

การผจญภัยของการเรียนรู้ของเครื่อง
ในโลกการรู้จำรูปแบบ

The Adventures of Machine Learning
in The World of Pattern Recognition

ธีช พงศ์ กตัญญูกุล

26 มกราคม พ.ศ. 2565

ຫົວໜ້າ

คำนำ

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

—พระราชาดำรัสของพระบาทสมเด็จพระเจ้าอยู่หัวรัชกาลที่๙

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

ถึงแม้เป้าหมายนี้ คือ การอภิปรายศาสตร์และศิลป์ในงานการรู้จำรูปแบบ แต่จากความก้าวหน้า แล้วลึก ทั้งในทางทฤษฎีและการประยุกต์ใช้ที่กว้างขวาง รวมถึงความก้าวหน้าที่เดิมโดยอย่างรวดเร็ว และต่อเนื่องของศาสตร์ การจะครอบคลุมเนื้อหาทั้งหมดนั้นเป็นไปไม่ได้เลย. ดังนั้น เช่นเดียวกับหนังสือการเรียนรู้ของเครื่องเบื้องต้น หนังสือเล่มนี้จัดทำเนื้อหา โดยยึดแนวคิดในการวางแผนเพียงเป็นจุดเริ่มต้น ที่จะช่วยให้ผู้อ่านได้พอเข้าใจภาพรวม และเห็นคุณค่าของศาสตร์นี้ รวมไปจนถึงอำนวยความสะดวกในการติดตามความคืบหน้าในกรณีที่ผู้อ่านต้องการแหล่งค้นคว้าเพิ่มเติม. ผู้เขียนหวังว่า ผู้อ่านจะได้เข้าใจพื้นฐานและเหตุผลเบื้องหลัง เข้าใจทฤษฎีที่สำคัญ ซาบซึ้งในความเพียรพยายามและความคิดสร้างสรรค์เบื้องหลังการพัฒนามองเห็นการประยุกต์ใช้ ได้ทดลองลงมือปฏิบัติ ได้ฝึกคิด ทบทวน ตั้งข้อสังเกต อภิปรายลักษณะเด่น ประเด็นสำคัญ ข้อดีข้อเสีย โอกาสและความเสี่ยง ความเกี่ยวข้อง ความเชื่อมโยงของกลไกและแนวคิดต่าง ๆ มองเห็นความท้าทาย ตลอดจนรู้สึกเพลิดเพลิน เกิดแรงบันดาลใจที่จะศึกษา และทำงานกับศาสตร์ด้านนี้ต่อไป

แนวคิดในการจัดทำเนื้อหานี้ ได้ออกแบบจากประสบการณ์การสอน และเพื่อใช้ประกอบการเรียน วิชาโครงข่ายภาษาไทย วิชาการเรียนรู้ของเครื่อง และวิชาการรู้จำรูปแบบ ระดับปริญญาตรีและบัณฑิตศึกษา ของ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น. เนื้อหาจึงถูกเรียบเรียงจากสมมติฐานว่า ผู้อ่านมีพื้นฐานคณิตศาสตร์ ซึ่งรวมถึงพีชคณิตเชิงเส้น ความน่าจะเป็น และแคลคูลัส รวมถึง มีประสบการณ์การเรียนโปรแกรมคอมพิวเตอร์มาบ้าง. อย่างไรก็ตาม ผู้เขียนหวังว่า เนื้อหานี้จะเป็นประโยชน์กับผู้อ่านที่สนใจทั่วไปเช่นกัน

รูปแบบการเขียน

เนื่องจากเนื้อหาของศาสตร์การเรียนรู้ของเครื่องและการรู้จำรูปแบบ เกี่ยวข้องกับคณิตศาสตร์และโปรแกรมคอมพิวเตอร์ มีคำศัพท์เฉพาะจำนวนมาก อาทิ ประวัติการ แนวคิด และแนวทางปฏิบัติที่หลาย ๆ ครั้งถูกพัฒนาขึ้นจากความคิดสร้างสรรค์และประสบการณ์ ซึ่งบางโอกาส ผู้อ่านอาจพบว่าขัดกับสัญชาตญาณ ประเด็นต่าง ๆ เหล่านี้ อาจทำให้ผู้อ่านเข้าใจข้อความที่ผู้เขียนพยายามสื่อสารได้ยาก หรือเข้าใจไม่ถูกต้อง. ดังนั้นเพื่อช่วยให้เนื้อหาอ่านง่ายขึ้น ข้อความที่ต้องการสื่อสารชัดเจนไม่คลุมเครือ ลดความกำกวມ รวมถึงการใช้ถ้อยคำไม่ฟุ่มเฟือยเย็นเย้อ หรือไม่มีการใช้ย่อหน้าที่มากจนเกินไป ผู้เขียนใช้มหัพภาค เพื่อปั่งบอกการจบประโยค และใช้จุลภาค เพื่อค้นรายการต่าง ๆ รวมถึงบางครั้งอาจใช้ฟอนต์ตัวอังกฤษ เพื่อเน้นคำศัพท์หรือกลุ่มคำให้ชัดเจนขึ้น เช่น “วิธีที่ดีที่สุดในการเรียนรู้การรู้จำรูปแบบและการเรียนรู้ของเครื่องก็คือการลงมือทำ.” ทั้งนี้ การใช้มหัพภาคและจุลภาค แม้จะแตกต่างจากรูปแบบที่นิยมปฏิบัติเดิมไปบ้าง แต่เมื่อได้ผิดจากหลักการใช้ภาษาไทยที่ถูกต้องแต่อย่างใด ดังสะท้อนจากข้อความต่อไปนี้ ที่ยกจากราชบัณฑิตและสำนักงานราชบัณฑิตสถาบันฯ

- “มหัพภาค (มะ-ทับ-พาก) คือ เครื่องหมาย จุด ซึ่งเขียนแสดงการจบประโยค. ในภาษาอังกฤษ เรียกเครื่องหมายนี้ว่า full stop แสดงการจบประโยคโดยสมบูรณ์. ในภาษาไทยใช้เครื่องหมาย มหัพภาคเพื่อแสดงว่าจบประโยคได้เช่นเดียวกับในภาษาอังกฤษ.”
— เครื่องหมายมหัพภาค โดย ศ. ดร.กาญจนานา นาคสกุล ราชบัณฑิต ประจำวรรณศิลป์ สาขา วิชาภาษาไทย สำนักศิลปกรรม <http://www.royin.go.th> (คลังความรู้ เครื่องหมายมหัพภาค. สีบคัน 21 กันยายน พ.ศ. 2563)
- “จุลภาค (comma) หรือ จุดลูกน้ำ ชื่อเครื่องหมายวรรคตอน รูปดังนี้ , มีหลักเกณฑ์การใช้ดังต่อไปนี้
 ๑. ใช้แยกกลุ่มประโยคเพื่อกันความเข้าใจสับสน
[ตัวอย่าง ...]
 ๒. ใช้คั่นคำในรายการ ที่เขียนต่อ ๆ กัน ตั้งแต่ ๓ รายการขึ้นไป โดยเขียนคั่นแต่ละรายการ ส่วนหน้าคำ “และ” หรือ “หรือ” ที่อยู่หน้ารายการสุดท้ายไม่จำเป็นต้องใส่เครื่องหมายจุลภาค
[ตัวอย่าง ...]
 ๓. ใช้ในการเขียนบรรนานุกรม บรรชนี และนามานุกรม เป็นต้น
[ตัวอย่าง ...]
 ๔. ใช้คั่นจำนวนเลขนำบจากหลักหน่วยไปทีละ ๓ หลัก”
— เครื่องหมายจุลภาค โดย สำนักงานราชบัณฑิตสถาบันฯ http://www.royin.go.th/?page_id=10392 (สีบคัน 21 กันยายน พ.ศ. 2563)

- “ปรัศนี (question mark) ชื่อเครื่องหมายวรคตอน รูปดังนี้ ? มีหลักเกณฑ์การใช้ดังต่อไปนี้
 ๑. ใช้เมื่อสิ่นสุดความหรือประโยคที่เป็นคำถาม หรือใช้แทนคำ答 [ตัวอย่าง ...]
 ๒. ใช้หลังข้อความเพื่อแสดงความสงสัยหรือไม่แน่ใจ มักเขียนอยู่ในวงเล็บ [ตัวอย่าง ...]”

— ปรัศนี โดย สำนักงานราชบัณฑิตสถาฯ

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

ภาษาไทยอาศัยการตีความในระดับวัจنبัญบติศาสตร์อยู่มาก. การตีความในระดับวัจنبัญบติศาสตร์ (pragmatics) อาศัยสมมุติฐานนึก ความเข้าใจในบริบท และความรู้เดิม. การอธิบายแนวคิดที่แตกต่างจากบริบทหรือความรู้เดิมอย่างมาก อาจทำให้ผู้อ่านไม่สามารถตีความในระดับวัจnbัญบติศาสตร์ได้. ปัจจัยต่าง ๆ ที่ถูกอภิปราย[21] ว่าเป็นส่วนหนึ่ง ที่ทำให้ข้อความภาษาไทยหลาย ๆ ครั้งมีความก้าว光明อย่างมาก ได้แก่ การเป็นภาษาคำโดยที่มีรูปแบบไวยากรณ์ที่ดูง่าย แต่ซับซ้อนด้วยการใช้คำวิเศษณ์โดยไม่มีการผันนิพัตติปัจจัย (นั่นคือ การสร้างคำที่มีความหมายซับซ้อนจากการประกอบคำบรรยายต่าง ๆ เข้าด้วยกัน ที่อาจทำให้เกิดความสับสนระหว่าง นามวลีและประโยค. ทั้งนามวลีและประโยคอาจอยู่ในรูป นาม-กริยา-นามได้ เช่น คนภาคถนน อาจเป็นนามวลี หมายถึง บุคคลผู้ทำหน้าที่ทำความสะอาดถนน หรืออาจเป็นประโยคได้), การไม่มีข้อบ่งบอกขอบเขตของคำที่แน่นอน และการไม่มีข้อบ่งบอกขอบเขตของประโยคที่ชัดเจน เป็นต้น. การใช้มหัพภาค เพื่อบอกการจบประโยคจะช่วยแก้ปัญหาขอบเขตของประโยค. ในขณะที่ การใช้พอนต์ตัวเอียงเพื่อเน้นคำศัพท์หรือกลุ่มคำ จะช่วยบรรเทาปัญหาขอบเขตของคำลง.

