนวณที่ค่อนข้างคงที่ เมื่อเทียบกับจำนวนข้อมูล. นั่นคือ หากใช้ผลรวม เมื่อจำนวนข้อมูลมาก ค่าสูญเสียที่เห็น (ซึ่งคือผลรวมค่าสูญเสีย) จะมีตัวเลขใหญ่. นั่นคือ เมื่อเพิ่มจำนวนข้อมูลฝึกเข้าไป ค่าสูญเสียขณะฝึกที่เห็น จะมีค่ามากขึ้น เมื่อเปรียบเทียบกับการฝึกด้วยข้อมูลน้อย ๆ (ซึ่งไม่ได้แปลว่า การฝึกแย่ลง). แต่หากใช้ค่าเฉลี่ย ค่าสูญเสียที่เห็น (ซึ่งคือค่าสูญเสียเฉลี่ย) จะมีตัวเลขที่อยู่ในระดับเดียวกันได้ ไม่ว่าจะใช้จำนวนจุดข้อมูลฝึกเท่าไร. (3) การเลือกค่าอัตราเรียนรู้ จะทำได้สะดวกกว่าในกรณีค่าเฉลี่ย. นั่นคือ หากพบค่าอัตราเรียนรู้ที่ใช้ได้กับชุด

ข้อมูล เมื่อมีจำนวนข้อมูลน้อยๆ แล้วถ้ามีจำนวนข้อมูลเพิ่มขึ้นมาก การใช้ค่าอัตราเรียนรู้เดิม โดยทั่วไป ก็จะสามารถใช้ได้ดี. แต่หากใช้ผลรวม เมื่อจำนวนข้อมูลเพิ่มขึ้น จะทำให้ผลรวมค่าสูญเสียและผลรวมเกรดียนต์มากขึ้น โดยธรรมชาติ เพราะมีพจน์ที่จะรวมมากขึ้น. ดังนั้น ค่าอัตราเรียนรู้เดิม อาจจะใช้ไม่ดี และอาจจะต้องปรับลดลงเป็นอัตราส่วนตามจำนวนข้อมูลที่เพิ่มขึ้น. การรู้ระลึกถึงประเดิมผลรวมหรือค่าเฉลี่ยนี้ จะช่วยให้การเลือกค่าอัตราเรียนรู้ และการอ่านผลความก้าวหน้าการฝึก ทำได้ดียิ่งขึ้น.

การใช้ `nn.Sequential` แม้จะสะดวก แต่หากต้องการกำหนดทอโพโลยี (topology) การเชื่อมต่อ) ที่อิสระ ยืดหยุ่น และหลากหลายมากขึ้น การใช้ `nn.Module` (ดังแสดงในรายการ 5.8) อาจจะเหมาะสมกว่า. ตัวอย่างทอโพโลยีที่เกินกว่า `nn.Sequential` จะสามารถบรรยายได้ มีมากมาย รวมถึง อเล็กซ์ เน็ต[113] (หัวข้อ 6.5).

การบันทึกแบบจำลองที่ฝึกแล้วก็สามารถทำได้ เช่น

```
torch.save(net.state_dict(), './sav/nnet1.pth')
```

เมื่อ `net` เป็นแบบจำลองที่ต้องการบันทึกค่าเก็บไว้ และ '`./sav/nnet1.pth`' เป็นเส้นทางและชื่อไฟล์ที่บันทึก. การเรียกใช้แบบจำลองที่บันทึกไว้สามารถทำได้ เช่น

```
net = Net().to(device)
net.load_state_dict(torch.load('./sav/nnet1.pth'))
```

เมื่อ `Net()` เป็นโครงสร้างของแบบจำลอง. สังเกตว่า การบักทึกค่า จะบันทึกเฉพาะค่าของพารามิเตอร์ ดังนั้น การเรียกใช้แบบจำลองจึงประกอบด้วยการสร้างตัวแปรตุณของแบบจำลองขึ้นมาใหม่ และกำหนดค่าของพารามิเตอร์ตามค่าที่บันทึกไว้.

รายการ 5.8: ตัวอย่าง แสดงโครงสร้างแบบจำลอง แบบเดียวกับโปรแกรมในรายการ 5.6 แต่ใช้การกำหนดทอโพโลยีเอง (เมท็อด `forward`) โดยคลาสรับมารดจาก `nn.Module` ซึ่งยืดหยุ่นกว่าทอโพโลยีของ `nn.Sequential`.

```
1 class Net(torch.nn.Module):
2     def __init__(self):
3         super(Net, self).__init__()
4         self.fc1 = torch.nn.Linear(2, 8)
5         self.fc2 = torch.nn.Linear(8, 16)
6         self.fc3 = torch.nn.Linear(16, 3)
7
8     def forward(self, x):
9         z1 = torch.relu(self.fc1(x))
10        z2 = torch.relu(self.fc2(z1))
11        z3 = torch.nn.Softmax(dim=1)(self.fc3(z2))
```

---

return z3

---

## แบบฝึกหัด 5.8

จงทดสอบโปรแกรมโครงข่ายประสาทเทียม ที่เขียนโดยใช้โมดูล `nn` เปรียบเทียบกับโปรแกรมที่เขียนการคำนวณกรเดียนต์เอง (เช่น โปรแกรมในรายการ 5.4) โดย ออกแบบการทดลอง เลือกหรือสร้างข้อมูล ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

**โมดูลช่วยจัดหมู่อย่าง `utils.data.DataLoader`.** การทำการฝึกหมู่เล็ก (ดังเช่น โปรแกรมในรายการ 5.1) ก็สามารถดำเนินการได้ด้วยโมดูลอย่าง `utils.data.DataLoader`. การใช้งาน จะต้องสร้างตัวแปรวัตถุของ `DataLoader` โดยการสร้างตัวแปรวัตถุนี้ ต้องกำหนดข้อมูลที่ต้องการเข้าไป และข้อมูลนี้ต้องอยู่ในรูปแบบของ `utils.data.Dataset` ที่ต้องย่างคำสั่งข้างล่างใช้คลาส `MyDataset` (รายการ 5.9) เข้ามาช่วย.

```
mydat = MyDataset()
mydat.assign_data(DX, DY)
dataloader = torch.utils.data.DataLoader(mydat, batch_size=50,
                                         shuffle=True, num_workers=0)
```

เมื่อกำหนดขนาดหมู่เล็กเป็น 50. ตัวแปร `DX` และ `DY` เป็นข้อมูลอินพุตและเอาต์พุต ชนิดไฟฟอร์ชเนนเชอร์ สัดส่วน  $N \times D_x$  และ  $N \times D_y$  ตามลำดับ โดย  $N$  เป็นจำนวนจุดข้อมูล และ  $D_x$  กับ  $D_y$  เป็นมิติของอินพุต และเอาต์พุต.

การเรียกใช้ ก็สามารถทำได้ เช่นเดียวกับตัวแปรวนช้า<sup>10</sup> อื่น ๆ ของไฟรอน เช่นตัวอย่าง

```
for t in range(num_epochs):
    for data in trainloader:
        inputs, labels = data
        yhat = net(inputs)
        loss = loss_fn(yhat, labels[:,0])
        loss.backward()
        with torch.no_grad():
            for param in net.parameters():
                param -= learn_rate * param.grad
        net.zero_grad()
```

<sup>10</sup>ตัวแปรวนช้า (iterable variable) หมายถึง ตัวแปรวัตถุ ที่สามารถให้ค่าสมาชิกของมันออกมากได้ทีละตัว ซึ่งสามารถใช้งานได้สะดวกกับการวนช้าด้วยคำสั่ง `for`. ไฟรอน มีข้อมูลหลายชนิดที่เป็นตัวแปรวนช้า เช่น ลิสต์ ทูเพล และดิกชันนารี.

เมื่อ `num_epochs` และ `learn_rate` เป็นจำนวนสมัยฝึกและอัตราการเรียนรู้ ตามลำดับ. ในตัวอย่างคำสั่งนี้ โปรแกรม `loss_fn` รับผลลัพธ์ในรูปแบบไฟฟอร์ชเนนเซอร์ หนึ่งลำดับชั้น<sup>11</sup> ดังนั้น คำสั่ง `loss = loss_fn(yhat, labels[:,0])` จะต้องจัดผลลัพธ์ให้อยู่ในรูปแบบดังกล่าว.

รายการ 5.9: คลาส `MyDataset` เพื่อใช้กับ `utils.data.DataLoader`. หมายเหตุ หากยังไม่ได้ทำการนำเข้า `torch.utils.data` อาจต้องทำการนำเข้าก่อน.

---

```

1 class MyDataset(torch.utils.data.Dataset):
2     def __init__(self):
3         super(MyDataset, self).__init__()
4         self.datax = None
5         self.datay = None
6
7     def assign_data(self, datX, datY):
8         self.datax = datX
9         self.datay = datY
10
11    def __getitem__(self, index):
12        return self.datax[index,:], self.datay[index,:]
13
14    def __len__(self):
15        return self.datax.shape[0]
```

---

## แบบฝึกหัด 5.9

จากแบบฝึกหัด 3.16 จะเขียนโปรแกรมโดยใช้มอดูล `nn` และการทำเกรเดียนต์อัตโนมัติ พร้อมด้วยจัดการข้อมูลฝึกด้วย `utils.data.DataLoader` และเปรียบเทียบผล กับผลลัพธ์จากแบบฝึกหัด 3.16 สรุปและอภิปราย.

## แบบฝึกหัด 5.10

จากแบบฝึกหัด 5.9 ที่เราดาวน์โหลดข้อมูลเอง เตรียมข้อมูลเอง จัดรูปแบบต่าง ๆ จนข้อมูลสามารถนำเข้าไปใช้กับ ตัวแปรวัตถุของ `DataLoader` ได้. อย่างไรก็ตาม พัฒนาการของการเรียนรู้ของเครื่องและการรู้จำรูปแบบ ก้าวหน้าไปมาก และมีชุดข้อมูลที่มีการศึกษาอย่างกว้างขวาง และนิยมใช้เพื่อเรียนรู้ หรือเพื่อการทดสอบกลไกใหม่ ๆ. สำหรับชุดข้อมูลที่นิยมหลาย ๆ ชุด ไฟฟอร์ชมีกลไกช่วยเหลือ ด้วยมอดูล `torchvision` เพื่อลดภาระในการเตรียมข้อมูลเหล่านี้ลง. ตัวอย่างคำสั่งข้างล่าง เตรียมชุดข้อมูล

---

<sup>11</sup>นั่นคือ เมื่อตรวจสอบสัดส่วน เช่น รันคำสั่ง `labels[:,0].shape` แล้วจะเห็นสัดส่วนเป็น `torch.Size([50])` ไม่ใช่ `torch.Size([50, 1])` ที่หมายถึง ไฟฟอร์ชเนนเซอร์ สองลำดับชั้น.

เอมนิสต์ตั้งแต่ดาวน์โหลด (หากยังไม่มี) ไปจนถึงจัดเข้าตัวแปรวัตถุของ **DataLoader** และพร้อมที่จะถูกเรียกใช้งาน

```
import torchvision
import torchvision.transforms as transforms

transform = transforms.Compose( [transforms.ToTensor(),
                                transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])

trainset = torchvision.datasets.MNIST(root='./data', train=True,
                                      download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=50,
                                          shuffle=True, num_workers=0)
testset = torchvision.datasets.MNIST(root='./data', train=False,
                                      download=True, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=50,
                                         shuffle=False, num_workers=0)
```

เมื่อตัวแปร **trainloader** และ **testloader** คือตัวแปรวัตถุของ **DataLoader** สำหรับข้อมูลฝึกและข้อมูลทดสอบตามลำดับ.

จะเขียนโปรแกรม เพื่อฝึกและทดสอบชุดข้อมูลเอมนิสต์ โดยใช้ข้อมูลโหลดสำเร็จ. สังเกตผล สรุป และอภิปราย. หมายเหตุ การวัดค่าความแม่นยำของข้อมูลทดสอบที่แปลงเป็นหมู่เล็ก จะช่วยลดภาระการใช้หน่วยความจำที่เดียวมาก ๆ ได้.

**มอดูลอย optim.** ดังที่อภิปรายในหัวข้อ 5.5 มีขั้นตอนวิธีมากmany ที่สามารถนำฝึกแบบจำลองได้. มอดูล **optim** จัดเตรียมขั้นตอนวิธีที่นิยมต่าง ๆ ไว้ให้. โดยตัวอย่างคำสั่งต่อไปนี้ แสดงการใช้งาน วิธีลงเกรเดียนต์<sup>12</sup> ที่ใช้อัตราเรียนรู้เป็น 0.001 และโมเมนตัมเป็น 0.0 โดย **net** คือแบบจำลองที่ต้องการฝึก

```
device = torch.device('cuda')
net = torch.nn.Sequential( torch.nn.Linear(784, 8), torch.nn.ReLU(),
                          torch.nn.Linear(8, 10) ).to(device)
loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(net.parameters(), lr=0.001, momentum=0.0)
```

<sup>12</sup>โปรแกรม **optim.SGD** หมายถึง วิธีลงเกรเดียนต์แบบสโตแคตติก (stochastic gradient descent method) ซึ่งมีกลไกหลัก คือ วิธีลงเกรเดียนต์ และเน้นการสุ่มลำดับข้อมูล (ดูหัวข้อ 5.2) นอกจากนั้น **optim.SGD** ยังมีกลไกโมเมนตัมด้วย (หัวข้อ 5.5).

และการฝึกก็สามารถทำได้ดังแสดงในรายการ 5.10. สังเกต โปรแกรมในรายการ 5.10 ล้างค่าเกรเดียนต์ `net.zero_grad()` ตั้งแต่ต้นของลูป.

รายการ 5.10: ตัวอย่างการฝึกแบบจำลอง ด้วยมอดูล `optim` โดย `nepochs` เป็นจำนวนสมัยฝึก. อินพุต `inputs` และ เอาต์พุต `labels` มีสัดส่วน ( $N, 784$ ) และสัดส่วน  $N$  ตามลำดับ เมื่อ  $N$  เป็นขนาดหมู่เล็ก ที่กำหนดไว้กับ `trainloader`.

```

1 train_losses = []
2 for t in range(nepochs):
3     for data in trainloader:
4         net.zero_grad()
5         inputs, labels = data
6         yhat = net(inputs)
7         loss = loss_fn(yhat, labels)
8         loss.backward()
9         optimizer.step()
10        train_losses.append(loss.item())
11    # end for data
12 # end for t

```

---

## แบบฝึกหัด 5.11

จะเลือกหรือสร้างข้อมูล เลือกแบบจำลอง ฝึกโดยใช้การหาค่าดีที่สุดจากมอดูล `optim` ทดสอบ สรุป และอภิปราย.

## แบบฝึกหัด 5.12

จากหัวข้อ 5.3 จงศึกษาและเขียนโปรแกรมสำหรับกลไกการตกลอก จะเลือกหรือสร้างข้อมูล ทำแบบจำลอง โดยใช้เทคนิคการตกลอกที่เขียนขึ้น ทดสอบ สังเกตผล สรุป และอภิปราย. หมายเหตุ การใช้การตกลอก อาจทำให้ต้องการแบบจำลองที่ใหญ่ขึ้น(ซับซ้อนขึ้น) จากการทำแบบจำลองที่ไม่ใช้การตกลอก รวมถึง อาจทำให้ต้องการจำนวนสมัยฝึกที่มากขึ้น. คำใบ้ การฝึกอาจทำได้ช้าลง แต่ให้สังเกตการลดลงของค่าฟังก์ชัน สูญเสีย และในการทดสอบ อย่าลืมชดเชย การตกลอก. แบบฝึกหัดนี้ ต้องการให้ได้ทดลองฝึกเขียนโปรแกรม ด้วยตนเอง แต่หากต้องการ แบบฝึกหัด 5.13 แสดงตัวอย่างโปรแกรม.

## แบบฝึกหัด 5.13

จงศึกษา และเปรียบเทียบโปรแกรมที่เขียนขึ้นสำหรับแบบฝึกหัด 5.12 กับโปรแกรมตัวอย่าง (รายการ 5.11) และการนำไปใช้ในแบบจำลอง แสดงในรายการ 5.12. การฝึกและทดสอบ แสดงในรายการ 5.13) ทั้งวิธีการเขียน และพฤติกรรมการทำงาน.

สังเกตว่า ถึงแม้คือ การตกออก แต่ค่าความน่าจะเป็น (ตัวแปร **oneprob** ในรายการ 5.11) ซึ่งจะรับค่า **0.8** และ **0.5** ในรายการ 5.12) ระบุถึงความน่าจะเป็นของการคงอยู่. ข้อควรระวัง การใช้งานมอดูล **สำเร็จ ควรศึกษาตรวจสอบพุทธิกรรมการทำงานให้ชัดเจนก่อน.**

**รายการ 5.11:** ตัวอย่างโปรแกรมการตกออก. สำหรับอนุพันธ์  $\frac{\partial E}{\partial z'} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial z'}$  เมื่อ  $E$  คือค่าฟังก์ชันสูญเสีย และ  $z'$  กับ  $z$  คือ ค่าหน่วยย่อหยักทำการตกออก และก่อนทำการตกออก ตามลำดับ. ค่า  $\frac{\partial z}{\partial z'} = m$  เมื่อ  $m$  คือ หน้ากาก หรือค่าสัมประสิทธิ์ของการตกออก ( $m \in \{0, 1\}$  และ  $m$  มีการแจกแจงแบบแบร์นูลลี่ และความน่าจะเป็นของค่าหนึ่ง แทนด้วยอาร์กิวเม้นต์ **onprob**).

```

1  class mdropout(torch.autograd.Function):
2      @staticmethod
3      def forward(ctx, z, onprob=0.5):
4          d = torch.distributions.Bernoulli(torch.tensor([onprob]))
5          mask = d.sample(sample_shape=z.shape).view(z.shape)
6          mask = mask.to(z.device)
7          ctx.save_for_backward(mask)
8          return mask * z
9
10     @staticmethod
11     def backward(ctx, dEzm):
12         mask, = ctx.saved_tensors
13         dEz = mask * dEzm.clone()
14         return dEz, None, None

```

**รายการ 5.12:** ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้การตกออกที่เขียนขึ้นเอง. สังเกต (1) การตกออก ทำเฉพาะตอนฝึก (2) การอนุมาน ต้องชดเชยการตกออก. โปรแกรม ใช้กลไกของ **nn.Module** ที่มีสถานะ **self.training** ในการตรวจสอบว่า กำลังฝึก หรือกำลังใช้งานอนุมาน.

```

1  class Net(torch.nn.Module):
2      def __init__(self):
3          super(Net, self).__init__()
4          self.do0 = mdropout.apply
5          self.fc1 = torch.nn.Linear(784, 16)
6          self.do1 = mdropout.apply
7          self.fc2 = torch.nn.Linear(16, 10)
8
9      def forward(self, x):
10         z2 = None
11         if self.training:
12             xm = self.do0(x, 0.8, 1)
13             z1 = torch.relu(self.fc1(xm))
14             z1m = self.do1(z1, 0.5, 1)
15             z2 = self.fc2(z1m)

```

```

16     else:
17         xm = 0.8 * x
18         z1 = torch.relu(self.fc1(xm))
19         z1m = 0.5 * z1
20         z2 = self.fc2(z1m)
21     return z2

```

---

รายการ 5.13: ตัวอย่างการฝึกและทดสอบโครงข่ายประสาทเทียมที่ใช้การตอกออกที่เขียนขึ้นเอง. สังเกต เพื่อรับรู้ว่า การคำนวนเป็นการฝึก หรืออนุมาน จะใช้ คำสั่ง `net.train()` และคำสั่ง `net.eval()` ซึ่งคำสั่งทั้งสอง จะเข้าไปเปลี่ยนค่า `net.training` ที่โปรแกรมภายในคลาส `Net` สามารถนำไปตรวจสอบได้.

```

1 loss_fn = torch.nn.CrossEntropyLoss()
2 device = torch.device('cuda')
3 net = Net().to(device)
4 net.train() # Set mode to 'train' (net.Training = True)
5 optimizer = torch.optim.SGD(net.parameters(), lr=1e-4)
6 nepochs = 50
7 train_losses = []
8 for t in range(nepochs):
9     for i, data in enumerate(trainloader):
10         net.zero_grad()
11         inputs, labels = data
12         yhat = net(inputs)
13         loss = loss_fn(yhat, labels)
14         loss.backward()
15         optimizer.step()
16         train_losses.append(loss.item())
17     # end for data
18 # end for t
19
20 # Test
21 net.eval() # Set mode to 'eval' (net.Training = False)
22 correct = 0
23 num = 0
24 for data in testloader:
25     inputs, labels = data
26     yp = net(inputs)
27     _, yc = torch.max(yp, 1)
28     correct += torch.sum((yc == labels).float())
29     num += len(y)
30 print('Accuracy', correct/num)

```

---

## แบบฝึกหัด 5.14

ดังที่อภิรายในหัวข้อ 5.3 การตกลอก อาจดำเนินการชดเชย โดยใช้การหารค่าความน่าจะเป็น ออกจากค่าหน่วยอย่าง -tonfik แทนการคูณเข้า ขณะทำการอนุมาน. ตัวอย่างโปรแกรมในรายการ 5.14 แสดงโปรแกรมการตกลอก ที่เขียนโดยใช้การหาร ตอนฟิก เพื่อชดเชยค่าที่ตกลอกไป. จงเปรียบเทียบความต่างกับโปรแกรมในรายการ 5.11 ทั้งวิธีการเขียน และพฤติกรรมการทำงาน.

**รายการ 5.14:** ตัวอย่างโปรแกรมการตกลอก โดยการหารค่าความน่าจะเป็น ขณะฟิก เพื่อชดเชยการตกลอก. หมายเหตุ เพื่อให้แนวคิดตระกรากการทำงานชัดเจน โปรแกรมในตัวอย่างใช้การหาร. ในทางปฏิบัติ การคูณด้วย 1.25 และ 2 จะให้ประสิทธิภาพและเสถียรภาพดีกว่า การหารด้วย 0.8 และ 0.5 ตามลำดับ.

```

1  class Net(torch.nn.Module):
2      def __init__(self):
3          super(Net, self).__init__()
4          self.do0 = nn.Dropout(p=0.8)
5          self.fc1 = torch.nn.Linear(784, 16)
6          self.do1 = nn.Dropout(p=0.5)
7          self.fc2 = torch.nn.Linear(16, 10)
8
9      def forward(self, x):
10         z2 = None
11         if self.training:
12             xm = self.do0(x)
13             z1 = torch.relu(self.fc1(xm))
14             z1m = self.do1(z1)
15             z2 = self.fc2(z1m)
16         else:
17             xm = x
18             z1 = torch.relu(self.fc1(xm))
19             z1m = z1
20             z2 = self.fc2(z1m)
21         return z2

```

**การตกลอก ด้วย nn.Dropout.** มодูล nn มีมอดูลอย่างสำหรับทำการตกลอก คือ **nn.Dropout**. ตัวอย่างคำสั่งข้างล่าง แสดงการใช้ **nn.Dropout** เพื่อใช้งานกลไกการตกลอก ในลักษณะเดียวกับโปรแกรมในรายการ 5.12.

```

class Net(torch.nn.Module):
    def __init__(self):

```

```

super(Net, self).__init__()
self.do0 = torch.nn.Dropout(p=0.2)
self.fc1 = torch.nn.Linear(784, 16)
self.do1 = torch.nn.Dropout(p=0.5)
self.fc2 = torch.nn.Linear(16, 10)

def forward(self, x):
    xm = self.do0(x)
    z1 = torch.relu(self.fc1(xm))
    z1m = self.do1(z1)
    z2 = self.fc2(z1m)
    return z2

```

สังเกตการใช้ `nn.Dropout` ไม่ต้องกำหนดการคำนวณแยกระหว่างการฝึก และการอนุมาน (เช่นที่ต้องทำในรายการ 5.12) เพราะว่า `nn.Dropout` มีกลไกภายในที่จัดการเรื่องนี้ให้.

นอกจากนั้น `nn.Dropout` รับความน่าจะเป็นที่จะตัดออก (เปรียบเทียบกับ โปรแกรมในรายการ 5.11 ที่เป็นความน่าจะเป็นของการคงอยู่) ดังนั้น ณ ที่นี่ `self.do0` สำหรับอินพุตจึงใช้ `p=0.2` ซึ่งคือ โอกาสตัดออกเป็น 0.2 (หรือโอกาสคงอยู่ 0.8).

### แบบฝึกหัด 5.15

จากแบบฝึกหัด 5.12 จะทำแบบจำลองที่ใช้กลไกการตัดออก โดยใช้ `nn.Dropout` เปรียบเทียบโปรแกรม การทำงาน และผลการทำงาน สรุปผล และอภิปราย.

### แบบฝึกหัด 5.16

จงศึกษาการทำงานและผลของการใช้การตัดออก โดยเปรียบเทียบกับ (1) การไม่ใช้เทคนิคการตัดออก และ (2) การทำค่า  $\bar{z}$  หนักเสื่อม. ทดสอบกับข้อมูลที่มีความยากต่าง ๆ กัน มีปริมาณข้อมูลต่าง ๆ กัน และเมื่อแบบจำลองมีความซับซ้อนต่าง ๆ กัน. สังเกตผล สรุปและอภิปราย.

### แบบฝึกหัด 5.17

จงเขียนโปรแกรมโครงข่ายประสาทเทียม ที่มีชั้นสัญญาณรับกวน ที่รับค่าหน่วยย่อ  $z$  เป็นอินพุต และให้ค่า  $z'$  เป็นเอาต์พุต โดย  $z' = m \odot z$  เมื่อ  $m$  เป็นเมตริกซ์ขนาดเดียวกับ  $z$  และแต่ละส่วนประกอบ  $m \sim \mathcal{N}(1, \sigma)$  และ  $\sigma$  เป็นอภิมานพารามิเตอร์ กำหนดจากผู้ใช้.

ออกแบบการทดลอง เพื่อทดสอบประสิทธิภาพการใช้ชั้นสัญญาณรบกวน เปรียบเทียบกับการตอกอก ทั้งเรื่องการฝึก และผลของแบบจำลองที่ฝึกได้. ดำเนินการทดลอง สังเกตผล สรุปและอภิปราย. ศึกษางานวิจัยของศรีวาราстваและคณะ[189] อภิปรายผลที่ได้ เปรียบเทียบกับผลจากศรีวาราстваและคณะ.

รายการ 5.15 แสดงตัวอย่างโปรแกรม. สังเกตว่า การใช้ชั้นสัญญาณรบกวน สะดวกกว่าการตอกอก ในแบบที่ไม่ต้องทำการซัดเซยขณะใช้งานอนุมาน.

รายการ 5.15: ตัวอย่างโปรแกรมชั้นสัญญาณรบกวน.

---

```

1  class mnoise(torch.autograd.Function):
2      @staticmethod
3      def forward(ctx, z, sigma=1):
4          d = torch.distributions.normal.Normal(torch.tensor([1.0]),
5                                              torch.tensor([sigma]))
6          mask = d.sample(sample_shape=z.shape).view(z.shape)
7          mask = mask.to(z.device)
8          ctx.save_for_backward(mask)
9          return mask * z
10
11     @staticmethod
12     def backward(ctx, dEzm):
13         mask, = ctx.saved_tensors
14         dEz = mask * dEzm.clone()
15         return dEz, None, None
16
17 class Net(nn.Module):
18     def __init__(self):
19         super(Net, self).__init__()
20         self.do0 = mnoise.apply
21         self.fc1 = torch.nn.Linear(784, 16)
22         self.do1 = mnoise.apply
23         self.fc2 = torch.nn.Linear(16, 10)
24
25     def forward(self, x):
26         z2 = None
27         if self.training:
28             xm = self.do0(x, 1.0)
29             z1 = torch.nn.ReLU()(self.fc1(xm))
30             z1m = self.do1(z1, 1.0)
31             z2 = self.fc2(z1m)
32         else:

```

```

33     xm = x
34     z1 = torch.nn.ReLU()(self.fc1(xm))
35     z1m = z1
36     z2 = self.fc2(z1m)
37
38     return z2

```

---

## แบบฝึกหัด 5.18

**การคำนวณค่าความน่าจะเป็น** แม้การหาค่าผลตอบแทนจะเน้นค่าที่วิภาค และการคำนวณกลุ่ม เป็นกลุ่ม ภาระกิจที่มีการใช้งานมากที่สุด แต่การใช้งานโครงข่ายประสาทเทียม ไม่ได้จำกัดอยู่แต่เฉพาะกลุ่มภาระกิจ ที่นิยมเหล่านี้ โครงข่ายประสาทเทียม สามารถประยุกต์ใช้งานได้กว้างขวาง<sup>13</sup> หลักการของวิธีค่าฟังก์ชัน ควรจะเป็นสูงสุด (maximum likelihood) เป็นแนวทางหนึ่งที่ทั่วไปมากพอ ที่สามารถใช้ออกแบบฟังก์ชันจุด ประสงค์สำหรับภาระกิจต่าง ๆ ที่ต้องการได้.

หลักการของวิธีค่าฟังก์ชันควรจะเป็นสูงสุด คือ หากกำหนดให้  $\mathbf{X}$  และ  $\mathbf{Y}$  เป็นข้อมูลที่สนใจ และ  $p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta})$  เป็น ค่าประมาณความน่าจะเป็น โดย  $\boldsymbol{\theta}$  เป็นพารามิเตอร์แล้ว ค่าของพารามิเตอร์  $\boldsymbol{\theta}$  สามารถหาได้จาก

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta}) \quad (5.27)$$

เมื่อ  $\boldsymbol{\theta}^*$  คือค่าพารามิเตอร์ที่ดีที่สุด (สำหรับแบบจำลองและข้อมูลที่มี).

แบบฝึกหัดนี้ เราจะศึกษาการทำโครงข่ายประสาทเทียมสำหรับท่านายการแจกแจงของข้อมูล. นั่นคือ จากที่เคยใช้โครงข่ายประสาทเทียม  $f$  ท่านายค่าเออต์พุต  $y$  จากอินพุต  $x$  แบบฝึกหัดนี้จะใช้โครงข่ายประสาท เทียมท่านายการแจกแจงของเออต์พุต  $y$  จากอินพุต  $x$ . แนวทางคือ แทนที่จะใช้โครงข่ายประสาทเทียมท่านาย ค่าความน่าจะเป็น<sup>14</sup>  $p(y|x)$  โดยตรง เราจะใช้โครงข่ายประสาทเทียมท่านาย  $\boldsymbol{\theta}(x)$  ซึ่งนำไปใช้คำนวณค่า ประมาณความน่าจะเป็น  $p(y; \boldsymbol{\theta}(x)) \approx p(y|x)$  อิกต่อหนึ่ง.

แบบฝึกหัดนี้ การประมาณความน่าจะเป็น  $p(y; \boldsymbol{\theta}(x))$  จะคำนวณด้วย แบบจำลองความหนาแน่นผสม (mixture density model). แบบจำลองความหนาแน่นผสม เป็นแบบจำลองที่ว้าไปในการประมาณค่าเออต์-พุต จากอินพุต โดยรวมค่าประมาณจากส่วนผสมต่าง ๆ เข้าด้วยกัน อาจมองว่า แบบจำลองความหนาแน่นผสม มีพื้นฐานจากการกลบรวม และกฎผลลัพธ์ของทฤษฎีเบลส์ได้ อันคือ  $p(y|x) = \sum_{i=1}^M p(y|c=i)p(c=i|x)$

<sup>13</sup>เนื้อหา ในแบบฝึกหัดนี้ได้รับอิทธิพลหลัก ๆ จากคู่มือเพโลและคณะ[76, §6.2.2.4].

<sup>14</sup>เพื่อให้เนื้อหาความทั่วไป และไม่เย็นเชื่อมาก ในที่นี้จะใช้คำว่า ความน่าจะเป็น ในความหมายของความน่าจะเป็น หรือในกรณีที่ตัวแปรที่เกี่ยวข้องเป็นตัวแปรสุ่มต่อเนื่อง จะหมายถึง ความหนาแน่นความน่าจะเป็น. ดูหัวข้อ 2.2 เพิ่มเติม สำหรับความต่างระหว่างความน่าจะเป็น และความหนาแน่นความน่าจะเป็น.

เมื่อ  $c = i$  แทนส่วนผสม  $i$  และ  $M$  คือจำนวนส่วนผสมทั้งหมด. ส่วนผสม  $c = i$  อาจมองเสมอว่าเป็นสถานะภายในของความสัมพันธ์ระหว่าง  $x$  กับ  $y$  ได้. ค่า  $p(y|c = i)$  ถูกประมาณด้วยความหนาแน่นของ การแจกแจงเกาส์เซียน. ดังนั้น สรุปคือ แบบจำลองความหนาแน่นผสม คำนวน

$$p(y; \boldsymbol{\theta}(x)) = \sum_{i=1}^M p(c = i|x) \cdot \mathcal{N}(y; \mu_i(x), \sigma_i(x)) \quad (5.28)$$

เมื่อ  $p(c = i|x)$  แทนความน่าจะเป็นของส่วนผสม  $i$  และ  $\mathcal{N}(y; \mu_i(x), \sigma_i(x))$  เป็นค่าความหนาแน่น ความน่าจะเป็นของการแจกแจงเกาส์เซียน ที่มีค่าเฉลี่ย  $\mu_i(x)$  กับค่าเบี่ยงเบนมาตรฐาน  $\sigma_i(x)$ . แบบจำลอง เก้าส์เซียนผสม สามารถใช้ประมาณค่าความน่าจะเป็นจากการแจกแจงได้ ๆ ได้ หากมีจำนวนส่วนผสมเพียง พ. จำนวนส่วนผสม  $M$  เป็นอภิมานพารามิเตอร์ของแบบจำลอง.

สังเกต รูปแบบสมการ 5.28 เขียนสำหรับกรณีเอาร์พูตมิติเดียว ( $y \in \mathbb{R}$ ). กรณีทั่วไป ก็สามารถคำนวน ได้ในลักษณะเดียวกัน นั่นคือ  $p(\mathbf{y}; \boldsymbol{\theta}(\mathbf{x})) = \sum_{i=1}^M p(c = i|\mathbf{x}) \cdot \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_i(\mathbf{x}), \boldsymbol{\Sigma}_i(\mathbf{x}))$ .

สำหรับจุดข้อมูลที่  $n^{th}$  ค่าความน่าจะเป็น  $p(y_n|x_n) \approx p(y_n; \boldsymbol{\theta}(x_n))$  และด้วยสมมติฐานໄอ.ไอ.ดี. (i.i.d. ย่อจาก independent and identically distributed random variables ซึ่ง ณ ที่นี่ หมายถึง สมมติฐานว่าจุดข้อมูลแต่ละจุดเป็นอิสระต่อกัน และมีการแจกแจงเหมือนกัน) จะได้ว่า

$$p([y_1, y_2, \dots, y_N] | [x_1, x_2, \dots, x_N]) = \prod_{n=1}^N p(y_n | x_n)$$

เพื่อความสะดวกในการคำนวน ค่าลอกการทึบของฟังก์ชันควรจะเป็น (log likelihood) จะถูกนิยมมากกว่า. นอกจากนั้น สำหรับการฝึกแบบจำลอง นิยมวงกรอบเป็นปัญหาค่าน้อยที่สุด ค่าฟังก์ชันสูญเสีย สามารถกำหนดเป็น ค่าลบลอกการทึบของฟังก์ชันควรจะเป็น (negative log likelihood). ดังนั้น ค่าฟังก์ชันสูญเสีย สามารถนิยามได้เป็น

$$\begin{aligned} \text{loss} &= -\log \prod_{n=1}^N p(y_n | x_n) \\ &= -\sum_n \log p(y_n | x_n) \end{aligned} \quad (5.29)$$

$$= -\sum_n \log \sum_{i=1}^M p(c = i | x_n) \cdot \mathcal{N}(y_n; \mu_i(x_n), \sigma_i(x_n)) \quad (5.30)$$

สมการ 5.30 ได้จากการใช้แบบจำลองความหนาแน่นผสม. โครงข่ายประสาทเทียม สามารถใช้เพื่อ ประมาณ พารามิเตอร์ของแบบจำลองความหนาแน่นผสม  $\boldsymbol{\theta} = [p(c = i|x), \mu_i(x), \sigma_i(x)]^T$  สำหรับ  $i = 1, \dots, M$ .

จะศึกษาการทำโครงการข่ายประชาทเที่ยม สำหรับประมาณการแจกแจง ซึ่งมีรายละเอียด คือ (1) จงสร้างข้อมูล  $\{x_n, y_n\}$  สำหรับ  $n = 1, \dots, N$  โดยกำหนดความสัมพันธ์ระหว่างตัวแปรต้น  $x_n$  และตัวแปรตาม  $y_n$  ดังนี้

(a) สำหรับแต่ละค่าของตัวแปรต้น  $x$  ตัวแปรตาม  $y$  แสดงออกได้สองลักษณะ.

(b) ลักษณะแรก  $y \sim \mathcal{N}(\mu_0(x), \sigma_0(x))$

โดย  $\mu_0(x) = 0.05x^2 + 4$  และ  $\sigma_0(x) = 0.2 + \log(1 + \exp(x - 5))$ .

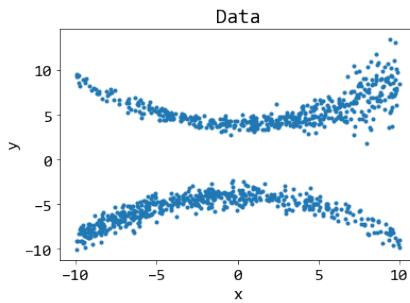
(c) ลักษณะที่สอง  $y \sim \mathcal{N}(\mu_1(x), 0.5)$  โดย  $\mu_1(x) = -0.05x^2 - 4$ .

(d) โอกาสที่  $y$  จะแสดงออกในลักษณะแรก เป็น  $0.25 + \frac{0.5}{1 + \exp(-0.4x)} \times 100\%$ . นอกจากนั้น  $y$  จะแสดงออกในลักษณะที่สอง.

(2) จงกำหนดโครงการข่ายประชาทเที่ยม ฝึก ทดสอบ สังเกตผล สรุป และอภิปราย.

จากข้อกำหนดของข้อมูล สังเกตว่า (ก) ลักษณะข้อมูลเป็นไปตามแบบจำลองความหนาแน่นผล จำนวนลักษณะแสดงออก คือจำนวนส่วนผสม นั่นคือ  $M = 2$ . (ข) ความน่าจะเป็นของลักษณะแรก  $p_0 = p(c = 0|x) = 0.25 + \frac{0.5}{1 + \exp(-0.4x)}$  และความน่าจะเป็นของลักษณะที่สอง  $p_1 = p(c = 1|x) = 1 - p(c = 0|x)$ . ทั้ง  $p_0$  และ  $p_1$  เป็นพึ่งกันของ  $x$ . (ค) ลักษณะแรก ทั้งค่าเฉลี่ย  $\mu_0$  และค่าเบี่ยงเบนมาตรฐาน  $\sigma_0$  เป็นพึ่งกันของ  $x$ . ส่วนลักษณะที่สอง ค่าเฉลี่ย  $\mu_1$  เป็นพึ่งกันของ  $x$  แต่ค่าเบี่ยงเบนมาตรฐาน  $\sigma_1$  เป็นค่าคงที่. (ง) ตัวแปรต้น  $x \in \mathbb{R}$  ดังนั้น โครงการข่ายประชาทเที่ยมรับอินพุตหนึ่งมิติ. (จ) พารามิเตอร์ของแบบจำลองความหนาแน่นผล จะมีทั้งหมด 6 ตัว ได้แก่  $\boldsymbol{\theta} = [p_0, p_1, \mu_0, \mu_1, \sigma_0, \sigma_1]^T$  ซึ่งทั้ง 6 ค่านี้ จะคำนวณมาจากโครงการข่ายประชาทเที่ยม. ดังนั้น โครงการข่ายประชาทเที่ยมให้อาชีพพุทธภูมิ. สรุปคือ โครงการข่ายประชาทเที่ยม  $f: \mathbb{R} \mapsto \mathbb{R}^6$ .

รูป 5.18 แสดงตัวอย่างจุดข้อมูลที่สร้างตามข้อกำหนด. สังเกตว่า ที่ตัวแปรต้น  $x$  แต่ละค่า ตัวแปรตาม จะแสดงออกเป็นสองลักษณะ ซึ่งทั้งสองลักษณะมีการแจกแจงแบบสุ่ม โดยลักษณะแรก มีค่ามากกว่าลักษณะที่สอง และแนวโน้มข้อมูลจะโค้งขึ้น ในขณะที่ลักษณะที่สอง แนวโน้มข้อมูลจะโค้งลง. (เกี่ยวข้องกับ  $\mu_0$  และ  $\mu_1$ .) จุดข้อมูลลักษณะแรก จะเบาบาง (มีสัดส่วนจำนวนจุดน้อยกว่า) จุดข้อมูลลักษณะที่สอง ในช่วงค่า  $x < 0$ . จุดข้อมูลลักษณะที่สอง ดูเบาบางลง เมื่อ  $x > 0$ . (เกี่ยวข้องกับ  $p_0$  และ  $p_1$ .) การแจกแจงของจุดข้อมูลลักษณะที่สอง ดูคงที่ตลอดช่วงค่าของ  $x$  แต่จุดข้อมูลลักษณะแรก ดูเหมือนมีการแจกแจงเพิ่มขึ้นอย่างเห็นได้ชัดในช่วง  $x$  มีค่ามาก ๆ. (เกี่ยวข้องกับ  $\sigma_0$  และ  $\sigma_1$ .)



รูปที่ 5.18: ตัวอย่างจุดข้อมูล สำหรับโครงข่ายประสาทเทียม เพื่อทำนายการแยกจำ. แกนนอน แสดงค่าตัวแปรต้น  $x$  และแกนตั้ง แสดงค่าตัวแปรตาม  $y$ .

ตัวอย่างโปรแกรม สำหรับสร้างข้อมูล แสดงในรายการ 5.16. โปรแกรมเขียนเป็นคลาส และเมท็อดที่ใช้สร้างข้อมูล คือ `sim_y` ซึ่งจะสร้างข้อมูลตัวแปรตาม ขึ้นมาจากข้อมูลตัวแปรต้นที่รับเข้าไป. โปรแกรมสามารถทดสอบได้ง่ายๆ ด้วยคำสั่ง

```
r = relation()
xs = np.linspace(-10, 10, 1000)
ys = r.sim_y(xs)
```

ซึ่งค่า `xs` และ `ys` สามารถนำไปวาดกราฟ เพื่อดูความสมพันธ์ได้.

รายการ 5.16: โปรแกรม `relation` เพื่อสร้างข้อมูลสำหรับการทำนายการแยกจำ

---

```

1 class relation:
2     def __init__(self):
3         self.num_modes = 2
4         self.mode_chances =[lambda x:0.25+0.5/(1+np.exp(-0.4*x)),
5                             lambda x: 1-(0.25+0.5/(1 + np.exp(-0.4*x)))]
6         self.fmu_y = [lambda x: 0.05*x**2 + 4,
7                       lambda x: -0.05*x**2 - 4]
8         self.fsigma_y = [lambda x: 0.2 + np.log(1 + np.exp(x-5)),
9                           lambda x: 0.5*np.ones(x.shape)]
10
11     def sim_y(self, xs):
12         N = len(xs)
13         p = np.random.uniform(0,1,N)
14         ys = np.array([])
15         for n in range(N):
16             xn = xs[n]
17             mode = self.num_modes-1
18             for i in range(self.num_modes-1):
```

```

19         pi = self.mode_chances[i](xn)
20         p[n] -= pi
21         if p[n] < 0:
22             mode = i
23             break
24         mun = self.fmu_y[mode](xn).reshape((-1,))
25         D = mun.shape[0]
26         sigman = self.fsigma_y[mode](xn).reshape((D,D))
27         yn = np.random.multivariate_normal(\n
28                                         mun,sigman,1).item()
29         ys = np.r_[ys, yn]
30     return ys

```

---

ตัวอย่างโปรแกรมคำนวณฟังก์ชันสุญเสีย (คำนวณสมการ 5.30) แสดงในรายการ 5.17. สังเกต (1) แบบจำลองความหนาแน่นผลสม ถูกโปรแกรมเป็นส่วนหนึ่งของฟังก์ชันสุญเสีย. (2) ค่าความหนาแน่น  $\mathcal{N}(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-0.5\left(\frac{y-\mu}{\sigma}\right)^2\right)$  คำนวณโดยปฏิบัติการพื้นฐาน ไม่ได้ใช้ฟังก์ชันสำเร็จ.

รายการ 5.17: โปรแกรมคำนวณค่าลับลอกการทิมของฟังก์ชันควรจะเป็น ที่ใช้แบบจำลองความหนาแน่นผลสมเป็นพื้นฐาน. โปรแกรมรับค่า  $yhat$  ที่อยู่ในรูปป้าเรอนทูเพิล (`dmode`, `dmu`, `dsigma`) เมื่อ `dmode`, `dmu`, และ `dsigma` เป็นค่าทำนาย  $[p_0, p_1]^T$ ,  $[\mu_0, \mu_1]^T$ , และ  $[\sigma_0, \sigma_1]^T$  ตามลำดับ. ส่วนเฉลย  $Y$  เป็นค่าตัวแปรตามของจุดข้อมูลต่าง ๆ. ค่า `eps` ใช้สำหรับป้องกัน `torch.log` ไม่ให้ค่าเป็น `-inf`.

---

```

1 def loss1(dhat, Y):
2     dmode, dmu, dsigma = dhat
3     eps = 1e-45
4     likelihood = dmode*torch.exp(\n
5         -0.5*((Y.view(-1,1) - dmu)/dsigma)**2)
6     likelihood /= dsigma*torch.sqrt(\n
7         torch.Tensor([2.0*np.pi])).to(Y.device)
8     loglikelihood = torch.log(likelihood.sum(dim=1) + eps).sum()
9     NLL = -loglikelihood
10    return NLL

```

---

ค่าเออต์พุตของโครงข่าย  $\hat{y} = [p_0, p_1, \mu_0, \mu_1, \sigma_0, \sigma_1]^T$  มีลักษณะต่าง ๆ กัน. ค่า  $p_0$  และ  $p_1$  เป็นค่าความน่าจะเป็น ซึ่ง  $p_i \in [0, 1]$  และ  $\sum_i p_i = 1$ . ดังนั้น โปรแกรมตัวอย่าง (รายการ 5.18) ใช้ฟังก์ชันซอฟต์แมกซ์<sup>15</sup> สำหรับ  $[p_0, p_1]^T$  เพื่อคุมเงื่อนไขนี้. ค่า  $\mu_0$  และ  $\mu_1$  ไม่มีข้อจำกัดอะไร. ค่า  $\sigma_0$  และ  $\sigma_1$

<sup>15</sup>เนื่องจาก  $p_0$  และ  $p_1$  ไม่ได้เปรียบเทียบกับรหัสหนึ่งร้อน การใช้ซอฟต์แมกซ์ อาจเพิ่มการคำนวณโดยไม่จำเป็น และอาจส่งผลเสียต่อเสียงภาพและคุณภาพของตีกอึกด้วย. กรณีนี้ การทำอนุรูปให้เหมาะสม อาจจะเป็นทางเลือกที่ดีกว่า.

เป็นค่าเบี่ยงเบนมาตรฐาน ซึ่ง  $\sigma_i > 0$ . โปรแกรมตัวอย่าง ใช้ฟังก์ชันบวกอ่อน  $h(a) = \log(1 + \exp(a))$  สำหรับ  $[\sigma_0, \sigma_1]^T$  เพื่อคุณเงื่อนไขนี้.

รายการ 5.18: โปรแกรมโครงข่ายประสาทเทียม เพื่อทำนายการแยกแยะ. โครงข่ายสองชั้น รับอินพุตหนึ่งมิติ ใช้จำนวนหน่วยช้อนเป็น 8 ใช้relu เป็นฟังก์ชันกระตุนชั้นช้อน และให้อาตพุตภมิติ โดยอาตพุตแยกออกเป็นสามชุด ได้แก่ `ymode`, `ymu`, และ `ysigma` สำหรับ  $[p_0, p_1]^T$ ,  $[\mu_0, \mu_1]^T$ , และ  $[\sigma_0, \sigma_1]^T$  ตามลำดับ.

```

1 class Net(nn.Module):
2     def __init__(self):
3         super(Net, self).__init__()
4         self.fc1 = nn.Linear(1, 8)
5         self.fc2 = nn.Linear(8, 6)
6
7     def forward(self, x):
8         z1 = nn.ReLU()(self.fc1(x))
9         z2 = self.fc2(z1)
10        ymode = nn.Softmax(dim=1)(z2[:, :2])
11        ymu = z2[:, 2:4]
12        ysigma = nn.Softplus()(z2[:, 4:])
13        return ymode, ymu, ysigma

```

ด้วยข้อมูล (รายการ 5.16), แบบจำลอง (รายการ 5.18), และฟังก์ชันสูญเสีย (รายการ 5.17) การฝึกก็สามารถทำได้ในลักษณะเดียวกับภาระกิจjoin ๆ อย่างไรก็ตาม รายการ 5.19 แสดงโปรแกรม `train` สำหรับตัวอย่างการฝึกโครงข่ายเพื่อทำนายการแยกแยะ. การฝึกสามารถทำได้ เช่นตัวอย่างคำสั่ง

```

device = torch.device('cuda')
net = Net().to(device)
net, train_losses = train(net, device, 500, 0.001)

```

สำหรับการรันด้วยจีพียู 500 สมัยฝึก ด้วยค่าอัตราเรียนรู้เป็น 0.001.

หมายเหตุ การทำแบบฝึกหัดนี้ ไม่จำเปาะต้องใช้โครงสร้างแบบจำลองตามตัวอย่าง หรือไม่จำเป็นต้องฝึกดังโปรแกรม `train` ไม่จำเป็นต้องใช้ขั้นตอนวิธีอัตโนมัติ หรือไม่จำเป็นต้องสร้างข้อมูลใหม่ทุกสมัยฝึก สามารถเลือกวิธีทำ และดำเนินการได้อย่างอิสระ.

รายการ 5.19: โปรแกรม `train` ฝึกโครงข่ายประสาทเทียม เพื่อทำนายการแยกแยะ. โดย `train` รับ `net`, `device`, `nepochs`, และ `lr` สำหรับโครงสร้างของแบบจำลอง, อุปกรณ์ที่ใช้รัน, จำนวนสมัยฝึก, และอัตราการเรียนรู้ ตามลำดับ. โปรแกรม `train` ใช้ขั้นตอนวิธีอัตโนมัติในการปรับค่าน้ำหนัก. และสร้างข้อมูลใหม่ทุก ๆ สมัยฝึก ด้วย `getdata`.

```

1 def train(net, device, nepochs, lr):
2     r = relation()
3     optimizer = torch.optim.Adam(net.parameters(), lr=lr)

```

```

4     net.train()
5     train_losses = []
6     for t in range(nepochs):
7         optimizer.zero_grad()
8         x, y = getdata(device, r)
9         yhat = net(x)
10        loss = loss1(yhat, y)
11        loss.backward()
12        optimizer.step()
13        train_losses.append(loss.item())
14        if t % 50 == 49:
15            print('* loss', loss.item())
16        if torch.isnan(loss).item() > 0:
17            print('NaN break!')
18            break
19    # end for t
20    return net, train_losses
21
22 def getdata(dev, process):
23     xs = np.random.uniform(-10, 10, 1000)
24     ys = process.sim_y(xs)
25     txs = torch.from_numpy(xs).float().to(dev)
26     tys = torch.from_numpy(ys).float().to(dev)
27     return txs.view(-1,1), tys

```

---

รูป 5.19 แสดงตัวอย่างผลลัพธ์<sup>16</sup> จากการฝึกแบบจำลองดังตัวอย่าง. เนื่องจากลำดับของลักษณะไม่ได้สำคัญ ดังนั้น เพื่อลดความสับสนจากลำดับ ค่าเฉลี่ยที่ใช้สร้างข้อมูลจะติดฉลากเป็น mode 0 และ mode 1 ในขณะที่ ผลทำนายจากแบบจำลอง จะติดฉลากเป็น mode a และ mode b.

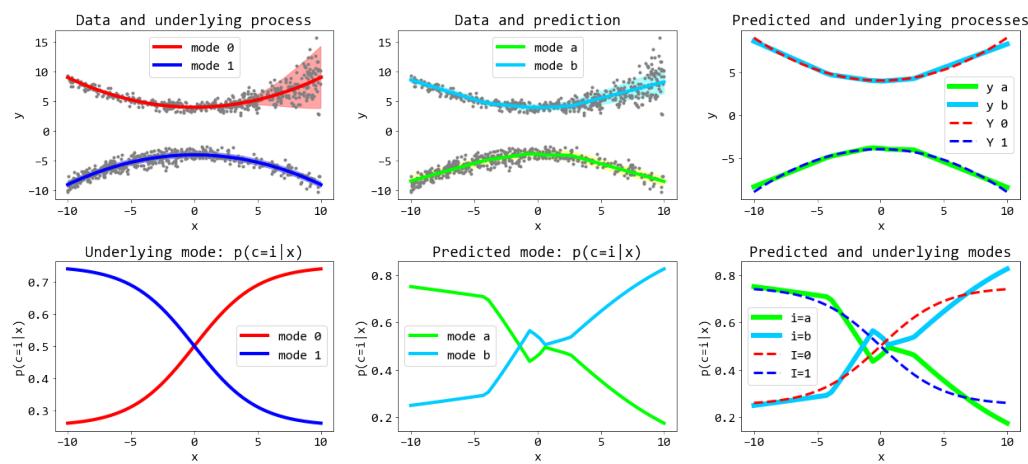
จากการเปรียบเทียบ จะเห็นว่า แบบจำลองทำนายค่าเฉลี่ย  $\mu_0$  และ  $\mu_1$  ได้ดีมาก เปรียบเทียบ ภาพบนซ้ายกับภาพบนกลาง จะเห็นแนวเส้นคล้ายกันมาก (เส้นทึบฟ้า mode b คล้ายเส้นทึบแดง mode 0 และเส้นทึบเขียว mode a คล้ายเส้นทึบน้ำเงิน mode 1) ความน่าจะเป็นของส่วนผสม แบบจำลองก็ทำนายได้ดีพอสมควร เปรียบเทียบภาพล่างซ้ายและภาพล่างกลาง.

ภาพขวาบนและล่าง แสดงค่าเฉลี่ย (ภาพบน) และความน่าจะเป็นของส่วนผสม (ภาพล่าง) ทั้งของเฉลี่ยและที่ทำนายในภาพเดียวกัน. ค่าเบี่ยงเบนมาตรฐาน แสดงด้วยความหนาของพื้นที่แรงงาน ในภาพบนซ้าย

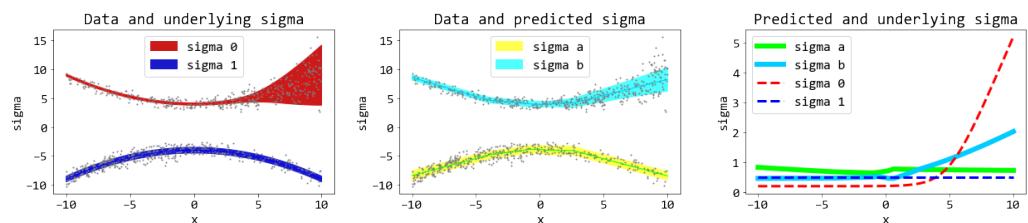
<sup>16</sup>หลังจากฝึกไป 4000 สมัยฝึก โดยเริ่มต้นค่าน้ำหนัก ด้วยวิธีกำหนดค่าหนักเชเวียร์ ด้วยอัตรา  $\sqrt{2}$  โดยการฝึก 2500 สมัยแรก ฝึกกับข้อมูลที่  $\sigma_0 = \sigma_1 = 0.5$  และ 1500 สมัยต่อมา จึงฝึกกับข้อมูลที่มีค่า  $\sigma_0$  และ  $\sigma_1$  ตั้งกำหนด. การฝึกกับข้อมูลดังกำหนด ตั้งแต่แรก พบร้าให้ผลไม่ต่างกันอย่างมีนัยสำคัญ.

(เฉลย) และภาพบนกลาง (ค่าทำนาย) ซึ่ง ลักษณะที่สอง (mode 1 ภาพช้าย และ mode a ภาพกลาง) อาจจะมองเห็นความหนาได้ยาก แต่ลักษณะแรก โดยเฉพาะช่วงปลาย เห็นขัดเจนว่า เฉลยมีค่าเบี่ยงเบนมาตรฐานที่หนามาก แต่ค่าที่ทำนาย แม้จะดูหนาขึ้นในช่วงปลาย แต่ก็ดูแคบกว่าเฉลยมาก.

รูป 5.20 เน้นแสดงผลจากค่าเบี่ยงเบนมาตรฐาน (จุดข้อมูลแสดงด้วยขนาดที่เล็กลง และสีพื้นที่เรางาเลือกให้เข้มขึ้น ในภาพช้ายและภาพกลาง). ภาพขวา แสดงค่าเบี่ยงเบนมาตรฐาน ของทั้งเฉลยและทำนายในภาพเดียวกัน. ถึงแม้ค่าที่ทำนายอาจจะยังดูห่างจากเฉลยมาก แต่เห็นได้ชัดว่าแบบจำลองสามารถจับแนวโน้มของ  $\sigma_0$  (sigma b) ที่เพิ่มในช่วงปลาย และ  $\sigma_1$  (sigma a) ที่คงที่ตลอดช่วงได้.

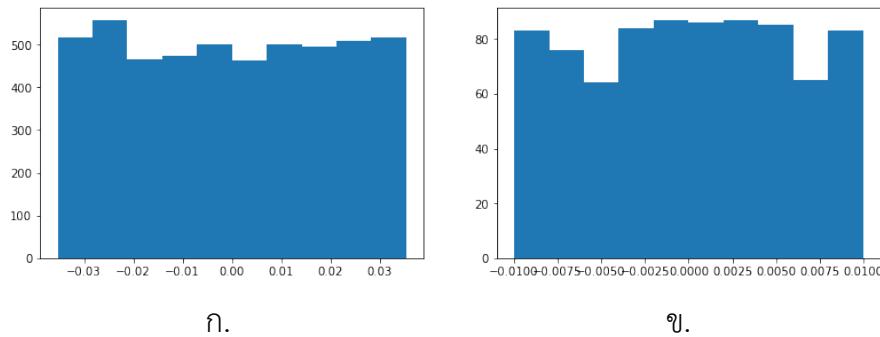


รูปที่ 5.19: ผลลัพธ์การเรียนการแยกแจง. ภาพบนช้ายและกลาง แสดงจุดข้อมูล (จุดสีเทา) พร้อมค่าเฉลย ( $\mu_0$  เส้นสีแดง กับ  $\mu_1$  สีน้ำเงินในภาพช้าย และเส้นฟ้ากับสีเขียวในภาพกลาง) และค่าเบี่ยงเบนมาตรฐาน ( $\sigma_0$  และ  $\sigma_1$ ) ซึ่งแสดงด้วยความกว้างของพื้นที่เรางา. ภาพบนขวา แสดงค่าเฉลยของทั้งค่าที่ทำนาย (ใช้สัญลักษณ์  $y_a$  และ  $y_b$  สำหรับ mode a และ mode b ตามลำดับ) และค่าเฉลย (ใช้สัญลักษณ์  $Y_0$  และ  $Y_1$  สำหรับ mode 0 และ mode 1 ตามลำดับ). ภาพล่างช้ายและกลาง แสดงค่าความน่าจะเป็นของส่วนผสม ( $p_0$  เส้นสีแดง กับ  $p_1$  เส้นสีน้ำเงิน ในภาพช้าย และเส้นฟ้ากับเส้นสีเขียวในภาพกลาง). ภาพล่างขวา แสดงค่าความน่าจะเป็นของส่วนผสมทั้งค่าที่ทำนาย (ใช้สัญลักษณ์  $i=a$  และ  $i=b$  สำหรับ mode a และ mode b ตามลำดับ) และค่าเฉลย (ใช้สัญลักษณ์  $I=0$  และ  $I=1$  สำหรับ mode 0 และ mode 1 ตามลำดับ).



รูปที่ 5.20: ผลลัพธ์การเรียนค่าเบี่ยงเบนมาตรฐาน. ภาพช้ายและกลาง แสดง จุดข้อมูล (จุดสีเทา) และค่าเบี่ยงเบนมาตรฐาน ที่แทนด้วยความหนาของพื้นที่เรางา. ภาพขวา แสดงค่าเบี่ยงเบนมาตรฐาน ในแกนตั้ง และค่าตัวแปรต้น  $x$  ในแกนนอน. ค่าเบี่ยงเบนมาตรฐานที่ทำนาย ใช้สัญลักษณ์  $\sigma_a$  และ  $\sigma_b$  สำหรับ mode a และ mode b ตามลำดับ. ค่าเบี่ยงเบนมาตรฐานของเฉลย ใช้สัญลักษณ์  $\sigma_0$  และ  $\sigma_1$  สำหรับ mode 0 และ mode 1 ตามลำดับ.

การกำหนดค่าเริ่มต้น. เมื่อเราทำการสร้างตัวแปร ค่าของตัวแปรจะถูกกำหนดขึ้นมาด้วย เช่น คำสั่งกำหนดค่า `fc1 = torch.nn.Linear(800, 5000)` จะสร้างพารามิเตอร์ของชั้นคำนวน ได้แก่ `fc1.weight` และ `fc1.bias` ซึ่งเป็นเทนเซอร์ สัดส่วน  $(800, 5000)$  และ  $(5000)$  ตามลำดับ พร้อมค่าเริ่มต้น. โดยดีฟอลต์ของไฟฟอร์ช ค่าเริ่มต้นทั้งของค่าน้ำหนักและไบอัส จะถูกกำหนดดังเช่นสมการ 5.7 นั้นคือ  $\theta \sim \mathcal{U}(-\frac{1}{\sqrt{m_i}}, -\frac{1}{\sqrt{m_i}})$  เมื่อ  $\theta$  คือค่าน้ำหนักหรือไบอัสแต่ละค่า และ  $m_i$  คือจำนวนแฟร์เจของชั้นคำนวน. ตัวอย่างนี้  $m_i = 800$  และหากตรวจสอบการกระจายของค่าเริ่มต้นที่สร้างขึ้น ด้วยคำสั่ง เช่น `plt.hist(fc1.bias.detach())` จะเห็นแผนภูมิแห่งคล้ายตัวอย่างในรูป 5.21 (ภาพ ก). สังเกต ค่าต่ำสุดสูงสุดประมาณ  $-0.035$  และ  $0.035$  ( $\frac{1}{\sqrt{800}} \approx 0.035$ ).



รูปที่ 5.21: การแจกแจงค่าเริ่มต้นของไบอัส. ภาพ ก.  $b \sim \mathcal{U}(-0.035, 0.035)$  และภาพ ข.  $b \sim \mathcal{U}(-0.01, 0.01)$

หากต้องการกำหนดค่าเริ่มต้นนี้เป็นอื่นก็สามารถทำได้ ดังตัวอย่างคำสั่งเช่น

```
with torch.no_grad():
    fc1.bias.data = 2*0.01*torch.rand(800) - 0.01
```

เปลี่ยนค่าไบอัสเป็น  $b \sim \mathcal{U}(-0.01, 0.01)$  ซึ่งเมื่อตรวจสอบ จะเห็นภาพคล้ายตัวอย่างในรูป 5.21 (ภาพ ข). หมายเหตุ จุดสำคัญอยู่ที่ค่าสูงสุดต่ำสุด ไม่ใช่ความสูงต่างของแผนภูมิแห่งแต่ละแท่ง (ที่โดยรวมแสดงการแจกแจงเอกสารูป แต่จำนวนข้อมูลที่น้อย 800 ค่า อาจทำให้เห็นความไม่สมดุลของแต่ละแท่งบ้าง).

การกำหนดค่าเริ่มต้นให้กับโครงข่ายประสาทเทียม อาจทำได้ดังตัวอย่าง

รายการ 5.20: ตัวอย่างการกำหนดค่าเริ่มต้นให้โครงข่ายประสาทเทียม

---

```
1 with torch.no_grad():
2     net.fc1.bias.data = torch.rand(net.fc1.bias.shape)
3     net.fc2.bias.data = torch.rand(net.fc2.bias.shape)
```

---

เมื่อ `net` เป็นตัวแปรแทนโprocrog ข่ายประสาทเทียม ที่มีชั้นคำนวณ `fc1` และ `fc2` และต้องการกำหนดค่าเริ่มต้นของไบอสแต่ละค่า ให้เป็นค่าสุ่มจากการแจกแจงเอกรูป  $\mathcal{U}(0, 1)$ . การกำหนดค่าน้ำหนักก็สามารถทำได้ในลักษณะเดียวกัน.

อย่างไรก็ตาม เพื่อความสะดวก สำหรับการกำหนดค่าเริ่มต้นชั้นคำนวณต่าง ๆ ด้วยวิธีเดียวกัน เมท็อด `apply` ของ `nn.Module`<sup>17</sup> สามารถช่วยลดภาระ การโปรแกรมซ้ำซ้อนลงได้ ดังคำสั่ง

```
with torch.no_grad():
    net.apply(initx)
```

เมื่อ `net` คือตัวแปรprocrog ข่ายประสาทเทียมที่ต้องการกำหนดค่าน้ำหนักเริ่มต้น และ `initx` คือฟังก์ชันกำหนดค่าเริ่มต้นที่ต้องการใช้กับค่าน้ำหนัก (และไบอส) ทุกชั้นคำนวณ. รายการ 5.21 แสดงตัวอย่างโปรแกรมของฟังก์ชันที่ใช้กำหนดค่าน้ำหนักและไบอส (วิธีเซเวียร์ ดูสมการ 5.9 ประกอบ). เนื่องจาก เมท็อด `apply` จะรันฟังก์ชันกับทุก ๆ มอดูลร้อยของ `net` (ตัวอย่างข้างต้น คือ `fc1` และ `fc2`) และตัวของ `net` เอง ดังนั้น ในฟังก์ชันที่จะใช้กำหนดค่าเริ่มต้น จึงต้องทำการเลือกรณี (ตรวจสอบ `type(m)`) เพื่อจะดำเนินการได้ถูกต้อง.

รายการ 5.21: ฟังก์ชันกำหนดค่าน้ำหนักเริ่มต้นเซเวียร์

---

```
1 def initx(m): # xavier initialization
2     if type(m) == nn.Linear:
3         no, ni = m.weight.data.size()
4         s = torch.sqrt(torch.Tensor([6/(ni + no)]))
5         m.weight.data = 2*s*torch.rand(no, ni) - s
6         m.bias.data = 0.1*torch.randn(m.bias.data.size())
```

---

## แบบฝึกหัด 5.19

จากตัวอย่างวิธีกำหนดค่าเริ่มต้น จงเขียนโปรแกรมกำหนดค่าเริ่มต้นแบบโคMING และศึกษางานของโกลโรค และเบนจิโอ<sup>[74]</sup> จากนั้น เลือกชุดข้อมูล ออกแบบการทดลอง เพื่อศึกษาผลกระทบจากฟังก์ชันกระตุ้น และวิธีการกำหนดค่าเริ่มต้น และนำเสนอสิ่งที่ได้เรียนรู้ อภิปราย และสรุปผล.

ตัวอย่างนำเสนอผลและอภิปราย. รูป 5.22 แสดงตัวอย่างผลที่คาดว่า เป็นส่วนหนึ่งที่ทำให้โกลโรคและเบนจิโอ<sup>[74]</sup> ตั้งสมมติฐานว่า หากความแปรปรวนของค่าน้ำหนัก ที่ชี้ต่าง ๆ มีค่าใกล้เคียงกัน จะช่วยให้การฝึกทำได้ง่ายขึ้น. สังเกตว่า ในกรณีที่การฝึกทำได้ดี ความห่างระหว่างเปอร์เซ็นไทล์ที่ 25 และ 75 (สื่อถึงความ

<sup>17</sup>`nn.Module` จะเป็นคลาสแม่ของคลาสprocrog ข่ายประสาทเทียมที่เราสร้างขึ้น จึงสามารถใช้เมท็อดต่าง ๆ ของคลาส `nn.Module` ได้.

แปรปรวน) ของชั้นคำนวนต่าง ๆ จะมีความห่างไกลเคียงกัน แต่อาจจะมีการขยายໄລเป็นชั้น ๆ จากชั้นต้น ๆ ที่จะขยายก่อนและໄລไปชั้นหลัง ๆ ซึ่งต่างจากผลที่เห็น ในกรณีการฝึกล้มเหลว ที่ความแปรปรวนระหว่างชั้นคำนวนต่างกันอย่างชัดเจน นอกจากนั้น ก็ยังไม่เห็นการขยายของความแปรปรวน.

รูป 5.23 แสดงตัวอย่างผลสรุปที่สำคัญ ได้แก่ (1) พังก์ชั้นกระตุ้น **tanh** และ **relu** ทำงานดีกว่า **sigmoid** ไม่ว่าจะใช้โครงข่ายที่มีความลึกเท่าใด. (2) ผลจากการกำหนดค่าเริ่มต้นด้วยวิธีเซเวียร์และไคเมิง (สัญกรณ์ย่อ **x** และ **k** ในภาพ) จะเห็นชัดเจนขึ้น เมื่อใช้งานกับโครงข่ายที่ลึกขึ้น. (3) ทั้งวิธีเซเวียร์ และวิธีไคเมิง ให้ผล ดังที่คาดหมาย นั่นคือ วิธีเซเวียร์ ช่วยในกรณี **tanh** เมื่อเปรียบเทียบกับวิธีพื้นฐาน (สมการ 5.7 สัญกรณ์ย่อ **u** ในภาพ). และวิธีไคเมิง ช่วยในกรณี **relu** เมื่อเปรียบเทียบกับทั้งวิธีเซเวียร์และวิธีพื้นฐาน. สังเกตว่า ทั้งวิธีเซเวียร์และวิธีไคเมิงพัฒนา โดยอาศัยสมมติฐานเชิงเส้น ที่แม้จะไม่ตรงกับสถานการณ์จริง แต่ในทางปฏิบัติ กลับพบว่า ทั้งสองวิธีทำงานได้ดีอย่างชัดเจน.

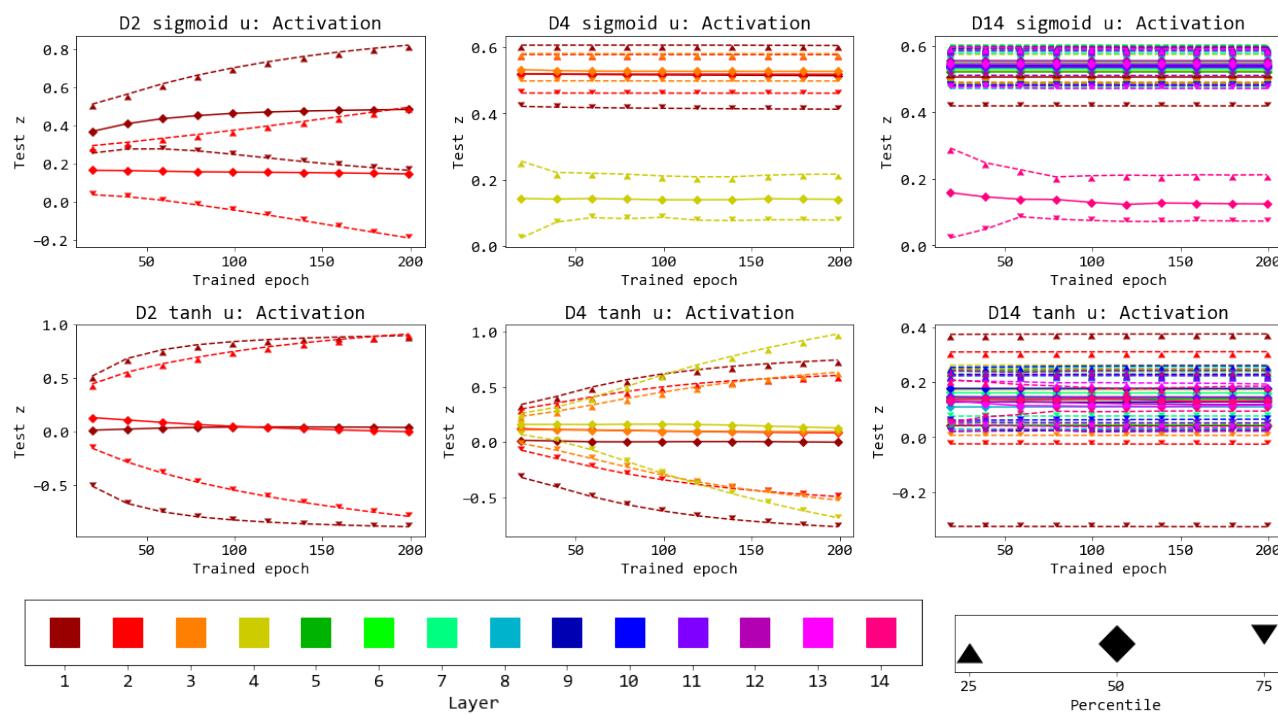
นอกจาก ผลสรุปเรื่องวิธีการกำหนดค่าเริ่มต้น อีกประเด็นหนึ่งที่โกลโลร์และเบนจิโอ[74]ได้ศึกษา ก็คือ การตรวจดูค่าโดยรวม ของผลการกระตุ้นและค่าเกรเดียนต์ระหว่างฝึก (ดังเข้นที่แสดงในรูป 5.22) และตรวจสอบการแยกของผลการกระตุ้นภายหลังการฝึก (ดังเข้นรูป 5.24) ที่ทั้งคู่คิดว่าจะช่วยให้สามารถทำความเข้าใจพฤติกรรมการเปลี่ยนแปลงของโครงข่ายจากการฝึกได้ดีขึ้น.

รูป 5.24 แสดงให้เห็นว่า กรณีที่การฝึกทำได้ดี (โครงข่ายสองชั้นทุกแบบ, โครงข่ายสี่ชั้น เมื่อใช้ **tanh** หรือ **relu**, โครงข่ายสิบสี่ชั้น เมื่อใช้ **tanh** และวิธีเซเวียร์ หรือเมื่อใช้ **relu** และวิธีไคเมิง. ดูรูป 5.23 ประกอบ) หน่วยคำนวนส่วนใหญ่ (ในเกือบทุกกรณี ยกเว้น **tanh** และวิธีเซเวียร์) อยู่ในระดับอิมิตัว (saturation) นั่นคือ ค่าผลการกระตุ้น  $z = 0$  หรือ  $z = 1$  สำหรับพังก์ชันซิกมอยด์, ค่าการกระตุ้น  $z = -1$  หรือ  $z = 1$  สำหรับพังก์ชันไฮเปอร์บolic tangent, และค่าการกระตุ้น  $z = 0$  สำหรับrelu.

หมายเหตุ การฝึกเขียนโปรแกรมเอง (เช่น รายการ 5.21) จะช่วยให้เข้าใจกลไกภายในได้ดี แต่การใช้งานในทางปฏิบัติ การใช้พังก์ชันสำเร็จรูป จะช่วยเพิ่มความสะดวกในการทำงานและการสื่อสารได้ดีขึ้น (โดยเฉพาะในกรณีทำงานด้วยกันหลายคน). ไฟฟอร์มมีพังก์ชันสำเร็จรูป สำหรับการกำหนดค่าเริ่มต้นด้วยวิธีที่รู้จักดีต่างๆ รวมถึงวิธีเซเวียร์และไคเมิง เช่น **nn.init.xavier\_uniform\_(w)** สำหรับการกำหนดค่าน้ำหนักเริ่มต้นให้กับพารามิเตอร์ **w** ด้วยวิธีเซเวียร์.

## แบบฝึกหัด 5.20

จงเลือกชั้นตอนวิธีการฝึก จากวิธีลงเกรเดียนต์กับกลไกโมเมนตัม, วิธีอาร์เมอสพรอป, วิธีอาร์เมอสพรอป กับกลไกโมเมนตัม, วิธีอัตโนมัติ หรือวิธีอื่น ๆ ที่สนใจ และเขียนโปรแกรมวิธีดังกล่าว เพรียบเทียบผลการทำงาน

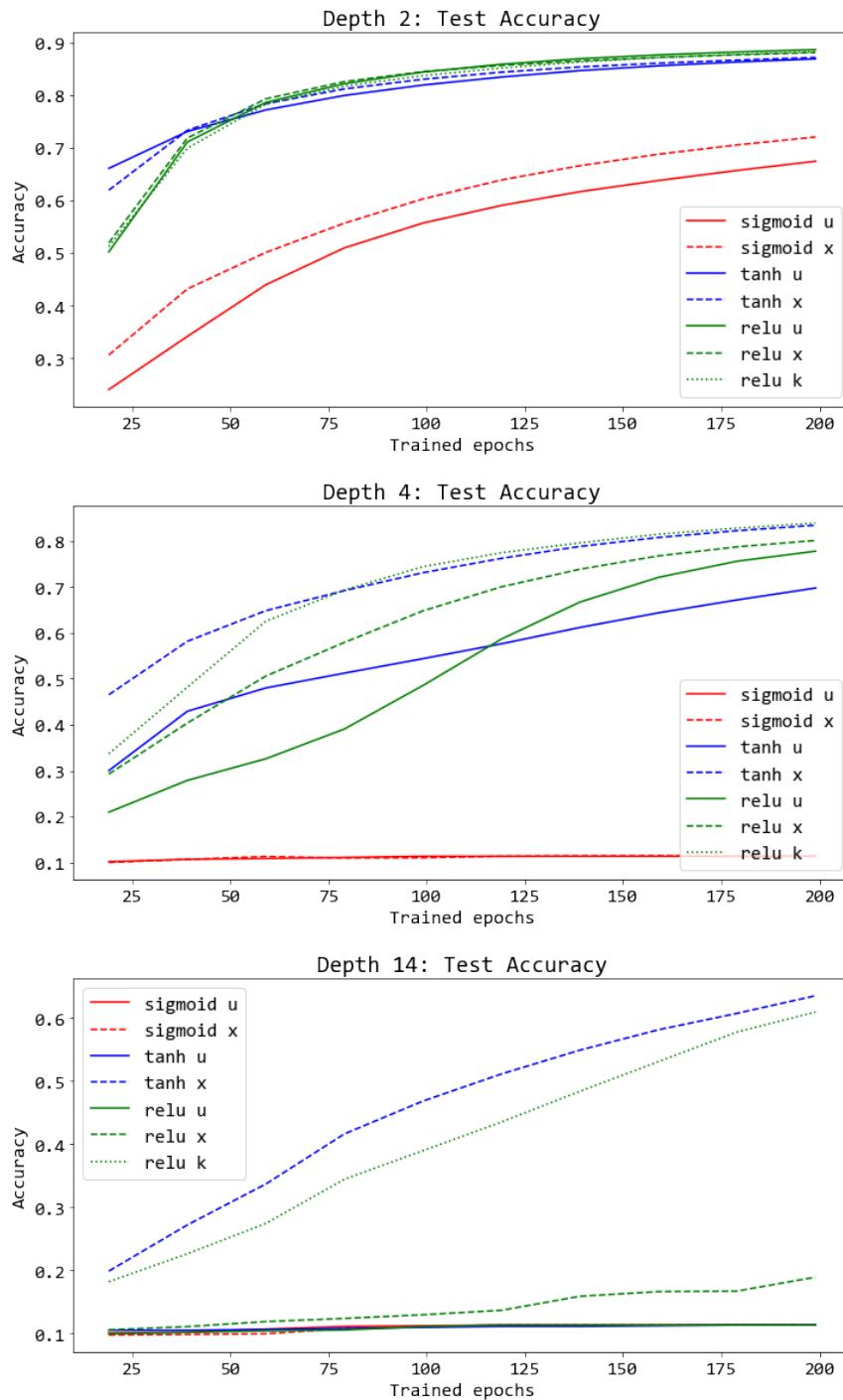


รูปที่ 5.22: ค่าผลการกระตุ้นระหว่างการฝึก (แสดงด้วยค่าเบอร์เซ็นไทล์ที่ 25, 50, และ 75) เมื่อใช้ฟังก์ชันซิกมอยด์ (ภาพในแฉบ) และไสเปอร์บอลิกแทนเจนต์ (ภาพในแฉบถัดมา) โดยมีโครงข่ายมีความลึก 2 ชั้น (ภาพซ้าย ระบุหนึ่งภาพด้วย D2), 4 ชั้น (ภาพกลาง ระบุด้วย D4), และ 14 ชั้น (ภาพขวา ระบุด้วย D14)。ภาพในแฉบล่าง แสดงสี สำหรับค่าผลการกระตุ้นที่ชั้นคำนวณต่างๆ และสัญลักษณ์ที่ระบุค่าเบอร์เซ็นไทล์。ในหากแบบจำลองนี้ มีสามแบบจำลองที่การฝึกทำได้สำเร็จ ได้แก่ แบบจำลองสองชั้นที่ใช้ซิกมอยด์ (ภาพซ้ายบน) แบบจำลองสองและสี่ชั้นที่ใช้ไสเปอร์บอลิกแทนเจนต์ (ภาพซ้ายและกลาง แฉบที่สอง)。

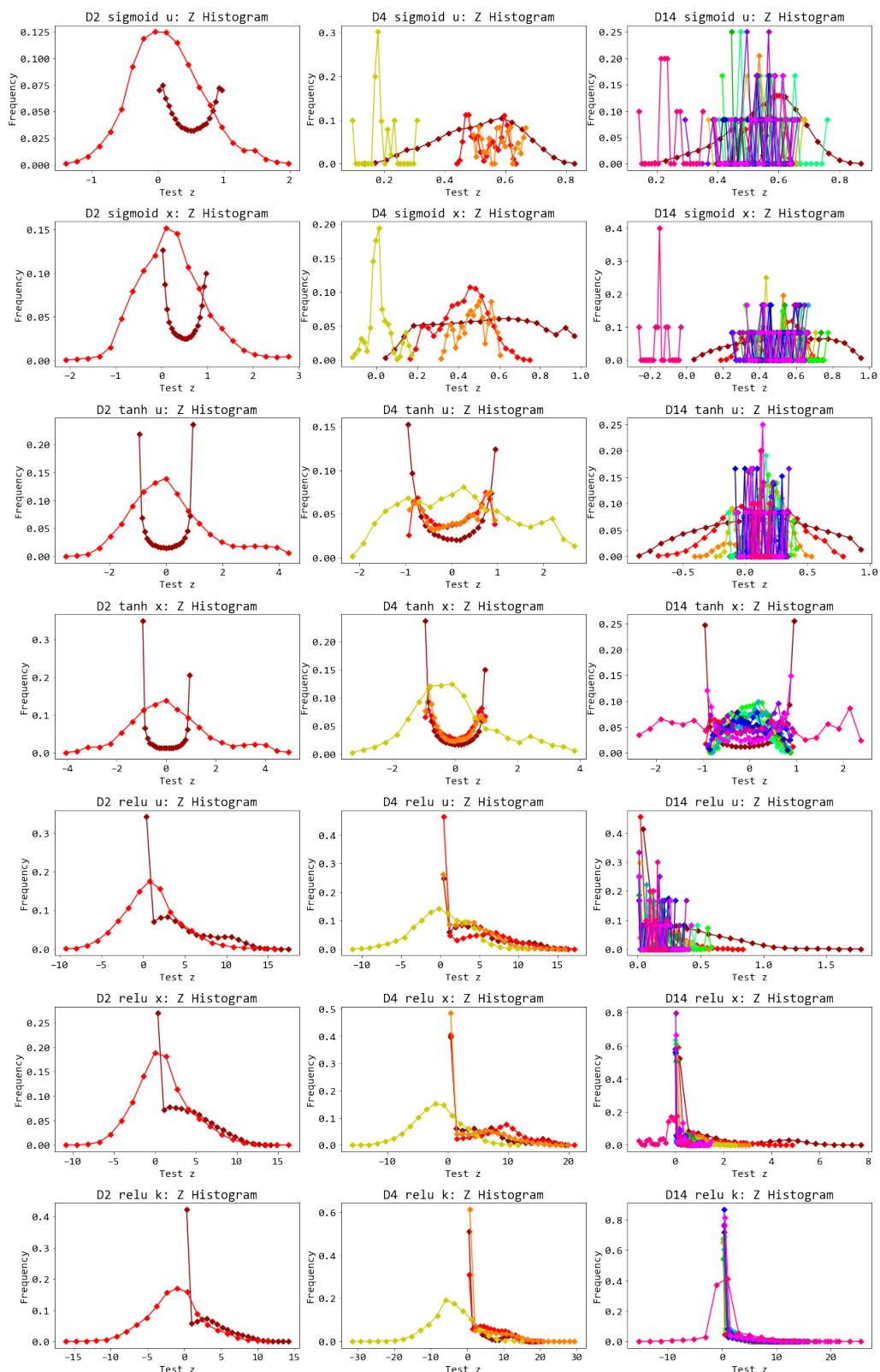
กับโปรแกรมสำเร็จของวิธีนั้น (ได้แก่ `optim.SGD`, `optim.RMSprop`, และ `optim.Adam`) และเปรียบเทียบกับวิธีลงเกรเดียนต์ เลือกชุดข้อมูลขึ้นมาเพื่อทดสอบ ออกแบบการทดลอง เพื่อวัดผลทั้งในเชิงความเร็วและคุณภาพในการเรียนรู้ รวมถึงความทนทานต่อค่าอภิมานพารามิเตอร์ต่างๆที่เลือกใช้ และความทนทานกับการกำหนดค่าเริ่มต้นแบบต่างๆ อภิปราย และสรุป。

### แบบฝึกหัด 5.21

จะออกแบบการทดลอง เพื่อทดสอบการทำงานของแบบนอร์ม วัดผลทั้งในความเร็วในการฝึก คุณภาพการฝึก ความทนทานต่อค่าอัตราการเรียนรู้ ผลจากขนาดของหมู่เล็ก รวมถึงตำแหน่งที่ทำแบบนอร์ม (ทำที่ตัวกระตุ้น นั่นคือก่อนฟังก์ชันกระตุ้น เปรียบเทียบกับทำที่ผลการกระตุ้น นั่นคือหลังฟังก์ชันกระตุ้น) ทดลองเขียนโปรแกรมแบบนอร์ม (ตัวอย่างแสดงในรายการ 5.23) และเปรียบเทียบกับโปรแกรมแบบนอร์มสำเร็จรูป (เช่น คำสั่ง `self.bn1 = nn.BatchNorm1d(8)` เปรียบเทียบกับ `self.bn1 = MyBN(8)` ในรายการ 5.22 เมื่อ `MyBN` กำหนดดังแสดงในรายการ 5.23)。สังเกตผล อภิปราย และสรุป。



ຮູບທີ 5.23: ດ້ວຍຄວາມແມ່ນຍຳກັບຂໍອມລົດສອບ ເນື້ອໃໝ່ໂຄຮງຂ່າຍຄວາມລຶກ 2 ຊັ້ນ (ກາພບນສຸດ) 4 ຊັ້ນ (ກາພກລາງ) ແລະ 14 ຊັ້ນ (ກາພລ່າງ) ປະກອບກັບຝັງກັນກະຕຸນໃນຂັ້ນຊ່ອນແບບຕ່າງໆ (**sigmoid** ຜິກມອຍດ, **tanh** ໄຂເປ່ອຮບອລິກແທນເຈນຕ, ແລະ **relu** ເຮລູ) ແລະ ວິເກີດກຳຫຼາດຕ່າງໆ (u ວິເກີດພື້ນຖານ ສາມກາຣ 5.7, x ວິເກີດເງິເວີຣ ສາມກາຣ 5.9, ແລະ k ວິເກີດເຄີມ ສາມກາຣ 5.12). ດ້ວຍຄວາມແມ່ນຍຳໄສແສດງ ເປັນຄ່າເຂົ້າຈາກການທຳຫຳສຶບຄຽງ.



รูปที่ 5.24: แต่ละภาพ แสดงการแจกแจงของค่าการกระตุน สำหรับแต่ละกรณี (ความลึก พังก์ชันกระตุนที่ใช้ และวิธีกำหนดค่าเริ่มต้น ระบุเนื้อแต่ละภาพ ด้วยรหัส เช่นเดียวกับรูป 5.22 และ 5.23). แกนนอน แสดงค่าผลการกระตุน และแกนตั้ง แสดงค่าความถี่ หารด้วยจำนวนทั้งหมด.

รายการ 5.22: ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้แบบชั้นอิรุ่ม. แบบชั้นอิรุ่มเหมือนชั้นคำนวณที่เพิ่มขึ้น. คลาส MyBN เป็นชั้นคำนวณแบบชั้นอิรุ่ม กำหนดดังแสดงในรายการ 5.23. หมายเหตุ การใช้แบบชั้นอิรุ่ม ทำให้ใบอัสเกินความจำเป็นและซ้ำซ้อน และสามารถตัดออกได้. แต่ในตัวอย่างนี้ไม่ได้ตัดค่าใบอัสออก. หากต้องการตัดใบอัสออก สามารถทำได้โดยคำสั่ง เช่น `self.bn1 = nn.Linear(1, 8, bias=False)`

---

```

1 class MyNetManualBN(nn.Module):
2     def __init__(self):
3         super(MyNetManualBN, self).__init__()
4         self.fc1 = nn.Linear(1, 8)
5         self.bn1 = MyBN(8)
6         self.fc2 = nn.Linear(8, 8)
7         self.bn2 = MyBN(8)
8         self.fc3 = nn.Linear(8, 1)
9
10    def forward(self, x):
11        self.a1 = self.fc1(x)
12        self.b1 = self.bn1(self.a1)
13        self.z1 = torch.relu(self.b1)
14        self.a2 = self.fc2(self.z1)
15        self.b2 = self.bn2(self.a2)
16        self.z2 = torch.relu(self.b2)
17        self.z3 = self.fc3(self.z2)
18        return self.z3

```

---

รายการ 5.23: โปรแกรมแบบชั้นอิรุ่ม. การใช้ `nn.Parameter` ช่วยให้เมท็อด `parameters()` ของแบบจำลองรู้จักพารามิเตอร์ที่กำหนดขึ้น (และการปรับค่าพารามิเตอร์ตามเกรเดียนต์ก็จะถูกทำโดยอัตโนมัติ เช่นเดียวกับค่าน้ำหนักและใบอัส). การใช้ `register_buffer` ลงทะเบียนตัวแปร เพื่อให้ตัวแปรที่ลงทะเบียนถูกรวบเข้าไป เมื่อทำการกำหนดอุปกรณ์ (เช่น `net.to(device)`) หรือการบันทึกแบบจำลอง (เช่น `torch.save(net.state_dict(), 'savnet.pth')`) และไม่ถูกรวบเข้าไปในกลุ่มพารามิเตอร์ของแบบจำลอง (และไม่ถูกปรับค่าอัตโนมัติจากเกรเดียนต์).

---

```

1 class MyBN(nn.Module):
2     def __init__(self, num_features, eps=1e-5, momentum=0.1):
3         super(MyBN, self).__init__()
4         self.num_features = num_features
5         self.eps = eps
6         self.momentum = momentum
7         self.weight = nn.Parameter(torch.ones(num_features))
8         self.bias = nn.Parameter(torch.zeros(num_features))
9         self.register_buffer('running_mean', torch.zeros(←
10                           num_features))
11        self.register_buffer('running_var', torch.ones(←
12                           num_features))

```

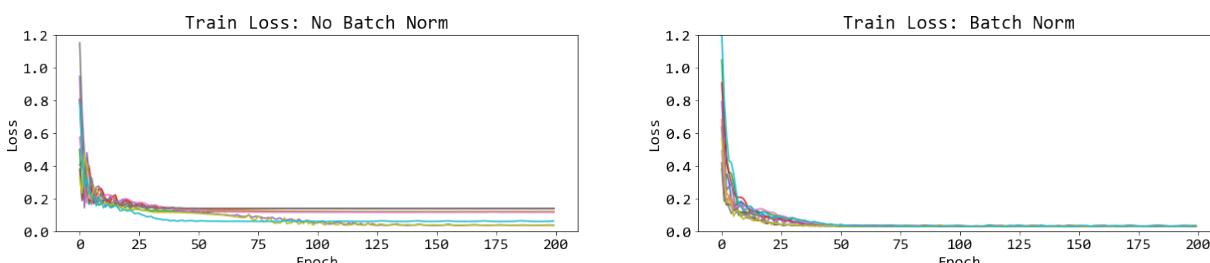
```

12     def forward(self, z):
13         mu = self.running_mean
14         svar = self.running_var
15         if self.training:
16             with torch.no_grad():
17                 mu = torch.mean(z, dim=0)
18                 svar = torch.var(z, dim=0)
19             # Tracing running_mean and running_var
20             p = self.momentum
21             q = 1 - p
22             self.running_mean = p*mu + q*self.running_mean
23             self.running_var = p*svar + q*self.running_var
24         # end self.training
25         zn = (z - mu)/torch.sqrt(svar + self.eps)
26         zns = zn * self.weight + self.bias
27         return zns

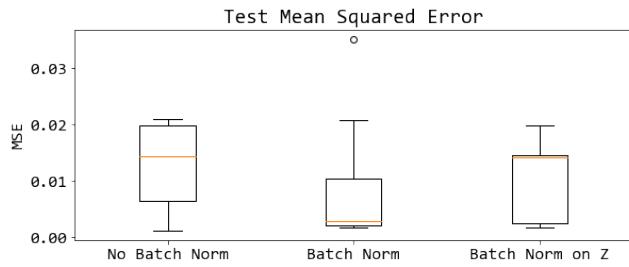
```

รูป 5.25, 5.26, 5.27, และ 5.28 แสดงตัวอย่างการนำเสนอผล. รูป 5.25 แสดงค่าสูญเสียระหว่างการฝึกจากการทดสอบสิบชั้น ภาพซ้าย เมื่อไม่ได้ใช้แบนดอร์ม และภาพขวา เมื่อใช้แบนดอร์ม. เห็นได้ชัดเจนว่า แบนดอร์มช่วยให้การฝึกทำได้เร็วขึ้นและแน่นอนขึ้น.

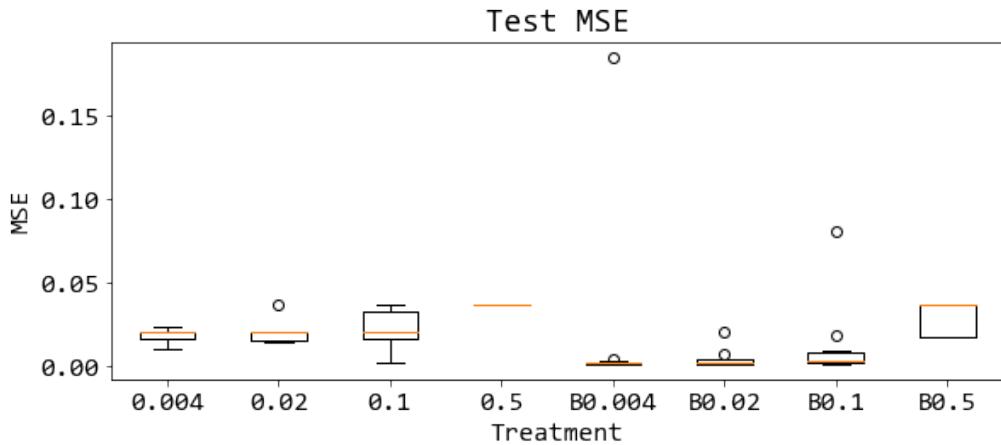
รูป 5.26 แสดงค่าทดสอบ ซึ่งในที่นี้ใช้ค่าเฉลี่ยกำลังสองน้อยที่สุด. ภาพแสดงด้วยแผนภูมิกล่อง กล่องซ้ายสุด แสดงค่าผิดพลาด เมื่อไม่ใช้แบนดอร์ม. กล่องกลาง เมื่อใช้แบนดอร์ม (ทำที่ตัวกระตุ้น นั่นคือ ใช้ทำที่  $\mathbf{A}$  เมื่อ ขั้นคำนวน ทำ  $h(\mathbf{A})$  โดย  $\mathbf{A} = \mathbf{W} \cdot \mathbf{Z} + \mathbf{b}$  หรือ ทำก่อนเข้าฟังก์ชันกระตุ้น). กล่องขวา เมื่อใช้แบนดอร์ม แต่ทำแบนดอร์มที่ผลการกระตุ้น แทนที่จะทำที่ตัวกระตุ้น (นั่นคือ ทำที่  $\mathbf{Z}$  หรือทำหลังฟังก์ชันกระตุ้น). รูป 5.26 แสดงในเห็นว่า ไม่เพียงแต่ แบนดอร์มช่วยให้การฝึกดำเนินการได้เร็วขึ้น แบนดอร์มยังช่วยคุณภาพการฝึกด้วย และเพื่อให้ได้ประสิทธิภาพที่ดี การทำแบนดอร์มควรทำที่ค่าตัวกระตุ้น (ค่าก่อนเข้าฟังก์ชันกระตุ้น).



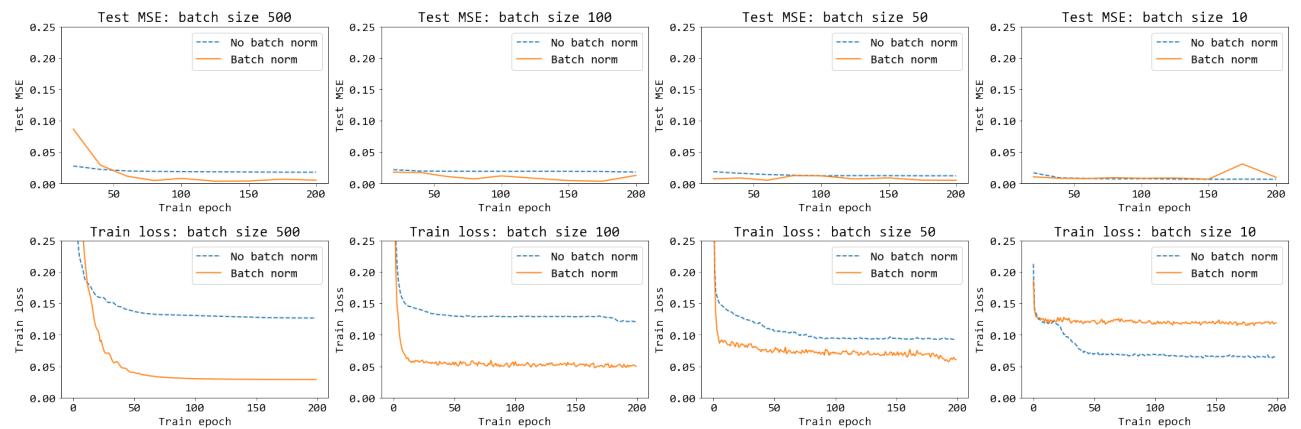
รูปที่ 5.25: ค่าฟังก์ชันสูญเสียต่อสมัยฝึก จากการทดสอบชั้น 10 ครั้ง เมื่อไม่ใช้แบนดอร์ม (ภาพซ้าย) และใช้แบนดอร์ม (ภาพขวา).



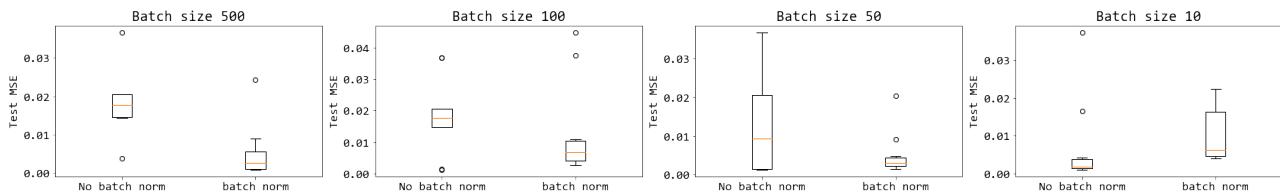
รูปที่ 5.26: แผนภูมิก่อล่อง แสดงค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากการทดสอบ 10 ครั้ง เมื่อไม่ใช้แบนชันอร์ม (กล่องซ้าย), เมื่อใช้แบนชันอร์ม (กล่องกลาง) และเมื่อใช้แบนชันอร์ม แต่ทำแบนชันอร์มที่ผลการกระตัน (กล่องขวา).



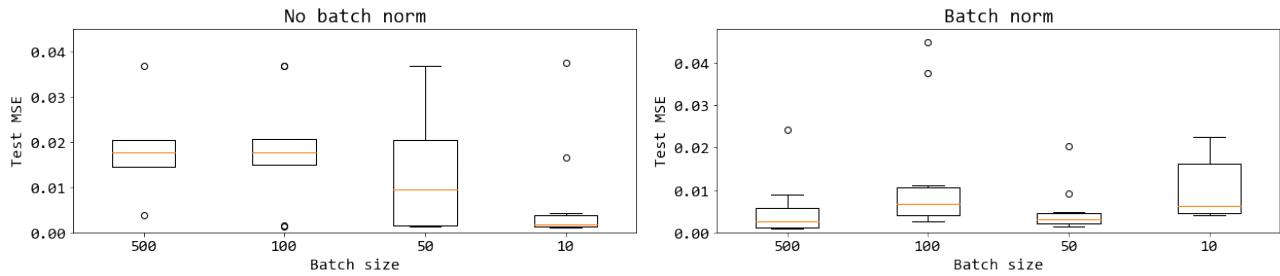
รูปที่ 5.27: แผนภูมิก่อล่องแสดงค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากการทดสอบ 10 ครั้ง เมื่อไม่ใช้แบนชันอร์ม และเมื่อใช้แบนชันอร์ม กับอัตราเรียนรู้ต่าง ๆ. ตัวเลขที่แสดง หมายถึง อัตราเรียนรู้ที่ใช้ และอักษร B ที่กำกับหมายถึง มีการใช้แบนชันอร์ม.