``Podrán cortar todas las flores,
pero no podrán detener la primavera."
(in spanish) ---Pablo Neruda

“คุณอาจจะตัดดอกไม้ออกได้หมดทุกดอก
แต่คุณหยุดฤดูใบไม้ผลิที่กำลังมาไม่ได้หรอก.”
—พาโบล เนรูดา

รูปแบบอักษร

เนื่องจากเนื้อหาเกี่ยวข้องอย่างมาก กับทั้งคำศัพท์เฉพาะ คณิตศาสตร์ และโปรแกรมคอมพิวเตอร์ ผู้เขียนต้องการสร้างการตระหนักรู้ถึงความต่าง ระหว่างคำศัพท์ภาษาอังกฤษ สัญลักษณ์ทางคณิตศาสตร์ และรหัสคอมพิวเตอร์ เช่น ฟังก์ชันไชน์ เขียนเป็นภาษาอังกฤษด้วย sine function ในขณะที่สัญลักษณ์ทางคณิตศาสตร์นิยมใช้ sin และรหัสโปรแกรมนิยมใช้ตัวปฏิบัติการ * สำหรับการคูณ แต่สัญลักษณ์ * ไม่ใช้สัญลักษณ์ทางคณิตศาสตร์ที่ยอมรับในวงกว้าง สำหรับการคูณปกติ. ประเด็นเหล่านี้ ผู้เขียนพบว่า นักศึกษาในปัจจุบันมีความสับสนอย่างมาก และผู้เขียนต้องการ อย่างน้อย เป็นส่วนหนึ่งของความพยายามในการช่วยแก้ไขความสับสนเหล่านี้. เพื่อลดความสับสนดังกล่าว รูปแบบอักษรที่ใช้ จะแยกตามประเภทดังนี้

ตัวอักษรภาษาอังกฤษทั่วไปจะใช้รูปแบบ เช่น cosine function. รูปแบบสำหรับโปรแกรมคอมพิวเตอร์ ตัวแปรที่อ้างถึงตัวแปรจากโปรแกรมคอมพิวเตอร์ จะใช้รูปแบบ เช่น $\cos(x, 'DEG')$ โดยตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ขึ้นกับชื่อตัวแปรในโปรแกรม ไม่เกี่ยวข้องกับโครงสร้างชนิดข้อมูลของตัวแปร (ต่างจากรูปแบบที่ใช้กับสัญลักษณ์ทางคณิตศาสตร์ ที่ใช้รูปแบบดังอธิบายในหัวข้อสัญลักษณ์). ตัวอย่างโปรแกรม อาจแสดงได้ดังนี้

รูปแบบที่ 1 ฟังก์ชันเรเดียลเบซิส (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 บางครั้งอาจแสดงเลขบรรทัดออกมาด้วย)

```
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: $\mathbf{x} \in \mathbb{R}^D$

นอกจากนี้ เพื่อช่วยอธิบายโปรแกรมคอมพิวเตอร์ คำอธิบายภายในส่วนข้อคิดเห็น (comment หรือ code comment ซึ่งมีไว้யากรณ์ที่เด่นชัดตามภาษาโปรแกรมคอมพิวเตอร์ รวมถึงอาจได้จัดทำให้เห็นชัดเจนขึ้นอีกด้วยสีที่แตกต่าง) อาจจะใช้รูปแบบอักษรอื่นตามความเหมาะสม เพื่อให้การอธิบายทำได้ชัดเจนขึ้น ดังตัวอย่างข้างต้น

รูปแบบการเขียนตัวเลข อาจจើยនในรูปแบบปกติ เช่น พ.ศ. 2563 หรือในรูปคณิตศาสตร์ เช่น จำนวนพิกเซล $h = 1200$ พิกเซล หรือในรูปรหัสโปรแกรม เช่น $width = 2400$ ตามความเหมาะสม ทั้งนี้ การใช้รูปแบบที่แตกต่างกันนี้จะช่วยแยกความหมายได้ชัดเจนขึ้น และช่วยลดความสับสน คุณเครื่องลงได้มาก

‘‘Form follows function.’’

---Louis Sullivan

“รูปลักษณ์มาทีหลังประโภชน์หน้าที”

—หลุยส์ ซัลลิแวน

สัญลักษณ์

ศาสตร์การรู้จำรูปแบบและการเรียนรู้ของเครื่องอาทัยพื้นฐานทางคณิตศาสตร์ ดังนั้นเพื่อให้ลดความสับสนของตัวแปรคณิตศาสตร์ที่ใช้ สัญลักษณ์ของตัวแปรคณิตศาสตร์จะใช้ตามแนวทางดังตารางข้างล่าง ยกเว้นแต่จะระบุเป็นอื่น

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	$\mathbf{X} \mathbf{Y} \mathbf{Z}$	$\mathbf{\Theta} \mathbf{\Phi} \mathbf{\Omega}$

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

เวกเตอร์ $\mathbf{x} \in \mathbb{R}^4$ 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 ทั้งนี้คำบรรยายภาพจะช่วยให้ความกระจางแทน

‘‘Four basic premises of writing:
clarity, brevity, simplicity, and
humanity.’’

---William Zinsser

“ข้อตั้งสี่ ประการ ของการ เขียน 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 การรู้จำตัวเลขหลายมือ	10
1.4 ประเภทของการเรียนรู้ของเครื่อง	12
1.5 การเรียนรู้ของเครื่องและศาสตร์ที่เกี่ยวข้อง	15
1.6 อภิธานศัพท์	17
1.7 แบบฝึกหัด	18
2 พื้นฐาน	29
2.1 พีชคณิตเชิงเส้น	29
2.2 ความน่าจะเป็น	44
2.3 การหาค่าดีที่สุด	59
2.4 อภิธานศัพท์	75
2.5 แบบฝึกหัด	78
3 การเรียนรู้ของเครื่องและโครงข่ายประสาทเทียม	107
3.1 การปรับเส้นโค้งด้วยฟังก์ชันพหุนาม	107
3.2 คุณสมบัติความทั่วไปและการเลือกแบบจำลอง	114

สารบัญ

3.3	โครงข่ายประสานเที่ยม	123
3.4	การประยุกต์ใช้โครงข่ายประสานเที่ยม	141
3.5	คำแนะนำสำหรับการใช้แบบจำลองทำนาย	146
3.6	อภิธานศัพท์	151
3.7	แบบฝึกหัด	154
4	การเรียนรู้ของเครื่องในโลกกว้าง	213
4.1	การวิเคราะห์พฤติกรรมลูกค้า	213
4.2	ชัพพอร์ตเวกเตอร์แมชีน	233
4.3	อภิธานศัพท์	248
4.4	แบบฝึกหัด	252
ii	การเรียนรู้เชิงลึก	265
5	การเรียนรู้เชิงลึก	267
5.1	ปัญหาการเลือนหายของเกรเดียนต์	269
5.2	การฝึกทีละหมู่เล็ก	272
5.3	เทคนิคการตกอອก	277
5.4	การทำหนดค่า�ำหนักเริ่มต้น	281
5.5	กลไกช่วยการฝึก	283
5.6	อภิธานศัพท์	291
5.7	แบบฝึกหัด	293
6	โครงข่ายคอนโวโลชัน	345
6.1	ชั้นคอนโวโลชัน	348
6.2	ชั้นดึงรวม	359
6.3	เกรเดียนต์ของโครงข่ายคอนโวโลชัน	360
6.4	สรุปการคำนวณของโครงข่ายคอนโวโลชันสองมิติ	371
6.5	โครงข่ายคอนโวโลชันที่สำคัญ	375
6.6	อภิธานศัพท์	377
6.7	แบบฝึกหัด	378
7	การเรียนรู้เชิงลึกในโลกการรู้จำทัศนรูปแบบ	397
7.1	การตรวจจับวัตถุในภาพ	397

7.2 การซ่อม เสริม และก่อกำเนิดภาพ	403
7.3 อภิธานศพท์	422
7.4 แบบฝึกหัด	425
iii การรีจารูปแบบเชิงลำดับ	429
8 แบบจำลองสำหรับข้อมูลเชิงลำดับ	431
8.1 ข้อมูลเชิงลำดับ	431
8.2 แบบจำลองมาร์คอฟ	433
8.3 แบบจำลองมาร์คอฟช่องเร้น	437
8.4 อภิธานศพท์	448
8.5 แบบฝึกหัด	449
9 การรีจารูปแบบเชิงลำดับในโลกการประมวลผลภาษาธรรมชาติ	453
9.1 การประมวลผลภาษาธรรมชาติ	453
9.2 โครงข่ายภาษาที่เวียนกลับ	455
9.3 โครงข่ายภาษาที่เวียนกลับสองทาง	465
9.4 แบบจำลองความจำระยะสั้นที่ยาว	469
9.5 การใช้งานโครงข่ายภาษาที่เวียนกลับ	472
9.6 กลไกความใส่ใจ	475
9.7 อภิธานศพท์	481
9.8 แบบฝึกหัด	483
บรรณานุกรม	487
บรรณานุกรมภาษาไทย	509
บรรณานุกรมภาษาอังกฤษ	518

สารบัญรูป

1.1	ภาพใหญ่การรู้จำรูปแบบและการเรียนรู้ของเครื่อง	5
1.2	ตัวอย่างรูปตัวเลขจากลายมือเขียน	10
1.3	แผนภาพแสดงระบบรู้จำตัวเลขลายมือ	11
1.4	แผนภาพการค้นหาฟังก์ชันรู้จำตัวเลขลายมือ	12
1.5	แผนภาพการค้นหาฟังก์ชันรู้จำตัวเลขลายมือ	13
1.6	ระดับเสียงเต็มรูป	27
2.1	การฉายภาพ	40
2.2	ตัวอย่างการฉายเวกเตอร์	40
2.3	ตัวอย่างความน่าจะเป็น ลังสี่ลูกบอล	47
2.4	ผลจากการจำลองเหตุการณ์สุ่มหิบลูกบอล	47
2.5	การลู่เข้าของอัตราส่วนการหิบได้สีเขียว	48
2.6	ตัวอย่างความน่าจะเป็นแบบมีเงื่อนไข	48
2.7	ตัวอย่างเพิ่มเติมความน่าจะเป็นแบบมีเงื่อนไข	54
2.8	ความเกี่ยวเนื่องของสองเหตุการณ์	54
2.9	ภาพแสดงความเกี่ยวเนื่องของสามเหตุการณ์	55
2.10	ความหนาแน่นความน่าจะเป็น ของการแจกแจงเกาส์เชียน	60
2.11	ความหนาแน่นความน่าจะเป็นและการแจกแจงความน่าจะเป็นสะสม	60
2.12	ปัญหาค่ามากที่สุดกับ ปัญหาค่าน้อยที่สุด	63
2.13	ค่าทำให้น้อยที่สุดต่าง ๆ	64
2.14	ภาพคอนทัวร์และภาพสามทศนิยติของฟังก์ชันจุดประสงค์	67
2.15	เส้นทางการหาค่าทำให้น้อยที่สุด	69
2.16	ความก้าวหน้าในการหาค่าทำให้น้อยที่สุด	70
2.17	ตัวอย่างปัญหาที่มีข้อจำกัด	73
2.18	ตัวอย่างฟังก์ชันลงโทษ	73
2.19	ตัวอย่างฟังก์ชันสูญเสียที่ถูกกลงโทษ	73

2.20	ตัวอย่างแสดงความต่างของค่าแปรปรวนและเงอนໂທປີ	79
2.21	ຝັກໜັນບວກອ່ອນ ພັກໜັນຊົກມອຍດີ ແລະ ພັກໜັນເກາສ්-ເෂියන	86
2.22	ตัวอย่างภาพและผลการแก้ค่าພິກເສລ	87
2.23	ผลการจำลองປັບຫາມອນຕື້ອລ	88
2.24	ຄວາມສັນພັນຮູ່ຂອງຄວາມຄລາດເຄລື່ອນການคำນວນກັບນາດຂ້ອນມູລ	89
2.25	ผลการทำงานວິຮີລຶງເກຣເດີຢັນຕົ້ນ ທີ່ຄ່າຂາດກ້າວຕ່າງ ໆ	94
2.26	ผลการทำงานວິຮີລຶງເກຣເດີຢັນຕົ້ນ ທີ່ໃຊ້ຄ່າຂາດກ້າວຕ່າງ ໆ ໃນຮອບຄໍານວນຕົ້ນ ໆ	95
2.27	ຜລລັບຮົມຈາກວິຮີລຶງເກຣເດີຢັນຕົ້ນ ທີ່ໃຊ້ຄ່າຂາດກ້າວຕ່າງ ໆ	95
2.28	ຝັກໜັນຈຸດປະສົງຄໍຂອງປັບຫາຫລາຍກວາງ	96
2.29	ຕົວຢ່າງຜລກາຮແກ້ປັບຫາຄ່ານ້ອຍທີ່ສຸດແບບມື້ອັຈຬກັດ ເມື່ອໃຊ້ວິກາລົງໂທ່າງ	97
2.30	ຕົວຢ່າງປັບຫາຄ່ານ້ອຍທີ່ສຸດແບບມື້ອັຈຬກັດ ແລະ ເຈື່ອນໄຂຄາຮູ່ຄຸນທັກເກອຮ້	99
2.31	ຕົວຢ່າງກວາງຄູ່ກັນ	103
2.32	ຜລກາຮາດກາຮົບຝັກໜັນໃໝ່ນ ທີ່ດູຕ່າງຈາກຄວາມຄາດຫວັງ	104
2.33	ຜລຈາກກາຮາດກາຮົບຝັກໜັນໃໝ່ນ ທີ່ໄດ້ດູແປລກຈາກທີ່ຄາດ	105
3.1	ຕົວຢ່າງຈຸດຂ້ອນມູລ	108
3.2	ຝັກໜັນພහຸນາມຕ່າງ ໆ	109
3.3	ຄວາມຜິດພາດຂອງການທໍານາຍ	110
3.4	ພຸດທິກຣມຂອງແບບຈຳລອງພහຸນາມຮະດັບຂັ້ນໜຶ່ງທີ່ຝຶກແລ້ວ	111