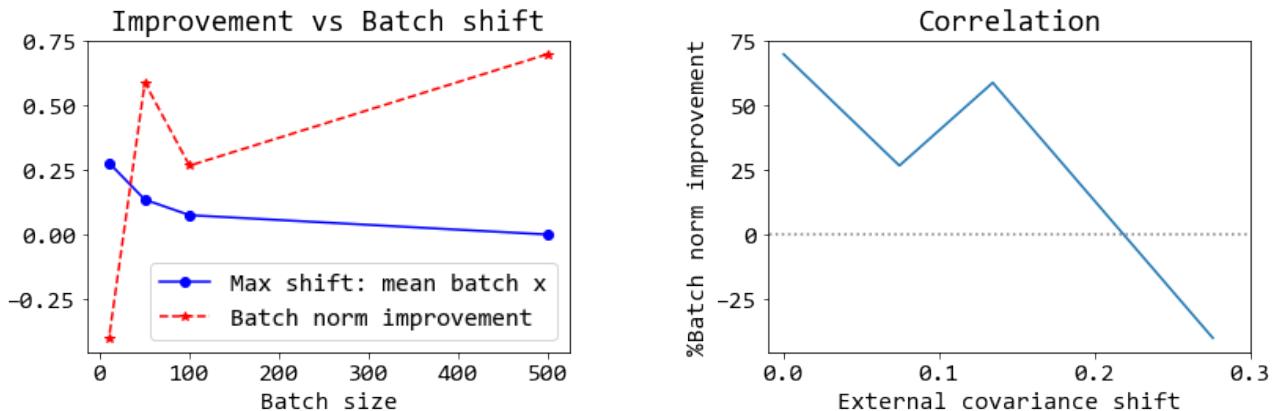
รูปที่ 5.28: ค่าเฉลี่ยค่าผิดพลาด ระหว่างการฝึก เมื่อใช้และไม่ใช้แบนชันอร์ม ที่ขนาดหมู่เล็กต่าง ๆ (ภาพต่าง ๆ ในแถวบน). ค่าเฉลี่ยค่าฟังก์ชันสูญเสีย ระหว่างการฝึก เมื่อใช้และไม่ใช้แบนชันอร์ม ที่ขนาดหมู่เล็กต่าง ๆ (ภาพต่าง ๆ ในแถวล่าง).



รูปที่ 5.29: แผนภาพกล่อง แสดงค่าผิดพลาด จากการทดสอบ 10 ครั้ง เมื่อใช้และไม่ใช้แบนชันอร์ม กับขนาดหมู่เล็กต่าง ๆ.



รูปที่ 5.30: แผนภาพกล่อง แสดงค่าผิดพลาด จากการทดสอบ 10 ครั้ง เมื่อใช้ขนาดหมู่เล็กต่าง ๆ ในกรณีที่ใช้และไม่ใช้แบนชันอร์ม. เปรียบเทียบกับรูป 5.29 รูปนี้นำเสนอจากอีกมุมมองหนึ่ง นั่นคือ การเลือกใช้หรือไม่ใช้แบนชันอร์ม ควรพิจารณาประกอบกับขนาดของหมู่เล็กที่จะเลือกใช้ด้วย.



รูปที่ 5.31: ภาพข่าย แสดงการทำงานของแบนชันอร์มกับการเลื่อนของความแปรปรวนร่วมเกี่ยวกับภายนอก. อัตราการปรับปรุงคุณภาพการทำงานเมื่อใช้แบนชันอร์มเทียบกับไม่ใช้ แสดงด้วยเส้นประสีแดง โดย ค่าเป็นบวกหมายถึงคุณภาพดีขึ้นเมื่อใช้แบนชันอร์ม ศูนย์หมายถึงคุณภาพเท่ากัน และค่าเป็นลบหมายถึงคุณภาพแย่ลง. เส้นทึบสีน้ำเงิน แสดงค่ามากที่สุดของความต่างระหว่างค่าเฉลี่ยของหมู่กับ ค่าเฉลี่ยของข้อมูลทั้งหมด (ความต่างคิดเป็นค่าสัมบูรณ์). ภาพขวา แสดงค่ามากที่สุดของความต่าง (แคนนอน) กับเพอร์เซ็นต์การปรับปรุงคุณภาพเมื่อใช้แบนชันอร์ม (แกนตั้ง).

รูป 5.27 แสดงให้เห็นว่า แบนชันอร์มทำงานได้ดีที่ค่าอัตราเรียนรู้ต่าง ๆ. รูป 5.28 แสดง การทำงานของแบนชันอร์ม ในสถานการณ์ของขนาดหมู่เล็กต่าง ๆ ในช่วงสมัยฝิกต่าง ๆ. สังเกตว่า เมื่อใช้หมู่เล็กขนาดเล็กเกินไป (ภาพขวาสุด บนและล่าง) แบนชันอร์มทำงานได้ไม่ดี และนำไปสู่การฝิกที่แย่กว่าการฝึกที่ไม่ใช้แบนชันอร์ม.

รูป 5.29 แสดงค่าความผิดพลาดเมื่อนำแบบจำลองที่ฝึกไปทดสอบ. รูป 5.29 ยืนยันว่า หากใช้แบบนอร์ม แล้วเลือกขนาดหมู่เล็กที่เล็กเกินไป จะทำให้ผลการฝึกแย่ลงได้.

แบบนอร์ม ออกแบบมาเพื่อแก้ไขการเลื่อนของความแปรปรวนร่วมเกี่ยวภายใน ที่เกิดจากการปรับค่าพารามิเตอร์ระหว่างการฝึก แต่การทำแบบนอร์ม ที่ปรับค่าเฉลี่ยและความแปรปรวนของหมู่เล็ก ก็เสียงที่จะทำสารสนเทศจากข้อมูลเสียหาย. หากความต่างของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากตัวข้อมูลเองไม่ใช่จากการเปลี่ยนแปลงของค่าพารามิเตอร์ในชั้นคำนวนก่อนหน้า.

หากการแจกแจงสารสนเทศของข้อมูลในหมู่เล็กแต่ละหมู่ ค่อนข้างคงเส้นคงวา และสามารถแทนการแจกแจงสารสนเทศของข้อมูลโดยรวมได้ การเลื่อนของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากการเปลี่ยนแปลงของค่าพารามิเตอร์ของชั้นคำนวนก่อนหน้า การทำแบบนอร์มจะมีประสิทธิผลตามที่ออกแบบไว้.

แต่หากการเลื่อนของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากสารสนเทศที่ต่างกันของข้อมูลระหว่างหมู่เอง ความเสี่ยงของการทำแบบนอร์ม จะสูงขึ้นมาก และเมื่อประกอบกับแนวทางปฏิบัติของการสุ่มหมู่เล็ก ที่มักทำการสุ่มแค่ครั้งแรก และใช้ลำดับและการจัดกลุ่มนั้นตลอด ยิ่งจะซ้ำเติมความเสี่ยงนี้เข้าไปอีก.

ตัวอย่างของกรณีที่ใช้ขนาดหมู่เล็กเป็นสิบ ภาพขวาสุดที่แสดงในรูป 5.29 แสดงให้เห็นว่า เมื่อขนาดหมู่เล็กลง โอกาสที่การแจกแจงสารสนเทศของข้อมูลระหว่างหมู่จะสมำเสมอหรือจะเป็นตัวแทนของข้อมูลทั้งหมดได้ จะน้อยลง และเมื่อการแจกแจงสารสนเทศของข้อมูลระหว่างหมู่ไม่สมำเสมอ การทำแบบนอร์มจึงทำสารสนเทศบางอย่างเสียหายไป และส่งผลให้การฝึกทำได้แย่. รูป 5.30 เปรียบเทียบให้เห็นว่า ความสัมพันธ์ระหว่างคุณภาพการฝึกกับขนาดของหมู่เล็ก เปลี่ยนไป เมื่อใช้แบบนอร์ม. ในทางปฏิบัติ หลายครั้ง ขนาดของหมู่เล็ก อาจถูกจำกัดจากหน่วยความจำ และการเลือกใช้หรือไม่ใช้แบบนอร์ม ควรคำนึงถึงความเสี่ยง จากประเด็นความสมำเสมอระหว่างหมู่เล็ก ของการแจกแจงสารสนเทศจากตัวข้อมูลเองประกอบ.

รูป 5.31 แสดงประเด็นความสัมพันธ์ระหว่างประสิทธิผลการทำงานของแบบนอร์มกับการแจกแจงข้อมูลระหว่างหมู่เล็ก. จากรูป เมื่อ การแจกแจงข้อมูลระหว่างหมู่เล็ก ทำให้เกิดความต่างระหว่างหมู่เล็กมาก ประสิทธิผลการทำงานของแบบนอร์มจะต่ำลง และอาจต่ำจนการใช้แบบนอร์มจะเป็นผลเสียมากกว่าผลดี ดังแสดงออกมาเป็นเปอร์เซ็นต์ปรับปรุงที่ติดลบ.

แล้วกรณีที่ข้อมูลมีรูปแบบแบลก ๆ ที่พบร้าด้วยกัน หรือกรณีประเด็นเรื่องวิธีประมาณค่า  $\mu_i$  และ  $\sigma_i^2$  ที่ใช้ทำแบบนอร์มภายหลังการฝึก อาจก่อให้เกิดความเสี่ยงอย่างไรบ้าง จผลกระทบความคิด ภูมิประยุ และสรุป.

## แบบฝึกหัด 5.22

**การศึกษาหัวข้อที่สนใจ.**<sup>18</sup> บางครั้งในบางจังหวะเวลา ศาสตร์ที่เราศึกษา มีการเปลี่ยนแปลงพัฒนาที่รวดเร็วมาก และอาจจำเป็นต้องศึกษาความก้าวหน้าและพัฒนาการล่าสุดจากแหล่งอื่น ๆ เพิ่มเติม.

จะเลือกหัวข้อเรื่องที่สนใจ ( เช่น การบรรยายภาพอัตโนมัติ, การแต่งเพลงอัตโนมัติ, การทำนายพฤติกรรม ประตีน ) แล้วศึกษา ทำความเข้าใจในหัวเรื่องดังกล่าว.

การศึกษา ทำความเข้าใจ อาจทำโดย (1) ค้นหาและรวบรวมแหล่งข้อมูล เกี่ยวกับหัวข้อที่สนใจ ซึ่งแหล่งข้อมูลอาจรวมถึง หนังสือ บทความวิชาการ บทความทั่วไป วีดีโอ เว็บเพจ เป็นต้น.

(2) ศึกษาแต่ละแหล่งข้อมูลที่ได้มาร่วม ๆ (ถ้าเป็นบทความวิชาการ รวมถึงบทความวิจัย อย่างน้อย อ่านบทคัดย่อและบทนำ) และอาจจะทำบันทึกย่อ ว่า แต่ละแหล่งข้อมูลเกี่ยวข้องกับหัวข้อที่สนใจมากน้อยขนาดไหน และเราเข้าใจเนื้อหาเข้าใจมากน้อยขนาดไหน รวมถึงอาจจัดลำดับความสำคัญ และหมายเหตุถึงสิ่งที่คิดว่าจะดำเนินการต่อ เช่น ไม่ค่อยเกี่ยวข้อง ตัดทิ้งไปก่อน หรือ เกี่ยวข้องมาก ศึกษาให้เข้าใจ หรือ เกี่ยวข้องประมาณ 50% พอก็เข้าใจแล้ว เก็บไว้เป็นตัวอย่าง หรือ ไม่แน่ใจว่าเกี่ยวข้อง เข้าใจดี ขอบเทคนิคที่เข้าใช้ อาจใช้เป็นประโยชน์กับงานของเราได้ เก็บไว้อ้างอิงทีหลัง. หรือ ไม่แน่ใจว่าเกี่ยวข้อง ไม่ค่อยเข้าใจ เก็บไว้ดูอีกทีหลังจากเข้าใจหัวข้อนี้ดีขึ้น.

โดยทั่วไป ถ้าเป็นบทความวิชาการหรือบทความวิจัย ที่ไม่ใช่บทความทบทวน หรือไม่ใช่บทความสำรวจ เราอาจจะต้องอ่านตั้งแต่สามจนถึงสิบบทความ จึงอาจจะพอเข้าใจหัวข้อนั้นในระดับเบื้องต้นได้. นักศึกษาปริญญาเอก ซึ่งถูกคาดหวังว่าจะเข้าใจในหัวข้อเป็นอย่างดี อาจต้องอ่านบทความ ไม่น้อยกว่า 100 บทความ. และก็ไม่แปลกที่จะเห็น บทความทบทวน หรือบทความสำรวจของหัวข้อใด ๆ ที่คนผู้เขียนอาจต้องอ่านบทความต่าง ๆ ที่เกี่ยวข้องกับหัวข้อนั้น ๆ มากกว่า 300 บทความ.

(3) หากไม่สามารถเข้าใจบทความวิชาการส่วนใหญ่ได้ ให้ระบุศาสตร์พื้นฐานที่อาจจะขาดไป เช่น บทความที่หนึ่ง เกี่ยวข้องมาก แต่ไม่เข้าใจเลย ดูเหมือนจะใช้ทฤษฎีความน่าจะเป็นและแคลคูลัสของการแปรผัน (calculus of variations). บทความที่สอง เกี่ยวข้อง แต่อ่านไม่เข้าใจ ใช้พิชิตเชิงเส้น, ความน่าจะเป็น และทฤษฎีสารสนเทศ. บทความที่ห้า เกี่ยวข้องมาก แต่ยังอ่านไม่เข้าใจ ใช้ทฤษฎีความน่าจะเป็น, กระบวนการล็อกแคลคูลิก และการหาค่าดีที่สุด. บทความที่แปด เกี่ยวข้องบ้าง แต่อ่านไม่เข้าใจ เพราะประยุกต์ใช้กับงานชีวการแพทย์ ใช้การหาค่าดีที่สุด และชีวเคมี. เช่นนี้ ก็จะช่วยให้เราพอเห็นว่า เราขาดพื้นฐานอะไรไปบ้าง และเราสามารถจัดลำดับความสำคัญ และเลือกที่จะศึกษาพื้นฐานเหล่านี้ก่อน. หมายเหตุ เราไม่จำเป็นต้องสร้างพื้นฐานทุก ๆ อายุ เช่น เราอาจเลือกว่า สิ่งที่เราสนใจไม่ได้เกี่ยวข้องกับงานชีวการแพทย์ เราอาจ

<sup>18</sup> ดัดแปลงจาก [4].

จะไม่เลือกเรียนรู้พื้นฐานด้านชีวเคมี ก็ได้ หรือ เรายคิดว่าทฤษฎีกระบวนการโลหะเคมีติก มีใช้บ้างกับงานบางประเภท บางแนวทาง ซึ่งเราอาจจะยังไม่สนใจ ก็สามารถทำได้. แต่ระวังว่า หากอ่านบทความวิชาการไม่รู้เรื่อง และพบว่าพื้นฐานอะไรบางที่เราต้องการ แต่เราไม่อยากเรียนรู้พื้นฐานเหล่านั้นเลย (โดยเฉพาะพื้นฐานที่จำเป็น) อาจเป็นสัญญาณเตือนว่า จริง ๆ แล้ว ใจเราอาจจะไม่อยากศึกษาหัวข้อที่เลือกจริง ๆ.

(4) จากแหล่งข้อมูลที่ได้ศึกษาเบื้องต้น เลือกแหล่งที่เกี่ยวข้องมาก ๆ ออกแบบศึกษาต่อ ให้ละเอียดขึ้น. สำหรับบทความวิจัย ให้ลองสรุปบทความ โดยระบุการค้นพบที่สำคัญ, วิธีที่ใช้, และคุณค่าเมื่อมองจากภาพรวมใหญ่ของหัวข้อ รวมถึงความสัมพันธ์กับงานอื่น ๆ. อาจอภิปรายประเด็นเพิ่มเติมด้วย เช่น ข้อจำกัด หรือ ศักยภาพ หรือการตีความจากมุมมองอื่น หรืออาจจะเป็นความเห็นส่วนตัว หรือสิ่งที่ชอบ ประเด็นที่ไม่ชอบ หากมี.

(5) ถ้าสามารถทำได้ หากกลุ่มคนที่สนใจเรื่องเดียวกัน และอภิปรายเรื่องที่เรียนรู้ต่าง ๆ ด้วยกัน (อาจเป็นกลุ่มที่สามารถพบปะตัวต่อตัว หรือกลุ่มแบบออนไลน์ก็ได้). เรื่องที่จะอภิปรายอาจจะเลือกอย่างอิสระตามความสนใจของกลุ่ม หรือหากไม่รู้จะเริ่มจากเรื่องใด อาจลองพิจารณาจากคำถามเหล่านี้ เป้าหมายที่สำคัญของหัวข้อนี้คืออะไร? หัวข้อนี้มีศักยภาพและประโยชน์ต่อสังคมในวงกว้างอย่างไร? ความท้าทายที่สำคัญของหัวข้อนี้มีอะไรบ้าง? แนวทางและวิธีการต่าง ๆ ที่ใช้เพื่อจัดการกับความท้าทาย มีอะไรบ้าง? และแต่ละแนวทางมีข้อดีข้อเสียอะไร? งานเด่น ๆ ในหัวข้อมีอะไรบ้าง ทำไมมันถึงเด่นกว่างานอื่น ๆ? อะไรคือสิ่งที่คนในวงการนี้ สนใจและอยากได้มากที่สุด? ทำไมถึงอยากได้? ในความเห็นส่วนตัวแล้ว คิดว่า นอกจากแนวทางหลัก ๆ แล้ว มีแนวทางอื่นอีกไหม? แนวทางไหนบ้างที่น่าสนใจ และทำไม? เป็นต้น

คำแนะนำในการอ่านบทความวิจัย. ก่อนอ่าน อาจจะถามตัวเองว่า ต้องการอะไรบ้าง จากบทความที่กำลังจะอ่าน. หากมีสิ่งที่ต้องการรู้เฉพาะจากบทความ เช่น อยากรู้วิธีประเมินผล เมื่ออ่านก็ให้ความสำคัญเป็นพิเศษกับวิธีประเมินผล และหลังอ่านเสร็จให้กลับมาตอบตัวเอง ว่าได้รู้สิ่งที่ค้นหาไว้อย่างไร.

แต่หากเป็นการอ่านเพื่อความเข้าใจภาพโดยทั่วไป ไม่ได้มีประเด็นที่เจาะจงเป็นพิเศษ อาจลองวิธีดังนี้ (1) อ่านผ่าน ๆ รอบแรก โดยอ่านชื่อเรื่อง บทคัดย่อ และรูปภาพ. (2) อ่านบทนำ และบทสรุป แล้วดูรูปภาพและเนื้อหาส่วนอื่น ๆ คร่าว ๆ. (3) อ่านเนื้อหาต่าง ๆ ในบทความ โดยอาจจะยังไม่ต้องสนใจรายละเอียด โดยเฉพาะนิพจน์หรือสมการคณิตศาสตร์มากนัก. (4) ทำความเข้าใจส่วนต่าง ๆ รวมถึงพจน์ นิพจน์ และสมการคณิตศาสตร์ต่าง ๆ. (5) ตั้งคำถามกับตัวเอง เช่น บทความนี้พยายามศึกษาหรือแก้ปัญหาอะไรอยู่? แนวทางหรือวิธีที่ใช้ มันมีอะไรเป็นปัจจัยสำคัญ? เนื้อหาที่อ่านมีประโยชน์อะไรบ้างกับเรา? มีอ้างอิงรายการไหนบ้างที่เราอყယาจะตามศึกษาต่อ? (6) หากสนใจบทความ อาจจะลองอภิปรายเพิ่มเติม เช่น ผลการศึกษาอาจมีข้อ

จำกัดอะไรบ้าง หรืออาจแสดงถึงศักยภาพอะไรบ้าง? จุดน่าสนใจ ความคิดสร้างสรรค์ของงานนี้ มีที่ใดบ้าง?

หลังอ่านจบแล้ว อาจลองทบทวนดูว่า มันช่วยตอบคำถามอะไรบ้างในภาพรวม ยังมีอะไรบ้างที่เป็นสิ่งที่เราสนใจอยู่? สำหรับนักศึกษาบริณญาเอก หากสิ่งที่เราสนใจ เป็นสิ่งที่ในการก็ยังไม่รู้ (ศึกษาแหล่งข้อมูลให้มากพอก เพื่อแนใจว่า ในวงการยังไม่รู้) และเป็นสิ่งที่หากรู้แล้วจะมีประโยชน์ สิ่งนั้นอาจเป็นตัวเลือกที่น่าสนใจ สำหรับหัวข้อวิจัยได้ ถ้าเราพอที่จะช่วยคลายความสงสัยนั้นลงได้บ้าง (การเลือกหัวข้อวิจัยมีความเสี่ยงสูงมาก ควรปรึกษาอาจารย์ที่ปรึกษาก่อนตัดสินใจ).

## บทที่ 6

### โครงข่ายคอนโวโลชัน

``Adapt or perish, now as ever, is nature's inexorable imperative."

---H. G. Wells

“ปรับตัว หรือ สูญพันธ์ไป เป็นความจริงของธรรมชาติที่ไม่อาจหลีกเลี่ยงได้ ทั้งตอนนี้และอนาคตเสมอมา.”

—เอช จี เวลส์

ดังที่กล่าวในบทที่ 5 โครงข่ายคอนโวโลชัน เป็นศาสตร์และศิลป์ที่สำคัญของการเรียนรู้ของเครื่อง. โครงข่ายคอนโวโลชัน อาศัยโครงสร้างที่เหมาะสมกับ ข้อมูลที่มีรูปแบบเชิงท้องถิ่นสูง ซึ่งข้อมูลในงานคอมพิวเตอร์วิทัค์ ที่มีลักษณะเช่นนี้. งานคอมพิวเตอร์วิทัค์ เป็นงานแรก ๆ ที่ โครงข่ายคอนโวโลชัน ได้พิสูจน์ให้เห็นศักยภาพของการเรียนรู้เชิงลึก และปัจจุบัน โครงข่ายคอนโวโลชัน ได้กลายเป็นศาสตร์และศิลป์ของคอมพิวเตอร์วิทัค์ในหลาย ๆ ด้าน. ยาน เลอคุน (Yann LeCun) ผู้เชี่ยวชาญการเรียนรู้เชิงลึก และผู้บุกเบิกการใช้งานโครงข่ายคอนโวโลชัน ยกย่อง[114] โครงข่ายคอนโวโลชัน ว่าเป็นส่วนที่นำพาณิชย์สำคัญมาให้กับวงการประมวลผลภาพ วิดีโอ คำพูด และเสียง.

โครงข่ายคอนโวโลชัน (Convolutional Neural Network คำย่อ CNN) เป็นโครงข่ายประสาทเทียม ที่ จำกัดการเข้ามต่อของหน่วยย่อย และมีการใช้ค่าน้ำหนักร่วมกัน. โครงข่ายคอนโวโลชัน ถูกออกแบบมาสำหรับ ข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง และรูปแบบเชิงท้องถิ่นมีลักษณะร่วมกัน. ดังนั้นเมื่อใช้กับข้อมูลที่มีลักษณะ เชิงท้องถิ่น การจำกัดการเข้ามต่อและการใช้ค่าน้ำหนักร่วมกัน จึงช่วยเพิ่มประสิทธิภาพ การคำนวณอนุมาณ และการฝึกโครงข่ายชั้นอย่างมาก เพราะ ได้ลดภาระการคำนวณของการเข้ามต่อที่มีผลกระทบน้อยกว่าไป และสามารถใช้สารสนเทศในแต่ละจุดข้อมูลได้คุ้มค่ามากขึ้น.

ข้อมูลหลาย ๆ ชนิด มีลักษณะเชิงท้องถิ่นสูง และเป็นลักษณะเชิงท้องถิ่นภายในตัวโครงสร้างของลำดับมิติ เช่น ภาพสเกลเทา ถูกแทนด้วยข้อมูลสองลำดับชั้นของค่าความเข้มพิกเซล (ลำดับค่าพิกเซลในแนวนอน และ ลำดับค่าพิกเซลในแนวตั้ง), ภาพสี ถูกแทนด้วยข้อมูลสามลำดับชั้น (ลำดับค่าพิกเซลในแนวนอน, ลำดับค่า

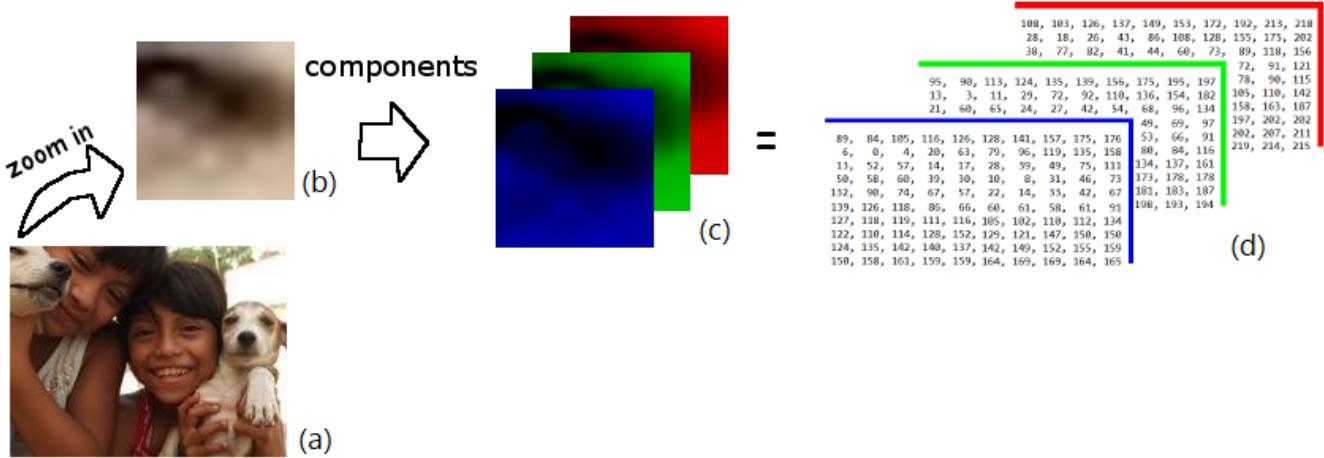
พิกเซลในแกนตั้ง, และชุดของช่องสีแดงเขียวน้ำเงิน), ภาพมัลติสเปกตรัม (multispectral image) ถูกแทนด้วยข้อมูลหลายลำดับชั้น (ลำดับค่าพิกเซลในแกนนอน, ลำดับค่าพิกเซลในแกนตั้ง, และชุดต่าง ๆ แต่ละชุดแทนค่าของแต่ละสเปกตรัม) วิดีโอoglukanแทนด้วยลำดับของภาพ, เสียงถูกแทนด้วยสเปกโตแกรมเสียง (audio spectrogram), ภาษาเขียนถูกแทนด้วยลำดับของอักษร เป็นต้น.

การที่กล่าวว่า ข้อมูลมีลักษณะเชิงท้องถิ่น คือตัวอย่าง เช่น ภาพถ่ายมุมสูงของหมู่บ้าน ภายในภาพจะเห็นบริเวณและตัวบ้านหลาย ๆ หลัง. รูปแบบของบ้านแต่ละหลัง จะสังเกตได้ทันทีจากตัวบ้านและบริเวณใกล้เคียง โดยไม่ต้องอาศัยส่วนอื่นของภาพที่อยู่ห่างออกไปประกอบ. นอกจากนั้น ภาพของหมู่บ้านภาพเดียว อาจแสดงรูปแบบของบ้านหลาย ๆ หลัง และแม้ว่าบ้านแต่หลังอาจแตกต่างกัน แต่มักจะมีลักษณะบางอย่างร่วมกัน อยู่ เช่น อาจจะเป็น ขนาด รูปทรง หรือวัสดุที่ใช้ทำหลังคา. นั่นคือ นอกจากรูปแบบของบ้านจะมีลักษณะเชิงท้องถิ่นแล้ว รูปแบบเชิงท้องถิ่นยังมีลักษณะร่วมกันอีกด้วย ซึ่งลักษณะร่วมเช่นนี้ ที่ทำให้วิธีการใช้ค่าน้ำหนักร่วมของโครงข่ายคอนโวaluชัน สามารถใช้ประโยชน์จากแต่ละจุดข้อมูลได้คุ้มค่า เช่น การเรียนรู้รูปแบบของบ้านจากส่วนภาพของบ้านหลาย ๆ หลัง ที่ปรากฏในภาพเดียวกัน.

โครงข่ายคอนโวaluชัน กับมิติและลำดับชั้นของข้อมูล. บทที่ 2 ได้อธิบาย ความหมายของมิติจากมุมมองต่าง ๆ และแนะนำความหมายของลำดับชั้น. ลำดับชั้น (rank) คือโครงสร้างลำดับมิติ หรือชุดมิติ.

โครงสร้างของโครงข่ายประสาทเทียมดังเดิม มองข้อมูลในรูปมิติที่ไม่โครงสร้างระหว่างมิติ ตัวอย่าง เช่น ข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ (แบบฝึกหัด 3.15) แต่ละจุดข้อมูลจะมี 6 เขตข้อมูล ซึ่ง เขตข้อมูลความร้ายแรงเป็นเอาร์พุต และเขตข้อมูล (1) ค่าการประเมินไบแรตส์, (2) อายุของผู้ป่วย, (3) รูปทรงของมวลเนื้อ, (4) ลักษณะขอบของมวลเนื้อ, และ (5) ความหนาแน่นของมวลเนื้อ เป็นอินพุต. นั่นคือ อินพุตของข้อมูลชุดนี้มี 5 มิติ เขตข้อมูลในแต่ละมิติ ไม่ได้เกี่ยวกันในเชิงโครงสร้าง นั่นคือ การดำเนินการ โดยสลับลำดับของเขตข้อมูล จะไม่ได้ทำให้สารสนเทศของข้อมูลเสียหาย.

เปรียบเทียบกับข้อมูลที่มิติเกี่ยวพันกันในเชิงโครงสร้าง การสลับลำดับจะทำให้สูญเสียสารสนเทศของข้อมูลไป. ดังหากพิจารณาข้อมูลในรูปแบบที่มีมิติลำดับตามธรรมชาติ เช่น ข้อมูลภาพสีในรูป 6.1. ข้อมูลภาพมีโครงสร้างมิติที่ชัดเจน นั่นคือมี (1) ชุดมิติของสี ที่มีสำหรับสีน้ำเงิน เขียว และแดง (2) ชุดลำดับมิติของพิกเซล ตามแนวตั้ง และ (3) ชุดลำดับมิติของพิกเซลตามแนวนอน. หากแต่ละค่าความเข้มพิกเซลมีค่าอยู่ระหว่าง 0 ถึง 255 สามารถใช้ เทคนิคลำดับชั้นสาม  $\mathbf{X} \in \{0, \dots, 255\}^{3 \times 133 \times 175}$  สำหรับแทนภาพสีหนึ่งภาพขนาดสูง 133 พิกเซล และกว้าง 175 พิกเซลได้. ข้อมูลลักษณะนี้ แม้ว่า แต่ละชุดข้อมูล (แต่ละภาพ) จะมี 69825 ค่า (ภาพหนึ่งภาพดังตัวอย่าง ต้องการหน่วยความจำ 69825 ไบต์) แต่ค่าเหล่านี้ มีความสัมพันธ์กัน



**รูปที่ 6.1:** ตัวอย่างข้อมูลที่เป็นภาพซึ่งมีโครงสร้างของมิติ (dimensional structure). ภาพล่างซ้าย (a) เป็นภาพสีขนาดสูง 133 พิกเซล กว้าง 175 พิกเซล. เมื่อลองขยายมุมบนซ้ายของภาพ บริเวณขนาด  $10 \times 10$  พิกเซล (ขยาย 10 เท่า) จะได้ภาพ ดังแสดงในภาพ (b). ภาพ (b) ที่เป็นภาพสี นั้นมีส่วนประกอบของสามช่องสี ดังแสดงในภาพ (c). และค่าความเข้มของพิกเซลที่ตำแหน่งต่าง ๆ ของแต่ละช่องสี แสดงดังตัวอย่างในภาพ (d). ภาพสีดังกล่าว (ภาพ a) จะมีโครงสร้างเป็นสามชุดมิติ หรือแทนด้วย เทนเซอร์ ลำดับชั้นสาม เช่น  $\mathbf{X} \in \{0, \dots, 255\}^{3 \times 133 \times 175}$  สำหรับภาพสี (3 ช่องสี) ขนาดสูง 133 พิกเซล กว้าง 175 พิกเซล แต่ละค่า แทนความเข้มของสีระหว่าง 0 ถึง 255.

เชิงลำดับด้วย เช่น ถ้าค่าพิกเซลที่อยู่ติด ๆ กันมีค่าใกล้เคียงกัน นั่น อาจหมายถึงลายเส้นที่สื่อความหมาย และแม้จะมีบางค่าพิกเซลขาดหรือเกินไปบ้าง ความหมายก็ไม่ได้เปลี่ยนแปลง. ความหมายได้จากผลเชิงรวมของค่าพิกเซลบริเวณใกล้เคียงกัน. การที่ค่าต่าง ๆ บริเวณใกล้เคียงมีความสัมพันธ์ต่อกันในเชิงความหมายรวม จะเรียกว่า ลักษณะเชิงท้องถิ่น.

โครงข่ายประสาทเทียมดังเดิม ไม่มีกลไกที่รองรับลำดับมิติ แต่ก็สามารถนำข้อมูลที่มีลำดับมิติเข้าไปประมวลผลได้ โดยการยุบชุดลำดับมิติรวมกัน เช่น จุดข้อมูล  $\mathbf{X} \in \mathbb{R}^{3 \times 133 \times 175}$  ยุบเป็นจุดข้อมูล  $\mathbf{x} \in \mathbb{R}^{69825}$ . แม้ว่า ข้อมูลที่มีลำดับมิติจะสามารถถูกยุบได้ แต่การทำเช่นนี้จะทำให้โครงสร้างธรรมชาติของข้อมูลสูญหายไป (หรือไม่ชัดเจน) และมีผลทำให้การฝึกโครงข่ายประสาทเทียม เพื่อทำงานกับข้อมูลลักษณะนี้ทำได้ยากขึ้น.

โครงข่ายคอนโวลูชัน ออกแบบมาโดยเฉพาะ เพื่อรองรับข้อมูลที่มีโครงสร้างลำดับมิติ.

เพื่อลดการสับสนกับคำว่ามิติในความหมายเดิม คำว่า ลำดับมิติ หรือ ชุดมิติ หรือ ชุดลำดับมิติ หรือ ลำดับชั้น จะถูกใช้สำหรับเน้นถึงโครงสร้างที่มีลักษณะลำดับมิติ.

กลไกที่สำคัญของโครงข่ายคอนโวลูชันที่ใช้ประโยชน์จากลักษณะโครงสร้างมิติของข้อมูล ประกอบด้วย (1) การเชื่อมต่อท้องถิ่น (local connections), (2) การใช้ค่าน้ำหนักร่วม (shared weights), (3) การตึงรวม (pooling), (4) การใช้โครงสร้างต่อกันหลายชั้น (multiple layers). กลไกเหล่านี้ ดำเนินการผ่านชั้นคำนวณสองแบบ คือ ชั้นคอนโวลูชัน (convolution layer) และ ชั้นตึงรวม (pooling layer). ชั้นคอนโวลูชัน

ทำกลไกของการเชื่อมต่อห้องถิน และการใช้ค่าน้ำหนักร่วม. ชั้นดึงรวม ทำกลไกของการเชื่อมต่อห้องถิน และการดึงรวม. โครงข่ายคอนโวลูชัน จะใช้ทั้งชั้นคอนโวลูชันและชั้นดึงรวม หลาย ๆ ชั้นต่อกัน และอาจใช้ชั้นคำนวนแบบดั้งเดิม ที่มักเรียกว่า ชั้นเชื่อมต่อเต็มที่ (fully connected layer)<sup>1</sup> สำหรับคำนวนเอาร์พุตในรูปแบบที่ต้องการ.

รูป 6.2 แสดงแผนภาพเปรียบเทียบ ชั้นเชื่อมต่อเต็มที่ ชั้นเชื่อมต่อห้องถิน และ ชั้นเชื่อมต่อห้องถินที่ใช้ค่าน้ำหนักร่วม. รูป 6.2 แสดงกรณีข้อมูลที่มีโครงสร้างหนึ่งลำดับชั้น<sup>2</sup>.

**หมายเหตุ** แนวคิดของชั้นคอนโวลูชันนั้นมีความทั่วไปมากพอ ที่จะนำไปปรับใช้กับลักษณะข้อมูลที่อาจมีโครงสร้างมิติแบบต่าง ๆ ได้. แต่เพื่อความกระชับ หัวข้อนี้อภิปรายโครงข่ายคอนโวลูชัน สำหรับข้อมูลภาพซึ่งมักมีโครงสร้างมิติสำหรับหนึ่งภาพคือ  $C \times H \times W$  เมื่อ  $C$  แทนจำนวนลักษณะอิสระ เช่น จำนวนสี และ  $H$  กับ  $W$  แทนขนาดของชุดลำดับมิติ (ขนาดของภาพ ได้แก่ ความสูงกับความกว้าง). กรณีเช่นนี้ มักถูกเรียกว่า คอนโวลูชันสองมิติ (two-dimension convolution) สำหรับสองชุดมิติลำดับสัมพันธ์ ซึ่งหมายความว่า ข้อมูล เช่น ภาพ. สังเกตว่า แต่ภาพ อาจแทนด้วย เทนเซอร์สามลำดับชั้น  $\mathbf{X} \in \mathbb{B}^{C \times H \times W}$  แต่ความสัมพันธ์ เชิงลำดับมีเฉพาะชุดมิติที่สองและสาม. ชุดมิติแรก (ชุดมิติของสี) ไม่มีความสัมพันธ์เชิงลำดับ.

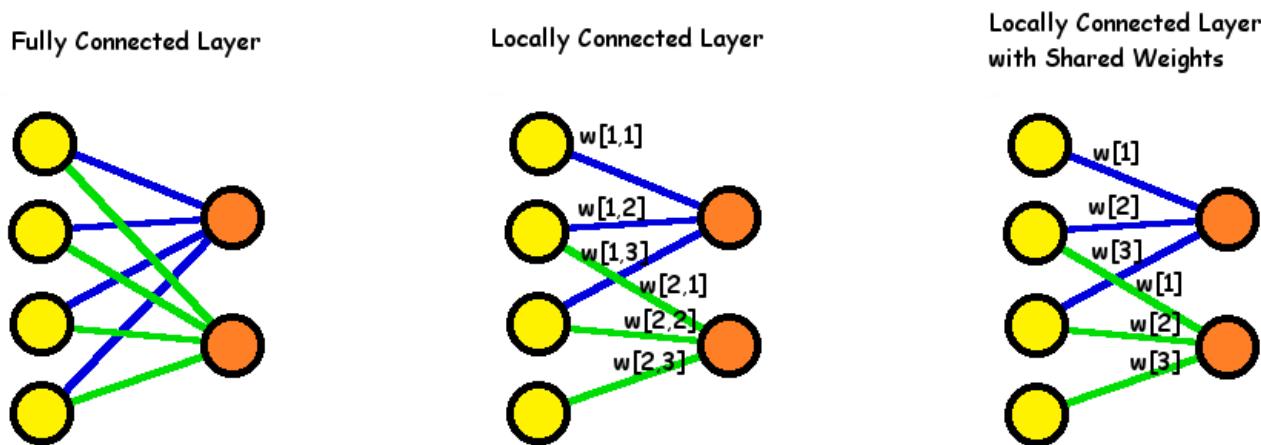
คอนโวลูชันหนึ่งมิติ (one-dimension convolution) สำหรับหนึ่งชุดมิติลำดับ จะหมายความว่า ข้อมูล เช่น เสียง โดย ชุดมิติที่มีลำดับ คือ เวลา. คอนโวลูชันสามมิติ (three-dimension convolution) จะหมายความว่า ข้อมูล เช่น วิดีโอ โดย ชุดมิติที่มีลำดับ คือ ชุดมิติพิกเซลแนวตั้ง, ชุดมิติพิกเซลแนวนอน, และชุดมิติเวลา.

## 6.1 ชั้นคอนโวลูชัน

ชั้นคอนโวลูชัน (convolution layer) เป็นชั้นคำนวน ที่มีกลไกของการเชื่อมต่อห้องถิน, การใช้ค่าน้ำหนักร่วม และการรักษาโครงสร้างมิติของเอาร์พุต. รูป 6.3 แสดงกลไกของการเชื่อมต่อห้องถิน และการใช้ค่าน้ำหนักร่วม. ค่าเอาร์พุตของชั้น  $z_i = h(a_i)$  โดย  $h(\cdot)$  แทนฟังก์ชันกราฟตัน และ คอนโวลูชันเอาร์พุต  $a_i = w_1 \cdot x_i + w_2 \cdot x_{i+1} + w_3 \cdot x_{i+2} + b$  เมื่อ  $i = 1, 2, 3$  และ  $b$  แทนค่าไบอัส (ไบอัส ไม่ได้แสดงในภาพ) เช่น  $a_1 = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b$  (ภาพซ้าย เน้นให้เห็นการคำนวนสำหรับ  $a_1$ ),  $a_2 = w_1 \cdot x_2 + w_2 \cdot x_3 + w_3 \cdot x_4 + b$  (ภาพกลาง), และ  $a_3 = w_1 \cdot x_3 + w_2 \cdot x_4 + w_3 \cdot x_5 + b$  (ภาพขวา).

<sup>1</sup> คำว่า ชั้นเชื่อมต่อเต็มที่ มักใช้เพื่อจำแนกชั้นที่มีการเชื่อมต่อแบบดั้งเดิม ออกจาก ชั้นคอนโวลูชัน

<sup>2</sup> ชั้นคอนโวลูชัน ก็คือชั้นเชื่อมต่อห้องถินที่ใช้ค่าน้ำหนักร่วม. รูป 6.2 เป็นแผนภาพอย่างง่าย ที่แสดงข้อมูลหนึ่งลำดับชั้น. การแสดงภาพของข้อมูลที่มีโครงสร้างมิติที่ซับซ้อนทำได้ยาก. อย่างไรก็ตาม รูป 6.7 แสดงตัวอย่างสำหรับทั้งข้อมูลหนึ่งลำดับชั้น และข้อมูลสองลำดับชั้น.

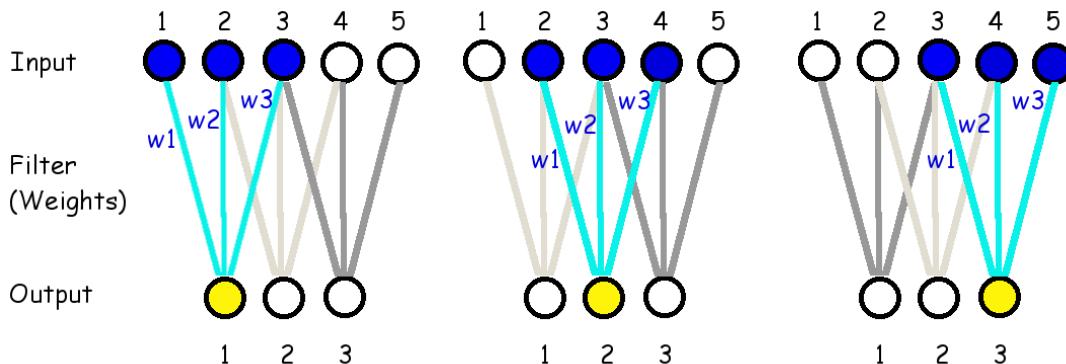


รูปที่ 6.2: ชั้นการเชื่อมต่อแบบต่าง ๆ รูปเปรียบเทียบการเชื่อมต่อเต็มที่ (ภาพซ้าย) การเชื่อมต่อห้องถิน (ภาพกลาง) และการเชื่อมต่อห้องถินที่มีการใช้ค่าน้ำหนักร่วม (ภาพขวา). อินพุต แสดงด้วย วงกลมทางซ้ายของแต่ละภาพ (อินพุตมี 4 หน่วยในแต่ละภาพ). เอาต์พุต แสดงด้วย วงกลมทางขวาของแต่ละภาพ (เอาต์พุตมี 2 หน่วยในแต่ละภาพ). เส้นตรงที่เชื่อมระหว่างอินพุตและเอาต์พุต แทนการเชื่อมต่อ หรือค่าน้ำหนักของเอาต์พุตสำหรับอินพุตต่างๆ. ภาพซ้ายแสดงชั้นเชื่อมต่อเต็มที่ เอาต์พุตแต่ละตัวมีการเชื่อมต่อ กับอินพุตทุกๆตัว จำนวนค่าน้ำหนักที่ต้องการ เท่ากับ จำนวนเอาต์พุตคูณจำนวนอินพุต (8 ค่าในภาพตัวอย่าง). ภาพกลางแสดงชั้น เชื่อมต่อห้องถิน เอาต์พุตแต่ละตัวมีการเชื่อมตอกับอินพุตแค่บางตัวเท่านั้น จำนวนค่าน้ำหนักที่ต้องการ เท่ากับ จำนวนเอาต์พุตคูณ จำนวนอินพุตที่อยู่ในห้องถิน (6 ค่าในภาพตัวอย่าง). ภาพขวาแสดงชั้นเชื่อมต่อห้องถินที่มีการใช้ค่าน้ำหนักร่วม เอาต์พุตแต่ละตัว มีการเชื่อมตอกับอินพุตแค่บางตัวเท่านั้น แต่การเชื่อมตอของเอาต์พุตแต่ละตัว จะใช้ค่าน้ำหนักชุดเดียวกัน. จำนวนค่าน้ำหนักที่ ต้องการ เท่ากับ จำนวนอินพุตที่อยู่ในห้องถิน (3 ค่าในภาพตัวอย่าง).

ว่า (1) ค่าน้ำหนัก  $w_1$ ,  $w_2$ , และ  $w_3$  ถูกใช้ร่วมกัน (การใช้ค่าน้ำหนักร่วม) และจำนวนค่าน้ำหนัก คือ 3. (2) แม้ว่าใบอัสไม่ได้แสดงในภาพ แต่ควรบันทึกไว้ว่า ค่าใบอัสก์ใช้ร่วมกัน นั่นคือ ทั้ง  $a_1, a_2, a_3$  ก็ใช้ใบอัส  $b$  ค่าเดียวกัน. (3) เอาต์พุตแต่ละตัว เชื่อมตอกับอินพุตจำนวนจำกัด ซึ่งจำนวนจะเท่ากับจำนวนค่าน้ำหนัก (การ เชื่อมต่อห้องถิน). (4) อินพุตขนาดเป็น 5 แต่เอาต์พุตมีขนาดเป็น 3. ถ้าใช้เอาต์พุตขนาดน้อยกว่า 3 จะไม่ สามารถครอบคลุมอินพุตได้ครบถ้วน. ถ้าใช้เอาต์พุตขนาดมากกว่า 3 เอาต์พุตตัวที่สี่ ตัวที่ห้า และตัวที่หก จะมีอินพุตไม่ครอบ.

เนื่องจากประวัติการพัฒนาชั้นคอนโวโลจูชั่น มาจากกลุ่มงานทางด้านการประมวลผลภาพ ค่าน้ำหนักร่วม เหล่านี้ นักถูกเรียกว่า พิลเตอร์ (filter) หรือ เคอร์แนล (kernel). จำนวนค่าน้ำหนัก ซึ่งกำหนดขนาดของ อินพุตที่เชื่อมตอกับเอาต์พุตแต่ละตัว จะถูกเรียกเป็น ขนาดของพิลเตอร์ (filter size). ขนาดของพิลเตอร์เป็น พารามิเตอร์ของแบบจำลองที่ผู้ใช้เลือก. รูป 6.4 แสดงให้เห็นการเชื่อมต่อเมื่อใช้พิลเตอร์ขนาดต่าง ๆ. การ คำนวณค่าคอนโวโลจูชั่นเอาต์พุตทำโดย

$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot x_{k+j-1} \quad (6.1)$$



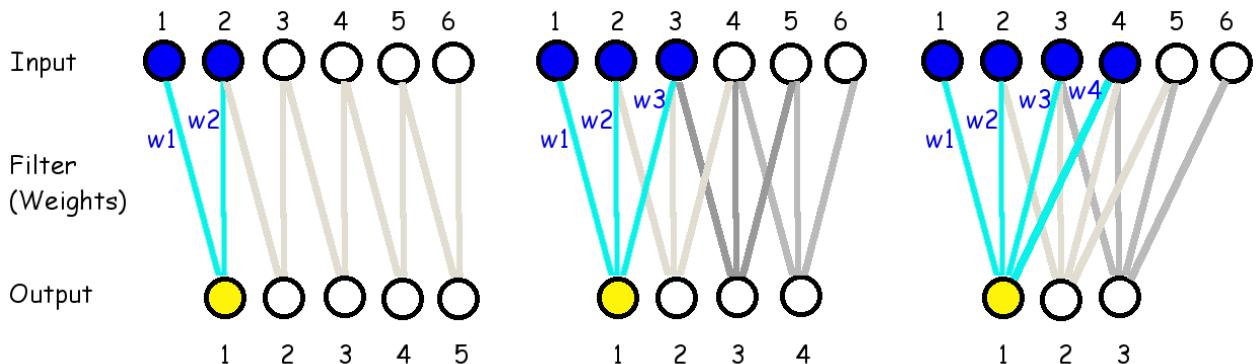
รูปที่ 6.3: แผนภาพการเชื่อมต่อของชั้นคอนโวลูชัน (convolution layer) เมื่ออินพุตมีโครงสร้างชุดมิติเดียวกับ  $\mathbf{x} \in \mathbb{R}^H$  เมื่อ  $H$  เป็นขนาดชุดลำดับมิติ (ในรูป  $H = 5$ ). เอาร์พุตแต่ละตัวเชื่อมต่อกับอินพุตจำนวนจำกัด (ในรูป ขนาดจำกัดที่ 3 ตัว) และใช้ค่าน้ำหนักร่วม ( $w_1, w_2, w_3$  ในรูปเขียน  $w1, w2$ , และ  $w3$  ตามลำดับ) นั่นคือ คอนโวลูชันเอาร์พุต  $a_i = w_1 \cdot x_i + w_2 \cdot x_{i+1} + w_3 \cdot x_{i+2} + b$  เมื่อ  $i = 1, 2, 3$  และ  $b$  แทนค่าไบอัส (ไม่ได้แสดงในภาพ). ภาพซ้ายเน้นการเชื่อมต่อของ  $a_1$  ภาพกลาง  $a_2$  และภาพขวา  $a_3$ .

สำหรับ  $k = 1, \dots, H - H_F + 1$  เมื่อ  $b$  คือค่าไบอัส. ค่าคงที่  $H_F$  คือขนาดของฟิลเตอร์. ตัวแปร  $w_j$  คือค่าน้ำหนัก. ตัวแปร  $x_i$  คืออินพุต สำหรับ  $i = 1, \dots, H$  โดย  $H$  คือขนาดของอินพุต.

สังเกต (1) ขนาดของฟิลเตอร์ยิ่งใหญ่ การเชื่อมต่อที่ยังครอบคลุมอินพุตจำนวนมากขึ้น (2) ขนาดของเอาร์พุตลดลง เช่น อินพุตขนาด 6 ใช้ฟิลเตอร์ขนาด 2 มีเอาร์พุตขนาด 5, อินพุตขนาด 6 ใช้ฟิลเตอร์ขนาด 3 มีเอาร์พุตขนาด 4, อินพุตขนาด 6 ใช้ฟิลเตอร์ขนาด 4 มีเอาร์พุตขนาด 3 เป็นต้น. เมื่อพิจารณาดูจะพบว่า อินพุตขนาด  $H$  เมื่อใช้ฟิลเตอร์ขนาด  $H_F$  จะมีเอาร์พุตขนาด  $H - H_F + 1$ .

หมายเหตุ วงการและศาสตร์การเรียนรู้ของเครื่อง จะเรียกกระบวนการดังสมการ 6.1 ว่า คอนโวลูชัน. ในขณะที่คณิตศาสตร์โดยทั่วไป และศาสตร์การประมวลผลสัญญาณ (signal processing) มักเรียกกระบวนการ เช่นนี้ ว่า ลงทะเบียนร่องรอย (cross-correlation) และใช้คำว่า คอนโวลูชัน กับปฏิบัติการ เช่น  $a_k = \sum_j w_j \cdot x_{k-j-1}$  ซึ่งมีจุดต่างสำคัญอยู่ที่การกลับลำดับของตัวถูกดำเนินการตัวหนึ่ง. เนื่องจากค่าน้ำหนักฟิลเตอร์ที่ใช้ของศาสตร์การเรียนรู้ของเครื่อง มักได้จากการบวนเรียนรู้ การทำหรือไม่ทำขั้นตอนการกลับลำดับ ไม่มีผลกับผลลัพธ์สุดท้าย. ดังนั้นการตัดขั้นตอนกลับลำดับออก ช่วยให้โปรแกรมซับซ้อนน้อยลง และทำงานได้เร็วขึ้น.

**การเติมเต็มด้วยศูนย์.** ในทางปฏิบัติ เทคนิกการเติมเต็ม (padding) หรือเรียกว่า การเติมเต็มด้วยศูนย์ (zero-padding) นักถูกนำมาใช้ เพื่อรักษาขนาดของเอาร์พุตให้เท่ากับขนาดของอินพุต (เช่น รักษาขนาดภาพของเอาร์พุต ให้เท่ากับขนาดภาพของอินพุต). นั่นคือ เมื่อใช้ฟิลเตอร์ขนาด  $H_F$  อินพุต  $\mathbf{x}$  ขนาด  $H$  จะถูกขยายเป็น  $\hat{\mathbf{x}}$  ขนาด  $H + H_F - 1$  โดยเพิ่มค่า 0 เข้าไปจนเต็มขนาด. ตัวอย่างเช่น  $H_F = 3$ ,  $\mathbf{x} =$



รูปที่ 6.4: ชั้นคอนโวลูชันที่ใช้ฟิลเตอร์ขนาดต่าง ๆ (filter sizes). ภาพซ้าย ฟิลเตอร์ขนาด 2. ภาพกลาง ฟิลเตอร์ขนาด 3. ภาพขวา ฟิลเตอร์ขนาด 4.

$\{x_1, x_2, x_3, x_4, x_5, x_6\}$  (ขนาด  $H = 6$ ) จะถูกขยายเป็น  $\hat{x} = \{0, x_1, x_2, x_3, x_4, x_5, x_6, 0\}$  ขนาดเพิ่มเป็น  $6 + 3 - 1 = 8$ . นั่นคือ เพิ่มศูนย์ 2 ตัว. หรือเมื่อใช้ฟิลเตอร์ขนาด  $H_F = 5$ , อินพุต  $x = \{x_1, x_2, x_3, x_4, x_5, x_6\}$  (ขนาด  $H = 6$ ) จะถูกขยายเป็น  $\hat{x} = \{0, 0, x_1, x_2, x_3, x_4, x_5, x_6, 0, 0\}$  ขนาดเพิ่มเป็น  $6 + 5 - 1 = 10$ . นั่นคือ เพิ่มศูนย์ 4 ตัว.

ขนาดของฟิลเตอร์ มักถูกนิยมเลือกให้เป็นเลขคี่. ขนาดของฟิลเตอร์เป็นเลขคี่ ทำให้การเติมเต็มด้วยศูนย์สามารถได้ออย่างสมดุลย์ทั้งสองปลาย<sup>3</sup>. ขนาดของฟิลเตอร์เป็นเลขคู่ก็สามารถทำได้ เพียงแต่จะมีปลายด้านหนึ่งที่จะถูกเติมมากกว่าอีกด้านเท่านั้น. รูป 6.5 แสดงแผนภาพการเข้ามต่อ เมื่อทำการเติมเต็มด้วยศูนย์. ในภาพ ขนาดของฟิลเตอร์เป็น 3 ดังนั้นต้องเติมศูนย์  $H_F - 1 = 2$  ตำแหน่ง โดยกระจายการเติมไปทั้งสองปลาย. เมื่อมีการเติมเต็ม การคำนวณค่าคอนโวลูชันอาจทำโดย

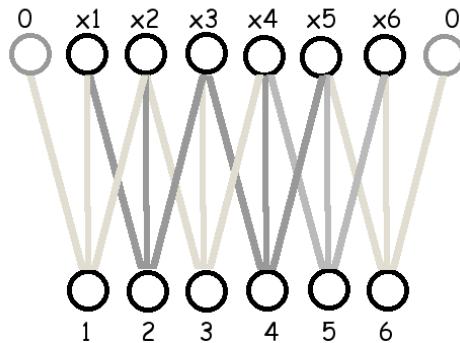
$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot \hat{x}_{k+j-1} \quad (6.2)$$

สำหรับ  $k = 1, \dots, H$  และ  $\hat{x}_i$  คืออินพุตที่ถูกเติมเต็มด้วยศูนย์ สำหรับ  $i = 1, \dots, H + H_F - 1$  โดย  $H$  คือขนาดของอินพุตเติม. สำหรับกรณีขนาดฟิลเตอร์เป็นเลขคี่ ( $H_F \bmod 2 = 1$ )

$$\hat{x}_i = \begin{cases} 0, & \text{สำหรับ } i \leq \frac{H_F-1}{2} \text{ หรือ } i > \frac{H_F-1}{2} + H, \\ x_{i-(H_F-1)/2}, & \text{นอกเหนือจากข้างต้น.} \end{cases} \quad (6.3)$$

โดย  $i = 1, \dots, H + H_F - 1$ .

<sup>3</sup>หากให้ฟิลเตอร์ขนาดเป็นเลขคี่ และก้าวย่างเป็นหนึ่ง จะทำให้จำนวนศูนย์ที่ต้องเติมเป็นเลขคู่. ดูสมการ 6.5 เพิ่มเติม.



รูปที่ 6.5: แผนภาพแสดงการเติมเต็มด้วยศูนย์ (zero padding). อินพุตเดิม  $x_1, \dots, x_6$  ขนาด 6 ถูกเติมขนาดให้เป็น 8 ด้วยค่าศูนย์ทั้งสองปลาย. เอ้าต์พุตจะมีขนาด 6 (เท่ากับขนาดอินพุตตั้งเดิม)

**ขนาดก้าวย่าง.** ตัวอย่างข้างต้น เอ้าต์พุตแต่ละหน่วยจะรับอินพุตต่างจากเอ้าต์พุตหน่วยข้าง ๆ โดยขยายลำดับอินพุตไปแค่หนึ่งตำแหน่ง. การขยายตำแหน่งของอินพุตสำหรับเอ้าต์พุตหน่วยข้าง ๆ กันนี้ ไม่จำเป็นต้องจำกัดเพียง 1 ตำแหน่งเท่านั้น. การขยายตำแหน่งนี้ สามารถทำทีละหลาย ๆ ตำแหน่ง และการขยายตำแหน่งของเอ้าต์พุตหน่วยข้าง ๆ กันนี้ จะเรียกว่า ขนาดก้าวย่าง (stride). รูป 6.6 แสดงการเชื่อมต่อ เมื่อใช้ฟิลเตอร์ขนาด 3 และ 5 กับขนาดก้าวย่างต่าง ๆ.

หากอินพุตที่เติมมีขนาด  $\hat{H}$  และฟิลเตอร์มีขนาด  $H_F$  และก้าวย่างมีขนาด  $S$  แล้วขนาดของเอ้าต์พุต  $H'$  คำนวณได้จาก

$$H' = \left\lfloor \frac{\hat{H} - H_F}{S} \right\rfloor + 1 \quad (6.4)$$

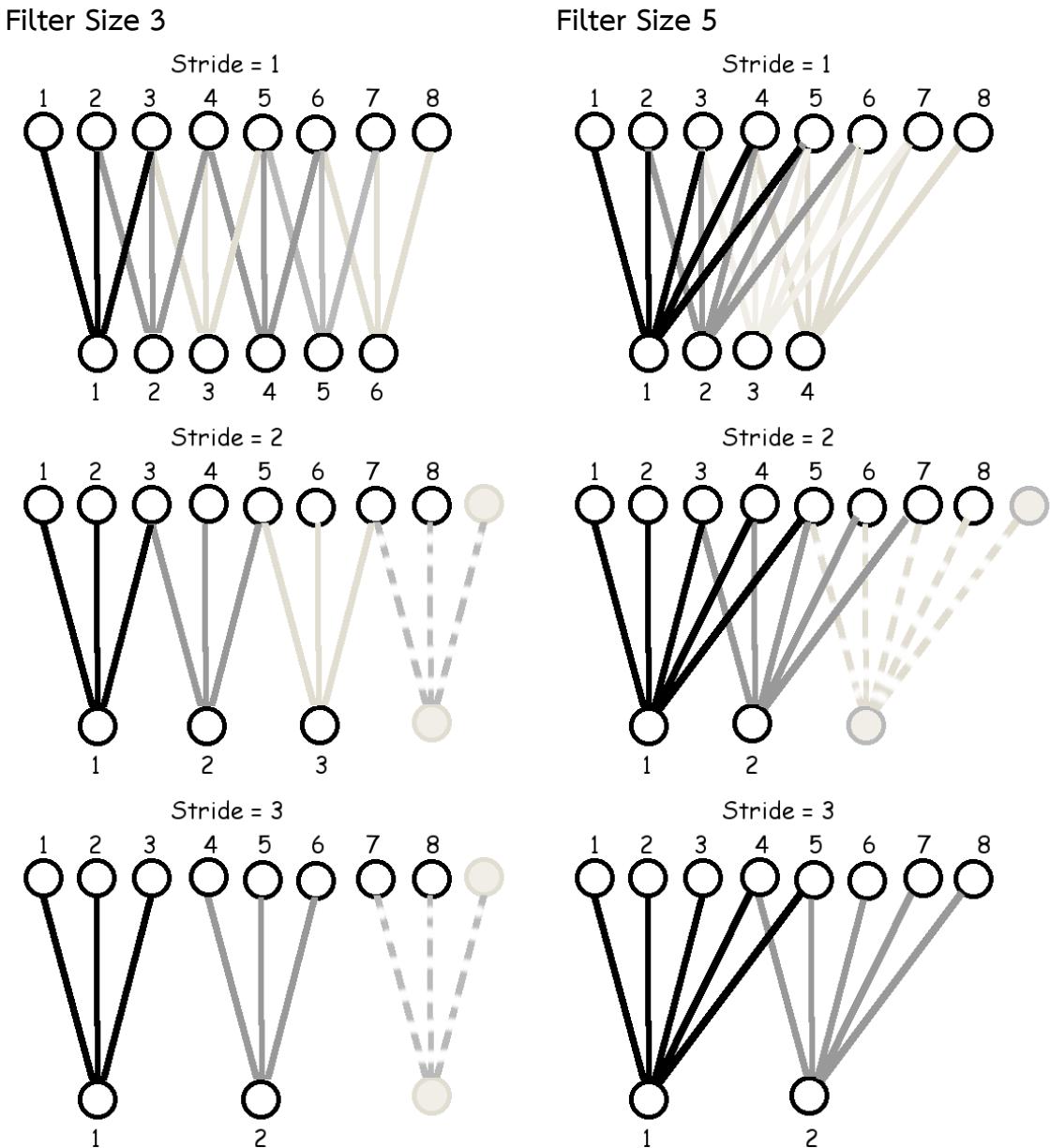
ดังนั้น หากต้องการทำการเติมเต็ม เพื่อให้ได้ขนาดของเอ้าต์พุตที่ต้องการ  $\hat{H}'$  เราสามารถคำนวณขนาดของอินพุตหลังเติมเต็ม  $\hat{H}$  ได้จาก

$$\hat{H} = S \cdot (\hat{H}' - 1) + H_F \quad (6.5)$$

และ จำนวนของศูนย์ที่ต้องเติมใส่อินพุตจะเท่ากับ  $\hat{H} - H$ .

นั่นคือ หากต้องการเติมเต็ม เพื่อให้ได้ขนาดเอ้าต์พุตเท่ากับขนาดอินพุตเดิม  $\hat{H}' = H$  จะได้ว่า จำนวนของศูนย์ที่ต้องเติมใส่อินพุตจะเท่ากับ  $S \cdot (H - 1) + H_F - H$ . หรือ หากต้องการเติมเต็ม เพื่อให้ได้ขนาดเอ้าต์พุต  $\hat{H}' = \lceil \frac{H}{S} \rceil$  จะได้ว่า จำนวนของศูนย์ที่ต้องเติมใส่อินพุตจะเท่ากับ  $S \cdot (\lceil \frac{H}{S} \rceil - 1) + H_F - H$ .

ตัวอย่างเช่น อินพุตขนาด  $H = 6$  ใช้ฟิลเตอร์ขนาด  $H_F = 3$  ใช้ขนาดย่างก้าว  $S = 1$  และต้องการให้ขนาดเอ้าต์พุตเท่ากับขนาดอินพุตเดิม จะได้  $\hat{H} = 1 \cdot (6 - 1) + 3 = 8$  และต้องเติมศูนย์ 2 ตัว. อินพุต



รูปที่ 6.6: ชั้นคอนโวลูชันที่ใช้ฟิลเตอร์ขนาด 3 (ภาพด้านซ้าย) และ ฟิลเตอร์ขนาด 5 (ภาพด้านขวา) โดยแต่ละฟิลเตอร์ใช้กับ ก้าวย่าง (strides) ขนาด 1, 2, และ 3 จากบนลงล่าง. วงกลมที่อยู่ด้านบนของภาพ แทน หน่วยอินพุต. วงกลมที่อยู่ด้านล่างของ ภาพ แทน หน่วยเอาต์พุต. เส้นตรงทึบ (เอดสี ทำเพื่อให้มองเห็นได้ชัดเจนเท่านั้น) แสดงการเชื่อมต่อ. วงกลมสีเทาและเส้นประ เน้น ความสัมพันธ์ของการเติมเต็ม ขนาดฟิลเตอร์ ขนาดย่างก้าว และความครอบคลุมอินพุต.

ขนาด  $H = 6$  ใช้ฟิลเตอร์ขนาด  $H_F = 3$  ใช้ขนาดย่างก้าว  $S = 2$  และต้องการให้ขนาดเอ้าต์พุตเท่ากับขนาดอินพุตเดิม จะได้  $\hat{H} = 2 \cdot (6 - 1) + 3 = 13$  และต้องเติมศูนย์ 7 ตัว. อินพุตขนาด  $H = 8$  ใช้ฟิลเตอร์ขนาด  $H_F = 5$  ใช้ขนาดย่างก้าว  $S = 3$  และต้องการให้ขนาดเอ้าต์พุตเท่ากับขนาดอินพุตเดิม จะได้  $\hat{H} = 3 \cdot (8 - 1) + 5 = 26$  และต้องเติมศูนย์ 18 ตัว.

ชั้นคอนโวโลชัน ที่มีอินพุต  $\mathbf{x} \in \mathbb{R}^H$  ใช้ฟิลเตอร์  $\mathbf{w} \in \mathbb{R}^{H_F}$  มีขนาดก้าวย่างเป็น  $S$  และทำการเติมเต็มด้วยศูนย์เพื่อให้เอ้าต์พุต  $\mathbf{a}$  มีขนาด  $H$  สามารถคำนวณค่าคอนโวโลชันเอ้าต์พุตแต่ละค่า ได้จาก

$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot \hat{x}_{S \cdot (k-1) + j} \quad (6.6)$$

เมื่อ  $k = 1, \dots, H$  และ  $b$  แทนค่าใบอัส. สังเกตว่า  $\mathbf{a}$  รักษาโครงสร้างมิติของ  $\mathbf{x}$  ไว้. อินพุต  $\mathbf{x}$  มีลำดับมิติเดียว คอนโวโลชันเอ้าต์พุต  $\mathbf{a}$  ก็มีลำดับมิติเดียว.

กรณีที่อภิปรายมาข้างต้น โครงสร้างมิติของอินพุตไม่ได้ซับซ้อน นั่นคือมีเพียงลำดับมิติเดียว. ปัจจัยที่สำคัญต่อมาของชั้นคอนโวโลชัน คือ กลไกที่ชัดเจนในการจัดการกับอินพุตที่มีโครงสร้างมิติที่ซับซ้อน และการยอมให้เอ้าต์พุตรักษาโครงสร้างเชิงมิติบางส่วนของอินพุตไว้ได้.

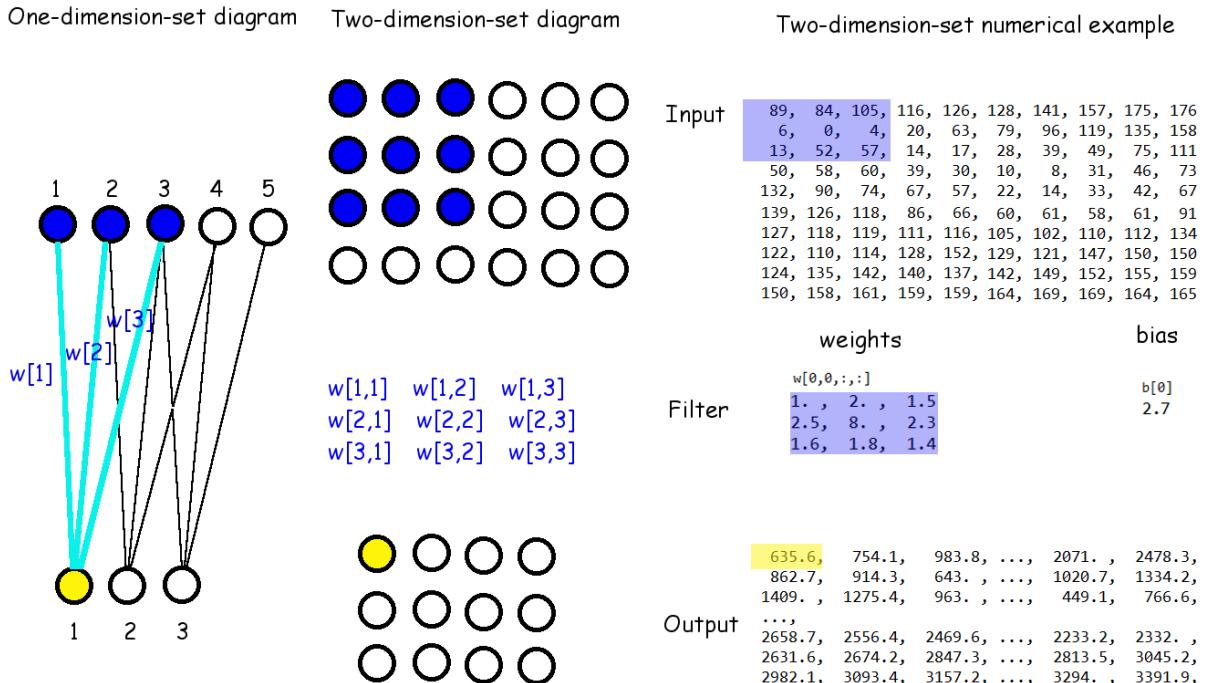
รูป 6.7 เปรียบเทียบโครงสร้างมิติเชิงเดียว (ภาพซ้าย) กับโครงสร้างมิติที่มี 2 ชุดลำดับ (ภาพกลาง) และตัวอย่างเชิงตัวเลข (ภาพขวา). จากตัวอย่างเชิงเลขในรูปจะเห็นว่า

$$\begin{aligned} \text{เอ้าต์พุต } a_{1,1} &= 2.7 + (1) \cdot 89 + (2) \cdot 84 + (1.5) \cdot 105 \\ &\quad + (2.5) \cdot 6 + (8) \cdot 0 + (2.3) \cdot 4 \\ &\quad + (1.6) \cdot 13 + (1.8) \cdot 52 + (1.4) \cdot 57 \\ &= 635.6 \end{aligned}$$

ซึ่งเอ้าต์พุตตำแหน่ง  $(1, 1)$  ถูกเน้นในภาพขวา.

$$\begin{aligned} \text{เอ้าต์พุต } a_{1,2} &= 2.7 + (1) \cdot 84 + (2) \cdot 105 + (1.5) \cdot 116 \\ &\quad + (2.5) \cdot 0 + (8) \cdot 4 + (2.3) \cdot 20 \\ &\quad + (1.6) \cdot 52 + (1.8) \cdot 57 + (1.4) \cdot 14 \\ &= 754.1 \end{aligned}$$

...



รูปที่ 6.7: แผนภาพแสดงการคำนวณชั้นคอนโวลูชัน. อินพุตแสดงอยู่ด้านบน. การเขื่อมต่อ (ค่าน้ำหนัก) แสดงตรงกลาง. และ คอนโวลูชันเอาร์พุตแสดงอยู่ด้านล่าง. ภาพซ้ายสุด แสดงแผนภาพการเขื่อมต่อของชั้นคอนโวลูชัน เมื่ออินพุตมีชุดลำดับมิติเดียว  $x \in \mathbb{R}^H$  เมื่อ  $H$  เป็นขนาดชุดลำดับมิติ (ในรูป  $H = 5$ ). ภาพกลาง แสดงแผนภาพของชั้นคอนโวลูชัน เมื่ออินพุตมีโครงสร้างชุด ลำดับมิติสองชุด  $x \in \mathbb{R}^{H \times W}$  เมื่อ  $H$  และ  $W$  เป็นขนาดชุดลำดับมิติ (ในรูป สูง  $H = 4$  กว้าง  $W = 6$ ). ภาพกลางนี้เล่นเชื่อมต่อ กัน เพื่อความไม่ยุ่งเหงิง. เอาร์พุตแต่ละตัวเขื่อมต่อ กับ อินพุตจำนวนจำกัด (ในรูป ขนาดจำกัดที่  $3 \times 3$  ตัว) และใช้ค่าน้ำหนัก ร่วม 9 ตัวดังแสดง. ภาพขวา แสดงตัวอย่างเชิงตัวเลข เมื่ออินพุต (ด้านบน) มีโครงสร้างชุดมิติสองชุด ขนาด  $10 \times 10$ . ตรงกลาง ภาพ แสดงค่าน้ำหนัก หรือ พิลเตอร์ ขนาด  $3 \times 3$  และค่าไบอส. คอนโวลูชันเอาร์พุต แสดงด้านล่าง. ไม่มีการเติมด้วยศูนย์ และ ใช้ค่าก้าวย่างเป็น  $1 \times 1$ .

$$\begin{aligned}
 \text{เอาร์พุต } a_{2,1} &= 2.7 + (1) \cdot 6 + (2) \cdot 0 + (1.5) \cdot 4 \\
 &\quad + (2.5) \cdot 13 + (8) \cdot 52 + (2.3) \cdot 57 \\
 &\quad + (1.6) \cdot 50 + (1.8) \cdot 58 + (1.4) \cdot 60 \\
 &= 862.7
 \end{aligned}$$

เป็นต้น.

สังเกตว่า (1) รูป 6.7 ไม่มีการทำการเติมด้วยศูนย์. (2) ขนาดก้าวย่างเป็น  $1 \times 1$ . นั่นคือ ขยายตาม แนวตั้งที่ลักษณะหน่วยอินพุต (พิกเซล) และขยายตามแนวโนนก์ที่ลักษณะหน่วยอินพุต. (3) คอนโวลูชันเอาร์พุต มี 2 ลำดับมิติ เช่นเดียวกับอินพุต แม้จะขนาดเล็กกว่า เพราะไม่ได้ทำการเติมเต็ม. และ เพื่อเน้นถึงโครงสร้าง มิติที่รักษาไว้ นี่ คอนโวลูชันเอาร์พุต อาจจะถูกเรียกว่า แผนที่คอนโวลูชัน (convolution map) เพื่อกันการ สับสนกับเอาร์พุตสุดท้ายของโครงข่ายทั้งหมด ซึ่งเอาร์พุตสุดท้าย คือ เอาร์พุตของชั้นสุดท้ายของโครงข่าย.

นั่นคือ เมื่ออินพุต  $\mathbf{X} \in \mathbb{R}^{H \times W}$  และเลือกใช้ฟิลเตอร์ขนาด  $H_F \times W_F$  ซึ่งหมายถึง  $\mathbf{W} \in \mathbb{R}^{H_F \times W_F}$  และเลือกก้าวย่างขนาด  $S_H \times S_W$  แล้ว (ทำนองเดียวกับสมการ 6.6) คอนโวโลชันเอาร์พุต  $\mathbf{A} \in \mathbb{R}^{H \times W}$  สามารถคำนวณได้จาก

$$a_{k,l} = b + \sum_{i=1}^{H_F} \sum_{j=1}^{H_W} w_{ij} \cdot \hat{x}_{S_H \cdot (k-1) + i, S_W \cdot (l-1) + j} \quad (6.7)$$

$k = 1, \dots, H, l = 1, \dots, W$  โดย  $\hat{x}$  คืออินพุตที่ผ่านการเติมเต็ม และ  $b$  คือไบอัส.

อย่างไรก็ตาม เมื่อพิจารณารูป 6.1 จะเห็นว่า ข้อมูลภาพสีหนึ่งภาพมีโครงสร้างมิติเป็น 3 ชุดมิติ ได้แก่ ชุดสำหรับช่องสีขนาด 3, ชุดลำดับแนวตั้งขนาด 133 และชุดลำดับสำหรับแนวอนขนาด 175. นั่นคือ มีอินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ . โดยทั่วไป ชั้นคอนโวโลชัน จะเลือกจัดการกับชุดมิติช่องสีเป็นเสมอ ชุดมิติที่ไม่มีลำดับ แต่ ชุดลำดับแนวตั้งและแนวอนแบบชุดมิติมีลำดับ. การคำนวณคอนโวโลชันเอาร์พุต  $\mathbf{A} \in \mathbb{R}^{H \times W}$  สำหรับกรณีนี้ (โครงสร้างมีชุดมิติที่มีลำดับ และไม่มีลำดับ) จะทำดังสมการ 6.8

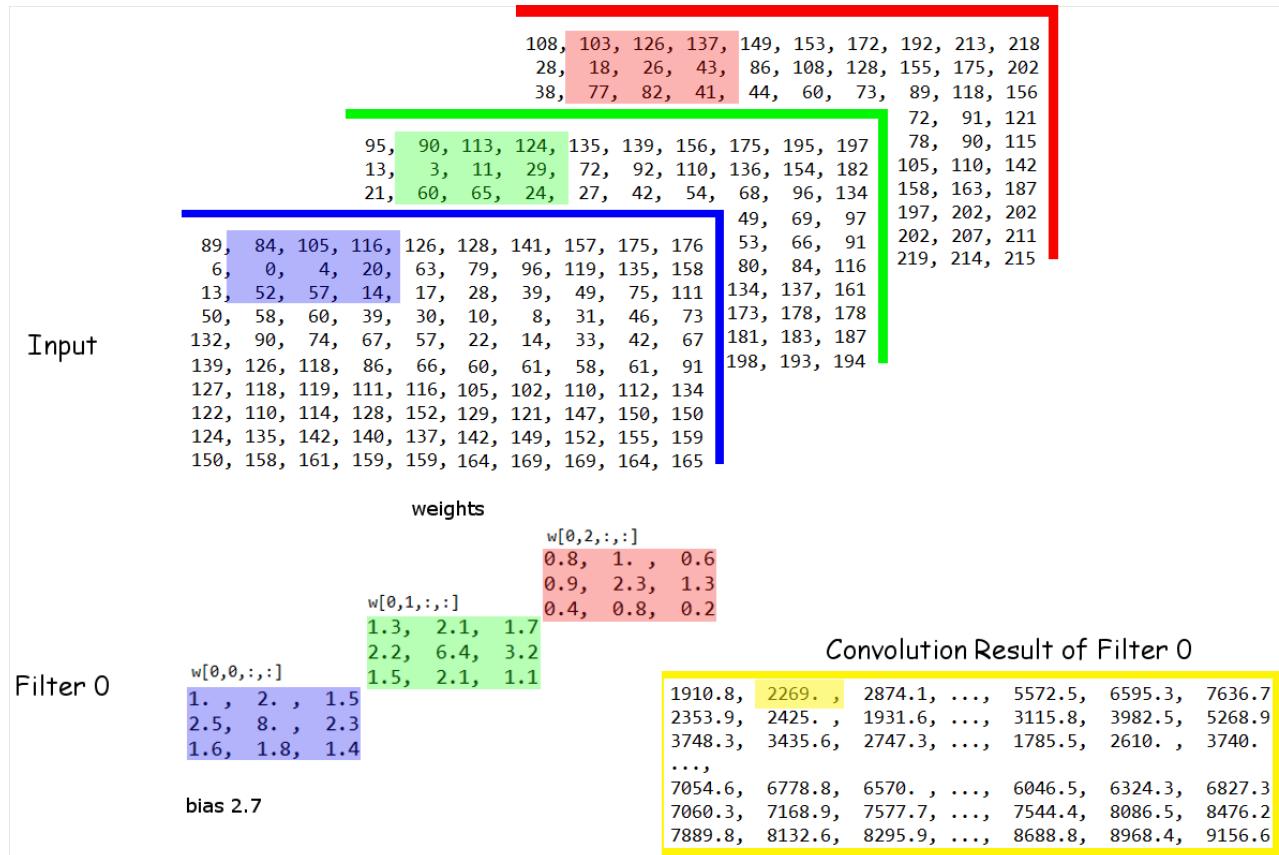
$$a_{k,l} = b + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{H_W} w_{cij} \cdot \hat{x}_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j} \quad (6.8)$$

$k = 1, \dots, H$  และ  $l = 1, \dots, W$  เมื่อเลือกใช้ฟิลเตอร์ขนาด  $H_F \times W_F$  ซึ่งหมายถึง  $\mathbf{W} \in \mathbb{R}^{C \times H_F \times W_F}$  เพราะ ชุดมิติแรกของฟิลเตอร์จะต้องมีขนาดเท่ากับขนาดชุดมิติที่ไม่มีลำดับของอินพุต, ก้าวย่างขนาด  $S_H \times S_W$ , และ  $\hat{x}$  คืออินพุตที่ผ่านการเติมเต็ม.

รูป 6.8 แสดงการคำนวณคอนโวโลชัน เมื่ออินพุตมีโครงสร้าง 3 ชุดมิติ โดยชุดมิติแรกไม่มีลำดับ. รูปเน้นให้เห็นความสัมพันธ์ระหว่างเอาร์พุต  $a_{1,2}$  ที่เข้ามายิงกับ อินพุต  $x_{1,1,2}, \dots, x_{1,3,4}$  (ช่องสีน้ำเงิน),  $x_{2,1,2}, \dots, x_{2,3,4}$  (ช่องสีเขียว),  $x_{3,1,2}, \dots, x_{3,3,4}$  (ช่องสีแดง) ดังที่เน้นคำในภาพ. จากตัวอย่างเชิงเลขในรูปจะเห็นว่า

$$\text{เอาร์พุต } a_{1,2} = 2.7 \quad (\text{ไบอัส})$$

$$\begin{aligned} &+ (1) \cdot 84 + (2) \cdot 105 + (1.5) \cdot 116 \\ &+ (2.5) \cdot 0 + (8) \cdot 4 + (2.3) \cdot 20 \\ &+ (1.6) \cdot 52 + (1.8) \cdot 57 + (1.4) \cdot 14 \quad (\text{ช่องสีน้ำเงิน รวม 751.4}) \\ &+ (1.3) \cdot 90 + (2.1) \cdot 113 + (1.7) \cdot 124 \\ &+ (2.2) \cdot 3 + (6.4) \cdot 11 + (3.2) \cdot 29 \\ &+ (1.5) \cdot 60 + (2.1) \cdot 65 + (1.1) \cdot 24 \quad (\text{ช่องสีเขียว รวม 987.8}) \end{aligned}$$



รูปที่ 6.8: การคำนวณค่าคอนโวลูชันที่ตำแหน่ง  $(1, 2)$ . อินพุตมีโครงสร้างมิติที่ชั้บช้อน นั่นคือ มี 3 ช่องสี และแต่ละช่องสีมีค่าความเข้มพิกเซลตามลำดับแนวตั้งและแนวนอน. พิลเตอร์ขนาด  $3 \times 3$  ที่ใช้ จะต้องมีชุดมิติแรกขนาดเป็น 3 เช่นกัน (รับกับ 3 ช่องสีของอินพุต). เอ้าต์พุตจะรักษาโครงสร้างมิติเชิงลำดับไว้ นั่นคือ อินพุตขนาด  $C \times H \times W$  จะแปลงมาเป็น เอ้าต์พุตขนาด  $(H - 2) \times (W - 2)$  เนื่องจากขนาดพิลเตอร์เป็น  $3 \times 3$  ขนาดก้าวย่างเป็น  $1 \times 1$  และไม่มีการเติมเต็ม.

$$\begin{aligned}
 & + (0.8) \cdot 103 + (1) \cdot 126 + (0.6) \cdot 137 \\
 & + (0.9) \cdot 18 + (2.3) \cdot 26 + (1.3) \cdot 43 \\
 & + (0.4) \cdot 77 + (0.8) \cdot 82 + (0.2) \cdot 41 \quad (\text{ช่องสีแดง รวม } 527.1) \\
 & = 2269.0
 \end{aligned}$$

เป็นต้น.

กลไกของคอนโวลูชันที่กล่าวมา เป็นเพียงการคำนวณค่าของคอนโวลูชันเอ้าต์พุต การใช้งานชั้นคอนโวลูชันในโครงข่ายแบบลีกนั้น หลังจากทำการคำนวณคอนโวลูชัน (ผลรวมเชิงเส้นระหว่างอินพุตกับค่าน้ำหนัก) แล้ว ค่าที่ได้จะนำไปผ่านฟังก์นกระดูน เพื่อให้ได้ผลลัพธ์ของชั้นคอนโวลูชัน ดังแสดงในสมการ 6.9,

$$z_{k,l} = h(a_{k,l}) \quad (6.9)$$

เมื่อ  $h(\cdot)$  คือฟังก์ชันการกระดูน (เช่น เรคติไฟฟ์ลิเนียร์, สมการ 5.1) และ  $a_{k,l}$  คือคอนโวลูชันเอ้าต์พุต (สม-

การ 6.8). เอ้าต์พุต  $\mathbf{Z} \in \mathbb{R}^{H \times W}$  นี้ อาจถูกเรียกว่า แผนที่ลักษณะสำคัญ (feature map).

การใช้กลไกของการเชื่อมต่อห้องถิน ร่วมกับการใช้ค่าน้ำหนักร่วม จะเปรียบเสมือนการทำเทคนิคหน้าต่างเลื่อน(หัวข้อ 4.1) เพื่อค้นหารูปแบบ ที่อาจปรากฏอยู่ตำแหน่งต่าง ๆ ในอินพุตได้. นั่นคือ เมื่อ้อนกับการเลื่อนฟิลเตอร์ (เทียบเท่ากับหน้าต่าง) ไปตำแหน่งต่าง ๆ ของอินพุต เพื่อค้นหารูปแบบลักษณะที่สำคัญ. การใช้ฟิลเตอร์หนึ่งตัว ก็เปรียบเสมือน รูปแบบของลักษณะสำคัญนั้นรูปแบบ. การใช้งานโครงข่ายคอนโวลูชัน ในทางปฏิบัติ จะใช้ฟิลเตอร์หลาย ๆ ตัว ซึ่งเปรียบเสมือน การค้นหารูปแบบของลักษณะสำคัญหลาย ๆ รูปแบบ และผลลัพธ์ ก็จะได้แผนที่ลักษณะสำคัญหลาย ๆ แผนที่.

ทบทวนการคำนวณชั้นคอนโวลูชัน สำหรับอินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$  และฟิลเตอร์ขนาด  $H_F \times W_F$  ทั้งหมด  $F$  ตัว นั่นคือ  $\mathbf{W} \in \mathbb{R}^{F \times C \times H_F \times W_F}$  และเอ้าต์พุต  $\mathbf{Z} \in \mathbb{R}^{F \times H \times W}$  หาได้จาก

$$a_{f,k,l} = b_f + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{fcij} \cdot \hat{x}_{c,S_H \cdot (k-1)+i, S_W \cdot (l-1)+j} \quad (6.10)$$

$$z_{f,k,l} = h(a_{f,k,l}) \quad (6.11)$$

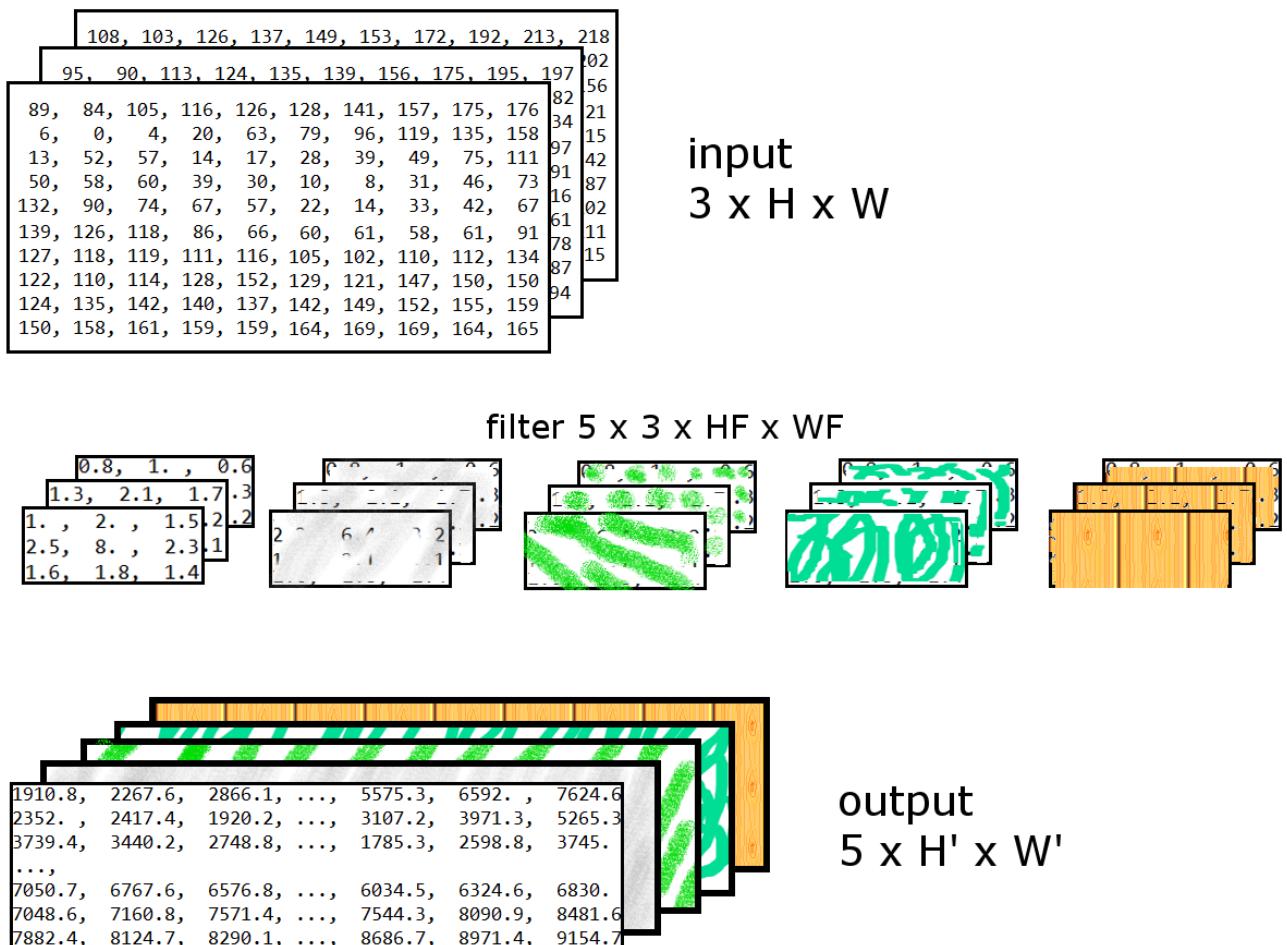
$f = 1, \dots, F$ ,  $k = 1, \dots, H$  และ  $l = 1, \dots, W$  เมื่อ  $S_H$ ,  $S_W$  คือขนาดก้าวย่าง ตามแนวตั้งและนอน ตามลำดับ. พิงก์ชัน  $h(\cdot)$  คือ พิงก์ชันกระตุ้น, ตัวแปร  $b_f$  คือใบอัสร่วมของฟิลเตอร์  $f$  ส่วนตัวแปร  $\hat{x}$  คือ อินพุตที่ผ่านการเติมเต็ม.

รูป 6.9 แสดงแผนภาพโครงสร้างมิติของอินพุต ฟิลเตอร์ และเอ้าต์พุต ของชั้นคอนโวลูชัน. ในรูป ใช้ 5 ฟิลเตอร์ และผลลัพธ์ก็จะได้แผนที่ลักษณะ 5 แผนที่. แต่ละแผนที่ระบุการตอบสนองต่อรูปแบบสำคัญที่ตำแหน่งต่าง ๆ สำหรับแต่ละรูปแบบสำคัญ. สังเกตว่าเอ้าต์พุตของชั้นคอนโวลูชัน  $\mathbf{Z} \in \mathbb{R}^{F \times H' \times W'}$  ซึ่งมีโครงสร้าง มิติเป็น 3 ชุด โดยชุดแรกไม่สำคัญ สองชุดหลังเป็นชุดลำดับที่สัมพันธ์กัน เช่นเดียวกับอินพุต ดังนั้น เอ้าต์พุต จากชั้นคอนโวลูชันชั้นหนึ่ง สามารถส่งต่อไปเป็นอีกชั้นหนึ่งได้เลย โดยไม่ต้องมีการตัดเปลี่ยนใด ๆ.

**สนามรับรู้.** สนามรับรู้ (receptive field) สำหรับโครงข่ายคอนโวลูชัน คือบริเวณพื้นที่ห้องถินของอินพุต ที่หน่วยย่อยที่สนใจครอบคลุมถึง. ขนาดของสนามรับรู้ สามารถคำนวณได้จาก

$$R_k = 1 + \sum_{j=1}^k (F_j - 1) \prod_{i=0}^{j-1} S_i \quad (6.12)$$

เมื่อ  $R_k$  เป็นขนาดของสนามรับรู้ และ  $F_j$  เป็นขนาดฟิลเตอร์ของชั้นที่  $j$  และ  $S_i$  เป็นขนาดก้าวย่างของชั้นที่  $i$  และกำหนดให้  $S_0 = 1$ .



รูปที่ 6.9: โครงสร้างมิติของอินพุต พิลเตอร์ และเอาต์พุตของชั้นคณโนว์ลุชั่น. มีพิลเตอร์ขนาด  $HF \times WF$  อยู่ 5 ตัว ดังนั้น เอาต์พุตซึ่งเป็นแผนที่ลักษณะ ก็จะมีอยู่ 5 แผนที่ แต่ละแผนที่ลักษณะคำนวนจากพิลเตอร์แต่ละตัว.

ตัวอย่างเช่น โครงข่ายคณโนว์ลุชั่นสองชั้น ที่ชั้นแรกใช้พิลเตอร์ขนาด  $3 \times 3$  ก้าวย่างเป็น 1 และชั้นที่สองก็ใช้พิลเตอร์  $3 \times 3$  และก้าวย่างเป็น 1 แล้ว แต่ละหน่วยอยู่ในชั้นที่สอง จะครอบคลุมพื้นที่ขนาด  $3 \times 3$  ในชั้นที่หนึ่ง และจะครอบคลุมพื้นที่ขนาด  $5 \times 5$  ของอินพุต. นั่นคือ  $R_2 = 1 + (F_2 - 1)S_0 \cdot S_1 + (F_1 - 1)S_0 = 1 + (2)(1)(1) + (2)(1) = 5$ .

## 6.2 ชั้นดึงรวม

ชั้นดึงรวม (pooling layer) ถูกนำมาใช้สำหรับ (1) การทำซับแซมบลิง (subsampling) เพื่อลดจำนวนข้อมูลลง และ (2) การสรุปลักษณะสำคัญในรูปแบบไกล์เคียง. กลไกการทำงานของชั้นดึงรวมจะคล้ายกับชั้นคณโนว์ลุชั่นที่จะขยับไปทีละก้าวย่าง เพื่อประมวลผลอินพุตในบริเวณจำกัด (การเข้มต่อท้องถิ่น) แต่การประมวลของชั้นดึงรวมจะแยกมิติที่ไม่มีลำดับออกจากกัน ไม่นำมาประมวลผลรวมกัน. นั่นคือ อินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$

โดย  $C$  คือชุดมิติที่ไม่มีลำดับ และ  $H$  และ  $W$  คือชุดมิติที่มีลำดับสัมพันธ์. เอาร์พุตของชั้นดึงรวม  $\mathbf{Z} \in \mathbb{R}^{C \times H' \times W'}$  สามารถคำนวณได้ดังนี้

$$z_{c,k,l} = g(\{\hat{x}_{c,S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}\}_{i=1, \dots, H_F, j=1, \dots, W_F}) \quad (6.13)$$

$c = 1, \dots, C$ ,  $k = 1, \dots, H'$  และ  $l = 1, \dots, W'$  เมื่อ  $S_H, S_W$  คือขนาดก้าวย่าง ตามแนวตั้งและนอน ตามลำดับ,  $g(\cdot)$  คือ ฟังก์ชันดึงรวม, และ  $\hat{x}$  คืออินพุตที่ผ่านการเติมเต็ม เพื่อให้เอาร์พุตครอบคลุมอินพุตทุก ตัว. สังเกตว่า การดึงรวมจะไม่ยุบชุดมิติอิสระ  $C$ . เอาร์พุตที่ได้ยังคงมีขนาด  $C$  สำหรับชุดมิติแรกเช่นเดิม.

การทำการเติมเต็มสำหรับชั้นดึงรวม จะมีจุดประสงค์ต่างจากสำหรับชั้นคอนโวลูชัน ตรงที่ ชั้นดึงรวมไม่ ต้องการรักษาขนาดชุดมิติของอินพุตไว้ ชั้นดึงรวมต้องการลดขนาดชุดมิติลำดับนี้ลง ( $H' < H$  และ  $W' < W$ ) แต่การเติมเต็มสำหรับชั้นดึงรวม ทำเพื่อให้เอาร์พุตสามารถครอบคลุมอินพุตได้ทุกตัว. ตัวอย่างเช่น ชั้น ดึงรวมแบบมากที่สุด (max-pooling Layer) ขนาด  $5 \times 5$  ใช้ก้าวย่างขนาด  $2 \times 2$  หากอินพุต  $\mathbf{X}$  มีขนาด  $3 \times 60 \times 80$  จะพบว่า หากไม่ทำการเติมเต็ม อินพุตตัวสุดท้ายของแต่ละชุดมิติอิสระ  $\{x_{1,60,80}, x_{2,60,80}, x_{3,60,80}\}$  จะถูกรับผิดชอบโดยเอาร์พุต ที่ตำแหน่ง (29, 39) ซึ่ง (จากสมการ 6.13),

$$\begin{aligned} z_{c,29,39} &= \max\{\hat{x}_{c,56+i,76+j}\}_{i=1, \dots, 5, j=1, \dots, 5} \\ &= \max \left\{ \begin{array}{l} \hat{x}_{c,57,77}, \hat{x}_{c,57,78}, \hat{x}_{c,57,79}, \hat{x}_{c,57,80}, \hat{x}_{c,57,81}, \\ \hat{x}_{c,58,77}, \hat{x}_{c,58,78}, \hat{x}_{c,58,79}, \hat{x}_{c,58,80}, \hat{x}_{c,58,81}, \\ \hat{x}_{c,59,77}, \hat{x}_{c,59,78}, \hat{x}_{c,59,79}, \hat{x}_{c,59,80}, \hat{x}_{c,59,81}, \\ \hat{x}_{c,60,77}, \hat{x}_{c,60,78}, \hat{x}_{c,60,79}, \hat{x}_{c,60,80}, \hat{x}_{c,60,81}, \\ \hat{x}_{c,61,77}, \hat{x}_{c,61,78}, \hat{x}_{c,61,79}, \hat{x}_{c,61,80}, \hat{x}_{c,61,81} \end{array} \right\}, \end{aligned}$$

$c = 1, 2, 3$  ต้องการค่าอินพุตที่ตำแหน่งแนวนอนที่ 61 และตำแหน่งแนวนอนที่ 81 ซึ่งอินพุตตั้งเดิมนั้นไม่มี. ดังนั้น  $\hat{x}$  จึงต้องมีการเติมเต็มในตำแหน่งตั้งกล่าว เพื่อให้การดึงรวมสามารถครอบคลุมอินพุตตั้งเดิมได้ทั้งหมด.

แต่หากว่าการดึงรวมสามารถครอบคลุมอินพุตตั้งเดิมได้ทั้งหมดแล้วก็ไม่จำเป็นต้องมีการเติมเต็ม. ตัว-อย่างเช่น ชั้นดึงรวมแบบมากที่สุด ขนาด  $3 \times 3$  ใช้ก้าวย่างขนาด  $3 \times 3$  (ไม่มีการซ้อนทับ) หากอินพุต  $\mathbf{X}$  มี ขนาด  $3 \times 90 \times 120$  จะพบว่า แม้ไม่ทำการเติมเต็ม อินพุตตัวสุดท้ายของแต่ละชุดมิติอิสระ  $\{x_{1,90,120}, x_{2,90,120}, x_{3,90,120}\}$  จะถูกรับผิดชอบโดยเอาร์พุต ที่ตำแหน่ง (30, 40) พอดี โดยไม่ต้องการการเติม

เต็ม/เพิ่ม,

$$\begin{aligned} z_{c,30,40} &= \max\{\hat{x}_{c,87+i,3 \cdot 117+j}\}_{i=1,\dots,3,j=1,\dots,3} \\ &= \max \left\{ \begin{array}{l} \hat{x}_{c,88,118}, \hat{x}_{c,88,119}, \hat{x}_{c,88,120}, \\ \hat{x}_{c,89,118}, \hat{x}_{c,89,119}, \hat{x}_{c,89,120}, \\ \hat{x}_{c,90,118}, \hat{x}_{c,90,119}, \hat{x}_{c,90,120} \end{array} \right\}. \end{aligned}$$

นั่นคือ หาก  $(H - H_F) \bmod S_H = 0$  และ  $(W - W_F) \bmod S_W = 0$  ก็ไม่จำเป็นต้องทำการเติมเต็ม.