3.5	ພຸດທິກຣມການທໍານາຍຂອງແບບຈຳລອງພහຸນາມທີ່ຮະດັບຂັ້ນຕ່າງ ໆ	115
3.6	ພຸດທິກຣມການທໍານາຍຂອງແບບຈຳລອງທີ່ຮະດັບຂັ້ນຕ່າງ ໆ	116
3.7	ຄ່າຜິດພາດຊຸດຝຶກກັບຄ່າຜິດພາດຊຸດທົດສອນ	116
3.8	ແບບຈຳລອງພහຸນາມເມື່ອໃຊ້ຈຳນວນຈຸດຂ້ອນມູລມາກຂຶ້ນ	117
3.9	ພහຸນາມຮະດັບຂັ້ນເກົ້າ ກັບການທຳຄ່ານ້ຳໜັກເສື່ອມ	119
3.10	ການທຳນ້ຳໜັກເສື່ອມດ້ວຍລາກຮານຈົ່າຕ່າງ ໆ	120
3.11	ວິທີຮອສວາລີເທື່ນໜ້າພັບ	121
3.12	ກາພແສດງເໜີລົບປະສາທເຂື່ອມຕ່ອງສ້າງຄວາມກັນຝ່າຍໃໝ່ແນປ່ງ	123
3.13	ຮູປແສດງໂຄຮສ້າງຂອງເໜີລົບປະສາທທົ່ວ ໆ ໄປ	124
3.14	ໂຄຮສ້າງຂອງເພອ່ງເໜີປ່ອຕອນ	125
3.15	ຕົວຢ່າງໂຄຮຂ່າຍເພອ່ງເໜີປ່ອຕອນສອງຂັ້ນ	127
3.16	ຝັກໜັນຈຳກັດແບ່ງກັບຝັກໜັນຊົກມອຍດີ	131
3.17	ຜລຈາກໂທນັດທີ່ສ່າງຝ່າຍໂທນັດອື່ນ ໆ ໃນຫັ້ນຄໍານວນຄັດໄປ	134
3.18	ພຸດທິກຣມຂອງຝັກໜັນສູງເສີຍ	140
3.19	ໂຄຮຂ່າຍປະສາທເຖິມກັບນາດຂອງອິນພຸຕ	142

สารบัญ

3.20	ฟังก์ชันซิกมอยด์	144
3.21	การฝึกที่จำนวนสมัยต่าง ๆ	145
3.22	เส้นโค้งเรียนรู้	148
3.23	ตัวอย่างพฤติกรรมของแบบจำลองที่ความซับซ้อนต่าง ๆ	149
3.24	ตัวอย่างเส้นโค้งเรียนรู้	150
3.25	ผลเมื่อสุมกำหนดค่าหน้าหนักเริ่มต้น	164
3.26	ผลเมื่อกำหนดค่าหน้าหนักเริ่มต้นด้วยวิธี heuristic โดร์	165
3.27	ผลการทดลองข้าม กับวิธีการกำหนดค่าหน้าหนักเริ่มต้น	166
3.28	แผนภูมิกล่องผลการทดลองข้าม	166
3.29	ความสำคัญของการทำข้าม เมื่อสุมกำหนดค่าเริ่มต้น	168
3.30	ชุดข้อมูลเรียกใช้	171
3.31	ตัวอย่างการนำเสนอผลการทำนายชุดข้อมูลเรียกใช้	172
3.32	ชุดข้อมูลเรียกใช้ เปรียบเทียบการฝึกแบบหมู่ และการฝึกแบบออนไลน์	172
3.33	ข้อมูลภาพเอ็กซ์เรย์เต้านม ลักษณะสำคัญมิติแรก	180
3.34	ข้อมูลภาพเอ็กซ์เรย์เต้านม ลักษณะสำคัญมิติที่สอง	181
3.35	ข้อมูลภาพเอ็กซ์เรย์เต้านม ลักษณะสำคัญต่าง ๆ ที่เป็นค่าแทนชื่อ	181
3.36	การทำนายข้อมูลภาพเอ็กซ์เรย์เต้านม	182
3.37	แผนภูมิกล่องเปรียบเทียบ วิธีการจัดการกับข้อมูลขาดหายแบบต่าง ๆ	184
3.38	การแจกแจงของข้อมูลเอมนิสต์ชุดฝึก	185
3.39	อินพุตของเอมนิสต์	186
3.40	ตัวอย่างภาพตัวเลขข้อมูลเอมนิสต์	186
3.41	ตัวอย่างภาพที่สับสนของเอมนิสต์	189
3.42	การแจกแจงของข้อมูลไม่เลกุล	195
3.43	เจ้าตัวพุตของแบบจำลอง สำหรับลิแกนต์และตัวหลอก	197
3.44	ค่าคณนเอย์ เมื่อใช้ระดับค่าขีดแบ่งต่าง ๆ	197
3.45	กราฟอาร์โธซีการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีน	198
3.46	ผลหลังแก้ข้อมูลไม่สมดุล	200
3.47	ค่าคณนเอย์ ที่ระดับค่าขีดแบ่งต่าง ๆ หลังแก้ข้อมูลไม่สมดุล	201
3.48	การทำนายการจับตัวของโมเลกุล หลังปรับปรุงด้วยวิธีสุมเกิน	201
4.1	แผนภาพความร้อน	214
4.2	วิธีการลบจากหลัง	216
4.3	เทคนิคหน้าต่างเลื่อน	218
4.4	ตัวอย่างการเลือกส่วนภาพด้วยวิธีหน้าต่างเลื่อน	219

4.5	วิธีหน้าต่างเลื่อนกับการตรวจจับภาพเป้าหมาย	220
4.6	ตัวอย่างพิกเซลเกรเดียนต์ของภาพ	222
4.7	ตัวอย่างการทำเอชโอลีเชลล์	223
4.8	ขั้นตอนการทำเอชโอลีบล็อก	224
4.9	ตัวอย่างลักษณะสำคัญของเอชโอลี	224
4.10	ผลจากการประมวลความหนาแน่นแก่น	228
4.11	ตัวอย่างแสดงผลลัพธ์การตรวจจับและผลเฉลย	230
4.12	กราฟความเที่ยงตรงและการเรียงกลับ และพื้นที่ใต้กราฟ	231
4.13	อภิรະนา	233
4.14	ความสัมพันธ์ระหว่างพารามิเตอร์กับคุณลักษณะของอภิรະนา	234
4.15	ช่องว่างของการแบ่งกลุ่ม	236
4.16	ระยะทางจากอภิรະนาไปจุดข้อมูล	237
4.17	ผลลัพธ์ของชั้พพอร์ตเวกเตอร์แมชชีน	241
4.18	ข้อมูลที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์	243
4.19	ผลการฝึกชั้พพอร์ตเวกเตอร์แมชชีนกรณีทั่วไป	245
4.20	พฤติกรรมของชั้พพอร์ตเวกเตอร์แมชชีนที่ค่า C ต่าง ๆ	245
4.21	ชั้พพอร์ตเวกเตอร์แมชชีน ด้วยเกาส์เซียนเครื่องเนลที่ค่า σ ต่าง ๆ	249
4.22	ชั้พพอร์ตเวกเตอร์แมชชีน ด้วยเครื่องเนลเชิงเส้น เพื่อเปรียบเทียบกับเกาส์เซียน	250
4.23	ตัวอย่างการประมวลค่าความหนาแน่นความนำจะเป็นสำหรับข้อมูลสองมิติ	253
4.24	ตัวอย่างข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น	256
4.25	ตัวอย่างอภิรະนาที่ทำได้จากรูปปั๊ม	256
4.26	ตัวอย่างข้อมูลที่ไม่สามารถแบ่งแยกสมบูรณ์ได้เชิงเส้น	263
5.1	ค่าพิงก์ชั้นสูญเสียต่อสมัยฝึก เมื่อใช้โครงข่ายประสาทเทียมที่ความลึกต่าง ๆ	270
5.2	การฝึกโครงข่ายลึกล้มเหลว	270
5.3	ปัญหาการเลื่อนหายของเกรเดียนต์	271
5.4	การเลื่อนหายของเกรเดียนต์ในโครงข่ายสิบชั้น	271
5.5	ฟังก์ชันกระตุนต่าง ๆ	272
5.6	ตัวอย่างความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบชั้น เมื่อใช้เรลู	273
5.7	การเลื่อนหายของเกรเดียนต์บรรเทาลงด้วยการเปลี่ยนฟังก์ชันกระตุน	273
5.8	ค่าพิงก์ชั้นสูญเสียที่ความลึกต่าง ๆ เมื่อใช้เรลู เปรียบเทียบกับซิกมอยด์	273
5.9	ตัวอย่างเวลาการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ	275
5.10	ตัวอย่างคุณภาพการทำงานเมื่อใช้ขนาดหมู่เล็กต่าง ๆ	276
5.11	ตัวอย่างผลจากการใช้วิธีจัดหมู่เล็กแบบต่าง ๆ	276

สารบัญ

5.12	ภาพประกอบแสดงแนวคิดของการตอกออก	278
5.13	พุติกรรมซิกแซก	285
5.14	ภาพแสดงการทำงานของกลไกโมเมนตัม	286
5.15	แนวทางที่นิยมดำเนินการกับค่าน้ำหนักจากการฝึกก่อน	290
5.16	ตัวอย่างข้อมูลสำหรับแสดงปัญหาการเลือนหายของเกรเดียนต์	293
5.17	ปัญหาการเลือนหายของเกรเดียนต์	300
5.18	ตัวอย่างจุดข้อมูล สำหรับโครงข่ายประสาทเทียมเพื่อทำการแยกจำ	326
5.19	ผลลัพธ์การเรียนการจำจำ	330
5.20	ผลลัพธ์การเรียนค่าเบี่ยงเบนมาตรฐาน	331
5.21	ตัวอย่างการจำจำค่าเริ่มต้นของไปอัส	331
5.22	ผลการกระตุนระหว่างการฝึก	334
5.23	ผลเปรียบเทียบฟังก์ชันกระตุนและวิธีกำหนดค่าเริ่มต้น	335
5.24	การจำจำของค่าการกระตุน	336
5.25	ค่าฟังก์ชันสูญเสียต่อสมัยฝึก เมื่อไม่ใช้และใช้แบบอร์ม	339
5.26	ค่าผิดพลาดเมื่อไม่ใช้และใช้แบบอร์ม	339
5.27	ค่าผิดพลาดเมื่อไม่ใช้และใช้แบบอร์ม ที่อัตราเรียนรู้ต่าง ๆ	339
5.28	ค่าเฉลี่ยค่าผิดพลาด เมื่อใช้และไม่ใช้แบบอร์ม ที่ขนาดหมู่เล็กต่าง ๆ	340
5.29	แบบอร์มและขนาดของหมู่เล็ก	340
5.30	แบบอร์มเปลี่ยนความสัมพันธ์ของการฝึกและขนาดหมู่เล็ก	340
5.31	คุณภาพของแบบอร์มกับการเลื่อนของความแปรปรวนร่วมเกี่ยวกับยกอก	341
6.1	โครงสร้างของมิติ	347
6.2	ขั้นการเขื่อมต่อแบบต่าง ๆ	349
6.3	แผนภาพการเขื่อมต่อของขั้นตอนโนโลจิชัน	349
6.4	ขั้นตอนโนโลจิชันที่ใช้พิลเตอร์ขนาดต่าง ๆ	350
6.5	แผนภาพแสดงการเติมเต็มด้วยศูนย์	352
6.6	ขนาดก้าวย่าง	353
6.7	แผนภาพแสดงการคำนวนขั้นตอนโนโลจิชัน	355
6.8	ตัวอย่างการคำนวนค่าตอนโนโลจิชันในข้อมูลที่มีโครงสร้างมิติ	357
6.9	โครงสร้างมิติของอนพุต พิลเตอร์ และเอาร์พุตของขั้นตอนโนโลจิชัน	358
6.10	แผนภาพการแบ่งกระจายย้อนกลับ	363
6.11	การแบ่งกระจายย้อนกลับจากขั้นเอาร์พุต	363
6.12	การแบ่งกระจายย้อนกลับจากขั้นซ้อน	364
6.13	การแบ่งกระจายย้อนกลับผ่านโครงสร้างการต่อเชื่อม	366

6.14	โครงสร้างของอเล็กซ์เน็ต	376
6.15	ตัวอย่างการจัดเทนเซอร์ให้อยู่ในรูปที่การคุณเมทริกซ์สมือในการคำนวนเทนเซอร์	382
7.1	การจำแนกภาพ และการตรวจจับวัตถุในภาพ	398
7.2	โยโลใช้อาร์พุตที่มีโครงสร้างเทียบเท่าการแบ่งส่วนภาพอินพุต	400
7.3	ตัวอย่างแสดงกรณีสำหรับเทคนิคกล่องสมอ	400
7.4	โยโลเลือกกล่องสมอเพื่อรับผิดชอบบัวตุ	401
7.5	การขยายความละเอียดภาพด้วยชั้นดีคอนโนวูลชั้น	406
7.6	วิธีการฝึกแบบปรับปักรหัส	407
7.7	การฝึกโครงสร้างปรับปักรหัสเชิงสร้างแบบมีเงื่อนไข	409
7.8	การทำพิษคอมิตเตอร์กับเวกเตอร์ค่าสุ่มของโครงข่ายก่อกำเนิด	411
7.9	โครงสร้างของใบแกนและการอนุமานที่เรียนเชิงปรับปักรหัส	412
7.10	ภาพแสดงสมมติฐานการเรียนรู้การแยกแจงข้อมูลของโครงข่ายปรับปักรหัสเชิงสร้าง	414
7.11	ค่อนโนวูลชั้นก้าวเศษช่วยขยายขนาดแผนที่ลักษณะสำคัญ	415
7.12	การแยกແຍະໜູ້ເລືກ	416
8.1	ตัวอย่างการกิจการรู้จำรูปแบบเชิงลำดับ	434
8.2	ตัวอย่างสมมติฐานแบบต่าง ๆ ของความสัมพันธ์ระหว่างจุดข้อมูล	436
9.1	ภาพรวมของการประมวลผลภาษาธรรมชาติ	456
9.2	ตัวอย่างอินพุตເອົາຕົວພຸດຂອງກิจกรรมຮູ່ມວດຄຳ	456
9.3	ตัวอย่างโครงข่ายประสาทເວີນກັບ	457
9.4	ตัวอย่างโครงข่ายประสาทເວີນກັບ ພັນຍົມຕົວຢູ່ຂອ້ມູນລຳດັບ	457
9.5	ແຜນກາພໂຄງສ້າງໂດຍຮົມຂອງໂຄງຂ່າຍປະສາກເວີນກັບ	458
9.6	ແຜນກາພຄືລຳດັບຂອງໂຄງຂ່າຍປະສາກເວີນກັບ	458
9.7	ແຜນກາພຄືລຳດັບຂອງໂຄງຂ່າຍປະສາກເວີນກັບສອງທາງ	466
9.8	ແຜນກາພໂຄງສ້າງບັນດາຄວາມຈຳຂອງແບບຈຳລອງຄວາມຈຳຮະຍະສັ້ນທີ່ຍາວ	470
9.9	ແຜນກາພຄືລຳດັບຂອງແບບຈຳລອງຄວາມຈຳຮະຍະສັ້ນທີ່ຍາວ	471
9.10	ແຜນກາພຄືລຳດັບ ກຣນິທີ່ທັງອິນພຸດແລະເອົາຕົວພຸດເປັນຊຸດລຳດັບ	472
9.11	ແຜນກາພຄືລຳດັບຂອງໂຄງຂ່າຍປະສາກເວີນກັບ ກຣນິກາຈຳແນກລຳດັບ	473
9.12	ແຜນກາພຄືລຳດັບ ກຣນິທີ່ອິນພຸດໄມ້ໃໝ່ຊຸດລຳດັບ	474
9.13	ແຜນກາພຄືລຳດັບຂອງສາປັຕຍກຣມຕ້ວເຂົ້າຮ້າສັ່ວົນອດຮ້າສ	474
9.14	ໂຄງຂ່າຍປະສາກເວີນກັບແບບລຶກ	476
9.15	ໂຄງຂ່າຍປະສາກເວີນກັບແບບລຶກ ທີ່ໃຊ້ໜັ້ນຄໍານວນໄມ້ເວີນກັບຈຳນວນมาก	477
9.16	ຕ້ວເຂົ້າຮ້າສັ່ວົນອດຮ້າສ ທີ່ໃຊ້ຮ້າສເນື້ອຄວາມປະກອບອິນພຸດຂອງຕ້ວອດຮ້າສ	478

สารบัญรูป

9.17 แผนภาพแสดงโครงสร้าง เมื่อใช้กลไกความไม่ใช้	479
9.18 แผนภาพคลี่ลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวผลลัพธ์	483

สารบัญตาราง

2.1	ภาษาเฉพาะเรื่องเขตกับเรื่องความน่าจะเป็น	46
2.2	อัตราส่วนของการสุ่ม抽บันได้ลูกบลอสสีเขียว	47
2.3	สรุปค่าความน่าจะเป็นร่วม	53
2.4	คุณสมบัติที่มักสับสนของตัวแปรสุ่มต่อเนื่อง	58
3.1	ค่าพารามิเตอร์ของแบบจำลอง กับการทำค่าน้ำหนักเสื่อมที่ลากรานจ์ค่าต่าง ๆ	119
3.2	ตระรากเอ็กซ์ขอร์	126
3.3	ตัวอย่างการทำงานของเพอร์เซปตรอน	128
3.4	สรุปคำแนะนำสำหรับปรับปรุงแบบจำลอง	151
3.5	ค่าสถิติของผลการฝึกแบบจำลอง จากการทำทดลองช้า	167
3.6	ตัวอย่างผลลัพธ์จากการกับข้อมูลขาดหาย	183
3.7	เมทริกซ์สับสน	189
4.1	ตัวอย่างการคำนวณค่าเฉลี่ยค่าประมาณความเที่ยงตรง	232
6.1	ตัวอย่างแสดงความสัมพันธ์ของดัชนีหน่วยคำนวณ	368

สารบัญรายการ

2.1	วิธี亥เมทริกซ์ผกผันด้วยวิธีเก้าส์จอร์เดน	82
2.2	วิธีลงเกรเดียนต์	90
2.3	วิธีลงเกรเดียนต์ที่มีเงื่อนไขการจบ	91
2.4	วิธีลงเกรเดียนต์ เมื่อตัวแปรเป็นเวกเตอร์	93
3.1	แบบจำลองพหุนาม	156
3.2	โปรแกรมฝึกพหุนามระดับขั้นหนึ่ง	156
3.3	ตัวอย่างโปรแกรมการปรับเส้นโค้ง	156
3.4	โครงข่ายประสาทเทียม	157
3.5	ฟังก์ชันจำกัดแข็ง	158
3.6	ตัวอย่างโครงข่ายประสาทสำหรับตระกระเอ็กซ์ซอร์	158
3.7	โปรแกรมฝึกโครงข่ายประสาทเทียม	159
3.8	โปรแกรมฟังก์ชันซิกมอยด์และอนุพันธ์	161
3.9	โปรแกรมหาค่าอนุพันธ์ของฟังก์ชันเอกลักษณ์	161
3.10	โปรแกรมหาค่าอนุพันธ์ของฟังก์ชันเริ่มต้นด้วยการสุ่ม	162
3.11	ค่าผิดพลาดกำลังสอง	162
3.12	การกำหนดค่าอนุพันธ์ของฟังก์ชันเริ่มต้นให้เป็นวิดิโอดริว	164
3.13	โปรแกรมนอร์มออนไลน์อินพุต	168
3.14	โปรแกรમฝึกโครงข่ายประสาทเทียมแบบออนไลน์	172
3.15	ฟังก์ชันสูญเสียครอสເອນໂທຣປີທິວິກາດ	176
3.16	โปรแกรມແທນຄ່າຂາດຫາຍ້ວຍທຸກຄ່າທີ່ເປັນໄປໄດ້	182
3.17	ฟังก์ชันซอฟຕ์ແມର්କ්	187
3.18	ฟังก์ชันสูญเสียครอสເອນໂທຣປີ	188
3.19	โปรแกรມໂທລດຂໍ້ມູນສາງປະກອບ	190
3.20	ตัวอย่างโปรแกรມเลือกລັກຊະສຳຄັນຂອງໂນເລຸກ	192
3.21	ตัวอย่างโปรแกรມນັບອະຕອນແລະນັບພັນຮະ	193
3.22	โปรแกรມຫາເກຣເດີຢັນຕື່ອງເລີຂໍ	210

4.1	วิธีการประมาณความหนาแน่นแก่น	252
4.2	โปรแกรมค้นหาอภิรานาบ ในปัญหาปัญหของชั้พพอร์ตเวกเตอร์แมชีน	254
4.3	โปรแกรมวิริลงเกรเดียนต์	255
4.4	ชั้พพอร์ตเวกเตอร์แมชีน ปัญหาปัญห กรณีแบ่งแยกได้โดยสมบูรณ์	257
4.5	ชั้พพอร์ตเวกเตอร์แมชีน	260
4.6	เครื่องเรนเลก้าส์เซียน	263
5.1	คลาสโคงข่ายประสาทเทียม	295
5.2	ฟังก์ชันกระตุ้น เทียนด้วยไฟฟอร์ซ	301
5.3	ตัวอย่างโปรแกรมรันโครงข่ายประสาทเทียมที่เขียนด้วยไฟฟอร์ซ	303
5.4	คลาสโคงข่ายประสาทเทียมแบบไฟฟอร์ซ	303
5.5	โปรแกรมโครงข่ายประสาทเทียมไฟฟอร์ซและการหาเกรเดียนต์อัตโนมัติ	308
5.6	โปรแกรมโครงข่ายประสาทเทียมด้วยไฟฟอร์ซ nn	312
5.7	ตัวอย่างคำสั่งฝึกและทดสอบโครงข่ายประสาทเทียม	312
5.8	โปรแกรมโครงข่ายประสาทเทียมด้วยไฟฟอร์ซ nn.Module	314
5.9	คลาส MyDataset เพื่อใช้กับ utils.data.DataLoader	315
5.10	ตัวอย่างคำสั่งฝึกแบบจำลอง ด้วยมอดูล optim	317
5.11	ตัวอย่างโปรแกรมการตกอก	318
5.12	ตัวอย่างโปรแกรมใช้การตกอกที่เขียนขึ้นเอง	319
5.13	ตัวอย่างการฝึกและทดสอบโครงข่ายที่ใช้การตกอกที่เขียนขึ้นเอง	319
5.14	ตัวอย่างโปรแกรมการตกอก โดยการซัดเซยล่วงหน้า	320
5.15	ตัวอย่างโปรแกรมชั้นสัญญาณรบกวน	322
5.16	โปรแกรมสร้างข้อมูลสำหรับการทำนายการแจกแจง	326
5.17	โปรแกรมคำนวณค่าลับลอการิทึมของฟังก์ชันควรจะเป็น	327
5.18	โปรแกรมโครงข่ายประสาทเทียม เพื่อทำนายการแจกแจง	328
5.19	การฝึกโครงข่ายประสาทเทียม เพื่อทำนายการแจกแจง	328
5.20	ตัวอย่างการทำหนดค่าเริ่มต้นให้โครงข่ายประสาทเทียม	332
5.21	ฟังก์ชันกำหนดค่าหน้าหนักเริ่มต้นเชเวียร์	332
5.22	ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้แบบนอร์ม	337
5.23	คลาสแบบนอร์ม	337
6.1	ตัวอย่างโปรแกรมชั้นคำนวณคอนโวลูชัน	381
6.2	ตัวอย่างการเรียกใช้ชั้นคำนวณคอนโวลูชัน	385
6.3	ตัวอย่างการฝึกโครงข่ายคอนโวลูชัน	386
6.4	ตัวอย่างการทดสอบโครงข่ายคอนโวลูชัน	386

สารบัญตาราง

6.5	ตัวอย่างโปรแกรมชั้นเชื่อมต่อเต็มที่	387