เมื่อทำการเติมตามความจำเป็น เพื่อให้การดึงรวมครอบคลุมอินพุตทุกหน่วยแล้ว ขนาดของเอาร์พุต (หรืออาจเรียกว่า แผนที่เอาร์พุต เพื่อเน้นโครงสร้างเชิงมิติ 2 ชุดลำดับที่สัมพันธ์กัน) จะเป็น  $H' \times W'$  โดย

$$H' = \left\lceil \frac{H - H_F}{S_H} \right\rceil + 1 \quad (6.14)$$

$$W' = \left\lceil \frac{W - W_F}{S_W} \right\rceil + 1. \quad (6.15)$$

ดังนั้น หากทำการดึงรวมด้วยฟิลเตอร์ขนาด  $2 \times 2$  ก้าวย่างขนาด  $2 \times 2$  ซึ่งเป็นการดึงรวมที่มักนิยมใช้ และอินพุตมีขนาด  $H \times W$  จะได้ แผนที่เอาร์พุตที่มีขนาด  $\lceil \frac{H}{2} \rceil \times \lceil \frac{W}{2} \rceil$  หรือ  $\frac{H}{2} \times \frac{W}{2}$  หาก  $H$  และ  $W$  เป็นเลขคู่ (หรือหาก  $H$  และ  $W$  เป็นเลขคี่ ขนาดของเอาร์พุตจะเป็น  $(1 + \frac{H}{2}) \times (1 + \frac{W}{2})$  ซึ่งโดยมาก  $H \gg 1, W \gg 1$ , ขนาดก็จะ  $\approx \frac{H}{2} \times \frac{W}{2}$ ). นั่นคือ ขนาดแผนที่เอาร์พุตจะลดลงเหลือครึ่งหนึ่งของขนาด แผนที่อินพุต

ฟังก์ชันดึงรวม นิยมใช้ ฟังก์ชันทางสถิติ อาทิ ฟังก์ชันดึงรวมแบบมากที่สุด และ ฟังก์ชันดึงรวมแบบเฉลี่ย (average pooling) เป็นต้น. การทำงานของฟังก์ชันเหล่านี้ก็ตรงไปตรงมา คือ ฟังก์ชันดึงรวมแบบมากที่สุด จะให้ค่าที่มากที่สุดในเซตอุปกรณ์ ฟังก์ชันดึงรวมแบบเฉลี่ย จะให้ค่าเฉลี่ยของค่าต่าง ๆ ในเซตอุปกรณ์.

### 6.3 เกรเดียนต์ของโครงข่ายคอนโวลูชัน

ในขั้นตอนการฝึกโครงข่าย ขั้นตอนวิธีที่ใช้ มักอาศัยค่าเกรเดียนต์ (ดูหัวข้อ 3.3). การใช้งานขั้นตอนโวลูชัน และขั้นดึงรวม ก็มีผลต่อเกรเดียนต์. นอกจากนั้น ขั้นตอนโวลูชันเองก็มีค่าน้ำหนักที่ต้องการการฝึกด้วย.

## เกรเดียนต์ของชั้นคอนเวลูชัน

เปรียบเทียบกับหัวข้อ 3.3 พิจารณาความสัมพันธ์ ระหว่างฟังก์ชันเป้าหมายและชั้นคอนเวลูชัน ฟังก์ชันเป้าหมาย  $E_n$  จะขึ้นกับค่าน้ำหนัก  $w_{fcij}$  (สมการ 6.10 และ 6.11) ผ่านชั้นคอนเวลูชันเอาร์พุต  $a_{fkl}$  สำหรับทุก ๆ  $k$  และ  $l$  ที่ใช้  $w_{fcij}$  ร่วม. ดังนั้นจากกฎลูกโซ่

$$\frac{\partial E_n}{\partial w_{fcij}} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \frac{\partial E_n}{\partial a_{fkl}} \frac{\partial a_{fkl}}{\partial w_{fcij}} \quad (6.16)$$

เมื่อแทนค่าสมการ 6.10 เข้าไปในสมการ 6.16 จะได้

$$\frac{\partial E_n}{\partial w_{fcij}} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \frac{\partial E_n}{\partial a_{fkl}} \hat{x}_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j} \quad (6.17)$$

โดย  $\hat{x}_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}$  เป็นอินพุตของชั้นคำนวณ (layer).

เมื่อกำหนด

$$\delta_{fkl} \equiv \frac{\partial E_n}{\partial a_{fkl}} \quad (6.18)$$

จะได้

$$\frac{\partial E_n}{\partial w_{fcij}} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \delta_{fkl} \hat{x}_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j} \quad (6.19)$$

สำหรับ ดัชนี  $f = 1, \dots, F$ , ดัชนี  $c = 1, \dots, C$ , ดัชนี  $i = 1, \dots, H_F$  และดัชนี  $j = 1, \dots, W_F$  เมื่อ  $\hat{x}_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}$  คือ อินพุตของชั้นคอนเวลูชัน ที่ผ่านการเติมเต็ม,  $S_H$  และ  $S_W$  เป็นค่าก้าวย่างตามแนวตั้งและนอน.

เมื่อเปรียบเทียบกับสมการ 3.28 จะเห็นว่าคล้ายกันมาก ต่างกันเพียง (1) สมการ 6.19 มีการบวกพจน์ต่าง ๆ ที่ใช้ค่าน้ำหนักร่วมกัน (2) ค่าน้ำหนักร่วมกันไม่ได้เฉพาะเจาะจงกับเอาร์พุต (เพราะใช้ค่าน้ำหนักร่วมกัน), (3) จำนวนค่าน้ำหนักมีน้อยเมื่อเทียบกับจำนวนอินพุตและเอาร์พุต (ปกติ  $H_F \ll H$  และ  $W_F \ll W$ ).

ในทำนองเดียวกัน เกรเดียนต์ของใบอัส สามารถหาได้จาก

$$\frac{\partial E_n}{\partial b_f} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \delta_{fkl} \cdot \quad (6.20)$$

พิจารณา  $\delta_{fkl}$  ของชั้นที่  $m^{th}$  ว่า  $a_{fkl}^{(m)}$  เชื่อมไปสู่เอ้าต์พุตสุดท้ายและฟังก์ชันจุดประสีค์  $E_n$  ผ่านเอ้าต์พุตของชั้น  $z_{fkl}^{(m)}$  และจากกฎลูกโซ่ จะได้

$$\delta_{fkl}^{(m)} = \frac{\partial E_n}{\partial a_{fkl}^{(m)}} = \frac{\partial E_n}{\partial z_{fkl}^{(m)}} \cdot \frac{\partial z_{fkl}^{(m)}}{\partial a_{fkl}^{(m)}}$$

โดยตัวยก  $.(m)$  เน้นระบุชั้นคำนวณ.

จากสมการ 6.11 จะได้

$$\delta_{fkl}^{(m)} = \frac{\partial E_n}{\partial z_{fkl}^{(m)}} \cdot h'(a_{fkl}^{(m)}) \quad (6.21)$$

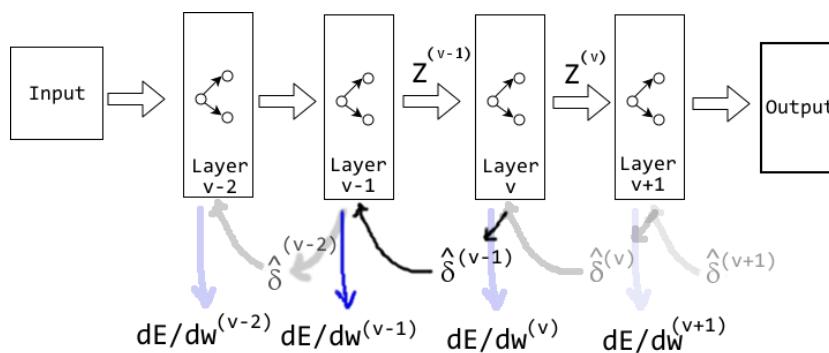
โครงข่ายคอนโวลูชันจะมีชั้นคำนวณ 3 ชั้นดิจิทัล ได้แก่ ชั้นคอนโวลูชัน, ชั้นดึงรวม, และชั้นเชื่อมต่อเต็มที่ โดยการใช้งาน บางครั้งอาจมีหรือไม่มีชั้นดึงรวมหรือชั้นเชื่อมต่อเต็มที่ก็ได้ นอกจากนั้น จำนวนชั้นทั้งหมดและการเรียงลำดับของชั้นชนิดต่าง ๆ ก็อาจแตกต่างกันไป. การคำนวณค่า  $\frac{\partial E_n}{\partial z_{fkl}^{(m)}}$  ของชั้น  $m^{th}$  จะขึ้นกับชั้นคำนวณถัดไป (หรือฟังก์ชันจุดประสีค์ หากชั้น  $m^{th}$  เป็นชั้นสุดท้าย). เพื่อความสะดวก กำหนด

$$\hat{\delta}_{fkl}^{(m)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(m)}}. \quad (6.22)$$

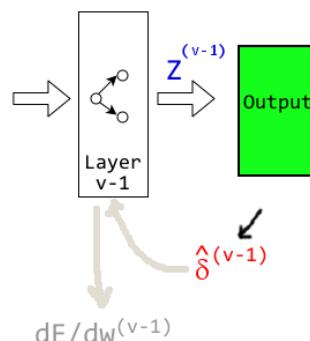
ดังนั้น เช่นเดียวกับโครงข่ายประสาทเทียมแบบเชื่อมต่อเต็มที่ การคำนวณการแพร่กระจายย้อนกลับของโครงข่ายคอนโวลูชัน แต่ละชั้นจะคำนวณค่า  $\hat{\delta}$  ออกม�다วย เพียงแต่ ค่า  $\hat{\delta}_{fkl}^{(m)}$  จะคำนวณมาจากชั้นที่  $(m+1)^{st}$  และชั้นที่  $m^{th}$  ก็จะต้องคำนวณค่า  $\hat{\delta}_{fkl}^{(m-1)}$  เพื่อให้ชั้น  $(m-1)^{st}$  สามารถนำไปคำนวณหาค่าเกรเดียนต์ของค่าน้ำหนักได้.

เนื่องจาก  $\hat{\delta}_{fkl}^{(m)}$  จะได้จากการคำนวณจากชั้นที่  $(m+1)^{st}$  จึงจะต้องกว่าที่ เมื่อพิจารณาชั้นคำนวณ  $n^{th}$  เราจะอภิปรายถึงการคำนวณหาค่า  $\hat{\delta}^{(v-1)}$  ซึ่งเน้นว่า ค่า  $\hat{\delta}^{(v-1)}$  นี้คำนวณจากชั้น  $n^{th}$  แต่นำไปใช้ในการหาค่าอนุพันธ์สำหรับชั้น  $(v-1)^{st}$ . รูป 6.10 แสดงภาพการผ่านค่า  $\hat{\delta}$  ที่คำนวณจากชั้น  $v^{th}$  เพื่อนำไปใช้คำนวณค่าเกรเดียนต์ของชั้น  $(v-1)^{st}$ .

เมื่อพิจารณาโครงข่ายคอนโวลูชัน ชนิดของชั้นคำนวณก่อนหน้าของชั้นคอนโวลูชันจะเป็น ชั้นคำนวณคอนโวลูชัน ชั้นดึงรวม หรืออินพุตได้เท่านั้น. โครงข่ายคอนโวลูชันไม่มีกรณีที่ชั้นเชื่อมต่อเต็มที่อยู่ก่อนหน้าชั้นคอนโวลูชัน เพราะชั้นเชื่อมต่อเต็มที่ไม่สามารถสร้างเชิงมิติของข้อมูล ดังนั้นหากจัดเรียงชั้นเชื่อมต่อเต็มก่อนชั้นคอนโวลูชัน จึงไม่สามารถใช้ประโยชน์จากการคอนโวลูชันกับโครงสร้างเชิงมิติของข้อมูลได้.



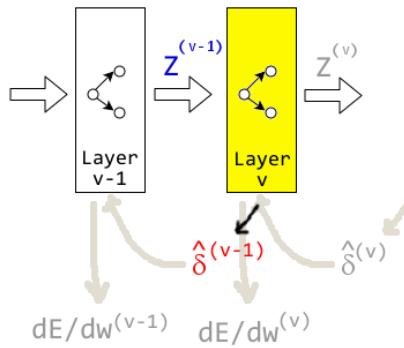
รูปที่ 6.10: แผนภาพการแพร่กระจายย้อนกลับของค่า  $\hat{\delta}$  ในโครงข่ายประสาทเทียม 4 ชั้น. จุดประสงค์หลักของการแพร่กระจายย้อนกลับ คือการคำนวณหาค่าเกรเดียนต์หรืออนุพันธ์ของฟังก์ชันจุดประสงค์เทียบกับค่าน้ำหนัก. นั่นคือ  $\frac{\partial E}{\partial w^{(v-2)}}, \frac{\partial E}{\partial w^{(v-1)}}, \frac{\partial E}{\partial w^{(v)}},$  และ  $\frac{\partial E}{\partial w^{(v+1)}}$ . การคำนวณค่าเกรเดียนต์ของน้ำหนักแต่ละชั้น ต้องอาศัยค่า  $\hat{\delta}$  ซึ่งการคำนวณค่าเกรเดียนต์ของน้ำหนักชั้น  $v-1$  ต้องอาศัยค่า  $\hat{\delta}^{(v-1)}$ . ค่า  $\hat{\delta}^{(v-1)}$  คำนวณมาจากชั้น  $v$ . ในทำนองเดียวกัน เพื่อคำนวณ  $\frac{\partial E}{\partial w^{(v)}}$  ชั้น  $v$  เองก็อาศัยค่า  $\hat{\delta}^{(v)}$  จากชั้น  $v+1$ . ชั้นสุดท้าย  $v+1$  ก็ต้องการค่า  $\hat{\delta}^{(v+1)}$  เพื่อคำนวณ  $\frac{\partial E}{\partial w^{(v+1)}}$  แต่ค่า  $\hat{\delta}^{(v+1)}$  สามารถคำนวณได้โดยตรงจากการแทนค่าและหาอนุพันธ์. ค่า  $\hat{\delta}$  ของชั้นสุดท้ายสามารถคำนวณได้โดยตรง เพราะ  $\hat{\delta}$  คืออนุพันธ์ของฟังก์ชันจุดประสงค์เทียบกับเอาร์พตของชั้น และเอาร์พตของชั้นสุดท้ายเชื่อมโยงกับฟังก์ชันจุดประสงค์โดยตรง จึงทำให้ค่า  $\hat{\delta}$  สามารถคำนวณได้โดยตรง. นอกจากนี้ สังเกตว่าชั้นคำนวณแรกสุด (ชั้น  $v-2$ ) ต้องการค่า  $\hat{\delta}^{(v-2)}$  จากชั้น  $v-1$ . แต่ชั้นคำนวณแรกสุด ไม่จำเป็นต้องผ่านค่า  $\hat{\delta}$  ออกมานะ. ชั้นคำนวณ  $v-2$  สามารถคำนวณค่า  $\hat{\delta}^{(v-3)}$  ออกมาได้ แต่เนื่องจากไม่มีชั้นคำนวณ  $v-3$  ที่ต้องค่า  $\hat{\delta}^{(v-3)}$  นี้ จึงทำให้ชั้น  $v-2$  ซึ่งเป็นชั้นคำนวณแรกสุด ไม่จำเป็นต้องผ่านค่า  $\hat{\delta}$  ออกมา.



รูปที่ 6.11: แผนภาพการแพร่กระจายย้อนกลับของค่า  $\hat{\delta}$  โดยชั้นเอาร์พตจะผ่านค่า  $\hat{\delta}$  จากชั้นเอาร์พตกลับไปให้ชั้น  $v-1$  เพื่อให้ชั้น  $v-1$  สามารถใช้คำนวณเกรเดียนต์ต่อน้ำหนักของชั้นได้. ค่า  $\hat{\delta}^{(v-1)} = \frac{\partial E}{\partial z^{(v-1)}}$  ซึ่งที่ชั้นเอาร์พต ค่านี้สามารถคำนวณได้ตรงไปตรงมา โดยการแทนค่าฟังก์ชันจุดประสงค์  $E$  ที่กำหนดในพจน์ของ  $z^{(v-1)}$  และหาอนุพันธ์.

ชนิดของชั้นคำนวณหลังจากชั้นคอนเวอลูชัน อาจเป็น ชั้นคอนเวอลูชัน ชั้นดึงรวม ชั้นเชื่อมต่อเติมที่ หรือเอาร์พตก็ได้.

**กรณีชั้นเอาร์พต** สำหรับกรณีชั้นคำนวณ  $(v-1)^{st}$  เป็นชั้นคำนวณสุดท้าย และชั้นคำนวณ  $(v-1)^{st}$  ต้องการค่า  $\hat{\delta}^{(v-1)} = \frac{\partial E_n}{\partial z^{(v-1)}}$  จากชั้นเอาร์พต. รูป 6.11 แสดงการผ่านค่า  $\hat{\delta}$  จากชั้นเอาร์พตกลับไปให้ชั้นคำนวณสุดท้าย.



รูปที่ 6.12: แผนภาพการแพร่กระจายย้อนกลับของค่า  $\hat{\delta}$  โดยชั้นคำนวณที่  $v$  ผ่านค่า  $\hat{\delta}^{(v-1)}$  กลับไปให้ชั้น  $v-1$  เพื่อให้ชั้น  $v-1$  สามารถใช้คำนวณเกรเดียนต์ต่อน้ำหนักของชั้นได้. ในขณะที่ชั้น  $v$  องค์รับ  $\hat{\delta}^{(v)}$  มาเพื่อคำนวณเกรเดียนต์ต่อน้ำหนักของชั้น.

ที่ชั้นาเอาร์พุต<sup>4</sup> ค่า  $\hat{\delta}^{(v-1)}$  ก็สามารถหาเกรเดียนต์ได้โดยตรง จากการแทนค่าฟังก์ชันจุดประสังค์ในพจน์ของตัวแปร  $z^{(v-1)}$  และการหาค่าอนุพันธ์.

ตัวอย่างเช่น ฟังก์ชันจุดประสังค์  $E_n = \lambda \sum_{q=1}^F \sum_{r=1}^{H'} \sum_{s=1}^{W'} (y_{qrs} - z_{qrs}^{(v-1)})^2$  เมื่อ  $\lambda$  เป็นค่าคงที่,  $F$  เป็นจำนวนชุดของเอาร์พุต,  $H'$  และ  $W'$  เป็นขนาดของเอาร์พุต,  $y_{qrs}$  เป็นเฉลย (ground-truth), และ  $z_{qrs}^{(v-1)}$  คือเอาร์พุตจากชั้น  $(v-1)^{st}$ .

ดังนั้น จากการแทนค่าฟังก์ชันจุดประสังค์ในพจน์ของเอาร์พุต

$$\begin{aligned}\hat{\delta}_{fkl}^{(v-1)} &= \frac{\partial \lambda \sum_{q=1}^F \sum_{r=1}^{H'} \sum_{s=1}^{W'} (y_{qrs} - z_{qrs}^{(v-1)})^2}{\partial z_{fkl}^{(v-1)}} \\ &= -2\lambda(y_{fkl} - z_{fkl}^{(v-1)}).\end{aligned}$$

ดูเนื้อหาโยโล่ ในหัวข้อ 7.1 เพิ่มเติมสำหรับตัวอย่างการใช้โครงข่ายคอนโวลูชัน โดยใช้ชั้นคอนโวลูชันเป็นชั้นคำนวณสุดท้าย.

แต่หากชั้นคอนโวลูชันไม่ได้เป็นชั้นคำนวณสุดท้าย ค่า  $\hat{\delta}^{(v-1)}$  จะได้จากการคำนวณจากชั้นที่  $v^{th}$  ตามชนิดของชั้น  $v^{th}$ . รูป 6.12 แสดงแผนภาพการผ่านค่า  $\hat{\delta}^{(v-1)}$  กลับไปให้ชั้นก่อนหน้า.

กรณีชั้นเชื่อมต่อเต็มที่ พิจารณาชั้นที่  $v^{th}$  เป็นชั้นเชื่อมต่อเต็มที่ ค่า  $\hat{\delta}_{fkl}^{(v-1)}$  คำนวณได้จาก เอาร์พุต  $z_{fkl}^{(v-1)}$  ของชั้น  $(v-1)^{st}$  เชื่อมต่อไปยังฟังก์ชันจุดประสังค์  $E_n$  ผ่านชั้น  $v^{th}$  โดย เอาร์พุตของชั้น  $(v-1)^{st}$  จะกล้ายเป็นอินพุตของชั้น  $v^{th}$ .

<sup>4</sup> การมองชั้นาเอาร์พุตเป็นชั้นเชิงตระกูลหลังจากชั้นคำนวณสุดท้าย ก็เพื่อให้ง่ายต่อการเรียบเรียงเนื้อหา. นั่นคือ ทุกกรณี จะเรียกเป็นชั้น  $v^{th}$  และจะแสดงวิธีการหา  $\hat{\delta}^{(v-1)}$  สำหรับชั้นก่อนหน้า.

เอาต์พุต  $z_{fkl}^{(v-1)}$  จากชั้นคอนโวโลชัน เมื่อเข้าเป็นอินพุตของชั้นเชื่อมต่อเต็มที่ จะถูกスタイルโครงสร้างลงเป็น  $z_q^{(v-1)}$  โดย  $z_{fkl} = z_q$  เมื่อ  $q = l + W' \cdot (k - 1) + H'W' \cdot (f - 1)$  สำหรับ  $f = 1, \dots, F; k = 1, \dots, H'; l = 1, \dots, W'$ .

ดังนี้

$$\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \frac{\partial E_n}{\partial z_q^{(v-1)}}. \quad (6.23)$$

ค่า  $\frac{\partial E_n}{\partial z_q^{(v-1)}}$  ที่สามารถหาได้ เช่นเดียวกับชั้นเชื่อมต่อเต็มที่ ซึ่งได้อธิบายในหัวข้อ 3.3.

ทบทวนเรื่องการเดียนต์ของชั้นเชื่อมต่อเต็มที่ (จากหัวข้อ 3.3) พิจารณาที่ชั้น  $v^{th}$  ซึ่งเป็นชั้นเชื่อมต่อเต็มที่. เอาต์พุตของชั้น  $(v-1)^{st}$  ส่งอิทธิพลกับฟังก์ชันจุดประสงค์  $E_n$  ผ่านชั้น  $v^{th}$  และ เมื่อใช้กฎลูกโซ่จะได้

$$\frac{\partial E_n}{\partial z_q^{(v-1)}} = \sum_{r=1}^R \frac{\partial E_n}{\partial a_r^{(v)}} \cdot \frac{\partial a_r^{(v)}}{\partial z_q^{(v-1)}}$$

สำหรับ  $q = 1, \dots, F \cdot H' \cdot W'$  เมื่อ  $a_r^{(v)}$  คือค่าการกระตุ้นของชั้นเชื่อมต่อเต็มที่ และ  $R$  คือจำนวนหน่วยช่องในชั้น  $v^{th}$ . จาก  $a_r^{(v)} = \sum_q w_{rq}^{(v)} z_q^{(v-1)}$  และ  $\delta_r^{(v)} \equiv \frac{\partial E_n}{\partial a_r^{(v)}}$  ทำให้ได้

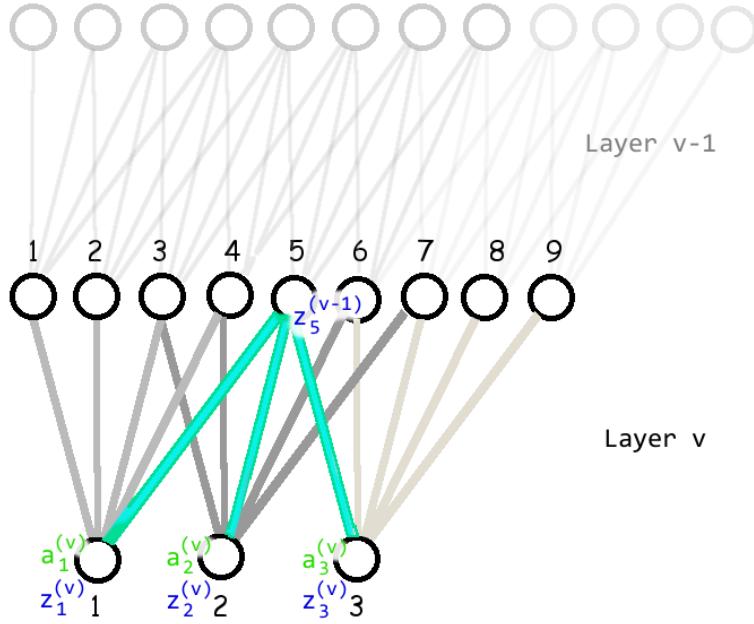
$$\frac{\partial E_n}{\partial z_q^{(v-1)}} = \sum_{r=1}^R \delta_r^{(v)} \cdot w_{rq}^{(v)}$$

และค่า  $\delta_r^{(v)}$  ที่สามารถหาได้จากการแพร่กระจายย้อยกลับ ดังอธิบายในหัวข้อ 3.3 (สมการ 3.29 สำหรับชั้นคำนวนสุดท้าย หรือ สมการ 3.31 สำหรับชั้นเชื่อมต่อเต็มที่ที่ไม่ใช่ชั้นคำนวนสุดท้าย)

กรณีชั้นคอนโวโลชัน ที่ชั้นคอนโวโลชัน  $v^{th}$  ค่า  $\hat{\delta}_{fkl}^{(v-1)}$  ที่สามารถหาได้ในลักษณะเดียวกับชั้นเชื่อมต่อเต็มที่ เพียงแต่หน่วยเอาต์พุตของชั้นคอนโวโลชันที่  $(v-1)^{st}$  เชื่อมต่อกับหน่วยในชั้น  $v^{th}$  เฉพาะหน่วยที่สัมพันธ์กัน และค่าน้ำหนักมีการใช้ร่วมกัน.

หน่วย  $z_{fkl}^{(v-1)}$  ส่งอิทธิพลไปถึงฟังก์ชันจุดประสงค์  $E_n$  ผ่านหน่วย  $a_{qrs}^{(v)}$  แต่หน่วย  $a_{qrs}^{(v)}$  เชื่อมต่อกับหน่วย  $z_{fkl}^{(v-1)}$  เฉพาะหน่วยที่สัมพันธ์กัน. รูป 6.13 แสดงการเชื่อมต่อของเอาต์พุตของชั้น  $(v-1)^{st}$  กับเอาต์พุตของชั้น  $v^{th}$ . ในรูปแสดงแผนภาพของชั้นคอนโวโลชันหนึ่งมิติ แต่การเชื่อมต่อชั้นคอนโวโลชันสองมิติ ก็ทำในลักษณะเดียวกัน.

เมื่อพิจารณาการเชื่อมต่อ จะพบว่า ตำแหน่งเอาต์พุตของชั้น  $v^{th}$  สัมพันธ์กับตำแหน่งเอาต์พุตของชั้น  $(v-1)^{st}$  โดย สำหรับแต่ละลักษณะสำคัญ  $q$ , หน่วย  $a_{qrs}^{(v)}$  เชื่อมต่อ  $z_{fkl}^{(v-1)}, f = 1, \dots, F$  เมื่อ  $F$  คือ



รูปที่ 6.13: แผนภาพการเชื่อมต่อ แสดงหนึ่งหน่วยชั้น  $v - 1$  มีอิทธิพลกับสามหน่วยชั้น  $v$  โดยชั้น  $v$  ใช้ก้าวย่างขนาด 2 และฟิลเตอร์ขนาด 5 และมีเพียงฟิลเตอร์เดียว. ตัวอย่างเน้นแสดงเอาต์พุตหน่วยที่ 5 (ชั้น  $v - 1$ ) เชื่อมต่อกับคอนโวลูชันเอาต์พุตหน่วยที่ 1, 2, 3 (ชั้น  $v$ ). นั่นคือ  $z_5^{(v-1)}$  เชื่อมต่อกับ  $a_1^{(v)}, a_2^{(v)}, a_3^{(v)}$ .

จำนวนลักษณะสำคัญของชั้น  $(v - 1)^{st}$ ,  $r \in \Omega(k, S_H, H_F)$ ,  $s \in \Omega(l, S_W, W_F)$ , ขนาดก้าวย่างกับขนาดฟิลเตอร์ของชั้น  $v^{th}$  คือ  $S_H \times S_W$  กับ  $H_F \times W_F$  ตามลำดับ, และ พังก์ชันเซต

$$\Omega(k, S, H_F) = \left\{ \frac{k-i}{S} + 1 : (k-i \geq 0) \text{ and } ((k-i) \bmod S = 0) \right\}_{i=1, \dots, H_F} \quad (6.24)$$

ตัวอย่าง เช่น หน่วยเอาต์พุต  $z_{fkl}^{(v-1)}$  สำหรับ  $f = 1, \dots, 4; k = 1, \dots, 5; l = 1, \dots, 5$ . นั่นคือ ชั้น  $(v - 1)^{st}$  มีสี่ลักษณะสำคัญ ที่เกิดจากการใช้ฟิลเตอร์สี่ตัว และได้ແນที่ลักษณะสำคัญแต่ละอันเป็นขนาด  $5 \times 5$ . เมื่อ หน่วยเอาต์พุตในชั้น  $(v - 1)^{st}$  เชื่อมต่อกับชั้น  $v^{th}$  ที่เป็นชั้นคอนโวลูชัน โดยชั้น  $v^{th}$  มีแปดลักษณะสำคัญ ( $q = 1, \dots, 8$ ) ใช้ฟิลเตอร์ขนาด  $3 \times 3$  สำหรับแต่ละลักษณะสำคัญ และใช้ขนาด ก้าวย่างเป็น  $1 \times 1$  แล้ว ดังนั้นหน่วย(คอนโวลูชันเอาต์พุต)ของชั้น  $(v - 1)^{st}$  เชื่อมต่อกับหน่วย(คอนโวลูชันเอาต์พุต)ของชั้น  $v^{th}$  ดังนี้

- หน่วย  $z_{1,1,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{1,1,1}^{(v)}, \dots, a_{8,1,1}^{(v)}$ .

จาก  $z_{1,1,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{qrs}^{(v)}$  สำหรับ ทุก ๆ  $q = 1, \dots, 8; r = 1; s = 1$  เพราะว่า  $z_{1,1,1}^{(v-1)}$  มี  $f = 1, k = 1, l = 1$ . ที่  $k = 1$  ทำให้  $r \in \Omega(k = 1, S = 1, H = 3)$ . เมื่อแทนค่าลงในสมการ 6.24 จะได้ว่า  $r \in \{\frac{1-i}{1} + 1 : (1-i \geq 0) \text{ and } ((1-i) \bmod 1 = 0)\}_{i=1, \dots, 3}$  และเมื่อพิจารณาสมาชิกกับ

เงื่อนไข. ที่  $i = 1$ , สมาชิก  $\frac{1-1}{1} + 1 = 1$ . ที่  $i = 2$ , ค่า  $1 - 2 < 0$  ไม่ผ่านเงื่อนไขแรก. ที่  $i = 3$ , ค่า  $1 - 3 < 0$  ไม่ผ่านเงื่อนไขแรก. ดังนั้นที่  $k = 1$  ทำให้  $r \in \{1\}$  และในทำนองเดียวกัน ที่  $l = 1$  ทำให้  $s \in \{1\}$ .

- หน่วย  $z_{1,2,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{1,1,1}^{(v)}, \dots, a_{8,1,1}^{(v)}, a_{1,2,1}^{(v)}, \dots, a_{8,2,1}^{(v)}$ .

จาก  $z_{1,1,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{qrs}^{(v)}$  สำหรับ ทุก ๆ  $q = 1, \dots, 8; r = 1, 2; s = 1$  เพราะว่า  $k = 2$  ทำให้  $r \in \{1, 2\}$  จาก  $r \in \{\frac{2-i}{1} + 1 : (2-i \geq 0) \text{ and } ((2-i) \bmod 1 = 0)\}_{i=1, \dots, 3}$ . ที่  $i = 1$ , สมาชิก  $\frac{2-1}{1} + 1 = 2$ . ที่  $i = 2$ , สมาชิก  $\frac{2-2}{1} + 1 = 1$ . ที่  $i = 3$ , ค่า  $2 - 3 < 0$  ไม่ผ่านเงื่อนไขแรก.

- หน่วย  $z_{1,3,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{1,1,1}^{(v)}, \dots, a_{8,1,1}^{(v)}, a_{1,2,1}^{(v)}, \dots, a_{8,2,1}^{(v)}, a_{1,3,1}^{(v)}, \dots, a_{8,3,1}^{(v)}$ .

กรณี  $k = 3$  ทำให้  $r \in \{1, 2, 3\}$

- หน่วย  $z_{1,4,1}^{(v-1)}$  เชื่อมต่อกับ  $a_{1,2,1}^{(v)}, \dots, a_{8,2,1}^{(v)}, a_{1,3,1}^{(v)}, \dots, a_{8,3,1}^{(v)}, a_{1,4,1}^{(v)}, \dots, a_{8,4,1}^{(v)}$ .

กรณี  $k = 4$  ทำให้  $r \in \{2, 3, 4\}$

$\vdots$

- หน่วย  $z_{1,5,5}^{(v-1)}$  เชื่อมต่อกับ

$a_{1,3,3}^{(v)}, \dots, a_{8,3,3}^{(v)}, a_{1,4,3}^{(v)}, \dots, a_{8,4,3}^{(v)}, a_{1,5,3}^{(v)}, \dots, a_{8,5,3}^{(v)},$   
 $a_{1,3,4}^{(v)}, \dots, a_{8,3,4}^{(v)}, a_{1,4,4}^{(v)}, \dots, a_{8,4,4}^{(v)}, a_{1,5,4}^{(v)}, \dots, a_{8,5,4}^{(v)},$   
 $a_{1,3,5}^{(v)}, \dots, a_{8,3,5}^{(v)}, a_{1,4,5}^{(v)}, \dots, a_{8,4,5}^{(v)}, a_{1,5,5}^{(v)}, \dots, a_{8,5,5}^{(v)}$ .

กรณี  $k = 5$  ทำให้  $r \in \{3, 4, 5\}$ .  $l = 5$  ทำให้  $s \in \{3, 4, 5\}$ .

$\vdots$

- หน่วย  $z_{4,5,5}^{(v-1)}$  เชื่อมต่อกับ

$a_{1,3,3}^{(v)}, \dots, a_{8,3,3}^{(v)}, a_{1,4,3}^{(v)}, \dots, a_{8,4,3}^{(v)}, a_{1,5,3}^{(v)}, \dots, a_{8,5,3}^{(v)},$   
 $a_{1,3,4}^{(v)}, \dots, a_{8,3,4}^{(v)}, a_{1,4,4}^{(v)}, \dots, a_{8,4,4}^{(v)}, a_{1,5,4}^{(v)}, \dots, a_{8,5,4}^{(v)},$   
 $a_{1,3,5}^{(v)}, \dots, a_{8,3,5}^{(v)}, a_{1,4,5}^{(v)}, \dots, a_{8,4,5}^{(v)}, a_{1,5,5}^{(v)}, \dots, a_{8,5,5}^{(v)}$ .

กรณี  $k = 5$  ทำให้  $r \in \{2, 3, 4\}$  และ  $l = 5$  ทำให้  $s \in \{2, 3, 4\}$

เป็นต้น.

สังเกต (1) ด้วยนีเชิงลำดับ  $k$  และ  $l$  ของชั้น  $(v-1)^{st}$  จะบอกด้วยนีเชิงลำดับ  $r$  และ  $s$  สำหรับทุก ๆ ด้วยนีอิสระ  $q$  ของหน่วยในชั้น  $v^{th}$ . ตาราง 6.1 แสดงความสัมพันธ์ตำแหน่งหน่วยที่  $k$  ในชั้น  $(v-1)^{st}$  กับตำแหน่งหน่วยที่  $r$  ในชั้น  $v^{th}$ . ด้วย  $l$  ก็เป็นในลักษณะเดียวกัน. (2) หน่วยที่มีตำแหน่งเชิงลำดับเดียวกัน แต่

ต่างตำแหน่งอิสระ เช่น  $z_{1,5,5}$  กับ  $z_{4,5,5}$  เชื่อมต่อ กับ หน่วยเดียวกัน.

ตารางที่ 6.1: ตัวอย่างแสดงความสัมพันธ์ของหน่วยในชั้น  $v - 1$  กับดัชนีของหน่วยในชั้น  $v$  ที่เชื่อมต่อกัน. ตำแหน่งหน่วยที่  $k$  (ในชั้น  $v - 1$ ) เชื่อมต่อกับตำแหน่งหน่วยที่  $r$  (ในชั้น  $v$ ) เมื่อชั้น  $v$  ใช้ก้าวย่าง และพิลเตอร์ขนาดต่าง ๆ. วงเล็บใต้เขตค่าของ  $r$  สรุปย่อ เหตุผลของค่าสมาชิกที่ได้และไม่ได้ เช่น  $(-, 1, x)$  ที่  $k = 2, S_H = 2, H_F = 3$  หมายถึง ที่  $i = 1$  ไม่ผ่านเงื่อนไขที่สอง  $((2 - 1) \bmod 2 = 1)$ , ที่  $i = 2$  ได้ค่าสมาชิกเป็น 1, และ ที่  $i = 3$  ไม่ผ่านเงื่อนไขที่หนึ่ง  $(2 - 3 < 0)$  เป็นต้น

ก้าวย่าง	$S_H = 1$		$S_H = 2$	
พิลเตอร์	$H_F = 3$	$H_F = 5$	$H_F = 3$	$H_F = 5$
$k = 1$	$r \in \{1\}$ $(1, x, x)$	$r \in \{1\}$ $(1, x, x, x, x)$	$r \in \{1\}$ $(1, x, x)$	$r \in \{1\}$ $(1, x, x, x, x)$
$k = 2$	$r \in \{1, 2\}$ $(2, 1, x)$	$r \in \{1, 2\}$ $(2, 1, x, x, x)$	$r \in \{1\}$ $(-, 1, x)$	$r \in \{1\}$ $(-, 1, x, x, x)$
$k = 3$	$r \in \{1, 2, 3\}$ $(3, 2, 1)$	$r \in \{1, 2, 3\}$ $(3, 2, 1, x, x)$	$r \in \{1, 2\}$ $(2, -, 1)$	$r \in \{1, 2\}$ $(2, -, 1, x, x)$
$k = 4$	$r \in \{2, 3, 4\}$ $(4, 3, 2)$	$r \in \{1, 2, 3, 4\}$ $(4, 3, 2, 1, x)$	$r \in \{2\}$ $(-, 2, -)$	$r \in \{1, 2\}$ $(-, 2, -, 1, x)$
$k = 5$	$r \in \{3, 4, 5\}$ $(5, 4, 3)$	$r \in \{1, 2, 3, 4, 5\}$ $(5, 4, 3, 2, 1)$	$r \in \{2, 3\}$ $(3, -, 2)$	$r \in \{1, 2, 3\}$ $(3, -, 2, -, 1)$

จากการเชื่อมต่อข้างต้น เมื่อพิจารณาชั้นคอนโวลูชันที่  $v^{th}$ , หน่วย  $z_{fkl}^{(v-1)}$  เชื่อมต่อไปสู่เอาร์พุตสุดท้าย และค่าฟังก์ชันจุดประสงค์ ผ่าน  $a_{qrs}^{(v)}$  และจากกฎลูกโซ่ จะได้

$$\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \sum_{q=1}^Q \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \frac{\partial E_n}{\partial a_{qrs}^{(v)}} \frac{\partial a_{qrs}^{(v)}}{\partial z_{fkl}^{(v-1)}} \quad (6.25)$$

เมื่อ  $Q$  คือจำนวนลักษณะสำคัญในชั้น  $v^{th}$ ,  $\Omega_r = \Omega(k, S_H, H_F)$  และ  $\Omega_s = \Omega(l, S_W, W_F)$ , โดย  $H_F \times W_F$  และ  $S_H \times S_W$  เป็นขนาดพิลเตอร์และขนาดก้าวย่างในชั้น  $v^{th}$  ตามลำดับ.

จากนิยาม 6.18 และการคำนวณคอนโวลูชันสมการ 6.10 (อินพุต  $\hat{\mathbf{X}}$  ในสมการ 6.10 คือ อินพุตของชั้น  $v^{th}$  ซึ่งคือเอาร์พุตของชั้นก่อนหน้า, นั่นคือ  $\hat{\mathbf{X}} \equiv \mathbf{Z}^{(v-1)}$  และ เพื่อให้ตัวแปรดัชนีพิลเตอร์จำแนกได้ง่าย ที่นี่ใช้ตัวชี้  $\hat{f}$  แทนตัวชี้นี่พิลเตอร์ สำหรับการคำนวณคอนโวลูชัน), เมื่อแทนค่าลงในสมการ 6.25 จะได้ว่า

$$\begin{aligned} \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} &= \sum_q \sum_r \sum_s \delta_{qrs}^{(v)} \frac{\partial \left( b_{q,r,s}^{(v)} + \sum_{\hat{f}} \sum_i \sum_j w_{qfi j}^{(v)} z_{\hat{f}, S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)} \right)}{\partial z_{fkl}^{(v-1)}} \\ &= \sum_q \sum_r \sum_s \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1), l-S_W \cdot (s-1)}^{(v)}. \end{aligned} \quad (6.26)$$

สังเกตว่า เพื่อคำนวนหา  $\hat{\delta}_{fkl}^{(v-1)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$  สำหรับส่งไปให้ชั้น  $(v-1)^{st}$ , ชั้น  $v^{th}$  ต้องคำนวน  $\delta_{qrs}^{(v)}$  และ จากสมการ 6.21 ค่า  $\delta_{qrs}^{(v)} = \frac{\partial E_n}{\partial z_{qrs}^{(v)}} \cdot h' \left( a_{qrs}^{(v)} \right)$  เมื่อ  $h'(\cdot)$  คืออนุพันธ์ของฟังก์ชันกระตุ้นชั้น  $v^{th}$ . ส่วนค่า  $\frac{\partial E_n}{\partial z_{qrs}^{(v)}} \equiv \hat{\delta}_{qrs}^{(v)}$  ก็ได้มามาจากชั้น  $(v+1)^{st}$  อีกทอดหนึ่ง. หัวข้อ 6.4 สรุปย่อขั้นตอนการคำนวนเกรเดียนต์สำหรับโครงข่ายคอนโวลูชัน และรายการ 6.2 แสดงตัวอย่างโปรแกรมโครงข่ายคอนโวลูชัน.

ขนาดก้าวย่างที่นิยม พิจารณา  $\hat{\delta}_{fkl}^{(v-1)}$  สำหรับกรณีชั้น  $v^{th}$  ใช้ก้าวย่างขนาด  $1 \times 1$  ซึ่งเป็นขนาดที่มักถูกใช้กับชั้นคอนโวลูชัน. กรณีจะทำให้

$$\Omega_r = \{k-i+1 : k-i \geq 0\}_{i=1,\dots,H_F}, \quad (6.27)$$

$$\Omega_s = \{l-i+1 : l-i \geq 0\}_{i=1,\dots,W_F}. \quad (6.28)$$

และ

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{q=1}^Q \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \delta_{qrs}^{(v)} \cdot w_{q,f,k-r+1,l-s+1} \quad (6.29)$$

เมื่อแทนสมการ 6.27 และ 6.28 ลงในสมการข้างต้น และเขียน  $r$  และ  $s$  ในรูป  $i$  และ  $j$  จะได้

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{q=1}^Q \sum_{i \in \{1,\dots,H_F : i \leq k\}} \sum_{j \in \{1,\dots,W_F : j \leq l\}} \delta_{q,k-i+1,l-j+1}^{(v)} \cdot w_{q,f,i,j}$$

หรือ

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{q=1}^Q \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} \delta_{q,k-i+1,l-j+1}^{(v)} \cdot w_{q,f,i,j} \quad (6.30)$$

สำหรับ  $k \geq i$  และ  $l \geq j$ .

กรณีชั้นดึงรวม หากชั้น  $v^{th}$  เป็นชั้นดึงรวม หน่วย  $z_{fkl}^{(v-1)}$  เชื่อมต่อไปสู่เอาเตอร์พุตสุดท้ายและค่าฟังก์ชันจุดประสงค์ ผ่านหน่วย  $a_{frs}^{(v)}$ . สังเกตว่า เนื่องจากชั้นดึงรวมไม่ได้ประมวลผลในชุดมิติของมิติอิสระ ดังนั้น หน่วยของชั้นดึงรวมในลักษณะสำคัญ  $f$  เกี่ยวข้องกับหน่วยก่อนหน้าในลักษณะสำคัญ  $f$  เช่นกันเท่านั้น. ชั้นดึงรวมรักษาชุดมิติอิสระไว้ (ขนาดชุดมิติอิสระของอินพุตและเอาเตอร์พุตเท่ากัน).

พิจารณาการคำนวนเอาเตอร์พุตของชั้นดึงรวม (สมการ 6.13)

$$z_{frs}^{(v)} = g(\{z_{f,S_H \cdot (r-1)+i, S_W \cdot (s-1)+j}^{(v-1)}\}_{i=1,\dots,H_F, j=1,\dots,W_F}) \quad (6.31)$$

เมื่อ  $\{z_{f,S_H \cdot (r-1)+i, S_W \cdot (s-1)+j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F}$  เป็นเซตของหน่วยต่าง ๆ ที่ถูกดึงมาร่วมกัน,  $g(\cdot)$  เป็นฟังก์ชันดึงรวม, และ  $z_{frs}^{(v)}$  คือเอาต์พุตของชั้น  $v^{th}$  ซึ่งเป็นชั้นดึงรวม โดย  $S_H \times S_W$  กับ  $H_F \times W_F$  คือขนาดก้าวย่างกับขนาดพิลเตอร์ ของชั้น  $v^{th}$  ตามลำดับ.

สังเกตความสัมพันธ์ระหว่าง  $z_{fkl}^{(v-1)}$  กับ  $z_{frs}^{(v)}$  ในชั้นดึงรวม จะคล้ายกับความสัมพันธ์ในชั้นคอนโวลูชัน โดยต่างกันที่ (1) ชั้นดึงรวมไม่มีการคำนวณค่ากราฟตุน  $a$  และ (2) ชั้นดึงรวมไม่ได้ประมวลผลในชุดมิติของมิติอิสระ. แต่หากมองเฉพาะความสัมพันธ์ในชุดมิติใช้ลำดับ จะพบว่าหน่วยในชั้นดึงรวมมีความสัมพันธ์กับหน่วยในชั้นติดกัน ในลักษณะเดียวกับชั้นคอนโวลูชัน (เปรียบเทียบด้วยชุดมิติใช้ลำดับ สมการ 6.13 กับ สมการ 6.10) ดังนั้น เมื่อยิงความสัมพันธ์ย้อนกลับ ชั้นดึงรวมก็สามารถใช้ฟังก์ชัน  $\Omega(\cdot)$  ในสมการ 6.24 ช่วยอธิบายความสัมพันธ์ได้ในลักษณะเดียวกัน.

เมื่อพิจารณา  $\hat{\delta}_{fkl}^{(v-1)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$  และจากกฎลูกโซ่ จะได้

$$\frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \frac{\partial E_n}{\partial z_{frs}^{(v)}} \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} \quad (6.32)$$

เมื่อ  $\Omega_r = \Omega(k, S_H, H_F)$  และ  $\Omega_s = \Omega(l, S_W, W_F)$ .

เมื่อแทน  $\frac{\partial E_n}{\partial z_{frs}^{(v)}}$  ด้วย  $\hat{\delta}_{frs}^{(v)}$  ลงไปจะได้

$$\frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \hat{\delta}_{frs}^{(v)} \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}. \quad (6.33)$$

ชั้น  $v^{th}$  รับค่า  $\hat{\delta}_{frs}^{(v)}$  มาจากชั้น  $(v+1)^{st}$ . ส่วนค่า  $\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$  สามารถคำนวณได้ในชั้น  $v^{th}$  นี้ โดยการหาอนุพันธ์ ดังนี้

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{\partial g(\{z_{f,S_H \cdot (r-1)+i, S_W \cdot (s-1)+j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F})}{\partial z_{fkl}^{(v-1)}} \quad (6.34)$$

ซึ่งผลการหาอนุพันธ์จะขึ้นกับฟังก์ชันดึงรวม. พิจารณาฟังก์ชันดึงรวม 3 แบบ ได้แก่ แบบมากที่สุด (max pooling), แบบเฉลี่ย (average pooling), และแบบอาร์เรียมแอล (root-mean-squared pooling หรือ rms pooling).

เมื่อใช้การดึงรวมแบบมากที่สุด นั่นคือ พังก์ชันดึงรวม  $g(\{z_1, \dots, z_n\}) = \max\{z_1, \dots, z_n\}$  และจะได้ว่า

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \begin{cases} 1 & \text{เมื่อ } z_{fkl}^{(v-1)} = \max\{\{z_{f,S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F}\}, \\ 0 & \text{เมื่อ } z_{fkl}^{(v-1)} \neq \max\{\{z_{f,S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F}\}. \end{cases} \quad (6.35)$$

เมื่อใช้การดึงรวมแบบเฉลี่ย นั่นคือ พังก์ชันดึงรวม  $g(\{z_1, \dots, z_n\}) = \frac{1}{n} \sum_{i=1}^n z_i$  และจะได้ว่า

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{1}{H_F W_F}. \quad (6.36)$$

เมื่อใช้การดึงรวมแบบอาร์เรียมแอล นั่นคือ พังก์ชันดึงรวม  $g(\{z_1, \dots, z_n\}) = \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}$  และจะได้ว่า

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{z_{fkl}^{(v-1)}}{H_F W_F \cdot z_{frs}^{(v)}} \quad (6.37)$$

ชันดึงรวมเองไม่มีค่าน้ำหนักที่ต้องปรับ จึงไม่ต้องคำนวนเกรเดียนต์เพียงบัน้ำหนักของชัน แต่ชันดึงรวมต้องผ่านค่า  $\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$  ไปให้ชัน  $(v-1)^{st}$ .

## 6.4 สรุปการคำนวนของโครงข่ายคอนโวโลชันสองมิติ

กำหนดให้อินพุตที่ผ่านการเติมเต็ม  $\mathbf{Z}^{(0)} \in \mathbb{R}^{C \times H' \times W'}$ . คำนวนการแพร่กระจายไปข้างหน้า ตามชนิดของชันคำนวน สำหรับชัน  $m = 1, \dots, M$  เมื่อ  $M$  คือจำนวนชันคำนวนทั้งหมด ดังนั้นเอาต์พุตของโครงข่ายคือเอาต์พุตของชันสุดท้าย. นั่นคือ  $\mathbf{Z}^{(M)}$  เป็นเอาต์พุตสุดท้าย.

กรณีชันคอนโวโลชัน คำนวนสมการ 6.10 และ 6.11 นั่นคือ

$$\begin{aligned} a_{f,k,l}^{(m)} &= b_f^{(m)} + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{fcij}^{(m)} \cdot z_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}^{(m-1)} \\ z_{f,k,l}^{(m)} &= h^{(m)}(a_{f,k,l}^{(m)}) \end{aligned}$$

สำหรับ  $f = 1, \dots, F$ ,  $k = 1, \dots, H$  และ  $l = 1, \dots, W$  เมื่อ  $b_f^{(m)}, w_{fcij}^{(m)}$  คือ ค่าไบอัส และค่าน้ำหนักของชั้น  $m^{th}$ ,  $h^{(m)}$  คือพังก์ชันกระตุ้นของชั้น  $m^{th}$ , โดย  $F, H, W$  คือจำนวนลักษณะสำคัญและขนาดของแผนที่เอาร์พุตของชั้น  $m^{th}$ ,  $H_F, W_F$  คือขนาดพิลเตอร์ของชั้น  $m^{th}$  ตามแนวตั้งและแนวนอนตามลำดับ และ  $S_H, S_W$  คือขนาดก้าวย่างของชั้น  $m^{th}$  ตามแนวตั้งและแนวนอนตามลำดับ.

**กรณีชั้นดึงรวม** คำนวณสมการ 6.13. นั่นคือ

$$z_{f,k,l}^{(m)} = g^{(m)}(\{z_{f,S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}^{(m-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F})$$

สำหรับ  $f = 1, \dots, F$ ,  $k = 1, \dots, H$  และ  $l = 1, \dots, W$  เมื่อ  $g^{(m)}$  คือพังก์ชันดึงรวมของชั้น  $m^{th}$ , โดย  $F, H, W$  คือจำนวนลักษณะสำคัญและขนาดของแผนที่เอาร์พุตของชั้น  $m^{th}$ ,  $H_F, W_F$  คือขนาดพิลเตอร์ของชั้น  $m^{th}$  ตามแนวตั้งและแนวนอนตามลำดับ และ  $S_H, S_W$  คือขนาดก้าวย่างของชั้น  $m^{th}$  ตามแนวตั้งและแนวนอนตามลำดับ.

**กรณีชั้นเชื่อมต่อเต็มที่** สลายโครงสร้างอินพุต (ถ้าจำเป็น). นั่นคือคำนวณ

$$z_q^{(m-1)} = z_{fkl}^{(m-1)}$$

เมื่อ  $q = l + W' \cdot (k-1) + H'W' \cdot (f-1)$  สำหรับ  $f = 1, \dots, F'; k = 1, \dots, H'; l = 1, \dots, W'$  โดย  $F', H', W'$  คือจำนวนลักษณะสำคัญ, ขนาดเอาร์พุตแนวตั้ง, ขนาดเอาร์พุตแนวโนนของชั้น  $(m-1)^{st}$  ตามลำดับ.

คำนวณชั้นเชื่อมต่อเต็มที่ สมการ 3.16 และ 3.17. นั่นคือ

$$\begin{aligned} a_f^{(m)} &= b_f^{(m)} + \sum_{q=1}^Q w_{fq}^{(m)} z_q^{(m-1)} \\ z_f^{(m)} &= h^{(m)}(a_f^{(m)}) \end{aligned}$$

สำหรับ  $f = 1, \dots, F$  เมื่อ  $b_f^{(m)}, w_{fq}^{(m)}, h^{(m)}$  คือค่าไบอัส ค่าน้ำหนัก และพังก์ชันกระตุ้นของชั้น  $m^{th}$  โดย  $F$  เป็นจำนวนหน่วยเอาร์พุตในชั้น  $m^{th}$ .

## เกรเดียนต์ของโครงข่ายคอนโวลูชัน

เกรเดียนต์ของโครงข่ายคอนโวลูชันสามารถหาได้ โดยใช้การแพร่กระจายย้อนกลับ โดยคำนวณทีละชั้นคำนวณ เริ่มจากเอาร์พุต แล้วไปชั้นสุดท้าย แล้วไล่ย้อนกลับไปทีละชั้นจนครบทุกชั้น. นั่นคือ หากชั้นคำนวณมี

จำนวน  $M$  ชั้น วิธีแพร่กระจายย้อนกลับจะเริ่มที่เอาร์พุต (นับเป็น ชั้น  $(M+1)^{st}$ ) และแล้วไอล์ย้อนกลับไปจนถึงชั้นแรก (ชั้น  $1^{st}$ ).

ค่าที่แต่ละชั้น  $\nu^{th}$  ต้องส่งย้อนกลับไปให้ชั้นก่อนหน้า คือ ค่า  $\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$  สำหรับ  $f = 1, \dots, F'$ ;  $k = 1, \dots, H'$ ;  $l = 1, \dots, W'$  เมื่อ  $F', H', W'$  คือ จำนวนลักษณะสำคัญ ขนาดเอาร์พุตในแนวตั้ง ขนาดเอาร์พุตในแนวนอน ของชั้น  $(v-1)^{st}$  ตามลำดับ.

**กรณีเอาร์พุต** ในที่นี้ หมายถึง การคำนวณแรกสุด (ก่อนการคำนวณชั้นสุดท้าย). ที่กรณีชั้น  $\nu^{th}$  เป็นเอาร์พุต ( $v = M+1$ ) เอาร์พุตไม่มีค่าน้ำหนักที่ต้องปรับ แต่ต้องการคำนวณ  $\hat{\delta}^{(v-1)} = \hat{\delta}^{(M)}$  เพื่อส่งกลับให้ชั้นคำนวณสุดท้าย.

ค่า  $\hat{\delta}_{fkl}^{(M)}$  สามารถหาได้จาก

$$\hat{\delta}_{fkl}^{(M)} = \frac{\partial E_n}{\partial z_{fkl}^{(M)}}$$

ซึ่งมักหาได้ไม่ยาก เนื่องจากฟังก์ชันจุดประสงค์  $E_n$  มักถูกนิยามในพจน์ของ  $z_{fkl}^{(M)}$ .

กรณีชั้นเชื่อมต่อเต็มที่ ที่กรณีชั้น  $\nu^{th}$  เป็นชั้นเชื่อมต่อเต็มที่ ชั้น  $\nu^{th}$  รับ  $\hat{\delta}_j^{(v)}$  มาจากชั้น  $(v+1)^{st}$  และชั้น  $\nu^{th}$  คำนวณเกรเดียนต์เทียบกับน้ำหนักของชั้นจากสมการ 3.28. นั่นคือ

$$\begin{aligned}\frac{\partial E_n}{\partial w_{jq}^{(v)}} &= \delta_j^{(v)} z_q^{(v-1)} \\ \frac{\partial E_n}{\partial b_j^{(v)}} &= \delta_j^{(v)}\end{aligned}$$

เมื่อ  $\delta_j^{(v)} = \hat{\delta}_j^{(v)} \cdot h'(a_j^{(v)})$  โดย  $h'(\cdot)$  เป็นอนุพันธ์ของฟังก์ชันกระตุ้นในชั้น  $\nu^{th}$ .

ชั้น  $\nu^{th}$  คำนวณ  $\hat{\delta}_q^{(v-1)}$  เพื่อส่งย้อนไปให้ชั้น  $(v-1)^{st}$  จากสมการ 3.31. นั่นคือ

$$\hat{\delta}_q^{(v-1)} = \sum_j w_{jq}^{(v)} \delta_j^{(v)}$$

สำหรับ  $q = 1, \dots, Q$  เมื่อ  $Q$  เป็นจำนวนเอาร์พุตทั้งหมดของชั้น  $(v-1)^{st}$ .

หากชั้น  $(v-1)^{st}$  เป็นชั้นคอนโวโลชันหรือชั้นดึงรวม ต้องทำการรื้อฟื้นโครงสร้างกลับมาใหม่ นั่นคือ

$$\hat{\delta}_{fkl}^{(v-1)} = \hat{\delta}_q^{(v-1)}$$

เมื่อ  $q = l + W' \cdot (k - 1) + H'W' \cdot (f - 1)$  สำหรับ  $q = 1, \dots, F' \cdot H' \cdot W'$  โดย  $F', H', W'$  คือจำนวนลักษณะสำคัญ, ขนาดເອາ໌ພຸດແນວຕັ້ງ, ขนาดເອາ໌ພຸດແນວນອນຂອງຫັນ  $(v - 1)^{st}$  ตามลำดับ.

กรณีຫັນດີງรวม ทີ່กรณีຫັນ  $v^{th}$  เป็นຫັນດີງรวม ຫັນ  $v^{th}$  ຮັບ  $\hat{\delta}_{frs}^{(v)}$  ມາຈາກຫັນ  $(v + 1)^{st}$ . ຫັນດີງรวมໄມ່ມີຄ່ານໍ້າໜັກ ໄມ່ຈະເປັນຕ້ອງคำນວນເກຣເດີຍນີ້ ແຕ່ຫັນດີງรวมຕ້ອງສ່ງ  $\hat{\delta}_{fkl}^{(v-1)}$  ໄປໃຫ້ຫັນ  $(v - 1)^{st}$ . ດ້ວຍ  $\hat{\delta}_{fkl}^{(v-1)}$  คำນວນໄດ້ຈາກສາມາດ [6.33](#). ນັ້ນຄືອ

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \hat{\delta}_{frs}^{(v)} \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$$

ສໍາຮັບ  $f = 1, \dots, F'; k = 1, \dots, H'$  และ  $l = 1, \dots, W'$  ເມື່ອ  $\Omega_r = \Omega(k, S_H, H_F); \Omega_s = \Omega(l, S_W, W_F)$  ແລະ  $F', H', W'$  คືອງຈຳນວນລักษณะสำคัญ, ขนาดເອາ໌ພຸດແນວຕັ້ງ, ขนาดເອາ໌ພຸດແນວນອນຂອງຫັນ  $(v - 1)^{st}$  ตามลำดับ.

ພັກຫັນເຊືດຕຳມາດໄດ້ຈາກສາມາດ [6.24](#),

$$\Omega(k, S, H) = \left\{ \frac{k - i}{S} + 1 : (k - i \geq 0) \text{ and } ((k - i) \bmod S = 0) \right\}_{i=1, \dots, H}.$$

ດ້ວຍ  $\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$  ຂຶ້ນກັບໜິດກາຣີງດີງรวม. ກາຣີງດີງรวมແບບນາກທີ່ສຸດ ໃຊ້ສາມາດ [6.35](#),

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \begin{cases} 1 & \text{ເມື່ອ } z_{fkl}^{(v-1)} = \max(\{z_{f, S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F}), \\ 0 & \text{ອື່ນ ຖ.} \end{cases}$$

ກາຣີງດີງรวมແບບເຂົ້າໝູ້ໃຊ້ສາມາດ [6.36](#),

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{1}{H_F W_F}$$

ເມື່ອ  $H_F \times W_F$  ເປັນຂາດຂອງພິລເຕອຮ້ອ້ຫັນທີ່  $v^{th}$ .

ກາຣີງດີງรวมແບບອາຣເອີມເອສ ໃຊ້ສາມາດ [6.37](#)

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{z_{fkl}^{(v-1)}}{H_F W_F \cdot z_{frs}^{(v)}}$$

ເມື່ອ  $H_F \times W_F$  ເປັນຂາດຂອງພິລເຕອຮ້ອ້ຫັນທີ່  $v^{th}$ .

กรณีชั้นคอนโวโลชัน ที่กรณีชั้น  $v^{th}$  เป็นชั้นคอนโวโลชัน ชั้น  $v^{th}$  รูป  $\hat{\delta}_{qrs}^{(v)}$  มาจากชั้น  $(v+1)^{st}$  และชั้น  $v^{th}$  คำนวนเกรเดียนต์เทียบกับน้ำหนักของชั้นจากสมการ 6.19 และสมการ 6.20. นั่นคือ

$$\begin{aligned}\frac{\partial E_n}{\partial w_{qfij}^{(v)}} &= \sum_{r=1}^H \sum_{s=1}^W \delta_{qrs}^{(v)} z_{f, S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)} \\ \frac{\partial E_n}{\partial b_q^{(v)}} &= \sum_{r=1}^H \sum_{s=1}^W \delta_{qrs}^{(v)}\end{aligned}$$

สำหรับ  $q = 1, \dots, F; f = 1, \dots, F'; i = 1, \dots, H_F$  และ  $j = 1, \dots, W_F$  เมื่อ  $F, H_F, W_F$  เป็นจำนวนลักษณะสำคัญ ขนาดพิลเตอร์แนวตั้ง ขนาดพิลเตอร์แนวโนนของชั้น  $v^{th}$  ตามลำดับ,  $F'$  เป็นจำนวนลักษณะสำคัญของชั้น  $(v-1)^{st}$ , ค่า  $z_{f, S_H \cdot (r-1) + i, S_W \cdot (s-1) + j}^{(v-1)}$  คือ อินพุตของชั้น  $v^{th}$  ที่ผ่านการเติมเต็มแล้ว,  $S_H$  และ  $S_W$  เป็นค่าก้าวย่างตามแนวตั้งและโนนของชั้น  $(v-1)^{st}$  และ  $\delta_{qrs}^{(v)} = \hat{\delta}_{qrs}^{(v)} \cdot h'(a_{qrs}^{(v)})$  โดย  $h'(a_{qrs}^{(v)})$  เป็นอนุพันธ์การกระตุ้นของชั้น  $v^{th}$ .

ชั้น  $v^{th}$  คำนวน  $\hat{\delta}_{fkl}^{(v-1)}$  เพื่อส่งย้อนไปให้ชั้น  $(v-1)^{st}$  จากสมการ 6.26. นั่นคือ

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{q=1}^F \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \delta_{qrs}^{(v)} \cdot w_{q, f, k - S_H \cdot (r-1), l - S_W \cdot (s-1)}^{(v)}$$

สำหรับ  $f = 1, \dots, F'; k = 1, \dots, H'$  และ  $l = 1, \dots, W'$  เมื่อ  $\Omega_r = \Omega(k, S_H, H_F); \Omega_s = \Omega(l, S_W, W_F); F, H_F, W_F, S_H, S_W$  คือจำนวนลักษณะสำคัญ ขนาดพิลเตอร์ตามแนวตั้ง ขนาดพิลเตอร์ตามแนวโนน ขนาดก้าวย่างตามแนวตั้ง ขนาดก้าวย่างตามแนวโนนของชั้น  $v^{th}$  ตามลำดับ,  $F', H', W'$  คือจำนวนลักษณะสำคัญ, ขนาดเอาร์พุตแนวตั้ง, ขนาดเอาร์พุตแนวโนนของชั้น  $(v-1)^{st}$  ตามลำดับ และ จากสมการ 6.24,  $\Omega(k, S, H) = \{\frac{k-i}{S} + 1 : (k-i \geq 0) \text{ and } ((k-i) \bmod S = 0)\}_{i=1, \dots, H}$ .

พารามิเตอร์ที่นิยม. แม้ว่าปัจจุบันอาจจะยังไม่มีทฤษฎีที่ศึกษาอย่างดีรับรองการเลือกพารามิเตอร์ต่าง ๆ แต่ค่าพารามิเตอร์ที่นิยมใช้สำหรับโครงข่ายคอนโวโลชัน ได้แก่ สำหรับชั้นคอนโวโลชัน นิยมใช้กับ พิลเตอร์ขนาดเป็น  $3 \times 3$  หรือ  $5 \times 5$  และขนาดก้าวย่างเป็น  $1 \times 1$ . สำหรับชั้นดึงรวม นิยมใช้กับ พิลเตอร์ขนาดเป็น  $2 \times 2$  หรือ  $3 \times 3$  และขนาดก้าวย่างเป็น  $2 \times 2$ .

## 6.5 โครงข่ายคอนโวลูชันที่สำคัญ

การออกแบบโครงสร้างของโครงข่ายคอนโวลูชันในปัจจุบัน นิยมทำด้วยมุ่งมอง การกำหนดสาระสำคัญในระดับสูง (high level of abstraction). นั่นคือ คล้ายกับการออกแบบระบบเครือข่ายที่ต้องการคุณสมบัติอะไรมากกันนัก แทนที่จะออกแบบจริงโดยเลือกอุปกรณ์พื้นฐาน เช่น ทรานซิสเตอร์ ตัวต้านทาน ตัวเก็บประจุ ไดโอด รีเลย์ และการเชื่อมต่อระหว่างอุปกรณ์พื้นฐานเหล่านี้ ปัจจุบันการออกแบบจริงนิยมเริ่มจากการเลือกวิธีการรวมก่อน แล้วค่อยเสริม ประกอบ หรือดัดแปลง ให้เข้ากับความต้องการ.

เช่นเดียวกัน การออกแบบโครงสร้างของโครงข่ายคอนโวลูชันในปัจจุบัน ก็นิยมเริ่มจากโครงข่ายคอนโวลูชันที่รู้จักกันดีแล้ว และดัดแปลงตามความเหมาะสม. หัวข้อต่อไปนี้ อภิปรายตัวอย่างของโครงข่ายคอนโวลูชันเด่น ๆ ที่รู้จักกันดี และนิยมถูกเลือกมาเป็นจุดเริ่มต้นของโครงสร้าง เช่น อเล็กซ์เน็ต[113], วีจีจีเน็ต[186], อินเซปชัน[195], เรสเน็ต[85], และเดนซ์เน็ต[92].

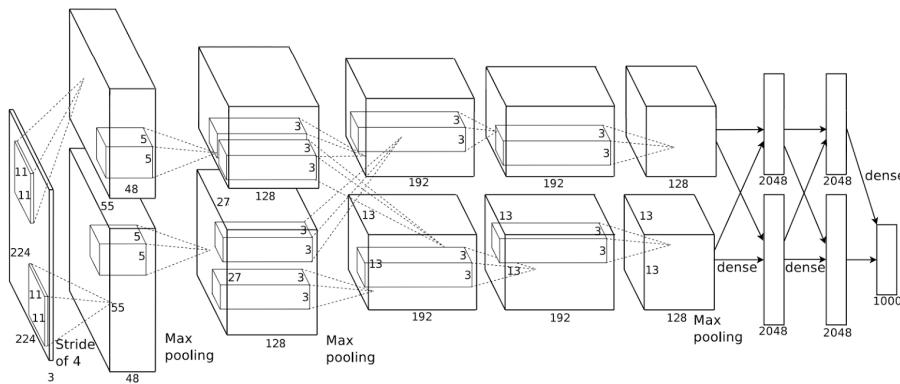
### อเล็กซ์เน็ต

อเล็กซ์เน็ต (AlexNet[113]) เป็นโครงข่ายคอนโวลูชัน ที่ได้รับความสนใจอย่างมาก หลังจากชนะการแข่งขันจำแนกชนิดวัตถุในภาพถ่าย อินเมจเนต (ImageNet) ในปี 2012 (ชุดข้อมูลมักถูกอ้างอิงว่า ImageNet LSVRC-2012).

อเล็กซ์เน็ตเป็นงานแรก ๆ ที่แสดงความสามารถการทำงานจากเครื่อง ที่ใกล้เคียงกับระดับของมนุษย์ได้. การแข่งขัน<sup>5</sup> ทดสอบผลด้วย ภาพถ่าย 100,000 ภาพ ที่แต่ละภาพมีฉลากเฉลยของชนิดวัตถุในภาพ. ชุดข้อมูลครอบคลุมถึง 1000 ชนิดวัตถุ. ผลตัดสินวัดจากค่าผิดพลาดของห้าชนิดอันดับสูงสุด (top-5 error rate) ซึ่งอเล็กซ์เน็ตทำได้ต่ำถึง 15.3%. หมายเหตุ ค่าผิดพลาดของห้าชนิดอันดับสูงสุด หมายถึง อัตราการทายผิด ซึ่งคือ อัตราส่วน จำนวนตัวอย่างที่ฉลากเฉลยไม่ถูกต้องในห้าชนิดอันดับแรกสุดที่ทาย ต่อจำนวนตัวอย่างทั้งหมด.

อเล็กซ์เน็ตใช้ขั้นตอนโวลูชัน 5 ขั้น แล้วตามด้วยขั้นเชื่อมต่อเติมที่ 3 ขั้น รวมแล้วใช้ พารามิเตอร์รวม 60 ล้านตัว. อเล็กซ์เน็ต ใช้เรลูเป็นฟังก์ชันกราฟตุน เพื่อช่วยให้การเรียนรู้ทำได้ง่ายขึ้น และใช้กลไกตอกออก เพื่อลดปัญหาโอเวอร์ฟิตติ้ง. ที่สำคัญคือ อเล็กซ์เน็ต ใช้การประมวลผลจีพียูอย่างมีประสิทธิภาพ. อเล็กซ์เน็ตถูกฝึกกับตัวอย่างภาพรวม 1.2 ล้านภาพ (จากชุดข้อมูลอินเมจเนต ของปี 2010 หรือ LSVRC-2010).

<sup>5</sup> <http://www.image-net.org/challenges/LSVRC/2012/> (ข้อมูลปรับปรุง 9 ก.พ. 2014. สืบค้น 25 ก.ค. 2020)



รูปที่ 6.14: โครงสร้างของเล็ตซ์เน็ต (ปรับปรุง เพิ่มความสมบูรณ์จากต้นฉบับ[113]). แผนที่อินพุต (ภาพสีกำหนดขนาดตามตัว) สัดส่วน  $224 \times 224 \times 3$  แสดงด้วยภาพกล่องสี่เหลี่ยมจั่ยมีสุด พื้นที่ขนาดในแนวนอนและแนวตั้ง รวมถึงจำนวนช่องสี. กล่องสี่เหลี่ยมลึกลายในแผนที่อินพุต แสดงฟิลเตอร์ของชั้นคอนโวโลชัน (ขนาด  $11 \times 11$ ) เส้นประจากฟิลเตอร์ ระบุทิศทางของผลลัพธ์การคำนวน. กล่องสี่เหลี่ยมลัดมา แสดงแผนที่ลักษณะสำคัญที่ได้จากชั้นคอนโวโลชันแรก. เพื่อแก้ปัญหาขนาดความจำที่จำกัด ของเล็ตซ์เน็ตแยกการคำนวนออกเป็นสองเส้นทาง (และใช้การดึงพิภูมิสองการ์ด แต่ละการ์ดคำนวนแต่ละเส้นทาง). ดังนั้น แผนที่ลักษณะสำคัญชั้นแรก จึงแยกเป็นสองแผนที่ (เส้นทางบน และเส้นทางล่าง) จำนวนช่องในแผนที่ชั้นแรก (ซึ่งเท่ากับจำนวนฟิลเตอร์) คือ 48 (ในแต่ละเส้นทาง หรือรวม 96 ช่องในชั้นแรกทั้งสองเส้นทาง). ชั้นคอนโวโลชันที่สอง ใช้ฟิลเตอร์ขนาด  $5 \times 5$  จำนวน 128 ในแต่ละเส้นทาง. ชั้นคอนโวโลชันที่สาม ใช้ฟิลเตอร์ขนาด  $3 \times 3$  โดยมีการนำผลลัพธ์จากแต่ละเส้นทางมาประกอบรวมกัน เป็นแผนที่ลักษณะสำคัญขนาด 192 ในแต่ละเส้นทาง. ชั้นคอนโวโลชันที่สี่ ใช้ฟิลเตอร์ขนาด  $3 \times 3$  จำนวน 192 ชุด ในแต่ละเส้นทาง โดยไม่มีการส่งผลลัพธ์ข้ามเส้นทาง. ชั้นคอนโวโลชันที่ห้า ใช้ฟิลเตอร์ขนาด  $3 \times 3$  จำนวน 128 ชุด ในแต่ละเส้นทาง. สังเกตว่า หลังชั้นคอนโวโลชันที่หนึ่ง ที่สอง และที่ห้า มีขั้นดึงรวมแบบมากที่สุด ทำให้ขนาดของแผนที่ลักษณะสำคัญในชั้นถัดไปลดลงเรื่อยๆ เท่าตัว ( $55 \times 55$  ลดลงเป็น  $27 \times 27$ ,  $27 \times 27$  ลดลงเป็น  $13 \times 13$ ). ส่วนแผนที่ลักษณะสำคัญของชั้นคอนโวโลชันที่ห้า จะถูกลดขนาดลง ก่อนถูกสลายโครงสร้าง เพื่อนำไปคำนวนกับชั้นเชื่อมต่อเต็มที่ต่อไป. หมายเหตุ ขนาดของแผนที่ลักษณะสำคัญในชั้นที่หนึ่ง  $55 \times 55$  ที่ลดลงจาก  $224 \times 224$  เกิดจากการใช้ขนาดย่างก้าว 4. ). ชั้นเชื่อมต่อเต็มที่ชั้นแรก มีจำนวนหน่วยช่อง 2048 ต่อเส้นทาง ซึ่งมีการส่งผลลัพธ์ข้ามเส้นทาง และทำให้โครงสร้างสองเส้นทาง ทำงานเหมือนหนึ่งชั้นเชื่อมต่อเต็มที่ขนาด 4096. ชั้นเชื่อมต่อเต็มที่ชั้นสอง ก็มีจำนวนหน่วยช่อง 2048 ต่อเส้นทางเช่นกัน. ชั้นเชื่อมต่อเต็มที่ชั้นที่สาม ซึ่งเป็นชั้นสุดท้าย และเป็นชั้นเอาร์พุตของเล็ตซ์เน็ต มีจำนวนหน่วยช่อง 1000 หน่วย (เท่ากับจำนวนชนิดคลิกที่ต้องการทำนาย) และใช้พังก์ชันกระตุ้นเป็นซอฟต์แมกซ์.

โครงสร้างของเล็ตซ์เน็ต แสดงดังรูป 6.14. โครงสร้างของเล็ตซ์เน็ต แยกการคำนวนออกเป็นสองเส้นทาง เพื่อแก้ปัญหาขนาดความจำ โดยใช้การดึงพิภูมิสองการ์ดร่วมกัน. การคำนวนของชั้นเชื่อมต่อเต็มที่ แม้จะแบ่งส่วนคำนวน (กระจายภาระทางハードแวร์) แต่การนำผลลัพธ์มาร่วมกันทำให้ผลลัพธ์ที่ได้สมீอันกับว่าไม่มีการแบ่งส่วน.

## 6.6 อภิธานศัพท์

**โครงข่ายคอนโวโลชัน (convolution neural network):** โครงข่ายประสาทเทียมที่มีชั้นคอนโวโลชัน.

**ชั้นคอนโวโลชัน (convolution layer):** ชั้นคำนวนที่อาศัยการเชื่อมต่อท้องถิ่นและการใช้ค่าน้ำหนักร่วม.

**ฟิลเตอร์ (filter):** ชุดค่าน้ำหนักของชั้นคอนโวลูชัน อาจเรียกว่า เคอร์เนล.

**เคอร์เนล (kernel):** ชุดค่าน้ำหนักของชั้นคอนโวลูชัน อาจเรียกว่า ฟิลเตอร์.

**การเติมเต็มด้วยศูนย์ (zero-padding):** การขยายมิติของอินพุตของชั้นคอนโวลูชัน ด้วยการเพิ่มมิติที่มีค่าเป็นศูนย์เข้าไป โดยมักมีจุดประสงค์ เพื่อควบคุมขนาดของเอาต์พุตของชั้นคอนโวลูชัน

**ขนาดก้าวย่าง (stride):** ขนาดการขยับตำแหน่งของอินพุต เพื่อนำมาคำนวณเอาต์พุตหน่วยถัดไป.

**แผนที่ลักษณะสำคัญ (feature map):** เอ้าต์พุตจากชั้นคอนโวลูชัน

**ชั้นตึงรวม (pooling layer):** ชั้นคำนวณ ที่สรุปสถิติของบริเวณห้องนิ่นต่าง ๆ ของอินพุตอย่างมาก.

**ชั้นเชื่อมต่อเต็มที่ (fully connected layer):** ชั้นคำนวณโครงข่ายประสาทเทียมแบบดั้งเดิม (นั่นคือ ไม่คำนึงถึงโครงสร้างมิติของอินพุต และไม่มีโครงสร้างมิติของเอาต์พุต).

## 6.7 แบบฝึกหัด

“Success is not final, failure is not fatal, it is the courage to continue that counts.”

---Winston Churchill

“ความสำเร็จไม่ใช่สิ่นสุด ความล้มเหลวไม่ใช่จุดจบ มีเพียงความกล้าหาญที่ไปต่อเท่านั้นที่สำคัญ.”

—วินสตัน เชอร์ชิล

### แบบฝึกหัด 6.1

จงตอบคำถามต่อไปนี้ เกี่ยวกับขั้นตอนโวลูชัน ลำดับขั้น และชุดมิติต่าง ๆ

- (ก) อินพุตเป็นเวกเตอร์ นั่นคือ  $\mathbf{x} \in \mathbb{R}^{10}$  และขั้นตอนโวลูชันใช้ฟิลเตอร์  $\mathbf{w}$  มีขนาด 3 จำนวน 15 ตัว โดยไม่มีการเติมเต็ม ขนาดย่างก้าวเป็น 1 และผลลัพธ์จากขั้นตอนโวลูชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำให้ ดูสมการ 6.1 (สำหรับฟิลเตอร์แต่ละตัว เอาร์พุต  $a_k = b + \sum_j w_j \cdot x_{k+j-1}$  โดย  $k = 1, \dots, H - H_F + 1$ ). สังเกต รูปแบบของเทนเซอร์ที่ใช้ คือ ชุดมิติแรกเป็นจำนวนลักษณะสำคัญ และตามด้วยชุดมิติอื่น ๆ (เช่น ชุดมิติลำดับ).
- (ข) อินพุตเป็นเทนเซอร์สองลำดับขั้น คือ  $\mathbf{X} \in \mathbb{R}^{8 \times 10}$ . ขั้นตอนโวลูชันใช้ฟิลเตอร์  $\mathbf{W}$  ขนาด  $8 \times 3$  จำนวน 15 ตัว โดยทำคอนโวลูชัน (การเขีอมต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สอง (ชุดมิติแรกเป็นเส้นข้อมูลของลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มีการเติมเต็มอินพุต และใช้ขนาดย่างก้าวเป็น 1. ผลลัพธ์จากขั้นตอนโวลูชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำให้ สำหรับฟิลเตอร์แต่ละตัว เอาร์พุต  $a_k = b + \sum_c \sum_j w_{c,j} \cdot x_{c,k+j-1}$  โดย  $c$  แทนดัชนีของช่องของลักษณะสำคัญ (ไม่มีความสัมพันธ์ในเชิงลำดับ) และ  $k = 1, \dots, H - H_F + 1$ .
- (ค) อินพุตเป็นเทนเซอร์สามลำดับขั้น นั่นคือ  $\mathbf{X} \in \mathbb{R}^{3 \times 100 \times 200}$ . ขั้นตอนโวลูชันใช้ฟิลเตอร์  $\mathbf{W}$  ขนาด  $3 \times 5 \times 5$  จำนวน 24 ตัว โดยทำคอนโวลูชัน (การเขีอมต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สองและสาม (ชุดมิติแรกเป็นเส้นข้อมูลของลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มีการเติมเต็มอินพุต และใช้ขนาดย่างก้าวเป็น 1. ผลลัพธ์จากขั้นตอนโวลูชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำให้ ดูสมการ 6.8 (สำหรับฟิลเตอร์แต่ละตัว เมื่อขนาดก้าวย่างเป็นหนึ่ง เอาร์พุต  $a_{k,l} = b + \sum_c \sum_i \sum_j w_{c,i,j} \cdot x_{c,k+i-1, l+j-1}$ ).
- (ง) อินพุตเป็นเทนเซอร์สี่ลำดับขั้น นั่นคือ  $\mathbf{X} \in \mathbb{R}^{4 \times 300 \times 400 \times 50}$ . ขั้นตอนโวลูชันใช้ฟิลเตอร์  $\mathbf{W}$  ขนาด  $4 \times 11 \times 11 \times 7$  จำนวน 64 ตัว โดยทำคอนโวลูชัน (การเขีอมต่อห้องถินและใช้

ค่าน้ำหนักร่วม) เนพาะกับชุดมิติที่สอง ที่สาม และที่สี่ (ชุดมิติแรกเป็นสมือนของลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มีการเติมเต็มอินพุต และใช้ขนาดย่างก้าวเป็น 1. ผลลัพธ์จากชั้นคอนโวโลชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำให้ สำหรับฟิลเตอร์แต่ละตัว เอาต์พุต  $a_{k,l,m} = b + \sum_c \sum_i \sum_j \sum_q w_{c,i,j,q} \cdot x_{c,k+i-1,l+j-1,m+q-1}$ .

### แบบฝึกหัด 6.2

จากแบบฝึกหัด 6.1 จงประมาณขนาดเทนเซอร์ของเอาต์พุต ในกรณีต่าง ๆ เมื่อใช้ขนาดย่างก้าวเป็น 2, เป็น 3 และเป็น 4. คำให้ ดูสมการ 6.4 ( $H' = \left\lceil \frac{H-H_F}{S} \right\rceil + 1$ ).

### แบบฝึกหัด 6.3

จากแบบฝึกหัด 6.1 จงประมาณการเติมเต็มด้วยค่าศูนย์ (จำนวนค่าศูนย์ที่ต้องเติม) ทั้ง 4 กรณี โดย

- (แบบที่ 1) เติมให้เอาต์พุตมีขนาดเท่ากับอินพุต เมื่อใช้ขนาดก้าวย่างเป็น 1 (พิจารณาเฉพาะในชุดมิติที่มีความสัมพันธ์เชิงลำดับ เช่น กรณี  $x$  อินพุต  $\mathbf{X} \in \mathbb{R}^{8 \times 10}$  แต่ชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ. ดังนั้น สัดส่วนของเอาต์พุตที่ต้องการคือ  $15 \times 10$  หรือเอาต์พุต  $\mathbf{A} \in \mathbb{R}^{15 \times 10}$ . ขนาด 15 มาจากจำนวนฟิลเตอร์ที่ใช้ ไม่เกี่ยวกับการเติมเต็มด้วยค่าศูนย์).
- (แบบที่ 2) เติมให้เอาต์พุตมีขนาดเท่ากับ  $\lceil \frac{H}{S} \rceil$  โดย  $H$  คือขนาดอินพุต และ  $S$  คือขนาดก้าวย่าง เมื่อใช้ขนาดก้าวย่างเป็น 2, เป็น 3, และเป็น 4 ตามลำดับ. (พิจารณาเฉพาะในชุดมิติที่มีความสัมพันธ์เชิงลำดับ เช่น กรณี  $c$  อินพุต  $\mathbf{X} \in \mathbb{R}^{3 \times 100 \times 200}$  แต่ชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ. ดังนั้น สัดส่วนของเอาต์พุตที่ต้องการคือ  $24 \times 50 \times 100$  เมื่อใช้ขนาดก้าวย่าง 2 และคือ  $24 \times 34 \times 67$  เมื่อใช้ขนาดก้าวย่าง 3 เป็นต้น).

คำให้ ดูสมการ 6.5 ซึ่งคือ  $\hat{H} = S \cdot (\hat{H}' - 1) + H_F$  และ  $\hat{H} - H$ .

### แบบฝึกหัด 6.4

จงคำนวณขนาดของเอาต์พุตจากชั้นคอนโวโลชัน สำหรับกรณีต่าง ๆ ดังนี้

- (ก) อินพุตเป็นเวกเตอร์ นั่นคือ  $\mathbf{x} \in \mathbb{R}^{10}$  และชั้นคอนโวโลชันใช้ฟิลเตอร์  $\mathbf{w}$  มีขนาด 3 จำนวน 15 ตัว โดยเติมเต็มด้วยค่าศูนย์จำนวนรวม 2 ตัว ขนาดย่างก้าวเป็น 1 และผลลัพธ์จากชั้นคอนโวโลชัน จะเป็นเทนเซอร์ขนาดเท่าใด?

- (ก) อินพุตเป็นเทนเซอร์สองลำดับชั้น คือ  $\mathbf{X} \in \mathbb{R}^{8 \times 10}$ . ชั้นคอนโวโลชันใช้ฟิลเตอร์  $\mathbf{W}$  ขนาด  $8 \times 3$  จำนวน 15 ตัว โดยทำคอนโวโลชัน (การเชื่อมต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สอง (ชุดมิติแรกเป็นเสมือนช่องลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). มีการเติมเต็มอินพุตด้วยค่าศูนย์จำนวน 7 ตัว และใช้ขนาดย่างก้าวเป็น 2. ผลลัพธ์จากชั้นคอนโวโลชัน จะเป็นเทนเซอร์ขนาดเท่าใด?
- (ค) อินพุตเป็นเทนเซอร์สามลำดับชั้น นั่นคือ  $\mathbf{X} \in \mathbb{R}^{3 \times 100 \times 200}$ . ชั้นคอนโวโลชันใช้ฟิลเตอร์  $\mathbf{W}$  ขนาด  $3 \times 5 \times 5$  จำนวน 24 ตัว โดยทำคอนโวโลชัน (การเชื่อมต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สองและสาม (ชุดมิติแรกเป็นเสมือนช่องลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). มีการเติมเต็มอินพุตด้วยค่าศูนย์จำนวน 11 ตัวในแต่ละชุดมิติ (ยกเว้นชุดมิติแรก) และใช้ขนาดย่างก้าวเป็น 3. ผลลัพธ์จากชั้นคอนโวโลชัน จะเป็นเทนเซอร์ขนาดเท่าใด?

คำให้  $H' = \left\lfloor \frac{H-H_F+P}{S} \right\rfloor + 1$  เมื่อ  $P$  คือจำนวนศูนย์ที่เติมเข้าไปทั้งหมด.

### แบบฝึกหัด 6.5

จงคำนวณขนาดของสนามรบรู้ของหน่วยย่อยในชั้นสุดท้ายของกรณีต่อไปนี้

- (ก) โครงข่ายคอนโวโลชันหนึ่งชั้น ที่ใช้ฟิลเตอร์ขนาด  $5 \times 5$  ขนาดก้าวย่างเป็น  $1 \times 1$ , เป็น  $2 \times 2$  และเป็น  $3 \times 3$  ตามลำดับ.
- (ข) โครงข่ายคอนโวโลชันสองชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด  $5 \times 5$  ขนาดก้าวย่างเป็น  $1 \times 1$ .
- (ค) โครงข่ายคอนโวโลชันสองชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด  $11 \times 11$  ขนาดก้าวย่างเป็น  $1 \times 1$ .
- (ง) โครงข่ายคอนโวโลชันสามชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด  $5 \times 5$  ขนาดก้าวย่างเป็น  $1 \times 1$ .
- (จ) โครงข่ายคอนโวโลชันห้าชั้น โดยฟิลเตอร์ชั้นแรก  $11 \times 11$  ก้าวย่าง  $1 \times 1$ , ฟิลเตอร์ชั้นสอง  $5 \times 5$  ก้าวย่าง  $1 \times 1$ , ฟิลเตอร์ชั้นสามถึงห้าใช้ฟิลเตอร์แบบเดียวกัน คือ  $3 \times 3$  ก้าวย่าง  $1 \times 1$ .
- (ฉ) โครงข่ายคอนโวโลชันห้าชั้นสิบชั้น โดยทุกชั้นใช้ฟิลเตอร์แบบเดียวกัน คือ  $3 \times 3$  ก้าวย่าง  $1 \times 1$ .
- (ช) โครงข่ายคอนโวโลชันสามชั้น ชั้นที่หนึ่งและสามใช้ฟิลเตอร์ขนาด  $3 \times 3$  ขนาดก้าวย่างเป็น  $1 \times 1$  แต่ชั้นที่สองใช้ฟิลเตอร์ขนาด  $2 \times 2$  ก้าวย่าง  $2 \times 2$ .

คำให้ ดูสมการ 6.12 ( $R_k = 1 + \sum_{j=1}^k (F_j - 1) \prod_{i=0}^{j-1} S_i$  และกำหนด  $S_0 = 1$ )

### แบบฝึกหัด 6.6

จากสมการ 6.10 และ 6.11 สำหรับคุณโวluชั่นสองมิติ จะเขียนสมการคำนวณแผนที่ลักษณะสำคัญ (เอาร์พุต) ของชั้นคุณโวluชั่น สำหรับ

- (ก) คุณโวluชั่นหนึ่งมิติ (มีชุดลำดับมิติชุดเดียว อินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H}$  โดยชุดมิติแรกไม่มีความสัมพันธ์ เชิงลำดับ).
- (ข) คุณโวluชั่นสามมิติ (มีชุดลำดับมิติสัมพันธ์สามชุด อินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H \times W \times D}$  โดยชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ).
- (ค) คุณโวluชั่นสี่มิติ (มีชุดลำดับมิติสัมพันธ์สี่ชุด อินพุต  $\mathbf{X} \in \mathbb{R}^{C \times H \times W \times D \times E}$  โดยชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ).

**การโปรแกรมตรรกะของโครงข่ายคุณโวluชั่น.** การคำนวณของโครงข่ายคุณโวluชั่น ประกอบด้วยการคำนวณของชั้นคำนวณสามชนิดหลัก ๆ ได้แก่ ชั้นคำนวณคุณโวluชั่น ชั้นดึงรวม และชั้นเข้มต่อเต็มที่. รายการ 6.1 แสดงตัวอย่างโปรแกรมของชั้นคำนวณคุณโวluชั่น. โปรแกรมในรายการ 6.1 อาศัยการจัดเรียงเทนเซอร์ใหม่ และใช้ประโยชน์จากการคุณเมทริกซ์. รูป 6.15 แสดงแนวคิด การจัดเรียงเทนเซอร์ใหม่ เพื่อที่การคุณเมทริกซ์จะให้ผลลัพธ์เหมือนการคำนวณคุณโวluชั่น. สังเกต สมการ 6.10 เอาร์พุต  $\mathbf{a}$  เป็นเทนเซอร์สัดส่วน  $M \times H' \times W'$  (เมื่อ  $M$  เป็นจำนวนลักษณะสำคัญ และ  $H'$  กับ  $W'$  เป็นขนาดความสูงและกว้างของแผนที่เอาร์พุต) สำหรับชุดข้อมูลแต่ละจุด. ดังนั้น สำหรับชุดข้อมูลหนึ่งขนาด  $N$  ผลลัพธ์จะเป็นเทนเซอร์สัดส่วน  $N \times M \times H' \times W'$ . เอาร์พุตจากการคุณเมทริกซ์  $\mathbf{W}_{M \times C \cdot H_f \cdot W_f} \cdot \mathbf{X}_{C \cdot H_f \cdot W_f \times H' \cdot W' \cdot N}$  จะเป็นเมทริกซ์ขนาด  $M \times H' \cdot W' \cdot N$  ซึ่งสามารถนำมารัดเรียงเป็นเทนเซอร์สัดส่วน  $N \times M \times H' \times W'$  ได.

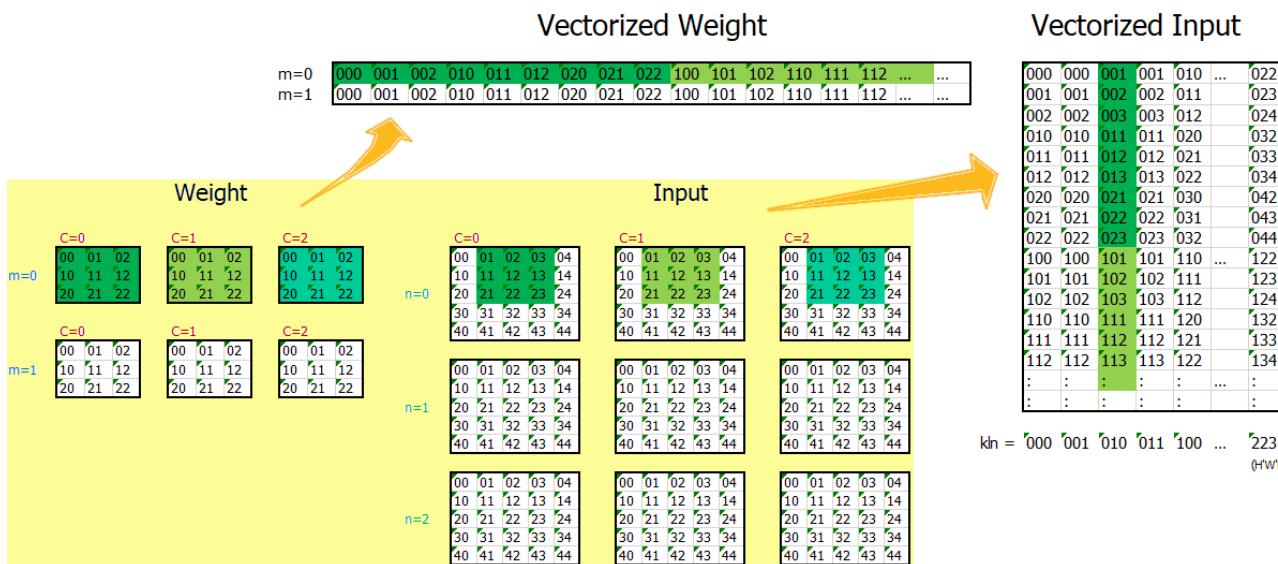
หมายเหตุ การเขียนโปรแกรมคำนวณคุณโวluชั่น เช่น สมการ 6.10 ด้วยการวนลูป ก็สามารถทำได้ แต่การทำงานอาจทำได้ช้ามาก. ผู้อ่านสามารถทดลองวิธีการเขียนโปรแกรมหลาย ๆ แนวทาง และเปรียบเทียบข้อดีข้อเสีย ในแต่ละ ฯ เช่น ประสิทธิภาพการทำงาน ความยากง่ายในการแก้ไขและปรับปรุง.

รายการ 6.1: ตัวอย่างโปรแกรมชั้นคำนวณคุณโวluชั่น

---

```

1 class MyConv2D(nn.Module):
2     def __init__(self, input_channels, num_kernels, kernel_size,
```



รูปที่ 6.15: ตัวอย่างการจัดเทนเซอร์ของค่าน้ำหนักและอินพุตให้อยู่ในรูปที่การคูณเมทริกซ์เสมือนการคำนวณเทนเซอร์. ภาพภายในพื้นหลังสีเหลืองอ่อน แสดงค่าน้ำหนัก และอินพุต. ในภาพ  $m$  แทนดัชนีของฟิลเตอร์ หรือลักษณะสำคัญ. ตัว  $C$  เป็นดัชนีของช่อง. ค่าน้ำหนักซึ่งเป็นเทนเซอร์สีลำดับชั้น  $w \in \mathbb{R}^{2 \times 3 \times 3 \times 3}$  (สองฟิลเตอร์ แต่ละฟิลเตอร์มีสามช่อง และแต่ละช่องขนาด  $3 \times 3$ ) แสดงด้วยกรอบหากรอบจัดเรียงเป็นสองแถว (ตามจำนวนฟิลเตอร์) และสามสมมุติ (ตามจำนวนช่อง). ตัวเลขภายในกรอบแสดงค่าดัชนีเชิงพื้นที่ในแนวตั้งและแนวนอน (จากเรียกเป็น  $(i, j)$  แต่ไม่ได้ระบุในภาพ). อินพุตซึ่งเป็นเทนเซอร์สีลำดับชั้น  $x \in \mathbb{R}^{3 \times 3 \times 5 \times 5}$  (สามจุดข้อมูล แต่ละจุดข้อมูลมีสามช่อง แต่ละช่องขนาด  $5 \times 5$ ) ใช้ตัว  $n$  แสดงดัชนีของจุดข้อมูล. พื้นที่แรงงานสีเขียว (เขียวแก่ เขียวอ่อน และเขียวไข่กา) แสดงความสัมพันธ์ของการคำนวณในก้าวย่างที่สองตามแนวนอน (จากเรียกเป็น  $(k, l) = (0, 1)$  เมื่อ ขนาดก้าวย่าง  $S = 1$ ). ถูกศร ที่อยู่บนเทนเซอร์สีลำดับชั้นในรูปเดิม กับรูปแบบใหม่ที่สะกดการคำนวณ. แต่ละแถวของเมทริกซ์ของค่าน้ำหนัก แทนฟิลเตอร์แต่ละตัว (ดัชนี  $m$  ช่วยระบุ). แต่ละค่าภายในแถว แสดงด้วยตัวเลขเดียวกัน  $(c, i, j)$  สำหรับช่อง ตำแหน่งแนวตั้ง และตำแหน่งแนวนอน. สำหรับเมทริกซ์ของอินพุตที่จัดใหม่ แต่ละสมมุติ แทนตำแหน่งของເອົາພຸດและຈຸດຂ້ອມູນ (ดัชนี  $(k, l, n)$  สำหรับตำแหน่งເອົາພຸດแนวตั้ง แนวนอน และຈຸດຂ້ອມູນที่  $n^{th}$ ). ภายใต้สมมุติ แสดงดัชนีของช่อง ตำแหน่งแนวตั้ง และตำแหน่งแนวนอนของอินพุตเดิม  $(c, i, j)$ . สังเกต การจัดเรียงดัชนีในเมทริกซ์อินพุต ที่ทำให้การคูณเมทริกซ์เป็นเสมือนการคำนวณconvolution.

```

3                         stride=1, padding=0):
4     super(MyConv2D, self).__init__()
5     self.input_channels = input_channels
6     self.num_kernels = num_kernels
7     self.kernel_size = kernel_size
8     self.stride = stride
9     self.padding = padding
10
11    # initialization with pytorch default
12    sqk = torch.sqrt(torch.Tensor([1/(input_channels * \
13                                kernel_size * kernel_size)]))
14    initw = 2*sqk*torch.rand(num_kernels, input_channels,
15                            kernel_size, kernel_size) - sqk
16    initb = 2*sqk*torch.rand(num_kernels, 1) - sqk

```

```

17         self.weight = nn.Parameter(initw)
18         self.bias = nn.Parameter(initb)
19
20     def forward(self, z):
21         """
22             Eq. 6.10:  $a_{f,k,l}^{(v)} = b_f^{(v)} + \sum_{c=1}^C \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} w_{fcij}^{(v)} \cdot z_{c, S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}^{(v-1)}$ 
23         """
24
25         M, C, Hf, Wf = self.weight.shape
26         N, D, H, W = z.shape
27         assert C == D, 'Numbers of channels are not matched.'
28
29         S = self.stride
30         P = self.padding
31
32         # Determine output size
33         Ho = int((H + 2*P - Hf)/S) + 1
34         Wo = int((W + 2*P - Wf)/S) + 1
35
36         # Simplify z structure
37         simplified_z = self._simplify_struct(z, Hf, Wf, S, P)
38         assert simplified_z.shape == (D * Hf * Wf, Ho * Wo * N)
39
40         simplified_w = self.weight.view(M, -1)
41         assert simplified_w.shape == (M, C * Hf * Wf)
42
43         # Compute convolution
44         simplified_out = self.bias + simplified_w.mm(simplified_z)
45
46         # Restructure convoluted output back
47         conv_out = simplified_out.view(M, Ho, Wo, N)
48         a = conv_out.permute(3, 0, 1, 2) # output (N, M, H', W')
49
50     return a
51
52
53     @staticmethod
54     def _simplify_struct(z, Hf, Wf, S, P):
55         """
56             Collapse z structure such that convolution can be
57             efficiently computed as matrix multiplication.

```

```

58     ...
59
60     # Zero-pad the input (on Last 2 dimensions)
61     zhat = torch.nn.functional.pad(z, (P,P,P,P,0,0,0,0),
62                                    'constant', 0)
63
64     # Get vectorized indices
65     c, rx, cx = MyConv2D._get_simplified_indices(z.shape,
66                                                    Hf, Wf, S, P)
67     # c.shape = (C Hf Wf, 1)
68     # rx.shape = (C Hf Wf, Ho Wo)
69     # cx.shape = (C Hf Wf, Ho Wo)
70
71     # Re-arrange input into a simplified structure
72     simz = zhat[:, c, rx, cx]    # shape (N, C Hf Wf, Ho Wo)
73
74     num_channels = z.shape[1]
75
76     sim_z = simz.permute(1, 2, 0).contiguous().view(\n
77                 num_channels * Hf * Wf, -1)
78
79     return sim_z # shape = (C Hf Wf, Ho Wo N)
80
81     @staticmethod
82     def _get_simplified_indices(input_shape, Hf, Wf, S, P):
83         ...
84
85         return indices of re-arranged vector ready for
86         dot operation (in lieu of convolution).
87         ...
88
88     N, C, H, W = input_shape
89
90     # Determine output size
91     Ho = int((H + 2*P - Hf)/S) + 1
92     Wo = int((W + 2*P - Wf)/S) + 1
93
94     # To match the re-arranged weight,
95     # input must be re-arranged accordingly.
96     # weight row: f = 0, 1, ..., (M-1)
97     # weight column: cij = 000, 001, 002, 010, 011, ...
98     # Thus, input row: cij

```

```

99      # input column: k,l = 00, 01, 02, ..., (Ho-1)(Wo-1)
100
101      # Work out indices of filter nodes
102      # j, i, c from innermost to outermost along row direction
103      j = np.tile(np.arange(Wf), Hf * C).reshape(-1, 1)
104      # e.g., j = [0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, ...]
105      i = np.tile(np.repeat(np.arange(Hf), Wf), C).reshape(-1, 1)
106      # e.g., i = [0, 0, 0, 1, 1, 1, 2, 2, 2, 0, 0, 0, ...]
107      c = np.repeat(np.arange(C), Hf*Wf).reshape(-1, 1)
108      # e.g., c = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, ...]
109
110      # Work out indices of output nodes
111      # l, k from innermost to outermost, along column
112      l = np.tile(np.arange(Wo), Ho).reshape(1, -1)
113      # e.g., L.T = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, ...]
114      k = np.repeat(np.arange(Ho), Wo).reshape(1, -1)
115      # e.g., k.T = [0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 2, ...]
116
117      # Indices of input nodes
118      rx = S * k + i # shape = (C Hf Wf, Ho Wo)
119      cx = S * l + j # shape = (C Hf Wf, Ho Wo)
120
121      return c.astype(int), rx.astype(int), cx.astype(int)

```

---

โปรแกรมในรายการ 6.2 แสดงตัวอย่างการเรียกใช้ MyConv2D. โปรแกรม MyConv2D เขียนขึ้นตามรูปแบบของไฟล์ nn.Conv2d ดังนั้น การใช้งานก็ทำในลักษณะเดียวกันได้. โปรแกรมในรายการ 6.3 และ 6.4 แสดงตัวอย่างฝึกและทดสอบโครงสร้าง (ค่าอภิมานพารามิตเตอร์ต่าง ๆ ใช้ได้กับชุดข้อมูลเอมนิสต์. ดูแบบฝึกหัด 5.10 สำหรับตัวอย่างการนำเข้าชุดข้อมูลเอมนิสต์).

รายการ 6.2: ตัวอย่างการเรียกใช้ชั้นคำนวณคอนโวลูชัน MyConv2D

---

```

1 class NetConv1(nn.Module):
2
3     def __init__(self):
4         super(NetConv1, self).__init__()
5         self.conv1 = MyConv2D(1, 16, 5, 1, 2)
6         self.pool1 = nn.MaxPool2d(2, 2)
7         self.conv2 = MyConv2D(16, 8, 3, 1, 1)
8         self.pool2 = nn.MaxPool2d(2, 2)
9         self.fc1 = nn.Linear(8*7*7, 10)
10

```

```

11     def forward(self, x):
12         z1 = torch.relu(self.conv1(x))
13         z2 = self.pool1(z1)
14         z3 = torch.relu(self.conv2(z2))
15         z4 = self.pool2(z3)
16         z5 = z4.view(-1, 8 * 7 * 7)
17         a6 = self.fc1(z5)
18         return a6
19
20 net = NetConv1().to(device)
21 loss_fn = torch.nn.CrossEntropyLoss()
22 optimizer = optim.Adam(net.parameters(), lr=0.001)

```

---

รายการ 6.3: การฝึกโครงข่ายที่ใช้ชั้นคำนวณ convolutional ชั้น MyConv2D สามารถทำได้แบบเดียวกับโครงข่ายประสาทเทียมอื่น ๆ

```

1 num_epochs = 20
2 N = len(trainloader) * 50 # 50 samples a batch
3
4 for epoch in range(num_epochs):
5
6     running_loss = 0.0
7     for i, data in enumerate(trainloader, 0):
8         inputs, labels = data
9         optimizer.zero_grad()
10        outputs = net(inputs.to(device))
11        loss = loss_fn(outputs.to('cpu'), labels)
12        loss.backward()
13        optimizer.step()
14
15        running_loss += loss.item()
16    # end for i
17    print('Epoch %d loss: %.3f' % (epoch + 1, running_loss / N))
18
19 torch.save(net.state_dict(), './conv1_net.pth')

```

---

รายการ 6.4: การทดสอบโครงข่ายที่ใช้ชั้นคำนวณ convolutional ชั้น MyConv2D สามารถทำได้ เช่นเดียวกับการทดสอบโครงข่ายประสาทเทียมอื่น ๆ

```

1 net.eval()
2 N = len(testloader) * 50 # 50 samples a batch
3
4 correct = 0
5 for i, data in enumerate(testloader):

```

```

6   inputs, labels = data
7   outputs = net(inputs.to(device))
8   yhat = outputs.to('cpu')
9   yhatc = torch.argmax(outputs, 1)
10  correct += torch.sum(yhatc.cpu() == labels).numpy()
11
12 print('Correct %d out of %d' %(correct, N))
13 print('Accuracy %.3f' %(correct/N))

```

---

## แบบฝึกหัด 6.7

จงศึกษาการทำงานของชั้นคำนวณคอนโวลูชันและวิธีการเขียนโปรแกรมในรายการ 6.1 และทดสอบการทำงานเปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.Conv2d` รวมถึงทดสอบโครงสร้างแบบอื่น ๆ (เปลี่ยนค่าอภิมานพารามิเตอร์ เช่น ขนาดฟิลเตอร์ จำนวนฟิลเตอร์ ขนาดก้าวย่าง จำนวนการเติมเต็มด้วยศูนย์) ภารกิจรายและสรุป. หมายเหตุ ในทางปฏิบัติ การใช้โปรแกรมสำเร็จรูปจะสะดวกกว่า การอ้างอิงทำได้ยากกว่า ถูกยอมรับดีกว่า (โปรแกรมมาตรฐาน เชื่อว่าได้รับการตรวจสอบมาดีกว่า) และดังเช่นที่จะได้เห็นจากการทดลอง ในกรณีนี้ โปรแกรมสำเร็จรูป `nn.Conv2d` ทำงานได้มีประสิทธิภาพมากกว่าอย่างเห็นได้ชัด (การเขียนโปรแกรมประสิทธิภาพสูง อาจต้องอาศัยการโปรแกรมระดับล่าง ซึ่งอยู่นอกเหนือขอบเขตของหนังสือเล่มนี้). แต่การศึกษาโปรแกรมในรายการ 6.1 ทำเพื่อให้เข้าใจกลไกการทำงานของชั้นคำนวณคอนโวลูชันอย่างกระฉับเจ้ง.

## แบบฝึกหัด 6.8

การเขียนโปรแกรมชั้นเชื่อมต่อเติมที่กีสามารถทำได้ในลักษณะเดียวกัน. รายการ 6.6 แสดงตัวอย่างโปรแกรมเชื่อมต่อเติมที่ที่เขียนการแพร่กระจายย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `fcf` ที่เขียนดังในรายการ 6.5. การใช้งานชั้นเชื่อมต่อเติมที่ `MyFCBack` ก็ทำเช่นเดียวกับการเรียกใช้ชั้นคำนวณ `nn.Linear` เช่น การเปลี่ยนบรรทัดคำสั่ง `self.fc1 = nn.Linear(8*7*7, 10)` ในรายการ 6.2 เป็น `self.fc1 = MyFCBack(8*7*7, 10)` เท่านั้น ที่เหลือก็สามารถดำเนินงานสร้างโครงข่าย ฝึก และทดสอบได้เช่นเดิม.

รายการ 6.5: ตัวอย่างโปรแกรมการคำนวณการเชื่อมต่อเติมที่และการแพร่กระจายย้อนกลับ `fcf`

---

```

1 class fcf(torch.autograd.Function):
2     @staticmethod
3     def forward(ctx, zp, w, b):
4         # input: zp (N,Mi):  $z_j^{(v-1)}$  , w (Mo,Mi):  $\partial w_{ji}^{(v)}$ 
        , b (Mo,1):  $\partial b_j^{(v)}$ 

```

```

5      # output: a (N,Mo):  $a_j^{(v)}$ 
6      zT = torch.transpose(zp, 0, 1)
7      a = w.mm(zT) + b
8      ctx.save_for_backward(zp, w, b)
9      return torch.transpose(a, 0, 1)
10
11     @staticmethod
12     def backward(ctx, dEa):
13         # input: dEa (N,Mo):  $\frac{\partial E}{\partial a_j^{(v)}}$ 
14         # output: dEzp:  $\frac{\partial E}{\partial z_i^{(v-1)}} = \sum_j \frac{\partial E}{\partial a_j^{(v)}} \cdot \frac{\partial a_j^{(v)}}{\partial z_i^{(v-1)}}$  , dEw:  $\frac{\partial E}{\partial w_{ji}^{(v)}} = \frac{\partial E}{\partial a_j^{(v)}} \cdot \frac{\partial a_j^{(v)}}{\partial w_{ji}^{(v)}}$ 
15         , dEb:  $\frac{\partial E}{\partial b_j^{(v)}} = \frac{\partial E}{\partial a_j^{(v)}} \cdot \frac{\partial a_j^{(v)}}{\partial b_j^{(v)}}$ 
16         N, _ = dEa.shape
17         zp, w, b = ctx.saved_tensors
18         dEzp = dEa.mm(w)
19         dEw = torch.transpose(dEa, 0, 1).mm(zp)
20         dEb = torch.transpose(dEa, 0, 1).mm(torch.ones(N,1).to(←
21             dEa.device))
22         return dEzp, dEw, dEb

```

รายการ 6.6: ตัวอย่างโปรแกรมชั้นเชื่อมต่อเติมที่ที่เขียนการแพร่กระจายย้อนกลับของ MyFCBack. ตัวอย่างนี้ เรียกใช้ฟังก์ชัน `fclf` ที่นิยามในรายการ 6.5.

```

1 class MyFCBack(nn.Module):
2     def __init__(self, input_channels, num_features):
3         super(MyFCBack, self).__init__()
4         self.input_channels = input_channels
5         self.num_features = num_features
6         sqk = torch.sqrt(torch.Tensor([1/input_channels]))
7         initw = 2*sqk*torch.rand(num_features,input_channels)-sqk
8         initb = 2*sqk*torch.rand(num_features,1) - sqk
9         self.weight = nn.Parameter(initw)
10        self.bias = nn.Parameter(initb)
11        self.fclf = fclf.apply
12
13    def forward(self, z):
14        a = self.fclf(z, self.weight, self.bias)
15        return a

```

จงทดสอบการทำงานของชั้นเชื่อมต่อเติมที่ MyFCBack เปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.Linear`

ทั้งในเชิงการทำงาน และเวลาในการทำงาน. รวมถึง จงทดลองแก้การคำนวณในฟังก์ชัน `fcf` เพื่อตรวจสอบ ดูว่าการคำนวณการเชื่อมต่อและการคำนวณแพร่กระจายย้อนกลับ ว่าได้ทำผ่าน `fcf.forward` และ `fcf.backward` จริง. ตัวอย่างเช่น ทดลองแก็บรั่วทั้ดคำสั่ง `return dEzp, dEw, dEb` เป็น `return 0*dEzp, 0*dEw, 0*dEb` และสังเกตผล. สรุป และอภิปราย.

หมายเหตุ แม้การเขียนโปรแกรมชั้นเริ่มต้นต่อเติมได้ถูกอภิปรายไปแล้วในหัวข้อ 5.7 การทบทวนอีกครั้ง ในแบบฝึกหัด เพื่อให้คุณเคยกับรูปแบบการเขียนโปรแกรมชั้นคำนวณเพื่อใช้กับไฟฟอร์ช ที่ระบุการคำนวณ การแพร่กระจายย้อนกลับด้วย. การทบทวนนี้ จะคาดว่าจะช่วยผู้อ่านเข้าใจกลไกของการเขียนโปรแกรมชั้นคำนวณพร้อมการระบุการแพร่กระจายย้อนกลับของไฟฟอร์ช ก่อนที่จะเขียนโปรแกรมชั้นตอนโดยชั้น ซึ่งขึ้นในแบบฝึกหัด 6.9.

## แบบฝึกหัด 6.9

คล้ายกับแบบฝึกหัด 6.8 แบบฝึกหัดนี้ศึกษาการเขียนโปรแกรมชั้นตอนโดยชั้นทั้งการคำนวณ และการแพร่กระจายย้อนกลับ. รายการ 6.8 แสดงตัวอย่างโปรแกรมชั้นตอนโดยชั้นที่เขียนการแพร่กระจายย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `convf` ที่เขียนดังในรายการ 6.7<sup>6</sup>. โปรแกรมชั้นตอนโดยชั้น MyConv2DB รับมารดกมาจาก MyConv2D (รายการ 6.1) เพื่อลดความซ้ำซ้อน ที่จะต้องกำหนดค่าเริ่มต้นค่าน้ำหนัก (ภายในเมธอด `__init__`). การใช้งานชั้นตอนโดยชั้น MyConv2DB ก็ทำ เช่นเดียวกับ MyConv2D เช่น การเปลี่ยนบรรทัดคำสั่ง `self.conv1 = MyConv2D(1, 16, 5, 1, 2)` และบรรทัดคำสั่ง `self.conv2 = MyConv2D(16, 8, 3, 1, 1)` ในรายการ 6.2 เป็น `self.conv1 = MyConv2DB(1, 16, 5, 1, 2)` และ `self.conv2 = MyConv2DB(16, 8, 3, 1, 1)` ตามลำดับ เท่านั้น ที่เหลือก็สามารถดำเนินงานสร้างโครงข่าย ฝึก และทดสอบได้เช่นเดิม.

รายการ 6.7: ตัวอย่างโปรแกรมการคำนวณตอนโดยชั้นพร้อมการแพร่กระจายย้อนกลับ `convf`

---

```

1 class convf(torch.autograd.Function):
2     @staticmethod
3     def forward(ctx, zp, w, b, S, P):
4         ...
5         Eq. 6.10:  $a_{f,k,l}^{(v)} = b_f^{(v)} + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{fcij}^{(v)} \cdot z_{c,S_H \cdot (k-1) + i, S_W \cdot (l-1) + j}^{(v-1)}$ 
6         zp:  $z_{cij}^{(v-1)}$ , w:  $w_{fcij}^{(v)}$ , b:  $b_f^{(v)}$ , S: stride, P: padding
7         ...

```

<sup>6</sup> รหัสโปรแกรมนี้ ดัดแปลง จาก รหัสโปรแกรมชิปส เทอร์เน็ต (Hipsternet), จาก <https://github.com/wiseodd/hipsternet/tree/master/hipsternet>, ปรับปรุงล่าสุด 12 ก.พ. 2017.

```

8
9     F, C, Hf, Wf = w.shape
10    N, D, H, W = zp.shape
11    assert C == D, 'Numbers of channels are not matched.'
12
13    # Determine output size
14    Ho = int((H + 2*P - Hf)/S) + 1
15    Wo = int((W + 2*P - Wf)/S) + 1
16
17    # Simplify z structure
18    simplified_z = MyConv2D._simplify_struct(zp, Hf, Wf, S, P)
19    assert simplified_z.shape == (D * Hf * Wf, Ho * Wo * N)
20
21    simplified_w = w.view(F, -1)
22    assert simplified_w.shape == (F, C * Hf * Wf)
23
24    # Compute convolution
25    simplified_out = b + simplified_w.mm(simplified_z)
26
27    # Restructure convoluted output back
28    conv_out = simplified_out.view(F, Ho, Wo, N)
29    a = conv_out.permute(3, 0, 1, 2) # output (N, M, H', W')
30
31    ctx.save_for_backward(zp, w, b, torch.tensor([S, P]),
32                          simplified_z)
33
34    return a
35
36    @staticmethod
37    def backward(ctx, dEa):
38        # input: dEa (N, F, H', W'):  $\delta_{qrs}^{(v)} = \frac{\partial E}{\partial a_{qrs}^{(v)}}$ 
39        # output: dEzp (N, C, H, W):
40        
$$\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E}{\partial z_{fkl}^{(v-1)}} = \sum_{q=1}^F \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1),l-S_W \cdot (s-1)}^{(v)}$$

41        #           dEw (F, C, Hf, Wf):
42        
$$\frac{\partial E}{\partial w_{qfij}^{(v)}} = \sum_{r=1}^H \sum_{s=1}^W \delta_{qrs}^{(v)} z_{f,S_H \cdot (r-1)+i,S_W \cdot (s-1)+j}^{(v-1)}$$

43        #           dEb (F, 1):  $\frac{\partial E}{\partial b_q} = \sum_{r=1}^H \sum_{s=1}^W \delta_{qrs}^{(v)}$ 
44        N, F, Ho, Wo = dEa.shape
45        zp, w, b, tensorSP, simplified_z = ctx.saved_tensors
46        S = tensorSP[0].item()

```

```

45     P = tensorSP[1].item()
46
47     _, C, Hf, Wf = w.shape
48
49     # Calculate dEb
50     dEb = dEa.sum(dim=(0,2,3)).view(-1,1) # sum over N,H',W'
51
52     # Restructure dEa from (N, F, H', W') to (F, H' W' N)
53     simplified_dEa = dEa.permute(1, 2, 3, 0).contiguous().←
54         view(F, -1)
55
56     # Calculate dEw
57     dEw = simplified_dEa.mm(simplified_z.transpose(0,1))
58     dEw = dEw.view(w.shape) # (F, C, Hf, Wf)
59
60     # Calculate dEzp (N, C, H, W):  $\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E}{\partial z_{fkl}^{(v-1)}}$ 
61     #  $\frac{\partial E}{\partial z_{fkl}^{(v-1)}} = \sum_{q=1}^F \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1),l-S_W \cdot (s-1)}^{(v)} =$ 
62     #  $\sum_{r \in \Omega_r} \sum_{s \in \Omega_s} (\sum_{q=1}^F \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1),l-S_W \cdot (s-1)}^{(v)})$ 
63     # First, sum over the feature axis
64     simplified_w = w.view(F,-1)
65     assert simplified_w.shape == (F, C * Hf * Wf)
66
67     wdEa_overF = simplified_w.transpose(0,1).mm(
68                               simplified_dEa) #(C Hf Wf, H'W'N)
69
70     # Sum over spatial indices
71     dEzp = convf.sum_omega(wdEa_overF, zp.shape, Hf, Wf, P, S)
72
73     return dEzp, dEw, dEb, None, None
74
75     @staticmethod
76     def sum_omega(prod_overF, zpshape, Hf, Wf, P, S):
77         '''
78             Summation over the two omega sets (~over H' and W')
79             input: prod_overF (C Hf Wf, H' W' N):
80                 ( $\sum_{q=1}^F \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1),l-S_W \cdot (s-1)}^{(v)}$ )
81             output: dEzp (N, C, H, W):
82                  $\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E}{\partial z_{fkl}^{(v-1)}} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} (\sum_{q=1}^F \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1),l-S_W \cdot (s-1)}^{(v)})$ 
83         '''

```

```

80
81     N, C, H, W = zpshape
82     H_hat, W_hat = H + 2*P, W + 2*P
83
84     # Restructure prod_overF for sum over omega
85     prod_overF_reshaped = prod_overF.view(C*Hf*Wf, -1, N)
86     prod = prod_overF_reshaped.permute(2, 0, 1).cpu().numpy()
87
88     # Prepare result structure
89     sum_result = np.zeros((N,C,H_hat,W_hat), dtype=prod.dtype)
90
91     # Get vectorized indices
92     c, rx, cx = MyConv2D._get_simplified_indices(zpshape,
93                                                 Hf, Wf, S, P)
94     # c.shape = (C Hf Wf, 1)
95     # rx.shape = (C Hf Wf, H' W')
96     # cx.shape = (C Hf Wf, H' W')
97
98     # Sum over omega using np.add.at mechanism
99     np.add.at(sum_result, (slice(None), c, rx, cx), prod)
100    tsum = torch.tensor(sum_result).to(prod_overF.device)
101
102    if P != 0:
103        # remove side effect from padding
104        return tsum[:, :, P:-P, P:-P]
105
106    return tsum

```

---

รายการ 6.8: ตัวอย่างโปรแกรมชั้นตอนโวลูชันที่เขียนการแพร่กระจายย้อนกลับของ MyConv2DB ซึ่งกลไกการคำนวณจริงทำผ่านฟังก์ชัน convf ที่นิยามในรายการ 6.7.

---

```

1 class MyConv2DB(MyConv2D):
2     def __init__(self, input_channels, num_kernels,
3                  kernel_size, stride=1, padding=0):
4         super(MyConv2DB, self).__init__(input_channels,
5                                         num_kernels, kernel_size, stride, padding)
6         self.convf = convf.apply
7
8     def forward(self, z):
9         a = self.convf(z, self.weight, self.bias,
10                      self.stride, self.padding)
11        return a

```

---

จงทดสอบชั้นคำนวณ MyConv2DB ทั้งในเชิงผลการทำงาน และประสิทธิภาพการทำงาน (วัดเวลาทำงาน) รวมถึงทดสอบว่า การแพร่กระจายย้อนกลับทำผ่าน `convf.backward` จริง (ดูแบบฝึกหัด 6.8 ประกอบ). แล้วเปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.Conv2d`.

หมายเหตุ การเขียนโปรแกรมเองในที่นี้เพื่อความกระจ่างในการทำงาน แต่ในทางปฏิบัติ แนะนำให้ใช้โปรแกรมสำเร็จรูป ด้วยเหตุผลด้านความสะดวก ประสิทธิภาพ การทดสอบที่ดีและครอบคลุมกว่า รวมถึงความยอมรับและความไว้วางใจของผู้เกี่ยวข้อง.

## แบบฝึกหัด 6.10

แบบฝึกหัดนี้ศึกษาการเขียนโปรแกรมชั้นดึงรวมแบบมากที่สุด ทั้งการคำนวณ และการแพร่กระจายย้อนกลับ. รายการ 6.10 แสดงตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด ที่เขียนการแพร่กระจายย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `maxpoolf` ที่เขียนดังในรายการ 6.9<sup>7</sup>. การใช้งานชั้นดึงรวม MyMaxpool ก็ทำเช่นเดียวกับโปรแกรมสำเร็จรูป `nn.MaxPool2d` เช่น การเปลี่ยนบรรทัดคำสั่ง `self.pool1 = nn.MaxPool2d(2, 2)` และ `self.pool2 = nn.MaxPool2d(2, 2)` ในรายการ 6.2 เป็น `self.pool1 = MyMaxpool(2, 2, 0)` และ `self.pool2 = MyMaxpool(2, 2, 0)` ตามลำดับ เท่านั้น. ส่วนที่เหลือก็สามารถดำเนินงานสร้างโครงข่าย ฝึก และทดสอบได้เช่นเดิม.

---

รายการ 6.9: ตัวอย่างโปรแกรมการคำนวณชั้นดึงรวมแบบมากที่สุดพร้อมการแพร่กระจายย้อนกลับ `maxpoolf`

```

1 class maxpoolf(torch.autograd.Function):
2     @staticmethod
3     def forward(ctx, zp, Hf=2, Wf=2, S=2, P=0):
4         ...
5         input: zp (N, C, H, W):  $z_{c,i,j}^{(v-1)}$ 
6         output: z (N, C, H', W'):  $z_{c,k,l}^{(v)} = g(\{z_{c,S_H \cdot (k-1)+i, S_W \cdot (l-1)+j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F})$ 
7         ...
8
9         N, C, H, W = zp.shape
10
11        # Determinte output size
12        Ho = int((H + 2*P - Hf)/S) + 1
13        Wo = int((W + 2*P - Wf)/S) + 1
14

```

---

<sup>7</sup> รหัสโปรแกรมนี้ดัดแปลงจาก รหัสโปรแกรมชีปส์ เทอร์เน็ต (Hipsternet), จาก <https://github.com/wiseodd/hipsternet/tree/master/hipsternet>, ปรับปรุงล่าสุด 12 ก.พ. 2017.

```

15      # Restructure zp
16      # An operation effect of pooling is different from conv
17      # such that channel/feature axis is treated independently
18      # (like datapoint axis).
19
20      restrict_z = zp.view(N * C, 1, H, W)
21      sim_z = MyConv2D._simplify_struct(restrict_z, Hf,Wf,S,P)
22      assert sim_z.shape == (Hf * Wf, Ho * Wo * N * C)
23
24      # Perform pooling function
25      # poolz, pool_cache = pool_func(sim_z)
26
27      max_idx = torch.argmax(sim_z, dim=0)
28      poolz = sim_z[max_idx, range(max_idx.size()[0])]
29      pool_cache = max_idx
30
31      # Restructure pooling output
32      zpool = poolz.view(Ho, Wo, N, C)
33      z = zpool.permute(2, 3, 0, 1).contiguous()
34
35      ctx.save_for_backward(zp, torch.tensor([Hf, Wf, S, P]),
36                            sim_z, pool_cache)
37
38      return z
39
40  @staticmethod
41  def backward(ctx, dEz):
42      # input: dEz (N, F, H', W'):  $\hat{\delta}_{frs}^{(v)}$ 
43      # output: dEzp (N, F, H, W):  $\hat{\delta}_{fkl}^{(v-1)} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \hat{\delta}_{frs}^{(v)} \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$ 
44      zp, tensorHfWfSP, sim_z, pool_cache = ctx.saved_tensors
45      Hf = tensorHfWfSP[0].item()
46      Wf = tensorHfWfSP[1].item()
47      S = tensorHfWfSP[2].item()
48      P = tensorHfWfSP[3].item()
49
50      N, F, H, W = zp.shape
51
52      sim_dEa = torch.zeros(sim_z.shape).to(zp.device)
53      sim_dEz = dEz.permute(2, 3, 0, 1).contiguous().view(-1,
54

```

```

55         # Perform dpooling function
56         # sim_dEa = dpool_func(poolz, pool_cache)
57         # sim_dEa:    $\hat{\delta}_{frs}^{(v)} \cdot \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \begin{cases} \delta_{frs}^{(v)} & \text{when } f, k, l \text{ are the max id's} \\ 0 & \text{otherwise} \end{cases}$ 
58
59         sim_dEa[pool_cache, range(pool_cache.size()[0])] =
60             sim_dEz    # (Hf Wf, H' W' N F)
61
62         # Sum over spatial indices
63         dEzp_sim = convf.sum_omega(sim_dEa,
64                                     (N*F, 1, H, W), Hf, Wf, P, S)
65         dEzp = dEzp_sim.view(zp.shape)
66
67         return dEzp, None, None, None, None

```

---

รายการ 6.10: ตัวอย่างโปรแกรมชั้นชั้นดึงรวมแบบมากที่สุดที่เขียนการแพร์กระจายย้อนกลับของ MyMaxpool ซึ่งกลไกการคำนวณจริงทำผ่านฟังก์ชัน `maxpoolf` ที่นิยามในรายการ 6.9.

```

1 class MyMaxpool(nn.Module):
2     def __init__(self, kernel_size, stride, padding=0):
3         super(MyMaxpool, self).__init__()
4         self.maxpoolf = maxpoolf.apply
5         self.Hf = kernel_size
6         self.Wf = kernel_size
7         self.stride = stride
8         self.padding = padding
9
10    def forward(self, zp):
11        z = self.maxpoolf(zp, self.Hf, self.Wf,
12                          self.stride, self.padding)
13        return z

```

---

จงทดสอบชั้นคำนวณ MyMaxpool ทั้งในเชิงผลการทำงาน และประสิทธิภาพการทำงาน (วัดเวลาทำงาน) รวมถึงทดสอบว่า การแพร์กระจายย้อนกลับทำผ่าน `maxpoolf.backward` จริง. แล้วเปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.MaxPool2d`.

หมายเหตุ การเขียนโปรแกรมเองในที่นี้เพื่อความกระจ่างในการทำงาน แต่ในทางปฏิบัติ แนะนำให้ใช้โปรแกรมสำเร็จรูป ด้วยเหตุผลด้านความสะดวก ประสิทธิภาพ การทดสอบที่ดีและครอบคลุมกว่า รวมถึงความยอมรับและความไว้วางใจของผู้เกี่ยวข้อง.



## บทที่ 7

### การเรียนรู้เชิงลึกในโลกการรู้จำทัศนรูปแบบ

``By three methods we may learn wisdom. First, it is by reflection, which is noblest. Second, it is by imitation, which is easiest. And, third, it is by experience, which is the bitterest.'' ---Confucius

“มีสามวิธีที่เราจะเรียนรู้. หนึ่ง ด้วยการคิดพิจารณา ซึ่งสูงส่งที่สุด. สอง ด้วยการเลียนแบบ ซึ่งง่ายที่สุด. สาม ด้วยประสบการณ์ ซึ่งขมขื่นที่สุด.”

—คงจื๊อ

โครงข่ายคอมโวลุชัน เหมาะกับข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง. ในทางปฏิบัติ ข้อมูลหลาย ๆ ชนิด มีลักษณะเชิงท้องถิ่นสูง รวมถึงข้อมูลเชิงทัศนะ เช่น ภาพ และวิดีโอ. โครงข่ายคอมโวลุชันได้รับการประยุกต์ใช้อย่างกว้างขวางกับข้อมูลเชิงทัศนะ ไม่ว่าจะเป็น การรู้จำประเภทของวัตถุหลักในภาพ (image classification[113, 186, 195, 85, 92]), การตรวจจับวัตถุในภาพ (object detection[160, 161, 164]), การตรวจจับท่าทาง (pose detection[28]), การรู้จำใบหน้า (face recognition[179]), การแบ่งส่วนภาพตามความหมาย (semantic segmentation[122]), การบรรยายภาพ (scene description[99, 95, 205]), การเพิ่มความละเอียดให้กับภาพ (enhance image resolution[57, 183, 184]), การซ่อม เสริม และกำเนิดภาพ (image reparation/generation[201, 213, 77, 145]), การจำแนกวิดีโอ (video classification[100]), การติดตามวัตถุ (object tracking[208]) เป็นต้น. การประยุกต์ใช้เหล่านี้ อาศัยความคิดสร้างสรรค์และความเข้าใจในภารกิจ ทฤษฎี กลไกการทำงานของการเรียนรู้ของเครื่อง และในหลาย ๆ ครั้งได้พัฒนาทฤษฎีเฉพาะขึ้นมา. การประยุกต์ใช้ที่น่าสนใจ มีมากมายและยังมีการพัฒนาอย่างต่อเนื่อง บทนี้ เลือกอภิปรายบางส่วนของการประยุกต์ใช้ที่น่าสนใจ เพื่อให้เห็นตัวอย่างของความคิดสร้างสรรค์ในการประยุกต์ใช้โครงข่ายคอมโวลุชัน.

การรู้จำประเภทของวัตถุหลักในภาพ (image classification) การตรวจจับวัตถุในภาพ (object detection)



output: “Baby Kangaroo”



output:  $(x, y, w, h, “Baby Kangaroo”)$

**รูปที่ 7.1:** เปรียบเทียบการรู้จำประเภทของวัตถุหลักในภาพ และการตรวจจับวัตถุในภาพ. ภาพซ้ายมือแสดงผลลัพธ์จากการรู้จำประเภทของวัตถุหลักในภาพ ซึ่งจะระบุแค่ชนิดของวัตถุหลักในภาพ. ภาพขวา มือแสดงผลลัพธ์จากการตรวจจับวัตถุในภาพ ซึ่งนอกจากระบุชนิดของวัตถุแล้วยังต้องระบุตำแหน่งด้วย. กรอบสีเขียวในภาพขวา คือ กล่องของเขต ซึ่งมักถูกระบุด้วยพิกัด (แนวอนันและแนวตั้ง  $x, y$ ) และขนาด (ความกว้างและความสูง  $w, h$ ).

## 7.1 การตรวจจับวัตถุในภาพ

การตรวจจับวัตถุในภาพ (object detection) เป็นภารกิจที่รับอินพุตเป็นภาพ และให้อเอร์พุตเป็นตำแหน่งของวัตถุที่พ�ในภาพ พร้อมชนิดของวัตถุ. โดยทั่วไป ตำแหน่งของวัตถุ จะระบุด้วยกล่องของเขต (bounding box) ซึ่งอาจอ้างอิงถึงด้วยพิกัดแนวอนันและแนวตั้งของจุดศูนย์กลางของกล่องของเขต และความกว้างกับความสูงของกล่องของเขต.

รูป 7.1 แสดงผลลัพธ์ของการตรวจจับวัตถุในภาพ (ภาพซ้าย) เปรียบเทียบกับการรู้จำประเภทของวัตถุหลักในภาพ (image classification ในภาพขวา). การรู้จำประเภทของวัตถุหลักในภาพ เป็นภารกิจที่รับอินพุตเป็นภาพ และให้อเอร์พุตเป็นชนิดของวัตถุหลักในภาพ ส่วนการตรวจจับวัตถุในภาพ จะเพิ่มการระบุตำแหน่งของวัตถุออกมาให้ด้วย.

การรู้จำประเภทของวัตถุหลักในภาพ ไม่มีการระบุตำแหน่งของวัตถุภายในภาพ และมักถูกตีกรอบปัญหาเป็นปัญหาการจำแนกประเภท (multi-class classification). แบบจำลองของการรู้จำประเภทของวัตถุหลักในภาพ นิยมใช้โครงข่ายคอนโวลูชัน ที่โครงสร้างเออร์พุตใช้ชั้นเชื่อมต่อเต็มที่ตามด้วยฟังก์ชันซอฟต์แมกซ์ ดังเช่น แบบจำลองอเล็กซ์เน็ต ที่ได้อภิรายในหัวข้อ 6.5.

การตรวจจับวัตถุในภาพ อาจทำได้หลายวิธี. วิธีแบบดั้งเดิม ใช้การจำแนกประเภท ร่วมกับเทคนิคหน้า-ต่างเลื่อน ดังอภิรายในหัวข้อ 4.1 (หรือเทคนิคอื่นในลักษณะเดียวกัน [164, 69]). หนึ่งในศาสตร์และศิลป์ของการตรวจจับวัตถุในภาพ คือ โยโล (YOLO[160]) ซึ่งเป็นระบบตรวจจับวัตถุในภาพแบบเวลาจริง (real-

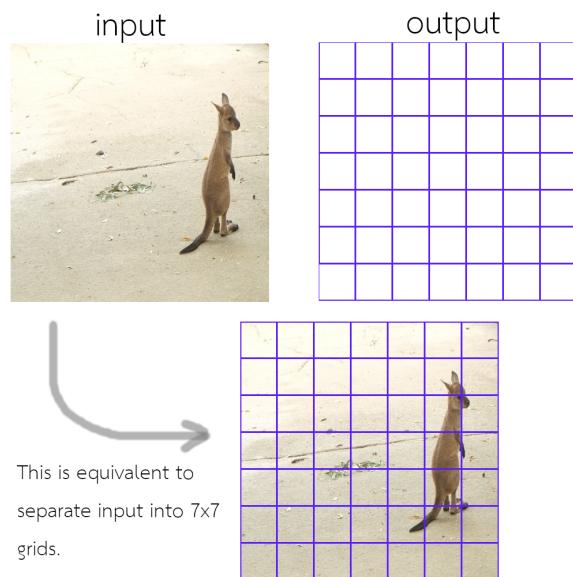
time object detection) และโยโล่ยังมีการทำงานภายในที่ใช้โครงสร้างกลไกหลักอย่างเดียว ทำให้การปรับแต่งประสิทธิภาพสามารถทำได้ง่าย.

**โยโล่.** โยโล่[160, 161]รับอินพุตเป็นภาพสี และให้อาต์พุตออกมาเป็นกล่องของขอบเขตระบุตำแหน่งของวัตถุที่ตรวจจับได้ในภาพ พร้อมชนิดของวัตถุที่พบ. โยโล่ใช้โครงข่ายคอนโวลูชันในการแปลงจากอินพุตไปเป็นอาต์พุต. ต่างจากระบบการตรวจจับวัตถุในภาพแบบดั้งเดิม โยโล่ ตั้งปัญหาการตรวจจับวัตถุในภาพ เป็นการหาค่าตัดตอน เพื่อคำนวณตำแหน่งของวัตถุ และการจำแนกประเภท เพื่อคำนวณชนิดของวัตถุที่พบ.

แนวทางคือ เพื่อสามารถตรวจจับตำแหน่งวัตถุได้สูงสุด  $M$  วัตถุต่อภาพ เราต้องการอาต์พุตเป็นเทนเซอร์ขนาดอย่างน้อย  $5M$  นั่นคือ สำหรับแต่ละการตรวจจับตำแหน่งวัตถุ โยโล่จะใช้ 5 ค่า เพื่อระบุด้วยพิกัดของจุดศูนย์กลางและขนาดของกล่องของขอบเขต ( $x, y, w, h$  สำหรับพิกัดแนวอนและแนวตั้ง ความกว้างและความสูง) พร้อมด้วยค่าความมั่นใจว่าภายในกล่องมีวัตถุอยู่. ค่าความมั่นใจนี้ (confidence ใช้สัญกรณ์  $C$ ) มีเพื่อที่ช่วยให้ผลการทำนายสามารถยืดหยุ่นจำนวนวัตถุในภาพได้ ตั้งแต่ 0 วัตถุ (ทุกตำแหน่งตรวจจับ มีค่าความมั่นใจต่ำมาก) ไปจนถึง  $M$  วัตถุ (ทุกตำแหน่งตรวจจับ มีค่าความมั่นใจสูงมาก). อาจมองได้ว่า ค่าความมั่นใจทำหน้าที่เป็นเหมือนสวิตซ์ปิดเปิดกล่องของขอบเขต ว่าจะเลือกผลลัพธ์กล่องไหนบ้างให้ออกไป.

เพื่อให้การฝึกโครงข่ายทำได้อย่างมีประสิทธิภาพ โยโล่ กำหนดพื้นที่รับผิดชอบของแต่ละตำแหน่งตรวจจับ. การกำหนดพื้นที่รับผิดชอบของตำแหน่งตรวจจับ จำนวน  $M$  ตำแหน่งตรวจจับ เทียบเท่ากับการแบ่งพื้นที่รับผิดชอบของภาพอินพุตออกเป็น  $M$  ส่วน. โยโล่แบ่งพื้นที่ภาพตามแนวอนและแนวตั้งอย่างละเอียด ๆ กัน และเรียกการแบ่งนี้เป็นเสมือนช่องตาราง หรือ กริด (grid) และเรียกพื้นที่รับผิดชอบแต่ละส่วนว่า กริดเซลล์ (grid cell). รูป 7.2 แสดงแนวคิดนี้ ในรูปแสดงการแบ่งรูปออกเป็น  $7 \times 7$  ส่วน ( $M = 49$ ). การกำหนดกริดเซลล์ให้รับผิดชอบพื้นที่อินพุตส่วนไหน ช่วยให้การฝึกทำได้มีประสิทธิภาพมากขึ้น โดยลดความสับสนว่าวัตถุควรจะถูกทายด้วยกริดเซลล์ไหน. มันจะช่วยให้ การกำหนดฟังก์ชันสูญเสีย และการทำนาย ทำได้ง่ายขึ้น เพราะถ้าไม่กำหนดความรับผิดชอบให้แน่นอน กริดเซลล์ใด ๆ หนึ่งใน  $M$  กริดเซลล์ อาจหายวัตถุก็ได้ และการคำนวณค่าฟังก์ชันสูญเสียจะยุ่งยากมาก. (ในระหว่างการฝึก ซึ่งการทายอาจจะยังผิดเพียงอยู่มาก มันจะยกที่จะรู้ว่ากริดเซลล์ไหนที่กำลังทายเฉลยวัตถุไหน และยังอาจมีกรณีที่ กริดเซลล์มากกว่าหนึ่งตัว พยายามทายวัตถุเดียวกันอีก.)

การกำหนดให้แต่ละกริดเซลล์ทายได้เพียงหนึ่งวัตถุ จะจำกัดความสามารถของการตรวจจับภาพวัตถุ โดยเฉพาะกรณีที่วัตถุซ้อนทับกันและมีจุดศูนย์กลางอยู่ใกล้กัน (ทำให้วัตถุตกลอยู่ในความรับผิดชอบของกริดเซลล์เดียวกัน). หลาย ๆ ครั้ง วัตถุที่ซ้อนทับกันนั้น อาจมีขนาดหรือรูปทรงที่แตกต่างกัน ทำให้ แม้จะทับซ้อนกัน ก็



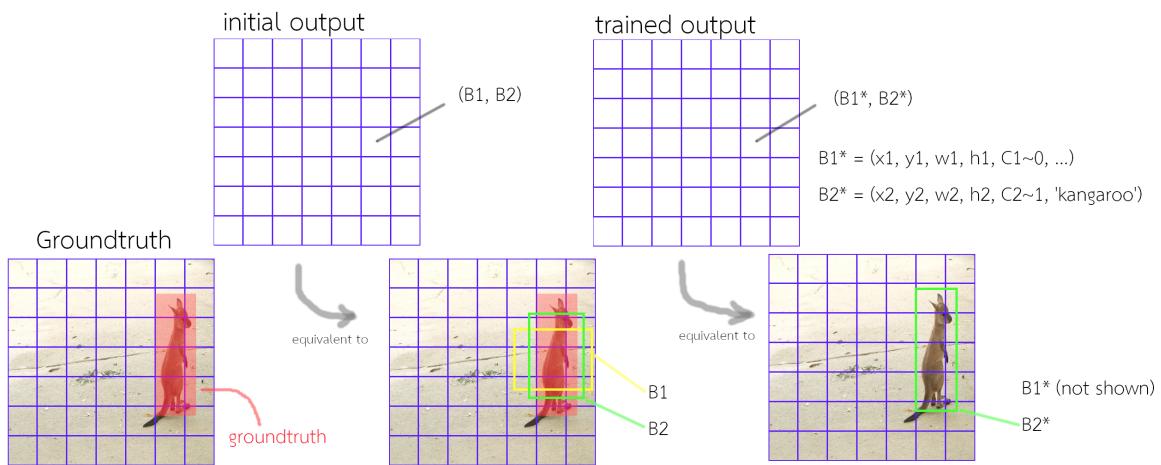
รูปที่ 7.2: ໂປໂລໃใช້ເອາະພູທີ່ມີໂຄຮງສ້າງເປັນ  $G \times G$  ສ່ວນ ຜຶ່ງຕ້ວຍໆຢັ້ງນີ້ຂຶ້ນ  $7 \times 7$  ແລະ ກຳນົດໃຫ້ແຕ່ລະສ່ວນແບ່ງຂອບເຂດຮັບຜິດຂອບ ອືນພູຫ ຜຶ່ງເທິຍບ່າງເກົ່າການແບ່ງສ່ວນກາພອືນພູຫເປັນ  $7 \times 7$  ສ່ວນ (ໂປໂລໄເຮີຍກ ແຕ່ລະສ່ວນວ່າ ກຣິດເຊີລ໌). ແຕ່ສ່ວນຖຸກຮັບຜິດຂອບດ້ວຍແຕ່ລະ ກຣິດເຊີລ໌ ໂດຍໜຶ່ງກຣິດເຊີລ໌ (ໜຶ່ງ“ພິກເຊລ໌”ໃນເອາະພູ) ຈະມີຄ່າຕ່າງໆ ຈຳກັດຕັ້ງ ໃນການກວດສັບສົນ ແລະ ຊື່ນິດວັດຖຸ.



รูปที่ 7.3: ຕ້ວຍໆຢັ້ງແສດງຮຽນສໍາຫຼັບເຫັນວ່າມີຄ່າຕ່າງໆ ໃນກາພ ນັກດຳນັ້ນທີ່ອີ່ງດ້ານໜ້າ ແລະ ຂລາມທີ່ອີ່ງດ້ານໜ້າ ມີຈຸດຄູນຍົກລາງຂອງ ກລ່ອງຂອບເຂດອີ່ງຕໍ່ມາແນ່ງເຕີຍວັນ. ທັກກຳນົດໃຫ້ການທາຍວັດຖຸສາມາດຮັດໄດ້ເພີ່ງທີ່ນີ້ວັດຖຸຕ່ອງກຣິດເຊີລ໌ ຄວາມສາມາດຮັດຂອງຮະບບ ຈະຖຸກຈຳກັດຍ່າງມາກໃນກຣິນີ້ເຂັ້ນນີ້. ສັງເກດວ່າ ແມ່ວັດຖຸຈະທັບໜັນກັນແລະ ມີຈຸດຄູນຍົກລາງເດືອຍວັນແຕ່ໄມ້ໄດ້ບັນກັນສັນຫິ ເນື່ອຈາກວັດຖຸ ຕ່າງໆ ທີ່ທັບໜັນກັນ ຈະມີຂັາດຫີ່ວັນທີ່ແຕກຕ່າງກັນ. ກລ່ອງຂອບເຂດໃນກາພ ແສດງການທັບໜັນແລະ ກວາມມີຈຸດຄູນຍົກລາງໃກ້ເດີຍກ ກັນ(ອາຈະສັງເກດໄດ້ຍາກ) ແຕ່ມີຂັາດແລະ ຮູ່ປຽງແຕກຕ່າງກັນຂອງວັດຖຸທີ່ທັບໜັນກັນ (ກລ່ອງສື່ເສີວາ ຕຽບຈັບນັກດຳນັ້ນ ສ່ວນກລ່ອງສື່ເຫຼືອງ ຕຽບຈັບຂລາມ).

ສາມາດເຫັນວັດຖຸຕ່າງໆ ທີ່ທັບໜັນກັນໄດ້ຢັ້ງຢັ້ງ. ຮູບ 7.3 ແສດງຕ້ວຍໆຢັ້ງຮຽນດັ່ງກ່າວ.

ວິທີແກ້ໄຂເບື້ອງຕັ້ນຂຶ້ນ ແກ່ນທີ່ຈະໃຫ້ແຕ່ລະ ກຣິດເຊີລ໌ ທໍານາຍຕໍ່ມາແນ່ງວັດຖຸໄດ້ແຕ່ໜຶ່ງອັນ ກີ່ແກ່ອນຸ້າຕີໃຫ້ໜຶ່ງ ກຣິດເຊີລ໌ ທໍານາຍຕໍ່ມາແນ່ງວັດຖຸໄດ້ໜ້າຍ ຖໍ່ມາແນ່ງ ໂດຍແຕ່ລະ ການທາຍກີ່ມີຄ່າຄວາມມັນໃຈຂອງຕົວເວັງ. ເພີ່ງແຕ່ ໃນ ການຝຶກ ການຄໍາວຸນຄ່າ ພັກໜັນສູງເສີຍຈະຈັດກາໄໝມີປະສິທິກາພໄດ້ຢັ້ງໄຣ. ວິທີການທີ່ອົກແບບມາເພື່ອບຣເທາ



**รูปที่ 7.4:** โยโล่เลือกกล่องสมอเพื่อรับผิดชอบวัตถุ. ภาพซ้ายสุด แควร์ล่าง แสดงภาพอินพุต และส่วนรับผิดชอบต่าง ๆ พร้อมทั้งเฉลย (แสดงด้วยพื้นที่เปร่งใสสีแดง). ในภาพ สังเกต จุดศูนย์กลางของเฉลย จะตอกอยู่ภายใต้กริดเซลล์ที่หกจากซ้ายและสี่จากบน และโยโล่จะใช้กริดเซลล์นี้รับผิดชอบเฉลย. ภาพซ้าย แควร์บน แสดงอาต์พุตขณะเดือนฝึก ที่แต่ละกริดเซลล์จะเป็นเวกเตอร์ที่นำมายกกล่องสมอสองกล่อง (**B1** และ **B2** ในกริดเซลล์ที่รับผิดชอบเฉลย). กล่องสมอแต่ละตัวประกอบด้วยค่าตำแหน่งและขนาดกล่องของเขต ค่าความมั่นใจ และชนิดวัตถุ (ชนิดวัตถุ ระบุด้วยค่าความน่าจะเป็นของชนิดต่าง ๆ เต็รีมไว้). ภาพล่างกลาง แสดงตัวอย่างของกล่องสมอก่อนเริ่มฝึก เมื่อเทียบกับเฉลย. กล่องสมอ ภายในกริดเซลล์เดียวกัน จะถูกกำหนดค่าเริ่มต้นให้มีขนาดหรือรูปทรงแตกต่างกัน. ขนาดหรือรูปทรงที่แตกต่างกัน ทำให้ค่าไอยูรุ่ห่วงกล่องสมอต่าง ๆ กับเฉลย ต่างกัน และสามารถใช้เป็นตัวชี้วัดได้ เพื่อกำหนดความรับผิดชอบได้. ในภาพ ไอยูรุ่ห่วง (ซึ่งคือ สัดส่วนทับซ้อน) ระหว่างกล่องสมอ **B2** กับเฉลย มีค่ามากกว่า ค่าของกล่องสมอ **B1** กับเฉลย. ดังนั้น ในกระบวนการฝึก โยโล่จะกำหนดให้ กล่องสมอ **B2** รับผิดชอบเฉลย. ภาพบนขวา แสดงตัวอย่างอาต์พุตหลังฝึกเสร็จ (กริดเซลล์ที่ดูแลเฉลย เป็นลี่นค่าเป็น **B1\*** และ **B2\***) โดยเมื่อการฝึกสมบูรณ์ ค่าความมั่นใจของกล่องสมอที่ไม่ได้รับผิดชอบเฉลยได้ (**C1**) จะใกล้กับศูนย์ และค่าความมั่นใจของกล่องสมอที่ทำงาน (**C2**) จะใกล้กับหนึ่ง. ภาพขวาล่าง แสดงผลลัพธ์ เมื่อนำไปคาดหัว กับอินพุต. สังเกตว่า กล่องสมอที่รับผิดชอบเฉลย จะปรับขนาดและรูปทรงตามเฉลย. ตัวอย่างนี้ กริดเซลล์มีเพียงเฉลยเดียว จึงมีเพียงกล่องสมอเดียวที่ถูกใช้. หากกริดเซลล์รับผิดชอบวัตถุสองวัตถุทับซ้อนกัน กล่องสมอทั้งสองก็จะถูกใช้งาน และเลือกจับคู่โดยอาศัยค่าไอยูรุ่ห่วงเป็นตัวชี้วัด.

ประเด็นนี้ คือ เทคนิคกล่องสมอ (anchor box[164]).

**เทคนิคกล่องสมอ.** การทายตำแหน่งแต่ละกล่อง ในกริดเซลล์ จะเรียกว่า กล่องสมอ โดยหนึ่งกริดเซลล์ สามารถทำนายกล่องสมอได้  $B$  กล่อง และ กล่องสมอแต่ละกล่อง จะถูกกำหนดค่าเริ่มต้นให้มีขนาดหรือสัดส่วนต่างกัน. การฝึก จะใช้ค่าไอยูรุ่ห่วงกล่องสมอกับเฉลย เป็นตัวชี้วัดว่า กล่องสมอใดจะรับผิดชอบเฉลยวัตถุใด (และอาจมีกฎในการกำหนดความรับผิดชอบในกรณีที่ค่าไอยูรุ่ห่วงเท่ากัน). รูป 7.4 แสดงการใช้งานเทคนิคกล่องสมอในโยโล่.

สำหรับการตรวจจับวัตถุในภาพได้ครอบคลุม  $K$  ชนิดวัตถุ แต่ละกล่องสมอจะมี  $5 + K$  ค่า สำหรับตำแหน่งและขนาดของกล่องของเขต ( $x, y, w, h$ ), ค่าความมั่นใจว่าวัตถุอยู่ภายใต้กริดเซลล์  $C$ , และค่าความน่าจะเป็นของวัตถุแต่ละชนิด  $p(1), \dots, p(K)$ . ดังนั้น สำหรับ  $M$  กริดเซลล์ และ  $B$  กล่องสมอต่อกริดเซลล์ แล้ว เอาต์พุตของโยโล่  $\mathbf{Y} \in \mathbb{R}^{M \cdot B \cdot (5+K)}$  หรือ  $\mathbf{Y} = [\tilde{x}_{mb}, \tilde{y}_{mb}, \tilde{w}_{mb}, \tilde{h}_{mb}, \hat{C}_{mb}, \hat{p}_{mb}(1), \dots, \hat{p}_{mb}(K)]$

สำหรับ  $m = 1, \dots, M; b = 1, \dots, B$ .

เพื่อให้การฝึกแบบจำลองทำได้มีประสิทธิภาพมากขึ้น แทนที่จะให้แบบจำลองทำนายค่าพิกัดและขนาดของกล่องขอบเขตโดยตรง ค่าที่แบบจำลองทำนาย  $\tilde{x}_{mb}, \tilde{y}_{mb}, \tilde{w}_{mb}, \tilde{h}_{mb}$  จะถูกคำนวณไปเป็นค่าพิกัดและขนาดของกล่องขอบเขต ดังนี้ (ละตัวห้อยออก เพื่อความกระชับ)

$$\hat{x} = c_w \cdot \sigma(\tilde{x}) + c_x \quad (7.1)$$

$$\hat{y} = c_h \cdot \sigma(\tilde{y}) + c_y \quad (7.2)$$

$$\hat{w} = p_w \cdot \exp(\tilde{w}) \quad (7.3)$$

$$\hat{h} = p_h \cdot \exp(\tilde{h}) \quad (7.4)$$

เมื่อ  $\hat{x}$  กับ  $\hat{y}$  เป็นพิกัดแนวอนกับแนวตั้งของศูนย์กลางกล่องขอบเขตที่ทำนาย และ  $\hat{w}$  กับ  $\hat{h}$  เป็นความกว้างกับความสูงของกล่องขอบเขตที่ทำนาย โดย  $\sigma(\cdot)$  คือฟังก์ชันซิกมอยด์ (มีค่าระหว่างศูนย์ถึงหนึ่ง),  $c_x$  กับ  $c_y$  เป็นพิกัดมุมซ้ายบนของกริดเซลล์,  $c_w$  กับ  $c_h$  เป็นความกว้างกับความสูงของกริดเซลล์, และ  $p_w$  กับ  $p_h$  เป็นค่าฐานของความกว้างกับความสูงของกล่องสมอ.

สังเกตว่า ถ้า  $\tilde{x}$  มีค่าบวกขนาดใหญ่มาก จะทำให้  $\hat{x}$  อยู่ขอบขวาของกริดเซลล์. ถ้า  $\tilde{x}$  มีค่าลบขนาดใหญ่มาก จะทำให้  $\hat{x}$  อยู่ขอบซ้ายของกริดเซลล์. ถ้า  $\tilde{x}$  มีค่าเป็นศูนย์ จะทำให้  $\hat{x}$  อยู่ตรงกลางของกริดเซลล์. ความสัมพันธ์ระหว่างค่า  $\tilde{y}$  กับ  $\hat{y}$  ก็เป็นในทำนองเดียวกัน. ส่วนถ้า  $\tilde{w}$  มีค่าบวก จะทำให้  $\hat{w}$  กว้างกว่าค่าฐาน  $p_w$ . ถ้า  $\tilde{h}$  มีค่าลบ จะทำให้  $\hat{h}$  แคบกว่าค่าฐาน  $p_h$ . ถ้า  $\tilde{w}$  มีค่าศูนย์ จะทำให้  $\hat{w}$  กว้างเท่ากับค่าฐาน  $p_w$ . ความสัมพันธ์ระหว่างค่า  $\tilde{h}$  กับ  $\hat{h}$  ก็เป็นในทำนองเดียวกัน.

ค่าฐานของแต่ละกล่องสมอ  $p_w$  และ  $p_h$  อาจเลือกกำหนดเองตามเห็นว่าเหมาะสม. คณะของเรดมอน[161] ใช้การจัดกลุ่มข้อมูลด้วยวิธีเค-มีนส์ (K-means) สำรวจกล่องขอบเขตเฉลยของข้อมูลฝึกหัด แล้วใช้ค่าเซนทรอยด์ต่าง ๆ (centroids) ที่ได้มา เป็นค่าฐานของกล่องสมอต่าง ๆ.

โดยโล้ นิยาม ค่าความมั่นใจ  $\hat{C} = \text{Pr}(\text{Object}) \cdot \text{IOU}$  เมื่อ  $\text{Pr}(\text{Object})$  แทนค่าความน่าจะเป็นที่กล่องขอบเขตจะมีวัตถุ และ  $\text{IOU}$  แทนค่าไอโอ喻ระหว่างกล่องขอบเขตที่ทายกับกล่องขอบเขตเฉลย. ดังนั้นค่าความมั่นใจเฉลย  $C = 0$  ถ้าไม่มีวัตถุอยู่ภายในกริดเซลล์ และ  $C = \text{IOU}$  ถ้ามีวัตถุอยู่ภายในกริดเซลล์.

ในการฝึกการตรวจจับวัตถุในภาพ โดยโล่กำหนดฟังก์ชันสูญเสียดังนี้

$$\begin{aligned}
 \text{loss} = & \lambda_{\text{coord}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \left( (\hat{x}_{mb} - x_{mb})^2 + (\hat{y}_{mb} - y_{mb})^2 \right) \\
 & + \lambda_{\text{coord}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \left( (\sqrt{\hat{w}_{mb}} - \sqrt{w_{mb}})^2 + (\sqrt{\hat{h}_{mb}} - \sqrt{h_{mb}})^2 \right) \\
 & + \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot (\hat{C}_{mb} - C_{mb})^2 + \lambda_{\text{noobj}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{noobj}} \cdot (\hat{C}_{mb} - C_{mb})^2 \\
 & + \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \sum_{k \in \text{classes}} \cdot (\hat{p}_{mb}(k) - p_{mb}(k))^2
 \end{aligned} \tag{7.5}$$

เมื่อ  $p_{mb}(k)$  คือเฉลยชนิดวัตถุที่กริดเซลล์  $m$  กล่องสมอ  $b$  โดย  $p_{mb}(k) = 1$  ถ้าที่กล่องสมอ มีภาพวัตถุชนิด  $k$  และ  $p_{mb}(k) = 0$  ถ้าที่กล่องสมอ มีภาพวัตถุชนิดอื่น. สัญกรณ์  $\mathbf{1}_{mb}^{\text{obj}}$  ใช้ระบุว่ากล่องสมอ  $b$  ในกริดเซลล์  $m$  รับผิดชอบการทาย นั่นคือ  $\mathbf{1}_{mb}^{\text{obj}} = 1$  เมื่อ กล่องสมอ  $b$  ในกริดเซลล์  $m$  มีเฉลยที่รับผิดชอบอยู่ ถ้าไม่อย่างนั้นให้  $\mathbf{1}_{mb}^{\text{obj}} = 0$ . โดยเลือกกำหนดการรับผิดชอบของกล่องสมอ โดยให้กล่องสมอที่มีค่าໄວอยู่ร่วมกับกรอบตัวอย่างเฉลยมากที่สุด ทำหน้าที่รับผิดชอบการทายเฉลยนั้น. สัญกรณ์  $\mathbf{1}_{mb}^{\text{noobj}}$  ใช้ระบุว่ากล่องสมอ  $b$  ในกริดเซลล์  $m$  ไม่มีวัตถุอยู่ ซึ่ง  $\mathbf{1}_{mb}^{\text{noobj}} = 1 - \mathbf{1}_{mb}^{\text{obj}}$ .

เนื่องจาก คณะของเรดมอน[160] พบร่วมกับภาพต่าง ๆ ที่ใช้ฝึก ส่วนใหญ่มีวัตถุอยู่ไม่มาก. กล่องสมอส่วนใหญ่ไม่มีวัตถุ และสัดส่วนการไม่มีวัตถุต่อการมีวัตถุสูงมาก (ข้อมูลไม่สมดุลย์ unbalanced data. แบบฝึกหัด 3.17). ดังนั้น คณะของเรดมอน เลือกใช้แนวทางหนึ่งที่นิยมใช้บรรเทาปัญหา เช่นนี้ คือใช้ค่านำหนักที่ต่างกันเพื่อชดเชย. ค่า  $\lambda_{\text{coord}}$  และ  $\lambda_{\text{noobj}}$  เป็นเพียงเทคนิคเชิงเลข เพื่อชดเชยความไม่สมดุลย์ของข้อมูล (ซึ่งคณะของเรดมอน เลือกใช้  $\lambda_{\text{coord}} = 5$  และ  $\lambda_{\text{noobj}} = 0.5$ ).

สังเกตว่า การคำนวณค่าผิดพลาดของความกว้างและความสูง ทำผ่านค่ารากที่สอง. เนื่องจาก ค่าผิดพลาดสัมบูรณ์ ของการทำนายขนาดสำหรับกล่องขอบเขตขนาดเล็ก แม้ตัวเลขจะเท่ากับค่าผิดพลาดสัมบูรณ์ ของการทำนายขนาดสำหรับกล่องขอบเขตขนาดใหญ่ แต่ถือเป็นความผิดพลาดที่รุนแรงกว่า. ตัวอย่างเช่น การทายความกว้างผิดไป 10 สำหรับความกว้าง 500 นั้นถือว่าเล็กน้อยมาก จนผู้ใช้อาจไม่ได้เห็นความแตกต่างแต่ การทายความกว้างผิดไป 10 สำหรับความกว้าง 5 นั้นถือว่าผิดพลาดรุนแรงมาก และผลลัพธ์ก็เห็นได้อย่างชัดแจ้ง. คณะของเรดมอน[160] ใช้เทคนิคเชิงเลข โดยคำนวณความแตกต่างของค่ารากที่สองแทน เพื่อช่วยบรรเทาปัญahanี้.

ในกระบวนการฝึก คณะของเรดมอน[160] ใช้การฝึกก่อน โดยฝึกแบบจำลองกับงานจำแนกชนิดวัตถุ

หลักในการพก่อนจนแบบจำลองทำงานได้ดีแล้ว. จากนั้นจึงเพิ่มชั้นคำนวนห้าย ๆ (ด้านเอาต์พุต) เข้าไปแล้วจึงฝึกแบบจำลองสำหรับภาระกิจการตรวจจับวัตถุในภาพ.

หลังจากฝึกเสร็จ ในการงานอนุมาน ค่าเอาต์พุตจะถูกนำมาประมวลผล โดยค่าของกล่องสมอที่  $\hat{C} > \tau$  จะถูกนำมาคำนวนตำแหน่งและขนาดของกล่องขอบเขต (สมการ 7.1, 7.2, 7.3, 7.4) และวัตถุจะถูกอนุมานเป็นชนิด  $k^* = \arg \max_k \hat{p}(k)$  เมื่อ  $\tau$  เป็นระดับค่าขีดแบ่งที่กำหนด.

## 7.2 การซ้อม เสริม และก่อกำเนิดภาพ

การซ้อมภาพ คือการเติมส่วนของภาพที่ต้องการ (ส่วนของภาพที่เสียหาย) โดยคำนึงถึงบริเวณรอบข้าง และลักษณะของภาพโดยรวม. การเสริมภาพ มีความหมายครอบคลุมการเพิ่มความละเอียดให้กับภาพ (หัวข้อ 7.2). การก่อกำเนิดภาพ คือการสร้างภาพขึ้นมาใหม่ทั้งภาพ โดยภาพที่สร้างขึ้นเป็นภาพในลักษณะที่ต้องการ เช่น ดูคล้ายภาพจริง (หัวข้อ 7.2).

การซ้อม เสริม และก่อกำเนิดภาพ เป็นศาสตร์ที่กำลังได้รับความสนใจและมีการพัฒนาอย่างรวดเร็ว มีหลายแนวทาง เช่น พิกเซลอาร์เอนเนอน (PixelRNN[201]), ตัวเข้ารหัสอัตโนมัติแบบเปลี่ยนแปลง (Variational Autoencoder[213]), หรือโครงข่ายปรับแก้เชิงสร้าง (Generative Adversarial Network[77, 145]).

ความท้าทายที่สำคัญสำหรับภาระกิจเช่นนี้ โดยเฉพาะการก่อกำเนิดภาพ คือ ตัวภาระกิจเป็นเสมือนการเรียนรู้ความน่าจะเป็นของภาพ (ไม่ว่าจะเรียนรู้ชัดแจ้งโดยตรง ซึ่งได้ค่าฟังก์ชันความหนาแน่นความน่าจะเป็นของภาพ หรือโดยนัย ซึ่งคือทำการกิจได้ แต่ไม่ได้ค่าฟังก์ชันความหนาแน่นความน่าจะเป็น). จากมุขของปริภูมิมิติ ภาพเป็นจุดข้อมูลที่อยู่ในปริภูมิหลายมิติ ที่มีจำนวนมิติมหาศาล<sup>1</sup>. นั่นคือ ภาพสีขนาด  $W \times H$  (สัญกรณ์  $\mathbf{X} \in \mathbb{R}^{3 \times W \times H}$ ) แต่ละภาพเปรียบเสมือนจุดหนึ่งจุดในปริภูมิ  $3 \cdot W \cdot H$  มิติ. การประมาณฟังก์ชันความหนาแน่นความน่าจะเป็น  $p(\mathbf{X})$  ทำได้ยากมาก และต้องการข้อมูลจำนวนมหาศาล.

โครงข่ายปรับแก้เชิงสร้าง จัดเป็นศาสตร์และศิลป์ที่สำคัญของการเรียนรู้ของเครื่อง ดังอภิรายเกรินในบทที่ 5 และ ได้แสดงให้เห็นว่า โครงข่ายปรับแก้เชิงสร้างเป็นแนวทางที่ช่วยแก้ปัญหาของภาระกิจการก่อกำเนิดภาพได้. หัวข้อ 7.2 อภิรายโครงข่ายปรับแก้เชิงสร้าง รวมถึงอุปสรรคความท้าทายในการประยุกต์ใช้โครงข่ายปรับแก้เชิงสร้าง และแนวทางในการบรรเทาอุปสรรค.

<sup>1</sup>ลักษณะภาระกิจที่การดำเนินการทำได้ไม่ยาก เมื่อมิติของปริภูมิมีจำนวนน้อย แต่ทำได้ยากมาก หรือไม่อาจทำได้เลยในทางปฏิบัติ หากมิติของปริภูมิมีจำนวนมากมาก มักถูกอ้างอิงถึงว่าเมื่อมีเป็น คำสาปของมิติ (curses of dimensionality).

## การเพิ่มความละเอียดให้กับภาพ

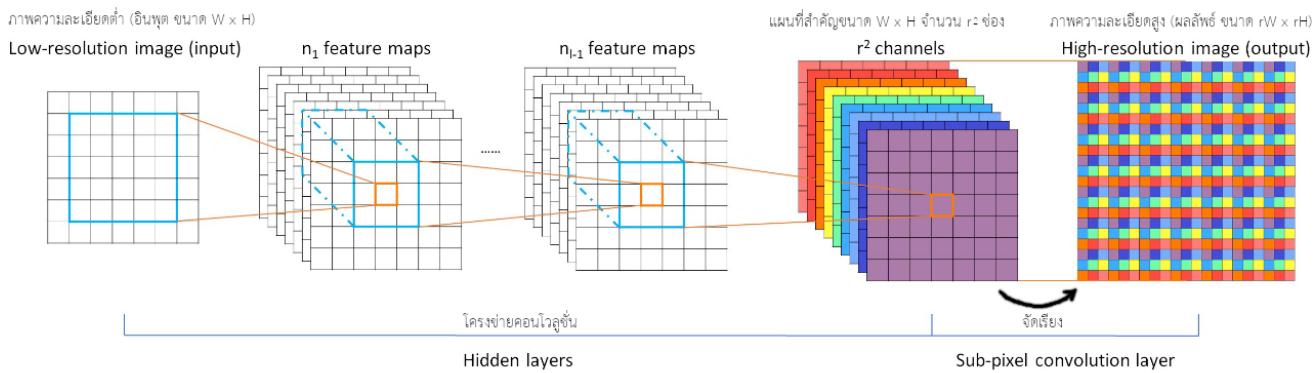
การเพิ่มความละเอียดให้กับภาพ คือ กระบวนการที่รับภาพความละเอียดต่ำ (low-resolution image) และประมาณภาพความละเอียดสูง (high-resolution) ออกมานา.

การเพิ่มความละเอียดให้กับภาพ อาจทำได้หลายวิธี. คณะของตง[57] ทำการอัปแซมบลิ่ง (upsampling) ซึ่งคือการเพิ่มพิกเซลเข้าไปในภาพ โดยค่าพิกเซลที่เพิ่มขึ้นจะได้จากการทำการประมาณค่าในช่วงแบบไบคิวบิก (bicubic interpolation). จากนั้นใช้โครงข่ายคอนโวลูชันในการประมาณภาพความละเอียดสูงออกมา (เพื่อปรับปรุงคุณภาพจากการประมาณค่าในช่วงแบบไบคิวบิก).

กล่าวคือ จากภาพความละเอียดต่ำ  $\tilde{\mathbf{X}}$  คณะของตงสร้างภาพความละเอียดสูง  $\mathbf{X}'$  ขึ้นด้วยวิธีการประมาณค่าในช่วงแบบไบคิวบิก และใช้  $\mathbf{X}'$  เป็นอินพุตของโครงข่ายคอนโวลูชัน เพื่อทำนาย  $\hat{\mathbf{X}}$  (ซึ่ง แม้จะความละเอียดเท่ากัน แต่  $\hat{\mathbf{X}}$  มีคุณภาพดีกว่า  $\mathbf{X}'$ ). หาก  $f$  คือฟังก์ชันที่แทนการคำนวณของโครงข่าย และ  $\Theta$  เป็นค่าพารามิเตอร์ต่าง ๆ ของโครงข่าย โครงข่ายคอนโวลูชันถูกฝึกให้ทำนาย  $\hat{\mathbf{X}} = f(\mathbf{X}'; \Theta)$  ให้ใกล้เคียงกับเฉลย (ที่เป็นภาพความละเอียดสูง)  $\mathbf{X}$  ให้มากที่สุด. นั่นคือ ฟังก์ชันสูญเสีย loss( $\Theta$ ) =  $\frac{1}{N} \sum_{n=1}^N \|f(\mathbf{X}'_n; \Theta) - \mathbf{X}_n\|^2$  เมื่อ  $N$  คือจำนวนข้อมูลฝึกทั้งหมด.

คุณภาพของการเพิ่มความละเอียดให้กับภาพ อาจประเมินจาก ค่าผิดพลาดกำลังสองเฉลี่ย เช่นเดียวกับภาระกิจการหาค่าเฉลี่ยหัวใจที่ว่าไป เช่น mse =  $\frac{1}{W \cdot H} \sum_i \sum_j (\hat{x}_{i,j} - x_{i,j})^2$  เมื่อ  $W$  กับ  $H$  เป็นความกว้างกับสูงของภาพ และ  $\hat{x}_{i,j}$  เป็นค่าพิกเซลของภาพที่เพิ่มความละเอียดขึ้นจากภาพความละเอียดต่ำ  $\tilde{\mathbf{X}}$ . โดย ภาพความละเอียดต่ำ  $\tilde{\mathbf{X}}$  เป็นภาพที่ถูกลดความละเอียดลงจากภาพความละเอียดสูง  $\mathbf{X}$ . ภาพความละเอียดสูง  $\mathbf{X}$  มีค่าพิกเซลต่าง ๆ เป็น  $x_{i,j}$  และ  $i$  กับ  $j$  คือ ดัชนีแนวอนกับแนวตั้งของภาพ. อย่างไรก็ได้ แม้ค่าผิดพลาดกำลังสองเฉลี่ยพอใช้งานได้ แต่นักวิจัยต่างพบว่า ค่าผิดพลาดกำลังสองเฉลี่ยไม่สัมพันธ์กับคุณภาพของภาพที่คนรับรู้[211]. คุณภาพของภาพจึงมีกว่าด้วยดัชนีอื่น ๆ ได้แก่ อัตราส่วนสัญญาณสูงสุดต่อสัญญาณรบกวน (peak signal-to-noise ratio คำย่อ PSNR[210]), ความคล้ายคลึงเชิงโครงสร้าง (structural similarity คำย่อ SSIM[210]), เงื่อนไขความเที่ยงตรง (fidelity criterion คำย่อ IFC[182]), มาตรวัดคุณภาพสัญญาณรบกวน (noise quality measure คำย่อ NQM[50]), อัตราส่วนสัญญาณสูงสุดปรับค่าน้ำหนักต่อสัญญาณรบกวน (weighted peak signal-to-noise ratio คำย่อ WPSNR[211]), หรือ ดัชนีความคล้ายคลึงเชิงโครงสร้างหลายสเกล (multi-scale structure similarity index คำย่อ MSSSIM[211]) เป็นต้น.

ต่างจากการของตงและคณะ[57] คณะของชีอ[183, 184] ใช้โครงข่ายคอนโวลูชัน เพื่อเพิ่มความละเอียดให้กับภาพ โดยรับอินพุตเป็นภาพความละเอียดต่ำโดยตรง และให้อาร์พุตสุดท้ายออกมาเป็นภาพความละเอียดต่ำโดยตรง และให้อาร์พุตสุดท้ายออกมาเป็นภาพความละเอียดต่ำโดยตรง.



รูปที่ 7.5: การขยายความละเอียดภาพด้วยชั้นดีคอนโวโลวูชั้น (ดัดแปลงจากซือและຄณะ[183])

เอียดสูงได้เลย. การใช้โครงข่ายดีคอนโวโลวูชั้นกับภาพความละเอียดต่ำโดยตรง ช่วยลดภาระการคำนวณลงไปได้มาก และຄณะของซือ ยังแสดงให้เห็นคุณภาพของผลลัพธ์ที่ดีขึ้นด้วย.

กลไกสำคัญที่ซือและຄณะใช้ อยู่ที่ชั้นคำนวนท้ายสุด. สำหรับภาพขนาด  $W \times H$  (ช่องสีเดียว<sup>2</sup>) และต้องการขยายความละเอียดขึ้น  $r$  เท่า (นั่นคือ ภาพจะถูกขยายเป็น ภาพผลลัพธ์ขนาด  $rW \times rH$ ) คณะของซือ ออกแบบโครงข่ายดีคอนโวโลวูชั้นที่ให้อิเอาต์พุตออกมานะ เป็นแผนที่ลักษณะสำคัญขนาดเท่ากับขนาดภาพอินพุต แต่มีจำนวนแผนที่เท่ากับ  $r^2$ . และภาพผลลัพธ์ จะสร้างขึ้นจากการจัดเรียงแผนที่ลักษณะสำคัญขนาด  $W \times H$  จำนวน  $r^2$  แผ่นที่ ให้เป็นแผนที่เดียว ขนาด  $rW \times rH$  ซึ่งคือภาพความละเอียดสูงที่ต้องการ. กลไกของการจัดเรียงผลลัพธ์แผนที่สำคัญนี้เรียกว่า กลไกพิกเซลร่ายอย ซึ่งเป็นรูปหนึ่งของชั้นดีคอนโวโลวูชั้น (deconvolution layer[224]). ชั้นดีคอนโวโลวูชั้น อาจทำได้หลายรูปแบบ ซือและຄณะ[184] แจกแจงและอภิปรายข้อแตกต่างของชั้นดีคอนโวโลวูชั้นแบบต่าง ๆ. รูป 7.5 แสดงจุดเด่น (รับอินพุตเป็นภาพความละเอียดต่ำ และให้อิเอาต์พุตเป็นภาพความละเอียดสูงได้โดยตรง) และกลไกสำคัญ (ชั้นดีคอนโวโลวูชั้น ที่จัดเรียงแผนที่ลักษณะสำคัญ  $r^2$  แผนที่ เป็นผลลัพธ์ ภาพเดียวที่เป็นมีความละเอียดเพิ่มเป็น  $r$  เท่า).

## โครงข่ายปรับเปลี่ยนสร้าง

โครงข่ายปรับเปลี่ยนสร้าง<sup>3</sup> (Generative Adversarial Networks คำย่อ GANs) หมายถึง โครงข่ายประสาทเทียมที่สามารถเรียนรู้ความน่าจะเป็นของข้อมูลที่มีจำนวนมิติสูง ๆ และมีหลาย ๆ โหมดได้ (high-dimensional and multi-modal distribution) โดยโครงข่ายถูกเตรียมด้วยวิธีการฝึกแบบปรับเปลี่ยน. ด้วย

<sup>2</sup>ที่นี่ ยกตัวอย่างภาพช่องสีเดียว เพื่อความกระชับของเนื้อหา. เทคนิคที่อธิบายนี้ สามารถประยุกต์ใช้กับภาพหลายช่องสี ได้อย่างตรงไปตรงมา.

<sup>3</sup>เนื้อหาในหัวข้อนี้ ได้รับอิทธิพลหลัก ๆ จากครสเวลและຄณะ[45].

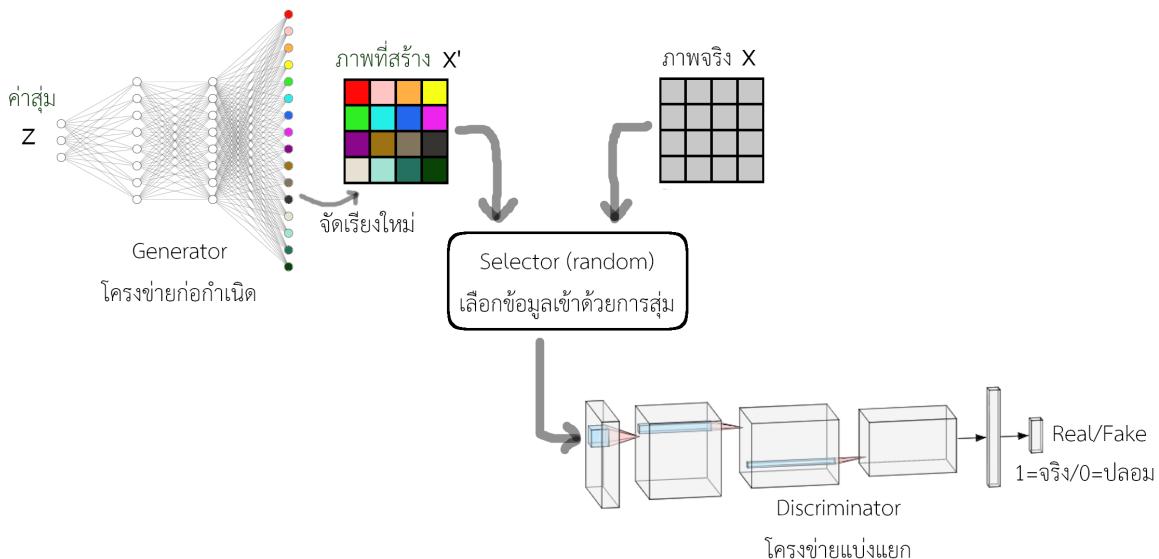
กลไกวิธีการฝึกแบบปรปักษ์ ที่ใช้การเรียนรู้แบบกึ่งมีผู้ช่วยสอนและการเรียนรู้แบบไม่มีผู้สอน โครงข่ายปรปักษ์เชิงสร้างสามารถใช้ประโยชน์จากข้อมูลจำนวนมากที่ไม่มีฉลากเฉลยได้ ซึ่งข้อมูลจำนวนมากนั้น จำเป็นต่อการเรียนรู้ความน่าจะเป็นของข้อมูลที่มีจำนวนมิติสูง ๆ ( เช่น ข้อมูลรูปภาพ ).

อย่างไรก็ตาม การเรียนรู้ความน่าจะเป็นของข้อมูลด้วยโครงข่ายปรปักษ์เชิงสร้าง อาจเป็นเพียงการเรียนรู้เชิงนัย นั่นคือ อาจไม่ได้ค่าความน่าจะเป็นหรือไม่ได้ฟังก์ชันความหนาแน่นความน่าจะเป็น. นั่นคือ โครงข่ายสามารถสังเคราะห์ หรือสร้างตัวอย่างข้อมูลขึ้นมาใหม่ได้ โดยตัวอย่างข้อมูลที่สร้างขึ้นมาใหม่ มีลักษณะในแบบเดียวกับตัวอย่างข้อมูลจริง (กล่าวคือ มีความเป็นไปได้สูงว่ามาจากการแจกแจงเดียวกัน). หากข้อมูลที่กล่าวถึง คือภาพ โครงข่ายปรปักษ์เชิงสร้าง สามารถสร้างตัวอย่างภาพขึ้นมาใหม่ ซึ่งภาพที่สร้างขึ้นนี้ อาจดูเหมือนภาพถ่ายจริง ซึ่งเบื้องหลังหมายถึงว่า โครงข่ายได้เรียนรู้การแจกแจงข้อมูลของภาพถ่ายจริง และสามารถสังเคราะห์ตัวอย่างข้อมูลจากการแจกแจงนั้นได้ แต่ฟังก์ชันการแจกแจงนั้น อาจไม่สามารถเข้าถึงได้โดยตรง.

วิธีการฝึกแบบปรปักษ์ (adversarial training) มีลักษณะเด่น คือ การใช้โครงข่ายสองโครงข่ายในการฝึก และโครงข่ายทั้งสองถูกฝึกโดยมีเป้าหมายที่ขัดแย้งกัน. คณะของกุดเพโล<sup>[77]</sup> เสนอโครงข่ายปรปักษ์เชิงสร้าง เพื่อสร้างตัวอย่างภาพต่าง ๆ ที่เหมือนภาพถ่ายจริงอ่อนๆ. วิธีการฝึกแบบปรปักษ์ ใช้โครงข่ายสองโครงข่าย หนึ่งเรียกว่า โครงข่ายแบ่งแยก (discriminator ใช้สัญกรณ์  $D$ ) และอีกหนึ่ง เรียกว่า โครงข่ายก่อกำเนิด (generator ใช้สัญกรณ์  $G$ ). โครงข่ายแบ่งแยก  $D$  รับอินพุตเป็นภาพ และทำหน้าที่ทำนายว่า ภาพที่รับเข้ามาเป็นภาพถ่ายจริง หรือว่าเป็นภาพปลอม (ภาพที่สร้างขึ้น). โครงข่ายก่อกำเนิด  $G$  รับอินพุตเป็นค่าสุ่ม<sup>4</sup> และทำหน้าที่สร้างภาพขึ้นมา. ในกระบวนการฝึก โครงข่ายแบ่งแยก  $D$  ถูกฝึก โดยมีเป้าหมายคือ การแบ่งแยกให้ถูกต้องมากที่สุด ในขณะที่โครงข่ายก่อกำเนิด  $G$  ถูกฝึก โดยมีเป้าหมายคือ การสร้างภาพปลอมให้เหมือน จนโครงข่ายแบ่งแยก  $D$  ทายถูกน้อยที่สุด.

รูป 7.6 แสดงแนวคิดของวิธีการฝึกแบบปรปักษ์ ซึ่งเป็นกลไกหลักของโครงสร้างปรปักษ์เชิงสร้าง. ในการฝึก โครงข่ายแบ่งแยก  $D$  จะรับอินพุต เป็นภาพ ที่ถูกสุ่มขึ้นมา โดยภาพที่ได้อาจสุ่มจากภาพจริง หรืออาจสุ่มมาจากภาพปลอมที่สร้างโดย โครงข่ายก่อกำเนิด  $G$  และให้ โครงข่ายแบ่งแยก  $D$  ทำนาย. ผลของการทำนายผิดหรือถูก จะถูกนำไปปรับค่าน้ำหนักเพื่อให้ โครงข่ายแบ่งแยก  $D$  ทำงานได้ดีขึ้น แบ่งแยกได้ดีขึ้น (ทายถูกมากขึ้น) และก็จะถูกนำไปปรับค่าน้ำหนัก โครงข่ายก่อกำเนิด  $G$  เพื่อให้  $G$  สร้างภาพได้ดีขึ้น (หลอก  $D$  ได้ดีขึ้น ทำให้  $D$  ทายถูกน้อยลง). หากโครงข่ายก่อกำเนิด  $G$  สามารถหลอกโครงข่ายแบ่งแยก  $D$  ได้โดยสมบูรณ์

<sup>4</sup>ที่ต้องรับอินพุตเป็นค่าสุ่ม เพื่อให้โครงข่ายก่อกำเนิด  $G$  เรียนรู้ที่จะสร้างເອົາພຸດທ່ານຫາຍ. นั่นคือ อินพุตเป็นค่าหนึ่ง โครงข่ายก่อกำเนิด  $G$  สร้างภาพหนึ่ง. อินพุตเป็นอีค่าหนึ่ง โครงข่ายก่อกำเนิด  $G$  สร้างภาพอีกภาพหนึ่ง.



รูปที่ 7.6: วิธีการฝึกแบบปรับปักร์. โครงข่ายก่อกำเนิด  $\mathcal{G}$  (แสดงด้วยโครงข่ายเชื่อมต่อเต็มที่ มุมบนซ้าย) รับอินพุตเป็นเวกเตอร์ค่าสุ่ม  $z$  และให้อาตพุต  $\mathbf{X}'$  ออกมานอกจาก แสดงเอกลักษณ์จัดเรียงใหม่เพื่อให้มีโครงสร้างเหมือนภาพจริง  $\mathbf{X}$ . ภาพจริง  $\mathbf{X}$  ถูกสุ่มออกมากจากชุดข้อมูล. โครงข่ายแบ่งแยก  $\mathcal{D}$  (แสดงด้วยโครงข่ายคอนโวลูชัน มุมล่างขวา) รับอินพุตที่เป็นภาพ โดยโครงข่ายแบ่งแยกไม่รู้ว่าภาพอินพุตที่ได้ ถูกเลือกมาจากภาพจริง หรือภาพที่สร้างขึ้น (การเลือกทำด้วยการสุ่ม) และโครงข่ายแบ่งแยกจะต้องพยายามทายว่า อินพุตที่เห็น เป็นภาพจริง หรือเป็นภาพที่สร้างขึ้น.

แล้ว โครงข่ายแบ่งแยก  $\mathcal{D}$  จะพยายามครึ่ง ๆ นั่นคือ ถ้าภาพที่สร้างขึ้นเหมือนภาพจริง โอกาสคือ เท่ากับเดาสุ่ม ซึ่งถ้าเดาดี ๆ โอกาสสูงคือแค่ประมาณ 0.5 (หรือ 50%). หมายเหตุ รูป 7.6 อาจแสดงโครงข่ายก่อกำเนิดด้วยโครงข่ายเชื่อมต่อเต็มที่ แต่การเปลี่ยนโครงสร้างไปเป็นโครงข่ายคอนโวลูชันก็สามารถทำได้ (ดู [156, 53] เพิ่มเติม).

การฝึกโครงข่ายปรับปักร์เชิงสร้าง กล่าวโดยเจาะจงแล้วก็คือการแก้ปัญหาค่าดีที่สุด ในนิพจน์ 7.6 ได้แก่ เมื่อ

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) \quad (7.6)$$

$$V(\mathcal{G}, \mathcal{D}) = E_{\mathbf{X} \sim p_{data}} [\log \mathcal{D}(\mathbf{X})] + E_{\mathbf{X} \sim p_{\mathcal{G}}} [\log (1 - \mathcal{D}(\mathbf{X}))] \quad (7.7)$$

โดย  $\mathcal{D}(\mathbf{X}) \in (0, 1)$  และ  $\mathcal{D}(\mathbf{X}) \approx 1$  หมายถึง โครงข่ายแบ่งแยกทายว่า  $\mathbf{X}$  เป็นภาพจริง และ  $\mathcal{D}(\mathbf{X}) \approx 0$  หมายถึง โครงข่ายแบ่งแยกทายว่า  $\mathbf{X}$  เป็นภาพปลอมที่สร้างขึ้น.

พจน์  $E_{\mathbf{X} \sim p_{data}} [\log \mathcal{D}(\mathbf{X})]$  หมายถึง ค่าคาดหมายของลօกาڑิที่มของผลลัพธ์จากโครงข่ายแบ่งแยก เมื่ออินพุตของโครงข่ายแบ่งแยกมีการแจกแจงตามข้อมูลจริง (ดังระบุด้วยสัญกรณ์  $\mathbf{X} \sim p_{data}$ ) หรือกล่าว

ง่าย ๆ คือ เมื่อainพุตเป็นภาพจริง หากโครงข่ายแบ่งแยกทำงานถูกต้องโดยสมบูรณ์ แล้ว  $\mathcal{D}(\mathbf{X}) \approx 1$  สำหรับทุก ๆ ภาพจริง และ พจน์  $E_{\mathbf{X} \sim p_{data}}[\log \mathcal{D}(\mathbf{X})] \approx 0$ .

ส่วนพจน์<sup>5</sup>  $E_{\mathbf{X} \sim p_{\mathcal{G}}}[\log(1 - \mathcal{D}(\mathbf{X}))]$  แสดงค่าคาดหมายของลอการิทึ่มของ  $1 - \mathcal{D}(\mathbf{X})$  เมื่อainพุตของมีการแจกแจงตามการแจกแจงจากโครงข่ายก่อกำเนิด (ตั้งระบุด้วยสัญกรณ์  $\mathbf{X} \sim p_{\mathcal{G}}$ ) หรือกล่าวง่าย ๆ คือ เมื่อainพุตถูกสร้างขึ้นจากโครงข่ายก่อกำเนิด. หากโครงข่ายแบ่งแยกทำงานถูกต้องโดยสมบูรณ์ แล้ว  $\mathcal{D}(\mathbf{X}) \approx 0$  สำหรับทุก ๆ ภาพที่สร้างขึ้น และ พจน์  $E_{\mathbf{X} \sim p_{\mathcal{G}}}[\log(1 - \mathcal{D}(\mathbf{X}))] \approx 0$ . แต่หากโครงข่ายแบ่งแยก ทำให้ได้  $\log(0) \rightarrow -\infty$  หรือทำให้ได้ค่าที่ต่ำมาก ๆ.

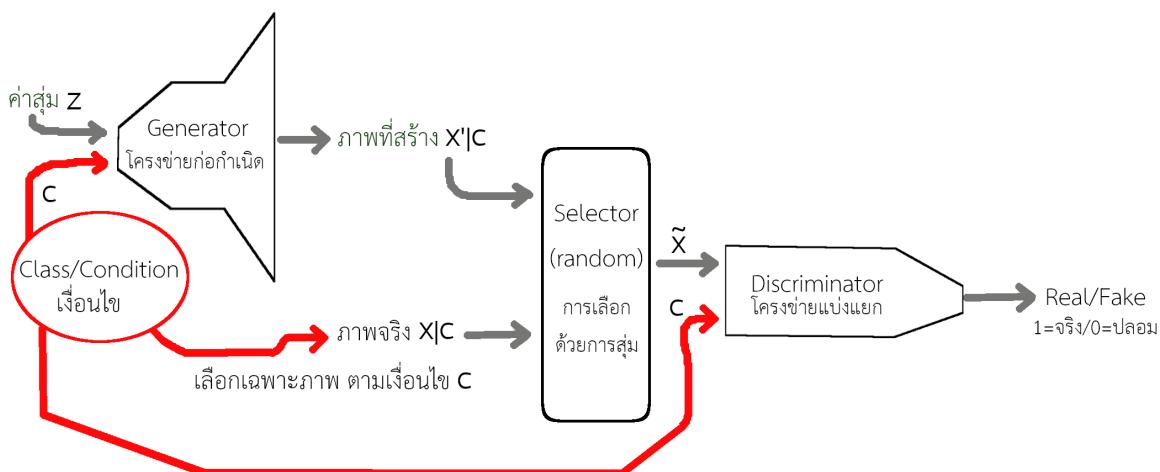
หมายเหตุ สัญกรณ์ที่ในนิพจน์ 7.6 ใช้เพื่อความกระชับ. เช่นเดียวกับการฝึกโครงข่ายประสาทเทียมอื่น ๆ การฝึกโครงข่ายก่อกำเนิดและโครงข่ายแบ่งแยก ก็ดำเนินการผ่านการปรับค่าพารามิเตอร์ต่าง ๆ ของโครงข่าย. นั่นคือ หากจะเขียนนิพจน์ 7.6 ให้ละเอียดถูกต้องมากยิ่งขึ้น อาจเขียนเป็น  $\min_{\boldsymbol{\theta}} \max_{\mathbf{w}} V(\mathcal{G}_{\boldsymbol{\theta}}, \mathcal{D}_{\mathbf{w}})$  เมื่อโครงข่ายก่อกำเนิด  $\mathcal{G}_{\boldsymbol{\theta}}$  และโครงข่ายแบ่งแยก  $\mathcal{D}_{\mathbf{w}}$  ถูกควบคุมโดยรูปแบบด้วยพารามิเตอร์  $\boldsymbol{\theta}$  และ  $\mathbf{w}$  ตามลำดับ.

โครงข่ายแบ่งแยก  $\mathcal{D}$  จะถูกฝึกเพื่อให้ค่าฟังก์ชันจุดประสงค์นี้สูงที่สุด ผ่านกลไกการทำนาย  $\mathcal{D}(\mathbf{X})$ . ในขณะที่ โครงข่ายก่อกำเนิด จะพยายามทำให้ค่าฟังก์ชันจุดประสงค์นี้ต่ำที่สุด โดยผ่านกลไก  $\mathbf{X} \sim p_{\mathcal{G}}$  ซึ่งคือ การสร้างภาพให้เหมือนภาพจริงที่สุด หรือพยายามเรียนรู้ให้การแจกแจง  $p_{\mathcal{G}}$  ใกล้เคียงกับ  $p_{data}$  มากที่สุด. โครงข่ายก่อกำเนิดในอุดมคติ จะมี  $p_{\mathcal{G}} \approx p_{data}$ .

ปัจจัยสำคัญประการหนึ่งคือ โครงข่ายก่อกำเนิด  $\mathcal{G}$  ไม่ได้รับข้อมูลเกี่ยวกับภาพจริงโดยตรงเลย โครงข่ายก่อกำเนิด  $\mathcal{G}$  ถูกบังคับให้เรียนรู้การแจกแจงของภาพจริงผ่านปฏิสัมพันธ์กับโครงข่ายแบ่งแยก  $\mathcal{D}$ . โครงข่ายแบ่งแยก  $\mathcal{D}$  เที่ยงทั้งภาพจริง และภาพที่สร้างขึ้น และได้รับเฉลยผ่านเกรเดียนต์หลังจากทายไป. อีกหอดหนึ่ง โครงข่ายก่อกำเนิด  $\mathcal{G}$  ที่ได้รับเกรเดียนต์ของมันผ่านเกรเดียนต์ของโครงข่ายแบ่งแยก  $\mathcal{D}$  อีกต่อหนึ่ง.

โครงข่ายก่อกำเนิด อาจถูกมองว่าเป็นการเรียนรู้ เพื่อที่จะแปลงข้อมูลจากปริภูมิของตัวแทนสู่ ที่อาจถูกเรียกว่า ปริภูมิตัวแทน (representation space) หรือปริภูมิช่องเร้น (latent space) ไปสู่ปริภูมิของข้อมูล. นั่นคือ  $\mathcal{G} : \mathbf{z} \rightarrow \mathbf{X}$  เมื่อ  $\mathbf{z}$  คือตัวแปรในปริภูมิช่องเร้น และ  $\mathbf{X}$  คือตัวแปรในปริภูมิข้อมูล. ส่วนโครงข่ายแบ่งแยกก็เป็นเสมือนการแปลงจากข้อมูลไปสู่ค่าระหว่างศูนย์กับหนึ่ง. นั่นคือ  $\mathcal{D} : \mathbf{X} \rightarrow (0, 1)$ . ภายหลังการฝึกเสร็จสิ้น โครงข่ายก่อกำเนิด  $\mathcal{G}$  สามารถนำไปใช้สร้างตัวอย่างข้อมูลได้ตามต้องการ.

<sup>5</sup> คณะของกูดเพโล[77] ใช้พจน์  $E_{\mathbf{z} \sim p_{\mathbf{z}}}[\log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z})))]$  เมื่อ  $\mathcal{G}(\mathbf{z})$  แทนภาพที่สร้างจากโครงข่ายก่อกำเนิดตามค่าสุ่ม  $\mathbf{z}$  และ  $E_{\mathbf{z} \sim p_{\mathbf{z}}}$  หมายถึงค่าคาดหมายคำนวนตามการแจกแจงของตัวแปรสุ่ม. แต่ ณ ที่นี่เขียนพจน์นี้ ตามเครื่องสวอลและคณะ[45] เพื่อความกระชับในการอธิบาย.



รูปที่ 7.7: การฝึกโครงสร้างปรับปักษ์เชิงสร้างแบบมีเงื่อนไข. คล้ายการฝึกโครงสร้างปรับปักษ์เชิงสร้างแบบดั้งเดิม (ไม่มีเงื่อนไข) การฝึกโครงสร้างปรับปักษ์เชิงสร้างแบบมีเงื่อนไข เพิ่มข้อมูลของเงื่อนไข และให้ข้อมูลเงื่อนไขนี้กับโครงข่ายก่อกำเนิด และโครงข่ายแบ่งแยกรวมถึงข้อมูลที่จริงที่จะเลือกมา ก็ต้องถูกควบคุมให้เป็นข้อมูลที่ตรงกับเงื่อนไขด้วย. การกำหนดเงื่อนไข เช่นนี้ ช่วยให้เราสามารถควบคุมเอาต์พุตของโครงข่ายก่อกำเนิด เพื่อให้สร้างเอาต์พุตตามเงื่อนไขที่เราต้องการได้.

โครงข่ายปรับปักษ์เชิงสร้างแบบมีเงื่อนไข. เมียร์ฉะและคณะ[129] ขยายความสามารถของโครงข่ายปรับปักษ์เชิงสร้าง โดยใช้ความน่าจะเป็นแบบมีเงื่อนไข. โครงข่ายปรับปักษ์เชิงสร้างที่มีความสามารถที่เพิ่มขึ้นมา เช่นนี้ ถูกเรียกว่า โครงข่ายปรับปักษ์เชิงสร้างแบบมีเงื่อนไข (Conditional Generative Adversarial Networks). กระบวนการฝึกของโครงข่ายปรับปักษ์เชิงสร้างแบบมีเงื่อนไข อาจตั้งจุดประสงค์เป็น

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) = E_{\mathbf{X} \sim p_{data|\mathbf{C}}} [\log \mathcal{D}(\mathbf{X} | \mathbf{C})] + E_{\mathbf{X} \sim p_{\mathcal{G}|\mathbf{C}}} [\log (1 - \mathcal{D}(\mathbf{X} | \mathbf{C}))] \quad (7.8)$$

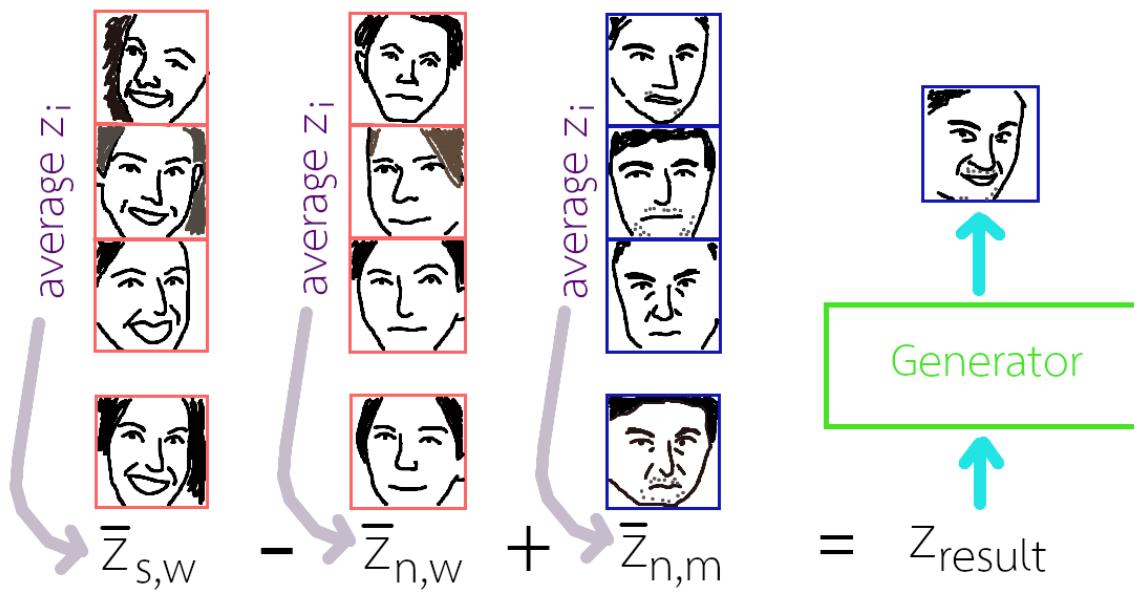
เมื่อ  $\mathcal{D}(\mathbf{X} | \mathbf{C})$  แทนผลการทำนายจากโครงข่ายแบ่งแยก ที่รับอินพุตหลักเป็น  $\mathbf{X}$  และรับอินพุตรองเป็น  $\mathbf{C}$  ซึ่งใช้ระบุเงื่อนไข. สัญกรณ์  $\mathbf{X} \sim p_{data|\mathbf{C}}$  หมายถึง ตัวแปร  $\mathbf{X}$  มีการแจกแจงตามข้อมูลจริงที่เป็นไปตามเงื่อนไข  $\mathbf{C}$ . สัญกรณ์  $\mathbf{X} \sim p_{\mathcal{G}|\mathbf{C}}$  หมายถึง ตัวแปร  $\mathbf{X}$  มีการแจกแจงตามการแจกแจงจากโครงข่ายก่อกำเนิดที่เงื่อนไข  $\mathbf{C}$ . รูป 7.7 แสดงกลไกเพิ่มเติม เพื่อเพิ่มคุณสมบัติการใช้เงื่อนไข ให้กับโครงข่ายปรับปักษ์เชิงสร้าง. โครงสร้างและรายละเอียดในการทำโครงข่ายปรับปักษ์เชิงสร้างแบบมีเงื่อนไข อาจแตกต่างไปได้ เช่น อินโฟแกน (InfoGAN[36]).