6.6	ตัวอย่างโปรแกรมชั้นเชื่อมต่อเต็มที่ ที่เขียนการแพร่กระจายย้อนกลับเอง . . .	388
6.7	ตัวอย่างโปรแกรมชั้นคอนโวโลชัน	389
6.8	ตัวอย่างโปรแกรมชั้นคอนโวโลชัน ที่เขียนการแพร่กระจายย้อนกลับเอง . . .	392
6.9	ตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด	393
6.10	ตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด ที่เขียนการแพร่กระจายย้อนกลับเอง	394

ภาค i

การเรียนรู้ของเครื่อง

บทที่ 1

บทนำ

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

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

เจเรมี โฮเวิร์ด (Jeremy Howard) หนึ่งในผู้เชี่ยวชาญทางด้านการเรียนรู้ของเครื่อง ได้ระบุสี่คุณสมบัติที่สำคัญสำหรับผู้ที่เหมาะสมกับงานการเรียนรู้ของเครื่อง ได้แก่ หนึ่ง ความอึด (tenacity), สอง ความอยากรู้อยากเห็น (curiosity), สาม ความคิดสร้างสรรค์ (creativity) และสุดท้ายสี่ ทักษะ (skills) ซึ่งหมายถึงคณิตศาสตร์และการเขียนโปรแกรมคอมพิวเตอร์. คุณสมบัติทั้งสี่นี้เรียงตามลำดับ นั่นคือ ความอึดสำคัญที่สุด.

อย่างไรก็ตาม บางคนอาจเริ่มด้วยคุณสมบัติที่เหมาะสมกับงาน แต่บางคนอาจใช้งานเป็นแรงจูงใจในการพัฒนาคุณสมบัติให้เกิดขึ้นในตัวเอง ดังคำกล่าวของ เจมส์ แอนโธนี ฟรูด ว่า “ความคิดฝันไม่อาจเนรมิตตัวคุณให้เป็นคนที่คุณนับถือได้ คุณต้องมุ่งมั่นมากบั้นพัฒนาตัวให้เป็นให้ได้” (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 ถ้าคุณคิด ๆ ประวัติศาสตร์ไม่มีอะไร นอกจากซ้ำตัวมันเอง สิ่งที่เรารอเรียก ความยุ่งเหงิง ก็เป็นแค่รูปแบบที่เรายังมองไม่ออก สิ่งที่เรารอเรียก ไร้แบบแผน ก็เป็นแค่รูปแบบที่เรายังอ่านไม่ออก อะไรที่เราไม่เข้าใจ เราว่าไรสาระ อะไรที่เราอ่านไม่ออก เราว่าไม่มีความหมาย.”

—ชัก ปalaññineoku

รูปแบบมีอยู่ในทุก ๆ อย่าง ไม่ว่าจะธรรมชาติ เอกภพ ชีวิต หรือจิตปัญญา. ไม่ว่าจะประวัติศาสตร์ สมบัติ การเอาตัวรอด กีฬา การเต้นรำ การเคลื่อนไหว การคิด ดนตรี ศิลปะ ความรู้ หรือภาษา ล้วนมีรูปแบบอยู่. **รูปแบบ (pattern)** หรือการซ้ำเชิงโครงสร้าง (structural repetition) ซึ่งทำให้เราเข้าใจ ความเป็นไปต่าง ๆ ซึ่งทำให้เราจดจำผู้คน อาหาร อันตราย วิธีเอาตัวรอด ภาษา สถานที่ สัญลักษณ์ วัตถุ แนวคิด ไปจนถึง เรื่องราวต่าง ๆ ได้.

การที่เรารู้ว่าภาพที่เห็นเป็นภาพของอะไร มีวัตถุอะไรอยู่บ้าง อยู่ที่ไหน หรือภาพบอกเล่าเหตุการณ์ อะไร เป็นเพระมีรูปแบบของภาพที่เราจำได้หรือรู้จักอยู่. การที่เราเข้าใจเสียงที่ได้ยินว่าเป็นเสียง ของอะไร เสียงพูดของใคร กำลังพูดภาษาอะไร สำเนียงของที่ไหน พูดถึงอะไร อารมณ์เป็นอย่างไร เป็นเพระมีรูปแบบของเสียง ของภาษาที่เราจำได้อยู่. การที่เรารู้ได้ฟังหรืออ่านข้อความของภาษา แล้วเข้าใจ ความหมาย เป็นเพระมีรูปแบบของภาษา ของความหมายที่เราจำได้อยู่ รวมถึงมีรูปแบบของการเชื่อม ความสัมพันธ์ต่าง ๆ เข้าด้วยกัน และรูปแบบการสร้างรูปแบบใหม่ ที่เรารับรู้อยู่ ไม่ว่าจะรับรู้ในระดับ จิตสำนึก หรือระดับจิตใต้สำนึก. ดังนั้นอาจกล่าวได้ว่า การรู้จำรูปแบบ เป็นความสามารถที่สำคัญที่สุด อย่างหนึ่งของสติปัญญา.

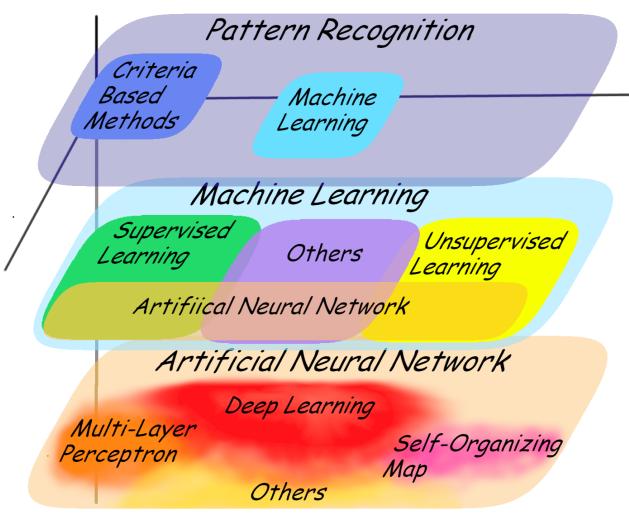
1.2 การรู้จำรูปแบบและการเรียนรู้ของเครื่อง

การรู้จำรูปแบบ (pattern recognition) โดยทั่วไป หมายถึง ภารกิจ เพื่อรับ�� ข้อมูลที่สำรวจมีสิ่งที่สนใจอยู่หรือไม่ หรือ เพื่อรับ知 ข้อมูลที่สำรวจเป็นสิ่งที่สนใจประเภทใด หรือ เพื่อรับ知 ว่า สิ่งที่สนใจอยู่ที่ได้ในข้อมูลที่สำรวจ หรือ เพื่อประเมินค่าที่สนใจออกมาจากข้อมูลที่สำรวจ เป็นต้น.

การรู้จำรูปแบบนั้น มีอยู่ทั่วไปในธรรมชาติ เป็นความสามารถของสิ่งมีชีวิต เป็นส่วนสำคัญในพฤติกรรมทางสังคม แต่ในที่นี้ การรู้จำรูปแบบ จะเจาะจงเฉพาะกับ วิธีการที่จะทำให้คอมพิวเตอร์ มีความสามารถในการรู้จำรูปแบบ. การรู้จำรูปแบบด้วยคอมพิวเตอร์ อาจทำได้หลายแนวทาง (รูป 1.1 แสดง

ภาพรวมของการรู้จำรูปแบบและความสัมพันธ์กับแนวทางต่าง).

แนวทางหนึ่งคือ แนวทางการกำหนดเกณฑ์ที่ชัดเจน (criteria-based or rule-based approach) รวมไปถึง การจับคู่กับแพลนแบบ (template matching). สำหรับบางงาน เกณฑ์แม้จะชัดเจน แต่อาจจะไม่เจาะจงที่ตัวรูปแบบเอง ตัวอย่างเช่น การค้นหารูปแบบที่ปรากฏบ่อย ๆ ในการศึกษาด้านพันธุกรรม บางครั้งอาจต้องการ豪โนมิฟ (motif) หรือ ลำดับของสารพันธุกรรมสายยาว ๆ ที่พบได้บ่อย ที่สุด ซึ่งเกณฑ์อาจจะระบุชัดเจน เรื่องความยาวของสายพันธุกรรม และเรื่องความถี่ที่ปรากฏ แต่ไม่ได้ระบุลำดับของรหัสพันธุกรรมที่ค้นหา.



รูปที่ 1.1 ภาพใหญ่การรู้จำรูปแบบและการเรียนรู้ของเครื่อง. ความสัมพันธ์ระหว่าง การรู้จำรูปแบบ, การเรียนรู้ของเครื่อง, การเรียนรู้ของเครื่องแบบต่าง ๆ, โครงข่ายประสาทเทียม, การเรียนรู้เชิงลึก มีความเกี่ยวพันธ์เชื่อมโยงกันในหลายมิติ. เส้นแบ่งระหว่างแต่ละแนวคิดอาจไม่ได้ชัดเจน โดยเฉพาะแนวคิดต่าง ๆ ภายใต้โครงข่ายประสาทเทียม ที่ในภาพ เน้นความไม่ชัดเจนนี้ด้วยขอบที่คลุมเครือ.

อย่างไรก็ได้ รูปแบบซึ่งเป็นการเข้าใช้โครงสร้าง อาจเป็นผลมาจากการความสัมพันธ์สำคัญ ที่เชื่อมโยงข้อมูลกับรูปแบบนั้น. ดังนั้น การรู้จำรูปแบบ ก็เปรียบเสมือนการเรียนรู้ความสัมพันธ์สำคัญ ที่เชื่อมโยงระหว่างข้อมูลน่าเข้าและรูปแบบที่สนใจนั้น. แนวทางหลักของการรู้จำรูปแบบที่จะอภิปรายในที่นี้ คือ แนวทางของการเรียนรู้ของเครื่อง. วิธีการเรียนรู้ของเครื่อง จะไม่ได้พึ่งกฎหรือเกณฑ์ที่ชัดเจน ดังแนวทางที่กล่าวไปข้างต้น แต่ใช้ความสามารถในการทำงานกับข้อมูลมาก ๆ ของคอมพิวเตอร์ ประกอบกับแบบจำลองทางคณิตศาสตร์ เพื่อช่วยในการค้นหา หรือช่วยในการเรียนรู้ความสัมพันธ์ระหว่างข้อมูล นำเข้า และรูปแบบที่มักเรียกว่า ข้อมูลนำออก โดยเฉพาะสำหรับความสัมพันธ์ที่มีลักษณะซับซ้อน และยากที่จะกำหนดเป็นกฎหรือเกณฑ์ดังกล่าว.

ลักษณะเด่นของวิธีการเรียนรู้ของเครื่อง อาจปรากฏชัดอยู่ในตัวอย่างงานของอาร์瑟 เซล (Arthur Samuel) ในปี ค.ศ. 1959 ที่ เขายืนโปรแกรมคอมพิวเตอร์เพื่อเล่นหมากออร์ส[174] โดยที่ตัวชามูเอลเองเล่นหมากออร์สไม่เก่งเลย. หากชามูเอลเขียนโปรแกรมด้วยแนวคิดดังเดิม เขายังต้อง