การประยุกต์ใช้โครงข่ายปรับปักษ์เชิงสร้าง. โครงข่ายปรับปักษ์เชิงสร้าง เป็นพัฒนาการที่สำคัญสำหรับการเรียนรู้เชิงลึก และได้ทำให้เกิดการประยุกต์ใช้อย่างกว้างขวาง ขยายเข้าไปแม้แต่ในวงการศิลปะ. การศึกษาวิจัยและขอบเขตการใช้งานของโครงข่ายปรับปักษ์เชิงสร้าง เป็นไปอย่างรวดเร็วและต่อเนื่อง จนโครงข่ายปรับปักษ์เชิงสร้างเป็นเสมือนศาสตร์ย่อย ๆ ในตัวเอง. ตัวอย่างการประยุกต์ใช้ทั่ว ๆ ไปส่วนหนึ่งของโครงข่ายปรับปักษ์

เชิงสร้าง ได้แก่ การจำแนกกลุ่ม (เช่น การนำโครงข่ายแบ่งแยกไปใช้), การสกัดลักษณะสำคัญ (ซึ่งอาจจะได้จากทั้งค่าเออร์พุตขั้นชื่นภายในโครงสร้างของโครงข่ายแบ่งแยก หรืออาจจะได้จากการทำพีชคณิตเวกเตอร์ที่จะอธิบายเพิ่มเติมต่อไป), การสังเคราะห์ข้อมูล (ซึ่งคือ การสร้างข้อมูล โดยให้โครงข่ายก่อกำเนิด และนี่คือจุดประสงค์หลักของโครงข่ายปรับกษ์เชิงสร้าง), การแปลงรูปหนึ่งไปสู่อีกรูปหนึ่ง, การเพิ่มความละเมียดให้กับภาพ เป็นต้น.

คณะของรีด[163] ใช้โครงข่ายปรับกษ์เชิงสร้าง ในการสร้างภาพขั้นมาตามคำบรรยาย. โครงข่ายปรับกษ์เชิงสร้างอะไรทีไน (Generative Adversarial What-Where Network[162]) สามารถสร้างภาพขั้นจากส่วนภาพเด็ก ๆ ที่แต่ละส่วนภาพสร้างขึ้นมาตามตำแหน่งที่กำหนด และตามลักษณะพื้นผิวที่บรรยาย. นอกจากนั้น มีการใช้โครงข่ายปรับกษ์เชิงสร้างไปใช้ในกระบวนการแก้ไขและตกแต่งรูป[23, 225]. การแปลงรูปหนึ่งไปสู่อีกรูปหนึ่ง[94] ก็มีการประยุกต์ใช้ที่หลากหลาย เช่น การแปลงกระบวนการแบบศิลปะ (artistic style transfer[117]), การแปลงกระบวนการแบบ (style transfer[101]), การสร้างภาพเหมือนจริงตามส่วนภาพ ที่ศิลปินสามารถคาดภาพคร่าวแล้วให้โครงข่ายปรับกษ์เชิงสร้างช่วยเติมรายละเอียด (semantic image synthesis[148, 147]), การสร้างภาพล้อเลียนบุคคลอัตโนมัติ (automatic caricature generation[185]), การเพิ่มอายุให้หน้า (age progression[51]).

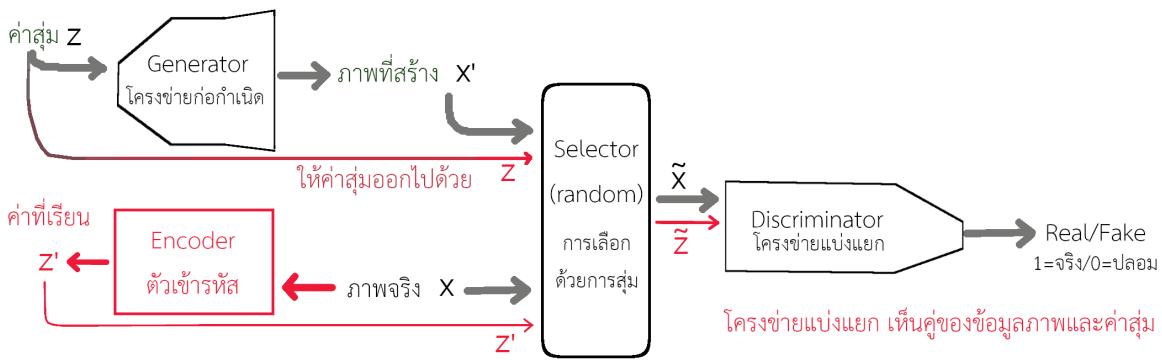
พีชคณิตเวกเตอร์ (Vector arithmetic) คือ การนำเวกเตอร์ค่าสุ่ม  $\mathbf{z}$  ที่เป็นอินพุตของโครงข่ายก่อกำเนิด สำหรับภาพต่าง ๆ มาทำบวกหรือลบกัน แล้วนำเวกเตอร์ผลลัพธ์เข้าไปเป็นอินพุตของโครงข่ายก่อกำเนิด ผลลัพธ์ที่ได้พบว่ามีลักษณะสำคัญจากการของเวกเตอร์ที่เป็นตัวถูกดำเนินการ ผสมกันในลักษณะเชิงเส้น. ตัวอย่างเช่น แรดฟอร์ดและคณะ[156] เลือกภาพผู้หญิงยิ้มสามภาพอookma แล้วหาเวกเตอร์เฉลี่ย  $\bar{\mathbf{z}}_{\text{smile,woman}}$  จากเวกเตอร์ค่าสุ่มของภาพทั้งสาม นำไปลบออกด้วยเวกเตอร์เฉลี่ย  $\bar{\mathbf{z}}_{\text{neutral,woman}}$  ที่เฉลี่ยจากเวกเตอร์ค่าสุ่มของภาพผู้หญิงหน้าเฉย (ไม่ได้ยิ้ม) แล้วนำไปบวกด้วยเวกเตอร์เฉลี่ย  $\bar{\mathbf{z}}_{\text{neutral,man}}$  ที่เฉลี่ยจากเวกเตอร์ค่าสุ่มของภาพผู้ชายหน้าเฉย (ไม่ได้ยิ้ม) สุดท้ายนำเวกเตอร์ผลลัพธ์เข้าโครงข่ายก่อกำเนิด และรูปภาพที่สร้างขึ้นมา พบร่วมกับภาพผู้ชายยิ้ม. นั่นคือ โครงข่ายก่อกำเนิด ได้เรียนรู้ที่จะเข้ารหัสลักษณะสำคัญของภาพไว้ในเวกเตอร์ค่าสุ่ม  $\mathbf{z}$  และการเข้ารหัสยังเป็นไปในลักษณะเชิงเส้น (จึงสามารถลบและบวก แล้วได้ผลลัพธ์ในลักษณะเชิงเส้นอookma). ด้วยคุณสมบัติเช่นนี้ อาจมองว่าโครงข่ายก่อกำเนิดได้เรียนรู้ที่จะกำหนดความหมายของลักษณะสำคัญไว้ที่ค่าของเวกเตอร์  $\mathbf{z}$ . เนื่องจากความหมายของลักษณะสำคัญนี้ ไม่ได้ถูกกำหนดอookmaอย่างชัดเจน ต้องอาศัยการสืบการสังเกตจึงจะพหุเห็นความเชื่อมโยง เวกเตอร์  $\mathbf{z}$  บางครั้งจึงถูกเรียกเป็นลักษณะชั้นเร้น (latent representation) และปริภูมิของ  $\mathbf{z}$  จึงมักถูกอ้างถึงเป็นปริภูมิชั้นเร้น หรือปริภูมิตัวแทน



รูปที่ 7.8: การทำพีชคณิตเวกเตอร์กับเวกเตอร์ค่าสุ่มของโครงข่ายก่อกำเนิด. ภาพผู้หญิงยิมถูกคัดเลือกมาสามภาพ โดยเก็บเวกเตอร์ค่าสุ่ม  $z_i$  ของแต่ละภาพมาด้วย นำเวกเตอร์ค่าสุ่มมาหาค่าเฉลี่ย  $\bar{z}_{s,w}$ . ทำแบบเดียวกันกับภาพผู้หญิงหน้าเฉย และภาพผู้ชายหน้าเฉย ได้ค่าเฉลี่ย  $\bar{z}_{n,w}$  สำหรับภาพผู้หญิงหน้าเฉย และ  $\bar{z}_{n,m}$  สำหรับภาพผู้ชายหน้าเฉย. คำนวณ  $z_{result} = \bar{z}_{s,w} - \bar{z}_{n,w} + \bar{z}_{n,m}$  แล้วนำ  $z_{result}$  เข้าโครงข่ายก่อกำเนิด ภาพผลลัพธ์  $\mathbf{X}' = \mathcal{G}(z_{result})$  ที่ได้พบว่า ภาพ  $\mathbf{X}'$  คล้ายภาพของผู้ชายยิ้ม. หมายเหตุ การทำพีชคณิตทำในบริภูมิช่องเร้น (ทำกับเวกเตอร์ค่าสุ่ม) ถึงจะได้ผลลัพธ์คล้ายการเลือกลักษณะสำคัญ. หากทำพีชคณิตในบริภูมิของภาพโดยตรง ( เช่น  $\mathbf{X}_{result} = \bar{\mathbf{X}}_{s,w} - \bar{\mathbf{X}}_{n,w} + \bar{\mathbf{X}}_{n,m}$  ) ภาพที่ได้จะเลอเทอะ และยากจะมองออก. ดูภาพตัวอย่างใน [156].

ตามที่อภิปรายไปก่อนหน้า. รูป 7.8 แสดงภาพประกอบที่วัดขึ้น (ดูภาพจริงจาก [156]).

แม้ว่าโครงข่ายก่อกำเนิดที่ถูกฝึกมาดีแล้วจะสามารถแปลงค่าจากบริภูมิช่องเร้น ไปสู่บริภูมิช่องมูลได้ แต่ เช่นเดียวกับโครงข่ายประสาทเทียมทั่วไป คือ หากต้องการจะคำนวณย้อนกลับ ซึ่งคือการแปลงจากภาพ  $\mathbf{X}$  กลับมาเป็นเวกเตอร์ช่องเร้น  $z$  นั้น ไม่สามารถทำได้โดยตรง. อย่างไรก็ตาม มีความพยายามที่จะเพิ่มกลไก ภายใน เพื่อให้โครงข่ายปรับปรุงเชิงสร้างสามารถคำนวณกลับไปกลับมาระหว่างบริภูมิช่องเร้น และบริภูมิช่องมูลได้. รูป 7.9 แสดงโครงสร้างของใบแกน (BiGAN[56]). การอนุมานที่เรียนเชิงปรบกษ (Adversarially learned inference คำย่อ ALI[59]) ซึ่งเป็นแนวคิดเดียวกันกับใบแกน ก็ถูกเสนอในช่วงเวลาเดียวกัน. กลไก ที่สำคัญ สำหรับทั้งใบแกนและการอนุมานที่เรียนเชิงปรบกษ คือ การเพิ่มตัวเข้ารหัส (encoder) เพื่อเรียนรู้ ความสัมพันธ์ระหว่างบริภูมิช่องเร้นและบริภูมิช่องมูล. หากจำเพาะลงไปก็คือ ตัวเข้ารหัส ทำหน้าที่เรียนรู้การ แยกแยะแบบมีเงื่อนไข  $p(z|\mathbf{X})$  เมื่อ  $\mathbf{X}$  แทนข้อมูลภาพ และ  $z$  คือ ค่าลักษณะช่องเร้น ที่โครงข่ายก่อกำเนิด ใช้. อย่างไรก็ตาม เครสนิเวลและคณะ[45] ให้ความเห็นว่า ภาพที่สร้างจากใบแกนหรือการอนุมานที่เรียนเชิง ปรบกษยังมีคุณภาพค่อนข้างต่ำ. นั่นอาจหมายถึงว่า การศึกษาวิจัย ถึงกลไกแปลงกลับจากบริภูมิช่องมูลไป สู่บริภูมิช่องเร้น ในโครงข่ายปรบกษเชิงสร้าง ยังอยู่ในขั้นเริ่มต้นเท่านั้น.



รูปที่ 7.9: โครงสร้างของใบแกนและการอนุมานที่เรียนเชิงปรับักษ์ เพื่อเรียนรู้ความสัมพันธ์ระหว่างปริภูมิช่องเร้นและปริภูมิช่องมูล กลไกสำคัญอยู่ที่ตัวเข้ารหัส. โครงข่ายก่อกำเนิด เรียนรู้การแจกแจงแบบมีเงื่อนไข  $p(\mathbf{X}|z)$ . ส่วนตัวเข้ารหัส ที่เห็นข้อมูลจริง แต่พยายามเรียนรู้การแจกแจงแบบมีเงื่อนไข  $p(z|\mathbf{X})$ . ทั้งโครงข่ายก่อกำเนิดและตัวเข้ารหัส พยายามเรียนรู้ เพื่อจะสร้างคุณของมูล  $(\tilde{\mathbf{X}}, \tilde{z})$  ที่โครงข่ายแบ่งแยก จำแนกได้ยากว่าเป็นภาพปลอม (คู่  $(\mathbf{X}', z)$  จากโครงข่ายก่อกำเนิด) หรือเป็นภาพจริง (คู่  $(\mathbf{X}, z')$  จากภาพจริงและตัวเข้ารหัส).

**ปัญหาในการฝึก.** การฝึกโครงข่ายปรับักษ์เชิงสร้าง ถูกรายงาน[156, 45] ว่าทำได้ยาก และมีโอกาสล้มเหลว มาก จากหลาย ๆ สาเหตุ รวมถึง

- การลู้เข้าหาก[156] ที่นักวิจัยมักพบว่า มันยากที่ทำให้การฝึกโครงข่ายก่อกำเนิด ลู้เข้า. ปัญหานี้ ส่วนหนึ่งอาจมาจากการรرمชาติของข้อมูลภาพ. ข้อมูลภาพมีปริภูมิที่ขนาดใหญ่มาก ๆ (รูปสีขนาด  $W \times H$  พิกเซล เทียบเท่าจุดข้อมูลในปริภูมิขนาด  $3 \cdot W \cdot H$  มิติ) แต่ตัวอย่างข้อมูลต่าง ๆ ที่มี (เช่น ภาพจริง ต่าง ๆ) เป็นข้อมูลสำหรับการสนับสนุนของฟังก์ชันความหนาแน่นความน่าจะเป็น<sup>6</sup> ครอบคลุมเพียงบริเวณเล็ก ๆ ในปริภูมิ และเมื่อเทียบกับขนาดของปริภูมิทั้งหมดแล้ว บริเวณที่ครอบคลุมมีขนาดเล็กมาก ๆ. กล่าวคือ แม้จะใช้ตัวอย่างข้อมูลจำนวนมากแล้ว แต่จำนวนตัวอย่างที่ใช้ ก็ยังน้อยมากเมื่อเทียบกับขนาดประชากร (โอกาสทั้งหมดที่เป็นไปได้ของข้อมูล) และตัวอย่างข้อมูลเหล่านี้ ก็ยากที่จะเป็นตัวแทนที่พอเพียงได้.

นอกจากนั้น ยังมีการศึกษา[45, 6] ที่พบว่า ก่อนการฝึก การแจกแจงก่อกำเนิด  $p_G$  อาจจะไม่มีการซ้อนทับกับการแจกแจงเป้าหมาย  $p_{data}$  เลย. หากเป็นเช่นนั้นจริง ผลคือ โครงข่ายแบ่งแยกจะสามารถถูกฝึกได้อย่างง่ายดายและรวดเร็ว เพื่อที่จะสามารถจำแนกตัวอย่างจริง  $\mathbf{X} \sim p_{data}$  ออกจากตัวอย่างปลอม  $\mathbf{X} \sim p_G$  ได้อย่างแม่นยำสมบูรณ์ (ความแม่นยำ 100%) นั่นคือ การฝึกโครงข่ายแบ่ง

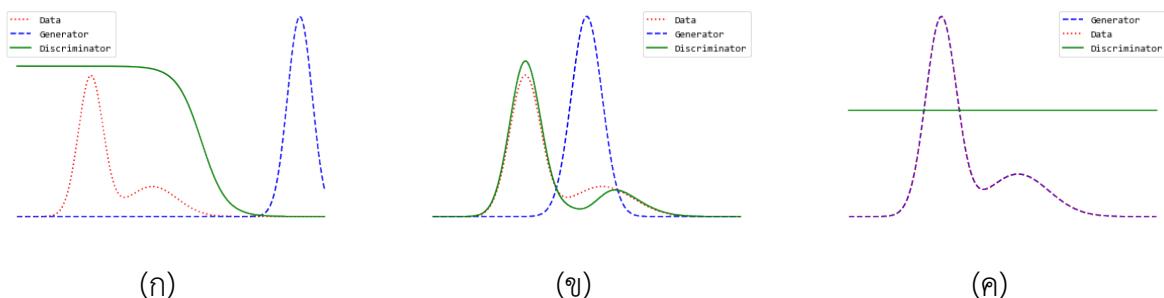
<sup>6</sup>ในทางคณิตศาสตร์และสถิติศาสตร์ การสนับสนุนของฟังก์ชันค่าจริง (the support of a real-valued function) หมายถึง เซตย่อยของโดเมน ที่เซตย่อยนั้นมีจุดข้อมูลที่ค่าของฟังก์ชันที่สนใจไม่เป็นศูนย์อยู่. ในบริบทของการเรียนรู้การแจกแจง การสนับสนุนของฟังก์ชันความหนาแน่นความน่าจะเป็น หมายถึง เซตย่อยในปริภูมิที่มีจุดข้อมูลจริงอยู่.

แยกจะลู่เข้าจัน พังก์ชันจุดประสังค์มีค่าเป็นศูนย์ ได้อย่างรวดเร็ว และส่งผลให้เกรเดียนต์ต่าง ๆ เป็นศูนย์ ซึ่งจะทำให้การฝึกโครงข่ายกำเนิดไม่สามารถทำต่อไปได้.

อีกประเด็นหนึ่ง เครสวเลลและคณะ[45] อภิปรายประเด็นจากการศึกษาทฤษฎีโครงข่ายปรปักษ์เชิงสร้าง[77] กับพังก์ชันจุดประสังค์ที่ใช้ ว่า หากโครงข่ายแบ่งแยกไม่ได้อยู่ในสภาพที่ดีที่สุดแล้ว การฝึกโครงข่ายกำเนิดก็อาจจะไม่แม่นยำ หรืออาจได้ผลลัพธ์ผิดความหมายได้. นี่อาจหมายถึง ความจำเป็นในการออกแบบพังก์ชันจุดประสังค์ใหม่ สำหรับโครงข่ายปรปักษ์เชิงสร้าง. อย่างไรก็ตาม ด้วยพังก์ชันจุดประสังค์ดังเช่นนิพจน์ 7.7 ประเด็นนี้ ที่เมื่อประกอบกับข้ออภิปรายข้างต้นแล้ว จะช่วยให้เห็นความยากของการฝึกโครงข่ายปรปักษ์เชิงสร้าง ที่หากโครงข่ายแบ่งแยกทำงานได้ดีเกินไป การฝึกโครงข่ายกำเนิดก็จะทำได้ยาก หรืออาจล้มเหลว และหากโครงข่ายแบ่งแยกทำงานไม่ดีเลย การฝึกโครงข่ายกำเนิดก็จะไปผิดทาง.

- การพังทลายของภาวะ (mode collapse[173]) ที่หมายถึง โครงข่ายกำเนิดสร้างแต่เอาร์พุตที่เหมือน ๆ กัน แม้ว่าจะรับอินพุตต่างกัน. จุดประสังค์ คือ ต้องการได้โครงข่ายกำเนิดที่สามารถสร้างแต่เอาร์พุตออกมาได้ โดยเอาร์พุตที่ได้ มีการแจกแจงคล้ายข้อมูลจริงมากที่สุด. ตัวอย่างเช่น แทนที่โครงข่ายกำเนิดจะสามารถสร้างภาพเหมือนจริงใหม่ ๆ ออกมากได้เรื่อย ๆ แต่โครงข่ายกำเนิดกลับสร้างภาพเหมือนจริงภาพเดิม ๆ ออกมา แม้ว่าจะรับอินพุต (ซึ่งคือค่าสุ่ม) ค่าใหม่แล้วก็ตาม.
- การฝึกโครงข่ายแบ่งแยกได้เร็วและดีเกินไป[45]. หากโครงข่ายแบ่งแยกทำงานได้ดีมาก ๆ อาจทำให้  $V(\mathcal{G}, \mathcal{D}) \approx 0$  ซึ่งมีผลให้เกรเดียนต์มีค่าใกล้ศูนย์ และทำให้การฝึกโครงข่ายก่อกำเนิดทำได้ยากมาก หรืออาจล้มเหลวได้. รูป 7.10 แสดงสมมติฐานกลไกการเรียนรู้การแจกแจงของโครงข่ายปรปักษ์เชิงสร้าง ในสถานการณ์ต่าง ๆ.

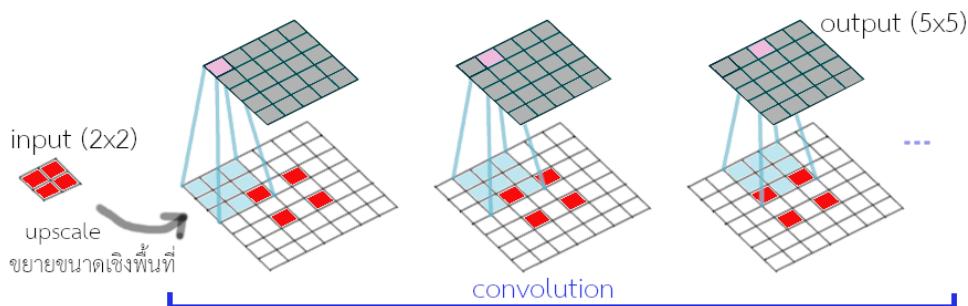
แม้มองผิวเผินอาจจะดูดี แต่การฝึกโครงข่ายแบ่งแยกให้ได้สมบูรณ์ก่อน (เมื่อเทียบกับความสามารถของโครงข่ายก่อกำเนิด) แล้วจึงค่อยฝึกโครงข่ายก่อกำเนิด ไม่ใช่แนวทางที่นิยมปฏิบัติ ด้วยเหตุผลข้างต้น. แนวทางปฏิบัติที่นิยมคือ การฝึกโครงข่ายแบ่งแยกไปก่อนระยะหนึ่ง แล้วจึงค่อยฝึกโครงข่ายก่อกำเนิดไปพร้อม ๆ กัน. นอกจากนั้น ด้วยเหตุผลด้านเสถียรภาพเชิงตัวเลข ในการฝึกโครงข่ายก่อกำเนิด มักนิยมใช้จุดประสังค์  $\max_{\mathcal{G}} E_{\mathbf{X} \sim p_{\mathcal{G}}} [\log \mathcal{D}(\mathbf{X})]$  มากกว่า  $\min_{\mathcal{G}} E_{\mathbf{X} \sim p_{\mathcal{G}}} [\log(1 - \mathcal{D}(\mathbf{X}))]$ .



**รูปที่ 7.10:** ภาพแสดงสมมติฐานการเรียนรู้การแจกแจงข้อมูลของโครงข่ายปรับกษ์เชิงสร้าง. ทั้งสามภาพ แสดง ภาพอย่างง่าย ของ ค่าความหนาแน่นความน่าจะเป็น (แกนตั้ง) ต่อค่าข้อมูล (แกนนอน) ของข้อมูลจริง (แสดงด้วยเส้นไข่ปลา) กับ ของที่สร้างขึ้น จากโครงข่ายก่อกำเนิด (แสดงด้วยเส้นประ) พร้อมทั้งแสดงค่าเอาร์พัฒของโครงข่ายแบ่งแยก (แสดงด้วยเส้นทึบ). ภาพชี้ัยสุด (ก) การแจกแจงของข้อมูลจริง ต่างจากการแจกแจงจากโครงข่ายก่อกำเนิดมาก ไม่มีส่วนที่ซ้อนทับกันเลย. โครงข่ายแบ่งแยก สามารถ จำแนกข้อมูลจริงออกจากข้อมูลปลอมได้อย่างสมบูรณ์ ด้วยความแม่นยำสูงสุด. กรณีเช่นนี้ จะทำให้การฝึกโครงข่ายก่อกำเนิดไม่ สามารถดำเนินการต่อได้. ภาพกลาง (ข) การแจกแจงของข้อมูลจริง ต่างจากการแจกแจงจากโครงข่ายก่อกำเนิด แต่มีส่วนซ้อน ทับกันอยู่มาก. โครงข่ายแบ่งแยก ไม่สามารถจำแนกข้อมูลจริงออกจากข้อมูลปลอมได้อย่างสมบูรณ์. การฝึกโครงข่ายก่อกำเนิด สามารถดำเนินการต่อไปได้. ภาพขวา (ค) โครงข่ายก่อกำเนิดสามารถเรียนรู้การแจกแจงของข้อมูลจริง และสามารถสร้างข้อมูลจาก การแจกแจงที่เหมือนของข้อมูลจริง. โครงข่ายแบ่งแยก ไม่สามารถจำแนกข้อมูลจริงออกจากข้อมูลปลอมได้เลย.

เทคนิคในการฝึกโครงข่ายปรับกษ์เชิงสร้าง. จากความท้าทายในการฝึกโครงข่ายปรับกษ์เชิงสร้างที่อภิปรายข้างต้น แรดฟอร์ดและคณะ[156] ได้เสนอดีชีแกน (DCGAN จาก Deep Convolutional Generative Adversarial Networks) เพื่อบรรเทาปัญหา. ดีชีแกน มีปัจจัยที่สำคัญคือ (1) การใช้คอนโวลูชันก้าวยาว (strided convolution) และการใช้ชั้นดึงรวม ในโครงสร้างของโครงข่ายแบ่งแยก  $D$ . ค่อนโวลูชันก้าวยาว หมายถึง ชั้นค่อนโวลูชันที่ใช้ก้าวย่างขนาดใหญ่กว่าหนึ่ง เช่น การใช้ขนาดก้าวย่าง  $S = 2$ . ผลลัพธ์ของการใช้ค่อนโวลูชันก้าวยาว จะให้ผลเหมือนการลดขนาดแผนที่ลักษณะลำคัญลง (หรือ spatially downsampling). (2) การใช้ชั้นค่อนโวลูชัน โดยเฉพาะใช้การทำค่อนโวลูชันก้าวเศษ (fractionally-strided convolution หรือ transposed convolution[60]) ในโครงสร้างของโครงข่ายก่อกำเนิด  $G$ .

หากจะอธิบายโดยง่ายแล้ว ภายใต้ปริบหนึ่ง ค่อนโวลุชั่นก้าวเศษ ก็คือ การขยายขนาดแผนที่ลักษณะลำคัญที่เป็นอนพุต แล้วจึงทำการคำนวนค่อนโวลุชั่น. การขยายขนาดแผนที่ลักษณะลำคัญ (ซึ่งขยายเฉพาะในมิติลำดับเชิงพื้นที่) ทำด้วยการเติมค่าศูนย์เข้าไประหว่างค่าพิกเซลเดิม (รวมการเติมด้วยศูนย์ ที่เติมค่าศูนย์ที่ปริเรณขอบของแผนที่ด้วย). ผลลัพธ์ของการใช้ค่อนโวลุชั่นก้าวเศษ จะให้ผลเหมือนการเพิ่มขนาดแผนที่ลักษณะลำคัญขึ้น (หรือ spatially upsampling). รูป 7.11 แสดงกลไกที่ค่อนโวลุชั่นก้าวเศษ ช่วยขยายขนาดแผนที่ลักษณะลำคัญขึ้น. หากสังเกตการทำค่อนโวลุชั่นในรูป เมื่อมองจากปฏิสัมพันธ์ระหว่างฟิลเตอร์และอินพุต อาจดูเหมือนกับว่าฟิลเตอร์ขยายบ่งานพิกเซลช้าลง คล้ายกับว่า ใช้ขนาดก้าวย่างที่เล็กกว่าหนึ่ง ซึ่งเป็นที่มาของชื่อ ค่อนโวลุชั่นก้าวเศษ.



รูปที่ 7.11: คอนโวลูชันก้าวเศษ ขยายขนาดเชิงพื้นที่ ขั้นตอนนี้เป็นการเปลี่ยนแปลงขนาดของภาพโดยใช้ฟิลเตอร์ขนาด  $3 \times 3$  ในการเติมคุณย์ที่ตัวรับห่วงพิกเซล และทำการคำนวณผลลัพธ์ คือขนาดฟิลเตอร์  $s$  คือขนาดฟิลเตอร์ และเติมคุณย์ที่ตัวรับห่วงพิกเซล  $s - 1$  ตัว. กรณีนี้  $o' = 2(2 - 1) + 3 = 5$ . (ดูรายละเอียดการคำนวณจาก [60]. ภาพตัดแปลงจาก [60, รูป 4.5])

หมายเหตุ คอนโวลูชันก้าวเศษ บางครั้งอาจถูกเรียก คอนโวลูชันสลับเปลี่ยน (Transposed convolution) ซึ่งมาจากการตีความทางคณิตศาสตร์. นั่นคือ หากดำเนินคอนโวลูชันด้วยการแปลงอินพุตและค่าน้ำหนักของฟิลเตอร์เป็นเมทริกซ์ โดยจัดรูปเมทริกซ์ทั้งสองให้ถูกต้อง (มีการใช้ค่าซ้ำและมีการเติมคุณย์เข้าไป ดูแบบฝึกหัด 6.7 ประกอบ) ซึ่งทำให้ได้เมทริกซ์ของค่าน้ำหนักเป็น เมทริกซ์มากเลขคุณย์ (sparse matrix) แล้ว การทำคอนโวลูชัน ก็เหมือนกับการคูณเมทริกซ์อินพุตเข้ากับเมทริกซ์ค่าน้ำหนัก และนำผลลัพธ์ที่ได้ไปจัดรูปให้เข้ากับโครงสร้างที่ถูกต้อง. ในทำนองเดียวกัน คอนโวลูชันก้าวเศษ ก็เสมือนการคูณเมทริกซ์อินพุตเข้ากับการสลับเปลี่ยนของเมทริกซ์ค่าน้ำหนัก. ดังนั้น กระบวนการนี้จึงเรียกว่า คอนโวลูชันสลับเปลี่ยน. (ศึกษาเพิ่มเติมได้จาก [60])

คอนโวลูชันก้าวเศษ บางครั้ง อาจถูกเรียกว่า การถอดคอนโวลูชัน (Deconvolution). อาย่างไรก็ตาม การถอดคอนโวลูชัน มีความหมายอื่น (ซึ่งเป็นคนละเรื่อง) และถูกยอมรับมากกว่า. ความหมายที่ถูกยอมรับมากกว่าของการถอดคอนโวลูชัน คือ การถอดพารามิเตอร์ของโครงข่ายคอนโวลูชันย้อนกลับ เพื่อศึกษาถูกต้องการทำงานของโครงข่ายคอนโวลูชัน ว่า ฟิลเตอร์แต่ละตัวที่ใช้ในโครงข่ายคอนโวลูชัน ได้เรียนรู้เพื่อจะตรวจจับลักษณะรูปแบบ เช่นไร. สำหรับการถอดคอนโวลูชัน ในความหมายที่นิยมนี้ สามารถศึกษาเพิ่มเติมได้จากบทความ [223, 224, 187] เป็นต้น.

นอกจากการใช้คอนโวลูชันก้าวเศษและคอนโวลูชันก้าวเศษในโครงสร้างของโครงข่ายแบ่งแยกและโครงข่ายก่อกำเนิดแล้ว แรดฟอร์ดและคณะ[156] ยังแนะนำการใช้แบบอร์ม (หัวข้อ 5.5), แนะนำใช้ชั้นคอนโวลูชันลีก ๆ (หลาย ๆ ชั้น) แทนการใช้ชั้นเชื่อมต่อเติมที่<sup>7</sup>, แนะนำการใช้เรลู สำหรับฟังก์ชันกราฟตุนของทุก ๆ

<sup>7</sup>อย่างไรก็ตาม ในโครงข่ายก่อกำเนิด การใช้ชั้นเชื่อมต่อเติมที่เป็นชั้นคำนวนแรก (ชั้นที่รับอินพุตเป็นเวกเตอร์ค่าสุ่ม) อาจจะสะทวักที่จะใช้ชั้นเชื่อมต่อเติมที่มากกว่าการใช้ชั้นคอนโวลูชัน ถึงแม้จะมีหลายวิธีที่จะประยุกต์ใช้ชั้นคอนโวลูชันกับอินพุตเวกเตอร์ได้ก็ตาม อาทิ การจัดเรียง

ขั้นคำนวณในโครงข่ายก่อกำเนิด ยกเว้นขั้นเอาร์พุตที่แนะนำให้ใช้ไปร์บอลิกแทนเจนต์, แนะนำการใช้เรลูร์ สำหรับฟังก์ชันกราฟตุ้นของทุก ๆ ขั้นคำนวณในโครงข่ายแบ่งแยก.

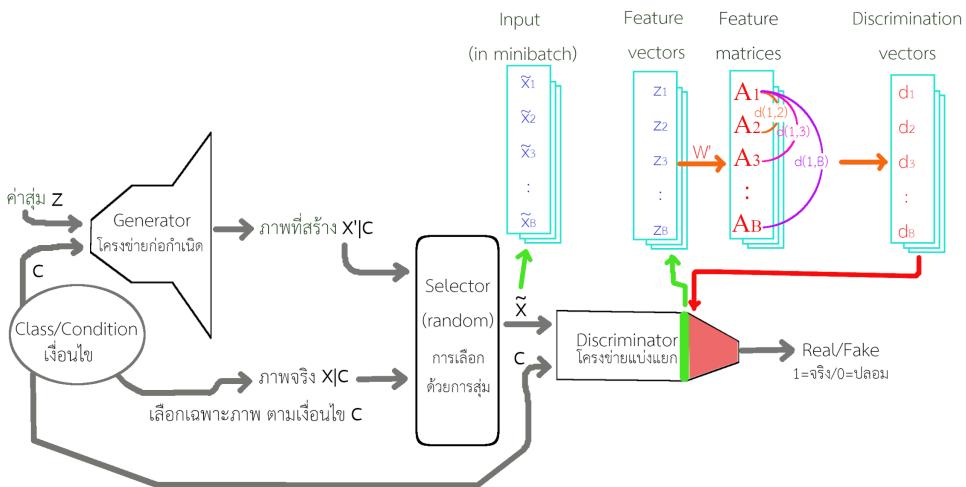
เทคนิคอื่น ๆ ที่มีรายงานว่า เป็นปัจจัยสำคัญช่วยการฝึกโครงข่ายประปักษ์เชิงสร้าง ได้แก่ การจับคู่ลักษณะสำคัญ[173], การแยกแยะหมู่เล็ก[173], การเฉลี่ยตามประวัติ[173], การทำฉลากกราบรีนทางเดียว[173], การทำหมู่เล็กเมื่อんじゃない[173], การใส่สัญญาณรบกวน[197, 6] ไปจนถึงการเปลี่ยนฟังก์ชันจุดประสงค์ (ดู [142, 7] เพิ่มเติม) เป็นต้น. เครสวอลและคณะ[45] ให้ความเห็นว่า โครงข่ายประปักษ์เชิงสร้างที่ฝึกได้ง่ายที่สุด น่าจะเป็นแบบจำลองที่เสนอโดยคณะของอาร์โจฟสกี[7] หรือของคณะของมาคอร์ชานี[125].

การจับคู่ลักษณะสำคัญ (feature matching) เปลี่ยนฟังก์ชันจุดประสงค์สำหรับโครงข่ายก่อกำเนิด เป็น  $\min_{\mathcal{G}} \|E_{\mathbf{X} \sim p_{data}}[\mathbf{f}(\mathbf{X})] - E_{\mathbf{X} \sim p_{\mathcal{G}}}[\mathbf{f}(\mathbf{X})]\|_2^2$  เมื่อ  $\mathbf{f}(\mathbf{X})$  เป็นลักษณะสำคัญที่ได้จากโครงข่ายแบ่งแยก (ค่าเวกเตอร์ของขั้นคำนวณชั้นหนึ่งที่อยู่ภายในโครงสร้างของโครงข่ายแบ่งแยก) ในขณะที่ยังใช้ฟังก์ชันจุดประสงค์แบบเดิมสำหรับโครงข่ายแบ่งแยก ( เช่น  $\max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) = E_{\mathbf{X} \sim p_{data}}[\log \mathcal{D}(\mathbf{X})] + E_{\mathbf{X} \sim p_{\mathcal{G}}}[\log(1 - \mathcal{D}(\mathbf{X}))]$ ).

การแยกแยะหมู่เล็ก (minibatch discrimination) เพิ่มสัญญาณสารสนเทศ ที่ช่วยบอกโครงข่ายแยกแยะว่าอินพุตที่ได้ เมื่อันหรือแตกต่างจากอินพุตอื่นในหมู่เล็กมากน้อยขนาดไหน. ดังนั้น โครงข่ายแยกแยะจะสามารถระบุอินพุตปลอมที่มีปัญหาโครงข่ายแยกแยะได้อย่างง่ายดาย.

รูป 7.12 แสดงกลไกของการแยกแยะหมู่เล็ก. คณะของแซลลิมันส์[173] แปลงเวกเตอร์ลักษณะสำคัญ  $\mathbf{z}_n \in \mathbb{R}^M$  (เลือกจากขั้นคำนวณในโครงข่ายแยกแยะ และ  $M$  เป็นขนาดของเวกเตอร์) ให้เป็นเมตริกซ์ลักษณะสำคัญ  $\mathbf{A}_n \in \mathbb{R}^{P \times Q}$  (ด้วยการคูณกับเทนเซอร์ค่าน้ำหนัก  $\mathbf{W}'$  ที่ปรับค่าได้ในกระบวนการฝึก) เมื่อ  $P$  และ  $Q$  เป็นจำนวนแถวและสมดุลที่ต้องการ. ความต่างระหว่างเมตริกซ์ลักษณะสำคัญ ถูกระบุด้วยเวกเตอร์  $\mathbf{d}(i, j) \in \mathbb{R}^P$  ที่มีส่วนประกอบ  $d_r(i, j) = \exp(-\|\text{row}_r(\mathbf{A}_i) - \text{row}_r(\mathbf{A}_j)\|_1)$  สำหรับ  $r = 1, \dots, P$  เมื่อ ดัชนีของตัวอย่างในหมู่เล็ก  $i, j \in \{1, \dots, B\}$  และ  $B$  คือจำนวนตัวอย่างในหมู่เล็ก. ตัวดำเนินการ  $\text{row}_r(\mathbf{A})$  หมายถึง แถวที่  $r^{th}$  ของเมตริกซ์  $\mathbf{A}$  และ  $\|[v_1, v_2, \dots, v_N]^T\|_1 = \sum_{n=1}^N |v_n|$  หรือ  $L^1$  นอร์ม (L1 norm). ความต่างระหว่างเมตริกซ์ ถูกสรุปเป็นเวกเตอร์แยกแยะ  $\mathbf{d}_i = [\sum_{j=1}^B \mathbf{d}_1(i, j), \dots, \sum_{j=1}^B \mathbf{d}_P(i, j)]^T$  และค่าเวกเตอร์  $\mathbf{d}_i$  ถูกป้อนร่วมกับเวกเตอร์ลักษณะสำคัญ  $\mathbf{z}_i$  (สำหรับตัวอย่างที่  $i^{th}$  ในหมู่เล็ก) เข้าไปสู่ขั้นคำนวณต่อไปในโครงข่ายแบ่งแยก.

การเฉลี่ยตามประวัติ (historical averaging) เป็นการเพิ่มพจน์ที่ช่วยลดการเปลี่ยนค่าพารามิเตอร์อย่างโครงสร้างของอินพุตใหม่ ให้晦มา กับการทำคอนโวโลชัน เป็นต้น.



**รูปที่ 7.12:** การแยกแยะหมู่เล็ก. แต่ละอินพุตในหมู่เล็ก  $\tilde{x}_n$  จะถูกแปลงเป็นเวกเตอร์ลักษณะสำคัญ  $z_n$  (เดิมมาจากเจ้าต์พุตของชั้นคำนวณขั้นหนึ่งของโครงข่ายแยกแยะ). การแยกแยะหมู่เล็ก จะแปลงเวกเตอร์ลักษณะสำคัญ  $z_n$  เป็นเมตริกซ์ลักษณะสำคัญ  $A_n$  และวัดความต่างระหว่างเมตริกซ์ลักษณะสำคัญนั้นเปรียบเทียบกับเมตริกซ์ลักษณะสำคัญอื่น ๆ ในหมู่เล็ก และสรุปอุปมาเป็นเวกเตอร์แยกแยะ  $d_n$ . ค่าของเวกเตอร์แยกแยะจะถูกนำไปประกอบเป็นอินพุตเสริมให้กับชั้นคำนวณต่อไปในโครงข่ายแยกแยะ. การพังทลายของภาวะที่โครงข่ายก่อทำเนิดมักสร้างเจ้าต์พุตคล้าย ๆ กันอุปมา จะทำให้เวกเตอร์แยกแยะมีขนาดเล็ก เมื่อเปรียบเทียบกับอินพุตจริงที่มีความหลากหลาย. ดังนั้นโครงข่ายแยกแยะ จะสามารถจำแนกอินพุตปลอมจากการพังทลายของภาวะได้ดีขึ้น. จากเวกเตอร์แยกแยะ และเกรเดียนต์จากโครงข่ายแยกแยะจะช่วยให้โครงข่ายก่อทำเนิดเรียนรู้เพื่อแก้การพังทลายของภาวะได้ดีขึ้น.

รุนแรงในระหว่างการฝึก เพื่อช่วยให้ระบบเข้าสู่สมดุลย์ได้ดีขึ้น. ตัวอย่างเช่น คณะของแซลลิมันส์[173] ซึ่งได้รับแรงบัลดาลใจจากทฤษฎีเกม (game theory) เพิ่มพจน์  $\|\boldsymbol{\theta} - \frac{1}{T} \sum_{i=1}^T \boldsymbol{\theta}^{(i)}\|^2$  เข้าไปในฟังก์ชันสูญเสียเดิม โดย  $\boldsymbol{\theta}$  แทนค่าปัจจุบันของพารามิเตอร์ต่าง ๆ ส่วน  $\boldsymbol{\theta}^{(i)}$  แทนค่าในสมัยฝึกที่  $i^{\text{th}}$  ของพารามิเตอร์ต่าง ๆ และ  $T$  คือจำนวนสมัยฝึกที่ผ่านมา. พจน์ที่คณะของแซลลิมันส์ เพิ่มเข้าไปเป็นระยะทางยูคลีเดียน ระหว่างค่าพารามิเตอร์ปัจจุบัน กับค่าเฉลี่ยที่ผ่านมา. การเพิ่มพจน์นี้เข้าไปในฟังก์ชันสูญเสีย ส่งผลให้กระบวนการฝึกลดการปรับค่าพารามิเตอร์อย่างมากลงได้. อย่างไรก็ตามการใช้การเฉลี่ยตามประวัติ ควรทำอย่างระมัดระวัง และควรเลือกค่าหนักเพื่อรักษาดุลย์ระหว่างค่าฟังก์ชันสูญเสียเดิม กับค่าของพจน์การเฉลี่ยตามประวัตินี้อย่างเหมาะสม.

การทำฉลาก rabine ทางเดียว (one-sided label smoothing) คือ การเปลี่ยนค่าเป้าหมายของฉลากเฉลี่ยของโครงข่ายแบ่งแยก จากเฉลี่ยว่าเป็นภาพจริง  $y = 1$  ลดลงเป็น  $1 - \epsilon$  แต่คงค่าเฉลี่ยภาพปลอม  $y = 0$  ไว้เหมือนเดิม เช่น เปลี่ยนจากเฉลี่ยภาพจริง จากค่า 1 เป็น 0.9 ( $\epsilon = 0.1$ ). เทคนิคนี้ดัดแปลงมาจากการทำฉลาก rabine (label smoothing) เพื่อป้องกันไม่ได้โครงข่ายแบ่งแยกมีความมั่นใจมากเกินไป.

การทำฉลาก rabine[132, 196] เป็นเทคนิคที่เปลี่ยนค่าเป้าหมายของฉลากเฉลี่ย เพื่อป้องกันไม่ให้แบบจำลองมีความมั่นใจมากเกินไป (over-confidence). การทำฉลาก rabine ดังเดิมออกแบบมาสำหรับการ

จำแนกกลุ่ม (multi-class classification) โดยการปรับค่าเป้าหมายของคลาสเฉลยสำหรับประเภท  $k^{th}$  เป็น  $q_k = (1 - \varepsilon)y_k + \varepsilon p(k)$  เมื่อ  $y_k$  คือค่าฉลากเฉลยของประเภท  $k^{th}$  (อยู่ในรูประหัสหนึ่งร้อน นั่นคือ  $y_k = 1$  เมื่อเฉลยคือชนิด  $k^{th}$  และ  $y_k = 0$  เมื่อเฉลยไม่ใช่ชนิด  $k^{th}$ ) และ  $p(k)$  คือการแจกแจงของข้อมูลชนิด  $k^{th}$  ส่วน  $\varepsilon$  คืออภิมานพารามิเตอร์ที่เลือกได้ตามความเหมาะสม.

เซเจดีและคณะ[196] เลือกประมาณ  $p(k)$  ด้วยการแจกแจงเอกรูป นั่นคือ ค่าฉลากเฉลย  $q_k = (1 - \varepsilon)y_k + \frac{\varepsilon}{K}$  เมื่อ  $K$  คือจำนวนประเภททั้งหมด. ตัวอย่างเช่น กรณีการจำแนก 5 ประเภท แล้วเลือก  $\varepsilon = 0.1$  ค่าเป้าหมาย จะถูกปรับเป็น 0.92 สำหรับประเภทที่ถูกต้อง และ 0.02 สำหรับประเภทที่ไม่ถูกต้อง. ดังนั้นแบบจำลองที่ถูกฝึกอย่างดีแล้วจะปรับค่าทำนายเข้ามาที่ 0.92 ซึ่งอาจตีความว่ามั่นใจมาก แต่ไม่ร้อยเปอร์เซ็นต์ มีเพื่อใจไว้บ้าง ซึ่งหากมองเชิงการคำนวณ การปรับค่าฉลากเฉลยจะช่วยป้องกันไม่ให้แบบจำลองถูกปรับค่าเข้าไปสู่ช่วงอิ่มตัว (saturation region). แบบจำลองที่ถูกปรับค่าเข้าไปสู่ช่วงอิ่มตัว จะทำให้การฝึกต้องทำได้ยาก และให้ผลคล้ายการเกิดโอลิเวอร์ฟิต. (ดูแบบฝึกหัด 7.5 เพิ่มเติมสำหรับการทำฉลากราบรื่น)

เพื่อลดการขึ้นกับข้อมูลภายในหมู่เล็กมากก็เกินไป เมื่อใช้แบบอร์ม คณะของแซลลิมันส์[173] ใช้การทำหมู่เล็กเสมือนจริง (virtual minibatch) ทำแบบอร์มกับจุดข้อมูล โดยใช้ค่าสถิติที่คำนวณจากจุดข้อมูลนั้นๆ และหมู่อ้างอิง (reference batch) ซึ่งหมู่อ้างอิง ถูกเลือกขึ้นมาก่อนการฝึก และใช้หมูนี้ตลอด (ไม่มีการเปลี่ยนแปลง). เนื่องจากการทำหมู่เล็กเสมือนจริง ทำการคำนวณมากขึ้น เพราะว่า ต้องทำการคำนวณไปข้างหน้า (forward pass) สำหรับสองหมู่เล็ก ดังนั้นคณะของแซลลิมันส์ จึงใช้การทำหมู่เล็กเสมือนจริงเฉพาะกับการฝึกโครงข่ายก่อกำเนิด

การใส่สัญญาณรบกวน (noise addition) คือการใส่สัญญาณรบกวน เช่น สัญญาณรบกวนที่มีการแจกแจงแบบเกาส์เชียน เข้าไปในทั้งภาพจริง และภาพที่สร้างจากโครงข่ายก่อกำเนิด. alonเดอบายและคณะ[197] อ้างว่าการใส่สัญญาณรบกวน ให้ผลดีกว่าการทำฉลากราบรื่นทางเดียว. การใส่สัญญาณรบกวนเข้าไปกับทั้งภาพจริงและภาพปลอม เป็นครั้งๆ กับการปรับการแจกแจงจริง กับการแจกแจงจากโครงข่ายก่อกำเนิดให้เข้ามาใกล้กันและกันมากขึ้น.

“The most difficult subjects can be explained to the most slow-witted man if he has not formed any idea of them already; but the simplest thing cannot be made clear to the most intelligent man if he is firmly persuaded that he knows

“เรื่องที่ยากที่สุดสามารถอธิบายให้คนที่หัวชาที่สุดเข้าใจได้ ถ้าเขามิ่งฝังใจคิดไปเองก่อนแล้ว. แต่เรื่องที่ง่ายที่สุดไม่อาจจะอธิบายให้คนที่ฉลาดที่สุดเข้าใจได้ ถ้าเขามั่วสับติดสิ่งที่เขาคิด โดยที่ไม่สนใจความจริงที่อยู่

already, without a shadow of doubt, what is laid before him."

ตรงหน้าเลยสักนิด."

---Leo Tolstoy

—ลีโอ โตล์สตอย

เกร็ดความรู้ รูปแบบ “ประหลาด” ของจิต เมื่อคิดนี้ อีเดเวิร์ด เคลลี่[108] เรียกว่า กายภาพนิยม (physicalism). กายภาพนิยม เป็นกรอบความคิด และมุ่งมองโลกที่มองว่า ทุกอย่างเป็นกายภาพ. นั่นรวมถึง ความคิดที่ว่า จิตก็เกิดมาจากกิจกรรมของสมอง สติและความรู้ตัวก็เป็นผลผลอยได้ จากกิจกรรมของเซลล์ประสาท และเนื่องจากแนวคิดนี้เชื่อว่า จิตมาจากการของ ดังนั้นมือตัวตาย สมองหยุดทำงาน จิตจะหายไป. และถึงแม้ว่าคนส่วนใหญ่ก็เชื่อเช่นนั้น แต่แนวคิดนี้ก็ไม่ได้ถูกพิสูจน์ หรือทดสอบอย่างเป็นทางการเลย จนกระทั่งงานศึกษาที่สำคัญ ของพาร์เนียและคณะ[150] กับทีมของแวนโนลมเมล[202].

เช่น พาร์เนีย เป็นแพทย์โรคหัวใจ ซึ่งเชี่ยวชาญในการรักษาผู้ป่วยหัวใจวาย เนื่องจากต้องการลดความเสี่ยงภาวะเจ้าชายนิทรา ของผู้ป่วย พาร์เนียจึงได้ศึกษาวิจัยเกี่ยวกับสติรู้ตัว (consciousness) และรวมไปถึง การศึกษาประสบการณ์เฉียดตาย (Near Death Experience คำย่อ NDE) ของผู้ป่วย.

ประสบการณ์เฉียดตาย เป็นประสบการณ์การรับรู้ของผู้ป่วยที่อยู่ในสถานการณ์ที่ใกล้จะตาย หรือพยายามรักษาผู้ป่วยหยุดหายใจ หัวใจหยุดเต้น และไม่มีกิจกรรมทางสมองแล้ว แต่แพทย์ พยาบาล เจ้าหน้าที่ สามารถรักษาพักลับมาได้. แม้จะบอกว่าเป็น ประสบการณ์เฉียดตาย แต่จริง ๆ แล้ว ส่วนใหญ่ผู้ที่มีประสบการณ์นี้ ก็คือ ผู้ป่วยที่ได้ตายไปแล้วในช่วงเวลาสั้น ๆ แต่ได้รับการรักษาพักลับมา สำเร็จ. ถึงแม้ จะมีรายงานประสบการณ์เฉียดตาย จากอาการป่วยหลักหลายประเภท แต่งานวิจัยของพาร์เนียและคณะจะเน้นที่ กลุ่มผู้ป่วยภาวะหัวใจวาย.

จากรายงานวิจัย[149] พาร์เนียและคณะพบว่า ในจำนวนผู้ป่วยที่รอดชีวิตและให้สัมภาษณ์ มีราว ๆ 46% ที่มีความทรงจำ โดย 9% จัดเป็นประสบการณ์เฉียดตาย (ตามเงื่อนไขที่กำหนดในงานวิจัย). มี 2% ที่รู้ตัว โดย “เห็น” หรือ “ได้ยิน” เหตุการณ์เกี่ยวกับ การรักษาอย่างชัดเจน. มีกรณีหนึ่งที่ยืนยันได้ว่า ช่วงที่ผู้ป่วยรู้ตัวอยู่นั้น ไม่พบกิจกรรมหรือสัญญาณทางสมอง.

การที่มีการรู้ตัวในช่วงที่ไม่พบกิจกรรมทางสมอง อาจบอกได้ว่า (1) จิตไม่ได้ถูกสร้างจากสมอง หรือ (2) วิธีการวัดในปัจจุบัน ไม่สามารถวัดกิจกรรมที่เกี่ยวข้องนี้ได้. การอ้างการรู้ตัวในช่วงระหว่างการรักษาพยาบาลตรวจสอบอย่างละเอียด. หนึ่งในการทดสอบก็คือ การทดสอบประสบการณ์ออกจากร่าง (out-of-body experience) ที่มักบรรยายถึง ความรู้สึกโดยอุบัติการณ์ของตัวเอง และมองเห็นภาพต่าง ๆ จากมุมสูง. คณะของพาร์เนียเตรียมการทดสอบ โดยการติดตั้งห้องที่มีโถกาลสูญที่จะเกิดเหตุการณ์ หัวใจวาย. บนห้อง จะวางรูป象牙ไว้ โดยรูปหันหน้าขึ้นpedan ซึ่งผู้ที่อยู่ในห้องไม่สามารถที่จะมองเห็นภาพในรูป. ภาพในรูปจะใช้เพื่อ พิสูจน์ความถูกต้องของคำบรรยายที่ได้จากประสบการณ์ออกจากร่าง. การทดสอบ พบว่า ผู้ที่อ้างประสบการณ์ออกจากร่างสามารถ บรรยายภาพได้อย่างถูกต้อง. การบรรยายภาพในรูปได้ถูกต้อง บอกได้ว่า (1) การรับรู้สามารถแยกออกจากร่างกายได้ และ (2) ประสบการณ์ที่บรรยายเป็นประสบการณ์จริง ไม่ใช่ความฝัน จินตนาการ หรือผลของการกิจกรรมที่สมองสร้างขึ้นมาเอง.

นอกจาก งานของพาร์เนียและคณะแล้ว ยังมีการศึกษาอื่น ๆ อีก[202, 199] ที่สนับสนุนสมมติฐานว่า (1) จิตไม่ได้เกิดจากสมอง และ (2) การรับรู้ของจิตสามารถแยกออกจากสมองได้. แม้ว่าจะมีหลักฐานสนับสนุนนักแน่น แต่ว่าการจิตวิทยาและประสาทวิทยา ส่วนใหญ่ ก็ยังเชื่อในแนวคิดเดิมอยู่ ส่วนหนึ่งก็เพราะว่า แม้หลักฐานจะบอกว่า จิตไม่ได้เกิดจากสมอง แต่ธรรมชาติของจิต การแยก ออกจากสมอง ความสัมพันธ์กับสมอง ความสัมพันธ์กับชีวิต ชีวิตหลังความตาย กลไกที่อยู่เบื้องหลัง เงื่อนไขของประสบการณ์เฉียดตาย เรื่องเหล่านี้ วงการวิทยาศาสตร์ยังไม่รู้อะไรเลย. ปัจจุบันวงการวิชาการรู้เรื่องจิตน้อยมาก และหลาย ๆ อย่างที่คิดว่ารู้ ก็อาจจะ ไม่ถูกต้อง.

``We should not be ashamed to acknowledge truth from whatever source it comes to us, even if it is brought to us by former generations and foreign peoples. For him who seeks the truth there is nothing of higher value than truth itself.''

“เราไม่ควรอายที่จะยอมรับความจริง ไม่ว่าเราได้รับมันมาจากไหน ถึงแม้ว่ามันจะมาจากคนรุ่นก่อนหรือมาจากคนต่างชาติ. สำหรับผู้แสวงหาความจริง ไม่มีอะไรมีค่ามากกว่าความจริง.”

---Al-Kindi

—อัลคินดี

**การกลับชาติตามเกิด** ในขณะที่ เรายังไม่เข้าใจความสัมพันธ์ของจิตและชีวิต สิ่งหนึ่งที่น่าสนใจ และอาจจะช่วยเติมภาพความสัมพันธ์นี้ให้เด่นขึ้น คือ การศึกษาเรื่องการกลับชาติตามเกิด (reincarnation). ภาควิชาการศึกษาการรับรู้ มหาวิทยาลัยเวอร์จิเนีย (Division of Perceptual Studies, University of Virginia) ดำเนินการศึกษาเรื่องการกลับชาติตามเกิดมากว่า 50 ปี ซึ่งทัคเกอร์และคณ[199] ได้สรุปสาระสำคัญของผลจากการศึกษาว่า เด็กที่ร่ำลึกชาติได้ มีมากกว่า 2,500 คนทั่วโลก เป็นเด็กอายุน้อยมาก ๆ (ไม่เกิน 6 ขวบ) พูดถึงชาติที่แล้ว ซึ่งเป็นชีวิตของคนธรรมดาย 70% จะพูดถึงชาติที่แล้วที่ไม่ได้ตายตามธรรมชาติ เช่น ถูกฆ่าตาย หลาย ๆ คน มีอารมณ์หรือพฤติกรรม ที่สัมพันธ์กับคนในชาติที่แล้วที่อ้างถึง เด็กบางคนมีปaineหรือทำหนินั้นแต่เกิด ที่เข้ากับแหล่งของคนในชาติที่แล้วที่อ้าง เช่น มีเด็กอินเดียหนึ่งร่ำลึกได้ว่า ชาติที่แล้ว เขาเกิดอุบัติเหตุ เครื่องจักรตัดนิวมือขวาขาดไป ตัวเด็กเองเกิดมา โดยไม่มีนิวมือขวา แต่มือซ้ายปกติ

เรื่องการกลับชาติตามเกิดไม่ได้เกี่ยวกับเชื้อชาติ หรือความเชื่อ เช่น กรณีของหนูน้อยเจมส์ ไลนิงเกอร์ (James Leininger) ที่เป็นลูกชายของครอบครัวชาวคริสต์ที่หลุยส์เซย์น่า สร้างรูปเหมือน เดิมครอบครัวไม่ได้เชื่อเรื่องการกลับชาติตามเกิดเลย. แต่ช่วงราว ๆ เจมส์อายุได้สองขวบ เจมส์ก็เริ่มฝันร้ายบ่อย ๆ. เจมส์ร้อง ดืน 时节 บนอากาศ “ไฟไหม้เครื่องบิน หนูน้อยออกไม่ได้” (“Airplane catches on fire. Little man can't get out.”) เวลากลางวัน เจมส์อาเครื่องบินมาเล่น แล้วก็เล่นทำเครื่องบินตกทำแบบนั้นขึ้น ๆ พ่อพ่อคุยกับเจมส์ เจมส์เล่าว่า เครื่องขาถูกยิงตกโดยพากผู้ปุ่น เจมส์ว่าเขาขึ้นเครื่องคอร์แซร์ (Corsair). ตอนอายุ 28 เดือน เจมส์บอกว่าเขาบินออกจากเรือ พ่อพ่อถามถึงเรือ เจมส์บอกว่าเรือชื่อ นาโนมา. ซึ่งช่วงสองครั้งที่สอง ก็มีเรือรบญี่ปุ่น เนชนาโนมา เบร์ (USS Natoma Bay) ที่ประจำการอยู่ในแปซิฟิก พ้อว่าดูรูป เจมส์ก้าวตั่รูปเครื่องบินตก ว่าดีเป็นสิบ ๆ รูป จนพ่อของเจมส์เริ่มคิดว่า หรือว่าเจมส์ร่ำลึกชาติได้จริง ๆ

ตอนเจมส์อายุ 4 ขวบครึ่ง พ่อของเจมส์ไปร่วมงานสังสรรค์ทหารเกย์ยிணของ ยูเอสເଓສ นาโนมา เบร์ ถึงได้รู้ว่า มีนักบินคนเดียวในปฏิบัติการที่ถูกฆ่าตาย นักบินคนนั้นชื่อ เจมส์ ฮูสตัน (James Huston). เมื่อคนละนักวิจัยเปรียบเทียบ สิ่งที่หนูน้อยเจมส์พูด กับประวัติของฮูสตันก็พบว่า

หนูน้อยเจมส์ ไลนิงเกอร์	เจมส์ ฮูสตัน
<ul style="list-style-type: none"> <li>เข็นต์ชื่อในรูปว่าดูว่า เจมส์ที่สาม (James 3)</li> <li>บอกว่าบินออกจากนาโนมา</li> <li>บอกว่าบินเครื่องคอร์แซร์</li> <li>บอกว่าถูกยิงตกโดยทหารญี่ปุ่น</li> <li>บอกว่าตายที่อิโรซิม่า</li> <li>บอก “เครื่องบินผ่านถูกยิงที่เครื่อง ตกลงน้ำ นั่นแหลมที่พมตาย”</li> <li>ฝันร้ายถึงเครื่องบินตกและจมน้ำบ่อย ๆ</li> <li>บอกว่าเพื่อนผู้ชาย แจ็ค ลาร์เซ่น (Jack Larsen) อุญญานั่นด้วย</li> </ul>	<ul style="list-style-type: none"> <li>เป็น เจมส์ จูเนียร์ (James, Jr.)</li> <li>เป็นนักบินของ ยูเอสເଓສ นาโนมา เบร์</li> <li>เคยบินเครื่องคอร์แซร์</li> <li>ถูกยิงตกโดยทหารญี่ปุ่น</li> <li>เป็นนักบินคนเดียวของ ยูเอสເଓສ นาโนมา เบร์ ที่ถูกยิงตกตายในปฏิบัติการอิโรซิม่า</li> <li>พยานที่เห็นเหตุการณ์รายงานว่า “ถูกยิงส่วนหน้าต่างกลางเครื่อง”</li> <li>เครื่องตกน้ำ และจมน้ำอย่างรวดเร็ว</li> <li>แจ็ค ลาร์เซ่น เป็นนักบินเครื่องที่อยู่ใกล้กับเครื่องของ ฮูสตัน วันที่ฮูสตันเครื่องตกตาย</li> </ul>

ในปี พ.ศ. 2560 หนูน้อยเจมส์ ไลนิงเกอร์ อายุ 18 ปี เรียนจบมัธยมและได้เข้าทำงานกับกองทัพเรือ.

“Your assumptions are your windows on the world.  
Scrub them off every once in a while, or the light won't come in.”

---Isaac Asimov

“ทิชชิ เป็นเส้น окหัวต่าง ที่มอง โลก ของ คุณ ขัดมันออกบ้าง ไม่อย่างนั้นแสงมันจะไม่ส่องเข้ามา.”

—ไอแซค อาซิมوف

**ประสบการณ์เฉียดตาย** นอกจากการกลับชาติตามเกิด ทัคเกอร์และคณะ[199] ยังได้สรุปงานศึกษาประสบการณ์เฉียดตาย ที่ดำเนินการมาร่วม 40 ปี ของภาควิชาการศึกษาการรับรู้ ไว้ว่า ประสบการณ์เฉียดตาย พบร้อยละ 20% ในผู้ป่วยหัวใจวาย บรู๊ส์ เกรย์สัน (Bruce Grayson) หนึ่งในคณะได้เสนอแบบจำลอง ที่ใช้วัดความเข้มข้นของประสบการณ์เฉียดตาย จากสี่ส่วนประกอบ ได้แก่ (1) การเปลี่ยนกระบวนการความคิด, (2) การเปลี่ยนสถานะของอารมณ์ความรู้สึก, (3) ลักษณะเชิงปฏิหารย์, และ (4) ลักษณะเชิงโลภอื่น.

การเปลี่ยนกระบวนการความคิด เช่น ความรู้สึกถึงการปราศจากเวลา (sense of timelessness), ความคิดที่รวดเร็วและชัดเจนกว่าปกติ, การทบทวนชีวิต (life review) ที่ผู้ป่วยรายงานว่า เห็นชีวิตที่ผ่านมาทั้งหมดฉายผ่านตา เมื่อونกับเป็นสรุปของชีวิต, ความรู้สึกว่าเข้าใจ รู้สึกร่วมกัน ฯ ชัดเจนแจ่มแจ้ง. การเปลี่ยนแปลงกระบวนการความรู้สึก เช่น ความรู้สึกถึงความสงบ ความพอใจ ความรู้สึกดี (sense of peace and well-being), รู้สึกมีความสุข (sense of joy), รู้สึกเป็นหนึ่งเดียว (sense of oneness or cosmic unity), รู้สึกถึงความรักและความอบอุ่น. ลักษณะของปฏิหารย์ เช่น การมีชีวิตชีวาของสัมผัสด้วยๆ ที่ผู้ป่วยรายงานว่า เห็นสีสันด้วย ๆ ที่ไม่เคยเห็นในโลกมาก่อน ได้ยินเสียงที่ไม่เคยได้ยินมาก่อน, การรับรู้ถึงเหตุการณ์ด้วย ๆ ที่เกิดขึ้น ระหว่างที่ผู้ป่วยหัวใจวาย, การรู้เห็นถึงอนาคต, การรู้สึกว่าได้ออกจากร่าง. ลักษณะของการสัมผัสถือกัน เช่น การได้เข้าไปในโลกอื่น, การได้พบรักกับสิ่งมีชีวิตที่ลึกลับ, การได้พบรักกับวิญญาณของคนที่ตายไปแล้ว, การได้พบรักกับจิตวิญญาณเชิงศาสนา, หรือว่า การได้ไปถึงจุดที่กลับไม่ได้ (a point of no return) ที่หากข้ามไปแล้ว จะกลับไม่ได้.

บรู๊ส์ เกรย์สัน บอกว่า ประสบการณ์เฉียดตายส่วนใหญ่จะมีลักษณะดังกล่าวผสม ๆ กัน โดยสัดส่วนแตกต่างกันไปตามแต่ละคน โดยได้ยกตัวอย่างประสบการณ์เฉียดตายของผู้หญิงคนหนึ่ง ที่เล่าไว้

“ในช่วงสองครั้ง ฉันนอนป่วยอยู่ในโรงพยาบาล. เข้าวันหนึ่ง นางพยาบาลเข้ามา และพบร้าฉันไม่มีสัญญาณชีพใด ๆ เลย. นางพยาบาลตามหามา ซึ่งหมอก็พบร้าว่าฉันตายแล้ว เช่นกัน และฉันก็ตายอยู่อย่างนั้นราวด้วย 20 นาที ตามที่หมอบอกฉันในภายหลัง.

ฉันรับรู้แสงสว่างแพร่พราว ที่ฉันรู้สึกถูกเย้ายวนตามมันไป. ตอนนั้นเมื่อฉันกับว่า เวลา�ันแตกต่างออกไป เมื่อฉันมันไม่มีเวลาอยู่ที่นั่น ไม่ว่าที่นั่น มันจะคือที่ไหน. แสงนั้นสวยงามมาก และมันก็ให้ความรู้สึกของความรักที่ปราศจากเงื่อนไข (unconditional love) และความสงบสุข. เมื่อมองไปรอบๆ ฉันก็พบว่า ฉันอยู่ในที่ที่สวยงาม เขียว เป็นเนินขึ้นลง. และฉันก็เห็นนายทหารหนุ่มกับทหารอีกหลายนายเดินเข้ามา. นายทหารหนุ่มเป็น อัลบิน ญาติคนโปรดของฉัน. ตอนนั้น ฉันไม่รู้ว่าอัลบินตายแล้ว และฉันก็ไม่เคยเห็นอัลบินในชุดเครื่องแบบมาก่อนด้วย. แต่ว่าสิ่งที่ฉันเห็น ก็ยืนยันได้จากภาพถ่ายที่ฉันได้เห็นหลายปีหลังจากนั้น. ฉันคุยกับอัลบินอย่างมีความสุขอยู่สักพัก แล้วอัลบินกับเพื่อนทหารก็เดิน靠近ออกไป. และคนข้าง ๆ ฉันก็อธิบายว่า ทหารเหล่านี้ได้รับอนุญาตให้ไปทักทายคนอื่น ๆ ที่เพียงตาย และช่วยแนะนำเขากับความตาย. ความทรงจำที่มีชีวิตชีวาต่อมา ก็คือการมองจากความสูงประมาณpedean ไปที่เตียง บนเตียง มีร่างซูบผอมนอนอยู่. มีหมอนและพยาบาลอยู่รوبرอ ๆ เตียง. ฉันตะโกนเรียก แต่ไม่มีใครได้ยินฉัน. ฉันเห็นทุกอย่างอย่างชัดเจน และรู้สึกอบอุ่น ปลอดภัย และสุขสงบ.

อีดใจต่อมา ฉันมองขึ้นไปเห็นหมอกับพยาบาลเหล่านั้น และก็รู้สึกผิดหวังอย่างแรง. ฉันพึงอกรมาจากสิ่งที่น่าเบิกบานใจ น่าพอใจอย่างที่สุด. สองวันหลังจากนั้น หมอก็เข้ามาคุยกับฉันว่า ฉันโชคดีที่ยังไม่ตาย. ฉันตอบหมอกับว่า ฉันตายไปแล้ว. หมอมองฉันแบบแปลกๆ แล้วก็นัดให้ฉันไปประเมินสภาพจิต. และฉันก็ได้เรียนรู้ที่จะหุบปากเรื่องนี้ ตั้งแต่นั้นเป็นต้นมา.”

เกรย์สันอภิปรายว่า อิทธิพลของความเชื่อและวัฒนธรรมไม่ได้มีผลต่อประสบการณ์เฉียดตาย แต่ความเชื่อและวัฒนธรรมมีอิทธิพลต่อการตีความของประสบการณ์เฉียดตาย เช่น ผู้ผ่านประสบการณ์เฉียดตายในโลกที่สามจะบรรยายถึง ถ้า หรือบ่อน้ำ แทนอุโมงค์ ที่ผู้ผ่านประสบการณ์เฉียดตายในอเมริกาบรรยาย.

คณะผู้วิจัยได้ทำการศึกษาและยืนยันถึงความน่าเชื่อถือของความทรงจำ ในประสบการณ์เฉียดตายที่คงเส้นคงวา แม้ว่าจะเปรียบเทียบการให้สัมภาษณ์ถึงประสบการณ์เฉียดตาย ที่คณะผู้วิจัยกลับไปสัมภาษณ์หลังการสัมภาษณ์เดิมที่เวลาต่างกันร่วม 10 ถึง 20 ปี. และ เพื่อตอบคำถามว่าความทรงจำในประสบการณ์เฉียดตาย เป็นความทรงจำของเหตุการณ์จริงๆ ไม่ใช่แค่ความทรงจำของจินตนาการ หรือจากภาพหลอน คณะผู้วิจัยใช้แบบสอบถามลักษณะพิเศษของความทรงจำ (memory characteristics questionnaire[96]) ที่ออกแบบมาเพื่อจำแนกแยก ความทรงจำของเหตุการณ์จริง ออกจากความทรงจำของเหตุการณ์ในจินตนาการ.

แบบสอบถามลักษณะพิเศษของความทรงจำ ทดสอบความทรงจำใน 5 แง่มุม ซึ่งสามารถแยกความทรงจำของเหตุการณ์จริง ออกจากเหตุการณ์สมมติ หรือเหตุการณ์ในจินตนาการได้อย่างน่าเชื่อถือ. แง่มุมดังๆ ได้แก่ แรงความชัดเจนของความทรงจำ (clarity of memories) ซึ่งรวมถึง รายละเอียดของสิ่งที่เห็น, แรงการรับสัมผัส (sensory aspects) เช่น เสียง กลิ่น รส, แรงของบริบท (contextual features) เช่น ความทรงจำเกี่ยวกับตำแหน่ง และ การจัดเรียงเชิงพื้นที่ (spatial arrangements), แรงความคิด และความรู้สึก (thoughts and feelings) ระหว่างที่ระลึกถึงเหตุการณ์, และแรงความเข้มข้นของความรู้สึก (intensity of feeling) ระหว่างเหตุการณ์และขณะระลึกถึง.

คณะผู้วิจัยประเมินผู้ผ่านประสบการณ์เฉียดตาย สำหรับเหตุการณ์ประสบการณ์เฉียดตาย เหตุการณ์จริงอื่นที่เกิดขึ้นในชีวิตในเวลาใกล้เคียงกัน และเหตุการณ์ในจินตนาการที่เกิดขึ้นในช่วงเวลาอื่น การทดสอบพบว่า ผู้ผ่านประสบการณ์เฉียดตาย จำประสบการณ์เฉียดตาย ได้ชัดเจน ละเอียด มีบริบท และด้วยความเข้มข้นของความรู้สึก ที่มากกว่า เหตุการณ์จริงอื่นที่เกิดในช่วงเวลาใกล้เคียงกัน. ประสบการณ์เฉียดตายถูกกระลึกถึงว่า จริงกว่าเหตุการณ์จริง ในระดับขั้นเดียวกับ ที่เหตุการณ์จริง จริงกว่า เหตุการณ์ในจินตนาการ. ในขณะที่ผู้ที่ไม่ได้มีประสบการณ์เฉียดตาย แต่ผ่านเหตุการณ์ภัยชีพในลักษณะคล้ายกัน จะรายงานความทรงจำช่วงเหตุการณ์ชีวิตนั้น ว่าจริงในระดับขั้นเดียวกับเหตุการณ์จริงอื่น ๆ เท่านั้น ไม่ได้พบว่าจริงมากกว่า.

การสันและคณบัญชีไม่พบปัจจัยใดที่จะสามารถทำนายถึงผู้ที่จะมีประสบการณ์เฉียดตายได้ ไม่ว่าจะเป็นปัจจัย อายุ เนื้อชาติ เพศ ศาสนา ความเคร่งศาสนา หรืออาการป่วยทางจิต และได้ให้ข้อสังเกตว่า แม้จะมีแนวคิดที่พิยายามเชื่อมโยง สภาวะทางสุริวิทยา ทางกายภาพ ทางชีวภาพ เข้ากับประสบการณ์เฉียดตาย แต่ที่สุดแล้ว มันก็ยังที่จะอธิบายถึง ความสามารถของสมองที่เพิ่มขึ้น การคิดและรับรู้ได้ชัดเจนขึ้น ในขณะที่สมองไม่สมบูรณ์ ไม่ว่าจากยาสลบ หรือจากภาวะหัวใจวาย.

แม้ว่าประสบการณ์เฉียดตาย อาจเป็นเงื่อนไขของการแยกกันระหว่างจิตกับสมอง หรืออาจเป็นหลักฐานสำคัญของชีวิตหลังความตาย แต่สิ่งที่น่าสนใจที่สุดเกี่ยวกับประสบการณ์เฉียดตาย ก็คือ ผลจากการผ่านประสบการณ์เฉียดตาย ผู้ที่ผ่านประสบการณ์เฉียดตายจะมีการเปลี่ยนแปลงในเชิงความเชื่อ ทัศนคติ ค่านิยม ได้แก่ มีความเชื่อและศรัทธาในเรื่องของจิตวิญญาณมากขึ้น (increase in spirituality), มีความเป็นห่วงเป็นใยมีเมตตาต่อผู้อื่นมากขึ้น (increase in sense of concern/compassion for others), ตระหนักในค่าของชีวิตมากขึ้น (increase in appreciation of life), ใช้ชีวิตมีคุณค่า มีความหมายมากขึ้น (increase in sense of meaning or purpose), มีความมั่นใจ มีความยืดหยุ่นในการรับมือกับสถานการณ์ต่าง ๆ ได้ดีขึ้น (increase in confidence and flexibility in coping skills), และเชื่อในชีวิตหลังความตาย (a belief in postmortem survival). ในขณะเดียวกัน ผู้ผ่านประสบการณ์เฉียดตาย จะลดการกลัวตายลง (decrease in fear of death), มีความสนใจในวัตถุนิยมลดลง (decreased interest in material possession), ลดความสนใจในสถานะ อำนาจ เกียรติ และชื่อเสียงลง (decreased interest in status, power, prestige, and fame), ลดความสนใจในการแกร่งแข่งชิงดิจิทัล (decreased interest in competition).

ถึงตรงนี้ อาจทำให้เกิดความว่า ถ้าผู้ฝ่าประลิบการณ์เมียดตายไม่กลัวตาย และพบว่าความตายนั้นเป็นสุขและสวยงาม ทำไม่เข้าไม่ฝ่าตัวตายไปเลย แต่กลับรักและชื่นชมคุณค่าของชีวิต และใช้ชีวิตอย่างมีความหมาย. คำถามนี้ ทัคเกอร์ เกรสัน และคณไม่ได้อภิปรายไว้. แต่หากลองคิดร่วมพิจารณาด้วยตนเองแล้ว จะพบว่าชีวิตคนนั้นเปล่า. คนที่เห็นความตาย กลับไม่กลัวตาย. คนที่กลัวตาย ไม่เคยเห็นความตาย. คนไม่กลัวความตาย กลับเข้าใจชีวิต ใช้ชีวิตได้ดี ใช้ชีวิตอย่างมีคุณค่า. แต่คนทั่วไปที่ส่วนใหญ่กลัวความตาย หลายคนเลือกใช้ชีวิตทึ้งเปล่าไป. หลายคนเลือกเล่นโทรศัพท์มือถือ แทนการมีปฏิสัมพันธ์กับคนรอบข้าง. หลายคนเลือกใช้ชีวิตเห็นแก่ตัว เลือกเป้าหมายชีวิตเป็นความร่าเริง สถานะ ชื่อเสียง ตอบสนองต่ออัตตาที่ขยายไม่รู้จบ แล้วเรียกมันว่า ความสำเร็จ. หลายคนทึ้งคุณค่าของชีวิต ทึ้งความสงบสุข เพื่อใช้ชีวิตที่ฟุ่มเฟือ. หลายคนทึ้งความกล้าที่จะทำในสิ่งที่ถูกต้อง ทึ้งความซื่อสัตย์มั่นคงที่จะยืนหยัดในหลักการที่อ้าง ทึ้งเมตตา ทึ้งปัญญา อันเป็นคุณธรรมอันสูงสุด ทึ้งโดยรู้ตัวและไม่รู้ตัว. หลายคนกลัวความตาย แต่กลับไม่เคยใช้ชีวิตให้มีคุณค่าเลย. ประเด็นนี้ก็เป็นอีกบริษัทของชีวิตที่ยากจะอธิบาย และข้อสังเกตนี้ก็ได้ถูกกล่าวไว้อย่างงดงามโดยท่านดาไลลามาองค์ที่สิบสี่ แห่งอิบเดดังนี้.

``[What surprises me most is] Man. Because he sacrifices his health in order to make money. Then he sacrifices money to recuperate his health. And then he is so anxious

“[สิ่งที่ทำให้อาตมาแบลกใจที่สุดคือ] คน เพราะว่า คนสละสุขภาพไปเพื่อหาเงิน แล้วที่หลัง ก็สละเงินไปเพื่อฟื้นฟูสุขภาพ และคนก็มัวแต่กังวล

about the future that he does not enjoy the present; the result being that he does not live in the present or the future; he lives as if he is never going to die, and then dies having never really lived."

--Tenzin Gyatso, the 14th Dalai Lama

กับอนาคต จนไม่มีความสุขกับปัจจุบัน ผลกระทบคือเขาไม่ได้อยู่ในปัจจุบันไม่ได้อยู่ในอนาคต เขาใช้ชีวิตอยู่เหมือนกับว่าเขาจะไม่มีวันตาย แล้วก็ตายไปแบบไม่เคยมีชีวิตจริง ๆ."

—เทนซิน กิยันโซ่, ดาไลลามาที่สิบสี่

### 7.3 อภิธานศัพท์

**การตรวจจับวัตถุในภาพ (object detection):** ภาระกิจการระบุชนิดของวัตถุและตำแหน่งในภาพ

**โยโล (YOLO):** แบบจำลองที่สำคัญในการตรวจจับวัตถุในภาพ ซึ่งมีแนวคิดที่สำคัญคือการกรอบปัญหาเป็นงานการหาค่าถดถอย และช่วยลดขั้นตอนการทำงานที่ซับซ้อนลงได้ ส่งผลให้แบบจำลองสามารถทำงานได้รวดเร็ว และการแก้ไขปรับปรุงก็ทำได้สะดวก

**กล่องสมอ (anchor box):** เทคนิคที่ยอมให้มีการทายวัตถุในภาพที่มีตำแหน่งซ้อนทับกันได้ โดยใช้กลไกของรูปทรงและขนาดเริ่มต้นที่ต่างกันของกล่องขอบเขต เพื่อกำหนดความรับผิดชอบต่อวัตถุ คล้ายการปักหมุดของแต่ละกล่องขอบเขต ว่ากล่องใดจะรับผิดชอบขนาดหรือรูปทรงคร่าว ๆ แบบได้

**โครงข่ายปรัปักษ์เชิงสร้าง (Generative Adversarial Networks คำย่อ GANs):** กลไกการฝึกโครงข่ายสองโครงข่าย โดยฝึกในลักษณะที่ทั้งสองโครงข่ายมีเป้าหมายขัดแย้งกัน. โครงข่ายหนึ่ง เรียกว่า โครงข่ายก่อกำเนิด ทำหน้าที่สร้างจุดข้อมูลขึ้นมา เลียนแบบจุดข้อมูลจริง ในขณะที่อีกโครงข่ายหนึ่ง เรียกว่า โครงข่ายแบ่งแยก ทำหน้าที่ตรวจสอบ ว่าจุดข้อมูลที่เห็นถูกสุ่มจากชุดข้อมูลจริง หรือถูกสร้างขึ้น. โครงข่ายก่อกำเนิด มีเป้าหมายเป็นการสร้างจุดข้อมูลเลียนแบบให้เหมือนข้อมูลจริง จนโครงข่ายแบ่งแยกจำแนกไม่ออก. โครงข่ายแบ่งแยก มีเป้าหมายเป็นการจำแนกจุดข้อมูลได้ถูกต้องมากที่สุด

**โครงข่ายแบ่งแยก (discriminator):** โครงข่ายหนึ่งในกลไกการฝึกแบบปรัปักษ์ ทำหน้าที่จำแนกจุดข้อมูลที่เห็นว่า ถูกสุ่มจากชุดข้อมูลจริง หรือถูกสร้างขึ้น

**โครงข่ายก่อกำเนิด (generator):** โครงข่ายหนึ่งในกลไกการฝึกแบบปรัปักษ์ ทำหน้าที่สร้างจุดข้อมูลเลียนแบบจุดข้อมูลจริงจนโครงข่ายแบ่งแยกจำแนกได้แยกที่สุด

**พิชคณิตเวกเตอร์ (Vector arithmetic):** สำหรับโครงข่ายประปักษ์เชิงสร้าง พิชคณิตเวกเตอร์ อ้างถึง ปฏิบัติ การเชิงเส้นที่ทำกับเวกเตอร์ลักษณะซ่อนเร้น แล้วนำเวกเตอร์ผลลัพธ์ไปเข้าโครงข่ายก่อกำเนิด เพื่อ สร้างจุดข้อมูลขึ้นมา

**ลักษณะซ่อนเร้น (latent representation):** ลักษณะของข้อมูล ที่ไม่ได้แสดง หรือกำหนดอย่างชัดแจ้ง. ใน บริบทของโครงข่ายประปักษ์เชิงสร้าง หมายถึง ค่าของเวกเตอร์ที่ใช้เป็นอินพุตของโครงข่ายก่อกำเนิด

**ปริภูมิซ่อนเร้น (latent space):** หรือปริภูมิตัวแทน (representation space) ปริภูมิของลักษณะซ่อนเร้น

**การพังทลายของภาวะ (mode collapse):** สถานการณ์ที่โครงข่ายก่อกำเนิดสร้างจุดข้อมูลคล้าย ๆ กัน เมื่อว่าจะรับอินพุตที่ต่างกัน

**คอนโวลูชันก้าวเศษ (fractionally-strided convolution):** หรือคอนโวลูชันสลับเปลี่ยน (transposed convolution) การดำเนินคอนโวลูชันด้วยการแปลงอินพุตและค่าน้ำหนักของฟิลเตอร์เป็นเมทริกซ์ โดยจัดรูปเมทริกซ์ทั้งสองให้ถูกต้อง แล้วทำการคูณเมทริกซ์อินพุตเข้ากับการสลับเปลี่ยนของเมทริกซ์ ค่าน้ำหนัก. หากเลือกอภิมานพารามิเตอร์ได้ถูกต้อง การดำเนินการเช่นนี้ อาจมองเสมอเป็นการทำ คอนโวลูชันที่ใช้ขนาดก้าวย่างเล็กกว่าหนึ่ง (เหมือนการเติมพิกเซลค่าศูนย์เข้าไประหว่างพิกเซลของ อินพุต ในกรณีข้อมูลภาพ)

**การถอดคอนโวลูชัน (deconvolution):** การถอดคอนโวลูชัน มีหลายความหมาย. ในขณะที่บางครั้ง การ ถอดคอนโวลูชันอาจหมายถึงคอนโวลูชันก้าวเศษ แต่ความหมายที่ถูกยอมรับอย่างกว้างขวาง คือการ ถอดค่าพารามิเตอร์ของโครงข่ายคอนโวลูชันย้อนกลับ เพื่อศึกษาลักษณะการทำงานของโครงข่ายคอนโว ลูชัน ว่า ฟิลเตอร์แต่ละตัวที่ใช้ในโครงข่ายคอนโวลูชัน ได้เรียนรู้เพื่อจะตรวจจับลักษณะรูปแบบ เช่น ไฟ

**การจับคู่ลักษณะสำคัญ (feature matching):** ในบริบทของโครงข่ายประปักษ์เชิงสร้าง การจับคู่ลักษณะ สำคัญ หมายถึงการตัดแปลงฟังก์ชันจุดประสงค์ของโครงข่ายก่อกำเนิด โดย แทนที่เป้าหมายการ ทำให้โครงข่ายแบ่งแยกทายผิดมากที่สุด ด้วยเป้าหมายการทำให้ความต่างระหว่าง ค่าเฉลี่ยของลักษณะสำคัญภายในโครงข่ายแบ่งแยก เมื่อเทียบกับค่าเฉลี่ยของลักษณะสำคัญเมื่อเทียบกับ ข้อมูลที่สร้างขึ้น มีค่าน้อยที่สุด. ตัวอย่างเช่น การใช้ฟังก์ชันสูญเสีย  $\text{Loss}_{\mathcal{G}} = \|E_{\mathbf{X}}[\mathbf{f}(\mathbf{X})] - E_{\mathbf{z}}[\mathbf{f}(\mathcal{G}(\mathbf{z}))]\|^2$  เมื่อ  $\mathbf{X}$  คือข้อมูลจริง และ  $\mathbf{z}$  แทนเวกเตอร์ค่าสุ่ม ส่วน  $\mathcal{G}(\cdot)$  คือการคำนวณของ โครงข่ายก่อกำเนิด และ  $\mathbf{f}(\cdot)$  คือลักษณะสำคัญที่ได้จากโครงข่ายแบ่งแยก

การแยกแยะหมู่เล็ก (minibatch discrimination): การเพิ่มสัญญาณข้อมูลที่บอกรความแตกต่างระหว่าง

จุดข้อมูลใด ๆ กับจุดข้อมูลอื่น ๆ ภายในหมู่เล็กเดียวกัน เพื่อลดบรรเทาปัญหาการพังทลายของภาวะ

การทำฉลากරาร์น (label smoothing): การปรับค่าเป้าหมายของฉลากเฉลย เพื่อบรรเทาปัญหาที่แบบ

จำลองมีความมั่นใจสูงเกินไป

การทำหมู่เล็กเสมือนจริง (virtual minibatch): การใช้หมู่อ้างอิงที่เลือกมาก่อนกระบวนการฝึก เพื่อใช้การ

คำนวณแบบนอร์ม ร่วมกับจุดข้อมูลที่สนใจ

## 7.4 แบบฝึกหัด

``A single act of kindness throws out roots in all directions, and the roots spring up and make new trees.''

---Amelia Earhart

“ความเมตตาปราณีเพียงครั้งกี่หยิ่ง รากไปทุกทิศทาง และรากก็จะงอกงาม ออกมานเป็นต้นใหม่.”

—เอมีลีย์ แวร์ยาร์ต

### แบบฝึกหัด 7.1

จะเลือกการประยุกต์ใช้โครงข่ายคอนโวโลจี้น์ที่สนใจ แล้วศึกษาวรรณกรรมที่เกี่ยวข้อง โดยให้เลือกบทความวิจัยที่เกี่ยวข้องไม่น้อยกว่า 20 บทความ แล้วสำหรับแต่ละบทความ ให้อภิปรายถึง จุดประสงค์ ความคาดหมาย ปัญหาที่ต้องการแก้ ความท้าทายที่เกี่ยวข้อง วิธีการที่นำเสนอ และผลลัพธ์ รวมถึงอภิปรายจุดเด่น และประเด็นอื่น ๆ ที่เห็นว่า่น่าสนใจในบทความ.

นอกจากนั้น จงอภิปรายความสัมพันธ์กับการประยุกต์แบบอื่นที่มีลักษณะใกล้เคียงกัน (อาจต้องทำการศึกษาวรรณกรรมเพิ่มเติม ให้ฝึกการศึกษาวรรณกรรมอย่างกว้างขวาง). ตัวอย่างเช่น หากเลือกการรู้จำใบหน้า (face recognition) อาจอภิปรายความสัมพันธ์ ความเหมือน ความต่าง กับการประยุกต์ใช้สำหรับ การจำแนกชนิดวัตถุในภาพ (image classification) หรือการพิสูจน์ยืนยันใบหน้า (face verification) หรือการรู้จำอารมณ์จากใบหน้า (facial expression recognition) เป็นต้น.

### แบบฝึกหัด 7.2

จะเลือกบทความวิจัยในแบบฝึกหัด 7.1 มา 5 บทความ แล้วสำหรับแต่ละบทความ (นอกจากประเด็นในแบบฝึกหัดที่ 7.1) ให้อภิปรายถึง ข้อมูล วิธีการปฏิบัติ การทดลอง และวิธีการประเมินผล.

### แบบฝึกหัด 7.3

จากแบบฝึกหัดที่ 7.2 จะศึกษาวิธีการประยุกต์ใช้ และลงมือปฏิบัติ ทดลอง และเปรียบเทียบผลที่ได้ กับผลที่รายงานในวรรณกรรม. ในการลงมือปฏิบัติ อาจปรับลดความยากของปัญหาลงได้ตามความเหมาะสม รวมถึงอาจศึกษาวิธีการปฏิบัติและโปรแกรมจากอินเตอร์เน็ต

ตัวอย่าง หากเลือกการรู้จำใบหน้า และสนใจ FaceNet[179] อาจใช้คำค้นหา เช่น “facenet code” และอาจเลือกชุดข้อมูลที่ง่ายขึ้น หรือเลือกข้อมูลขนาดเล็กลง หรือใช้แบบจำลองที่เล็กลง เพื่อให้การฝึกทำได้รวดเร็วขึ้น.

### แบบฝึกหัด 7.4

จงทบทวนเรื่องโครงข่ายปรับกษ์เชิงสร้าง (และอาจศึกษาเพิ่มเติม ถ้าจำเป็น) และอภิปรายถึงแนวทางวิธี หรือกลไก เพื่อจะอนุมานการแยกแยะร่วม  $p(\mathbf{X}, \mathbf{C})$  เมื่อ  $\mathbf{X}$  คือข้อมูลต้น เช่น ภาพ และ  $\mathbf{C}$  คือข้อมูลตาม เช่น ประเภทของวัตถุในภาพ โดยอาศัยแนวทางของโครงข่ายปรับกษ์เชิงสร้าง. การอภิปราย อาจเริ่มจากข้อคิดเห็นหรือคำถาม เช่น หากโครงข่ายปรับกษ์เชิงสร้าง สามารถเรียนรู้การแยกแยะ  $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$  ได้แล้ว และในเมื่อ  $\mathbf{z}$  ก็สู่มสร้างขึ้นมาเอง (อาจจะจากสุ่มจากการแยกแยะเอกรูป หรือการแยกแยะเกาส์เจี้ยน) ส่วนเงื่อนไขหรือข้อมูลตาม  $\mathbf{C}$  ซึ่งมักอยู่ในปริภูมิที่มีจำนวนมิติไม่มาก ก็อาจสามารถประมาณการแยกแยะจากข้อมูลที่มีได้ไม่ยากนัก ดังนั้น จากการแยกแยะ  $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$  เราก็จะสามารถใช้ทฤษฎีของเบล์ เพื่ออนุมานการแยกแยะร่วม  $p(\mathbf{X}, \mathbf{C})$  ได้. ทำไม่การหาอนุมานการแยกแยะร่วม  $p(\mathbf{X}, \mathbf{C})$  หรือแม้แต่การหา  $p(\mathbf{X})$  เมื่อ  $\mathbf{X}$  เป็นภาพ เช่น ภาพถ่ายทิวทัศน์ทั่วไป ถึงเป็นปัญหาที่ยากมาก<sup>8</sup> หากเป็นไปได้?

ตั้งกลุ่ม สามคำถามและอภิปรายข้อคิดเห็นลักษณะเช่นนี้ ความท้าทาย ความเสี่ยง แนวทางและกลไกที่จะลดหรือบรรเทาปัญหาและความเสี่ยงต่าง ๆ. ยกตัวอย่าง หรือหากเหมาะสม อาจจะลองออกแบบการทดลองเล็ก ๆ ง่าย ๆ เพื่อพิสูจน์ ยืนยัน หรือหักล้าง.

## แบบฝึกหัด 7.5

พิจารณาข้ออภิปรายถึงวิธีการทำลากกราบรีน ดังนี้ วิธีการทำลากกราบรีน[196] ปรับค่าเป้าหมายของฉลากเป็น  $q_k = (1 - \varepsilon)y_k + \varepsilon p(k)$  เมื่อ  $y_k$  คือค่าฉลากเฉลยในรูปหนึ่งร้อนของประเภท  $k^{th}$  และ  $p(k)$  คือการแยกแยะของข้อมูลชนิด  $k^{th}$ . สังเกตว่า วิธีการทำลากกราบรีน ปรับที่ค่าเป้าหมายของฉลากเฉย ไม่ได้แก้ไขการคำนวณฟังก์ชันกระตุนในแบบจำลอง. หากพิจารณาประเด็นนี้ร่วมกับฟังก์ชันสูญเสียสำหรับภาระกิจการจำแนกประเภทแบบหลายกลุ่ม ซึ่งคือ  $\text{Loss} = -\sum_k y_k \log \hat{y}_k$  เมื่อ  $y_k$  คือค่าเป้าหมายเฉลย และ  $\hat{y}_k$  คือค่าที่คำนวณ จะพบว่า กรณีไม่ทำลากกราบรีน (กรณีดั้งเดิม) ฟังก์ชันสูญเสียสามารถคำนวณโดย  $\text{Loss} = -\log \hat{y}_{k^*}$  เมื่อ  $k^*$  คือฉลากของประเภทที่เฉลย เพราะ  $y_{k^*} = 1$  และ  $y_{k \neq k^*} = 0$ .

แต่กรณีทำลากกราบรีน ฟังก์ชันสูญเสียไม่สามารถย่อรูปดังข้างต้นได้ และหากทำการคำนวณ  $\text{Loss} = -\sum_k y_k \log \hat{y}_k$  โดยตรง ซึ่งอาจเขียนเป็น  $\text{Loss} = -(1 - \varepsilon + \frac{\varepsilon}{K}) \log \hat{y}_{k^*} - \sum_{k \neq k^*} \frac{\varepsilon}{K} \log \hat{y}_k$  เมื่อ  $K$  คือจำนวนของประเภททั้งหมด แล้วอาจเกิดปัญหาการคำนวณเชิงเลขได้. ดังเช่น กรณีที่  $\hat{y}_k$  ตัวใดตัวหนึ่งมีค่าใกล้กับศูนย์มาก ๆ ( $\log 0 \rightarrow -\infty$ ) ซึ่งอาจจะทำให้การคำนวณไม่มีเสถียรภาพ. ปัญหานี้ แม้จะเกิดยากเนื่องจากแบบจำลองมีแนวโน้มที่จะถูกผึ้งให้  $\hat{y}_k$  ปรับเข้าหาเป้าหมาย เช่น  $\hat{y}_k \rightarrow \frac{\varepsilon}{K}$  และค่า  $\varepsilon$  ไม่เล็กจนเกิน

<sup>8</sup> คำใบ้ คำกล่าวว่าแบบจำลองเรียนรู้การแยกแยะ  $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$  กับการที่แบบจำลองสามารถให้ค่า  $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$  ออกมาได้ นั้นต่างกัน. ถึงที่โครงข่ายปรับกษ์เชิงสร้างให้ออกมาจริง ๆ คืออะไร? การแยกแยะ (?) ความน่าจะเป็น (?) หรือเพียงค่าคาดหมาย  $E[\mathbf{X}|\mathbf{C}, \mathbf{z}]$  หรืออะไร?

ไป. แต่การปรับเปลี่ยนนี้ ก็เพิ่มความเสี่ยงขึ้นมากจากกรณีตั้งเดิม (ที่มีแต่  $\log \hat{y}_{k^*}$  ซึ่ง  $\hat{y}_{k^*} \rightarrow (1 - \varepsilon + \frac{\varepsilon}{K})$  ที่มีค่ามากใกล้ ๆ หนึ่ง).

นอกจากนั้น อีกประเด็นหนึ่งสำหรับการใช้วิธีการทำclassificationในทางปฏิบัติ หากการทำclassification รีบถูกนำไปใช้ในโปรแกรมโดยไม่ระวัง เช่น อาจอาศัยโปรแกรมหรือโครงสร้างเดิมจากฟังก์ชันสูญเสีย ซึ่งคำนวณ  $\text{Loss} = -\log \hat{y}_{k^*}$  แทน  $\text{Loss} = -\sum_k y_k \log \hat{y}_k$  และ การการทำclassification อาจผิดเพี้ยนจากแนวคิดตั้งเดิมได้ เช่น แทนที่จะคำนวณ  $\text{Loss} = -(1 - \varepsilon + \frac{\varepsilon}{K}) \log \hat{y}_{k^*} - \sum_{k \neq k^*} \frac{\varepsilon}{K} \log \hat{y}_k$  แต่ด้วยการใช้โปรแกรมเดิม อาจทำให้สิ่งที่คำนวณจริงเป็น  $\text{Loss} = -(1 - \varepsilon + \frac{\varepsilon}{K}) \log \hat{y}_{k^*}$  ซึ่งผิดเพี้ยนไปจากแนวคิดของการการทำclassificationที่ใช้เดิมและคงจะ[196]เสนอ (แต่อาจจะทำงานได้ และอาจจะไปคล้ายกับแนวคิดของการการทำclassificationทางเดียว)

จึงศึกษาโปรแกรมการทำclassificationที่ค้นหาได้จากอินเตอร์เน็ต ทดลองใช้และสังเกตการทำงานของวิธีการทำclassification วิเคราะห์ และให้ข้อคิดเห็นเพิ่มเติมจากข้ออภิรายข้างต้น (อาจเห็นด้วย เห็นแย้ง หรือเห็นต่าง) พิรุณให้เหตุผล และอาจยกตัวอย่างประกอบ เพื่อสนับสนุน รวมถึงอภิรายสถานการณ์ต่าง ๆ ว่า หากเกิดขึ้นจริง จะมีผลดี ผลเสียอย่างไรบ้าง และสำหรับผลเสียจะมีวิธีจัดการ แก้หรือบรรเทาปัญหาอย่างไรบ้าง

## แบบฝึกหัด 7.6

จากแบบฝึกหัด 7.5 ที่อภิรายกรณีการจำแนกกลุ่ม จงอภิรายประเด็นข้อดี ข้อเสีย โอกาส และความเสี่ยง ในทางปฏิบัติ เมื่อนำวิธีการทำclassificationไปใช้ในกรณีการจำแนกค่าทวิภาค (binary classification) รวมถึงศึกษางานของคณะของแซลลิมันส์[173] สำหรับเหตุผลที่เลือกใช้วิธีการทำclassificationทางเดียว ทั้งเหตุผล ข้อดี ข้อเสีย โอกาส และความเสี่ยง สำหรับการฝึกโครงข่ายประปักษ์เชิงสร้าง และการนำแนวคิดไปใช้ในกรณีทั่วไป.



ภาค iii

การรื้อจำรูปแบบเชิงลำดับ



## บทที่ 8

### แบบจำลองสำหรับข้อมูลเชิงลำดับ

“Failure comes only when we forget our ideals and objectives and principles.”

—Jawaharlal Nehru

“ความล้มเหลวมาเกิดแต่เฉพาะตอนที่เราลืม  
อุดมการณ์ เป้าหมาย และหลักการของเรา。”

— Jawaharlal Nehru

รูปแบบของข้อมูลในเนื้อหาที่ผ่าน ๆ มา เป็นลักษณะที่เป็นอิสระในตัวเอง นั่นคือ แต่ละจุดข้อมูลมีความหมายสมบูรณ์แบบในตัวเอง หรือหากเจาะจงลงไป อาจกล่าวว่า ที่ผ่านมา แต่ละจุดข้อมูลเป็นอิสระต่อกันและมีการแจกแจงเหมือนกัน ที่เรียกว่า ไอ.ไอ.ดี. (independent and identically distributed คำย่อ i.i.d.).  
อย่างไรก็ตาม มีข้อมูลหลายประเภทที่จุดข้อมูลต่าง ๆ มีความสัมพันธ์ระหว่างกัน. เนื้อหาในบทนี้อภิปรายแนวทางและแบบจำลอง ที่ออกแบบมาสำหรับการทำงานอย่างมีประสิทธิภาพกับข้อมูลเชิงลำดับ.

#### 8.1 ข้อมูลเชิงลำดับ

ข้อมูลประเภทที่จุดข้อมูลมีความสัมพันธ์เชิงลำดับระหว่างกัน จะเรียกว่า **ข้อมูลเชิงลำดับ** (sequential data). นั่นคือ แนวว่าจุดข้อมูลจะมีค่าเหมือนกัน แต่หากลำดับที่ปรากฏต่างกัน ก็อาจทำให้ความหมายต่างกัน หรือแม้จุดข้อมูลหนึ่งจะมีค่าเท่าเดิมและปรากฏที่ตำแหน่งเดิม แต่หากจุดข้อมูลอื่น ๆ ในลำดับเปลี่ยนค่า ก็อาจจะทำให้ความหมายนั้นเปลี่ยนไปได้.