หัดเล่นหากซอฟต์แวร์ให้เก่ง และโปรแกรมวิธีเดินหากอย่างละเอียดให้กับคอมพิวเตอร์. ชามูเอลไม่ได้เลือกทำแบบนั้น เขาเลือกที่จะโปรแกรมคอมพิวเตอร์ ให้เล่นแข่งกันเอง และให้คอมพิวเตอร์เก็บผลว่า ตำแหน่งของหากอย่างไรที่เป็นตำแหน่งที่ดี ซึ่งนำไปสู่ชัยชนะ หรือตำแหน่งไหนเป็นตำแหน่งไม่ดี และมักจะทำให้แพ้ แล้วให้โปรแกรมเลือกเดินหากตามผลที่เก็บนั้น. หลังจากชามูเอลปล่อยให้โปรแกรมเล่นแข่งกันเองหลายมื่นกระดาน โปรแกรมเล่นหากซอฟต์ของชามูเอลก็สามารถเล่นหากซอฟต์ได้ดีมาก และเล่นได้ดีกว่าตัวของชามูเอลเอง. ซึ่งกรณีเช่นนี้ แทบจะเป็นไปไม่ได้เลยกับวิธีการเขียนโปรแกรมแบบดั้งเดิม. ดังนั้น ณ ตอนนั้น วิธีการสร้างโปรแกรมเล่นหากซอฟต์ของชามูเอล เป็นหนึ่งของการปฏิวัติแนวทางใหม่ๆ เลยทีเดียว. และนี่คือลักษณะเด่นของการเรียนรู้ของเครื่อง คือแทนที่จะเขียนโปรแกรมวิธีทำอย่างละเอียดให้คอมพิวเตอร์ กลับเขียนโปรแกรมให้คอมพิวเตอร์มีความสามารถในการเรียนรู้วิธีทำ สร้างสิ่งแวดล้อมให้คอมพิวเตอร์ได้เรียนรู้ แล้วปล่อยให้คอมพิวเตอร์เรียนรู้วิธีทำเอง.

อาร์เรอร์ ชามูเอล[174] ได้นิยามการเรียนรู้ของเครื่อง (*machine learning*) ไว้ว่า การเรียนรู้ของเครื่อง คือ ศาสตร์ของการทำให้คอมพิวเตอร์มีความสามารถที่จะเรียนรู้ได้ โดยที่ไม่ต้องเขียนโปรแกรมวิธีทำตรง ๆ. ทอม มิทเซลล์[131] ได้ช่วยขยายความ โดยนิยามว่า โปรแกรมคอมพิวเตอร์จะเรียกว่า มีการเรียนรู้จากประสบการณ์ E ซึ่งเกี่ยวข้องกับภารกิจ T และสมรรถนะของผลลัมภ์ที่วัดได้ P ก็ต่อเมื่อสมรรถนะของการทำภารกิจ T ที่วัดด้วย P ปรับปรุงขึ้นได้จากการประสบการณ์ E .

จากตัวอย่าง โปรแกรมเล่นหากซอฟต์ของชามูเอล ประสบการณ์ E คือ การได้เล่นแข่งเล่นแข่งกันเอง ภารกิจ T คือการเล่นหากซอฟต์ และสมรรถนะ P วัดได้จากการที่โปรแกรมเล่นชนะ.

ปัจจุบัน การเรียนรู้ของเครื่อง ถูกประยุกต์ใช้อย่างกว้างขวาง ในวงการธุรกิจ อุตสาหกรรม การทหาร วงการวิทยาศาสตร์ วิศวกรรม การแพทย์ การเกษตร บันเทิง ศิลปะ การกีฬา รวมถึงการประยุกต์ใช้ชีวิตประจำวัน ตัวอย่างเช่น การค้นหาข้อมูลบนเวป (web search), การกรองข้อมูลบนสื่อสังคมออนไลน์ (content filtering on social media), การตรวจสอบหารูปแบบการใช้บัตรเครดิตที่ผิดปกติ[165] ซึ่งอาจเนื่องมาจากการที่บัตรถูกขโมยไป, การบริหารการลงทุนทางการเงิน[198], งานแอพพลิเคชันที่ไม่สามารถโปรแกรมตรง ๆ ได้ (หรือ ทำได้ยากมาก) ได้แก่ ระบบอ่านลายมือเขียน[116], การควบคุมไฮลิคอปเตอร์ไร้นักบิน[42], การควบคุมหุ่นยนต์ที่มีการเครื่องไฟว์ที่ซับซ้อน[3], การบริหารจัดการทรัพยากรน้ำ[30], การปรับตั้งค่าของเวอร์ชั่ร์แมชีน[158], การพัฒนาระยนต์ที่ขับเคลื่อนได้เองโดยไร้คนขับ[226], การติดตามลักษณะโครงสร้างใต้น้ำอัตโนมัติ[124], การระบุหารังสีแกรมม่าจากข้อมูลกล้องโทรทัศน์[20], ระบบตรวจสอบการสั่นสะเทือนของแผ่นดินไหว[169], การหารูปแบบในข้อมูลชีวสารสนเทศ[107], การแปลภาษาอัตโนมัติ[44], ระบบรู้จำคำพูด[176], ระบบรู้จำความก้าวหน้าของครอตตันต์[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].

เกร็งความรู้ รูปแบบของลูกีเมียและยา raksha เรียบเรียงจากบางส่วนของ [29]. รูปแบบมักซ่อนความสัมพันธ์หรือกลไกที่สำคัญอยู่เบื้องหลัง. เจนเนต โรวลี่ (Janet Rowley) คุณแม่ลูกสี่ เลี้ยงลูกเป็นหลัก และทำงานเสริมกับโรงพยาบาลวิจัยมะเร็งอาร์กอน. เธอ ทำงานศึกษาด้วยอย่างเชลล์จากคนไข้ที่ป่วยด้วยโรคเลือดต่าง ๆ แล้วในช่วงต้นปี ค.ศ. 1972 เธอกล่าวถึงเกตพบรูปแบบผิดปกติในเซลล์ของคนไข้ที่ป่วยด้วยโรคลูกีเมียเย็บหัวพลันชนิดไม่มีลิออยด์ คือ ดูเหมือนว่า มีบางส่วนของโครโมโซมที่แปดและบางส่วนของโครโมโซมที่ย้ายสิบเอ็ด สลับกัน (เรียกว่า การสลับตำแหน่งทางพันธุกรรม หรือ translocation). โรวลี่ได้ใจมาก และคิดว่าเธออาจพบสาเหตุของลูกีเมีย ซึ่งเป็นมะเร็งเม็ดเลือดขาว. ณ เวลานั้น ถึงแม้วิการแพทย์จะรู้แล้วว่า เซลล์มะเร็งมักมีโครโมโซมที่แปรปรวนไป แต่ก็ยังไม่มีเครื่องตรวจรูปแบบที่เด่นชัด และส่วนใหญ่ (ณ ตอนนั้น) ก็เชื่อกันว่า โครโมโซมที่แปรปรวนไปเป็นผลมาจากการมะเร็ง ไม่ใช่เป็นสาเหตุของมะเร็ง. โรวลี่ได้เขียนบทความรายงานเรื่องนี้ไปที่วารสารการแพทย์นิวอิงแลนด์ ซึ่งเป็นวารสารชั้นนำ แต่กลับถูกปฏิเสธ ด้วยเหตุที่วารสารซึ่งแจ้งว่า สิ่งที่โรวลี่พูดไม่สำคัญ. โรวลี่ส่งบทความนั้นไปที่วารสารเล็ก ๆ แทน.

หลังจากนั้นไม่นาน โรวี่ ก็ได้ศึกษา เซลล์มะเร็งจากคนไข้ ที่ป่วยด้วยโรคลูคีเมียรึรังชนิดไม้อลอยด์ (Chronic Myelogenous Leukemia คำย่อ CML). แม้จะเป็นงานเสริม โรวี่ ก็สนุกับงานที่ทำมาก. เห็นภาพถ่ายของโครโมโซมจากเซลล์ของคนไข้ กลับไปดูที่บ้านด้วยในวันหยุด. ภาพถ่ายของโครโมโซม เป็นคู่ ๆ เมื่อൺปลาท่องโก๋ มีจุดร่วมกันตรงกลาง ๆ และมีแขนยื่นข้างบน มีขา y ยื่นข้างล่าง. โรวี่ ว่างภาพถ่ายกระจายเต็มโต๊ะอาหารที่บ้าน จนลูก ๆ ของเรอแซวว่า แม่กำลังเล่นกับตุ๊กตากระดาษอยู่. โรวี่ ดูภาพถ่ายเหล่านั้นอย่างละเอียด ซึ่งเป็นภาพที่ถ่ายเซลล์ที่ผ่านวิธีย้อมแบบใหม่ เธอพบว่าโครโมโซมที่เก้าในเซลล์มะเร็งยาวกว่า โครโมโซมที่เก้าที่พบในเซลล์ปกติ. ก่อนหน้านั้นมีนักวิจัยจากฟิลาเดลเฟียพบว่า โครโมโซมที่ยี่สิบสองจะสั้นผิดปกติ ในเซลล์จากผู้ป่วยลูคีเมียรึรังชนิดไม้อลอยด์ จนโครโมโซมที่ยี่สิบสองที่สั้นผิดปกติ ถูกเรียกว่า ฟิล่าเดลเฟียโครโมโซม.

สำหรับต่อ เรื่องป่าไม้ โครโนโชมที่ยังสืบทอดกันมา ไม่สามารถลับขึ้นส่วนกัน และขึ้นส่วนที่ลับกันยากไม่เท่ากัน จึงทำให้โครโนโชมที่เก้ายากขึ้น และโครโนโชมที่ยังสืบทอดสั่นลง. รวมถึงตรวจสอบเซลล์อื่น ๆ ที่ปกติของคนไข้ พบว่า เซลล์ปกติไม่ได้มีการลับขึ้นส่วนกันแบบนี้ มีพับแต่เฉพาะในเซลล์มะเร็ง. คราวนี้ รวมถึงส่งบทความไปตีพิมพ์กับวารสารเนเชอร์ (Nature) ซึ่งเป็นวารสารวิชาการทางวิทยาศาสตร์ชั้นนำ แม้วารสารเนเชอร์จะปฏิเสธرواลีในครั้งแรก แต่ด้วยหลักฐานที่เพิ่มขึ้น ที่สุด วารสารเนเชอร์ก็ยอมรับตีพิมพ์รายงานของرواลี. ต่อจากนั้น رواลีก็พึ่งการลับตามหน้างทางพันธุกรรมระหว่างโครโนโชมที่สืบทอดกัน และโครโนโชมที่สืบเจิด ในเซลล์มะเร็งจากผู้ป่วยคลีมีเยียบพลันชนิดปะรำอีโลไซติก.

โรวีส์งสัยว่า การลับตัวแทนทางพันธุกรรมน่ามีส่วนในการก่อมะเร็ง แต่ว่า ณ ตอนนั้น มันยากที่จะพิสูจน์ประเด็นนี้ แม้งานก่อนหน้านี้ของ เพย์ตัน รูส (Peyton Rous) ที่พบว่า ไวรัสสามารถก่อให้เกิดมะเร็งชาร์โคมาในไก่ ก็ยังไม่ได้รับการยอมรับเท่าที่ควร (ณ ตอนนั้น). อย่างไรก็ดี การค้นพบไวรัสก่อมะเร็งในสัตว์ เป็นเรื่องสำคัญ ที่จะช่วยไขปริศนาด้านกำเนิดมะเร็ง เพราะว่า ไวรัสที่รู้สึกษา ซึ่งเรียกว่า รูสชาร์โคมาไวรัส (Rous sarcoma virus คำย่อ RSV) รูสชาร์โคมาไวรัสมีสิ่นอยู่แค่สี่ชนิด ทำให้พื้นที่มีแนวทางค้นหา ว่ามีตัวไหนที่มีส่วนในการก่อมะเร็ง. สตีฟ มาร์ติน (Steve Martin) นักศึกษาจากมหาวิทยาลัยแคลิฟอร์เนีย ที่เบิร์คลี ได้แยกรูสชาร์โคมาไวรัสที่กล้ายพันธุ์อกมา และได้ขยายพันธุ์อกมาเป็นเซลล์ แต่เซลล์ที่ได้กลับไม่เป็นเซลล์มะเร็ง เพราะมีการกล้ายพันธุ์ในไวรัสของมาร์ติน. ตำแหน่งที่กล้ายพันธุ์ในไวรัสของมาร์ติน อยู่ในยีนที่เรียกว่า ชาร์ค (src). ยีนชาร์คที่สมบูรณ์จะทำให้เกิดเซลล์มะเร็ง ชาร์คจึงถูกเรียกว่า ยีนมะเร็ง (oncogene). การค้นพบยีนมะเร็งในรูสชาร์โคมาไวรัส อาจจะช่วยอธิบายและยืนยันการค้นพบของรูส แต่ยังไม่ได้ช่วยอธิบายสาเหตุของมะเร็งในมนุษย์สักเท่าไร