ข้อมูลหลากหลายชนิด เป็นข้อมูลเชิงลำดับ และงานการรู้จำรูปแบบกับข้อมูลเชิงลำดับ ก็มีหลากหลายประเภท เช่น การทำนายข้อมูลทางการเงิน (financial data prediction) ซึ่งอาจรับอินพุตเป็นลำดับของราคาปิดของหุ้นในวันก่อน ๆ และทำนายราคากับของวันถัดไป, การรู้จำเสียงพูด (speech recognition) ที่รับอินพุตเป็นสัญญาณเสียง (ลำดับค่าแอมพลิจูดต่อเวลา) และให้อาร์พุตเป็นลำดับของคำ, ระบบแต่งเพลงอัตโนมัติ (music generation) ที่อาจรับอินพุตเป็นประเภทของเพลง และให้อาร์พุตเป็นลำดับของโน้ตดนตรี, การรู้จำ

รูปแบบสัญญาณคลื่นไฟฟ้าหัวใจ (ECG pattern recognition) ที่รับอินพุตเป็นสัญญาณคลื่นไฟฟ้าหัวใจ (ลำดับของพลิจูดหลาย ๆ ช่องสัญญาณต่อเวลา) และอาจจะให้เอาต์พุตเป็นค่าระบุว่า ปกติหรือผิดปกติ หรืออาจจะระบุตำแหน่งและชนิดที่ผิดปกติอีกด้วย, ระบบวิเคราะห์ลำดับดีเอ็นเอ (DNA sequence analysis) ที่รับอินพุตเป็นลำดับของชนิดฐานนิวคลีโอไทด์ และอาจจะให้เอาต์พุตเป็นตำแหน่งต่าง ๆ ในลำดับที่สัมพันธ์กับโปรตีนที่สนใจ, การรู้จำกิจกรรมจากวีดีโอ (video activity recognition) ที่รับอินพุตเป็นข้อมูลวีดีโอ (ลำดับของภาพต่าง ๆ ตามเวลา) และให้เอาต์พุตเป็นฉลากของกิจกรรม, การจำแนกอารมณ์ (sentiment classification) ที่อาจรับอินพุตเป็นข้อความ (ลำดับของคำต่าง ๆ) และให้เอาต์พุตเป็นคะแนนประเมินความพอใจ, การแปลภาษาอัตโนมัติ (machine translation) ที่รับอินพุตเป็นข้อความในภาษาหนึ่ง (ลำดับของคำในภาษาต้นทาง) และให้เอาต์พุตเป็นข้อความในอีกภาษาหนึ่ง (ลำดับของคำในภาษาเป้าหมาย). สำหรับข้อมูลเชิงลำดับบางชนิด ตัวลำดับอาจเป็นตัวแทนของเวลา เช่น สัญญาณเสียง หรืออาจเป็นเพียงลำดับที่ไม่ได้เกี่ยวกับเวลา ก็ได้ เช่น ลำดับของคำต่าง ๆ ในข้อความ. อย่างไรก็ตาม เพื่อความสะดวก เนื้อหาในบทนี้อาจใช้คำว่า เวลา ใน การอ้างถึงลำดับ. รูป 8.1 แสดงตัวอย่างข้อมูลเชิงลำดับ และตัวอย่างภาระกิจการรู้จำรูปแบบที่เกี่ยวข้อง.

หากจำแนกภาระกิจออกตามลักษณะอินพุตและเอาต์พุต การรู้จำรูปแบบเชิงลำดับ อาจจำแนกได้ดังนี้.

- (1) ประเภทแรกคือ ภาระกิจที่อินพุตเป็นข้อมูลลำดับ  $\{\mathbf{x}_n\}_{n=1,\dots,N}$  เมื่อ  $N$  คือจำนวนจุดข้อมูลทั้งหมดในลำดับ แต่เอาต์พุตไม่ได้เป็นข้อมูลลำดับ  $y \in \mathbb{R}^K$  เมื่อ  $K$  คือจำนวนมิติของเอาต์พุต. ในรูป 8.1 ตัวอย่างในกลุ่มนี้คือ การทำนายข้อมูลทางการเงิน, การรู้จำรูปแบบสัญญาณคลื่นไฟฟ้าหัวใจ, การรู้จำกิจกรรมจากวีดีโอ และการจำแนกอารมณ์. ระบบวิเคราะห์ลำดับดีเอ็นเอ ก็อาจจัดอยู่ในกลุ่มนี้ หากให้เอาต์พุตอีกมาเป็นค่าตัวนึงของจุดเริ่มต้นและจุดจบของส่วนลำดับที่สนใจ. (2) ประเภทสองคือ ภาระกิจที่อินพุตเป็นข้อมูลลำดับ  $\{\mathbf{x}_n\}_{n=1,\dots,N}$  และเอาต์พุตก็เป็นข้อมูลลำดับ  $\{\mathbf{y}_n\}_{n=1,\dots,N}$  โดยทั้งสองลำดับมีจำนวนจุดข้อมูลในลำดับเท่ากัน. รูป 8.1 ไม่ได้แสดงตัวอย่างของภาระกิจในกลุ่มนี้. ตัวอย่างภาระกิจในกลุ่มนี้ ได้แก่ การระบุหมวดคำ (Part-Of-Speech Tagging) ที่วิเคราะห์ข้อความแล้วระบุหมวดคำของคำทุกคำในข้อความ ว่าอยู่ในหมวดคำใดในกลุ่ม (ซึ่งมักประกอบด้วย คำนาม, คำกริยา, คำคุณศัพท์, คำกริยาवิเศษณ์, คำบุพบท, คำสันธาน, คำสรรพนาม, คำอุทาน และคำนำหน้านาม สำหรับภาระกิจการระบุหมวดคำในภาษาอังกฤษ) และการรู้จำชื่อเฉพาะ (Named-Entity Recognition) ที่ต้องการระบุว่าคำไหนบ้างในข้อความที่เป็นชื่อเฉพาะ และเป็นชื่อเฉพาะของสิ่งประเภทใด (ซึ่งมักกำหนดประเภทที่สนใจไว้ เช่น ชื่อคน, ชื่องค์กร หน่วยงาน หรือบริษัท, ชื่อตราสินค้า, ชื่อสถานที่, เวลา, ปริมาณหรือจำนวน) เป็นต้น. (3) ประเภทสามคือ ภาระกิจที่อินพุตเป็นข้อมูลลำดับ  $\{\mathbf{x}_n\}_{n=1,\dots,N}$  และเอาต์พุตก็เป็นข้อมูลลำดับ  $\{\mathbf{y}_n\}_{n=1,\dots,M}$  เมื่อ  $M$  เป็นจำนวนจุดข้อมูลในลำดับ

ของเอาร์พุต แต่ลำดับทั้งสองอาจมีจำนวนจุดข้อมูลในลำดับไม่เท่ากัน. ตัวอย่างที่แสดงในรูป 8.1 คือ การแปลภาษาอัตโนมัติ ที่ข้อความของภาษาต้นทาง อาจมีจำนวนคำต่างจากข้อความของภาษาปลายทาง. (4) ประเภทสี่คือ ภาระกิจที่อินพุตไม่ใช่ข้อมูลลำดับ แต่เอาร์พุตเป็นข้อมูลลำดับ. ตัวอย่างที่แสดงในรูป 8.1 คือ ระบบแต่งเพลงอัตโนมัติ ที่อินพุตอาจจะเป็นເງິນເຕັກ หรืออาจจะเป็นຄ່າສເກລ່າຮ້ ແຕ່ໃຫ້ເອົາບຸດອອກມາเป็นลำดับของໂນີຕົນຕຣີ. ສຸດທ້າຍ ທາກภาระກີຈີທີ່ທີ່ອີນພຸດແລະເອົາບຸດໄຟ່ຂໍ້ຂໍ້ມູນລຳດັບ ພາຮະກິຈນີ້ໄຟ່ຈັດອູຍືໃນກາຮູ້ຈຳກູປະບົບເຊີງລຳດັບ ແລະ ໂດຍທີ່ໄປ ພາຮະກິຈປະເກາທນີ້ຈະສາມາດດຳເນີນກາຮັດໄດ້ ໂດຍວິທີກາຮັດຕ່າງໆ ທີ່ໄດ້ອົກປາຍໄປໃນບຖກອນ ແລະ ທີ່ໄດ້

ข้อมูลເຊີງລຳດັບ ອາຈແບ່ງອອກໄດ້ເປັນສອງປະເກາທ ຄື່ອ ข້ອມູນລືເຊີງລຳດັບແບບຄົງທີ່ ກັບ ข້ອມູນລືເຊີງລຳດັບແບບໄມ່ຄົງທີ່. ข້ອມູນລືເຊີງລຳດັບແບບຄົງທີ່ (stationary sequential data) ທີ່ແມ່ຄ່າຕ່າງໆ ຂອງຈຸດຂໍ້ອມູນອາຈັດພັນແປປໄປຕາມເວລາ ແຕ່ກາຮແຈກແຈງເບື້ອງໜັງລຳດັບຂໍ້ອມູນນັ້ນຄົງທີ່ ໄມ່ໄດ້ມີກາຮປັບປຸງແປ່ງຕາມເວລາ. ສ່ວນກົນຂອງຂໍ້ອມູນລືເຊີງລຳດັບແບບໄມ່ຄົງທີ່ (nonstationary sequential data) ກາຮແຈກແຈງເບື້ອງໜັງລຳດັບຂໍ້ອມູນນັ້ນມີກາຮປັບປຸງແປ່ງຕາມເວລາດ້ວຍ. ທາກອົບຍາຍຈ່າຍ ແລະ ອາຈເປົ້າຍເຕີບຈາກ ຕ້ວອຍ່າງຂອງຂໍ້ອມູນບຣິມານນໍາຟັນໃນແຕ່ລະວັນ ຕລອດທລາຍ ແລະ ປຶກທີ່ຮະຢະເວລາເຫັນນັ້ນຟ້າຟັນຕົກຕ້ອງຕາມຄຸງກາລ ເປັນຕ້ວອຍ່າງຂອງຂໍ້ອມູນລືເຊີງລຳດັບແບບຄົງທີ່. ສ່ວນຕ້ວອຍ່າງຂອງຂໍ້ອມູນບຣິມານນໍາຟັນໃນແຕ່ລະວັນ ໃນທລາຍ ແລະ ປຶກທີ່ຮະຢະເວລາເຫັນນັ້ນຟ້າຟັນຕົກຕ້ອງຕາມຄຸງກາລທີ່ຄຸນເຄຍ. ກາຮປັບປຸງແປ່ງຕາມເວລາດ້ວຍ ສ່ວນຕ້ວອຍ່າງຂອງຂໍ້ອມູນບຣິມານນໍາຟັນໃນແຕ່ລະວັນ ໃນທລາຍ ແລະ ປຶກທີ່ຮະຢະເວລາເຫັນນັ້ນຟ້າຟັນຕົກຕ້ອງຕາມຄຸງກາລເອງຍ່າງມາກ ທີ່ອາຈຈະເກີດຈາກທລາຍ ແລະ ສາເຫຼຸດຮຸມດື່ງກາວໂລກຮັອນສພາພຸມືອກາສປະເປົ້າຍແປ່ງຕາມເວລາຂອງຂໍ້ອມູນ ເກີດຈາກກະບວນກາຮເບື້ອງໜັງທີ່ມີກາຮປັບປຸງແປ່ງຕາມເວລາດ້ວຍ. ເນື້ອຫາໃນບຖນີ້ໃໝ່ສົມມືຕູ້ານຂອງກົນຂໍ້ອມູນລືເຊີງລຳດັບແບບຄົງທີ່ເປັນຫັກ ຍກເວັນແຕ່ຈະຮະບູເປັນເອົ້ນ.

ກາຮສ້າງແບບຈຳລອງສໍາຮັບຂໍ້ອມູນລືເຊີງລຳດັບ ສາມາດທຳໄດ້ທລາຍແນວທາງ. ທ້ວ່າຂ້ອ 8.2 ແລະ 8.3 ອົກປາຍແນວທາງກາຮໃຊ້ຄວາມນໍາຈະເປັນດ້ວຍແບບຈຳລອງມາრົກໂພ ແລະ ແບບຈຳລອງມາຮົກໂພໜ່ອເຮັນ. ແນວທາງຂອງໂຄຮງໝາຍປະສາທເວີນກັບ ຈະຖຸກວິປາຍໃນທ້ວ່າຂ້ອ 9.2.

## 8.2 ແບບຈຳລອງມາຮົກໂພ

ກາຮສ້າງແບບຈຳລອງສໍາຮັບຂໍ້ອມູນລືເຊີງລຳດັບ ໂດຍຮອງຮັບຄວາມສັນພັນຮື່ອງລຳດັບຮ່ວ່າງລຳດັບຕ່າງໆ ອ່າງສົມບູຮັນ ລາຍການທີ່ກຳນົດໃຫ້ກາຮສ້າງແບບຈຳລອງມາຮົກໂພ ແລະ ແບບຈຳລອງມາຮົກໂພໜ່ອເຮັນ. ຖ້າກາຮສ້າງແບບຈຳລອງມາຮົກໂພ ເປັນຫັກ ຍກເວັນແຕ່ຈະຮະບູເປັນເອົ້ນ.



**รูปที่ 8.1:** ตัวอย่างการรู้จำรูปแบบเชิงลำดับ. ภาระกิจ (จากแควร์บลังล่าง) ได้แก่ การนำมายังการเงิน อินพุตเป็นราคากิตต่อวัน, การรู้จำเสียงพูด อินพุตเป็นสัญญาณเสียง ที่เป็นข้อมูลแอมพลิจูดต่อเวลา, ระบบแต่งเพลงอัตโนมัติ เอ้าต์พุตเป็นลำดับของโน็ตดนตรี (โดยอินพุตอาจเป็นค่าสเกลาร์หรือເວກເຕົອ ທີ່ເປັນຕົວແທນຮະບຸລັກນະຂອງເພິ່ນ), การรู้จำรูปแบบสัญญาณคลื่นไฟฟ้า ຫົວໃຈ อินพุตเป็นลำดับແອມພລິຈຸດຫລາຍ ຈະອ່ານສัญญาณຕ່ອງເວລາ, การວິເຄາະທີ່ລຳດັບດີເລື່ອນເອົາອີນພຸດເປັນລຳດັບຂອງໝັ້ນດຽວນິວຄລືໂໄທດ໌, การรู้จำຈິກຮົມຈາກກົງໂອ ອີນພຸດເປັນພາພີຕ່ອງເວລາ, การຈຳແນກອາຣມັນ ອີນພຸດເປັນລຳດັບຄໍາ ແລະການແປ່ງພາຫາອັດໂນມັຕີ ທີ່ທັງອີນພຸດແລະເອົາຕົວຕ່າງກັ່ນເປັນข้อมูลລຳດັບຂອງຄໍາ. ສັງເກຕະກິຈຈາກເກີ່ວຂອງກັບຂໍ້ມູນເຊີງລຳດັບ ໂດຍຮັບອີນພຸດເປັນຂໍ້ມູນເຊີງລຳດັບ ພວກເຮົາໃຫ້ເອົາຕົວອົກມາເປັນຂໍ້ມູນເຊີງລຳດັບ ບໍ່ຮົວທັງສອງຍ່າງ. ຂໍ້ມູນເຊີງລຳດັບ ອາຈນີ້ລຳດັບທີ່ມີຄວາມໝາຍາຕາມລຳດັບເວລາຈິງ ຈະເຊັ່ນລຳດັບຮາຄາປັດຕ່ວັນ, ລຳດັບແອມພລິຈຸດຕ່ອງເວລາ, ລຳດັບໂນ້ຕົດຕົວຕ່ອງເວລາ ແລະ ລຳດັບພາພີຕ່ອງເວລາ ບໍ່ຮົວຈາກມີລຳດັບທີ່ໄມ້ໄດ້ມີຄວາມໝາຍາກັບເວລາ ເຊັ່ນ ລຳດັບຂອງໝັ້ນດຽວນິວຄລືໂໄທດ໌ ແລະ ລຳດັບຂອງຄໍາໃນຂໍ້ຄວາມ.

จุดข้อมูลลำดับต่อไป  $\mathbf{x}_{T+1}$  ของข้อมูลชุดลำดับ  $\{\mathbf{x}_t\}_{t=1,\dots,T}$ . การใช้แบบจำลองความน่าจะเป็น<sup>1</sup> ด้วย  $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$  อาจอาศัยทฤษฎีของเบส์ (หัวข้อ 2.2) เพื่อประมาณค่าในอนาคตด้วย  $p(\mathbf{x}_{T+1} | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) = p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T, \mathbf{x}_{T+1}) / p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ . ค่า  $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$  เอง ก็ประมาณได้ยากในทางปฏิบัติ และถึงแม้จะรู้ แต่ค่าของ  $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T, \mathbf{x}_{T+1})$  ยังยากกว่าที่จะประมาณ หากยังใช้แบบจำลองที่รองรับความสัมพันธ์เชิงลำดับระหว่างลำดับต่าง ๆ อย่างสมบูรณ์.

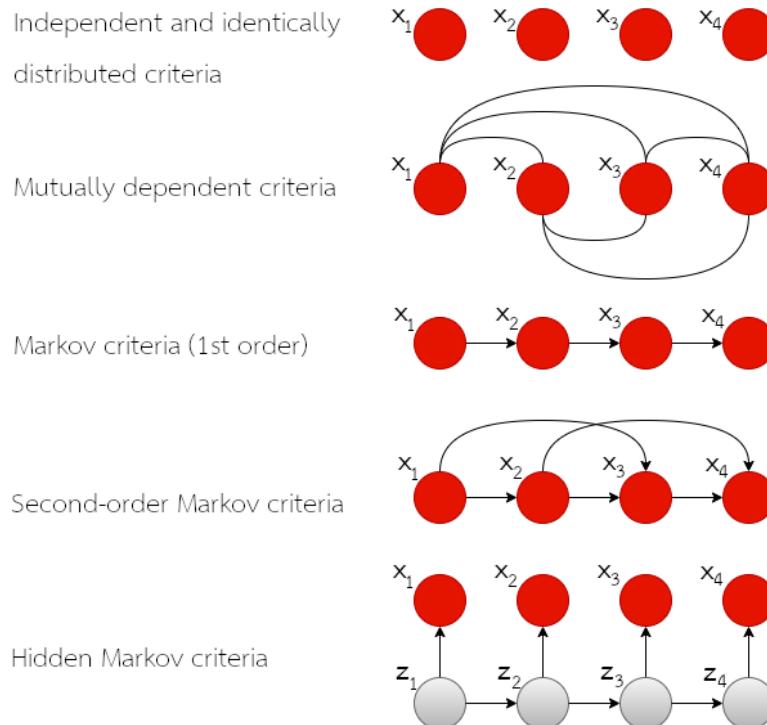
แนวทางการแก้ปัญหาเชิงคำนวนหลาย ๆ ครั้ง ที่การคำนวนค่าอย่างแม่นตรง (exact calculation) ไม่สามารถทำได้หรือทำได้ยาก การประมาณหรือการเพิ่มเงื่อนไขที่สมเหตุสมผล มักถูกนำมาใช้. วิธีหนึ่งคือเลือกการประมาณที่สุดขอบ เช่น การประมาณโดยไม่สนใจความสัมพันธ์เชิงลำดับเลย (ใช้สมมติฐาน ไอ.ไอ.ดี.) ซึ่งวิธีนี้ทำให้เราสามารถเลือกแบบจำลองต่าง ๆ ที่ไม่มีความสามารถเชิงลำดับ เช่น โครงข่ายประสาทเทียม (บทที่ 3) มาใช้ได้. แต่การประมาณที่สุดขอบเช่นนี้ เท่ากับเราทิ้งสารสนเทศความสัมพันธ์เชิงลำดับไปทั้งหมดเลย. เราไม่สามารถใช้ประโยชน์จากสารสนเทศเชิงลำดับได้เลย.

ยืดหยุ่นขึ้นมาบ้าง แบบจำลองมาร์คอฟ (Markov models หรืออาจเรียก Markov chain) ประมาณความสัมพันธ์เชิงลำดับโดยจำกัดเฉพาะลำดับที่ผ่านมาไม่เกินลำดับ. จุดสำคัญ คือ (1) แบบจำลองมาร์คอฟ จำกัดความสัมพันธ์เชิงลำดับ โดยจำกัดให้ค่าของจุดข้อมูลมีความสัมพันธ์เฉพาะกับค่าของจุดข้อมูลลำดับก่อนหน้า. จุดข้อมูลลำดับหลังไม่จำเป็น. นั่นคือ  $p(\mathbf{x}_t | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \dots, \mathbf{x}_T) = p(\mathbf{x}_t | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1})$ . (2) แบบจำลองมาร์คอฟ จำกัดความสัมพันธ์เชิงลำดับกลับไปเพียงจำนวนลำดับที่กำหนดเท่านั้น. ไม่ได้ย้อนกลับไปจนถึงจุดข้อมูลที่ลำดับแรกสุดทุกครั้ง. นั่นคือ แบบจำลองมาร์คอฟประมาณ  $p(\mathbf{x}_{T+1} | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) = p(\mathbf{x}_{T+1} | \mathbf{x}_{T+1-\tau}, \dots, \mathbf{x}_T)$  เมื่อ  $\tau$  เป็นจำนวนจุดข้อมูลในลำดับก่อนหน้าที่แบบจำลองมาร์คอฟถือว่ามีความสัมพันธ์. อภิมานพารามิเตอร์  $\tau$  ที่นิยมเลือกใช้คือ  $\tau = 1$  ซึ่งแบบจำลองมาร์คอฟที่ใช้  $\tau = 1$  จะเรียกว่า แบบจำลองมาร์คอฟอันดับหนึ่ง (first-order Markov model). นั่นคือ

$$p(\mathbf{x}_{T+1} | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) = p(\mathbf{x}_{T+1} | \mathbf{x}_T) \quad (8.1)$$

เมื่อลำดับ  $\{\mathbf{x}_t\}$  มีความสัมพันธ์เชิงลำดับตามเงื่อนไขของมาร์คอฟ. รูป 8.2 แสดงเงื่อนไขมาร์คอฟ เปรียบเทียบกับสมมติฐานอื่น ๆ. เนื่องจากแบบจำลองมาร์คอฟอันดับหนึ่งเป็นที่นิยมเป็นอย่างมาก ทำให้หลาย ๆ ครั้งการอ้างถึงแบบจำลองมาร์คอฟ หมายถึงแบบจำลองมาร์คอฟอันดับหนึ่ง. เพื่อความกระชับ เนื้อหาต่อไปนี้ ก็จะอ้างถึงแบบจำลองมาร์คอฟอันดับหนึ่งว่า แบบจำลองมาร์คอฟ ยกเว้นแต่จะระบุเป็นอื่น.

<sup>1</sup> หรือความหนาแน่นความน่าจะเป็น กรณีตัวแปรลุ่มต่อเนื่อง



รูปที่ 8.2: ตัวอย่างสมมติฐานแบบต่าง ๆ ของความสัมพันธ์ระหว่างจุดข้อมูล. แต่ละภาพ แสดงจุดข้อมูลสี่จุด. วงกลมเทibus สีแดง แทนจุดข้อมูล และตัวแปร  $x_i$  สำหรับ  $i = 1, \dots, 4$  แทนค่าของจุดข้อมูล. ภาพบนสุด แสดงสมมติฐานไอ.ไอ.ดี. ที่ถือว่า ค่าของจุดข้อมูลไม่มีความสัมพันธ์ระหว่างกันเลย (แต่ค่าของจุดข้อมูลทุกจุดมาจาก การแยกแจงเดียว กัน). ภาพที่สามจากบน แสดงสมมติฐาน ว่า ทุกจุดข้อมูลมีความสัมพันธ์ร่วมกัน (เกี่ยวพันกันหมด ไม่ว่ากับจุดข้อมูลลำดับก่อนหน้า หรือกับจุดข้อมูลลำดับหลัง). เส้นเชื่อม ไม่มีหัวลูกศร แสดงความสัมพันธ์สองทาง. ภาพที่สามจากบน แสดงสมมติฐานมาร์คอฟ(อันดับหนึ่ง). เส้นเชื่อมมีหัวลูกศร แสดงความสัมพันธ์ทางเดียว นั่นคือ ค่าของจุดข้อมูลต้นทางของเส้นเชื่อม ส่งผลต่อค่าของจุดข้อมูลปลายทาง (ที่หัวลูกศรชี้). ภาพที่สี่จากบน แสดงสมมติฐานมาร์คอฟยังตื้บสอง. ภาพล่างสุด แสดงสมมติฐานมาร์คอฟท่อนเร้น ที่ว่า ค่าของจุดข้อมูล  $x_i$  ขึ้นอยู่กับค่าของสถานะ  $z_i$  และค่าของสถานะ  $z_i$  ได้รับอิทธิพลมาจากการค่าของสถานะก่อนหน้า  $z_{i-1}$ . ค่าของสถานะ  $z_i$  จะสามารถสังเกตได้โดยตรง อาจมีความหมายซับเจน หรืออาจจะไม่สามารถสังเกตได้โดยตรง หรืออาจเป็นสถานะที่สมมติขึ้นมาก็ได.

ด้วยแบบจำลองมาร์คอฟ การแจกแจงร่วมของลำดับข้อมูลสามารถวิเคราะห์ได้จาก

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T) = p(\mathbf{x}_1) \cdot \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}). \quad (8.2)$$

การแจกแจงแบบมีเงื่อนไข  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$  สามารถถูกกำหนดให้มีค่าเท่ากันทุก ๆ ดัชนีลำดับ  $t$  ตามสมมติฐานข้อมูลเชิงลำดับแบบคงที่. การกำหนดเช่นนี้ ช่วยให้เราสามารถใช้ค่าพารามิเตอร์ร่วมกันได้<sup>2</sup> ซึ่งมีผลช่วยลดความซับซ้อนของการคำนวณ และช่วยลดปริมาณข้อมูลที่ต้องการสำหรับการฝึกแบบจำลองด้วย. แบบจำลองมาร์คอฟ ช่วยให้การประมาณการแจกแจงร่วมของข้อมูลลำดับคำนวณได้สะดวกขึ้น.

<sup>2</sup> การใช้ค่าพารามิเตอร์ร่วมกัน (parameter sharing) เป็นปัจจัยที่สำคัญที่ช่วยให้แบบจำลองหลาย ๆ ชนิดสามารถทำงานได้ดี. ตัวอย่าง เช่น โครงข่ายคอนโวโลยูชั่น (บทที่ 6) มีการใช้ชั้นคำนวณคอนโวโลยูชั่น ซึ่งอาศัยการเชื่อมต่อห้องถิน ที่ใช้ค่าพารามิเตอร์ร่วมกัน. เนื่องจากสมมติฐานของการข้ารูปแบบเชิงพื้นที่สมเหตุสมผลกับข้อมูลรูปภาพ โครงข่ายคอนโวโลยูชั่น จึงสามารถทำงานได้ดีกับภาระกิจกรรมพิวเตอร์ที่ต่ำ ๆ.

แต่อย่างไรก็ตาม เงื่อนไขการขึ้นอยู่กับค่าจุดข้อมูลก่อนหนึ่งเพียงหนึ่งลำดับ ก็เป็นปัจจัยจำกัดความสามารถของแบบจำลองมาร์คอฟลงด้วย. เราอาจจะเพิ่มอันดับ (เพิ่มค่า  $T$ ) ของแบบจำลองมาร์คอฟขึ้น ซึ่งก็อาจจะช่วยบรรเทาข้อจำกัดลงได้บ้าง ในเรื่องความสามารถที่จะเชื่อมโยงความสัมพันธ์เชิงลำดับระยะที่ยาวขึ้น. แต่แนวทางนี้ กลับส่งผลกระทบต่อประสิทธิภาพการคำนวณเป็นอย่างมาก (ศึกษารายละเอียดเพิ่มเติมจาก [15, §13.1]).

เพื่อแบบจำลองจะไม่ถูกจำกัดจำนวนลำดับย้อนหลังที่สัมพันธ์กันจากเงื่อนไขของมาร์คอฟ และกีรังสามารถคำนวณได้อย่างมีประสิทธิภาพ เราสามารถตัดแปลงแบบจำลองได้โดยกำหนดให้ จุดข้อมูล  $\mathbf{x}_t$  สัมพันธ์กับสถานะ  $\mathbf{z}_t$  และสถานะ  $\mathbf{z}_t$  เป็นไปตามเงื่อนไขของมาร์คอฟ. เงื่อนไขเช่นนี้ ช่วยให้แบบจำลองมีความยืดหยุ่น สามารถรองรับความสัมพันธ์ของจุดข้อมูลย้อนหลังกี่ลำดับก็ได้ โดยผ่านกลไกของตัวแปรสถานะ และยังคงรักษาการคำนวณที่มีประสิทธิภาพไว้ได้.

ค่าของสถานะ  $\mathbf{z}_t$  อาจมีความหมายซับเจกต์หรือไม่ก็ได้ และอาจสามารถสังเกตได้โดยตรงหรือไม่ก็ได้. ดังนั้น สถานะ  $\mathbf{z}_t$  จึงถูกเรียกว่า **สถานะซ่อนเร้น** (latent state) หรือ **ตัวแปรซ่อนเร้น** (latent variable) และเงื่อนไขนี้ เรียกว่า **เงื่อนไขมาร์คอฟซ่อนเร้น** (hidden Markov criteria). บางครั้ง เพื่อลดความสับสน ตัวแปรจุดข้อมูล  $\mathbf{x}_t$  อาจถูกเรียกว่า **ตัวแปรที่ถูกสังเกต** (observed variable). รูป 8.2 แสดงเงื่อนไขมาร์คอฟซ่อนเร้น (ภาพสุดท้าย) เปรียบเทียบกับสมมติฐานอื่น ๆ.

ค่าของสถานะซ่อนเร้น  $\mathbf{z}_t$  อาจจะเป็นค่าชนิดเดียวกับค่าของจุดข้อมูล  $\mathbf{x}_t$  ก็ได้ หรืออาจจะต่างกันก็ได้ เช่น ค่าของสถานะอาจเป็นค่าวิยุตที่มีจำนวนจำกัด แต่ค่าของจุดข้อมูล  $\mathbf{x}_t$  จะเป็นค่าวิยุตที่มีจำนวนจำกัด หรือค่าวิยุตที่มีจำนวนไม่จำกัด หรือค่าต่อเนื่องก็ได้. จำนวนมิติของสถานะซ่อนเร้น อาจจะเท่ากับ หรืออาจจะต่างจากจำนวนมิติของตัวแปรจุดข้อมูลก็ได้. เงื่อนไขมาร์คอฟซ่อนเร้น เพียงระบุว่า ความเป็นอิสระต่อกันแบบมีเงื่อนไข นั่นคือ  $\mathbf{z}_t \perp\!\!\!\perp \mathbf{z}_{t-2} | \mathbf{z}_{t-1}$  และ  $\mathbf{x}_t \perp\!\!\!\perp \mathbf{x}_{t-1} | \mathbf{z}_t$ .

ด้วยเงื่อนไขมาร์คอฟซ่อนเร้น การแยกแจงร่วมของลำดับชุดข้อมูลและชุดสถานะ สามารถเขียนได้

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{z}_1, \dots, \mathbf{z}_T) = p(\mathbf{z}_1) \cdot \left( \prod_{t=2}^T p(\mathbf{z}_t | \mathbf{z}_{t-1}) \right) \cdot \prod_{t=1}^T p(\mathbf{x}_t | \mathbf{z}_t). \quad (8.3)$$

แบบจำลองตามเงื่อนไขมาร์คอฟซ่อนเร้น เรียกว่า **แบบจำลองปริภูมิสถานะ** (state space model). แบบจำลองปริภูมิสถานะ มีแบบจำลองที่เฉพาะเจาะจงลงไปอีก ที่สำคัญได้แก่ แบบจำลองมาร์คอฟซ่อนเร้น ที่เจาะจงสำหรับกรณีสถานะซ่อนเร้นเป็นตัวแปรวิยุต และแบบจำลองพลวัตเชิงเส้น (linear dynamic model หรือ linear dynamic system) ที่เจาะจงสำหรับกรณีที่ทั้งสถานะซ่อนเร้นและตัวแปรที่ถูกสังเกตเป็นตัวแปร

ต่อเนื่องที่มีการแจกแจงเกาส์เซียน.

### 8.3 แบบจำลองมาร์คอฟซ่อนเร้น

แบบจำลองมาร์คอฟซ่อนเร้น<sup>3</sup>(hidden Markov model) เป็นแบบจำลองสำหรับข้อมูลเชิงลำดับ ที่ใช้เงื่อนไขมาร์คอฟซ่อนเร้น สำหรับกรณีที่สถานะซ่อนเร้นเป็นตัวแปรวิญญาต ที่ค่าวิญญาตมีจำนวนจำกัด. เมื่อพิจารณาสมการ 8.3 ด้วยสมมติฐานสถานะซ่อนเร้นวิญญาต เราจะเห็นว่า สำหรับ ระบบที่มีค่าของสถานะซ่อนเร้นได้  $K$  สถานะแล้ว ความน่าจะเป็น  $p(z_1)$  ที่อาจเรียกว่า **ค่าความน่าจะเป็นเริ่มต้น** (initial probabilities) สามารถแทนด้วยตารางความน่าจะเป็น โดยแต่ละรายการของตารางระบุค่าความน่าจะเป็น  $p(z_1 = k) = \pi_k$  สำหรับ  $k = 1, \dots, K$  เมื่อ  $\pi_k$  เป็นความน่าจะเป็นที่ระบบจะเริ่มต้นด้วยสถานะ  $k$ . ค่าของ  $\pi_k$  ต่าง ๆ เป็นพารามิเตอร์ของแบบจำลอง (กระบวนการหาค่าพารามิเตอร์เหล่านี้ จะอภิปรายในหัวข้อ 8.3.)

ความน่าจะเป็น  $p(z_t | z_{t-1})$  ที่เรียกว่า **ความน่าจะเป็นของการเปลี่ยนสถานะ** (transition probabilities) สามารถแทนด้วยเมตริกซ์  $\mathbf{A}$  ที่ส่วนประกอบ  $A_{ij}$  แทนความน่าจะเป็นของการเปลี่ยนสถานะจากสถานะ  $i$  ไปเป็นสถานะ  $j$ . เมทริกซ์  $\mathbf{A}$  อาจถูกเรียกว่า **เมทริกซ์การเปลี่ยนสถานะ** (transition matrix).

เพื่อความสะดวก สถานะซ่อนเร้นได้  $K$  สถานะ สามารถแสดงด้วยรหัสหนึ่งร้อน (one-hot coding ดูหัวข้อ 3.3) ซึ่งคือ สถานะซ่อนเร้น  $z_t = [z_{t,1}, \dots, z_{t,K}]^T \in \{0,1\}^K$  และ  $\sum_{k=1}^K z_{t,k} = 1$ . ดังนั้น  $\pi_k \equiv p(z_{1,k} = 1)$  และ  $A_{ij} \equiv p(z_{t,j} = 1 | z_{t-1,i} = 1)$ . หมายเหตุ ด้วยสมมติฐานข้อมูลเชิงลำดับแบบคงที่ ความน่าจะเป็นของการเปลี่ยนสถานะ  $p(z_t | z_{t-1})$  จะถูกแทนด้วยเมทริกซ์การเปลี่ยนสถานะ  $\mathbf{A}$  เหมือนกันสำหรับทุก ๆ ค่าของลำดับ  $t$ . นอกจ้านั้น ด้วยคุณสมบัติความน่าจะเป็น ทำให้รู้ว่า  $0 \leq \pi_k, A_{ij} \leq 1$  สำหรับทุก ๆ ค่าของ  $i, j, k$  กับ  $\sum_{k=1}^K \pi_k = 1$  และ  $\sum_{j=1}^K A_{ij} = 1$ .

เข่นเดียวกับพารามิเตอร์  $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]^T$  เมทริกซ์  $\mathbf{A}$  ก็เป็นพารามิเตอร์ของแบบจำลอง. เราสามารถเขียนฟังก์ชันการแจกแจงเริ่มต้น และฟังก์ชันการแจกแจงการเปลี่ยนสถานะ โดยเน้นพารามิเตอร์เหล่า

---

<sup>3</sup>เนื้อหาในส่วนของแบบจำลองมาร์คอฟซ่อนเร้น ได้รับอิทธิพลหลัก ๆ จาก [15].

นี้ได้<sup>4</sup>

$$p(\mathbf{z}_1 | \boldsymbol{\pi}) = \prod_{k=1}^K \boldsymbol{\pi}_k^{z_{1k}}. \quad (8.4)$$

$$p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{A}) = \prod_{j=1}^K \prod_{i=1}^K A_{ij}^{z_{t-1,i} z_{t,j}}. \quad (8.5)$$

ตัวอย่างเช่น สำหรับระบบที่มีสถานะซ่อนเร้น การคำนวณฟังก์ชันการแจกแจงเริ่มต้น สำหรับสถานะที่หนึ่ง ( $\mathbf{z}_1 = [1, 0, 0]^T$ ) ทำโดย  $p(\mathbf{z}_1 = [1, 0, 0]^T | \boldsymbol{\pi}) = \boldsymbol{\pi}_1^1 \cdot \boldsymbol{\pi}_2^0 \cdot \boldsymbol{\pi}_3^0 = \boldsymbol{\pi}_1$ .

สุดท้ายฟังก์ชันการแจกแจงแบบมีเงื่อนไข สำหรับตัวแปรที่ถูกสังเกต  $p(\mathbf{x}_t | \mathbf{z}_t)$  มักถูกเรียกว่า ความน่าจะเป็นของการปล่อย (emission probabilities). เช่นเดียวกัน ด้วยสมมติฐานข้อมูลเชิงลำดับแบบคงที่ ความน่าจะเป็นของการปล่อย ไม่ขึ้นกับตัวแปร  $t$ . เราอาจเขียนความน่าจะเป็นของการปล่อยด้วย  $p(\mathbf{x} | \mathbf{z})$  โดยละเอียดดังนี้

ความน่าจะเป็นของการปล่อย อาจถูกนิยามด้วยการแจกแจงที่หมายความกับลักษณะของตัวแปรที่ถูกสังเกต เช่น หากตัวแปรที่ถูกสังเกตเป็นค่าวิภาค เราอาจประมาณความน่าจะเป็นของการปล่อย ด้วยเมทริกซ์แจกแจง  $\Phi$  ที่ส่วนประกอบ  $\phi_{kd}$  ระบุค่าความน่าจะเป็นที่ตัวแปรที่ถูกสังเกตจะเป็นชนิดที่  $d^{th}$  เมื่อสถานะซ่อนเร้น เป็นสถานะที่  $k^{th}$ . นั่นคือ  $p(\mathbf{x} | \mathbf{z}, \Phi) = \prod_d \prod_k \phi_{kd}^{z_k x_d}$  เมื่อ  $\phi_{kd} \equiv p(x_d = 1 | z_k = 1)$  และทั้ง  $\mathbf{x}$  และ  $\mathbf{z}$  แสดงด้วยรหัสหนึ่งร้อน. หากตัวแปรที่ถูกสังเกตเป็นค่าต่อเนื่อง เราอาจเลือกการแจกแจงเกาส์เซียน สำหรับ ประมาณความน่าจะเป็นของการปล่อย. นั่นคือ  $p(\mathbf{x} | \mathbf{z}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,\dots,K}) = \prod_{k=1}^K (\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))^{z_k}$  เมื่อ  $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right)$  โดย  $D$  เป็นจำนวนมิติของ เวกเตอร์  $\mathbf{x}$  และ  $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,\dots,K}$  เป็นพารามิเตอร์ของแบบจำลอง. หมายเหตุ ค่า  $\pi$  ในที่นี้หมายถึง ค่าคงที่<sup>4</sup> ( $\pi \approx 3.1416$ ) ซึ่งต่างจาก  $\boldsymbol{\pi}_k$  (เช่นในสมการ 8.4) ที่เป็นพารามิเตอร์ของแบบจำลอง. สังเกตว่า ถึงแม้แบบจำลองมาร์คอฟช่องเรื้อรัง จะกำหนดให้สถานะซ่อนเร้น  $\mathbf{z}$  เป็นตัวแปรค่าวิภาค แต่ตัวแปรที่ถูกสังเกต  $\mathbf{x}$  อาจจะเป็นค่าวิภาค หรือเป็นค่าต่อเนื่องก็ได้.

เพื่อความสะดวก ต่อจากนี้ พารามิเตอร์ของฟังก์ชันประมาณความน่าจะเป็นของการปล่อย จะถูกอ้างถึง โดยรวมด้วยสัญกรณ์  $\boldsymbol{\phi}$  ไม่ว่าจะเลือกใช้การแจกแจงแบบใด และ  $\boldsymbol{\phi}_k$  จะหมายถึงชุดพารามิเตอร์ของการแจกแจง ของสถานะซ่อนเร้น  $k^{th}$  เช่น กรณีเมทริกซ์แจกแจง  $\boldsymbol{\phi}_k \equiv \prod_d \phi_{kd}^{x_d}$  หรือกรณีเกาส์เซียน  $\boldsymbol{\phi}_k \equiv \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ . นั่นคือ ฟังก์ชันประมาณความน่าจะเป็นของการปล่อย หรือเรียกย่อ ๆ ว่า ฟังก์ชันการปล่อย

<sup>4</sup>วิธีการเขียนเช่นนี้ เป็นการเขียนจากมุมมองคณิตศาสตร์เพื่อให้คำนิยามต่าง ๆ สมบูรณ์. การนำพจน์ต่าง ๆ เหล่านี้ไปเขียนโปรแกรม อาจดำเนินการต่างไป เพื่อให้คอมพิวเตอร์สามารถคำนวณได้อย่างมีประสิทธิภาพ.

(emission function) จะใช้สัญกรณ์  $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\phi})$  โดย  $\boldsymbol{\phi}$  หมายถึงพารามิเตอร์ของแบบจำลองที่ใช้ (อาจจะหมายถึง เมทริกซ์  $\boldsymbol{\Phi}$  หรือเซต  $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,\dots,K}$  หรือชุดพารามิเตอร์อื่น ๆ ตามการแจกแจงที่เลือก). ฟังก์ชันการปล่อย อาจเขียนได้โดยทั่วไปเป็น

$$p(\mathbf{x}_t|\mathbf{z}_t, \boldsymbol{\phi}) = \prod_{k=1}^K (p(\mathbf{x}_t|\boldsymbol{\phi}_k))^{\mathbf{z}_{t,k}}. \quad (8.6)$$

การแจกแจงร่วมของข้อมูลหนึ่งลำดับชุด สามารถเขียนได้เป็น

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = p(\mathbf{z}_1|\boldsymbol{\pi}) \cdot \left( \prod_{t=2}^T p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \right) \cdot \prod_{\tau=1}^T p(\mathbf{x}_\tau|\mathbf{z}_\tau, \boldsymbol{\phi}) \quad (8.7)$$

เมื่อ  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ ,  $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$  และ  $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$  เป็นชุดพารามิเตอร์ทั้งหมดของแบบจำลอง.

จากแบบจำลองในสมการ 8.7 เป็นแบบจำลองสร้างกำเนิด (generative model) ซึ่งด้วยทฤษฎีความน่าจะเป็น โดยเฉพาะทฤษฎีของเบส์ (หัวข้อ 2.2) เราสามารถทำการอนุमานต่าง ๆ ได้หากรู้ค่าของชุดพารามิเตอร์  $\boldsymbol{\theta}$  เช่น กรณีการทำนายข้อมูลการเงิน การทายค่าลำดับต่อไป  $\hat{\mathbf{x}}_{T+1}$  ที่สามารถอนุமานจาก<sup>5</sup>  $\hat{\mathbf{x}}_{T+1} \approx E[\mathbf{x}_{T+1}|\mathbf{X}, \boldsymbol{\theta}] = \sum_d \sum_k \sum_j \mathbf{x}_{T+1,d} \cdot p(\mathbf{x}_{T+1,d}|\mathbf{z}_{T+1,k}, \boldsymbol{\phi}) \cdot p(\mathbf{z}_{T+1,k}|\mathbf{z}_{T,j}, \mathbf{A}) \cdot p((\mathbf{z}_{T,j}|\mathbf{X}, \boldsymbol{\theta})$  และ  $p((\mathbf{z}_T|\mathbf{X}, \boldsymbol{\theta}) = \sum_{\{\mathbf{z}_1, \dots, \mathbf{z}_{T-1}\}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$  โดย  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}$ .

## การฝึกแบบจำลองมาร์คอฟซ่อนเร้น

การฝึกแบบจำลองมาร์คอฟซ่อนเร้น หรือการหาค่าชุดพารามิเตอร์  $\boldsymbol{\theta}$  สามารถทำได้ด้วยวิธีค่าฟังก์ชันควรจะเป็นสูงสุด (maximum likelihood ดูแบบฝึกหัด 5.18). แนวคิดของวิธีค่าฟังก์ชันควรจะเป็นสูงสุด คือ การหาค่าของชุดพารามิเตอร์ ที่ทำให้ค่าความน่าจะเป็นภายหลัง (posterior) มีค่ามากกว่าที่สุด โดยความน่าจะเป็นภายหลัง คือค่าความน่าจะเป็นที่คำนวนด้วยค่าต่าง ๆ ของชุดข้อมูลที่มีอยู่.

นั่นคือ ด้วยข้อมูลชุดลำดับ  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$  ที่สังเกตได้ ค่าค่าของชุดพารามิเตอร์  $\boldsymbol{\theta}$  ของแบบจำลองมาร์คอฟซ่อนเร้น สามารถหาได้จาก  $\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta})$  โดย  $p(\mathbf{X}|\boldsymbol{\theta})$  อาจหาได้โดยการ слายปัจจัย (marginalization)

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \quad (8.8)$$

<sup>5</sup> ตัวอย่างนี้ ต้องการแสดงในเห็นคร่าว ๆ เท่านั้นว่า การแจกแจงสามารถนำไปใช้ในการอนุमานต่าง ๆ ได้อย่างไร อย่างน้อยในทางทฤษฎี การอนุमานในทางปฏิบัติ อาจต้องการขั้นตอนวิธีคำนวณที่มีประสิทธิภาพมากกว่าการประยุกต์ใช้ทฤษฎีความน่าจะเป็นตรง ๆ ดังที่แสดงในตัวอย่างนี้.

ในทางปฏิบัติ การคำนวณสมการ 8.8 โดยตรง ทำได้ยาก เพราะว่า การถลวยปัจจัย อาศัยการบวกทุกค่าที่เป็นไปได้ของชุดลำดับสถานะช่อนเร้น  $\mathbf{Z} = \{z_1, \dots, z_T\}$  และ ที่แต่ละลำดับ สถานะช่อนเร้นก็มีโอกาสเป็นไปได้หลายค่า. ในที่นี้ กำหนดให้ ค่าของสถานะช่อนเร้นที่เป็นไปได้มีจำนวนเป็น  $K$  ค่า. ด้วยความยาวของลำดับเป็น  $T$  ทำให้มีชุดค่าของลำดับสถานะช่อนเร้นที่เป็นไปได้เท่ากับ  $K^T$  ชุด. จำนวนพจน์ที่ต้องทำการบวก จะเพิ่มขึ้นแบบซึ่งกำลังตามความยาวของชุดลำดับ.

แนวทางหนึ่งสามารถนำมาใช้คำนวณหาค่าพารามิเตอร์ที่เหมาะสมได้ คือ **ขั้นตอนวิธีอีเม็ม** (expectation-maximization algorithm คำย่อ EM algorithm) ซึ่งเป็นขั้นตอนวิธีทั่ว ๆ ไปสำหรับการหาค่าพารามิเตอร์ของแบบจำลอง โดยมีเนื้องความน่าจะเป็นประกอบ. กล่าวโดยสรุว ๆ ขั้นตอนวิธีอีเม็ม อาศัยการทำงานเป็นสองขั้นตอนหลัก ๆ คือ ขั้นตอนการหาค่าคาดหมาย (expectation phase) และขั้นตอนการหาค่าตัวทำมากที่สุด (maximization phase). ขั้นตอนวิธีอีเม็ม เริ่มด้วยค่าเริ่มต้นของพารามิเตอร์  $\boldsymbol{\theta}_0$  และคำนวณความน่าจะเป็นภายหลัง  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$ . จากนั้น ใช้ค่าความน่าจะเป็นภายหลังที่ได้ เพื่อประเมินค่าคาดหมายของลอการิทึมของค่าฟังก์ชันควรจะเป็นของข้อมูล

$$\varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \quad (8.9)$$

เมื่อ  $\boldsymbol{\theta}_0$  เป็นค่าพารามิเตอร์ ณ ปัจจุบัน ส่วน  $\boldsymbol{\theta}$  เป็นตัวแปรของค่าพารามิเตอร์ (ที่ต้องการจะปรับปรุงใหม่). การประเมินฟังก์ชันควรจะเป็น  $\varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0)$  จะใช้ในกระบวนการหาค่าพารามิเตอร์ที่ดีที่สุด. หลังจากได้ค่าพารามิเตอร์ชุดใหม่แล้ว จึงวนซ้ำคำนวณในลักษณะเช่นนี้ต่อไป จนค่าต่าง ๆ ถูเข้า หรือจนกว่าจะเป็นไปตามเงื่อนไขการจบการคำนวณ.

**ขั้นตอนวิธีอีเม็ม.** กล่าวโดยละเอียดแล้ว ด้วยค่าพารามิเตอร์  $\boldsymbol{\theta}_0$  ขั้นตอนวิธีอีเม็ม เริ่มที่ขั้นตอนการหาค่าคาดหมายใช้ค่าพารามิเตอร์นี้ในการคำนวณค่าคาดหมายของสถานะช่อนเร้น.

เพื่อความสะดวก นิยามความน่าจะเป็นภายหลังของสถานะช่อนเร้นด้วย  $\mathbf{q}_t$  และนิยามความน่าจะเป็นภายหลังร่วมด้วย  $\mathbf{R}^{(t-1,t)}$ . นั่นคือ

$$\mathbf{q}_t \equiv p(z_t|\mathbf{X}, \boldsymbol{\theta}_0) \quad (8.10)$$

$$\mathbf{R}^{(t-1,t)} \equiv p(z_{t-1}, z_t|\mathbf{X}, \boldsymbol{\theta}_0) \quad (8.11)$$

โดย สำหรับแต่ละตัวชีนลำดับ  $t$  เวกเตอร์  $\mathbf{q}_t \in [0, 1]^K$  ซึ่งส่วนประกอบ  $q_{tk} \equiv p(z_{tk} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$  และ เมทริกซ์  $\mathbf{R}^{(t-1,t)} \in [0, 1]^{K \times K}$  ซึ่งส่วนประกอบ  $R_{j,k}^{(t-1,t)} \equiv p(z_{t-1,j} = 1, z_{t,k} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$ .

หมายเหตุ เนื่องจาก  $z_{tk}$  เป็นตัวแปรค่าทวิภาค นั่นทำให้ ค่าคาดหมายของสถานะซ่อนเร้น  $E[z_{tk}] = p(z_{tk} = 1|\mathbf{X}, \boldsymbol{\theta}_0) \cdot 1 + p(z_{tk} = 0|\mathbf{X}, \boldsymbol{\theta}_0) \cdot 0 = p(z_{tk} = 1|\mathbf{X}, \boldsymbol{\theta}_0) = q_{tk}$ . ในทำนองเดียวกัน  $E[z_{t-1,j} \cdot z_{tk}] = p(z_{t-1,j} = 1, z_{t,k} = 1|\mathbf{X}, \boldsymbol{\theta}_0) \cdot 1 \cdot 1 + 0 + 0 + 0 = R_{j,k}^{(t-1,t)}$ . ดังนั้น การประมาณค่าความน่าจะเป็นภายหลัง  $\mathbf{q}_t$  และความน่าจะเป็นภายหลังร่วม  $\mathbf{R}^{(t-1,t)}$  จะเทียบเท่ากับการหาค่าคาดหมาย. การคำนวณในขั้นตอนนี้ จึงถูกเรียกว่าเป็น ขั้นตอนการหาค่าคาดหมาย.

ด้วยความน่าจะเป็นภายหลังทั้ง  $\mathbf{q}_t$  และ  $\mathbf{R}^{(t-1,t)}$  และนำแบบจำลองมาร์คอฟซ่อนเร้น ในสมการ 8.7 มาประกอบ เราจะได้ฟังก์ชันควรจะเป็น (สมการ 8.9) ว่า

$$\begin{aligned}\varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln \left\{ p(\mathbf{z}_1|\boldsymbol{\pi}) \cdot \left( \prod_{t=2}^T p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \right) \cdot \prod_{\tau=1}^T p(\mathbf{x}_\tau|\mathbf{z}_\tau, \boldsymbol{\phi}) \right\} \\ &= \sum_{k=1}^K q_{1k} \cdot \ln p(\mathbf{z}_1|\boldsymbol{\pi}) + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \\ &\quad + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(\mathbf{x}_\tau|\mathbf{z}_\tau, \boldsymbol{\phi}) \\ &= \sum_{k=1}^K q_{1k} \cdot \ln \pi_k + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln A_{jk} + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(\mathbf{x}_\tau|\boldsymbol{\phi}_k)\end{aligned}\tag{8.12}$$

โดย  $\boldsymbol{\pi}$  และ  $\mathbf{A}$  เป็นค่าความน่าจะเป็นเริ่มต้น และความน่าจะเป็นของการเปลี่ยนสถานะ ตามลำดับ. ส่วน  $p(\mathbf{x}_\tau|\boldsymbol{\phi}_k)$  คือความน่าจะเป็นของการปล่อยของสถานะที่  $k^{th}$ .

ขั้นตอนการหาค่าตัวทำมากที่สุด หากค่าของพารามิเตอร์  $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$  จากค่าที่ทำให้ฟังก์ชันควรจะเป็น (สมการ 8.12) มีค่ามากที่สุด (โดยค่า  $\mathbf{q}_t$  และ  $\mathbf{R}^{(t-1,t)}$  จะใช้ค่าที่ได้จากการหาค่าคาดหมาย และคิดเสื้อönเป็นค่าคงที่). ค่าพารามิเตอร์  $\boldsymbol{\pi}$  และ  $\mathbf{A}$  หากได้จากค่าทำมากที่สุดของฟังก์ชันควรจะเป็น ประกอบกับเงื่อนไขของพารามิเตอร์ ได้แก่  $\sum_k \pi_k = 1$  และ  $\sum_k A_{jk} = 1$  และได้ผลลัพธ์ (ดูแบบฝึกหัด 8.2) คือ

$$\pi_k = \frac{q_{1k}}{\sum_{j=1}^K q_{1j}}\tag{8.13}$$

$$A_{jk} = \frac{\sum_{t=2}^T R_{jk}^{(t-1,t)}}{\sum_{l=1}^K \sum_{t=2}^T R_{jl}^{(t-1,t)}}.\tag{8.14}$$

สำหรับพารามิเตอร์  $\boldsymbol{\phi}$  การหาค่าก็สามารถทำได้ในทำนองเดียวกัน เพียงแต่แบบจำลองมาร์คอฟซ่อนเร้น เปิดกว้างสำหรับการเลือกใช้ฟังก์ชันการปล่อย  $p(\mathbf{x}_t|\boldsymbol{\phi}_k)$  ได้หลายแบบ.

หากเลือกฟังก์ชันการปล่อยเกาล์เชียน นั่นคือ  $p(\mathbf{x}_t|\boldsymbol{\phi}_k) \equiv \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ . เมื่อทำการหาค่าตัวทำมากที่สุด ผลลัพธ์จะได้เป็น

$$\boldsymbol{\mu}_k = \frac{\sum_{t=1}^T q_{tk} \cdot \mathbf{x}_t}{\sum_{t=1}^T q_{tk}} \quad (8.15)$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{t=1}^T q_{tk} \cdot (\mathbf{x}_t - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_t - \boldsymbol{\mu}_k)^T}{\sum_{t=1}^T q_{tk}}. \quad (8.16)$$

หากเลือกฟังก์ชันการปล่อยเนกานามวิยุต (discrete multinomial emission function) นั่นคือ การกำหนด  $p(\mathbf{x}_t|\boldsymbol{\phi}_k) \equiv \prod_{d=1}^D \phi_{kd}^{x_{td}}$  เมื่อ พารามิเตอร์  $\phi_{kd}$  แทนค่าความน่าจะเป็นที่จะพบตัวแปรที่ถูกสังเกต  $\mathbf{x}_t$  เป็นชนิด  $d^{th}$  หากสถานะซ่อนเร้นเป็นชนิด  $k^{th}$ . ค่าของตัวแปรที่ถูกสังเกต  $\mathbf{x}_t$  ใช้รหัสหนึ่งร้อน ซึ่งคือ  $\mathbf{x}_t = [x_{t1}, \dots, x_{tD}]^T$  โดย  $D$  คือจำนวนค่าวิยุต ที่ตัวแปรสามารถแทนได้. ส่วนประกอบ  $x_{td} \in \{0, 1\}$  และ  $\sum_{d=1}^D x_{td} = 1$ . เมื่อทำการหาค่าตัวทำมากที่สุด ผลลัพธ์จะได้เป็น

$$\phi_{kd} = \frac{\sum_{t=1}^T q_{tk} \cdot x_{td}}{\sum_{t=1}^T q_{tk}}. \quad (8.17)$$

ค่าเริ่มต้นของพารามิเตอร์  $\boldsymbol{\pi}$  และ  $\mathbf{A}$  อาจกำหนดได้โดยการสุ่ม แต่ต้องควบคุมให้ค่าเป็นไปตามเงื่อนไขของความน่าจะเป็น นั่นคือ  $\pi_k \geq 0$ ,  $\sum_k \pi_k = 1$ ,  $A_{jk} \geq 0$  และ  $\sum_k A_{jk} = 1$ . ค่าเริ่มต้นของพารามิเตอร์ของ  $\boldsymbol{\phi}$  อาจกำหนดเป็นสมมือนพารามิเตอร์อีกส่วน และใช้กระบวนการเรียนรู้จากข้อมูลเพื่อช่วยกำหนดค่าได้.

ขั้นตอนวิธีอีเม็ม มีคุณสมบัติความถูกต้องและคุณสมบัติการถูกลู่เข้าที่ดี (ดู [188] เพิ่มเติม). อย่างไรก็ตาม ในขั้นตอนการหาค่าคาดหมาย การประมาณค่าของค่าความน่าจะเป็นภายหลัง  $\mathbf{q}_t$  และ  $\mathbf{R}^{(t-1,t)}$  แม้อาจสามารถทำได้โดยวิธีการลากปัจจัย เช่น  $\mathbf{q}_t = p(\mathbf{z}_t|\mathbf{X}, \boldsymbol{\theta}_0) = \sum_{\{\mathbf{z}_1, \dots, \mathbf{z}_{t-1}, \mathbf{z}_{t+1}, \dots, \mathbf{z}_T\}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$  และ  $\mathbf{R}^{(t-1,t)} = p(\mathbf{z}_{t-1}, \mathbf{z}_t|\mathbf{X}, \boldsymbol{\theta}_0) = \sum_{\{\mathbf{z}_1, \dots, \mathbf{z}_{t-2}, \mathbf{z}_{t+1}, \dots, \mathbf{z}_T\}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$  โดย  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$  ก็อาจจะหาได้จากกฎของเบส  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) = p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}_0) / \sum_{\mathbf{Z}} p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}_0)$  แต่วิธีนี้ใช้การคำนวณมหาศาล.

แนวทางการคำนวณ  $\sum_{\{\mathbf{z}_1, \dots, \mathbf{z}_{t-1}, \mathbf{z}_{t+1}, \dots, \mathbf{z}_T\}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$  เองอย่างเดียว ก็เท่ากับต้องบวกจนทั้งหมด  $(T-1) \cdot K$  พจน์เข้าด้วยกัน และค่าของแต่ละพจน์ ก็ต้องผ่านการคำนวณต่าง ๆ ดังอภิราย. ดังนั้น เมื่อต้องนำมาใช้กับขั้นตอนวิธีอีเม็ม ซึ่งต้องคำนวณค่าความน่าจะเป็นภายหลังใหม่ทุก ๆ สมัยฝึก การประมาณค่า  $\mathbf{q}_t$  และ  $\mathbf{R}^{(t-1,t)}$  ด้วยวิธีนี้ จึงไม่เหมาะสมที่จะนำมาใช้ได้ในทางปฏิบัติ.

ปัญหาการคำนวณค่าความน่าจะเป็นภายหลังอย่างมีประสิทธิภาพ ถูกบรรเทาด้วยขั้นตอนวิธีไปข้างหน้า กับถอยกลับ ที่จะอภิรายต่อไปในหัวข้อ 8.3.

## ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ

ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ (forward-backward algorithm) เป็นขั้นตอนวิธี ที่ใช้คำนวณค่าของความน่าจะเป็นภายหลังที่ใช้ในวิธีอื่นๆ ได้อย่างมีประสิทธิภาพ. จริง ๆ แล้ว ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ มีการศึกษาอย่างกว้าง และวิธีดำเนินการก็มีอยู่หลายแบบ หัวข้อนี้จะอภิปรายแบบหนึ่งที่มีการใช้อย่างกว้างขวาง[15] เรียกว่า **ขั้นตอนวิธีแอลfa-บีตา** (alpha-beta algorithm). เนื่องจาก หัวข้อนี้พิจารณาค่าพารามิเตอร์  $\boldsymbol{\theta}_0$  เป็นสมือนค่าคงที่ ดังนั้นเงื่อนไข  $\boldsymbol{\theta}_0$  จะถูกละไว้ในฐานที่เข้าใจ เพื่อความกระชับ.

ด้วยเงื่อนไขของแบบจำลองมาร์คอฟซ่อนเร้น ระบบจะมีคุณสมบัติดังนี้

$$p(\mathbf{X}|\mathbf{z}_t) = p(\mathbf{x}_1, \dots, \mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t) \quad (8.18)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{z}_t) = p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_t) \quad (8.19)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}, \mathbf{z}_t) = p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}) \quad (8.20)$$

$$p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t, \mathbf{z}_{t+1}) = p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_{t+1}) \quad (8.21)$$

$$p(\mathbf{x}_{t+2}, \dots, \mathbf{x}_T | \mathbf{x}_{t+1}, \mathbf{z}_{t+1}) = p(\mathbf{x}_{t+2}, \dots, \mathbf{x}_T | \mathbf{z}_{t+1}) \quad (8.22)$$

$$p(\mathbf{X} | \mathbf{z}_{t-1}, \mathbf{z}_t) = p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}) \cdot p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t) \quad (8.23)$$

$$p(\mathbf{x}_{T+1} | \mathbf{X}, \mathbf{z}_{T+1}) = p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \quad (8.24)$$

$$p(\mathbf{z}_{T+1} | \mathbf{X}, \mathbf{z}_{T+1}) = p(\mathbf{z}_{T+1} | \mathbf{z}_{T+1}) \quad (8.25)$$

เมื่อ  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ .

พิจารณา

$$\mathbf{q}_t = p(\mathbf{z}_t | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{z}_t) \cdot p(\mathbf{z}_t)}{p(\mathbf{X})}. \quad (8.26)$$

ด้วยสมการ 8.18 และกฎผลคูณ เราจะได้

$$\mathbf{q}_t = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{z}_t) \cdot p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_t) \cdot \beta(\mathbf{z}_t)}{p(\mathbf{X})} \quad (8.27)$$

โดยนิยาม

$$\alpha(\mathbf{z}_t) \equiv p(\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{z}_t) \quad (8.28)$$

$$\beta(\mathbf{z}_t) \equiv p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t). \quad (8.29)$$

ค่า  $\alpha(\mathbf{z}_t) \in \mathbb{R}^K$  แทนค่าความน่าจะเป็นร่วม ระหว่างข้อมูลชุดลำดับที่สังเกตจนถึงเวลา  $t$  กับสถานะซ่อนเร้นของเวลา  $t$ . ค่า  $\beta(\mathbf{z}_t) \in \mathbb{R}^K$  แทนค่าความน่าจะเป็นแบบมีเงื่อนไขของข้อมูลชุดลำดับที่สังเกตตั้งแต่เวลา  $t$  ไปจนจบ เมื่อมีสถานะซ่อนเร้นที่เวลา  $t$  เป็นเงื่อนไข. เพื่อความสะดวก กำหนดให้  $\alpha(z_{tk}) \equiv p(\mathbf{x}_1, \dots, \mathbf{x}_t, z_{tk} = 1)$  และ  $\beta(\mathbf{z}_t) \equiv p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | z_{tk} = 1)$  โดย  $\mathbf{z}_t$  แสดงด้วยรหัสหนึ่งร้อย.

ในทำนองเดียวกัน

$$\begin{aligned} \mathbf{R}^{(t-1,t)} &= p(\mathbf{z}_{t-1}, \mathbf{z}_t | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{z}_{t-1}, \mathbf{z}_t) \cdot p(\mathbf{z}_{t-1}, \mathbf{z}_t)}{p(\mathbf{X})} \\ &= \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}) \cdot p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_{t-1})}{p(\mathbf{X})} \\ &= \frac{\alpha(\mathbf{z}_{t-1}) \cdot p(\mathbf{x}_t | \mathbf{z}_t) \cdot \beta(\mathbf{z}_t) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1})}{p(\mathbf{X})}. \end{aligned} \quad (8.30)$$

แนวคิดของขั้นตอนวิธีไปข้างหน้ากับถอยกลับ คือ จัดรูปการคำนวณ  $\alpha(\mathbf{z}_t)$  และ  $\beta(\mathbf{z}_t)$  ให้อยู่ในรูปที่สามารถคำนวณได้อย่างมีประสิทธิภาพ โดยอาศัยความล้มเหลวแบบเรียกซ้ำ (recursive relation). การคำนวณค่า  $\alpha(\mathbf{z}_t)$  สามารถจัดรูปใหม่ได้ดังนี้

$$\begin{aligned} \alpha(\mathbf{z}_t) &= p(\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{z}_t) \\ &= p(\mathbf{x}_1, \dots, \mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_t) \cdot p(\mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_{t-1}, \mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_t | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_{t-1}) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_{t-1}) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} \alpha(\mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \end{aligned} \quad (8.31)$$

สำหรับ  $t = 2, \dots, T$ .

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{z}_1) \cdot p(\mathbf{x}_1 | \mathbf{z}_1) = \prod_{k=1}^K (\pi_k \cdot p(\mathbf{x}_1 | \boldsymbol{\phi}_k))^{z_{1k}}. \quad (8.32)$$

นั่นคือ  $\alpha(z_{1k}) = \pi_k \cdot p(\mathbf{x}_1 | \boldsymbol{\phi}_k)$ .

การคำนวณเริ่มจากลำดับเวลาแรก แล้วค่อย ๆ คำนวณขึ้นไปทีละลำดับ. การคำนวณสมการ 8.31 ทำการบวก  $K$  พจน์ สำหรับแต่ละสถานะซ่อนเร้น ซึ่งสถานะซ่อนเร้นมีจำนวนทั้งหมด  $K$  สถานะ. ดังนั้นที่แต่ละลำดับเวลา การคำนวณจะขยายเป็น  $K^2$  (นั่นคือ  $O(K^2)$ ) และการคำนวณจะเป็น  $O(TK^2)$  สำหรับทั้งชุดลำดับ.

ในทำนองเดียวกัน ค่า  $\beta(\mathbf{z}_t)$  สามารถจัดรูปการคำนวณใหม่ได้ดังนี้

$$\begin{aligned}
 \beta(\mathbf{z}_t) &= p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t) \\
 &= \sum_{\mathbf{z}_{t+1}} p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T, \mathbf{z}_{t+1} | \mathbf{z}_t) \\
 &= \sum_{\mathbf{z}_{t+1}} p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_{t+1}, \mathbf{z}_t) \cdot p(\mathbf{z}_{t+1} | \mathbf{z}_t) \\
 &= \sum_{\mathbf{z}_{t+1}} p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_{t+1}) \cdot p(\mathbf{z}_{t+1} | \mathbf{z}_t) \\
 &= \sum_{\mathbf{z}_{t+1}} p(\mathbf{x}_{t+2}, \dots, \mathbf{x}_T | \mathbf{z}_{t+1}) \cdot p(\mathbf{x}_{t+1} | \mathbf{z}_{t+1}) \cdot p(\mathbf{z}_{t+1} | \mathbf{z}_t) \\
 &= \sum_{\mathbf{z}_{t+1}} \beta(\mathbf{z}_{t+1}) \cdot p(\mathbf{x}_{t+1} | \mathbf{z}_{t+1}) \cdot p(\mathbf{z}_{t+1} | \mathbf{z}_t)
 \end{aligned} \tag{8.33}$$

สำหรับ  $t = 1, \dots, T-1$ . สำหรับ ลำดับเวลาท้ายสุด เพื่อประเมินค่า  $\beta(\mathbf{z}_T)$  พิจารณาสมการ 8.26 และ 8.27 เมื่อ  $t = T$ . นั่นคือ

$$p(\mathbf{z}_T | \mathbf{X}) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{z}_T) \cdot \beta(\mathbf{z}_T)}{p(\mathbf{X})} = \frac{p(\mathbf{X}, \mathbf{z}_T)}{p(\mathbf{X})} \cdot \beta(\mathbf{z}_T) \tag{8.34}$$

และจากกฎผลคูณ ซึ่ง ณ ที่นี่ คือ  $\frac{p(\mathbf{X}, \mathbf{z}_T)}{p(\mathbf{X})} = p(\mathbf{z}_T | \mathbf{X})$  ดังนั้น ค่าของ  $\beta(\mathbf{z}_T)$  ต้องเท่ากับหนึ่ง สำหรับทุก ๆ สถานะของ  $\mathbf{z}_T$ . นั่นคือ  $\beta(z_{Tk}) = 1$  สำหรับ  $k = 1, \dots, K$ . (หมายเหตุ นิยาม  $\beta(\mathbf{z}_t) \equiv p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T | \mathbf{z}_t)$  ในนิพจน์ 8.29 ไม่ได้ครอบคลุมลำดับ  $t = T$ .)

ตรงกันข้ามกับการคำนวณ  $\alpha(\mathbf{z}_t)$  ที่เริ่มจากลำดับเวลาแรก แล้วขึ้นลำดับไปข้างหน้า การคำนวณ  $\beta(\mathbf{z}_t)$  เริ่มจากลำดับเวลาท้ายสุด แล้วขึ้นลำดับโดยหลังมาเรื่อย ๆ. หลังจากได้ค่าของ  $\alpha(\mathbf{z}_t)$  และ  $\beta(\mathbf{z}_t)$  สำหรับ  $t = 1, \dots, T$  แล้วอาจประเมินค่าความน่าจะเป็นภายหลังด้วยสมการ 8.27 และ 8.30 ในขั้นตอนการหาค่าคาดหมาย ซึ่งต้องการค่า  $p(\mathbf{X})$  ประกอบ หรือ อาจนำค่า  $\alpha(\mathbf{z}_t)$  และ  $\beta(\mathbf{z}_t)$  ที่ได้ไปใช้ในขั้นตอนการหาค่าตัวทำมากที่สุดโดยตรงเลยก็ได้. เนื่องจากการคำนวณในขั้นตอนการหาค่าตัวทำมากที่สุดนั้น ค่าของ

$p(\mathbf{X})$  จะหักล้างกันเอง ตัวอย่างเช่น สมการ 8.13 ซึ่งคือ

$$\pi_k = \frac{q_{1k}}{\sum_{j=1}^K q_{1j}} = \frac{\alpha(z_{1k}) \cdot \beta(z_{1k})}{\sum_{j=1}^K \alpha(z_{1j}) \cdot \beta(z_{1j})}. \quad (8.35)$$

อย่างไรก็ตาม หากต้องการประเมินค่าของ  $p(\mathbf{X})$  ก็สามารถทำได้อย่างสะดวก. พิจารณาสมการ 8.26 และสมการ 8.27 จะเห็นว่า

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{X}) &= \frac{\alpha(\mathbf{z}_t) \cdot \beta(\mathbf{z}_t)}{p(\mathbf{X})} \\ p(\mathbf{z}_t, \mathbf{X}) &= \alpha(\mathbf{z}_t) \cdot \beta(\mathbf{z}_t) \end{aligned}$$

ดังนั้น

$$p(\mathbf{X}) = \sum_{\mathbf{z}_t} p(\mathbf{z}_t, \mathbf{X}) = \sum_{\mathbf{z}_t} \alpha(\mathbf{z}_t) \cdot \beta(\mathbf{z}_t) \quad (8.36)$$

ซึ่งหมายถึง เราสามารถเลือกดัชนีลำดับ  $t$  ได้ก็ได้ ที่จะใช้ประเมินค่า  $p(\mathbf{X})$ . ค่าดัชนีลำดับที่สะดวกในกรณีนี้คือ  $t = T$  ซึ่งจะได้

$$p(\mathbf{X}) = \sum_{\mathbf{z}_T} \alpha(\mathbf{z}_T) \quad (8.37)$$

เพราะว่า  $\beta(\mathbf{z}_{tk}) = 1$  สำหรับทุก ๆ ค่าของ  $k$ .

สังเกตว่า ค่า  $p(\mathbf{X})$  อาจหาได้โดยการ сл้ายบจัจย์  $p(\mathbf{X}) = \sum_{\mathbf{Z}} p(\mathbf{Z}, \mathbf{X})$  แต่การทำเช่นนี้เท่ากับการบวกของ  $K^T$  พจน์ ซึ่งแต่ละพจน์ต้องประเมินค่า  $p(\mathbf{Z}, \mathbf{X})$  เปรียบเทียบกับสมการ 8.37 ซึ่งเท่ากับการบวกของ  $K$  พจน์เท่านั้น. การจัดรูปการคำนวนในสมการ 8.37 จึงเปลี่ยนการคำนวนที่บริโภคเป็นสัดส่วนเติบโตแบบชี้กำลัง มาเป็นสัดส่วนแบบเชิงเส้น ลดการคำนวนลงได้มหาศาล โดยเฉพาะกับชุดลำดับข้อมูลยาว ๆ.

การอนุมานข้อมูลด้วยแบบจำลองมาร์คอฟช่องเรื้อรัง. แบบจำลองมาร์คอฟช่องเรื้อรัง สามารถนำประยุกต์ใช้ได้กว้างขวางในการอนุมานต่าง ๆ. ตัวอย่างหนึ่งที่สำคัญ คือ กรณีการอนุมานจุดข้อมูลต่อไปในชุดลำดับ เช่น ในกรณีการทำนายทางการเงิน (ภาพบนสุด รูป 8.1 ที่อินพุตเป็นชุดลำดับของราคากิตต่อวัน จากหลาย ๆ วันที่ผ่านมา และเอาต์พุตคือการทำนายราคาปิดของวันปัจจุบัน). นั่นคือ ด้วยข้อมูลชุดลำดับที่ถูกสังเกตมา  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$  เราต้องการทำนายจุดข้อมูล  $\mathbf{x}_{T+1}$ . ด้วยเงื่อนไขมาร์คอฟและทฤษฎีของเบส การ

อนุมานอาจทำผ่านค่าความน่าจะเป็น ซึ่งวิเคราะห์ได้ดังนี้

$$\begin{aligned}
 p(\mathbf{x}_{T+1} | \mathbf{X}) &= \sum_{\mathbf{z}_{T+1}} p(\mathbf{x}_{T+1}, \mathbf{z}_{T+1} | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{T+1}} p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \cdot p(\mathbf{z}_{T+1} | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{T+1}} \left\{ p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \cdot \sum_{\mathbf{z}_T} p(\mathbf{z}_T, \mathbf{z}_{T+1} | \mathbf{X}) \right\} \\
 &= \sum_{\mathbf{z}_{T+1}} \left\{ p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \cdot \sum_{\mathbf{z}_T} p(\mathbf{z}_{T+1} | \mathbf{z}_T) \cdot p(\mathbf{z}_T | \mathbf{X}) \right\} \\
 &= \sum_{\mathbf{z}_{T+1}} \left\{ p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \cdot \sum_{\mathbf{z}_T} p(\mathbf{z}_{T+1} | \mathbf{z}_T) \cdot \frac{p(\mathbf{z}_T, \mathbf{X})}{p(\mathbf{X})} \right\} \\
 &= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{T+1}} \left\{ p(\mathbf{x}_{T+1} | \mathbf{z}_{T+1}) \cdot \sum_{\mathbf{z}_T} p(\mathbf{z}_{T+1} | \mathbf{z}_T) \cdot \alpha(\mathbf{z}_T) \right\}. \quad (8.38)
 \end{aligned}$$

สมการ 8.38 ประเมินได้ ด้วยการบวก  $K^2$  พจน์. การคำนวณสามารถทำได้อย่างมีประสิทธิภาพ เช่นนี้ได้ เพราะการประเมินค่า  $p(\mathbf{z}_t | \mathbf{X})$  สามารถทำผ่านค่า  $\alpha(\mathbf{z}_t)$  ได้.

**ข้อจำกัดของแบบจำลองมาร์คอฟช่องเร้น.** แบบจำลองมาร์คอฟช่องเร้น เป็นแบบจำลองเชิงความน่าจะเป็น แม้จะมีศักยภาพสูง แต่ปัจจุบัน การประยุกต์ใช้กับภาระกิจการรู้จำรูปแบบเชิงลำดับที่ซับซ้อน เช่น การรู้จำเสียงพูด และการประมวลผลภาษาธรรมชาติ นิยมใช้แนวทางของโครงข่ายประสาทเวียนกลับ (หัวข้อ 9.2) มากกว่า เพราะว่า โครงข่ายประสาทเวียนกลับสามารถทำการกิจดังกล่าวได้ดีกว่ามาก. ปัจจัยที่เป็นสาเหตุของความสามารถที่จำกัดของแบบจำลองมาร์คอฟช่องเร้น อาจได้แก่ จำนวนสถานะช่องเร้น ถูกจำกัด หรือ การที่ต้องกำหนด ระบุจำนวนสถานะอย่างชัดเจนในตัวแบบจำลอง, การเรียนรู้ความสัมพันธ์ระยะยาว ต้องการอาศัยการทำผ่านตัวแปรช่องเร้น ซึ่งถูกจำกัดจำนวนสถานะไว้ และการใช้รหัสหนึ่งร้อน ยังทำให้ตัวแปรช่องเร้นไม่อาจแทนความสัมพันธ์ที่ซับซ้อนอย่างมีประสิทธิภาพได้ คือ ไม่สามารถบุแยกความสัมพันธ์เป็นองค์ประกอบอยู่ได้ และอาจจะรวมถึง การอาศัยมุมมองเชิงความน่าจะเป็น ซึ่งอยู่บนสมมติฐานของการวิเคราะห์ที่ครอบคลุมทุกรณี ที่อาจจะส่งผลในทางปฏิบัติ (หากไม่มีกลไกพิเศษ เพื่อแก้ไข ชดเชย หรือบรรเทาการคำนวณที่พัฒนาจากสมมติฐาน เช่นนี้<sup>6</sup>).

<sup>6</sup> งานวิจัย [134, 136] ได้ชี้ให้เห็นปัญหาของสมมติฐานว่าการวิเคราะห์ครอบคลุมทุกรณี (assumption of all-inclusiveness) และได้เสนอกลไกบรรเทา สำหรับกรณีโครงข่ายประสาทที่ยึดจำแนวประเทศ.

## 8.4 อภิธานศัพท์

**ข้อมูลเชิงลำดับ (sequential data):** ข้อมูลประเภทที่จุดข้อมูลมีความสัมพันธ์เชิงลำดับระหว่างกัน

**แบบจำลองมาร์คอฟ (Markov model):** แบบจำลองความสัมพันธ์เชิงลำดับของข้อมูล ที่ใช้เงื่อนไขมาร์คอฟ ซึ่งจำกัด ให้จุดข้อมูลมีความความสัมพันธ์เชิงความไม่เป็นอิสระต่อกันแบบมีเงื่อนไข (conditional dependence) ระหว่างกันได้ เฉพาะกับจุดข้อมูลลำดับที่ผ่านมา ตามจำนวนลำดับที่กำหนด.

**แบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov model คำย่อ HMM):** แบบจำลองสำหรับข้อมูลเชิงลำดับ ที่บรรยายความสัมพันธ์เชิงลำดับของค่าข้อมูล ผ่านตัวแปรซ่อนเร้น และใช้เงื่อนไขมาร์คอฟกับตัวแปรซ่อนเร้น.

**ตัวแปรที่สังเกตได้ (observable variable):** ค่าจุดข้อมูล ซึ่งเป็นค่าที่สามารถรู้ว่าแนอนได้

**สถานะซ่อนเร้น (latent state) หรือ ตัวแปรซ่อนเร้น (latent variable):** ค่าจุดข้อมูลที่สมมติขึ้น หรือเชื่อว่ามีอยู่ อาจจะสามารถรู้ค่าแนอนได้ หรืออาจจะไม่สามารถรู้ค่าได้เลย และอาจมีนัยความหมายจริง ก็ได้ หรืออาจจะไม่ได้มีนัยความหมายที่ชัดเจนก็ได้.

**ค่าความน่าจะเป็นเริ่มต้น (initial probabilities):** ความน่าจะเป็นของสถานะต่าง ๆ ของจุดข้อมูลที่ลำดับแรกสุด

**ความน่าจะเป็นของการเปลี่ยนสถานะ (transition probabilities):** ความน่าจะเป็นของสถานะต่าง ๆ ของจุดข้อมูลที่ลำดับถัดไป เมื่อจุดข้อมูลลำดับปัจจุบันมีสถานะดังระบุ.

**ความน่าจะเป็นของการปล่อย (emission probabilities):** ความน่าจะเป็นของค่าจุดข้อมูลที่สังเกตได้ เมื่อสถานะซ่อนเร้นมีค่าดังระบุ

**ฟังก์ชันควรจะเป็น (likelihood function):** ฟังก์ชันของตัวแปรที่สนใจ ที่คำนวณค่าความน่าจะเป็น ด้วยค่าข้อมูลที่สังเกตได้

**ขั้นตอนวิธีอีเม็ม (expectation-maximization algorithm คำย่อ EM algorithm):** ขั้นตอนวิธีทั่ว ๆ ไปสำหรับการหาค่าพารามิเตอร์ ของแบบจำลองเชิงความน่าจะเป็น โดยอาศัยขั้นตอนการคำนวณค่าคาดหมาย เมื่อกำหนดค่าพารามิเตอร์ที่สนใจเป็นค่าคงที่ สลับกับขั้นตอนการหาค่ามากที่สุด ที่ใช้ค่าคาด

หมายที่ได้ประเมินค่าฟังก์ชันควรจะเป็น สำหรับการหาค่าพารามิเตอร์ที่ทำให้ฟังก์ชันควรจะเป็นมีค่ามากที่สุด.

## 8.5 แบบฝึกหัด

``Life is an opportunity, benefit from it. Life is beauty, admire it. Life is a dream, realize it. Life is a challenge, meet it. Life is a duty, complete it. Life is a game, play it. Life is a promise, fulfill it. Life is sorrow, overcome it. Life is a song, sing it. Life is a struggle, accept it. Life is a tragedy, confront it. Life is an adventure, dare it. Life is luck, make it. Life is too precious, do not destroy it. Life is life, fight for it.''

---Mother Teresa

“ชีวิต เป็นโอกาส ใช้ประโยชน์จากมัน. ชีวิต เป็นความสวยงาม ชีนั่นเป็นมัน. ชีวิต เป็นความฝัน ทำมันให้เป็นจริง. ชีวิต เป็นความท้าทาย ต้อนรับมัน. ชีวิต เป็นหน้าที่ ทำมันให้สมบูรณ์. ชีวิต เป็นเกมส์ เล่นมัน. ชีวิต เป็นคำสัญญา รักษา มัน. ชีวิต เป็นความศร้า ผ่านมันให้ได้. ชีวิต เป็นเพลง ร้องมัน. ชีวิต เป็นการตีนรน ยอมรับมัน. ชีวิต เป็นโศกนาฏกรรม เพชญหน้ามัน. ชีวิต เป็นความท้าทาย กล้า घ呑มัน. ชีวิต เป็นโชค ทำมันให้เกิด. ชีวิต มีค่ามาก อย่าทำลายมัน. ชีวิต คือชีวิต สูมัน.”

—แม่เซเรชา

### แบบฝึกหัด 8.1

จากหัวข้อ 8.3 (สมการ 8.12) จงแสดงให้เห็นว่า

$$\begin{aligned} & \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln \left\{ p(\mathbf{z}_1|\boldsymbol{\pi}) \cdot \left( \prod_{t=2}^T p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \right) \cdot \prod_{\tau=1}^T p(\mathbf{x}_\tau|\mathbf{z}_\tau, \boldsymbol{\phi}) \right\} \\ &= \sum_{k=1}^K q_{1k} \cdot \ln p(\mathbf{z}_1|\boldsymbol{\pi}) + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(\mathbf{x}_\tau|\mathbf{z}_\tau, \boldsymbol{\phi}) \end{aligned}$$

เมื่อ  $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) = p(\mathbf{z}_1, \dots, \mathbf{z}_T|\mathbf{X}, \boldsymbol{\theta}_0)$  โดย  $\mathbf{z}_t$  แสดงด้วยรหัสหนึ่งร้อน และ  $q_{tk} \equiv p(z_{tk} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$  กับ  $R_{j,k}^{(t-1,t)} \equiv p(z_{t-1,j} = 1, z_{t,k} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$ . คำให้ แบบจำลองนี้ อาศัยเงื่อนไขมาრคอฟ.

### แบบฝึกหัด 8.2

จงแสดงให้เห็นว่า ค่าของ  $\boldsymbol{\pi}_k$  และค่าของ  $A_{jk}$  ดังระบุในสมการ 8.13 และ 8.14 จะทำให้พิงก์ชันควรจะเป็น (สมการ 8.12) มีค่าสูงสุด ขณะที่ยังรักษาเงื่อนไขความน่าจะเป็น  $\sum_k \boldsymbol{\pi}_k = 1$  และ  $\sum_k A_{jk} = 1$  ไว้ได.

คำให้ พิงก์ชันจุดประสงค์ที่รวมเงื่อนไขแล้ว<sup>7</sup> จะเป็นดังสมการ 8.39.

$$\mathcal{G}(\boldsymbol{\theta}) = \varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) - \lambda_1 \left( \sum_k \boldsymbol{\pi}_k - 1 \right) - \lambda_2 \left( \sum_k A_{jk} - 1 \right) \quad (8.39)$$

เมื่อ  $\lambda_1 \geq 0$  และ  $\lambda_2 \geq 0$  เป็น Lagrange multiplier เพื่อช่วยรักษาเงื่อนไข  $\sum_k \boldsymbol{\pi}_k = 1$  และ  $\sum_k A_{jk} = 1$ .

<sup>7</sup> พิงก์ชันจุดประสงค์ที่รวมเงื่อนไข อาจตั้งแบบอื่น เช่น แบบที่ใช้วิธีการลงโทษ (หัวข้อ 2.3) ได้ แต่การวิเคราะห์อาจจะซับซ้อนขึ้น.

ตัวอย่างของการวิเคราะห์ค่า  $\pi_k$  อาจแสดงดังนี้ ณ ที่ค่าทำให้มากที่สุด ค่าอนุพันธ์  $\partial \mathcal{G}(\boldsymbol{\theta}) / \partial \pi_k = 0$ .  
นั่นคือ

$$\begin{aligned}\frac{\partial \mathcal{E}(\boldsymbol{\theta}, \boldsymbol{\theta}_0)}{\partial \pi_k} - \lambda_1 \frac{\partial (\sum_j \pi_j - 1)}{\partial \pi_k} &= 0 \\ \frac{\partial \sum_j q_{1j} \cdot \ln \pi_j}{\partial \pi_k} - \lambda_1 &= 0 \\ \frac{q_{1k}}{\pi_k} &= \lambda_1 \\ \pi_k &= \frac{q_{1k}}{\lambda_1}.\end{aligned}\tag{8.40}$$

ด้วยเงื่อนไข  $\sum_j \pi_j = 1$  เมื่อแทนสมการ 8.40 ลงไปจะได้

$$\begin{aligned}\sum_j \frac{q_{1j}}{\lambda_1} &= 1 \\ \lambda_1 &= \sum_j q_{1j}.\end{aligned}\tag{8.41}$$

เมื่อนำผลจากสมการ 8.41 กลับไปประมวลกับสมการ 8.40 จะได้  $\pi_k = \frac{q_{1k}}{\sum_j q_{1j}}$  ซึ่งคือสมการ 8.13.

### แบบฝึกหัด 8.3

ด้วยเงื่อนไขของแบบจำลองมาร์คอฟช่องเร้น จงพิสูจน์คุณสมบัติในสมการ 8.18 ถึง 8.25.

### แบบฝึกหัด 8.4

จงศึกษาระบบแต่งเพลงอัตโนมัติ แนวทางปฏิบัติ การวัดผล และข้อมูลที่นิยม และสร้างระบบการจำแนก  
อรามณ์ พร้อมประเมินผล.

### แบบฝึกหัด 8.5

จงศึกษาศาสตร์การจำลองแบบ โดยเฉพาะสำหรับข้อมูลชุดลำดับ ในเชิงกว้าง ถึงแบบจำลอง ขั้นตอนวิธี  
และกลไกที่เป็นศาสตร์และศิลป์ หรือคิดว่านำเสนอ รวมไปถึง ปัจจัยหรือประเด็นที่ควรใส่ใจ การประยุกต์ใช้  
เด่น ๆ และภารกิจต่าง ๆ และอภิปรายโอกาสการประยุกต์ใช้ต่าง ๆ และความท้าทายต่าง ๆ ในงานวิจัย แล้ว  
สรุปและให้ความเห็น.

### แบบฝึกหัด 8.6

จากแบบฝึกหัด 8.5 จงเลือกประเด็น แบบจำลอง ขั้นตอนวิธี หรือกลไก ที่สนใจ แล้วศึกษาเรื่องดังกล่าว  
ตั้งคำถามที่เกี่ยวข้อง ดำเนินการหาคำตอบ และสรุปผล.

หมายเหตุ การตั้งคำถาม ควรเป็นคำถามปลายเปิด ซึ่งจะนำไปสู่คำตอบที่น่าสนใจ เช่น หากสนใจโครงสร้างของแบบจำลองความจำรยะสั้น แทนที่จะตั้งคำถามว่า “หากตัดประตู้ลีมออกไปแล้ว แบบจำลองจะยังทำงานได้หรือไม่?” ซึ่งคำตอบจะเป็น แค่ ใช่หรือไม่ใช่. คำถามแบบนี้ ไม่น่าสนใจ. คำถามที่ดีกว่า อาจจะเป็น “ประตู้ลีม ช่วยการทำงานในกรณีกับข้อมูลลักษณะแบบไหน และช่วยได้มากน้อยเท่าไรในแต่ละกรณี เมื่อวัดผลโดยสมบูรณ์ และเมื่อเปรียบเทียบกับประตู้อื่น ๆ?” ซึ่งเป็นคำถามปลายเปิด และจะนำไปสู่คำตอบที่เดาได้ยาก มีความลึก น่าสนใจ และตัวคำตอบเอง ก็จะมีประโยชน์มากกว่าด้วย. กฎที่ ๑ ไป คือ หากคำถามได้ สามารถตอบได้เลย โดยไม่ต้องทำการศึกษาเพิ่มเติม หรือศึกษาเพียงเล็กน้อย หรือ เดาคำตอบได้ง่าย ๆ คำถามนี้ไม่น่าสนใจ.



## บทที่ 9

# การรู้จำรูปแบบเชิงลำดับในโลกการประมวลผลภาษาธรรมชาติ

“Yet the truly unique feature of our language is not its ability to transmit information about men and lions. Rather, it's the ability to transmit information about things that do not exist at all.”

---Yuval Noah Harari

“ลักษณะเด่นจริง ๆ ของภาษามนุษย์ ไม่ได้อยู่ที่ความสามารถในการแปลงสารสนเทศ เกี่ยวกับ คนกับสิงโต. แต่มันคือ ความสามารถในการแปลงสารสนเทศเกี่ยวกับสิ่งที่ไม่ได้มีอยู่จริงเลย.”

—ยูวาล โนอาห์ ไฮรา

ภาษา เป็นแบบจำลองคร่าว ๆ ของความคิด และเป็นตัวแทนที่หยาบมาก ๆ สำหรับบรรยายโลกและอธิบายความเป็นจริง. เพียงแต่ มันยังคงเป็นเครื่องมือที่ดีที่สุดอย่างหนึ่งเท่าที่เรามี สำหรับการถ่ายทอดความคิดและสื่อสารเรื่องราว. แฮร์มัnn เหสเซอ กล่าวว่า “ทุก ๆ อย่างที่คิด และแสดงออกมาเป็นคำพูด จะลำเอียงไปข้างเดียว คริ่งเดียวของความจริง ขาดความครบถ้วน ขาดความสมบูรณ์ ขาดเอกภาพ.” (Hermann Hesse: “Everything that is thought and expressed in words is one-sided, only half the truth; it all lacks totality, completeness, unity.”)

## 9.1 การประมวลผลภาษาธรรมชาติ

ภาษา หรือในบริบทของคอมพิวเตอร์ จะเรียกว่า **ภาษาธรรมชาติ** (natural language) เพื่อเน้นความแตกต่างจากภาษาโปรแกรม (programming language) ที่ใช้สำหรับเขียนโปรแกรมให้กับคอมพิวเตอร์. ภาษาธรรมชาติ เป็นภาษาที่คนใช้พูดสื่อสารกัน เช่น ภาษาไทย ภาษาจีน ภาษาสเปน ภาษาอังกฤษ. ภาษาธรรมชาติ

ชาติ<sup>1</sup> มีการเกิด การพัฒนา การวิจัยและการตามธรรมชาติ. วิวัฒนาการของภาษาเกิดจากคนจำนวนมาก และผ่านผู้คนหลายรุ่น แม้ว่าอาจมีบางครั้งที่ได้รับการควบคุม ปรับปรุง ผ่านกลุ่มคนจำนวนน้อย ๆ ที่มีอำนาจหรือที่ได้รับมอบหมายบ้าง. ส่วนภาษาโปรแกรม เป็นภาษาที่ออกแบบจากคนหรือกลุ่มคน (จำนวนไม่มาก) เพื่อใช้สั่งงานคอมพิวเตอร์. ตัวอย่างภาษาโปรแกรม เช่น ภาษาซี ภาษาซีพลัสพลัส ภาษาจawa ภาษาอาร์ ภาษาไฟรอน. ภาษาโปรแกรม จะมีไวยากรณ์ที่ติดตัว ใช้ควบคุมโครงสร้างของคำสั่งต่าง ๆ. ขณะที่ไวยากรณ์ของภาษาธรรมชาติ มักจะยืดหยุ่น และมีข้อยกเว้นอยู่มาก.

ไวยากรณ์ (syntax) คือกฎเกณฑ์ที่เกี่ยวกับโทเค็น และโครงสร้าง. โทเค็น (token) เป็นหน่วยพื้นฐานของภาษาที่มีความหมาย เช่น ในภาษาธรรมชาติ โทเค็น หมายถึง คำ. ในภาษาโปรแกรม โทเค็น หมายถึง คำ, ตัวแปร, ค่าตัวแปร, ค่าคงที่, นิพจน์, พังก์ชัน, ออปเจ็ค, เมธอด เป็นต้น. โครงสร้างไวยากรณ์ คือการนำโทเค็นไปประกอบกันเพื่อสื่อความหมาย. ความสัมพันธ์ระหว่างโทเค็นและโครงสร้างไวยากรณ์ อาจแสดงได้ด้วยตัวอย่าง เช่น ในภาษาอังกฤษ “This 1s @t English s3ntence.” มีการใช้โทเค็นที่ไม่ถูกต้อง. ส่วน “is.sentence ThisEnglish an” แม้จะใช้โทเค็นที่ถูกต้องทั้งหมด แต่เป็นการประกอบกันที่ไม่ถูกต้องตามไวยากรณ์ภาษาอังกฤษ. ประโยค “This is an English sentence.” ถูกไวยากรณ์ภาษาอังกฤษ (โทเค็นถูกต้องทั้งหมด และประกอบกันเป็นโครงสร้างที่ถูกต้อง). การวิเคราะห์โครงสร้างไวยากรณ์ของข้อความ หรือประโยค จะเรียกว่า การแยกส่วน (parsing). เวลาที่เราอ่านข้อความต่าง ๆ เราทำการแยกส่วน เพื่อเข้าใจรูปประโยค ประกอบการทำเข้าใจความหมายของข้อความ.

นอกจากที่ภาษาโปรแกรมมีไวยากรณ์ที่ติดตัวมากกว่าภาษาธรรมชาติแล้ว ยังมีประเด็นที่แตกต่างกัน ดังนี้ (1) ความกำกวມ (ambiguity) ที่ภาษาธรรมชาติมักอาศัยบริบทและสามัญสำนึกประกอบในการทำความเข้าใจข้อความ ในขณะที่ภาษาโปรแกรม ถูกออกแบบให้มีความชัดเจนโดยสมบูรณ์ ตีความได้อย่างเดียว ไม่มีความกำกวມเลย, (2) ความซ้ำซ้อน (redundancy) ที่พบได้บ่อย ๆ ในภาษาธรรมชาติ แต่ภาษาโปรแกรมจะกระชับและไม่ซ้ำซ้อน, (3) ความตรงตามตัวอักษร (literalness) ที่ภาษาโปรแกรมบอกความหมายที่เจาะจง ตรงตามตัวอักษร ในขณะที่ภาษาธรรมชาติ มีการใช้สำนวน โวหาร คำเปรียบเปรย.

ด้วยความแตกต่างระหว่างภาษาธรรมชาติและภาษาโปรแกรม ทำให้การประมวลผลภาษาธรรมชาติต้อง- การเครื่องมือ แนวทาง และกลไกเฉพาะ ที่นอกเหนือไปจากการยึมมาจากวิธีการต่าง ๆ ในการประมวลผล โปรแกรม.

**การประมวลผลภาษาธรรมชาติ** (Natural Language Processing คำย่อ NLP) เป็นศาสตร์ที่ใช้วิธีการ

<sup>1</sup> คำอธิบายในส่วนนี้ ได้รับอิทธิพลหลัก ๆ จาก [58]

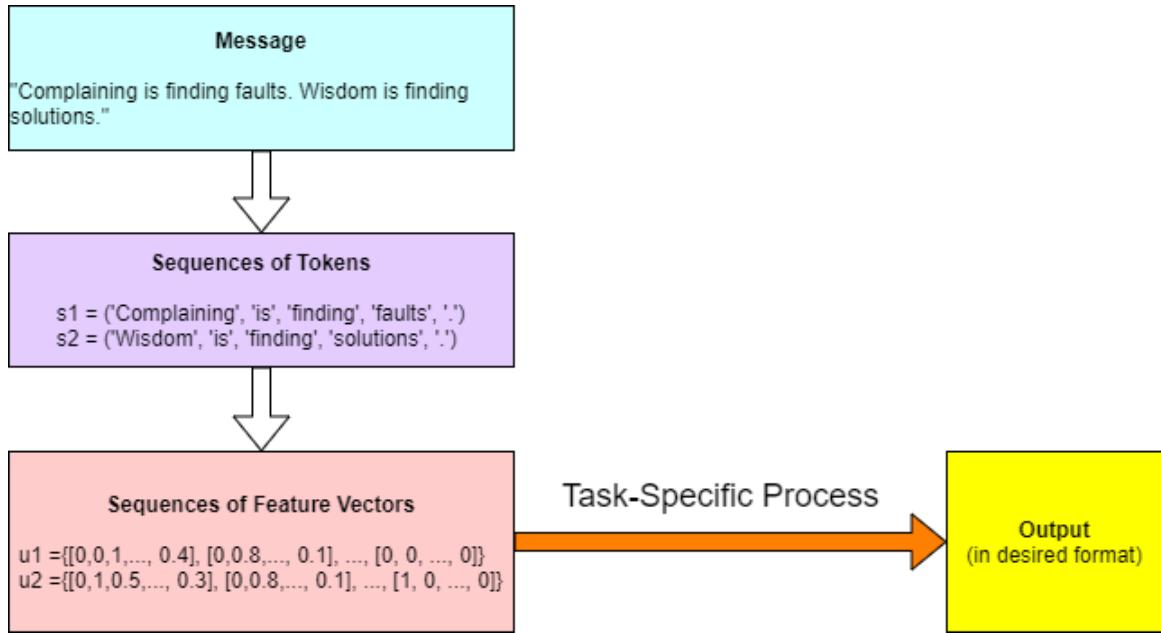
ต่าง ๆ เพื่อให้คอมพิวเตอร์สามารถนำข้อมูลความในภาษาธรรมชาติไปประมวลผล และให้ผลลัพธ์ตามจุดประสงค์ของภาระกิจที่ต้องการ.

ภาระกิจของการประมวลผลภาษาธรรมชาตินั้นมีหลากหลายมาก เช่น ระบบตรวจสอบภาษา (spelling and grammar correction), ระบบช่วยจบคำอัตโนมัติ (autocomplete), ระบบตอบคำถามอัตโนมัติ (question answering system), การสรุปข้อความ (text summarization), แชทบอต (chatbot), การจำแนกอารมณ์ (sentiment classification), ระบบแปลภาษาอัตโนมัติ (machine translation), การค้นหาเนื้อหาในเอกสาร (content searching), การตรวจสอบการลอกเลียนวรรณกรรม (plagiarism detection) และการสร้างข้อความอัตโนมัติ (text generation). ลักษณะเฉพาะของภาษาไทย มีส่วนอย่างมากต่อความต้องการและความจำเป็นของภาระกิจต่าง ๆ เช่น ภาษาอังกฤษมีขอบเขตคำและขอบเขตประโยคที่ชัดเจน การตัดคำ (word segmentation) และตัดประโยค (sentence segmentation) ในภาษาอังกฤษทำได้ง่ายมาก เมื่อเทียบกับภาษาไทย. ดังนั้นในขณะที่ ระบบอัตโนมัติสำหรับการตัดคำและการตัดประโยคในภาษาอังกฤษมีความสมบูรณ์เต็มที่และพร้อมใช้งาน ความสามารถของการตัดคำและการตัดประโยคอัตโนมัติในภาษาไทย กลับอยู่ในระดับเริ่มต้น และยังต้องการการพัฒนาอีกมาก. การตัดคำและการตัดประโยค นอกจากจะใช้ประกอบการจัดแสดงหน้าเอกสาร (ในการตัดคำขึ้นบรรทัดใหม่) การตัดคำและการตัดประโยค จัดเป็นภาระกิจพื้นฐานของการประมวลผลภาษาธรรมชาติ ที่จะช่วยให้งานที่มีความซับซ้อนอื่น ๆ สามารถประมวลผลต่อไปได้อย่างมีประสิทธิภาพ.

ภาพรวมของการประมวลผลภาษาธรรมชาติ โดยเฉพาะภาษาอังกฤษ<sup>2</sup> คือ อินพุตที่ข้อความ จะถูกแปลงเป็นชุดลำดับของโถกเด็น ซึ่งแต่ละโถกเด็นเป็นคำ. จากนั้นแต่ละโถกเด็น จะถูกแปลงเป็นเวกเตอร์ลักษณะสำคัญ ซึ่งเป็นเวกเตอร์ของค่าต่าง ๆ ที่เป็นตัวเลข ก่อนจะเข้ากระบวนการประมวลผลตามแต่ภาระกิจ. รูป 9.1 แสดงแนวทางการประมวลผลภาษาธรรมชาติ โดยทั่วไป ที่แปลงข้อความภาษาธรรมชาติ ไปเป็นชุดลำดับต่าง ๆ ของค่าเวกเตอร์ลักษณะสำคัญ ก่อนจะเข้าประมวลผล. แนวทางเช่นนี้ ทำให้สามารถใช้แบบจำลองเชิงลำดับต่าง ๆ ที่ทำงานกับข้อมูลที่มีค่าเป็นตัวเลข มาช่วยการประมวลผลตามแต่ภาระกิจได้.

รูป 9.2 แสดงลักษณะภารกิจการระบุหมวดคำ (Part-Of-Speech Tagging) ที่รับอินพุตเป็นข้อความ (ลำดับของคำ) และให้อาต์พุต ออกมายเป็นลำดับของหมวดคำ โดยลำดับของอาต์พุตจะสอดคล้องกับลำดับของอินพุต. ในภาพ ข้อความอินพุตถูกแบ่งออกเป็นโถกเด็น ซึ่ง ณ ที่นี่ แต่ละโถกเด็นคือ คำ และอาต์พุต ก็เป็นชุดลำดับข้อมูล ที่แต่ละจุดข้อมูลจะสอดคล้องกับแต่ละโถกเด็น.

<sup>2</sup> ภาษาไทยมีลักษณะเฉพาะหลายอย่าง โดยเฉพาะความคลุมเครือของขอบเขตคำและขอบเขตประโยค ทำให้อาจต้องการความคิดสร้างสรรค์ และกระบวนการวิเคราะห์ใหม่ ที่ต่างจากภาษาอังกฤษที่อภิปรายนี้.



รูปที่ 9.1: ภาพรวมของการประมวลผลภาษาธรรมชาติ. กล่องแสดงตัวอย่างลักษณะของข้อมูล และลูกศรแทนกระบวนการแปลงข้อมูล. ข้อความภาษาธรรมชาติ จะผ่านขั้นตอนต่อๆ กันเพื่อแปลงเป็นชุดลำดับของคำเวลาเตอร์ลักษณะสำคัญ ก่อนจะเข้าประมวลผลตามภารกิจที่ต้องการ.

POS tagging														
input	Mistakes	are	always	forgivable	,	if	one	has	the	courage	to	admit	them	.
output	noun	verb	adverb	adjective		preposition	number	verb	determiner	noun	preposition	verb	pronoun	
ground truth	noun	verb	adverb	adjective		preposition	pronoun	verb	determiner	noun	preposition	verb	pronoun	

รูปที่ 9.2: ตัวอย่างอินพุตเอาต์พุตของการกิจกรรมการระบุหมวดคำ. อินพุตเป็น คำพูดของบรรณาธิการ และเอาต์พุต เป็นตัวอย่างผลลัพธ์จากระบบการระบุหมวดคำ. บรรทัดสุดท้าย แสดงเฉลย.

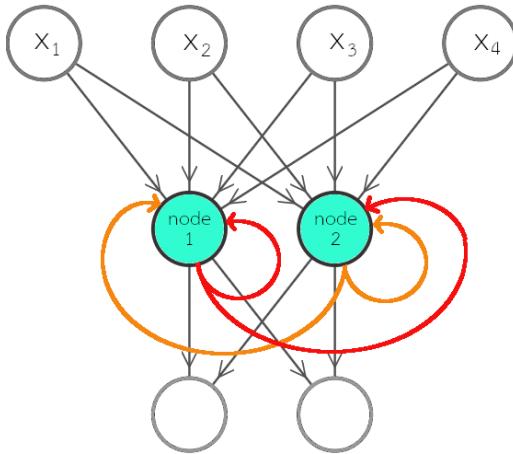
## 9.2 โครงข่ายภาษาเวียนกลับ

โครงข่ายภาษาเวียนกลับ (Recurrent Neural Network คำย่อ RNN) เป็นแบบจำลองโครงข่ายภาษา เที่ยม ที่อินพุตของแต่ละหน่วยย่อย นอกจากจะเป็นค่าของเอาต์พุตจากหน่วยย่อยในชั้นคำนวนก่อนหน้าแล้ว ยังสามารถเป็นค่าของเอาต์พุตของหน่วยย่อยในชั้นคำนวนเดียวกัน สำหรับจุดข้อมูลลำดับก่อนหน้าได้. เช่น เดียวกับการวิเคราะห์การคำนวนของโครงข่ายเป็นชั้นคำนวน ดังอภิปรายในบทที่ 6 โครงข่ายภาษาเวียนกลับ ก็สามารถมองเป็นการประกอบกันของชั้นคำนวนเวียนกลับ (recurrent layer) ได้.

การคำนวนของหน่วยย่อยในชั้นคำนวนเวียนกลับที่  $q^{th}$  อาจเขียนได้ดังนี้

$$a_j^{(q)}(t) = \sum_{i=1}^D w_{ji}^{(q)} \cdot z_i^{(q-1)}(t) + \sum_{m=1}^M v_{jm}^{(q)} \cdot z_m^{(q)}(t-1) + b_j^{(q)} \quad (9.1)$$

$$z_j^{(q)}(t) = h(a_j^{(q)}(t)) \quad (9.2)$$

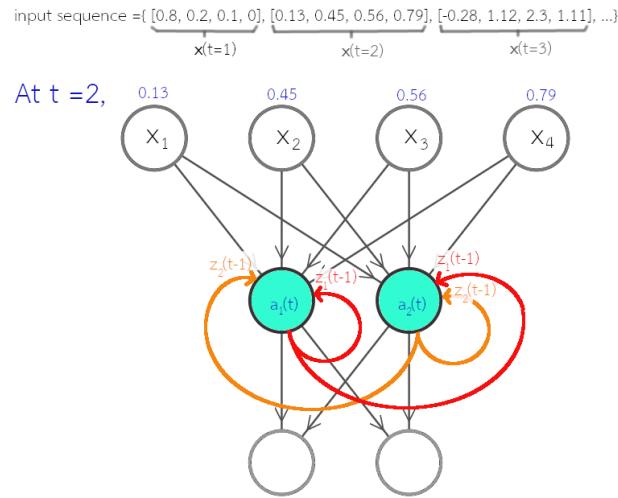


รูปที่ 9.3: ตัวอย่างโครงข่ายประสาทเวียนกลับ โดยเน้นเส้นทางข้อมูลป้อนเวียนกลับ. โครงข่ายประกอบด้วยสามชั้นคำนวณ โดยชั้นอินพุต (อยู่บนสุด) มีสีเทา (รับอินพุตสีมิตร ได้แก่  $x_1$  ถึง  $x_4$ ) ชั้นที่สอง เป็นชั้นเวียนกลับ มีสีของหน่วย (แสดงด้วยวงกลมสีฟ้าเขียว) และชั้นที่สาม (อยู่ล่างสุด). เส้นทางการส่งข้อมูลเวียนกลับ แสดงด้วย เส้นสีแดง (ค่าเวียนกลับจากหน่วยแรก) และเส้นสีส้ม (ค่าเวียนกลับจากหน่วยที่สอง).

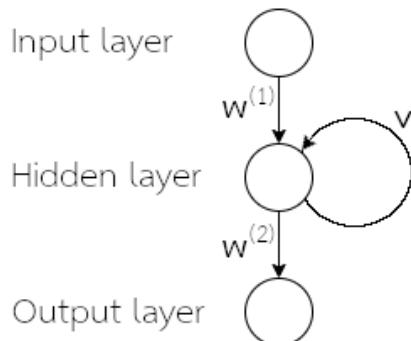
เมื่อ  $a_j^{(q)}(t)$  คือค่าตัวกระตุนของหน่วยย่อยที่  $j^{th}$  สำหรับจุดข้อมูลลำดับที่  $t^{th}$ . ตัวแปร  $z_j^{(q)}(t)$  คือ ผลการกระตุน หรือบางครั้งอาจเรียก ว่าเป็น **สถานะช่อง** ของหน่วยที่  $j^{th}$  ในชั้นคำนวณ  $q^{th}$  สำหรับจุดข้อมูลลำดับเวลา  $t^{th}$  โดย  $D$  คือจำนวนหน่วยย่อยในชั้น  $(q-1)^{th}$  และ  $M$  คือจำนวนหน่วยย่อยในชั้น  $q^{th}$ . ตัวแปร  $w_{jd}^{(q)}$  เป็นค่าน้ำหนักของการเชื่อมต่อระหว่างหน่วยที่  $d^{th}$  ของชั้น  $(q-1)^{th}$  กับหน่วยที่  $j^{th}$  ของชั้นคำนวณ  $q^{th}$ . ตัวแปร  $v_{jm}^{(q)}$  เป็นค่าน้ำหนักของการเชื่อมต่อของหน่วยที่  $m^{th}$  เวียนกลับมาเข้าหน่วยที่  $j^{th}$  ของชั้นคำนวณเดียวกัน. ส่วน  $b_j^{(q)}$  คือค่าไบอัสของหน่วยที่  $j^{th}$  และ  $h(\cdot)$  คือฟังก์ชันกระตุน.

เมื่อเปรียบเทียบสมการ 9.1 กับสมการ 3.16 ซึ่งเป็นการคำนวณของโครงข่ายแพร่กระจายไปข้างหน้า จะเห็นว่าจุดต่างที่สำคัญ คือ พจน์  $\sum_{m=1}^M v_{jm}^{(q)} \cdot z_m^{(q)}(t-1)$  ซึ่งเป็นการนำผลการกระตุนที่ลำดับเวลาก่อน เข้ามาคำนวณด้วย. รูป 9.3 แสดงตัวอย่างโครงสร้างการเชื่อมต่อของโครงข่ายประสาทเวียนกลับ ที่อินพุตมีสีมิตร และเอาต์พุตมีสีของมิตร โดยชั้นคำนวณที่สอง ซึ่งเป็นชั้นเวียนกลับ มีหน่วยย่อยสองหน่วย. รูป 9.4 แสดงตัวอย่าง โครงข่ายประสาทเวียนกลับ พร้อมตัวอย่างชุดข้อมูลลำดับ และตัวแปรที่สำคัญ.

รูป 9.3 แสดงโครงข่ายประสาทเวียนกลับ โดยเน้นการแสดงโครงสร้าง. อย่างไรก็ตาม หากชั้นเวียนกลับมีจำนวนหน่วยมาก ๆ การเขียนแผนภาพเช่นนี้ จะดูยุ่งเหยิงมาก (แต่ละหน่วยส่งค่าเวียนกลับไปให้ทุก ๆ หน่วยในชั้น). บ่อยครั้ง แผนภาพโครงข่ายประสาทเวียนกลับ จึงมักถูกแสดงโดยใช้วงกลมแค่หนึ่งวงแทนชั้นคำนวณทั้งชั้น (ไม่ว่าภายในชั้นจะใช้จำนวนหน่วยเท่าใด) ดังแสดงในรูป 9.5. นอกจากนั้น ในบางสถานการณ์ การใช้แผนภาพคลี่ลำดับ (unfolding diagram) ที่แสดงข้อมูลการเวียนกลับ ด้วยการกระจายออกตามลำดับเวลา อาจช่วยให้เข้าใจแนวคิดได้ดีกว่า. แผนภาพคลี่ลำดับ อาจแสดงดังรูป 9.6.



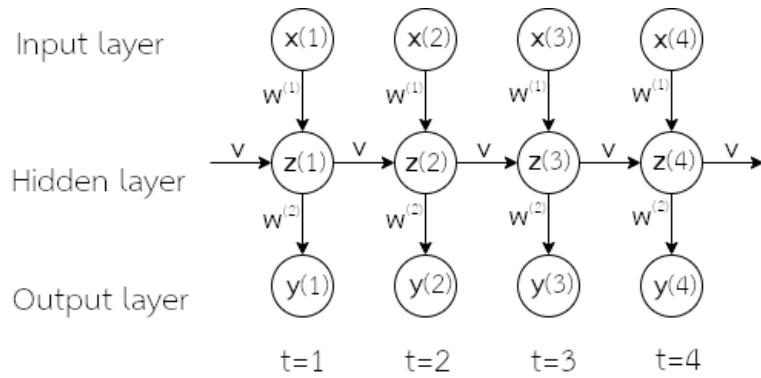
รูปที่ 9.4: ตัวอย่างโครงข่ายประสาทเวียนกลับ พร้อมตัวอย่างชุดข้อมูลลำดับ โดยเน้นตัวแปรที่สำคัญ. ในภาพ แสดงการคำนวณ ณ จุดข้อมูลลำดับที่  $t = 2$  ซึ่ง  $a_1(t) = w_{11}(0.13) + w_{12}(0.45) + w_{13}(0.56) + w_{14}(0.79) + v_{11}z_1(t-1) + v_{12}z_2(t-1) + b_1$  และ  $a_2(t) = w_{21}(0.13) + w_{22}(0.45) + w_{23}(0.56) + w_{24}(0.79) + v_{21}z_1(t-1) + v_{22}z_2(t-1) + b_2$ .



รูปที่ 9.5: แผนภาพโครงสร้างโดยรวมของโครงข่ายประสาทเวียนกลับ โดยวงกลมแทนชั้นคำนวณทั้งชั้น (โดยไม่ระบุจำนวนหน่วยคำนวณภายในชั้น). เส้นทางข้อมูล ระบุ  $w^{(1)}$ ,  $w^{(2)}$  และ  $v$  สำหรับค่าน้ำหนักการแพร่กระจายไปข้างหน้า ของชั้นคำนวณที่หนึ่ง กับของชั้นคำนวณที่สอง และค่าน้ำหนักเวียนกลับ (ของชั้นคำนวณที่หนึ่ง แต่ตัวยกกำลังไว้ เพื่อความกระชับ).

จากแผนภาพคลีลำดับ ในรูป 9.6 สังเกต (1) ทุก ๆ ลำดับเวลา การคำนวณใช้ค่าน้ำหนักชุดเดียวกัน (ที่เวลา  $t$  ต่าง ๆ ใช้ค่า  $w^{(1)}$ ,  $w^{(2)}$  และ  $v$  เมื่อนกัน) (2) ผลการกระตุ้นของจุดข้อมูลลำดับเวลาได ๆ  $z(t)$  จะส่งผลต่อเอาร์พุตผ่านหลายเส้นทาง (เส้นทางตรง ส่งผลต่อ  $y(t)$  และเส้นทางเวียนกลับเอาร์พุตอื่น ๆ หลังจากลำดับเวลานั้น ๆ ผ่านเส้นทางการเวียนกลับ).

**เกรเดียนต์ของชั้นเวียนกลับ.** ในลักษณะเดียวกับโครงข่ายแพร่กระจายไปข้างหน้าและโครงข่ายคอนโวต-ชั้น การฝึกโครงข่ายประสาทเวียนกลับ สามารถทำได้โดยปรับค่าน้ำหนักต่าง ๆ โดยอาศัยการแพร่กระจาย



รูปที่ 9.6: แผนภาพคลื่ล้ำดับของโครงข่ายประสาทเวียนกลับ.

ย้อนกลับ ซึ่งทำการคำนวณค่าเกรเดียนต์เป็นชั้น ๆ.

การแพร่กระจายย้อนกลับ สำหรับชั้นคำนวณเวียนกลับ สามารถทำได้อย่างมีประสิทธิภาพ ด้วยขั้นตอนวิธี หลาย ๆ วิธี[78] ไม่ว่าจะเป็น การเรียนรู้เวียนกลับเวลาจริง (real time recurrent learning[167]) หรือการแพร่กระจายย้อนกลับผ่านเวลา (backpropagation through time[218, 215] 俗稱 BPTT) โดยเกรฟช์[78] ให้ความเห็นว่า การแพร่กระจายย้อนกลับผ่านเวลา เข้าใจได้ง่ายกว่า และสามารถคำนวณได้อย่างมีประสิทธิภาพมากกว่า. เกรเดียนต์ของชั้นเวียนกลับ ดังที่จะอธิบายต่อไปนี้ ใช้แนวทางของการแพร่กระจายย้อนกลับผ่านเวลา เช่นเดียวกับการอธิบายของเกรฟช์[78].

ทำนองเดียวกัน กำหนดให้  $E$  เป็นพังก์ชันค่าผิดพลาด และ

$$\delta_j^{(q)}(t) \equiv \frac{\partial E}{\partial a_j^{(q)}(t)}. \quad (9.3)$$

จากการที่ ค่าการกระตุ้น  $a_j^{(q)}(t)$  ส่งผลต่อ  $E$  ผ่านผลการกระตุ้น  $z_j^{(q)}(t)$  และกฎลูกโซ่ เราจะได้

$$\begin{aligned} \frac{\partial E}{\partial a_j^{(q)}(t)} &= \frac{\partial E}{\partial z_j^{(q)}(t)} \cdot \frac{\partial z_j^{(q)}(t)}{\partial a_j^{(q)}(t)} \\ &= \frac{\partial E}{\partial z_j^{(q)}(t)} \cdot h'(a_j^{(q)}(t)). \end{aligned} \quad (9.4)$$

เมื่อพิจารณา เราจะเห็นว่า ผลการกระตุ้น  $z_j^{(q)}(t)$  ส่งอิทธิพลต่อ  $E$  ผ่านสองเส้นทาง คือ (1) ผ่านการแพร่กระจายไปข้างหน้า (ผ่านชั้นคำนวณต่อไป) และ (2) ผ่านการเวียนกลับ (ผ่านชั้นคำนวณเดิม แต่สำหรับ

ลำดับเวลาถัดไป). ดังนั้น ด้วยกฎลูกโซ่ เราจะได้

$$\frac{\partial E}{\partial z_j^{(q)}(t)} = \sum_k \frac{\partial L}{\partial a_k^{(q+1)}(t)} \cdot \frac{\partial a_k^{(q+1)}(t)}{\partial z_j^{(q)}(t)} + \sum_m \frac{\partial L}{\partial a_m^{(q)}(t+1)} \cdot \frac{\partial a_m^{(q)}(t+1)}{\partial z_j^{(q)}(t)} \quad (9.5)$$

$$= \sum_k \delta_k^{(q+1)}(t) \cdot w_{kj}^{(q+1)} + \sum_m \delta_m^{(q)}(t+1) \cdot v_{mj}^{(q)}. \quad (9.6)$$

จากสมการ 9.4 และ 9.6 เราจะได้

$$\delta_j^{(q)}(t) = h'(a_j^{(q)}(t)) \cdot \left( \sum_k \delta_k^{(q+1)}(t) \cdot w_{kj}^{(q+1)} + \sum_m \delta_m^{(q)}(t+1) \cdot v_{mj}^{(q)} \right). \quad (9.7)$$

สุดท้าย เมื่อพิจารณากรเดียนต์ต่อค่าน้ำหนักต่าง ๆ ซึ่งค่าน้ำหนักต่าง จะถูกใช้คำนวณสำหรับทุก ๆ ลำดับเวลาเหมือนกัน ดังนั้น

$$\frac{\partial E}{\partial w_{ji}^{(q)}} = \sum_t \frac{\partial E}{\partial a_j^{(q)}(t)} \cdot \frac{\partial a_j(t)^{(q)}}{\partial w_{ji}^{(q)}} \quad (9.8)$$

$$= \sum_t \delta_j^{(q)}(t) \cdot z_i^{(q-1)}(t) \quad (9.9)$$

และ

$$\frac{\partial E}{\partial v_{jm}^{(q)}} = \sum_t \frac{\partial E}{\partial a_j^{(q)}(t)} \cdot \frac{\partial a_j(t)^{(q)}}{\partial v_{jm}^{(q)}} \quad (9.10)$$

$$= \sum_t \delta_j^{(q)}(t) \cdot z_m^{(q)}(t-1) \quad (9.11)$$

เข่นเดียวกับค่าน้ำหนัก ค่าไบอัส  $b_j^{(q)}$  สามารถคำนวณได้จาก  $\frac{\partial E}{\partial b_j^{(q)}} = \sum_t \frac{\partial E}{\partial a_j^{(q)}(t)} \cdot \frac{\partial a_j(t)^{(q)}}{\partial b_j^{(q)}}$  ซึ่งจะได้ว่า

$$\frac{\partial E}{\partial b_j^{(q)}} = \sum_t \delta_j^{(q)}(t). \quad (9.12)$$

การวิเคราะห์ค่าที่ใช้ในการเริ่มต้นการคำนวณ สามารถทำได้ในลักษณะเดียวกับที่ทำกับโครงข่ายแพร่กระจายไปข้างหน้า. นั่นคือ พงก์ชั้นค่าผิดพลาด อาจนิยามเป็น

$$E = \frac{1}{T} \sum_t \sum_k E_k(t) \quad (9.13)$$

เมื่อ  $T$  เป็นจำนวนลำดับ โดย  $E_k(t)$  เป็นค่าผิดพลาดของมิติ  $k^{th}$  ที่ลำดับเวลา  $t^{th}$ . หากลักษณะภาระกิจถูกตีกรอบเป็นการหาค่าทดแทน เราอาจกำหนด

$$E_k(t) = \frac{\mathcal{M}(t)}{2} \cdot (\hat{y}_k(t) - y_k(t))^2 \quad (9.14)$$

โดย  $y_k(t)$  เป็นค่าเฉลยของมิติ  $k^{th}$  ที่ลำดับเวลา  $t^{th}$  และ  $\hat{y}_k(t)$  เป็นค่าที่ทำนาย. ส่วน  $\mathcal{M}(t) \in \{0, 1\}$  เป็นเสมือนหน้ากาก (mask) ที่ใช้ควบคุมว่า ณ ลำดับเวลา  $t^{th}$  เราต้องการคิดผลของการทำนายหรือไม่.

การใช้กลไกหน้ากาก แม้จะสามารถใช้ได้ทั่วไป แต่สำหรับบริบทของการอนุமานข้อมูลเชิงลำดับ กลไกนี้ มีความสำคัญอย่างมาก. ภาระกิจการอนุमานข้อมูลเชิงลำดับ มีหลากหลายประเภท (หัวข้อ 8.1). ภาระกิจ บางประเภท อาจต้องการการทำนายค่าสำหรับทุก ๆ ลำดับเวลา (เช่น ระบบตรวจสอบการสะกดคำ ที่ต้องให้ค่าทำนายสะกดถูกหรือผิดของมาสำหรับทุกด้านนี้ลำดับ) ภาระกิจบางประเภท อาจต้องการการทำนายค่า แค่ บางลำดับเวลา (เช่น การจำแนกอารมณ์ ที่อาจจะให้ค่าทำนายของมาเฉพาะที่ด้านนี้ลำดับสุดท้ายเท่านั้น) การใช้กลไกหน้ากาก ช่วยกำหนดด้านนี้ลำดับที่มีผลจริง ๆ ( $\mathcal{M}(t) = 1$  เฉพาะด้านนี้ลำดับ  $t$  ที่มีค่าเฉลย ส่วนนอกนั้นให้  $\mathcal{M}(t) = 0$ ) จะช่วยให้การทำงานกับข้อมูลลำดับยืดหยุ่นและสะดวกมากขึ้น.

เมื่อพิจารณากรเดียโนต์ ด้วยสมการ 9.13 และ 9.14 สำหรับ กรณีการหาค่าทดแทน ซึ่งมักกำหนดให้  $a_k^{(L)}(t) = z_k^{(L)}(t) = \hat{y}_k(t)$  เราจะเห็นว่า

$$\begin{aligned} \delta_k^{(L)}(t) &= \frac{\partial E}{\partial a_k^{(L)}(t)} = \frac{\partial E}{\partial \hat{y}_k(t)} = \frac{1}{T} \sum_{\tau} \sum_j \mathcal{M}(\tau) \cdot (\hat{y}_j(\tau) - y_j(\tau)) \cdot \frac{\partial \hat{y}_j(\tau)}{\partial \hat{y}_k(t)} \\ &= \frac{1}{T} \mathcal{M}(t) \cdot (\hat{y}_k(t) - y_k(t)) \end{aligned} \quad (9.15)$$

สำหรับ  $t = 1, \dots, T$ .

“Wisdom can be learned.

But it cannot be taught .”

---Anthony de Mello

“ปัญญาสามารถเรียนรู้ได้

แต่มันสอนกันไม่ได้.”

—แอนโธนี เดอ เมโล่

**เกร็ดความรู้ เมตตา.** ปัญญาและเมตตาเป็นคุณค่าสูงสุดของมนุษย์. ปัญญา คือ ความรู้ในเรื่องราวตามความเป็นจริง ครอบคลุม ถึงความสามารถในการคิด วิเคราะห์ สังเคราะห์ แก้ปัญหา พัฒนา ตระหนักรู้ โดยใช้ความรู้, ประสบการณ์, ความเข้าใจ, สามัญสำนึก และมุ่งมองที่หลากหลายครบถ้วน. แฮร์มันน์ เอสเซอ กล่าวว่า “ความรู้สามารถสื่อสารกันได้ แต่ไม่ใช่ปัญญา. เราหาปัญญาได้ เราใช้ชีวิตอยู่กับปัญญาได้ เราป้องกันตัวเองจากภัยธรรมดายังปัญญาได้ เราทำสิ่งที่ควรรู้ด้วยปัญญาได้ แต่เราสื่อสารปัญญาออกໄປไม่ได้ เราสอนปัญญาไม่ได้.” (Hermann Hesse: “Knowledge can be communicated, but not wisdom. One can find it,

live it, be fortified by it, do wonders through it, but one cannot communicate and teach it.") ผู้คนและสังคมชี้ช่อง และยกย่องปัญญา แม้หลายครั้งอาจจะสับสนระหว่างปัญญา ความรู้ และความฉลาด.

เมตตา คือ ความปรารถนาให้ชีวิตดี ฯ เป็นสุข ซึ่งรวมทั้งชีวิตสัตว์ ชีวิตคนอื่น และชีวิตของตัวเราเองด้วย. ในความหมายกว้าง ๆ แล้ว ความหมายของเมตตา ยังครอบคลุมไปถึงความปรารถนาให้ชีวิตพื้นทุกชีวิต (กรุณา), ความยินดีเมื่อชีวิตเป็นสุข (มุทิตา) และในบางครั้งก็อาจหมายรวมถึงการปล่อยวาง (อุเบกษา) ด้วย. สำหรับเมตตาแล้ว แม้จะเป็นหนึ่งในสองคุณค่าสูงสุดคู่กับปัญญา แต่สังคมดูเหมือนจะชี้ช่องและยกย่องเมตตาแทน้อยเกินไป โดยเฉพาะเมื่อเปรียบเทียบกับระดับการยกย่องปัญญา. (ดูจากปรัชญาของโรงเรียนและมหาวิทยาลัยต่าง ๆ ทั้งในและต่างประเทศ เป็นตัวอย่าง.)

หมายเหตุ คุณธรรม นั้น อ้างถึงความดี ซึ่งครอบคลุมความหมายกว้าง ๆ และบ่อ依托 ที่ถูกตีความผ่านค่านิยมของสังคมหรือกลุ่มคน. แม้บางครั้งอาจมองว่า คุณธรรมครอบคลุมถึงความเมตตาด้วย แต่เนื่องจากคุณธรรมถูกตีความผ่านค่านิยมของสังคม ความหมายของคุณธรรมจึงขึ้นกับบริบทเป็นอย่างมาก. ด้วยอย่างเช่น [217, 159] คุณธรรมตามค่านิยมของกรีกโบราณ ตามแนวคิดของเพลโต คือ ความรอบคอบ, ความกล้าหาญ, การรู้จักระบันยับยั้งใจ และความยุติธรรม. คุณธรรมตามค่านิยมของชาวนิร (บูชิโด) คือ ความซื่อสัตย์และยุติธรรม, ความกล้าหาญ, ความเมตตา, ความเคารพให้เกียรติกันและกัน, สัจจะวาจา, เกียรติและศักดิ์ศรี, หน้าที่และความภักดี และการระงับอารมณ์ การควบคุมตัวเอง. คุณธรรมตามค่านิยมจีนดั้งเดิม คือ ความเมตตา, ความประหยัดมรดกสัตต์, ความอ่อนน้อมถ่อมตน และความกดดัน. คุณธรรมแก่นตามแนวคิดจิตวิทยาจีน คือ ปัญญาและความรู้ (ความอยากรู้อยากเห็น, ความคิดสร้างสรรค์, การเปิดกว้างทางความคิด, การรักที่จะเรียนรู้ และการมีมนุษย์มองที่หลักหลาน), ความกล้าหาญ (ความอาจหาญในการเผชิญความเสี่ยงหรืออันตราย, ความมุนานะอุตสาหะ, ความซื่อสัตย์มั่นคง และความกระตือรือร้น), มนุษยธรรม (ความรัก, เมตตา และความฉลาดทางสังคม), ความเป็นธรรม (ความรับผิดชอบทางสังคม, ความยุติธรรม และความเป็นผู้นำ), การควบคุมอารมณ์ (การให้อภัย, ความอ่อนน้อมถ่อมตน, ความรอบคอบ และการควบคุมตนเอง) และอุต্তरภาพ (การชื่นชมในความงามของสิ่งรอบตัวและความดีของผู้คน, ความสำนึกเห็นค่าและรู้คุณ, ความหวัง, อารมณ์ขันและความปี้เล่น และศรัทธาหรือความแกร่งทางจิตวิญญาณ)

การจะพัฒนาปัญญาของนั้น ถ้าหากขาดเมตตาแล้ว ปัญญาจะพัฒนาไปได้อย่างจำกัดมาก (หากจะยังพัฒนาต่อไปได้) เพราะความรู้ในเรื่องรากฐานความเป็นจริง จะสมบูรณ์ได้อย่างไร หากขาดความเห็นใจเข้าใจชีวิตอื่น. นอกจากนั้น เช่นเดียวกับที่ ผลบื้องกลับลบ (negative feedback) จะช่วยให้ระบบทางวิศวกรรมมีเสถียรภาพที่ดี และทนทานต่อสภาพการใช้งานที่หลากหลายมากกว่า เมตตาเป็นเสมือนกับกลไกผลป้อนกลับของชีวิต. ลองจินตนาการดูว่า หากเราเป็นผู้น้อยอยู่ในประสบการณ์ ผู้คนสามารถว่ากล่าวตักเตือน ให้คำแนะนำนำกับเราได้. แต่หากเราเป็นผู้ใหญ่ใหญ่ที่สูงด้วยวัยรุ่น ด้วยคุณวุฒิ ด้วยชื่อเสียง ด้วยเงินทอง ด้วยอำนาจ โดยไม่มีเกณฑ์ใดที่บังคับให้เราต้องฟังใคร และเราก็ไม่มีความจำเป็นต้องฟังใคร จะมีอะไรที่ทำให้เราต้องฟังคนอื่น? ณ ตอนนั้น มีเพียงความเมตตาความเห็นอกเห็นใจเท่านั้น ที่จะเป็นเสมือนช่องทางที่ยังจะเปิดรับฟังอยู่เสมอ ไม่ว่าช่องทางอื่น ๆ อาจจะถูกปิดไปแล้ว ปิดไปด้วยความสูงส่งของอำนาจ เกียรติยศ ศักดิ์ศรี ชื่อเสียง เงินทอง. เราต้องการกลไกผลบื้องกลับลบ เพื่อเสถียรภาพที่ดีของสังคมและของตัวเราเอง เพื่อที่จะยังสามารถรับฟังคำตักเตือน คำแนะนำ ความเห็นต่าง ๆ ได้อยู่เสมอ. บ่อ依托 ที่เมตตาอาจช่วยให้เราสามารถสังเกตและรับรู้ถึงความรู้สึกของผู้คนได้ ก่อนที่เขาจะต้องเอ่ยปากด้วยคำ.

“Kindness in words creates confidence.

Kindness in thinking creates profoundness.

Kindness in giving creates love.”

---Lao Tzu

“ความเมตตาในคำพูด สร้างความมั่นใจ.

ความเมตตาในความคิด สร้างความลึกซึ้ง.

ความเมตตาในการให้ สร้างความรัก”

—เล่าเจ้อ

การขาดเมตตา\_n ไม่ได้มีผลเฉพาะแค่ต่อการจำกัดปัญญา, ต่อการขาดระบบบื้องกลับ และต่อการลดประสิทธิภาพในการสื่อสารเท่านั้น. เมตตาเป็นกลไกสำคัญในการลดและควบคุมอัตตา. อัตตา (ego) หรือ มโนคติของตัวตน (concept of self) เป็นแนวโน้มและพฤติกรรมการยึดติดกับสิ่งที่จิตใช้เป็นตัวแทนของตัวตน เป็นการยึดติดในตัวตน เป็นการยึดติดในความรู้สึกเป็นเจ้าของ. อาจกล่าวโดยรวมได้ว่า เราทุกคนมีอัตตาอยู่ (ยกเว้นบุคคล เช่น อวิยบุคคล ซึ่งเป็นผู้ไม่มีอัตตา) เพียงแต่ว่า โดยส่วนใหญ่แล้ว ขนาด

ของอัตตาของเราไม่ได้ให้ผู้จุนรบกวนการดำเนินชีวิตมากจนเกินไป. อาย่างไรก็ตาม คนบางคนอาจมีอัตตาที่ใหญ่มาก ๆ และอาจใหญ่มากจนเข้าข่ายของโรคหลงตัวเอง.

**โรคหลงตัวเอง** (ความผิดปกติทางบุคลิกภาพแบบหลงตัวเอง ซึ่งภาษาอังกฤษคือ Narcissistic Personality Disorder คำย่อ NPD. เนื้อหาหลัก ๆ ในส่วนนี้ เรียบเรียงจาก [41]) คือ สภาพจิต ที่ผู้ป่วยรู้สึกว่าตัวเองเป็นคนสำคัญมาก, ชอบให้คนมาสนใจและชื่นชมมาก ๆ, มีปัญหาความสัมพันธ์กับคนในครอบครัว และขาดความเห็นอกเห็นใจผู้อื่น. ภายนอก ผู้ป่วยอาจดูเป็นคนที่มีความมั่นใจในตัวเองสูงมาก แต่ภายในแล้ว ผู้ป่วยมีความนับถือตัวเองในระดับที่ประ ula มาก และหนไม่ได้กับการถูกวิพากษ์วิจารณ์.

สัญญาณและการของโรค ได้แก่ คิดว่าตัวเองสำคัญมาก (มากเกินกว่าความเป็นจริง), คิดว่าตัวเองสมควรจะถูกยกย่อง และต้องการถูกชื่นชมอยู่ตลอดเวลา, คิดว่าตัวเองต้องถูกยอมรับว่าเหนือกว่าคนอื่น ๆ โดยไม่ได้มีหลักฐานรูปธรรมรองรับ, โ้อ้อดความสำเร็จ พรสวรรค์ และความสามารถ, หมกมุ่นและฝันเพื่องกับการประสบความสำเร็จ 野心จ ความเฉลี่ยวฉลาด ความสวယ หรือคุ้ครองที่สมบูรณ์แบบ, เสื่อว่าตัวเองดีกว่าคนอื่น ๆ และควรจะได้คบหาสมาคมกับคนพิเศษในระดับเดียวกัน, ของพูดอยู่คนเดียวในวงสนทน และการดูถูกคนอื่นที่คิดว่าต่ำต้อยกว่า, เอาเปรียบคนอื่น เพื่อให้ได้สิ่งที่ตนต้องการ, ไม่สามารถหรือไม่ยอมที่จะรับรู้ถึงความต้องการหรือความรู้สึกของคนอื่น, อิจฉาคนอื่น หรือคิดว่าคนอื่น ๆ อิจชาตัวเอง, ก้าวร้าว หรือหยิ่งโส ดูไม่จริงใจ จื้มเม แลลเสสร้ง, ยืนกรานที่จะได้สิ่งที่ต้องการ เช่น รถที่ดีที่สุด ที่ทำงานที่ดีที่สุด, ไม่สามารถยอมรับการถูกวิพากษ์วิจารณ์ได้, หงุดหงิดหรือโกรธ หากไม่ได้รับการต้อนรับปฏิบัติเป็นพิเศษ, มีปัญหาการควบคุมอารมณ์, มีปัญหาการจัดการกับความเครียด, มีปัญหาการปรับตัวกับการเปลี่ยนแปลง, รู้สึกเครียดและไม่สงบอารมณ์ เวลาไม่ได้ดีใจ และแอบรู้สึกว่าไม่มั่นคง อ่อนแอก อาย อดสูญหายหน้า.

ผู้ป่วยโรคหลงตัวเอง นอกจากจะสร้างความทุกข์ให้กับคนอื่นแล้ว โรคอาจส่งผลกระทบกับตัวผู้ป่วยเอง ได้แก่ ปัญหาความสัมพันธ์ในครอบครัว, ปัญหาที่โรงเรียน หรือที่ทำงาน, ปัญหาภาวะซึมเศร้าและวิตกกังวล, ปัญหาสุขภาพทางกาย, ปัญหาการใช้ยาเสพติด หรือการดื่มสุรา และพฤติกรรมการฆ่าตัวตาย. คำแนะนำจากเมดิคอลลินิก สำหรับผู้ป่วยโรคหลงตัวเอง คือ การเข้าพบแพทย์. แต่โดยส่วนใหญ่แล้ว ผู้มีความผิดปกติทางบุคลิกภาพ รวมถึงผู้ป่วยโรคหลงตัวเอง มักไม่คิดว่าตัวเองป่วย และมักไม่ยอมเข้ารับการรักษา.

**การลดอัตตา.** ผู้ที่ป่วยแล้ว การเข้าพบแพทย์น่าจะดีที่สุด แต่สำหรับ คนที่ว่าไป ที่อาจต้องการลดหรือควบคุมอัตตา อาจทำได้ด้วยการพัฒนาเมตตาขึ้น. การพัฒนาเมตตา อาจทำโดย ฝึกให้อภัยคนอื่น ให้อภัยตัวเอง และปล่อยวางบ้าง, ฝึกยอมรับความจริง ฝึกพูดความจริง และฝึกที่จะเปิดใจกว้างยอมรับความคิดความเห็นที่หลากหลาย, ฝึกยอมรับความผิดของตัวเอง, ฝึกลดหรือละความรู้สึกที่จะควบคุมทุกสิ่งทุกอย่างลง, หัวเลาอยู่เบื้องหลัง ๆ ลง ฯ คนเดียวบ้าง, ฝึกชื่นชมความงามของสิ่งรอบตัว และมองเห็นความดีของคนอื่น ๆ, ฝึกจะลึกซึ้งบุญคุณหรือสิ่งดี ๆ ที่คนอื่น ๆ ทำให้เรา, ฝึกช่วยเหลือคนอื่นบ้าง, ฝึกทำดีกับคนแปลกหน้าบ้าง, ลองเป็นจิตอาสาบ้าง, ฝึกพูดสิ่งดี ๆ ให้กำลังใจคนอื่น, ลด ละ เลิกการวิจารณ์คนอื่นและการเบรียบเทียบคน, ฝึกที่จะไม่บ่น ไม่เสียดสี ไม่ประชดประชัน, ฝึกมองโลกในแง่ดี, ฝึกทักษะผู้คนอย่างยิ่มเย้ายวนใส่, ฝึกที่จะช่วยคนที่เดือดร้อนบ้าง หากมีโอกาส, ฝึกที่จะถ่อมตัว, ฝึกที่จะไม่พูดโอ้อวด รวมถึงลดหรือเลิกการโอ้อวด ผ่านสื่อสังคมออนไลน์, ฝึกที่จะปล่อยให้คนอื่นได้รับความสนใจ ได้รับการชื่นชม, ฝึกสามารถอย่างสม่ำเสมอ, แผ่เมตตาหรืออวยพรให้สรรพชีวิตอย่างสม่ำเสมอ, แผ่เมตตาให้กับคนที่เราไม่ชอบหรือคนที่เราโกรธ, พยายามมีสติรู้ถึงอารมณ์ที่เข้ามาในใจ, พยายามควบคุมอารมณ์ และศึกษาพัฒนาตนเองด้านจิตวิญญาณบ้าง.

อัตตา มีลักษณะที่แบลก. นั่นคือ ถ้าเรารอับคิดว่า เราดีกว่าคนอื่น นี่คืออัตตาสูง และถ้าเรารอับคิดว่า เราแย่กว่าคนอื่น นี่ก็คืออัตตาสูง. ทราบที่เรายังหมกมุ่นกับตัวเราเป็นสำคัญ นั่นคืออัตตาสูง. สิ่งที่จะลดอัตตาได้ คือเมตตา (ภาพของสรรพชีวิตมีความสุข เราอาจจะยังอยู่ในภาพ แต่ไม่ได้เด่นอีกต่อไปแล้ว).

**ไม่ได้รักษาความผิดปกติ แต่ดูแลส่วนที่ปกติ.** สำหรับการรักษาผู้ป่วยอาการจิตเวช มีเรื่องเล่าที่น่าสนใจจากอาจารย์พระ (Ajahn Brahm) ซึ่งเป็นพระนักเทศน์ นักบรรยาย และนักเขียนที่ได้รับการยอมรับนับถืออย่างกว้างขวาง ที่ทำงานเคย์ตามเจ้าหน้าที่ในโรงพยาบาลจิตเวชแห่งหนึ่งว่า เขารักษาความผิดปกติทางจิตอย่างไร เจ้าหน้าที่ตอบว่า เขายังไม่ได้รักษาส่วนที่ผิดปกติ ท่ารักษาส่วนที่ดี.

ผู้ป่วยจิตเวช ไม่ได้แสดงอาการผิดปกติออกมานานอดเวลา. ผู้ป่วยหลายคน ส่วนใหญ่ก็ปกติ เพียงแค่มีช่วงเวลาที่เกิดอาการผิดปกติทางจิตขึ้นมาเท่านั้น. สิ่งที่เจ้าน้าจิตเวชทำ คือ พยายามรักษา ส่งเสริม ดูแล ให้ช่วงเวลาที่ดีอยู่ดี闫านานขึ้น ดูแลให้ส่วนที่ปกติเติบโตขึ้น แล้วช่วงเวลาที่ผู้ป่วยเป็นปกติ จะยาวนานขึ้น และทำให้ช่วงเวลาผิดปกติสั้นลงไปเอง. ความปกติก็ดูแล ถูกให้ความสำคัญ จนมันอยู่ได้นานขึ้น แข็งแรงมากขึ้น ส่วนความผิดปกติจะเกิดน้อยลงและเบาลงเอง. แนวทางนี้ไม่ใช่ใช้ได้เฉพาะกับผู้ป่วยจิตเวชหรือ ในตัวคนเรา ในชุมชน หรือในสังคมก็เท่านั้น ที่มีทั้งส่วนที่ดี และส่วนที่ไม่ดี ถ้าเรา\_rักษา ดูแล ส่งเสริมให้ส่วนที่ดีเติบโตขึ้นแข็งแรงขึ้น ส่วนที่ไม่ดีมันจะน้อยลง เบาลงเอง.

“When life is good do not take it for granted as it will pass. Be mindful, be compassionate and nurture the circumstances that find you in this good time so it will last longer. When life falls apart always remember that this too will pass. Life will have its unexpected turns.”

—Ajahn Brahm

“ตอนที่ชีวิตดี ใจกับมัน เพราะมันจะผ่านไป. มีสติรับรู้ มีเมตตา และทะนุถนอมสิ่งต่าง ๆ ที่ช่วยให้เราได้มีช่วงเวลาที่ดี เพื่อให้เวลาดี ๆ มีได้นานขึ้น. ตอนที่ชีวิตแตกเป็นเสียง ๆ จำไว้เสมอว่า เวลาันั้นมันก็จะผ่านไปเหมือนกัน. ชีวิตจะมีการเปลี่ยนแปลงที่คาดไม่ถึงเสมอ.”

—อาจารย์พรหม

ข้อดีข้อเสียของโครงข่ายประสาทเวียนกลับ. โครงข่ายประสาทเวียนกลับ สามารถประมวลผลชุดข้อมูลลำดับได้โดยไม่จำกัดความยาวของลำดับ โดยที่ความซับซ้อนของแบบจำลอง ไม่ขึ้นกับความยาวของลำดับ (ดูแบบฝึกหัด 9.1 และ 9.2 ประกอบ) และที่สำคัญ คือ การใช้ค่าน้ำหนักร่วม สำหรับทุก ๆ ลำดับเวลา. อย่างไรก็ตาม ข้อเสียของโครงข่ายประสาทเวียนกลับ คือ การคำนวณใช้เวลามาก (การประมวลผลแบบขนาดทำได้ลำบาก) และ โครงข่ายประสาทเวียนกลับ ยังถูกรายงานบ่อย ๆ ว่าจำลองความสัมพันธ์ระยะยาวระหว่างจุดข้อมูลได้ไม่ดี และไม่สามารถจำลองความสัมพันธ์กับจุดข้อมูลลำดับข้างหน้า หรือลำดับเวลาในอนาคต (ดูหัวข้อ 9.3 ประกอบ).

นอกจากการฝึกโครงข่ายประสาทเวียนกลับที่ใช้เวลา多くแล้ว การฝึกโครงข่ายประสาทเวียนกลับ ยังมีปัญหาการเลื่อนหายของเกรเดียนต์ และปัญหาการระเบิดของเกรเดียนต์. การเวียนกลับย้อนลำดับเวลาให้ผลคล้ายการแพร่กระจายย้อนกลับผ่านชั้นคำนวณต่าง ๆ ของโครงข่ายประสาทเชิงลึก (ดูแผนภาพคลื่นลำดับ เช่น รูป 9.6 ประกอบ) แต่จุดต่างที่สำคัญคือ เมื่อย้อนกลับผ่านชั้นคำนวณ ค่าน้ำหนักรของชั้นคำนวณแต่ละชั้น เป็นอิสระต่อกัน แต่เมื่อย้อนกลับผ่านลำดับเวลา ค่าน้ำหนักรที่ลำดับเวลาต่าง ๆ เป็นชุดเดียวกัน. กลไกการเวียนกลับ ส่งผลต่อเสถียรภาพของการคำนวณค่าเกรเดียนต์ ซึ่งบางครั้งเกิดปัญหาในลักษณะการเลื่อนหายของเกรเดียนต์ ที่เกรเดียนต์มีค่าลดลงอย่างมาก เมื่อเวียนกลับย้อนลำดับเวลา จนไม่สามารถเขื่อมโยงความสัมพันธ์ระยะยาวได้. แต่บางครั้ง การฝึกโครงข่ายประสาทเวียนกลับ อาจเห็นปัญหาในลักษณะของการระเบิดของเกรเดียนต์. **ปัญหาระเบิดของเกรเดียนต์** (exploding gradient problem) ที่พบกับการฝึกโครงข่ายประสาทเวียนกลับ คือ การที่เกรเดียนต์มีค่าเพิ่มขึ้นอย่างมาก เมื่อเวียนกลับย้อนลำดับเวลา จนทำให้การ

คำนวณเสียงสัญญาณ และการฝึกล้มเหลวในที่สุด.

ปัญหาการเลือนหายของเกรเดียนต์ ในโครงข่ายประสาทเวียนกลับ สามารถบรรเทาลงได้ด้วยกลไกต่าง ๆ เช่นที่เป็นส่วนประกอบของแบบจำลองความจำระยะสั้นที่ยาว (หัวข้อ 9.4). ส่วนปัญหาการระเบิดของเกรเดียนต์ สามารถบรรเทาลงได้ง่าย ๆ ด้วยการเลิมเกรเดียนต์.

การเลิมเกรเดียนต์ (gradient clipping) เป็นกลไกง่ายในการลดขนาดเกรเดียนต์ลง ให้อยู่ในระดับที่การคำนวณจะยังสามารถทำต่อไปได้โดยมีเสถียรภาพ. พาสคานูและคณะ[151] ปรับขนาดของเกรเดียนต์ลงให้ไม่เกินค่าที่กำหนด โดย

$$\text{ถ้า } \|\mathbf{g}\| > \tau \text{ และ}$$

$$\mathbf{g} \leftarrow \frac{\mathbf{g} \cdot \tau}{\|\mathbf{g}\|} \quad (9.16)$$

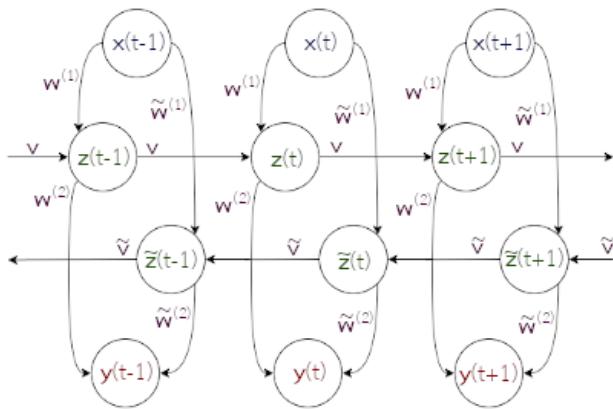
เมื่อ  $\mathbf{g}$  คือ เกรเดียนต์ นั่นคือ  $\mathbf{g} \equiv \nabla_{\theta} E$  และ  $\|\mathbf{g}\|$  คือ ขนาดของเกรเดียนต์. ส่วนสเกลาร์  $\tau$  คือ ค่าขีดแบ่งที่กำหนด. ค่าขีดแบ่ง  $\tau$  สามารถเลือกได้ง่าย ๆ เพียงเป็นค่าที่ไม่มากเกินไปที่จะทำให้ระบบเสียเสถียรภาพเท่านั้น เช่น อาจเลือกให้  $\tau = 1$  เมื่อันที่พาสคานูและคณะใช้ในการทดลองก็ได้.

พาสคานูและคณะ ใช้วิธีปรับลงขนาดของเกรเดียนต์ทั้งเวคเตอร์ ทำให้แม่ลอดขนาดของเวคเตอร์ลง แต่ทิศทางของเกรเดียนต์ยังคงเดิม.

### 9.3 โครงข่ายประสาทเวียนกลับสองทาง

โครงข่ายประสาทเวียนกลับ นำจุดข้อมูลลำดับก่อนหน้ามาร่วมพิจารณาผลการทำนายที่ลำดับเวลาปัจจุบันช่วยให้เราสามารถสร้างแบบจำลองความสัมพันธ์ของจุดข้อมูลลำดับปัจจุบัน กับจุดข้อมูลต่าง ๆ ในลำดับก่อนหน้าได้. อย่างไรก็ตาม ภารกิจกับข้อมูลเชิงลำดับหลายอย่าง อาจต้องการจำลองความสัมพันธ์ระหว่างจุดข้อมูลลำดับปัจจุบันกับจุดข้อมูลในลำดับหลัง ๆ เช่น กรณีภารกิจการระบุหมวดคำ ในรูป 9.2 การระบุหมวดคำของトイเค็น one ที่ถูกต้อง ต้องการรู้トイเค็นต่าง ๆ ที่ตามมาในภายหลัง นั่นคือ สำหรับ "... if one has the courage to admit them." คำว่า "one" ทำหน้าที่เป็นสรรพนาม แต่ถ้าสำหรับ "... if one day you can let it go." คำว่า "one" ทำหน้าที่เป็นตัวเลข. โครงข่ายประสาทเวียนกลับ ที่อาศัยเฉพาะแต่ความสัมพันธ์กับลำดับที่ผ่านมา ไม่อาจแก้ปัญหาลักษณะนี้ได้.

วิธีบรรเทาปัญหาลักษณะนี้อย่างง่าย ๆ ก็คือ การใช้กลไกหน้าต่างเวลา (time-window) ที่จับกลุ่มトイเค็นหลาย ๆ トイเค็นรวมกันเป็นจุดข้อมูลแต่ละจุด สำหรับโครงข่ายประสาทเวียนกลับ. อย่างไรก็ตาม แนวทาง



รูปที่ 9.7: แผนภาพคลื่นลำดับของโครงข่ายภาษาที่เวียนกลับสองทาง.

การใช้กลไกหน้าต่างเวลาใน อาศัยกรอบหน้าต่างเวลา ที่มีความยาวคงที่ ทำให้จำกัดความสัมพันธ์ระยะยาวระหว่างจุดข้อมูล. อีกแนวทางง่าย ๆ ก็คือ การหน่วงเวลาของระหว่างจุดข้อมูลลำดับของอินพุต กับจุดข้อมูลลำดับของเอาต์พุต แต่แนวทางนี้ ก็ยังต้องอาศัยการเลือกระยะเวลาหน่วงที่เหมาะสม.

แนวทางหนึ่ง ที่ถูกออกแบบและพบร่วมกับ[78] ใช้ได้ดี สำหรับกรณีเช่นนี้ คือ โครงข่ายภาษาที่เวียนกลับสองทาง (bidirectional recurrent neural networks[180]). กลไกที่สำคัญของโครงข่ายภาษาที่เวียนกลับสองทาง คือ เพิ่มชั้นคำนวนเวียนกลับที่รับชุดลำดับที่เรียงกลับหลัง. การคำนวนเอาต์พุตสุดท้ายของโครงข่าย จะรojognกว่าชั้นคำนวนเวียนกลับ (ทั้งชั้นที่รับชุดลำดับเรียงหน้าไปหลัง และชั้นที่รับชุดลำดับเรียงหลังไปหน้า) จะได้ประมวลผลครบถ้วนจุดข้อมูลในชุดลำดับก่อน.

รูป 9.7 แสดงแผนภาพคลื่นลำดับของโครงข่ายภาษาที่เวียนกลับสองทาง. การคำนวนของหน่วยอยู่ในชั้นคำนวนเวียนกลับทิศทางย้อนกลับที่  $q^{th}$  อาจเขียนได้ดังนี้

$$\tilde{a}_j^{(q)}(t) = \sum_{i=1}^{\tilde{D}} \tilde{w}_{ji}^{(q)} \cdot \tilde{z}_i^{(q-1)}(t) + \sum_{m=1}^{\tilde{M}} \tilde{v}_{jm}^{(q)} \cdot \tilde{z}_m^{(q)}(t+1) + \tilde{b}_j^{(q)} \quad (9.17)$$

$$\tilde{z}_j^{(q)}(t) = h(\tilde{a}_j^{(q)}(t)) \quad (9.18)$$

เมื่อ  $\tilde{a}_j^{(q)}(t)$  และ  $\tilde{z}_j^{(q)}(t)$  คือ ค่าตัวกระตุ้นและผลการกระตุ้น ของหน่วยอยู่ที่  $j^{th}$  ในชั้นคำนวน  $q^{th}$  สำหรับจุดข้อมูลลำดับที่  $t^{th}$  โดย  $\tilde{D}$  คือจำนวนหน่วยอยู่ในชั้น  $(q-1)^{th}$  และ  $\tilde{M}$  คือจำนวนหน่วยอยู่ในชั้น  $q^{th}$ . ตัวแปร  $\tilde{w}_{jd}^{(q)}$  เป็นค่าน้ำหนักของการเชื่อมต่อระหว่างหน่วยที่  $d^{th}$  ของชั้น  $(q-1)^{th}$  กับหน่วยที่  $j^{th}$  ของชั้นคำนวน  $q^{th}$ . ตัวแปร  $\tilde{v}_{jm}^{(q)}$  เป็นค่าน้ำหนักของการเชื่อมต่อของหน่วยที่  $m^{th}$  เวียนกลับมาเข้าหน่วยที่  $j^{th}$  ของชั้นคำนวนเดียวกัน,  $\tilde{b}_j^{(q)}$  คือค่าใบอัศของหน่วยที่  $j^{th}$  และ  $h(\cdot)$  คือฟังก์ชันกระตุ้น.

สังเกตสมการ 9.17 เปรียบเทียบกับสมการ 9.1 ซึ่งเป็นการคำนวณค่าตัวกระตุ้น ในชั้นคำนวณเวียนกลับทิศทางปกติ (ทิศทางไปข้างหน้า) จะเห็นว่า จุดสำคัญคือ การที่ค่าตัวกระตุ้น ในชั้นคำนวณเวียนกลับ ทิศทางย้อนกลับ ได้รับอิทธิพลจาก ผลการกระตุ้นของลำดับเวลาอนาคต  $\tilde{z}_m^{(q)}(t+1)$ .

การคำนวณของหน่วยอยู่ในชั้นรวมผลของทั้งสองทิศทาง (เช่น ชั้นเอาร์พุต ที่คำนวณค่า  $\mathbf{y}$  ในรูป 9.7) ก็สามารถดำเนินการได้ เช่นเดียวกับการคำนวณหน่วยอยู่ในชั้นคำนวณเชื่อมต่อเติมที่ทั่ว ๆ ไป นั่นคือ

$$\hat{a}_j^{(q)}(t) = \sum_{i=1}^D w_{ji}^{(q)} \cdot z_i^{(q-1)}(t) + \sum_{i=1}^{\tilde{D}} \tilde{w}_{ji}^{(q)} \cdot \tilde{z}_i^{(q-1)}(t) + \hat{b}_j^{(q)} \quad (9.19)$$

$$\hat{z}_j^{(q)}(t) = h(\hat{a}_j^{(q)}(t)) \quad (9.20)$$

เมื่อ  $\hat{a}_j^{(q)}(t)$  และ  $\hat{z}_j^{(q)}(t)$  คือ ค่าตัวกระตุ้นและผลการกระตุ้น ของชั้นที่รวมผลจากการคำนวณเวียนกลับทั้งสองทิศทาง โดย  $z_i^{(q-1)}(t)$  และ  $\tilde{z}_i^{(q-1)}(t)$  คือผลการกระตุ้น จากชั้นเวียนกลับทิศทางไปข้างหน้า และทิศทางย้อนกลับ ตามลำดับ. ส่วน  $w_{ji}^{(q)}$ ,  $\tilde{w}_{ji}^{(q)}$  และ  $\hat{b}_j^{(q)}$  คือพารามิเตอร์ของชั้นคำนวณ.

เพื่อให้การคำนวณค่าเอาร์พุตของโครงข่ายประสาทเวียนกลับสองทาง เป็นไปโดยเรียบร้อย การคำนวณ (การแพร่กระจายไปข้างหน้า) ดำเนินการตามลำดับดังนี้

- คำนวณค่าผลการกระตุ้น จากชั้นเวียนกลับทิศทางไปข้างหน้า โดยคำนวณตามลำดับเวลาจาก  $t = 1$  ไป  $t = T$ . นั่นคือ คำนวณค่า  $z_i^{(q-1)}(t)$  สำหรับ  $t = 1, \dots, T$  ตามลำดับ.
- คำนวณค่าผลการกระตุ้น จากชั้นเวียนกลับทิศทางย้อนกลับ โดยคำนวณตามลำดับเวลาจาก  $t = T$  ไป  $t = 1$ . นั่นคือ คำนวณค่า  $\tilde{z}_i^{(q-1)}(t)$  สำหรับ  $t = T, \dots, 1$  ตามลำดับ.
- คำนวณชั้นที่รวมผลจากสองทิศทาง. นั่นคือ คำนวณค่า  $\hat{z}_j^{(q)}(t)$  สำหรับทุก ๆ ค่าของ  $t$  (ลำดับใดก็ได้).

การฝึกโครงข่ายประสาทเวียนกลับสองทาง ก็สามารถทำได้ในลักษณะเดียวกับการฝึกโครงข่ายประสาทเวียนกลับ เพียงมีความซับซ้อนเพิ่มขึ้น เนื่องจาก (1) การเวียนกลับมีสองทิศทาง และ (2) การเวียนกลับทั้งสองทิศทาง เปรียบเสมือนส่วนประกอบในชั้นคำนวณเดียวกัน เพราะรับอินพุตจากชั้นเดียวกัน และให้อเอาร์พุตออกไปที่ชั้นเดียวกัน.

การคำนวณในการฝึกชั้นเวียนกลับสองทาง (ชั้น  $q^{th}$ ) สรุปได้ดังนี้

- (1) คำนวณการแพร่กระจายไปข้างหน้า

- (1.0) คำนวณชั้น  $(q - 1)^{th}$   
ได้  $\hat{z}_i^{(q-1)}(t)$  สำหรับทุก ๆ  $i$  และทุก ๆ  $t$ .  
ถ้าชั้น  $(q - 1)^{th}$  เป็นชั้นอินพุต  $\hat{z}_i^{(q-1)}(t) = x_i(t)$ .
  - (1.1) คำนวณการเวียนกลับทิศทางไปข้างหน้า ( $t = 1, \dots, T$  ตามลำดับ)  
ได้  $a_j^{(q)}(t)$  และ  $z_j^{(q)}(t)$  สำหรับทุก ๆ  $j$  (สมการ 9.1 และ 9.2)
  - (1.2) คำนวณการเวียนกลับทิศทางกลับหลัง ( $t = T, \dots, 1$  ตามลำดับ)  
ได้  $\tilde{a}_j^{(q)}(t)$  และ  $\tilde{z}_j^{(q)}(t)$  สำหรับทุก ๆ  $j$  (สมการ 9.17 และ 9.18)
  - (1.3) คำนวณชั้น  $(q + 1)^{th}$   
ได้  $\hat{a}_k^{(q+1)}(t)$  และ  $\hat{z}_k^{(q+1)}(t)$  สำหรับทุก ๆ  $k$  และทุก ๆ  $t$ .  
(สมการ 9.19 และ 9.20 ถ้าชั้น  $(q + 1)^{th}$  เป็นชั้นเชื่อมต่อเติมที่)
  - คำนวณชั้นต่อ ๆ ไป จนได้เอาร์พุตสุดท้าย
- (2) คำนวณการแพร่กระจายย้อนกลับ
    - (2.0) คำนวณการแพร่กระจายย้อนกลับจนถึงชั้น  $(q + 1)^{th}$   
ได้  $\hat{\delta}_k^{(q+1)}(t) \equiv \frac{\partial E}{\partial \hat{a}_k^{(q+1)}(t)}$  สำหรับทุก ๆ  $k$  และทุก ๆ  $t$ .
    - (2.1) คำนวณการแพร่กระจายย้อนกลับสำหรับทิศทางไปข้างหน้า (แต่การคำนวณต้องทำจาก  $t = T$  ไป  $t = 1$ )  
ได้  $\delta_j^{(q)}(t), \frac{\partial E}{\partial w_{ji}^{(q)}}, \frac{\partial E}{\partial v_{jm}^{(q)}} \text{ และ } \frac{\partial E}{\partial b_j^{(q)}}$  สำหรับทุก ๆ  $i, j$  และ  $m$  (สมการ 9.7, 9.9, 9.11 และ 9.12)
    - (2.2) คำนวณการแพร่กระจายย้อนกลับสำหรับทิศทางกลับหลัง (การคำนวณต้องทำจาก  $t = 1$  ไป  $t = T$ )

$$\begin{aligned}\tilde{\delta}_j^{(q)}(t) &\equiv \frac{\partial E}{\partial \tilde{a}_j^{(q)}(t)} \\ &= h'(\tilde{a}_j^{(q)}(t)) \cdot \left( \sum_k \hat{\delta}_k^{(q+1)}(t) \cdot \tilde{w}_{kj}^{(q+1)} + \sum_m \tilde{\delta}_m^{(q)}(t-1) \cdot \tilde{v}_{mj}^{(q)} \right)\end{aligned}\quad (9.21)$$

$$\frac{\partial E}{\partial \tilde{w}_{ji}^{(q)}} = \sum_t \tilde{\delta}_j^{(q)}(t) \cdot \hat{z}_i^{(q-1)}(t) \quad (9.22)$$

$$\frac{\partial E}{\partial \tilde{v}_{jm}^{(q)}} = \sum_t \tilde{\delta}_j^{(q)}(t) \cdot \tilde{z}_m^{(q)}(t-1) \quad (9.23)$$

$$\frac{\partial E}{\partial \tilde{b}_j^{(q)}} = \sum_t \tilde{\delta}_j^{(q)}(t) \quad (9.24)$$

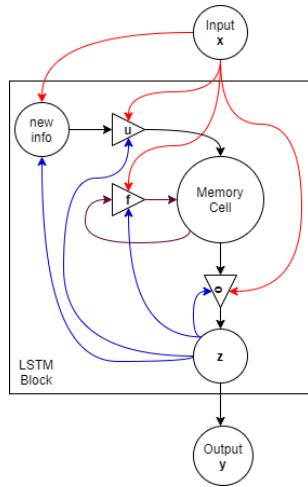
สำหรับทุก ๆ  $i, j$  และ  $m$

- แพร่กระจายย้อนกลับต่อไปจนครบทุกชั้น  
ถ้าชั้น  $(q-1)^{th}$  เป็นชั้นเชื่อมต่อเติมที่ แล้ว

$$\begin{aligned}\hat{\delta}_i^{(q-1)}(t) &\equiv \frac{\partial E}{\partial \hat{a}_i^{(q-1)}(t)} \\ &= h'(\hat{a}_i^{(q-1)}(t)) \cdot \left( \sum_j \delta_j^{(q)}(t) \cdot w_{ji}^{(q)} + \sum_j \tilde{\delta}_j^{(q)}(t) \cdot \tilde{w}_{ji}^{(q)} \right) \\ &\quad .\end{aligned}\quad (9.25)$$

- (3) ปรับค่าน้ำหนักตามเกรเดียนต์ที่คำนวณได้

โครงข่ายประสาทเวียนกลับสองทาง มีความสามารถในการเชื่อมความสัมพันธ์ ทั้งความสัมพันธ์กับลำดับในอดีต และลำดับในอนาคต. อย่างไรก็ตาม การใช้งานโครงข่ายประสาทเวียนกลับสองทาง เหมาะกับ (1) ลักษณะชุดข้อมูลที่ต้องการจำลองความสัมพันธ์กับลำดับในอนาคต และ (2) ภารกิจที่ต้องการเอาต์พุต หลังจากได้เห็นชุดข้อมูลครบทุกลำดับแล้ว. หากลักษณะชุดข้อมูลไม่ได้ต้องการความสัมพันธ์กับลำดับในอนาคต การคำนวณที่เพิ่มขึ้นของโครงข่ายประสาทเวียนกลับสองทาง จะกลายเป็นภาระที่ไม่จำเป็น. หากภารกิจต้องการเอาต์พุตก่อนที่แบบจำลองจะได้เห็นข้อมูลครบลำดับ โครงข่ายประสาทเวียนกลับสองทาง (ในรูปแบบดังเดิมนี้) จะไม่เหมาะสมกับภารกิจนั้น และอาจพิจารณาทางเลือกอื่น เช่น กลไกหน้าต่างเวลา หรือ การหน่วงเวลาระหว่างอินพุตและเอาต์พุต.



รูปที่ 9.8: แผนภาพโครงสร้างบล็อกความจำของแบบจำลองความจำระยะสั้นที่ยาว. วงกลม แทนหน่วยคำนวณ ดังเช่นแผนภาพโครงข่ายประสาทเทียมอื่น ๆ. สามเหลี่ยม แทนกลไกของประตุคุบคุム ที่ควบคุมการให้ผลของสารสนเทศ โดยการปิดเปิดประตู ขึ้น กับสัญญาณจากอินพุตเวลาปัจจุบัน และผลลัพธ์ที่ผ่านมา. ดูแผนภาพคลื่นลำดับ (รูป 9.9) ประกอบ.

## 9.4 แบบจำลองความจำระยะสั้นที่ยาว

ดังข้อดีข้อเสียของโครงข่ายประสาทเวียนกลับ ที่ได้อภิปรายไปว่า โครงข่ายประสาทเวียนกลับ มีปัญหาการเลือนหายของเกรเดียนต์ ที่ทำให้โครงข่ายประสาทเวียนกลับ ไม่สามารถเข้ามายองความสัมพันธ์ระยะยาว (ความสัมพันธ์ระหว่างจุดข้อมูลที่ลำดับต่างกันมาก).

ปัญหาเรื่องนี้ นำไปสู่การพัฒนาวิธีแก้ต่าง ๆ มากมาย และหนึ่งในนั้น คือ แบบจำลองความจำระยะสั้นที่ยาว ที่ปัจจุบัน ได้รับการยอมรับอย่างกว้างขวาง.

**แบบจำลองความจำระยะสั้นที่ยาว** (long short-term memory model [88, 89, 71, 72, 79] ที่มักย่อ LSTM) คือ โครงข่ายประสาทเวียนกลับ ที่เพิ่มกลไกควบคุมความจำค่าใหม่ ควบคุมการลืมค่าเก่า และควบคุมการระลึกความจำไปใช้ อย่างชัดเจน. รูป 9.8 แสดงโครงสร้างของแบบจำลองความจำระยะสั้นที่ยาว. แบบจำลองความจำระยะสั้นที่ยาว มีกลไกภายในหน่วยคำนวณย่อยที่ซับซ้อนกว่า หน่วยย่อยของโครงข่ายประสาทเทียมทั่วไปอยู่มาก ดังนั้น เพื่อกันการสับสน หน่วยคำนวณย่อยที่ครอบคลุมแนวคิดแบบจำลองความจำระยะสั้นที่ยาว จะเรียกว่า บล็อกความจำ (LSTM block). ภายในบล็อกความจำ มีหน่วยความจำ เรียก เชลล์ (cell) หรือเซลล์ความจำ (memory cell). กลไกการเก็บความจำของเชลล์นี้ คือ จุดเด่นของแบบจำลองความจำระยะสั้นที่ยาว ซึ่งบรรยายดังสมการ 9.26 ถึง 9.31.

เมื่อ ชุดลำดับอินพุต คือ  $[\mathbf{x}(1), \dots, \mathbf{x}(T)]$  การคำนวณของบล็อกความจำ ดำเนินการดังนี้

$$\mathbf{f}(t) = \sigma(\mathbf{W}_f \cdot [\mathbf{z}(t-1); \mathbf{x}(t)] + \mathbf{b}_f) \quad (9.26)$$

$$\mathbf{u}(t) = \sigma(\mathbf{W}_u \cdot [\mathbf{z}(t-1); \mathbf{x}(t)] + \mathbf{b}_u) \quad (9.27)$$

$$\mathbf{o}(t) = \sigma(\mathbf{W}_o \cdot [\mathbf{z}(t-1); \mathbf{x}(t)] + \mathbf{b}_o) \quad (9.28)$$

$$\tilde{\mathbf{c}}(t) = \tanh(\mathbf{W}_c \cdot [\mathbf{z}(t-1); \mathbf{x}(t)] + \mathbf{b}_c) \quad (9.29)$$

$$\mathbf{c}(t) = \mathbf{f}(t) \odot \mathbf{c}(t-1) + \mathbf{u}(t) \odot \tilde{\mathbf{c}}(t) \quad (9.30)$$

$$\mathbf{z}(t) = \mathbf{o}_t \odot \tanh(\mathbf{c}(t)) \quad (9.31)$$

เมื่อ ตัวดำเนินการ  $\odot$  หมายถึง การคูณแบบตัวต่อตัว (element-wise product) และตัวดำเนินการ  $[\cdot; \cdot]$  หมายถึงการนำค่าเวกเตอร์ต่อกัน นั่นคือ ถ้า  $\mathbf{v}^{(1)} = [v_1^{(1)}, \dots, v_M^{(1)}]^T$  และ  $\mathbf{v}^{(2)} = [v_1^{(2)}, \dots, v_N^{(2)}]^T$  แล้ว  $[\mathbf{v}^{(1)}; \mathbf{v}^{(2)}] = [v_1^{(1)}, \dots, v_M^{(1)}, v_1^{(2)}, \dots, v_N^{(2)}]^T$ . ตัวแปร  $\mathbf{z}(t)$  เป็นผลลัพธ์ของบล็อก (สำหรับลำดับเวลา  $t$ ).

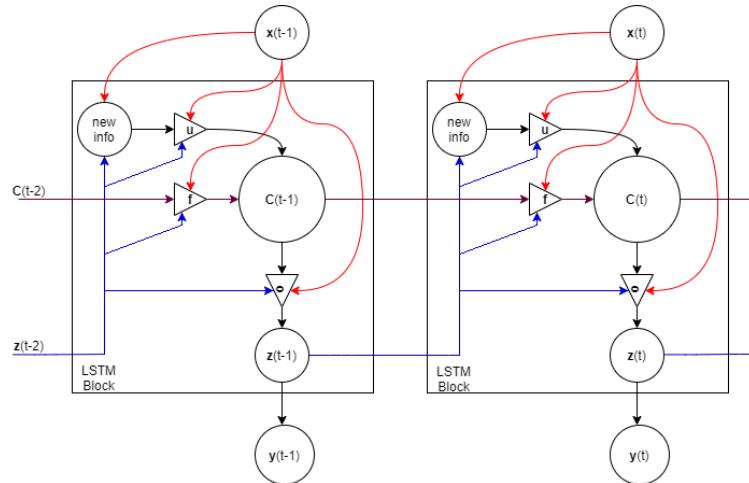
ตัวแปร  $\mathbf{c}(t)$  ทำหน้าที่เป็นค่าความจำของเซลล์ ที่ลำดับเวลา  $t$ . ส่วนตัวแปร  $\tilde{\mathbf{c}}(t)$  เป็นสารสนเทศใหม่ ที่ลำดับเวลา  $t$ . ตัวแปร  $\mathbf{f}(t)$ ,  $\mathbf{u}(t)$  และ  $\mathbf{o}(t)$  ทำหน้าที่เป็นเสมือนประตุควบคุมการให้ผลของสารสนเทศ. ประตุลืม (forget gate) และประตุรับค่าใหม่ (update gate) ซึ่งคือ  $\mathbf{f}(t)$  และ  $\mathbf{u}(t)$  ตามลำดับ ควบคุมว่า จะให้เซลล์รับจำสารสนเทศใหม่  $\tilde{\mathbf{c}}(t)$  หรือคงความจำเดิม  $\mathbf{c}(t-1)$ . ประตุผลลัพธ์ (output gate)  $\mathbf{o}(t)$  ควบคุมว่า ควรจะระลึกความจำอกรมาหรือไม่. ค่าอินพุต  $\mathbf{x}(t)$  และค่าผลลัพธ์ที่ผ่านมา  $\mathbf{z}(t-1)$  ควบคุมการทำงานของประตุต่าง ๆ. รูป 9.9 แสดงแผนภาพคลื่nl ลำดับของบล็อกความจำ.

นอกจากนั้น เพื่อปรับการควบคุมประตุได้แม่นยำยิ่งขึ้น โครงสร้างของบล็อกความจำ อาจมีกลไกช่อง แอบมอง (peephole connections[72]) เพิ่มเข้ามาด้วย. กลไกของช่องแอบมอง จะเพิ่มค่าของความจำที่ผ่านมา เข้ามาเป็นส่วนในการควบคุมประตุต่าง ๆ ด้วย ดังสมการ 9.32 ถึง 9.34.

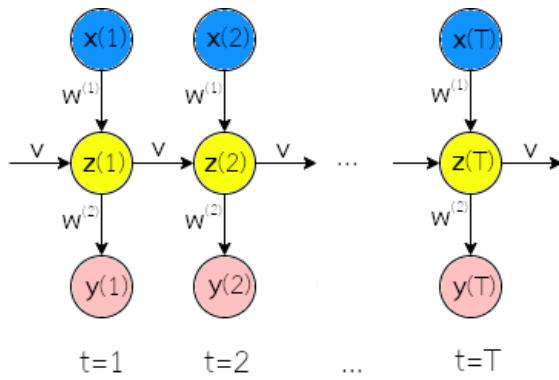
$$\mathbf{f}(t) = \sigma(\mathbf{W}_f \cdot [\mathbf{z}(t-1); \mathbf{x}(t); \mathbf{c}(t-1)] + \mathbf{b}_f) \quad (9.32)$$

$$\mathbf{u}(t) = \sigma(\mathbf{W}_u \cdot [\mathbf{z}(t-1); \mathbf{x}(t); \mathbf{c}(t-1)] + \mathbf{b}_u) \quad (9.33)$$

$$\mathbf{o}(t) = \sigma(\mathbf{W}_o \cdot [\mathbf{z}(t-1); \mathbf{x}(t); \mathbf{c}(t-1)] + \mathbf{b}_o). \quad (9.34)$$



รูปที่ 9.9: แผนภาพคลื่นลำดับของแบบจำลองความจำระยะสั้นที่ใช้ สำหรับสองลำดับเวลา.



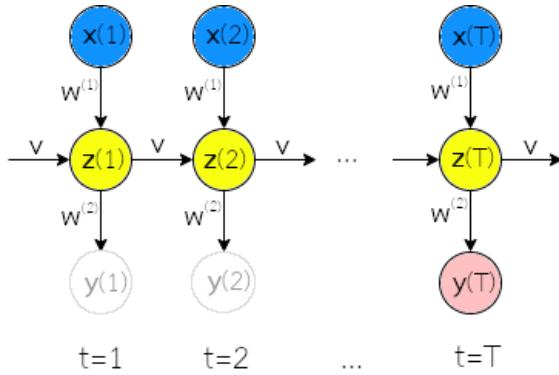
รูปที่ 9.10: แผนภาพคลื่นลำดับของโครงข่ายประสานเวียนกลับ สำหรับกรณีที่ทั้งอินพุตและเอาต์พุตเป็นชุดลำดับ และมีจำนวนลำดับเท่ากัน.

## 9.5 การใช้งานโครงข่ายประสานเวียนกลับ

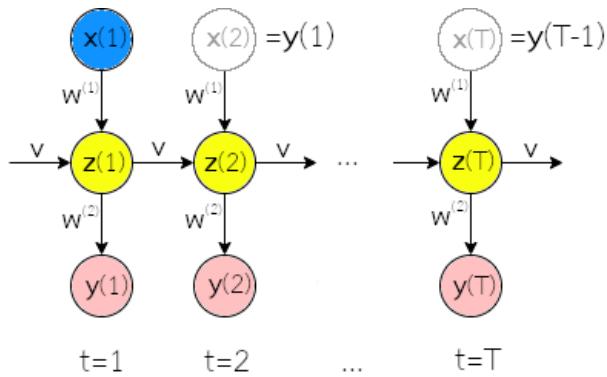
โครงข่ายประสานเวียนกลับ<sup>3</sup> สามารถใช้เป็นแบบจำลองในการรู้จำรูปแบบเชิงลำดับแบบต่าง ๆ ดังอภิปรายในหัวข้อ 8.1. ตัวอย่างเช่น กรณีที่ทั้งอินพุตและเอาต์พุตเป็นชุดลำดับ และมีจำนวนลำดับเท่ากัน เช่น การระบุหมวดคำ (ดูแบบฝึกหัด 9.7) โครงข่ายประสานเวียนกลับ สามารถนำมาใช้ในกรณีได้อย่างตรงไปตรงมา. กระบวนการฝึก อาจกำหนดให้  $\mathcal{M}(t) = 1$  สำหรับทุก ๆ ลำดับเวลา  $t$ . แผนภาพคลื่นลำดับ สำหรับกรณีนี้แสดงในรูป 9.10.

กรณีการจำแนกลำดับ เช่น การจำแนกอารมณ์ (ดูแบบฝึกหัด 9.6) ที่อินพุตเป็นข้อมูลชุดลำดับ แต่เอาต์พุตไม่ได้เป็นชุดลำดับ นั่นคือ อินพุต  $\mathbf{X} = \{\mathbf{x}(1), \dots, \mathbf{x}(T)\}$  และเอาต์พุต  $\mathbf{y}$ . กรณีนี้อาจดำเนินการโดย

<sup>3</sup> ณ ที่นี่ โครงข่ายประสานเวียนกลับ จะหมายรวมถึง โครงข่ายประสานเวียนกลับทุก ๆ ชนิด ซึ่งรวมถึง แบบจำลองความจำระยะสั้นที่ใช้ และหน่วยเรียนกลับมีประตุ ด้วย.



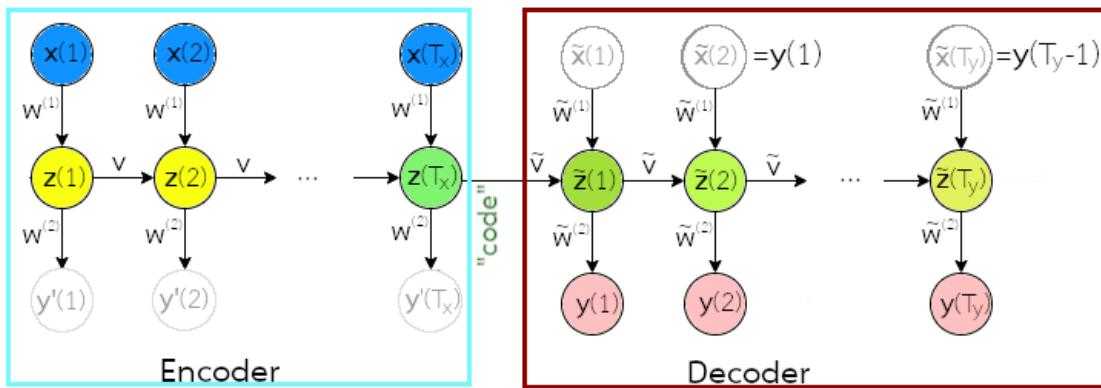
รูปที่ 9.11: แผนภาพคลีลำดับ กรณีการจำแนกลำดับ. จุดสำคัญอยู่ที่ การเลือกเฉพาะเอาต์พุตที่ต้องการไปใช้ (ค่าที่ทำงาน  $\hat{\mathbf{y}} = \mathbf{y}(T)$ ). แม้ค่าแบบจำลองจะยังคงให้เอาต์พุตอื่น ๆ  $\mathbf{y}(t \neq T)$  ออกมามาด้วย เพียงแต่เอาต์พุตเหล่านี้ไม่ได้ถูกนำไปใช้ทำอะไร (แผนภาพใช้สีแดง เพื่อสื่อถึงการปล่อยค่าเอาต์พุตเหล่านี้ทิ้ง).



รูปที่ 9.12: แผนภาพคลีลำดับของโครงข่ายประสาทเวียนกลับ ที่นิยมให้สำหรับกรณีที่เอาต์พุตเป็นชุดลำดับ แต่อินพุตไม่ใช่. อินพุตจริงของระบบ  $\tilde{x}$  จะถูกนำเข้าเป็นจุดข้อมูลที่ลำดับเวลาแรก นั่นคือ  $\mathbf{x}(1) = \tilde{x}$  และหลังจากลำดับเวลาแรก เอาต์พุตที่คำนวณได้ จะถูกนำไปเป็นอินพุตสำหรับลำดับเวลาถัดไป. ภาพแสดง  $\mathbf{x}(2)$  ถึง  $\mathbf{x}(T)$  ด้วยสีแดง เพื่อสื่อถึงว่า ค่าอินพุตที่ลำดับเวลาเหล่านี้ ไม่ใช่อินพุตของระบบ แต่เป็นค่าที่ถูกสร้างขึ้นในกระบวนการ.

กำหนดให้ ค่าทำงานที่ลำดับเวลาสุดท้าย เป็นเอาต์พุต และค่าทำงานต่าง ๆ ที่ลำดับเวลาอื่น ๆ ไม่มีความสำคัญ. กระบวนการฝึก ในกรณีการจำแนกลำดับ ก็สามารถทำได้สะดวก โดยการใช้กลไกหน้ากาก ที่กำหนดให้  $\mathcal{M}(T) = 1$  และ  $\mathcal{M}(t \neq T) = 0$ . รูป 9.11 แสดงแผนภาพคลีลำดับ สำหรับกรณีการจำแนกลำดับ.

กรณีที่เอาต์พุตเป็นชุดลำดับ แต่อินพุตไม่ใช่ เช่น ระบบแต่งเพลงอัตโนมัติ (ดูแบบฝึกหัด 8.4) โครงข่ายประสาทเวียนกลับ ก็อาจสามารถนำมาประยุกต์ใช้ได้โดยจัดโครงสร้างดังแสดงในรูป 9.12. การจัดโครงสร้างที่นิยม สำหรับกรณีเช่นนี้ คือ ใช้อินพุตของระบบ เป็นอินพุตที่ลำดับเวลาแรกสุดของโครงข่ายประสาทเวียนกลับ และหลังจากนั้น ใช้เอาต์พุตที่คำนวณได้ มาเป็นอินพุตสำหรับลำดับเวลาถัดไป. การฝึกโครงข่ายประสาทเวียนกลับ สำหรับกรณีเช่นนี้ ซึ่งเป็นลักษณะของการใช้งานแบบจำลองก่อภัยนิด ก็สามารถใช้วิธีทางของงานแบบจำลองก่อภัยนิด เช่น แนวทางของโครงข่ายปรับแก้เชิงสร้างได้ (ดูหัวข้อ 7.2 ประกอบ).



รูปที่ 9.13: แผนภาพคลื่นลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถัวตอนดรหัส. สถาปัตยกรรมตัวเข้ารหัสตัวถัวตอนดรหัส ใช้โครงข่ายประสาท เวียนกลับสองตัว ตัวหนึ่งทำหน้าที่เข้ารหัส (encoder แบบจำลองทางซ้ายในภาพ) อีกตัวหนึ่งทำหน้าที่ถอดรหัส (decoder แบบจำลองทางขวาในภาพ). ตัวเข้ารหัส สรุปเนื้อความของชุดลำดับอินพุต ไว้เป็นรหัสเนื้อความ แล้วส่งรหัสเนื้อความนี้ไปให้ตัวถอดรหัส เพื่อถอดออกมาระบบชุดลำดับเอาร์พุต.

สุดท้าย กรณีที่ทั้งอินพุตและเอาร์พุตเป็นชุดลำดับ แต่มีจำนวนลำดับอาจไม่เท่ากัน เช่น ภารกิจการแปลภาษาอัตโนมัติ ที่จำนวนคำในประโยคของภาษาต้นทาง อาจไม่เท่ากับจำนวนคำในประโยคของภาษาเป้าหมาย. กรณีนี้ บอยครั้งอาจถูกจ้างถึงเป็นภารกิจจำลองแบบชุดลำดับเป็นชุดลำดับ (sequence-to-sequence modeling task) เป็นสถาณการณ์ที่ท้าทายอย่างมาก โดยเฉพาะ เมื่อการทำนายชุดลำดับของเอาร์พุต จำเป็นต้องเห็นอินพุตครบถ้วนลำดับก่อน. แนวทางหนึ่ง คือการใช้สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส (encoder-decoder architecture[38]) หรือบางครั้งอาจเรียกว่า สถาปัตยกรรมแปลงชุดลำดับไปชุดลำดับ (sequence-to-sequence architecture[193]) ดังแสดงในรูป 9.13 (ดูแบบฝึกหัด 9.5 เพิ่มเติม).

สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส คล้ายการรวมแบบจำลองสองตัว. ตัวแรก สำหรับชุดลำดับอินพุต (เรียกว่า ตัวเข้ารหัส) และตัวที่สอง สำหรับชุดลำดับเอาร์พุต (เรียกว่า ตัวถอดรหัส). สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส อาศัยกลไกของรหัสเนื้อความ (code) หรืออาจเรียกว่า context ใน การสรุปความหมายของชุดลำดับอินพุตไว้ แล้วค่อยถอดรหัสเนื้อความออกมาเป็นอีกชุดลำดับ.

โดยทั่วไป ผลการกระตุ้นลำดับสุดท้ายของตัวเข้ารหัส เช่น  $\mathbf{z}(T_x)$  ในรูป 9.13 จะถูกใช้เป็นรหัสเนื้อความ. ตัวถอดรหัส อาจรับรหัสเนื้อความ มาเป็นส่วนหนึ่งของผลการกระตุ้นเริ่มต้นของตัวถอดรหัส เช่น  $\tilde{\mathbf{z}}(0) = [\mathbf{z}(T_x); \mathbf{0}]$  โดย  $\mathbf{0}$  อาจเป็นค่าเริ่มต้นที่เติมเข้าไป เพื่อให้เต็มขนาด (ขนาดสถานะภายในของตัวถอดรหัส อาจใหญ่กว่าขนาดของรหัสเนื้อความได้). อินพุตลำดับแรกของตัวถอดรหัส  $\tilde{\mathbf{x}}(1)$  อาจกำหนดด้วยค่า  $\mathbf{0}$ . ดูแบบฝึกหัด 9.5 เพิ่มเติม สำหรับการอภิปรายโครงสร้างของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัสแบบอื่น.

โดยทั่วไป ขนาดสถานะภายในของตัวถอดรหัส ไม่เล็กกว่าขนาดสถานะภายในของตัวเข้ารหัส นั่นคือ  $|\mathbf{z}(t)| \leq |\tilde{\mathbf{z}}(t)|$ . การกำหนดขนาดสถานะภายในของตัวถอดรหัส ให้ใหญ่กว่าขนาดรหัส ช่วยให้ตัวถอดรหัส

สมือนมีความจำเหลือพอที่จะใช้งานอีน ๆ ได้ (เช่น อาจจะเก็บสถานะของการทำงาน ณ ลำดับเวลาปัจจุบัน).

แม้สถาบัตยกรรมตัวเข้ารหัสตัวถอดรหัส จะสามารถช่วยให้ชุดลำดับของเอกสารพูดมีจำนวนลำดับที่เป็นอิสระจากจำนวนลำดับของอินพุต และยังช่วยให้การทำนายชุดลำดับของเอกสารพูด ได้เทื่อนอินพุตครบถ้วนลำดับก่อน แต่สถาบัตยกรรมตัวเข้ารหัสตัวถอดรหัส ใช้กลไกของรหัส ในการส่งผ่านความหมายสรุปจากชุดลำดับอินพุต ไปสร้างชุดลำดับเอกสารพูด. ดังนั้น ขนาดของรหัส มีผลโดยตรงต่อความสามารถในการแทนความหมาย. ในกรณีของระบบการแปลภาษาอัตโนมัติ ขนาดของรหัสที่เล็กเกินไป จะส่งผลกระทบต่อคุณภาพการแปลอย่างชัดเจน โดยเฉพาะกับการแปลประโยคยาว ๆ [37, 9] (ดูแบบฝึกหัด 9.9 เพิ่มเติม. หัวข้อ 9.6 อภิปรายกลไกความใส่ใจ ซึ่งพัฒนามาเพื่อแก้ข้อจำกัดนี้).

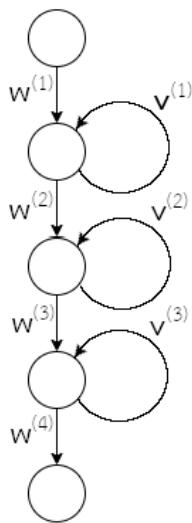
การจัดโครงสร้างเชิงลึกของโครงข่ายประสาทเวียนกลับ. โครงข่ายประสาทเวียนกลับ (สมการ 9.1 และ 9.2) สามารถจัดโครงสร้างแบบลึกได้ (รูป 9.14). อย่างไรก็ตาม ผู้เขียนชี้ว่า ศาสตร์การเรียนรู้ของเครื่องแอนดรอย์ อึ้ง[140] ให้ข้อสังเกตว่า แม้ปัจจุบัน โครงข่ายประเทียมเชิงลึกอาจมีจำนวนชั้นคำนวนเป็นหลักร้อยชั้นได้ (เช่น เรสเน็ต[85]) แต่สำหรับโครงข่ายประสาทเวียนกลับ โดยเฉพาะชั้นคำนวนเวียนกลับ มักตอกันไม่เกินสามชั้น<sup>4</sup> เนื่องจากส่วนหนึ่ง อาจเป็นผลกระทบจากการเวียนกลับได้เพิ่มความสามารถของแบบจำลองขึ้นอย่างมาก (รวมถึงเพิ่มความต้องการการคำนวนขึ้นอย่างมหาศาล โดยเฉพาะในกระบวนการฝึก). แต่โครงสร้างเชิงลึกมักพบ คือการใช้โครงข่ายประสาทเวียนกลับ ที่มีชั้นเวียนกลับ (หนึ่งถึงสามชั้น) และต่อด้วยชั้นคำนวน (ที่ไม่มีกลไกเวียนกลับ) หลาย ๆ ชั้น (รูป 9.15).

## 9.6 กลไกความใส่ใจ

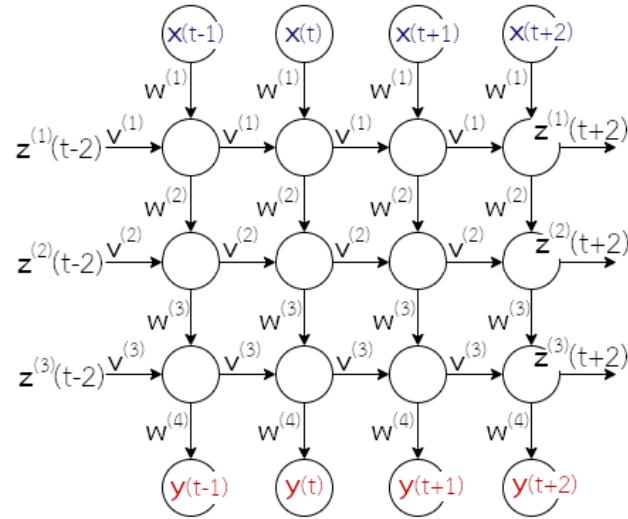
สำหรับภารกิจจำลองแบบชุดลำดับเป็นชุดลำดับ เช่น การแปลภาษาอัตโนมัติ (machine translation), การสรุปข้อความ (text summarization), แชทบอต (chatbot) และระบบตอบคำถาม (question answering) สถาบัตยกรรมตัวเข้ารหัสตัวถอดรหัส สามารถทำงานได้ดี แต่ความสามารถของระบบลดลงอย่างมาก เมื่อชุดลำดับอินพุตมีความยาวมาก ๆ. คณะของบาร์嗒โน[9] เชื่อว่าข้อจำกัดนี้ เกิดจากปัญหาความขาดที่การใช้รหัสเนื้อความ ซึ่งมีความยาวจำกัด และได้เสนอวิธีการแก้ด้วยกลไกความใส่ใจ.

<sup>4</sup> ข้อสังเกตนี้ ตั้งขึ้นจากสภาพสิ่งแวดล้อม และความนิยมในปัจจุบัน. ประวัติและวิวัฒนาการของโครงข่ายประสาทเทียม เช่น การใช้งานเพอร์เซปตรอนหลายชั้น (บทที่ 3) ในช่วงเวลา ก่อนปี ค.ศ. 2012 (ที่โครงข่ายประสาทเทียมเชิงลึกเริ่มได้รับความสนใจอย่างกว้างขวาง) ที่มีความเชื่อว่า ไม่มีความจำเป็นที่จะต้องใช้โครงข่ายที่มีจำนวนชั้นคำนวนเกินกว่าสองชั้น. ดังนั้น ข้อสังเกตนี้ ผู้เขียนบันทึกไว้ เพื่อผู้อ่านจะได้พอเร้นสภาพการนำไปใช้งาน แต่หวังว่า ผู้อ่านจะไม่ยึดติดจนเกินไป ดังที่ประวัติศาสตร์สอนได้ไว้ตลอดมา. นวัตกรรม ความก้าวหน้า พัฒนาการที่สำคัญ (breakthrough) เกิดจากการปล่อยวาง มากกว่าการยึดติด.

Structural diagram  
of a deep RNN

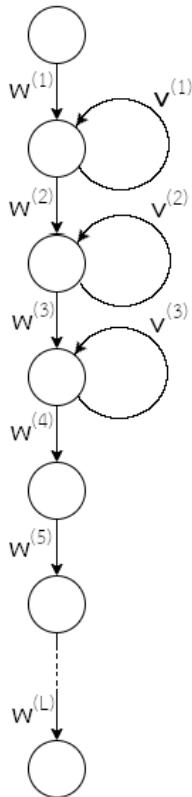


Unfolding diagram of a deep RNN

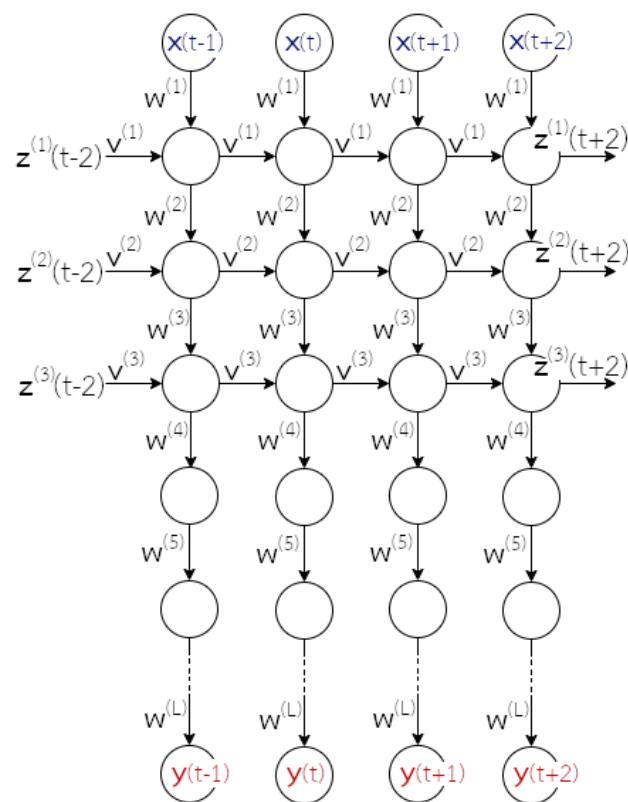


รูปที่ 9.14: โครงสร้างและแผนภาพคลี่ลำดับของโครงข่ายประสาทเวียนกลับแบบลึก. ตัวอย่างนี้ แสดงการใช้ชั้นเวียนกลับสามชั้น ต่อกัน.

Structural diagram of  
a deep RNN with  
non-recurrent layers



Unfolding diagram of a deep RNN  
with non-recurrent layers



รูปที่ 9.15: โครงสร้างและแผนภาพคลี่ลำดับของโครงข่ายประสาทเวียนกลับแบบลึก ที่ใช้ชั้นคำนวนไม่เวียนกลับจำนวนมาก.

แนวคิดของกลไกความใส่ใจ พัฒนาจากการกิจกรรมแปลภาษาอัตโนมัติ ซึ่งสังเกตว่า เวลาที่คนแปลภาษาแม้จะเป็นการแปลจากประ惰คายา ๆ ในภาษาต้นทาง แต่เวลาแปล ถอดความออกมานะเป็นคำ ๆ ในภาษาปลายทาง ผู้แปล แม้จะเห็นทั้งประ惰คาย แต่เวลาพิจารณาเลือกคำ แต่ละคำ ที่จะใช้สำหรับประ惰คป้ายทาง ผู้แปล จะใส่ใจกับเฉพาะบางส่วนของประ惰คต้นทางเท่านั้น.

ตัวอย่างเช่น ประ惰คต้นทาง (ภาษาอังกฤษ) คือ “Knowing that what is cannot be undone—because it already is — you say yes to what is or accept what isn’t.”<sup>5</sup>

ประ惰คป้ายทาง (ภาษาไทย) คือ “การรู้ว่า สิ่งที่เป็น ไม่สามารถเปลี่ยนแปลงได้ (เพราะว่ามันเป็นไปแล้ว) คุณยอมรับกับ สิ่งที่เป็น หรือยอมรับ สิ่งที่ไม่ได้เป็น.”

เอาร์พุต “การรู้ว่า” อาจมาจากการที่คืน “Knowing” และที่คืน “that” เป็นหลัก โดยส่วนอื่น ๆ ของประ惰คต้นทางมีความเกี่ยวข้องต่ำ.

**การทำงานของกลไกความใส่ใจ.** แทนที่จะอาศัยการส่งเนื้อความจากชุดลำดับอินพุต ไปชุดลำดับเอาร์พุต ผ่านรหัสเนื้อความที่มีความยาวจำกัด แล้วให้อิสระกับตัวถอดรหัสในการจัดเรียงเอาร์พุตเอง กลไกความใส่ใจ (attention mechanism) กำหนดบริบทสำหรับแต่ละลำดับ ให้กับตัวถอดรหัสโดยตรง.

ด้วยชุดลำดับอินพุต  $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(T_x)]$  สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส คำนวณรหัสเนื้อความ

$$\mathbf{c} = e'(\mathbf{z}(1), \dots, \mathbf{z}(T_x)) \quad (9.35)$$

เมื่อ

$$\mathbf{z}(t) = e(\mathbf{z}(t-1), \mathbf{x}(t), \mathbf{z}(t+1)) \quad (9.36)$$

โดย  $e'(\cdot)$  และ  $e(\cdot)$  เป็นฟังก์ชันของตัวเข้ารหัส ที่ทำหน้าที่สรุป และจำลองแบบเชิงลำดับ. ตัวอย่างเช่น  $e'(\mathbf{z}(1), \dots, \mathbf{z}(T_x)) = \mathbf{z}(T_x)$  และ  $e(\cdot)$  เป็นโครงข่ายประสาทเยื่องกลับสองทาง.

ในสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัสแบบดั้งเดิม (รูป 9.16) ตัวถอดรหัสรับรหัสเนื้อความ  $\mathbf{c}$  และระบบประเมินความนำจะเป็นของเอาร์พุตที่ลำดับ  $t$  ด้วย

$$p(\mathbf{y}(t) | \{\mathbf{y}(1), \dots, \mathbf{y}(t-1)\}, \mathbf{X}) \approx \mathcal{D}(\mathbf{y}(t-1), \tilde{\mathbf{z}}(t), \mathbf{c}) \quad (9.37)$$

<sup>5</sup> จาก Eckhart Tolle, The Power of Now.

เมื่อ  $\mathcal{D}(\cdot)$  เป็นฟังก์ชันที่อนุมานความน่าจะเป็นของเอาร์พุต<sup>6</sup> และ  $\tilde{\mathbf{z}}(t)$  เป็นสถานะซ่อน ของโครงข่ายประสาทเวียนกลับ ที่ลำดับเวลา  $t$ .

จากสมการ 9.37 เราจะเห็นว่า ตัวถอดรหัสรับรู้ชุดลำดับอินพุต  $\mathbf{X}$  ผ่านรหัสเนื้อความ  $\mathbf{c}$  ตลอดการประมาณเอาร์พุตทุก ๆ ลำดับ. ดังนั้น ชุดลำดับอินพุตที่มีความยาวมาก อาจจะยัดเยียดสารสนเทศจำนวนมาก ลงไปในรหัสเนื้อความ  $\mathbf{c}$  ซึ่งแม้โดยแนวคิด รหัสเนื้อความไม่ได้ถูกจำกัดขนาด แต่ในทางปฏิบัติ การนำแนวคิดสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัสไปใช้งาน จะกำหนดขนาดของรหัสเนื้อความนี้ และขนาดของรหัสเนื้อความที่จำกัด อาจทำให้เกิดปัญหาความขาดกับชุดลำดับยาว ๆ ได้.

กลไกความใส่ใจ เสนอใช้บริบทตามตำแหน่งลำดับ แทนที่จะใช้รหัสเนื้อความเดียวกันทุก ๆ ลำดับ. นั่นคือ ตัวถอดรหัส ประมาณความน่าจะเป็นของเอาร์พุตที่ลำดับ  $t$  ด้วย

$$p(\mathbf{y}(t) | \{\mathbf{y}(1), \dots, \mathbf{y}(t-1)\}, \mathbf{X}) \approx \mathcal{D}(\mathbf{y}(t-1), \tilde{\mathbf{z}}(t), \mathbf{c}_t) \quad (9.38)$$

โดย  $\mathbf{c}_t$  เป็นบริบทสำหรับเอาร์พุตที่ลำดับเวลา  $t$ . ค่าสถานะซ่อนคำนวนได้จาก

$$\tilde{\mathbf{z}}(t) = d(\tilde{\mathbf{z}}(t-1), \mathbf{y}(t-1), \mathbf{c}_t) \quad (9.39)$$

เมื่อ  $d(\cdot)$  เป็นส่วนของตัวถอดรหัสที่ทำหน้าที่จำลองแบบเชิงลำดับ<sup>7</sup>.

บริบท  $\mathbf{c}_t$  ควรจะขึ้นกับชุดลำดับของเนื้อความย่ออย  $\{\mathbf{z}(1), \dots, \mathbf{z}(T_x)\}$  ที่ตัวเข้ารหัสได้ไว้เคราะห์มา จากชุดลำดับอินพุต. เนื่องจากตัวเข้ารหัสเป็นโครงข่ายประสาทเวียนกลับสองทาง แต่ละเนื้อความย่ออย  $\mathbf{z}(t)$  จึงถูกสรุปมาจากอินพุตทั้งชุดลำดับ โดยเน้นลำดับต่าง ๆ บริเวณรอบ ๆ ลำดับ  $t$  (ของชุดลำดับอินพุต).

คลนของบทดานา[9] กำหนดให้บริบทของเอาร์พุตที่ลำดับเวลา  $t$  เป็นผลรวมตามน้ำหนักของเนื้อความย่ออยต่าง ๆ. นั่นคือ

$$\mathbf{c}_t = \sum_{t'=1}^{T_x} \alpha_{t,t'} \cdot \mathbf{z}(t') \quad (9.40)$$

โดย  $\alpha_{t,t'}$  เรียกว่า ค่าน้ำหนักความใส่ใจ (attention weight) เป็นค่าน้ำหนักที่ระบุการจัดตำแหน่งแบบอ่อน ๆ (soft alignment) ของลำดับเอาร์พุต เทียบกับลำดับอินพุต. และเพื่อให้การคำนวนมีเสถียรภาพ ค่าน้ำหนัก

<sup>6</sup>นิพจน์นี้ (บรรยายตามคลนของบทดานา[9]) ต้องการสื่อถึงความสัมพันธ์ระหว่างรหัสเนื้อความกับการแทนใจความของลำดับอินพุต ในมุมมองความน่าจะเป็น. ในทางปฏิบัติ บ่อยครั้งที่ฟังก์ชันอนุมาน  $\mathcal{D}(\cdot)$  รับสารสนเทศผ่านสถานะซ่อน ซึ่งหมายถึง  $p(\mathbf{y}(t) | \{\mathbf{y}(1), \dots, \mathbf{y}(t-1)\}, \mathbf{X}) \approx \mathcal{D}(\tilde{\mathbf{z}}(t))$  โดย  $\tilde{\mathbf{z}}(t)$  มักอนุมานได้จาก  $\tilde{\mathbf{z}}(t) = d(\tilde{\mathbf{z}}(t-1), \mathbf{y}(t-1), \mathbf{c})$  เมื่อ  $d(\cdot)$  เป็นส่วนของตัวถอดรหัส.

<sup>7</sup>เนื่องจาก  $d(\cdot)$  ต้องให้ผลลัพธ์ออกมาทุก ๆ ลำดับ ก่อนที่จะจบลำดับ ซึ่งแม้แต่การจบลำดับก็อาจจะถูกกำหนดด้วยค่าของผลลัพธ์ที่ออกมาก ดังนั้น  $d(\cdot)$  อาจเป็นโครงข่ายประสาทเวียนกลับ แต่ไม่ใช่โครงข่ายประสาทเวียนกลับสองทาง.

ความใส่ใจ  $\alpha_{t,t'}$  จะถูกควบคุมให้  $0 \leq \alpha_{t,t'} \leq 1$  สำหรับทุก ๆ  $t$  และ  $t'$  และ  $\sum_{t'} \alpha_{t,t'} = 1$  ด้วยฟังก์ชันซอฟต์แมกซ์

$$\alpha_{t,t'} = \frac{\exp(r_{t,t'})}{\sum_{\tau=1}^{T_x} \exp(r_{t,\tau})} \quad (9.41)$$

เมื่อ  $r_{t,t'}$  เป็นแบบจำลองการจัดเรียงตำแหน่ง (alignment model) ที่ระบุคะแนนว่าอินพุตลำดับ  $t'$  ควรมีผลกับเอกสารลำดับ  $t$  มากน้อยขนาดไหน. แบบจำลองการจัดเรียงตำแหน่ง ควรขึ้นอยู่กับสถานะซ่อนจากคุณลักษณะของอินพุตและเอกสารลำดับ  $t$  ดังนี้

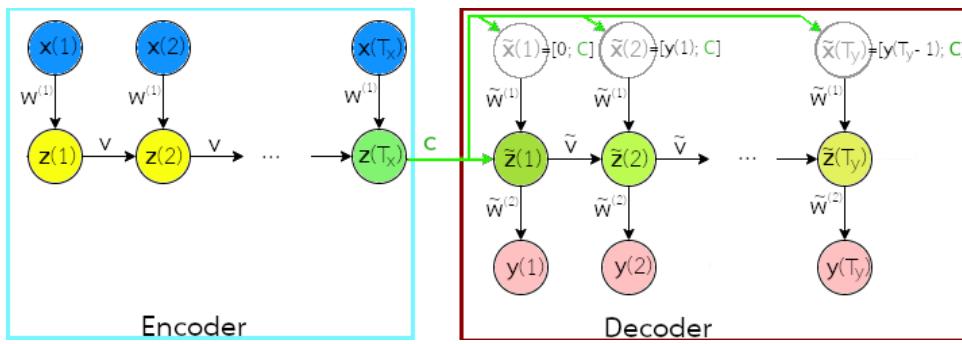
$$r_{t,t'} = f(\tilde{z}(t-1), z(t')) \quad (9.42)$$

โดย  $f(\cdot)$  เป็นฟังก์ชันคำนวณคะแนนความสัมพันธ์ระหว่างลำดับ  $t'$  ของอินพุต กับลำดับ  $t$  ของเอกสาร. สมการ 9.42 ใช้  $\tilde{z}(t-1)$  แทนที่จะเป็น  $\tilde{z}(t)$  เพราะว่า  $\tilde{z}(t-1)$  คือสถานะซ่อนล่าสุดของตัวออดรหัส (ดูลำดับการคำนวณประกอบ). ฟังก์ชันคะแนน  $f(\cdot)$  ที่สามารถทำได้จากโครงข่ายประสาทเทียม โดยทำการฝึกโครงข่ายไปพร้อม ๆ กับแบบจำลองส่วนอื่น ๆ ของระบบ.

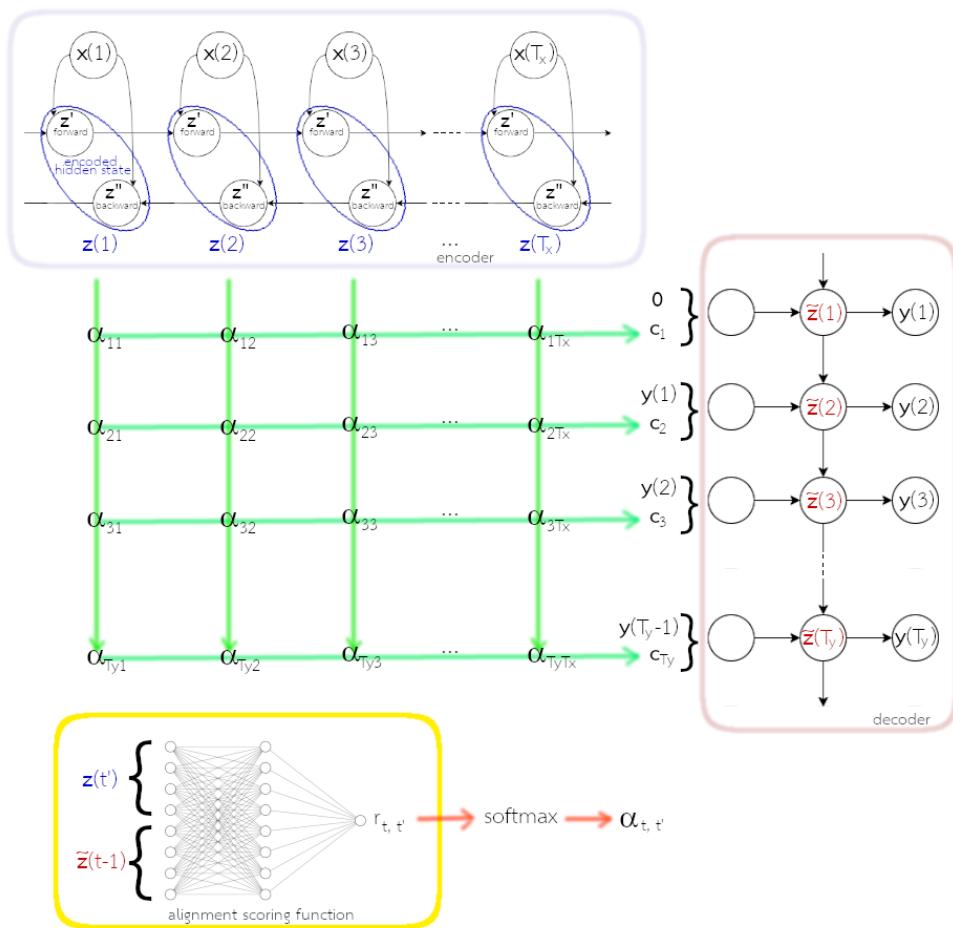
สังเกต ฟังก์ชัน  $f(\cdot)$  คำนวณคะแนน โดยใช้ค่าเวคเตอร์จากสถานะซ่อนไม่ได้ใช้ค่าดัชนี  $t$  หรือ  $t'$  โดยตรง. ดังนั้น (1) ไม่ต้องกังวลเรื่องความพยายามของลำดับ และ (2) การเชื่อมโยงระหว่างลำดับอินพุต และลำดับเอกสาร ทำผ่านสถานะซ่อน ไม่ได้ขึ้นกับตำแหน่งสัมบูรณ์.

รูป 9.17 แสดงโครงสร้าง เมื่อใช้กลไกความใส่ใจ. สถานะซ่อนของตัวเข้ารหัสทุก ๆ ลำดับเวลา จะถูกนำไปประกอบเป็นบริบท โดยสถานะซ่อนที่ลำดับเวลาใด จะถูกผสมเข้าไปมากน้อยเท่าไร ขึ้นกับน้ำหนักความใส่ใจ. สถานะซ่อนของตัวเข้ารหัส  $z(t) = [z'(t); z''(t)]$  เมื่อ  $z'(t)$  และ  $z''(t)$  คือ ผลกระทบตุนในทิศทางไปข้างหน้า และกลับหลัง ตามลำดับ.

ด้วยกลไกความใส่ใจ ไม่ว่าชุดลำดับอินพุตจะยาวเท่าไร แต่ละลำดับเวลาของเอกสาร จะสมீอ่อนมองเห็นทั้งชุดลำดับอินพุต เพียงแต่เน้นความหมายจากเฉพาะส่วนที่เกี่ยวข้องเท่านั้น ซึ่งแม้ขนาดของบริบทที่ลำดับ  $c_t$  จะไม่ได้ใหญ่ไปกว่าขนาดของรหัสเนื้อความ  $c$  แต่การที่ค่าของบริบทเปลี่ยนไปตามลำดับของเอกสารได้ ช่วยแก้ปัญหาความขาดของภารกิจจำลองแบบชุดลำดับเป็นชุดลำดับได้อย่างดี. ปัจจุบัน กลไกความใส่ใจ เป็นศาสตร์และศิลป์ของการเรียนรู้เชิงลึก โดยเฉพาะภารกิจเกี่ยวกับการประมวลผลภาษาธรรมชาติ มีการประยุกต์ใช้อย่างกว้างขวาง (รวมถึงภารกิจนอกเหนือจากการประมวลผลภาษาธรรมชาติทั่วไปด้วย เช่น [221]) และเป็นแรงบันดาลใจให้เกิดการพัฒนาอย่างต่อเนื่อง จนเป็นแนวทางของตัวแปลง (transformer[203, 40, 24]).



รูปที่ 9.16: สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส ที่ใช้รหัสเนื้อความประกอบอินพุตของตัวถอดรหัส.



รูปที่ 9.17: แผนภาพแสดงโครงสร้าง เมื่อใช้กลไกความ似ใจ. ค่าน้ำหนักความ似ใจ  $\alpha_{t,t'}$  จะถูกคำนวณทุก ๆ รอบของลำดับເອເຕີພຸດ  $t$ . ແນວ່າ ค่าน้ำหนักความ似ใจ ໃນການ ອາຈຸດເໜີມອນເມທຣິກ້ຈ ແຕ່  $\alpha_{t,t'}$  ອູກຄານວັນຈາກຄໍາຂະແນນ  $r_{t,t'}$  ຈຶ່ງເປັນຝັກ໌ຂັ້ນຂອງຄ່າສານະ ຂ້ອນ ໄນໄດ້ຂັ້ນກັບລຳດັບ  $t$  ແລະ  $t'$  ໂດຍຕຽງ. ຄໍາຄະແນນ  $r_{t,t'}$  ລາຍໄດ້ຈາກໂຄຮງຂ່າຍປະສາທເຖິມ ດັ່ງແສດງໃນການ (ກາຍໃນກຣອບສື່ເໜືອງ ດ້ວຍລ່າງ).

## 9.7 อภิธานศัพท์

**โครงข่ายประสาทเวียนกลับ (Recurrent Neural Network คำย่อ RNN):** โครงข่ายประสาทเทียม ที่โครงสร้างการคำนวณมีการเชื่อมต่อ ที่นำค่าที่คำนวณแล้วในลำดับเวลา ก่อนกลับเข้ามาคำนวณในลำดับเวลาปัจจุบันด้วย.

**แผนภาพคลี่ลำดับ (unfolding diagram):** แผนภาพโครงสร้างของโครงข่ายประสาทเทียม ที่กระจายการแสดงการเวียนกลับเชิงลำดับเวลา ออกเป็นลักษณะเดียวกับโครงสร้างตระรากภาษาพ. แผนภาพคลี่ลำดับ นิยมใช้แสดงการเชื่อมต่อของโครงข่ายประสาทเวียนกลับ.

**การแพร่กระจายย้อนกลับผ่านเวลา (backpropagation through time คำย่อ BPTT):** ขั้นตอนวิธีการคำนวณเกรเดียนต์ สำหรับโครงข่ายประสาทเวียนกลับ.

**ปัญหาการระเบิดของเกรเดียนต์ (exploding gradient problem):** ปัญหาที่อาจพบกับการฝึกโครงข่ายประสาทเวียนกลับ ที่เกรเดียนต์มีค่าเพิ่มขึ้นอย่างมาก เมื่อเวียนกลับย้อนลำดับเวลา.

**การเล้มเกรเดียนต์ (gradient clipping):** วิธีแก้ปัญหาการระเบิดของเกรเดียนต์ ด้วยวิธีจำกัดขนาดของเกรเดียนต์ ที่จะใช้คำนวนปรับค่าน้ำหนัก.

**โครงข่ายประสาทเวียนกลับสองทาง (bidirectional recurrent neural network คำย่อ BRNN):** โครงข่ายประสาทเวียนกลับ ที่ใช้นำค่าสถานะทั้งในลำดับก่อนหน้า (ทิศทางปกติ) และลำดับหลัง (ทิศทางย้อนกลับ) เวียนมาคำนวณผลลัพธ์ในลำดับปัจจุบัน.

**แบบจำลองความจำระยะสั้นที่ยาว (long short-term memory model คำย่อ LSTM):** แบบจำลอง โครงข่ายประสาทเวียนกลับ ที่มีการใช้กลไกของประตุคุม การปรับเปลี่ยนค่าสถานะและค่าผลลัพธ์ของหน่วยคำนวณ เพื่อช่วยเพิ่มประสิทธิผลในการรู้จำความสัมพันธ์ระยะยาวของข้อมูล.

**ช่องแอบมอง (peephole connections):** กลไกเพิ่มเติม สำหรับแบบจำลองความจำระยะสั้นที่ยาว เพื่อช่วยให้การควบคุมเปลี่ยนค่าสถานะและค่าผลลัพธ์ของหน่วยคำนวณ ทำได้แม่นยำมากขึ้น โดยนำค่าสถานะเข้าไปเป็นส่วนหนึ่งในการพิจารณาการเปิดปิดของประตุด้วย.

**สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส (encoder-decoder architecture):** โครงสร้างการต่อเชื่อมที่ใช้โครงสร้างประสาทสัมภาระกลับสองตัวต่อกัน โดยให้ตัวหนึ่งประมวลผลชุดลำดับอินพุต และอีกตัวประมวลผลชุดลำดับเอาต์พุต สำหรับการกิจจำลองแบบชุดลำดับเป็นชุดลำดับ.

**การประมวลผลภาษาธรรมชาติ (Natural Language Processing คำย่อ NLP):** ศาสตร์ที่ใช้วิธีการต่าง ๆ เพื่อให้คอมพิวเตอร์สามารถนำข้อความในภาษาธรรมชาติไปประมวลผล และให้ผลลัพธ์ตามจุดประสงค์ของภารกิจที่ต้องการ.

**โทเค็น (token):** หน่วยพื้นฐานของภาษาที่มีความหมาย เช่น โทเค็น อาจหมายถึง คำ.

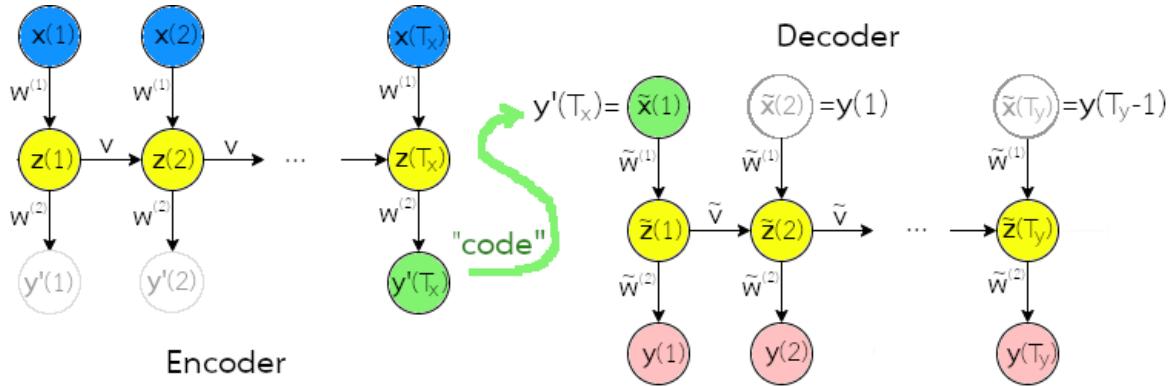
**ไวยากรณ์ (syntax):** กฎเกณฑ์ที่เกี่ยวกับโทเค็น และโครงสร้าง.

**การแยกส่วน (parsing):** การวิเคราะห์โครงสร้างไวยากรณ์ของข้อความหรือประโยค.

**การระบุหมวดคำ (Part-Of-Speech Tagging):** ภารกิจที่ระบุว่า คำต่าง ๆ ในข้อความ แต่ละคำอยู่ในหมวดคำใด จากหมวดคำ เช่น นาม สรรพนาม กริยา วิเศษณ์ คุณศัพท์ สันธาน บุพบท อุทาน.

**การกิจจำลองแบบชุดลำดับเป็นชุดลำดับ (sequence-to-sequence modeling task):** ภารกิจที่ทั้งอินพุตและเอาต์พุตเป็นชุดลำดับ แต่จำนวนลำดับของชุดอินพุต อาจไม่เท่ากับจำนวนลำดับในชุดเอาต์พุต

**กลไกความสนใจ (attention mechanism):** กลไกสำหรับภารกิจจำลองแบบชุดลำดับเป็นชุดลำดับ เพื่อช่วยบรรเทาปัญหา เมื่อทำงานกับชุดลำดับที่ยาว.



รูปที่ 9.18: แผนภาพคลื่ล้ำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส. สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส โดยใช้เอาร์พุตจากตัวเข้ารหัส นำมาเป็นอินพุตสำหรับตัวถอดรหัส.

## 9.8 แบบฝึกหัด

“If you talk to a man in a language he understands, that goes to his head. If you talk to him in his language, that goes to his heart.”

---Nelson Mandela

“ถ้าคุณคุยกับคนด้วยภาษาที่เขาฟังรู้เรื่อง สิ่งที่คุณพูดจะเข้าไปในหัวเขา. ถ้าคุณคุยกับเขาด้วยภาษาของเข้า สิ่งที่คุณพูดจะเข้าไปในใจเขา.”

—เนลสัน แมนเดลา

### แบบฝึกหัด 9.1

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายประสาทเทียม เช่น เพอร์เซปตรอนหลายชั้น ที่ไม่มีกลไกการเวียนกลับ กับข้อมูลเชิงลำดับ พร้อมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่ายประสาทเทียมเวียนกลับ.

### แบบฝึกหัด 9.2

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายคอนโวลูชัน ที่ไม่มีกลไกการเวียนกลับ กับ ข้อมูลเชิงลำดับ พร้อมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่ายประสาทเทียมเวียนกลับ.

### แบบฝึกหัด 9.3

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายประสาทเวียนกลับ กับข้อมูลที่มีความสัมพันธ์ เชิงท้องถิ่นหลายมิติ เช่น ข้อมูลภาพ พร้อมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่ายคอนโวลูชัน.

## แบบฝึกหัด 9.4

รูป 9.18 แสดงการจัดโครงสร้างสถานปัตยกรรมตัวเข้ารหัสตัวถอดรหัส โดยใช้เอต์พุตจากตัวเข้ารหัส นำมาเป็นอินพุตสำหรับตัวถอดรหัส

จงอภิราย ข้อดี ข้อเสีย และปัญหาของการจัดโครงสร้างแบบนี้ เปรียบเทียบแบบใช้สถานะภายในเป็นรหัส (รูป 9.13)

## แบบฝึกหัด 9.5

นอกจาก การจัดโครงสร้างดังแสดงในรูป 9.13 แล้วสถานปัตยกรรมตัวเข้ารหัสตัวถอดรหัส อาจผ่านรหัส ข้อความ ไปเป็นส่วนหนึ่งของอินพุตของตัวถอดรหัสได้ (รูปแบบที่สอง) หรือ แม้แต่จะผ่านรหัสข้อความ ไปเป็นทั้งสถานะเริ่มต้น และส่วนหนึ่งของอินพุต ดังแสดงในรูป 9.16 (รูปแบบที่สาม)

จงอภิราย ข้อดี ข้อเสีย การจัดโครงสร้างแบบต่าง ๆ นี้ รวมถึงอภิรายปัจจัยที่เกี่ยวข้อง สถานการณ์ที่บางรูปแบบอาจทำงานได้ดีกว่า พร้อมออกแบบการทดลอง ดำเนินการทดลอง และนำเสนอผล.

## แบบฝึกหัด 9.6

จงศึกษาการกิจการจำแนกอารมณ์ แนวทางปฏิบัติ การวัดผล และข้อมูลที่นิยม และสร้างระบบการจำแนกอารมณ์ พร้อมประเมินผล.

## แบบฝึกหัด 9.7

จงศึกษาการกิจการระบุหมวดคำ แนวทางปฏิบัติ การวัดผล และข้อมูลที่นิยม และสร้างระบบการจำแนกอารมณ์ พร้อมประเมินผล.

## แบบฝึกหัด 9.8

จงศึกษาการทำงานของระบบสั่งเคราะห์เสียง เช่น เวฟเน็ต (Wavenet[200]) อภิรายถึงแนวทางและวิธีที่ใช้ รวมถึงการประเมินผลและข้อมูล.

## แบบฝึกหัด 9.9

จงศึกษาสถานปัตยกรรมตัวเข้ารหัสตัวถอดรหัส (อาจเริ่มจากบทความที่ทรงอิทธิพล[37, 193]) ออกแบบการทดลอง เพื่อศึกษาปัจจัยความยาวของชุดลำดับข้อมูลกับประสิทธิภาพของระบบ ดำเนินการทดลอง รายงานผล และสรุป.

## แบบฝึกหัด 9.10

จงรวมกลุ่มระดมความคิด และอภิปรายแนวทางที่จะแก้ปัญหาระบบแปลภาษา เพื่อแก้ปัญหาคอขาดในสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส

หมายเหตุ แบบฝึกหัดนี้ ต้องการฝึกการคิดเชิงวิพากษ์ และฝึกความคิดสร้างสรรค์ อีกทั้งจะทำให้เห็นคุณค่าของกลไกความใส่ใจด้วย แต่เพื่อให้แบบฝึกหัดนี้ บรรลุจุดประสงค์ได้ ควรทำแบบฝึกหัดนี้ก่อนที่จะศึกษาเรื่องกลไกความใส่ใจ หรือหากได้ศึกษาเรื่องกลไกความใส่ใจไปแล้ว อาจจะลองเปิดใจมองหาแนวทางอื่น ๆ ที่อาจจะช่วยบรรเทาปัญหานี้ได้.

### แบบฝึกหัด 9.11

จงศึกษาพัฒนาการและกลไกที่สำคัญของศาสตร์แบบจำลองชุดข้อมูลลำดับ จากโครงข่ายประสาทเวียนกลับ แบบจำลองความจำระยะสั้นที่ยาว สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส กลไกความใส่ใจ และตัวแปลงชนิดต่าง ๆ (เช่น Transformer, BERT, GPT-3) สรุปประเด็น แนวทางที่สำคัญ ลักษณะปัญหา และการประยุกต์ใช้ รวมถึงช่วงเวลา และอภิปราย เหตุผล แรงขับดันเบื้องหลังพัฒนาการเหล่านี้.

### แบบฝึกหัด 9.12

จงศึกษาศาสตร์การประมวลผลภาษาธรรมชาติ ในเชิงกว้าง ถึงการกิจต่าง ๆ ที่สำคัญ และบริบทในแง่ความต้องการของสังคม รวมถึง ความก้าวหน้าในแต่ละภารกิจ เมื่อเทียบกับจุดมุ่งหมาย และอภิปรายโอกาส การประยุกต์ใช้ต่าง ๆ ที่อาจจะนอกเหนือจากประยุกต์ใช้เดิม และความท้าทายต่าง ๆ ในงานวิจัย เพื่อบรรลุจุดประสงค์ รวมถึงเชื่อมโยงสิ่งที่ได้เรียนรู้ ความก้าวหน้า การประยุกต์ใช้ จำกบริบทในแง่ความต้องการของสังคม.

### แบบฝึกหัด 9.13

จงเลือกการกิจการประมวลผลภาษาธรรมชาติที่สนใจ และศึกษาภารกิจ แนวทางปฏิบัติ ปัจจัยที่เกี่ยวข้อง การวัดผล และข้อมูลที่นิยม และทดลองสร้างระบบสำหรับภารกิจที่เลือก ประเมินผล ให้ความเห็น และสรุปสิ่งที่ได้เรียนรู้.



## បរទានក្រម

- [1] Access to Insight, W. Tipitaka: The pali canon. <http://www.accesstoinsight.org/tipitaka/index.html>.
- [2] Adrian, E. D., and Zotterman, Y. The impulses produced by sensory nerve endings. *Journal of Physiology* 61 (1926), 151–171.
- [3] Akiyama, T., Hachiya, H., and Sugiyama, M. Efficient exploration through active learning for value function approximation in reinforcement learning. *Neural Networks* 23 (2010), 639–648.
- [4] Alake, R. How you should read research papers according to andrew ng. online: <https://towardsdatascience.com>, July 2nd 2020.
- [5] Anderson, C. W., Hittle, D., Kretchmar, M., and Young, P. *Handbook of learning and approximate dynamic programming*. John Wiley & Sons, 2004, ch. Robust reinforcement learning for heating, ventilation, and air conditioning control of buildings.
- [6] Arjovsky, M., and Bottou, L. Towards principled methods for training generative adversarial networks. In *NIPS* (2016).
- [7] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. In *International Conference on Machine Learning* (2017).
- [8] Bache, K., and Lichman, M. UCI machine learning repository, 2013.
- [9] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *ICLR* (2015).

- [10] Barnard, M., Wang, W., Kittler, J., Naqvi, S. M., and Chambers, j. Audio-visual face detection for tracking in a meeting room environment. In *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013* (Istanbul, Turkey, 2013), pp. 1222–1227.
- [11] Benenson, R., Omran, M., Hosang, J., and Schiele, B. Ten years of pedestrian detection, what have we learned? In *European Conference on Computer Vision (ECCV)* (2014).
- [12] Bengio, Y. Curriculum learning. In *ICML'2009* (2009), pp. 41–48.
- [13] Bengio, Y. *Practical Recommendations for Gradient-Based Training of Deep Architectures*, vol. 7700 of *Lecture Notes in Computer Science*. Springer, 2012.
- [14] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspective. arXiv:1206.5538v3, 2014.
- [15] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, New York, USA, 2006.
- [16] Blais, B. S., and Cooper, L. Bcm theory. [http://www.scholarpedia.org/article/BCM\\_theory#Original\\_BCM\\_.28Bienenstock\\_et\\_al.\\_1982.29](http://www.scholarpedia.org/article/BCM_theory#Original_BCM_.28Bienenstock_et_al._1982.29), 2008. Scholarpedia 3(3):1570.
- [17] Blanzieri, E., and Bryl, A. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review* 29, 1 (2008), 63–92.
- [18] Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [19] Blum, A., and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory* (1998), Morgan Kaufmann, pp. 92–100.

- [20] Bock, R., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jirina, M., Klaschka, J., Kotrc, E., Savicky, P., Towers, S., Vaicius, A., and W., W. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A* 516 (2004), 511–528.
- [21] Boonkwan, P. Introduction to natural language processing. Seminar: AI Chatbot for Business, May 2017.
- [22] Bradley, D. Learning in modular systems. The Robotics Institute, Carnegie Mellon University, 2009. Doctoral dissertation.
- [23] Brock, A., Lim, T., Ritchie, J. M., and Weston, N. Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations* (2017).
- [24] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *arXiv.org* (2020).
- [25] C., N., Isarankura-Na-Ayadhy, C., Naenna, T., and Prachayasittikul, V. A practical overview of qualitative structure-activity relationship. *EXCLI Journal* 8 (2009), 74–88.
- [26] Caetano dos Santos, F. L., Paci, M., Nanni, L., Brahnam, S., and Hyttinen, J. Computer vision for virus image classification. *Biosystems Engineering Special Issue: Innovations in Medicine* (2015), 1–12.
- [27] Cao, D. S., Xu, Q. S., Hu, Q. N., and Liang, Y. Z. Chemopy: freely available python package for computational biology and chemoinformatics. *Bioinformatics* 29, 8 (2013), 1092–1094.

- [28] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR* (2017).
- [29] Carroll, S. B. *The Serengeti Rules*. Princeton University Press, 2016.
- [30] Castelletti, A., Pianosi, F., and Restelli, M. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water resource research* (2013).
- [31] Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys 41*, 3 (2009), 1–58.
- [32] Chang, C.-C., and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [33] Chanloha, P., Chinrungrueng, J., Usaha, W., and Aswakul, C. Cell transmission model-based multiagent q-learning for network-scale signal control with transit priority. *Computer Journal 57*, 3 (2014), 451–468.
- [34] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (2002), 321–357.
- [35] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., and Temam, O. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In *ASPLOS ’14* (2014), pp. 269–283.
- [36] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems* (2016).

- [37] Cho, K., van Merriënboer, B., and Bahdanau, D. On the properties of neural machine translation: Encoder–decoder approaches. In *EMNLP 2014: Conference on Empirical Methods in Natural Language Processing* (2014).
- [38] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP* (2014).
- [39] Chong, E. K. P., and Zak, S. *An Introduction to Optimization*, 2nd ed. Wiley-Interscience, 2001.
- [40] Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. What does bert look at? an analysis of bert’s attention. In *BlackBoxNLP@ACL* (2019).
- [41] Clinic, M. Mayo clinic website. internet.
- [42] Coates, A., Abbeel, P., and Ng, A. Y. Apprenticeship learning for helicopter control. *Communication of the ACM* (2009).
- [43] Cortes, C., and Vapnik, V. Support-vector networks. *Machine Learning* 20, 3 (Sep 1995), 273–297.
- [44] Costa-Jussa, M. R., and Farrus, M. Statistical machine translation enhancements through linguistic levels: A survey. *ACM Computing Surveys* 46, 3 (2014).
- [45] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65.
- [46] Culjak, M., Mikus, B., Jez, K., and Hadjic, S. Classification of art paintings by genre. In *34th International Convention on Information and Communication Technology, Electronics and Microelectronics* (Opatija, Croatia, 2011).

- [47] Cybenko, G. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems* 2, 4 (1989), 303–314.
- [48] Dahl, G. E., Sainath, T. N., and Hinton, G. E. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)* (Vancouver, BC, Canada, 2013).
- [49] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *In CVPR* (2005), pp. 886–893.
- [50] Damera-Venkata, N., Kite, T., Geisler, W., Evans, B., and Bovik, A. Image quality assessment based on a degradation model. *IEEE Transactions on Image Processing* 9, 4 (2000), 636–650.
- [51] Deb, D., Aggarwal, D., and Jain, A. K. Child face age-progression via deep feature aging. arXiv:2003.08788, 2020.
- [52] Deng, L., Hinton, G., and Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: an overview. In *38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)* (Vancouver, BC, Canada, 2013), pp. 8599–8603.
- [53] Denton, E. L., Chintala, S., Szlam, A., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems* (2015), pp. 1486–1494.
- [54] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT* (2019).
- [55] DiMasi, J., Grabowski, H. G., and Hansen, R. W. Innovation in the pharmaceutical industry: New estimates of r&d costs. *Journal of Health Economics* 47, number (2016), 20–33.

- [56] Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. In *International Conference on Learning Representations* (2017).
- [57] Dong, C., Loy, C. C., He, K., and Tang, X. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision* (2014).
- [58] Downey, A. *Think Python: How to Think Like a Computer Scientist*, 2nd edition ed. Green Tea Press, 2015.
- [59] Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially learned inference. In *International Conference on Learning Representations* (2017).
- [60] Dumoulin, V., and Visin, F. A guide to convolution arithmetic for deep learning. arXiv: 1603.07285v2, 2018.
- [61] Eisenstein, J. *Introduction to Natural Language Processing*. MIT Press, 2019.
- [62] Elter, M., Schulz-Wendtland, R., and Wittenberg, T. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics* 34, 11 (2007), 4164–4172.
- [63] Endo, A., Kuroda, M., and Tsujita, Y. Ml-236a, ml-236b, and ml-236c, new inhibitors of cholesterologenesis produced by penicillium citrinum. *Journal of Antibiotics* 29, 12 (1976), 1346–1348.
- [64] Erhan, D., Bengio, Y., Courville, A., Manzagol, P., and Vincent, P. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11 (2010), 625–660.
- [65] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* (2010).

- [66] Ettouati, W., Ma, J., Bourne, P., and Christopher, R. J. Drug discovery. Coursera.org, Winter 2013.
- [67] Fei-Fei, L., and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition* (2005).
- [68] Fei-Fei, L., and Perona, P. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition* (2008).
- [69] Felzenszwalb, P. F., Girshick, R. B., McAllister, D., and Ramanan, D. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9 (2010), 1627–1645.
- [70] Geman, S., Bienenstock, E., and Doursat, R. Neural networks and the bias/variance dilemma. *Neural Computation* 4 (1992), 1–58.
- [71] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)* (1999), vol. 2, pp. 850–855.
- [72] Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3 (2002).
- [73] Ghazanfar, M. A., and Prugel-Bennett, A. Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications* 41, 7 (2014), 3261–3272.
- [74] Glorot, X., and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)* (May 2010), vol. 9, pp. 249–256.
- [75] Godwin, D., and Cham, J. Your brain by the numbers. *Scientific American* (November 2012).

- [76] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.
- [77] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.
- [78] Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.
- [79] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28, 10 (2017).
- [80] Grimmett, G., and Stirzaker, D. *Probability and Random Processes*, 3rd ed. ed. Oxford University Press, 2001.
- [81] Grzymala-Busse, J. W., and Hu, M. A comparison of several approaches to missing attribute values in data mining. In *the 2nd International Conference on Rough Sets and New Trends in Computing* (2000), pp. 340–347. Banff.
- [82] Hanson, R., and Mendius, R. សមօងແຫ່ງພູທຣະ, 1 ed. ອັນດີນທຣຣມະ, 378 ດັ.ຊ້ຍພຖາກ່ານ ຕລິ່ງໜັນ ກຽມທະເວົາ 10170, ພ.ສ. 2557. ແປລໂດຍ ຄັ້ງ ສຍາມວາລາ ຈາກ Hanson and Mendius, Buddha's Brain: The Practical Neuroscience of Happiness, Love, and Wisdom, New Harbinger Publications, 2009.
- [83] Haykin, S. O. *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2009.
- [84] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 1026–1034.

- [85] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016).
- [86] Hinton, G. E. Neural networks for machine learning. Coursera, video lecture, 2012.
- [87] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.
- [88] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, Eds. IEEE Press, 2001.
- [89] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation* (1997).
- [90] Holst, A., and Jonasson, A. Classification of movement patterns in skiing. *Frontiers in Artificial Intelligence and Applications* 257 (2013), 115–124.
- [91] Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4, 2 (1991), 251–257.
- [92] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR* (2017).
- [93] Ioffe, S., and Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning* (2015), vol. 37, pp. 448–456.
- [94] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (2016).

- [95] Johnson, J., Karpathy, A., and Fei-Fei, L. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR* (2016).
- [96] Johnson, M. K., Foley, M. A., Suengas, A. G., and Raye, C. L. Phenomenal characteristics of memories for perceived and imagined autobiographical events. *Journal of Experimental Psychology: General* 117, 4 (1988), 371–376.
- [97] Jones, N. The learning machines. *Nature* 505 (2014).
- [98] Kar, S., and Roy, K. How far can virtual screening take us in drug discovery? *Expert Opinion on Drug Discovery* 8, 3 (2013), 245–261.
- [99] Karpathy, A., and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *CVPR* (2015).
- [100] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 1725–1732.
- [101] Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *CVPR* (2019).
- [102] Katanyukul, T. Ruminative reinforcement learning. *Journal of Computers* (2014).
- [103] Katanyukul, T., and Chong, E. K. P. Intelligent inventory control via ruminative reinforcement learning. *Journal of Applied Mathematics* (2014).
- [104] Katanyukul, T., Duff, W. S., and Chong, E. K. P. Approximate dynamic programming for an inventory problem: Empirical comparison. *Computers & Industrial Engineering* 60, 4 (2011), 719–743.
- [105] Katanyukul, T., Duff, W. S., and Chong, E. K. P. Intelligent inventory control: Is bootstrapping worth implementing? In *Intelligent Information Processing VI - 7th IFIP*

- TC 12 International Conference, Guilin, China (2012), Z. Shi, D. B. Leake, and Vadera, Eds., IFIP Advances in Information and Communication Technology, Springer, pp. 58–67.
- [106] Katanyukul, T., and Ponsawat, J. Customer analysis via video analytics. *Acta Polytechnica Hungarica* (2017).
- [107] Kelchtermans, P., Bittremieux, W., De Grave, K., Degroeve, S., Ramon, J., Laukens, K., Valkenborg, D., Barsnes, H., and Martens, L. Machine learning applications in proteomics research: How the past can boost the future. *Proteomics* 14, 4-5 (2014), 353–366.
- [108] Kelly, E. F. Consciousness is more than a product of brain activity, September 2016.
- [109] Kim, W., and Egan, J. M. The role of incretins in glucose homeostasis and diabetes treatment. *Pharmacological Reviews* 60, 4 (2008), 470–512.
- [110] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In *Proceeding of the 3rd International Conference for Learning Representations* (2015).
- [111] Kingma, D. P., and Welling, M. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.
- [112] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science* 220 (1983), 671–680.
- [113] Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Conference on Neural Information Processing Systems* (2012).
- [114] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature* 521 (May 2015), 436–444.
- [115] LeCun, Y., Cortes, C., and Burges, C. The mnist database. <http://yann.lecun.com/exdb/mnist/>, March 2015. retrieve on Mar 11th, 2015.

- [116] LeCun, Y., Matan, O., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jacket, L., and Baird, H. Handwritten zip code recognition with multilayer networks. In *10th International Conference on Pattern Recognition, Atlantic City, NJ (Volume:ii)* (1990), pp. 35–40.
- [117] Li, C., and Wand, M. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision* (2016), pp. 702–716.
- [118] Li, K., Du, N., and Zhang, A. Detecting ecg abnormalities via transductive transfer learning. In *ACM Conference on Bioinformatics, Computational Biology and Biomedicine, BCB 2012* (Orlando, FL, USA, 2012), pp. 210–217.
- [119] Li, Y., Zhong, W., Wang, D., Feng, Q., Liu, Z., Zhou, J., Jia, C., Hu, F., Zeng, J., Guo, Q., Fu, L., and Luo, M. Serotonin neurons in the dorsal raphe nucleus encode reward signals. *Nature Communications* (2016). 7:10503 doi: 10.1038/ncomms10503.
- [120] Lin, Y.-P., and Jung, T.-P. Improving eeg-based emotion classification using conditional transfer learning. *Frontiers in Human Neuroscience* 11, 334 (2017).
- [121] Linden, D. J. *The Accidental Mind: How Brain Evolution Has Given Us Love, Memory, Dreams, and God*. Belknap Press, 2008.
- [122] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR* (2015).
- [123] Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling* 55 (2015), 263–274.
- [124] Magazzeni, D., Py, F., Fox, M., Long, D., and Rajan, K. Policy learning for autonomous feature tracking. *Autonomous Robots* 37, 1 (2014), 47–69.

- [125] Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *International Conference on Learning Representations* (2016).
- [126] Malisiewicz, T., Gupta, A., and Efros, A. A. Ensemble of exemplar-svms for object detection and beyond. In *ICCV* (2011).
- [127] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S. K., Appuswamy, R., Taba, B., Amir, A., Flickner, M. D., Risk, W. P., Manohar, R., and Modha, D. S. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science 345*, 6197 (2014).
- [128] Minsky, M., and Papert, S. *Perceptrons: an introduction to computational geometry*. The MIT Press, 1969.
- [129] Mirza, M., and Osindero, S. Conditional generative adversarial nets. arXiv:1411.1784, 2014.
- [130] Mitchell, J. B. O. Machine learning methods in chemoinformatics. *WIREs Computational Molecular Science 4* (2014).
- [131] Mitchell, T. M. *Machine Learning*. McGraw-Hill, 1997.
- [132] Müller, R., Kornblith, S., and Hinton, G. When does label smoothing help? In *33rd Conference on Neural Information Processing Systems* (2019).
- [133] Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [134] Nakjai, P., and Katanyukul, T. Automatic hand sign recognition: Identify unusuality through latent cognizance. In *Artificial Neural Networks in Pattern Recognition - 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19-21, 2018, Proceedings* (2018), L. Pancioni, F. Schwenker, and E. Trentin, Eds., vol. 11081 of *Lecture Notes in Computer Science*, Springer, pp. 255–267.

- [135] Nakjai, P., and Katanyukul, T. Hand sign recognition for thai finger spelling: an application of convolution neural network. *J. Signal Process. Syst.* 91, 2 (2019), 131–146.
- [136] Nakjai, P., Ponsawat, J., and Katanyukul, T. Latent cognizance: what machine really learns. In *Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2019, Beijing, China, August 16-18, 2019* (2019), L. Ma and X. Huang, Eds., ACM, pp. 164–169.
- [137] Nash, S. G., and Sofer, A. *Linear and Nonlinear Programming*. McGraw-Hill, 1996.
- [138] NeuroBank. <http://neuronbank.org>.
- [139] Ng, A. Machine learning class. Coursera.org, 2013.
- [140] Ng, A., Katanforoosh, K., and Bensouda, Y. Deeplearning.ai: Sequnce models. Coursera.org, 2020.
- [141] Nguyen, D., and Widrow, B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of the International Joint Conference on Neural Networks* (1990), pp. 21–26.
- [142] Nowozin, S., Cseke, B., and Tomioka, R. F-gan: Training neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems* (2016), pp. 271–279.
- [143] of Neurological Disorders, N. I., and Stroke. Brain basic: Know your brain. [http://www.ninds.nih.gov/disorders/brain\\_basics/know\\_your\\_brain.htm](http://www.ninds.nih.gov/disorders/brain_basics/know_your_brain.htm), October 2012. NIH Publication No. 01 3440a.
- [144] Ortigosa, I., Lopez, R., and Garcia, J. A neural networks approach to residuary resistance of sailing yachts prediction. In *the International Conference on Marine Engineering MARINE* (2007).

- [145] Palsson, S., Agustsson, E., Timofte, R., and Gool, L. V. Generative adversarial style transfer networks for face aging. In *CVPR Workshop* (2018).
- [146] Pan, S. J., and Yang, Q. A. A survey on transfer learning. *IEEE Transaction on Knowledge and Data Engineering* 22 (2010), 345–1359.
- [147] Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. Gaugan: semantic image synthesis with spatially adaptive normalization. In *SIGGRAPH ’19: ACM SIGGRAPH 2019 Real-Time Live!* (2019).
- [148] Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. Semantic image synthesis with spatially-adaptive normalization. In *CVPR* (2019).
- [149] Parnia, S., Spearpoint, K., de Vos, G., Fenwick, P., Goldberg, D., Yang, J., Zhu, J., Baker, K., Killingback, H., McLean, P., Wood, M., Zafari, A. M., Dickert, N., Beisteiner, R., Sterz, F., Berger, M., Warlow, C., Bullock, S., Lovett, S., McPara, R. M., Marti-Navarette, S., Cushing, P., Wills, P., Harris, K., Sutton, J., Walmsley, A., Deakin, C. D., Little, P., Farber, M., Greyson, B., and Schoenfeld, E. R. Aware-awareness during resuscitation-a prospective study. *Resuscitation* 85, 12 (2014).
- [150] Parnia, S., Waller, D. G., Yeates, R., and Fenwick, P. Ruud van wees, vincent meyers, ingrid elfferich. *Resuscitation* 48, 2 (2001), 149–156.
- [151] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *ICML’2013* (2013).
- [152] Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. A review of novelty detection. *Signal Processing* 99 (2014), 215–249.
- [153] Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 4, 5 (1964), 1–17.

- [154] Poo, M. M. ibioseminar: Learning and memory: From synapse to perception. <http://www.ibiology.org>, 2010.
- [155] Prasad, V., and Mailankody, S. Research and development spending to bring a single cancer drug to market and revenues after approval. *JAMA Internal Medicine* 177, 11 (2017), 1569–1575.
- [156] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations* (2016). workshop track.
- [157] Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., and Pande, V. Massively multitask networks for drug discovery. In *International Conference on Machine Learning* (2015).
- [158] Rao, J., Bu, X., Xu, C. Z., Wang, L., and Yin, G. Vconf: a reinforcement learning approach to virtual machine autoconfiguration. In *Proceedings of the international conference autonomic computing, Barcelona, Spain, ACM 2009* (2009), pp. 137–146.
- [159] Rashid, T., and Anjum, A. 340 ways to use via character strengths. howpublished, July 2008.
- [160] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Conference of Computer Vision and Pattern Recognition (CVPR)* (2016).
- [161] Redmon, J., and Farhadi, A. Yolo9000: Better, faster, stronger. In *Conference of Computer Vision and Pattern Recognition (CVPR)* (2017).
- [162] Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. Learning what and where to draw. In *Advances in Neural Information Processing Systems* (2016), pp. 217–225.

- [163] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. In *International Conference on Machine Learning* (2016).
- [164] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster r-cnn: towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems* (2015).
- [165] Renuga Devi, T., Rabiyathul Basariya, A., and Kamaladevi, M. Fraud detection in card not present transactions based on behavioral pattern. *Theoretical and Applied Information Technology* 61, 3 (2014), 447–455.
- [166] Ricci, F., Rokach, L., and Shapira, B. *Introduction to Recommender Systems Handbook*. Springer, 2011, pp. 1–35.
- [167] Robinson, A. J., and Fallside, F. The utility driven dynamic error propagation network. Engineering Department, Cambridge University, 1987. Technical Report: CUED/F-INFENG/TR.1.
- [168] Romero, A., Ballas, N., Ebrahimi Kahou, S., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *ICLR’2015* (2015).
- [169] Ruano, A. E., Madureira, G., Barros, O., Khosravani, H. R., Ruano, M. G., and Ferreira, P. M. Seismic detection using support vector machines. *Neurocomputing* 135, 5 (2014), 273–283.
- [170] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.
- [171] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252.

- [172] Russell, S. J., and Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.
- [173] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems* (2016), pp. 2226–2234.
- [174] Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development* 3 (1959), 211–229.
- [175] Sánchez, H. P., Cano, G., and García-Rodríguez, J. Improving drug discovery using hybrid softcomputing methods. *Applied Soft Computing* 20 (2014), 119–126.
- [176] Sarikaya, R., Hinton, G. E., and Deoras, A. Application of deep belief networks for natural language understanding. *IEEE Transactions on Audio, Speech and Language Processing* 22, 4 (2014), 778–784.
- [177] Saxe, R. The brain v.s. the mind. [theagenda.tvo.org](http://theagenda.tvo.org), July 2012.
- [178] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.
- [179] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *CVPR* (2015).
- [180] Schuster, M., and Paliwal, K. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on* 45 (12 1997), 2673 – 2681.
- [181] Schölkopf, B., Smola, A., and Williamson, R. C. New support vector algorithms. *Neural Computation* 12, 5 (2000), 1207–1245.
- [182] Sheikh, H., Bovik, A., and De Veciana, G. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing* 14, 12 (2005), 2117–2128.

- [183] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR* (2016).
- [184] Shi, W., Caballero, J., Theis, L., Huszar, F., Aitken, A., Ledig, C., and Wang, Z. Is the deconvolution layer the same as a convolutional layer? arXiv preprint arXiv:1609.07009, 2016.
- [185] Shi, Y., Deb, D., and Jain, A. K. Warpgan: Automatic caricature generation. In *CVPR* (2019).
- [186] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015).
- [187] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR* (2015).
- [188] Springer, T., and Urban, K. Comparison of the em algorithm and alternatives. *Numerical Algorithms* 67 (2014), 335–364.
- [189] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.
- [190] Strang, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016.
- [191] Su, X., and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* (2009).
- [192] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *ICML* (2013).

- [193] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems* (2014), vol. 2, pp. 3104–3112.
- [194] Sutton, R. S., and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [195] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *CVPR* (2015).
- [196] Szegedy, C., Vanhoucke, V., Ioffe, S., and Shlens, J. Rethinking the inception architecture for computer vision. In *CVPR* (2016).
- [197] Sønderby, C. K., Caballero, J., and Theis, L. Amortised map inference for image super-resolution. In *International Conference on Learning Representations* (2017).
- [198] Tan, Z., Quek, C., and Cheng, P. Y. K. Stock trading with cycles: a financial application of anfis and reinforcement learning. *Expert System Applications* 38 (2011), 4741–4755.
- [199] Tucker, J. B., Greyson, B., Kelly, E. F., and Penberthy, J. K. Is there life after death? fifty years of research at uva history of the health sciences lecture. Public Lecture on February 22, 2017. the Division of Perceptual Studies, University of Virginia, February 2017. Youtube by UVA Medical Center Hour.
- [200] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499* (2016).
- [201] van den Oord Nal Kalchbrenner, A. Pixel rnn. In *ICML* (2016).
- [202] van Lommel, P., van Wees, R., Meyers, V., and Elfferich, I. Near-death experience in survivors of cardiac arrest: a prospective study in the netherlands. *The Lancet* 358, 9298 (December 2001), 2039–2045.

- [203] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceeding of Thirty-first Conference on Neural Information Processing Systems* (2017).
- [204] Vinogradov, S. Brain mind and behavior: Defining the mind. University of California Television (on youtube), October 2007. UCSF Mini Medical School for the Public. Show ID: 13029.
- [205] Vinyals, O., Toshev, A., Begio, S., and Erhan, D. Show and tell: A neural image caption generator. In *CVPR* (2015).
- [206] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition* (2001).
- [207] Wan, L., Zeiler, M., Zhang, S., LeCun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning* (2013).
- [208] Wang, N., and Yeung, D.-Y. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems* (2013), pp. 809–817.
- [209] Wang, S. I., and Manning, C. D. Fast dropout training. In *International Conference on Machine Learning* (2013), pp. 118–126.
- [210] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.
- [211] Wang, Z., Simoncelli, E., and Bovik, A. Multiscale structural similarity for image quality assessment. In *IEEE Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers* (2003), vol. 2, pp. 1398–1402.

- [212] Warde-Farley, D., Goodfellow, I. J., Courville, A., and Bengio, Y. An empirical analysis of dropout in piecewise linear networks. In *International Conference on Learning Representations* (2014).
- [213] Welling, M., and Kingma, D. P. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.
- [214] Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [215] Werbos, P. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE* 78, 10 (1990), 1550–1560.
- [216] Whitley, D. A genetic algorithm tutorial. *Statistics and Computing* 4, 2 (1994), 65–85.
- [217] Wikipedia. Wikipedia the free encyclopedia. internet.
- [218] Williams, R. J., and Zipser, D. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Back-propagation: Theory, Architectures and Applications*, Y. Chauvin and D. E. Rumelhart, Eds. Lawrence Erlbaum Publishers, 1995, pp. 433–486.
- [219] Wilson, D. R., and Martinez, T. R. The general inefficiency of batch training for gradient descent learning. *Neural Networks* 16, 10 (2003), 1429–1451.
- [220] Xiong, H. Y., Alipanahi, B., Lee, L. J., Bretscheneider, H., Merico, D., Yuen, R. K. C., Hua, Y., Gueroussov, S., Najafabadi, H. S., Hughes, T. R., Morris, Q., Barash, Y., Krainer, A. R., Jojic, N., Scherer, S. W., Blencowe, B. J., and Frey, B. J. The human splicing code reveals new insights into the genetic determinants of disease. *Science* 347, 6218 (2015).
- [221] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning* (Lille, France,

- 07–09 Jul 2015), F. Bach and D. Blei, Eds., vol. 37 of *Proceedings of Machine Learning Research*, PMLR, pp. 2048–2057.
- [222] Yu, Y., Zimmermann, R., Wang, Y., and Oria, V. Scalable content-based music retrieval using chord progression histogram and tree-structure lsh. *IEEE Transactions on Multimedia* 15, 8 (2013).
- [223] Zeiler, M. D., and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV* (2014).
- [224] Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. Deconvolutional network. In *CVPR* (2010).
- [225] Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision* (2016).
- [226] Zhu, W., Miao, J., Hu, J., and Qing, L. Vehicle detection in driving simulation using extreme learning machine. *Neurocomputing* 128 (2014), 160–165.
- [227] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning, 2019.

## บรรณนิภาษาไทย

- กฎของความน่าจะเป็นรวม, 50
- กฎของเบส์, 53, 444
- กฎผลคูณ, 53
- กฎผลรวม, 53
- กฎลูกโซ่ของความน่าจะเป็น, 53
- กราฟอาร์โอี, 207
- กลไกความใส่ใจ, 495, 497, 502
- กล่องขอบเขต, 237, 262
- กล่องสมอ, 440
- การกำจัดการระบุช้าช้อน, 236, 262
- การกำจัดความช้าช้อน, 237
- การทำหนดค่าน้ำหนักเริ่มต้น, 146, 171, 293
- วิธีเชเวียร์, 295
- การทำหนดค่าน้ำหนักเริ่มต้นด้วยการสุ่ม
- โปรแกรม, 170
- การทำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีเหยี่ยนวิดโดรว์
- โปรแกรม, 173
- การทำหนดค่าเริ่มต้น, 81
- การคำนวณเมทริกซ์, 33
- การคูณเมทริกซ์, 33
- คุณสมบัติ, 34
- การคูณแบบตัวต่อตัว, 33, 491
- การค้นหายา, 211
- การจัดกลุ่มข้อมูล, 418
- เค-มีนส์, 418
- การจัดการกับข้อมูลขาดหาย, 189
- การจัดถุง, 291
- การจับคู่ลักษณะสำคัญ, 433, 441
- การจำลองแบบชุดลำดับเป็นชุดลำดับ, 494
- การจำแนกกลุ่ม, 13, 18, 147
- การจำแนกค่าทวิภาค, 147
- การตกออก, 305
- การตรวจจับภาระตู้, 415
- วิธีหน้าต่างเลื่อน, 228
- โยโล', 415
- การตรวจจับภาระตู้ในภาพ, 414
- การตรวจหาภาระตู้, 260
- การตรวจจับภาระตู้ในภาพ, 440
- โยโล', 440
- การทดสอบโนลูชัน, 441
- การถ่ายโอนการเรียนรู้, 303
- การทดสอบนัยสำคัญ, 205
- การทำลายกราบรีน, 434, 442, 444, 445
- การทำลายกราบรีนทางเดียว, 434
- การทำซ้ำ, 175
- การทำอร์มอลาย, 149
- การทำอร์มอลายอินพุต, 176
- โปรแกรม, 177
- การทำหมู่เล็กเสริมอ่อนจริง, 435, 442
- การทำเรกุล่าเรซ์, 124, 163

- ค่าหน้าหนักเสื่อม, 124
- การทำเหมืองข้อมูล, 15
- การประมวลผลภาษาธรรมชาติ, 475, 502
- การจำแนกอารมณ์, 504
- การระบุหมวดคำ, 504
- การประสานการเรียนรู้, 291
- การประเมินผล
- กราฟการทำนายกับค่าเฉลี่ย, 180
- กราฟอาร์โอีซี, 207
- การทดสอบนัยสำคัญ, 205
- การระบุหมวดคำ, 188
- ความเที่ยงตรง, 188
- ความแม่นยำ, 187
- คะแนนเอฟ, 188
- ค่าความแม่นยำ, 124
- พื้นที่ใต้เส้นโค้ง, 208
- เมทริกซ์ความสัมสูน, 187
- การปรับละเอียด, 302
- การปรับส่วนค่าหน้าหนัก, 290
- การปรับเส้นโค้ง, 159
- ฟังก์ชันพหุนามระดับขั้นได ๆ, 162
- การปรับเส้นโค้งด้วยฟังก์ชันพหุนาม, 164
- การฝึก, 115, 137, 159
- การกำหนดค่าหน้าหนักเริ่มต้น, 146
- หมู่, 145
- ออนไลน์, 145
- โครงข่ายประสาทเทียม, 137
- การฝึกก่อน, 280, 302, 419
- การฝึกทีละหมู่เล็ก
- ขนาดของหมู่เล็ก, 287
- การฝึกแบบหมู่, 145
- การฝึกแบบหมู่เล็ก, 313
- การฝึกแบบออนไลน์, 145
- การพังทลายของภาวะ, 441
- การระบุหมวดคำ, 450, 475, 492, 502
- การระลึกกลับ, 188
- การรู้จำตัวเลขลายมือ, 9, 17, 193
- การรู้จำประเภทของวัตถุหลักในภาพ, 414
- การรู้จำรูปแบบ, 4, 12, 17
- การลูเช้า, 69, 81
- การสกัดลักษณะสำคัญ, 262
- การสลับเปลี่ยน, 30
- การสลายปัจจัย, 53, 80, 458
- การหยุดก่อนกำหนด, 151, 218
- การหาค่าดีที่สุด, 62, 80
- ภาวะคุกคัก, 105
- วิธีลงเกรเดียนต์, 66
- แบบมีข้อจำกัด, 73
- การหาค่าดีที่สุดแบบมีข้อจำกัด, 73
- แนวทางการลงโทษ, 74
- แนวทางการแปลงมุมมอง, 74
- การหาค่าลดตอน, 13, 18, 138
- การหาเกรเดียนต์เชิงเลข, 219
- การอันเดอร์ฟิต, 313

- การเขียนโปรแกรมเชิงเลข, 88  
 การเฉลี่ยตามประวัติ, 433  
 การเติมเต็ม, 364  
 การเติมเต็มด้วยศูนย์, 364  
 การเรียนรู้, 115, 159  
 การเรียนรู้การแนะนำสินค้า, 14  
 การเรียนรู้ของเครื่อง, 4, 6, 17  
 การเรียนรู้เชิงลึก, 16, 279, 305  
 การเรียนรู้แบบกึ่งมีผู้ช่วยสอน, 14, 423  
 การเรียนรู้แบบมีผู้ช่วยสอน, 13  
 การเรียนรู้แบบมีผู้สอน, 18  
 การเรียนรู้แบบเสริมกำลัง, 14  
 การเรียนรู้แบบไม่มีผู้สอน, 14, 18, 423  
 การเรียนหลักสูตร, 304  
 การเลือกแบบจำลอง, 120, 165  
 การเลื่อนของความแปรปรวนร่วมเกี่ยวกับภายใน, 300  
 การเลี้มเกรเดียนต์, 485, 501  
 การแจกส่วน, 474, 502  
 การแจกแจง  
 ปกติ, 61  
 เก้าส์เซียน, 61  
 การแจกแจงปกติ, 61  
 การแจกแจงเก้าส์เซียน, 61  
 การแจกแจงเอกรูป, 176  
 การแบ่งข้อมูล  
 โปรแกรม, 172  
 การแปลงเชิงเส้น, 40  
 การแผ่ทั่ว, 41  
 การแพร่กระจายย้อนกลับ, 141  
 โปรแกรม, 169  
 การแพร่กระจายย้อนกลับผ่านเวลา, 479, 501  
 การแยกแยกหมู่เล็ก, 433, 442  
 การใส่สัญญาณรบกวน, 435  
 ขนาดก้าว, 67  
 ขนาดก้าวย่าง, 366  
 ขนาดขัยบเลื่อน, 229, 262  
 ขนาดของหมู่เล็ก, 287  
 ขอบเขตของการแบ่ง, 246, 263  
 ขอบเขตตัดสินใจ, 260, 264  
 ขั้นตอนวิธีอีเม็ม, 459, 468  
 แบบจำลองมาร์คอฟช่อนเร็น, 459, 468  
 ขั้นตอนวิธีแอลฟ่า-ปีตา, 462  
 ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ, 462  
 ข้อจำกัดที่ทำงาน, 249  
 ข้อมูล  
 ตรวจสอบ, 152  
 ทดสอบ, 121  
 ฝึก, 121  
 ข้อมูลขนาดใหญ่  
 ปัญหา, 313  
 ข้อมูลชุดตรวจสอบ, 125  
 ข้อมูลตรวจสอบ, 152  
 ข้อมูลทดสอบ, 121, 159  
 ข้อมูลนำออก, 18, 114

- ข้อมูลนำเข้า, 18, 114
- ข้อมูลฝึก, 121, 159
- ข้อมูลทัต, 16
- ข้อมูลเชิงลำดับ, 449, 467
- ข้อมูลเชิงลำดับแบบคงที่, 451
- ครอบสาสติเดชั่น, 127
- ความซับซ้อนของแบบจำลอง, 123, 160
- ความน่าจะเป็น, 45, 47, 79
- ผลลัพธ์, 79
- เหตุการณ์, 79
- แบบมีเงื่อนไข, 52
- ไม่มีส่วนร่วมกัน, 50
- ความน่าจะเป็นก่อน, 80
- ความน่าจะเป็นของการปล่อย, 457, 467
- แบบจำลองมาร์คอฟซ่อนเร้น, 457, 467
- ความน่าจะเป็นของการเปลี่ยนสถานะ, 456, 467
- แบบจำลองมาร์คอฟซ่อนเร้น, 456, 467
- ความน่าจะเป็นภายนอก, 80
- ความน่าจะเป็นร่วม, 52
- ความน่าจะเป็นแบบมีเงื่อนไข, 52, 80
- ความผิดปกติทางบุคคลิกภาพแบบหลังตัวเอง, 481, 483
- ความลำเอียง, 154
- ความลำเอียงกับความแปรปรวน, 154
- ความลำเอียงสูง, 154
- ความลีกของโครงข่าย, 279
- ความเที่ยงตรง, 188
- ความเป็นอิสระเชิงเส้น, 37
- ความแปรปรวน, 51, 82, 154
- ความแปรปรวนร่วมเกี่ยว, 52
- ความแปรปรวนสูง, 154
- คอนโวโลชันก้าวยาว, 431
- คอนโวโลชันก้าวเศษ, 431, 441
- คอนโวโลชันสลับเปลี่ยน, 441
- คะแนนเอฟ, 188
- คำสาปของมิติ, 420
- คุณธรรม, 482
- คุณลักษณะ, 131
- คุณสมบัติความทั่วไป, 120, 159
- การหยุดก่อนกำหนด, 151
- ค่าความจำเพาะ, 208
- ค่าความน่าจะเป็นเริ่มต้น, 456, 467
- แบบจำลองมาร์คอฟซ่อนเร้น, 456, 467
- ค่าความเที่ยงตรง/ค่าการเรียกกลับ, 240
- ค่าความแม่นยำ, 124
- ค่าความໄ้, 208
- ค่าคาดหมาย, 51, 80
- ค่าทำให้น้อยที่สุด, 65, 80
- ห้องถิน, 66
- ค่าทำให้น้อยที่สุดห้องถิน, 66
- ค่าน้ำหนัก, 18, 131
- ค่าน้ำหนักเสื่อม, 124
- ค่าผิดปกติ, 210
- ค่าผิดพลาดของหัวชนิดอันดับสูงสุด, 391

- ค่าผิดพลาดชุดทดสอบ, 121  
 ค่าผิดพลาดชุดฝึก, 121  
 ค่าฟังก์ชันควรจะเป็นสูงสุด, 337  
 ค่าลอกการีทึมของฟังก์ชันควรจะเป็น, 338  
 ค่าเฉลี่ย, 116, 159  
 ค่าเฉลี่ยความผิดพลาดกำลังสอง, 124  
 ค่าเฉลี่ยค่าประมาณความเที่ยงตรง, 240, 263  
 จำนวนบวกจริง, 187  
 จำนวนบวกเท็จ, 187  
 จำนวนลบจริง, 187  
 จำนวนลบเท็จ, 187  
 จิต, 141, 435  
 จุดข้อมูล, 114, 159  
 จุดที่ดีที่สุด, 72  
 ฉลาก, 18  
 ชั้นคำนวน, 134, 136  
 ชั้นซ่อน, 136  
 ชั้นตีคอนโวลูชัน, 422  
 ชั้นเติร์รวม, 373  
 ชั้นเชื่อมต่อเติมที่, 362  
 ชั้นาเอาร์พุต, 138  
 ชีวิต, 435  
 ชุดข้อมูล  
 การจับตัวกับโปรตีน, 197  
 การรู้จำภาพตัวเลขลายมือเขียน, 193  
 ภาพเอ็กซเรย์เต้านม, 182  
 ยอด, 177  
 เออมนิสต์, 11, 193  
 ชุดข้อมูลโหลดสำเร็จ  
 MNIST, 329  
 ชุดมิติ, 360  
 ชุดลำดับมิติ, 360  
 ช่อง labore มอง, 491, 501  
 ซอฟต์แมกซ์, 111  
 ซัพพอร์ตเวกเตอร์, 247  
 ซัพพอร์ตเวกเตอร์แมชชีน, 244, 263, 264, 266,  
 270, 272, 275  
 ซัพพอร์ตเวกเตอร์, 264  
 ปัญหาปัจจุบัน  
 โปรแกรม, 268, 272  
 พังก์ชันเครอร์เนล, 264  
 พังก์ชันเครอร์เนลเกาส์เชียน, 264  
 พังก์ชันเครอร์เนลเชิงเส้น, 264  
 พังก์ชันเครอร์เนลเกาส์เชียน, 260  
 พังก์ชันเครอร์เนลเชิงเส้น, 260  
 ลูกเล่นเครอร์เนล, 264  
 เกาส์เชียนเครอร์เนล  
 โปรแกรม, 276  
 โปรแกรม, 275  
 ชิกมอยด์, 162  
 อนุพันธ์, 162  
 ดีเทอร์มิแนนต์, 37  
 ตัวประมาณค่าสาгал, 136  
 ตัวเข้ารหัสอัตแบบเปลี่ยนแปลง, 420

- ตัวเข้าอัตรหัส, 236
- ตัวแปรตัดสินใจ, 63
- ตัวแปรที่ลูกสั่งเกต, 455
- ตัวแปรที่สั่งเกตได้, 467
- ตัวแปรสุ่ม, 50, 79
- ตัวแปรสุ่มต่อเนื่อง, 60, 79
- ตัวแปรสุ่มวิยุต, 79
- ทบทวน
- การหาค่าดีที่สุด, 62
  - ความน่าจะเป็น, 45
  - พีชคณิตเชิงเส้น, 29
  - ทฤษฎีบทคารูซคุนท์เกอร์, 101
  - ทวิบตของความจำเอียงกับความแปรปรวน, 155
  - นอร์ม, 39, 433
  - นอร์มอิเลคตอินพุต, 150
  - บล็อกความจำ, 490
  - baugh กิ้งแน่นอน, 259
  - baugh แน่นอน, 259
  - ประสบการณ์เฉิดฉาย, 435
  - ปริภูมิ
    - การແພ່ວ້ວ, 41
    - ปริภูมิค่า, 32
    - ปริภูมิตัวอย่าง, 47, 79
    - ปริภูมิຍ່ອຍ, 41
    - ปริภูมิลักษณะสำคัญ, 244, 263
    - ปัญญา, 482
    - ปัญญาประดิษฐ์, 15
    - ปัญหาการระเบิดของเกรเดียนต์, 484, 501
    - ปัญหาการเลือนหายของเกรเดียนต์, 281, 305, 306, 484
    - ปัญหาค่าน้อยที่สุด, 63
    - ปัญหาค่ามากที่สุด, 63
    - ปัญหาที่สามารถแบ่งแยกได้เชิงเส้น, 132
    - ปัญหามอนต์ไฮอล, 58
    - ผลต่างเขต, 46
    - ผลbaugh พิດ, 58
    - ผลรวมเชิงเส้น, 37
    - พารามิเตอร์, 18
    - อภิมาน, 69
    - พิกเซลอาร์เอนเนอ, 420
    - พีชคณิตเชิงเส้น
    - ระบบสมการ, 34
    - พีชคณิตเชิงเส้น, 29
    - พื้นที่ใต้เส้นโค้ง, 208
    - พິງກໍ່ຫັນ
    - ซอฟຕ์ແມເກຊ, 111
    - ซิกมอยด์, 137
    - ພິງກໍ່ຫັນກະຮຕຸນ, 132
    - ພິງກໍ່ຫັນເຄື່ອງໝາຍອ່ອນ, 295
    - ເຮັດໄຟດລີເນີຍຣ, 305
    - ເຮຸ, 305
    - ພິງກໍ່ຫັນກະຮຕຸນເອາຕົບຸຕ, 138
    - ພິງກໍ່ຫັນซอຟຕິແມເກຊ, 148
    - ພິງກໍ່ຫັນຊົກມອຍດ, 147

- พังก์ชันเอกลักษณ์, 138  
 พังก์ชันการแจกแจง, 51, 79  
 พังก์ชันการแจกแจงสะสม, 60  
 พังก์ชันขั้นบันไดหนึ่งหน่วย, 105, 132  
 พังก์ชันควรจะเป็น, 80, 467  
 พังก์ชันความสูญเสีย, 65  
 พังก์ชันความหนาแน่นความน่าจะเป็น, 79  
 พังก์ชันค่าใช้จ่าย, 65  
 พังก์ชันจำกัดแข็ง, 132  
 โปรแกรม, 166  
 พังก์ชันจุดประสงค์, 63, 80  
 พังก์ชันซอฟต์แมกซ์, 111, 148  
 พังก์ชันซิกมอยด์, 89, 137  
 พังก์ชันบวกอ่อน, 84, 89  
 พังก์ชันพลังงาน, 65  
 พังก์ชันพหุนาม, 114, 159  
 พังก์ชันพหุนามระดับขั้นได ๆ, 162  
 พังก์ชันมวลความน่าจะเป็น, 51, 79  
 พังก์ชันลงโทษ, 75  
 พังก์ชันสูญเสีย, 80  
 ครอบสอนໂທຣີ, 147  
 ค่าเฉลี่ยค่าผิดพลาดกำลังสอง, 139  
 อนุพันธ์, 163  
 พังก์ชันสูญเสียครอบสอนໂທຣີ, 147  
 พังก์ชันเก้าส์ເຊີຍນສ, 89  
 พังก์ชันเครื่องหมายอ่อน, 295  
 พังก์ชันເຄອຣ່ານລ, 251, 264  
 พังก์ชันເຄອຣ່ານລເກາສ්ເຊີຍນ, 264  
 พังก์ชันເຄອຣ່ານລເຊີງເສັ້ນ, 264  
 พังก์ชันເຄອຣ່ານລເກາສ්ເຊີຍນ, 260, 264  
 พังก์ชันເຄອຣ່ານລເຊີງເສັ້ນ, 260, 264  
 พังก์ชันແບ່ງແຍກ, 236, 262  
 ພິລເຫວຼົມ, 363  
 ກາຮກິຈ  
     ກາຮຈຳແນກກຸລຸມ, 147  
     ກາຮຈຳແນກຄ່າທິກາຄ, 147  
     ກາຮຫາຄ່າຕດຄອຍ, 138  
     ກາຮກິຈຈຳລອງແບບໜຸດລຳດັບເປັນໜຸດລຳດັບ, 502  
     ກາຍາຮຮຽມໝາຕີ, 473  
     ມິຕີ, 31, 78, 131, 360  
     ມິຕີປະກຸມືກ່າ, 32  
     ຢູ່ເນື່ອນ, 46  
     ຮ້າສໜຶ່ງຮ້ອນ, 147  
     ຮະດັບຄ່າຂີດແບ່ງ, 206  
     ຮະບບສມກາຣ, 34  
     ຮະບບສມກາຣເຊີງເສັ້ນ, 82  
     ຮະບບແຕ່ງພັງອັຕໂນມັຕີ, 470  
     ຮະຍະທາງຢູ່ຄລືເດີຍນ, 39  
     ຮະເບີຍນ, 177  
     ຮາກທີ່ສອງຂອງຄ່າເນື່ອງຄວາມຜິດພາດກຳລັງສອງ, 124  
     ຮາຍຈານຜລ  
         ເວລານອ່ມອໄລ້ຈົດ, 152  
         ຮູ່ປະບົບ, 4, 17  
         ລາກຮານຈົ່ງພາຣາມີເຫວຼົມ, 75

- ลำดับชั้น, 31, 32, 360  
 ลำดับชั้นของความคิด, 325  
 ลูกเล่นเครอร์เนล, 258, 264  
 วิทยาการข้อมูล, 16  
 วิธีการประมาณความหนาแน่นแก่น, 238, 263, 265,  
     266  
 โปรแกรม, 266  
 วิธีการฝึกแบบปรับกշ์, 423  
 วิธีการลงโทษ, 75, 84  
 วิธีค่าพังก์ชันควรจะเป็นสูงสุด, 458  
 แบบจำลองมาร์คอฟซ่อนเร้น, 458  
 วิธีระจับค่าไม่มากสุดท่องถิน, 263  
 วิธีลงเกรเดียนต์, 66, 80  
 โปรแกรม, 268  
 วิธีลงเกรเดียนต์สโตแคสติก, 296  
 วิธีหน้าต่างเลื่อน, 226, 228, 260  
 วิธีเซเวียร์, 295  
 สติปัญญา, 76  
 สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส, 494, 502  
 สนามรับรู้, 372  
 สมการกำลังสอง, 44  
 สมการพหุนาม, 44  
 สมองมนุษย์, 118  
 สมัย, 145  
 สัญกรณ์จุดคู่, 30  
 สัดส่วนข้อมูลไม่สมดุล, 197, 419  
 สเกลเทา, 193  
 สเกลาร์, 29  
 ส่วนเติมเต็ม, 47  
 หน่วยคำนวน, 134  
 หน่วยช่อง, 136  
 หน่วยเวียนกลับมีประตุ, 492  
 หมู่เล็ก, 284, 305  
 อภิมานพารามิเตอร์, 69, 120, 135  
 อภิรนาบ, 244, 263  
 อัตตา, 481  
 อัตราการตรวจจับได้, 208  
 อัตราสัญญาณหลอก, 208  
 อัตราเรียนรู้, 145  
 อันเดอร์พิต, 154  
 อินพุต, 18, 114  
 คุณลักษณะ, 131  
 มิติ, 131  
 อินเตอร์เซกชัน, 46  
 อิมเมจเนต, 391  
 อเล็กซ์เน็ต, 391  
 เกร็ด  
     การค้นหายา, 211  
 จิต, 141  
 จิต สมอง และชีวิต, 435  
 มะเร็งและยารักษา, 6  
 สติปัญญา, 76  
 สมองมนุษย์, 118  
 เชลล์ประสาท, 128

- เมตตา, 481
- เกร็ดความรู้, 6, 76, 118, 128, 141, 211, 435, 481
- เขตข้อมูล, 177
- เค-มีนส์, 418
- เคอร์เนล, 363
- เงื่อนไข
- ทำงาน, 102
  - ไม่ทำงาน, 102
- เงื่อนไขการจบ, 69
- เขต, 46
- ยูเนียน, 46
- อินเตอร์เซกชัน, 46
- เขตปอย, 46
- เขตว่าง, 46
- เซลล์ความจำ, 490
- เซลล์ประสาท, 128
- เทคนิคกล่องสมอ, 417
- เทนเซอร์, 32
- เนสเตอรอฟโมเมนตัม, 297
- เพอร์เซปตรอน, 130
- เพอร์เซปตรอนหลายชั้น, 130
- เมตตา, 481, 482
- เมทริกซ์, 30
- การคำนวณ, 33
  - การคูณ, 33
  - การบวก, 33
- ดีเทอร์มิเนนต์, 37
- เมทริกซ์การเปลี่ยนสถานะ, 456
- แบบจำลองมาร์คอฟช่อนเร็ว, 456
- เมทริกซ์ความสับสน, 187
- เรคติไฟร์ลิเนียร์, 283
- เรลู, 267, 283, 307, 313, 432
- โปรแกรม, 268
- เรลูรัว, 433
- เรเดียวเบซิส, 162
- อนุพันธ์, 162
- เวกเตอร์, 29
- ตั้งฉาก, 40
- เวกเตอร์ตั้งฉาก, 40
- เวกเตอร์หนึ่งหน่วย, 39
- เวคตอเรซชัน, 135
- เวลานอร์มอลайซ์ด์, 152
- เส้นโค้งเรียนรู้, 154, 155
- เหตุการณ์, 47
- เอ็จโจจี, 232
- เอนโโทรปี, 82
- เอมนิสต์, 11, 18, 193
- เอาต์พุต, 18, 114
- เอาต์พุตจริง, 116
- แบบความเชื่อมั่น, 176
- แบบนอร์ม, 300, 305, 432
- โครงข่ายคอนโวลูชัน, 302
- โปรแกรม, 352

- แบบจำลอง, 12, 18, 113, 114  
 การทำนาย, 12  
 การอนุมาน, 12  
 การแปลงค่า, 12  
 แบบจำลองความจำระยะสั้นที่ยาว, 492  
 ช่องแอบมอง, 491, 501  
 ปลื้อกความจำ, 490  
 เชลล์, 490  
 แบบจำลองความระยะสั้นที่ยาว, 490, 501  
 แบบจำลองความหนาแน่นผสม, 337  
 แบบจำลองปริภูมิสถานะ, 455  
 แบบจำลองมาร์คอฟ, 451, 467  
 แบบจำลองมาร์คอฟซ่อนเร้น, 456, 467  
 ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ, 462  
 ขั้นตอนวิธีเอลฟ่า-บีตา, 462  
 พงกชันการปล่อย  
 อเนกนามวิยุต, 461  
 เกสเชียน, 461  
 แบบจำลองสร้างกำเนิด, 236, 262  
 แบบจำลองแบ่งแยก, 236, 262  
 แผนที่ความร้อน, 237  
 แผนที่ลักษณะสำคัญ, 372  
 แผนภาพคลื่นลำดับ, 477, 501  
 โครงการวิเคราะห์พฤติกรรมลูกค้า, 223  
 โครงข่ายก่อกำเนิด, 423, 440  
 โครงข่ายคอนโวลูชัน, 359, 403  
 โครงข่ายปรปักษ์เชิงสร้าง, 440  
 การจับคู่ลักษณะสำคัญ, 433, 441  
 การพังทลายของภาวะ, 430, 441  
 การแยกแยกหมู่เล็ก, 433  
 ปริภูมิซ่อนเร้น, 441  
 พีชคณิตเวกเตอร์, 441  
 ลักษณะซ่อนเร้น, 428, 441  
 โครงข่ายก่อกำเนิด, 423, 440  
 โครงข่ายแบ่งแยก, 423, 440  
 โครงข่ายปรปักษ์เชิงสร้างกำเนิด, 236  
 โครงข่ายปรปักษ์เชิงสร้างแบบมีเงื่อนไข, 426  
 โครงข่ายประสาทเทียม, 130  
 การทำงานอัตโนมัติ, 149  
 การฝึก, 137, 167  
 ตรรกะເອັກຊ່ອງຮ, 165  
 โปรแกรม, 166  
 การตกอัก, 332–334  
 คลาส, 312  
 ชั้นสัญญาณรบกวน, 337  
 ไฟฟอร์ซ, 315, 319  
 โครงข่ายประสาทเวียนกลับ, 476, 501  
 สถานะซ่อน, 477  
 โครงข่ายประสาทเวียนกลับสองทาง, 486, 501  
 โครงข่ายแบ่งแยก, 423, 440  
 โครงข่ายเพร์เซป捷ชันไปข้างหน้า, 136  
 โมเมนตัม, 297  
 โยໂລ໌, 415, 440  
 โรคหลงตัวเอง, 481, 483

- ខ្លួន, 134  
ខ្លួនទូន, 136  
ខោវេរិចិត, 122, 160  
ការធានរៀលាអីរីថី, 124  
ឲប៉ែស, 131  
ឲពុទុរីខ, 319  
ឲយាករណ៍, 474, 502  
ឲ.ឲ.ឲ.ឲ., 338, 449, 453  
ឲឈូយុ, 237, 263, 417  
ឲមេរីបអិកពេនជនទំនើត, 162  
ឧនុផន្ទី, 162

## គរចនីភាសាហោងករម

- accuracy, 124  
activation  
leaky relu, 295  
PRelu, 295  
relu, 283  
sigmoid, 137  
softsign, 295  
tanh, 138  
activation function, 132  
active constraint, 249  
adam, 280  
adversarial training, 423  
AlexNet, 391  
alpha-beta algorithm, 462  
anchor box, 417, 440  
Approaches to exact calculation issues  
approximation, 453  
constraining, 453  
Area Under Curve, 208  
artificial intelligence, 15  
artificial neural network  
code, 166  
class, 312  
dropout, 332–334  
noise layer, 337  
pytorch, 315, 319  
normalization, 149  
train, 137, 167  
XOR, 165  
Artificial Neural Network គម្រោង ANN, 130  
attention, 280  
attention mechanism, 495, 497, 502  
AUC, 208  
Autoencoder, 236  
backpropagation, 141  
code, 169  
backpropagation through time, 479, 501  
bagging, 291  
batch norm, 300, 305, 432  
code, 352  
convoltion neural network, 302  
batch normalization, 280  
batch training, 145  
Bayes’ rule, 53, 444  
Bayes’ theorem, 53, 444  
bias, 131  
bias (model behavior), 154  
bias-variance, 154  
Bias/Variance Dilemma, 155

- bidirectional recurrent neural networks, 486, 501
- big data, 16
- binary classification, 147
- Biological Neurons, 128
- bounding box, 237, 262
- built-in datasets
- MNIST, 329
- chain rule of probability, 53
- classification, 13, 18, 147
- clustering, 418
- K-means, 418
- CNN, 280, 359, 403
- colon notation, 30
- compassion, 481, 482
- complement, 47
- Conditional Generative Adversarial Networks, 426
- conditional probability, 52, 80
- confidence intervals, 176
- confusion matrix, 187
- constrained optimization, 73, 74, 84
- penalty, 74
  - projection, 74
- constraint
- active, 102
  - inactive, 102
- continuous random variable, 60, 79
- convergence, 69, 81
- Convolutional Neural Network, 359, 403
- cost function, 65
- covariance, 52
- cross entropy
- code, 197
- cross entropy loss, 147
- cross-validation, 127
- cumulative distribution function, 60
- curriculum learning, 304
- curses of dimensionality, 420
- curve fitting, 159
- degree-M polynomial, 162
- customer behavior analytic project, 223
- data
- test, 121
  - training, 121
  - validation, 152
- data mining, 15
- data science, 16
- data separation
- code, 172
- datapoint, 114, 159
- dataset
- mammography, 182
  - MNIST, 11, 193

- protein binding, 197  
yacht, 177  
DCGAN, 431  
decision boundary, 260, 264  
decision variable, 63  
deconvolution, 441  
deconvolution layer, 422  
deep learning, 16, 279, 305  
Degree-M polynomial, 162  
detection rate, 208  
determinant, 37  
dimension, 31, 78, 131, 151, 360  
discrete random variable, 79  
discriminant function, 236, 262  
discriminative model, 236, 262  
discriminator, 423, 440  
distribution  
    Gaussian, 61  
    normal, 61  
distribution function, 51, 79  
drop out, 305  
    break co-adaptation, 290  
    learn features more thoroughly, 290  
    robustness, 290  
dropout, 280  
drug discovery, 211  
dual problem, 105  
early stopping, 151, 218  
Eigenvalues, 43  
Eigenvectors, 43  
element-wise product, 33, 491  
emission probabilities, 457, 467  
    HMM, 457, 467  
empty set, 46  
encoder-decoder architecture, 494, 502  
energy function, 65  
ensemble learning, 291  
entropy, 82  
epoch, 145  
Euclidean distance, 39  
evaluation, 121  
    accuracy, 187  
    accuracy, 124  
    Area Under Curve, 208  
    AUC, 208  
    confusion matrix, 187  
    F-score, 188  
    MSE, 124  
    overfitting, 122  
    precision, 188  
    predict-groudtruth plot, 180  
    recall, 188  
    Receiver Operating Characteristic, 207  
    RMSE, 124

- ROC, 207
- significance test, 205
- test error, 121
- event, 47
- expectation, 51, 80
- expectation-maximization algorithm, 459, 468  
    HMM, 459, 468
- expected value, 51
- exploding gradient problem, 484, 501
- F-score, 188
- false alarm rate, 208
- false negative, 187
- false positive, 58, 187
- feature, 131
- feature extraction, 262
- feature map, 372
- feature matching, 433, 441
- feature space, 244, 263
- feedforward network, 136
- field, 177
- filter, 363
- fine tuning, 302
- forward-backward algorithm, 462
- fractionally-strided convolution, 431, 441
- fully connected layer, 362
- function
- sigmoid, 137
- softmax, 111
- GAN, 280, 422, 440  
    Conditional GAN, 426  
    DCGAN, 431  
    discriminator, 423  
    feature matching, 433  
    generator, 423  
    latent representation, 428  
    latent space, 428  
    minibatch discrimination, 433  
    mode collapse, 430  
    representation space, 428
- Gaussian distribution, 61
- gaussian function, 89
- Gaussian kernel, 260
- gaussian kernel, 264
- generalization, 120, 159  
    early stopping, 151
- Generative Adversarial Network, 236, 280,  
    422
- Generative Adversarial Networks, 440  
    discriminator, 440  
    feature matching, 441  
    generator, 440  
    latent representation, 441  
    latent space, 441  
    mode collapse, 441

- vector arithmetic, 441
- generative model, 236, 262
- generator, 423, 440
- gradient clipping, 485, 501
- gradient descend method
  - code, 268
- gradient descent algorithm, 66, 80
- gray scale, 193
- ground truth, 116, 159
- gru, 492
- hand-written digit recognition, 9, 193
- handling missing data, 189
- handwritten digit recognition, 17
- hard limit
  - code, 166
- hard limit function, 132
- heat map, 237
- hidden layer, 136
- Hidden Markov model, 456
- hidden node, 136
- hidden unit, 136
- hierarchy of abstraction, 325
- high bias, 154
- high variance, 154
- Histogram of Oriented Gradients, 232
- historical averaging, 433
- HMM, 467
- emission
  - discrete multinomial variables, 461
  - gaussian variables, 461
- forward-backward algorithm, 462
- alpha-beta algorithm, 462
- HOG, 232
- Human Brain, 118
- hyper-parameter, 69, 120, 135
- hyperbolic tangent, 138
- hyperplane, 244, 263
- i.i.d., 338, 449, 453
- image classification, 414
- ImageNet, 391
- independent and identically distributed, 338, 449
- initial probabilities, 456, 467
- HMM, 456, 467
- initialization, 81
- input, 18, 114
  - dimension, 131
  - feature, 131
- input normalization, 176
  - code, 177
- Intelligence, 76
- internal covariance shift, 300
- intersection, 46
- IoU, 237, 263, 417

- joint probability, 52  
K-means, 418  
Karush-Kuhn-Tucker theorem, 101  
KDE, 265, 266  
kernel, 251, 363  
Kernel Density Estimation, 238, 265, 266  
kernel density estimation, 263  
    code, 266  
kernel function, 264  
    gaussian kernel, 264  
    linear kernel, 264  
kernel tricks, 258, 264  
KKT, 101  
L1 norm, 433  
label, 18  
label smoothing, 434, 442, 444, 445  
Lagrange parameter, 75  
large dataset  
    issues, 313  
latent space, 425  
latent state, 455, 467  
    HMM, 455, 467  
latent variable  
    HMM, 455  
latent variable, 428, 455, 467  
    GAN, 428  
    HMM, 467  
law of total probability, 50  
layer, 134, 136  
leaky relu, 295, 433  
learning, 115, 159  
    long-term synaptic potentiation, 143  
    LTP, 143  
Learning Curve, 155  
learning curve, 154  
learning rate, 145  
life, 435  
likelihood function, 80, 467  
linear algebra, 29  
    linear equations, 34  
linear combination, 37  
linear equations, 34, 82  
linear kernel, 260, 264  
linear transformation, 40  
linearly independence, 37  
linearly separable problem, 132  
local minimizer, 66  
local optimum, 176  
log likelihood, 338  
long-term synaptic potentiation, 143  
loss  
    derivative, 163  
    mean square error, 139  
loss function, 65, 80

- loving kindness, 481
- lstm, 490, 492, 501
- block, 490
  - cell, 490
  - peephole, 491, 501
- lstm block, 490
- lstm cell, 490
- LTP, 143
- machine learning, 4, 6, 17
  - supervised learning, 13
    - classification, 13
    - regression, 13
- mAP, 240, 263
- margin of separation, 246, 263
- marginalization, 53, 80, 458
- Markov model, 451, 467
- matrix, 30
  - addition, 33
  - determinant, 37
  - multiplication, 33
  - operation, 33
- matrix operation, 33
- matrix product, 33
  - useful properties, 34
- maximization problem, 63
- maximum likelihood, 337, 458
- HMM, 458
- mean Average Precision, 240, 263
- mean square error, 124
- meta-parameter, 69, 120, 135
- Mind, 141
- mind, 435
- minibatch, 280, 284, 287, 305, 313
  - batch size, 287
- minibatch discrimination, 433, 442
- minimization problem, 63
- minimizer, 65, 80
  - local, 66
- missing data, 182
- mixture density model, 337
- MLP, 130
- mlp
  - code, 166
  - train, 167
  - code, 169
- MNIST, 11, 18, 193
- mode collapse, 441
- model, 12, 18, 113, 114
  - inference, 12
  - mapping, 12
  - prediction, 12
- model complexity, 123, 160
- model selection, 120, 165
- momentum, 297

- Monty Hall Problem, 58
- MSE, 124
- multi-class classification, 147
- multi-layer perceptron, 130
- multiclass classification, 147
- music generation, 470
- Named-Entity Recognition, 450
- Narcissistic Personality Disorder, 481, 483
- natural language, 473
- applications, 475
- issues, 474
- Natural Language Processing, 475, 502
- NDE, 435
- near death experience, 438
- Nesterov momentum, 297
- network depth, 279
- NLP, 475, 502
- POS tagging, 504
- sentiment analysis, 504
- node, 134
- noise addition, 435
- non-local-maximum suppression, 263
- norm, 39, 433
- normal distribution, 61
- normalization, 149
- normalized input, 150
- normalized time, 152
- numerical gradient, 219
- code, 221
- numerical programming, 88
- Object Detection, 415
- YOLO, 415
- object detection, 260, 414, 440
- sliding window, 228
- YOLO, 440
- objective function, 63, 80
- observable variable, 467
- observed variable, 455
- one-hot coding, 147
- one-sided label smoothing, 434
- online training, 145
- optimal point, 72
- optimization, 62, 80
- constrained, 73, 74
- duality, 105
- dual problem, 105
- primal problem, 105
- gradient descent algorithm, 66
- orthogonal vector, 40
- outliers, 210
- output, 18, 114
- output activation function, 138
- identity function, 138
- sigmoid function, 147

- softmax function, 148
- output layer, 138
- overfitting, 122, 160
- regularization, 124
- padding, 364
- parameter, 18
  - hyper-, 69
  - meta-, 69
- parameter sharing, 454
- parsing, 474, 502
- Part-Of-Speech Tagging, 450, 475, 492, 502
- pattern, 4, 17
- pattern recognition, 4, 12, 17
- peephole, 491, 501
- penalty function, 75
- penalty method, 75, 84
- perceptron, 130
- PixelRNN, 420
- pmf, 51
- polynomial, 44
  - code, 164
- train
  - code, 164
- polynomial curve fitting, 164
- polynomial function, 114, 159
- pooling layer, 373
- positive definite, 259
- positive semidefinite, 259
- posterior distribution, 80
- pre-training, 280, 302, 419
- precision, 188
- precision/recall, 240
- PReLU, 295
- primal problem, 105
- prior probability, 80
- probability, 45, 47, 79
  - conditional, 52
  - disjoint, 50
  - joint, 52
  - mass function, 51
  - outcome, 79
  - sum rule, 50
- probability density function, 79
- probability mass function, 51, 79
- probablity
- event, 79
- product rule, 53
- pytorch, 319
- quadratic, 44
- quote
  - achieving the goal, 279
  - adaptation, 359
  - break down the problem, 29
  - build up knowledge, 113

- courage, 162, 394  
explore, 19  
failure, 223  
gradient descent, 73  
insignificance, 218  
kindness, 443, 482, 484  
language, 473  
learning, 413  
life, 469  
living life, 440  
moderation, 144, 212  
native language, 503  
pattern, 4  
pre-condition, 436, 437, 481  
principles, 82, 449  
regulation, 9  
science, 306  
truth, 437  
value, 265  
wisdom and compassion, 78  
radial basis, 162  
derivative, 162  
random variable, 50, 79  
rank, 31, 360  
recall, 188  
Receiver Operating Characteristic, 207  
receptive field, 372  
recommendation learning, 14  
record, 177  
rectified linear, 283, 305  
Recurrent Neural Network, 476, 501  
hidden state, 477  
redundancy removal, 236, 237, 262  
regression, 13, 18, 138  
regularization, 124, 163  
weight decay, 124  
reincarnation, 437  
reinforcement learning, 14  
ReLU, 432  
relu, 267, 283, 305, 307, 313  
code, 268  
repeat, 175  
report  
normalized time, 152  
representation space, 425  
revise  
linear algebra, 29  
optimization, 62  
probability, 45  
RMSE, 124  
RNN, 280  
ROC, 207  
root mean square error, 124  
sample space, 47, 79

- scalar, 29
- self, 481
- semi-supervised Learning, 423
- semi-supervised learning, 14
- sensitivity, 208
- sequence-to-sequence architecture, 494, 502
- sequence-to-sequence modeling, 494, 502
- sequential data, 449, 467
- stationary, 451
- set, 46
- difference, 46
  - empty, 46
  - intersection, 46
  - union, 46
- set difference, 46
- side story, 6, 76, 118, 128, 141, 211, 435, 481
- CML and Cure, 6
- compassion, 481
- drug discovery, 211
- Human Brain, 118
- Intelligence, 76
- Mind, 141
- Mind, Brain, and Life, 435
- Neurons, 128
- sigmoid, 137, 162
- derivative, 162
- sigmoid function, 89
- significance test, 205
- sliding window, 226, 228, 260
- soft optimization, 176
- softmax, 111, 148
- code, 197
- softplus function, 84, 89
- softsign, 295
- space
- span, 41
  - span, 41
- specificity, 208
- state space model, 455
- stationary sequential data, 451
- step size, 67
- stochastic gradient descent, 296
- stride, 229, 262, 366
- strided convolution, 431
- subset, 46
- subspace, 41
- sum rule, 53
- supervised learning, 13, 18
- Support Vector Machine, 244, 263, 266, 270, 272, 275
- support vector machine, 264
- code, 275
  - gaussian kernel

- code, 276
- kernel function, 264
  - gaussian kernel, 264
  - linear kernel, 264
- kernel tricks, 264
- primal
  - code, 268, 272
- support vectors, 264
- support vectors, 247
- SVM, 244, 266, 270, 272, 275
  - Gaussian kernel, 260
  - linear kernel, 260
- symmetry breaking, 293
- syntax, 474, 502
- tanh, 138, 162
  - derivative, 162
- task
  - binary classification, 147
  - multiclass classification, 147
  - regression, 138
- tensor, 32
- terminating condition, 69
- test data, 121, 159
- test error, 121
- threshold, 206
- thresholding, 420
- top-5 error rate, 391
- torchvision, 329
- train, 137
  - artificial neural network, 137
  - backpropagation, 141
- training, 115, 159
  - batch, 145
  - online, 145
- weight initialization, 146
- training data, 121, 159
- training error, 121
- transfer learning, 303
- transition matrix, 456
  - HMM, 456
- transition probabilities, 456, 467
  - HMM, 456, 467
- transpose, 30
- transposed convolution, 441
- true negative, 187
- true positive, 187
- unbalanced data, 197, 419
- underfit, 154
- underfitting, 313
- unfolding diagram, 477, 501
- uniform distribution, 176
- union, 46
- unit, 134
- unit step function, 105, 132

- unit vector, 39
- universal approximator, 136
- unsupervised learning, 14, 18, 423
- validation, 125
- validation data, 152
- vanishing gradient problem, 281, 305, 306, 484
- variance, 51, 82
- variance (model behavior), 154
- Variational Autoencoder, 420
- vector, 29
  - orthogonal, 40
- vector space, 32
- vectorization, 135
- virtual minibatch, 435, 442
- virtues, 482
- weight, 18
- weight decay, 124
- weight initialization, 146, 171, 293
  - code, 170
  - Nguyen-Widrow
    - code, 173
  - random
    - code, 170
  - Xavier, 295
- weight scaling, 290
- weights, 131
- wisdom, 482
- words of wisdom
  - Al-Kindi, 437
  - Albert Einstein, 265
  - Amelia Earhart, 443
  - Chuck Palahniuk, 4
  - Confucius, 413
  - H. G. Wells, 359
  - Isaac Asimov, 437
  - Isaac Newton, 113
  - Jawaharlal Nehru, 449
  - Johann Wolfgang von Goethe, 279
  - Lao Tzu, 482, 484
  - Leo Tolstoy, 436, 481
  - Marcus Du Sautoy, 306
  - Marcus Tullius Cicero, 144
  - Mark Twain, 19
  - Mohandas Karamchand Gandhi, 218
  - Morihei Ueshiba, 223
  - Mother Teresa, 469
  - Nelson Mandela, 162, 503
  - Paracelsus, 212
  - Ralph Waldo Emerson, 82
  - René Descartes, 29
  - Robina Courtin, 78
  - Sean B. Carroll, 9
  - Tenzin Gyatso, 440

Winston Churchill, 394

Yuval Noah Harari, 473

Xavier initialization, 295

YOLO, 415, 440

zero-padding, 364

## แหล่งที่มาของภาพ

- ภาพปก ผู้เขียนจัดทำขึ้นเอง รวมถึงการวาดภาพนักประดาน้ำ และภาพทะเล.
- ภาพปกใน ผู้เขียนจัดทำขึ้นเอง จากการแต่งภาพ โดยภาพที่นำมาแต่ง ประกอบด้วย ภาพถ่ายห้องฟ้า ตอนกลางคืน และภาพถ่ายต้นเฟิร์น ซึ่งผู้เขียนถ่ายเองทั้งสองภาพ ในส่วนตัวที่นำมาตกแต่งห้องฟ้า ก็ ตัดมาจากส่วนหนึ่งของเพลงที่ผู้เขียนแต่งขึ้นเอง.
- ภาพต่าง ๆ ในเล่ม ผู้เขียนจัดทำขึ้นเอง ยกเว้นแต่จะระบุเป็นอื่น.