การค้นพบชาร์ค ทำให้ทีมวิจัยของฮาร์โอลด์ วาร์มูส (Harold Varmus) และเจ ไมเคิล บิชอฟ (J. Michael Bishop) ศึกษา และ สงสัยว่า แล้วชาร์คไปอยู่ในไวรัสได้อย่างไร ในเมื่อตัวไวรัสเองไม่ได้ต้องการยึนนี้เลย ไวรัสไม่ได้ต้องการชาร์คเพื่อ การยึดเซลล์ ไวรัสไม่ได้ต้องการชาร์คเพื่อการแบ่งเซลล์. วาร์มูสและบิชอฟคิดว่า ไวรัสน่าจะได้ชาร์คมาโดยบังเอิญ จากเซลล์ ในหนังสือเซลล์ที่มันเคยไปเยี่ยมมา ถ้าเป็นแบบนั้นจริง มันก็น่าจะมีชาร์คปราภากอยู่ในเซลล์ปกติ dav.

แต่ก็เกือบ ๆ สีปีหลังจากนั้น กว่าที่มีการพบรากคในเซลล์ปกติ ชาร์คในเซลล์ปกตินี้ เรียกว่า ซีชาร์ค (c-src สำหรับ cellular src) เพื่อระบุให้ต่างจาก วิชาร์ค (v-src) ที่เป็นยืนมะเร็ง. ปรากฏว่า ยืนซีชาร์คไม่ได้มีเฉพาะในไก่ แต่มีการพบรในสัตว์หลาย ๆ ชนิด รวมถึงมนุษย์ด้วย. การค้นพบนี้ ทำให้ วาร์มูสและบีชอบคิดต่อไปว่า ซีชาร์คคงมีหน้าที่สำคัญอะไรสักอย่างในเซลล์ปกติ และตอนที่ไวรัสได้ชาร์คไป อาจจะไปเปลี่ยนแปลงบางอย่างในชาร์ค จนทำให้มันกลายเป็นยืนมะเร็ง.

ชาร์คเป็นยืนแรก และหลังจากนั้นก็มีการค้นพบยืนมะเร็งจากไวรัสอื่น ๆ และเช่นเดียวกับวิชาร์ค ที่มีซีชาร์ค หลาย ๆ ยืนมะเร็ง ก็พบว่ามียืนแบบนั้น ๆ ได้ในเซลล์ปกติตัวอย่าง แล้วพบรในสัตว์หลายชนิด รวมถึงมนุษย์ด้วย. ยืนเหล่านั้น เช่น ยืนเอม วายซี (myc) ยืนเอปีเอล (abl) ยืนอาร์เออเอส (ras) ช่วยยืนยันว่า แนวคิดว่า ยืนมะเร็งของไวรัส มีที่มาจากการยืนของเซลล์ปกติ. ยืนแบบเดียวกับยืนมะเร็งแต่พบรในเซลล์ปกติ จะเรียกว่า ยืนก่อนมะเร็ง (proto-oncogene).

วีเอปีเอล (v-abl) ในไวรัสของหมู เป็นหนึ่งในยืนมะเร็งที่ค้นพบหลังจากยืนชาร์ค และยืนก่อนมะเร็ง ที่เป็นคู่ของมัน คือ ซีเอปีเอล (c-abl) ที่พบรได้ในเซลล์ปกติของหมู และก็ยังพบรได้ในเซลล์ปกติของมนุษย์ด้วย. ยืนซีเอปีเอล พบรในโคโรโนไซม ที่เก้าของมนุษย์ ซึ่ง เป็นโคโรโนไซมเดียวกับที่ไวรัสพบร การสลับตำแหน่งทางพันธุกรรมในผู้ป่วยลูกคีเมียฉบับพลับชนิดไม้อลloyd's. เรื่องนี้ทำให้ทีมนักวิจัยสนใจและสืบต่อไปที่โคโรโนไซมที่ยังส่อง จนพบว่า ในเซลล์มะเร็ง ยืนซีเอปีเอลได้รับผลกระทบจากโคโรโนไซมที่เก้า ไปอยู่โคโรโนไซมที่ยังส่อง และยังพยายามอยู่ตำแหน่งเดียวกันหมด ในเซลล์จากผู้ป่วยทั้งสิบเจ็ดคนที่ตรวจสอบ. ตำแหน่งที่รับผลกระทบจากโคโรโนไซมที่ยังส่อง แสดงถึงความเสี่ยงต่อการเกิดมะเร็ง ซึ่งเป็นปัจจัยต่อเอปีเอล หรือ bcr-abl) ซึ่งเมื่อเซลล์นำยืนไปสร้างโปรตีน จะได้โปรตีนที่ผิดปกติ โดยต่อสายโปรตีนจากซีเอปีเอล เข้ากับสายโปรตีนจากบีซีอาร์. ผลคือโปรตีนสายยาวพิเศษจากบีซีอาร์ต่อเอปีเอล.

หมายเหตุ ชีววิทยาบีดหลักกว่า กลไกหลักของชีวิตคือ ดีเอ็นเอจะถูกถอดรหัสเป็นโปรตีน เอ็นไซม์และสารเอนไซม์จะถูกแปลรหัสเพื่อไปสร้างโปรตีน. และโปรตีนเป็นเครื่องมือและกลไกหลักในการทำงานของชีวิต. ยืน ซึ่งเป็นลักษณะที่ถ่ายทอดทางพันธุกรรม จะถูกบันทึกไว้ด้วยดีเอ็นเอ. ถ้าเปรียบดีเอ็นเอเปรียบเหมือนโปรแกรมคอมพิวเตอร์จะ ประกอบด้วยตระรากของโปรแกรม ไวยากรณ์ของภาษา รวมถึงข้อคิดเห็น หรือ code comments) ยืนก็จะเปรียบเหมือนตระรากของโปรแกรม.

กลไกเบื้องหลังรูปแบบที่แสดง. จากการศึกษาพฤติกรรมของโปรตีน โปรตีนจากซีเอปีเอล จะเป็น.enzyme ไทโรซีนคิโนเนส (tyrosine kinase). เอนไซม์ไทโรซีนคิโนเนสทำหน้าที่เพิ่มฟอสเฟตให้กับโปรตีน. การเพิ่มหรือลดฟอสเฟตกับโปรตีน เป็นเสมือนการเปิดหรือปิดการทำงานของโปรตีน แต่เปิดหรือปิดขึ้นกับชนิดของโปรตีน. โปรตีนจากซีเอปีเอล (โปรตีนจากเซลล์ปกติ) จะไม่ค่อยทำงาน ในขณะที่ โปรตีนจากบีซีอาร์ต่อเอปีเอล (โปรตีนจากเซลล์มะเร็ง) จะทำงานเกือบทตลอดเวลา ทำงานเพิ่มฟอสเฟต. โปรตีนจากบีซีอาร์ต่อเอปีเอล จะเพิ่มฟอสเฟตไปให้กับ อาร์บีโปรตีน (Rb protein) ซึ่ง การเพิ่มฟอสเฟตมาก ๆ ให้กับอาร์บีโปรตีน เมื่อมีการปิดการทำงานของอาร์บีโปรตีน. อาร์บีโปรตีน ทำหน้าที่สำคัญในกระบวนการแบ่งตัวของเซลล์. เซลล์จะแบ่งตัวโดย การทำสำเนาดีเอ็นเอก่อน แล้วค่อยแบ่งตัว. กระบวนการนี้จะถูกควบคุมอย่างเป็นระเบียบ. อาร์บีโปรตีน ทำหน้าที่หยุดการสำเนาดีเอ็นเอในเซลล์. เอนไซม์ไทโรซีนคิโนเนส ทำงานมากเกินไป ส่งผลเท่ากับ การปิดการทำงานของอาร์บีโปรตีน. การปิดการทำงานของอาร์บีโปรตีน ส่งผลเท่ากับ การปิดกลไกควบคุมการแบ่งตัวของเซลล์. รูปแบบที่ผิดปกติในโคโรโนไซมที่ไวรัสพบรบนตัวกินเข้า เป็นสาเหตุของลูกคีเมีย. การสลับตำแหน่งทางพันธุกรรมทำให้เกิดยืนผิดปกติ. ยืนผิดปกติส่งผลให้เกิดเอนไซม์ผิดปกติ. เอนไซม์ผิดปกติส่งผลให้เกิดการปิดกลไกหยุดการแบ่งตัวของเซลล์ และท้ายสุด ส่งผลให้เกิดโรคลูกคีเมีย.

ความเข้าใจปัจจุบันนำไปสู่วิธีแก้ที่ดีกว่า. วิธีการที่การแพทย์ใช้กับมะเร็ง แต่เดิมมีอยู่สามแนวทางหลัก คือ การผ่าตัด การฉายรังสี และการจัดยาสารพัดอย่าง เพื่อจะฆ่าเซลล์มะเร็ง. ก่อนความรู้เกี่ยวกับมะเร็งข้างต้นนี้ วิธีการจัดยาไม่ได้เจาะจงเฉพาะกับเซลล์มะเร็ง ดังนั้นผลลัพธ์ก็ต่าง ๆ กันไป และมีผลข้างเคียงสูง.

ด้วยความเข้าใจกลไกและสาเหตุของลูกคีเมียเรือรังนินดไมอลลอด์ นักวิทยาศาสตร์ที่ปริษัทใหญ่กี สวิตเซอร์แลนด์ ได้แก่ นิก ไลดอน (Nick Lydon) และอเล็กซ์ มาเตเตอร์ (Alex Matter) คิดว่า ถ้ายืนมะเร็งสร้างเอนไซม์ผิดปกติออกมา เป็นสาเหตุของโรค. เอนไซม์ผิดปกตินี้ ทำงานมากเกินไป ดังนั้น สารที่ยับยั่งเอนไซม์นี้ได้ อาจช่วยหยุดการเติบโตของมะเร็งได้.

แทนที่ ไอลดอนกับ Mata เตอร์จะใช้วิธีค้นหาสารนี้จากธรรมชาติ หรือใช้วิธีลองผิดลองถูก ตามวิธีของการค้นหายา ณ ยุคหนึ่ง ไอลดอนกับ Mata เตอร์ใช้วิธีการออกแบบตามเหตุผล (rational design) ใช้กระบวนการทางเคมี เพื่อสังเคราะห์โมเลกุล ที่จะเข้าไปจับกับตำแหน่งของอคุทอร์ (active site) ของเอมไซม์ไฮโรซินคินเนสที่พิเศษ และหยุดการทำงานของไฮโรซินคินเนสที่พิเศษ.

หลังจากหลายปีที่ทำการทดลองทางเคมีและทดสอบ "ไอลดอนกับ Mata เตอร์" ได้สารต่าง ๆ ที่ผ่านการคัดเลือกเบื้องต้น มาจำนวนหนึ่ง และทั้งคู่ได้ติดต่อกับนายแพทย์บอัน ดรุกเกอร์ (Brian Druker) ซึ่งทำงานที่มหาวิทยาลัยวิทยาศาสตร์สุขภาพโอลรอกอน เพื่อทดสอบกับเซลล์จากผู้ป่วย. ดรุกเกอร์พบว่า มีสารอยู่ตัวหนึ่งจากที่ไอลดอนกับ Mata เตอร์ให้มา สามารถฆ่าเซลล์มะเร็งได้ โดยไม่ฆ่าเซลล์ปกติ เมื่อใช้ที่ความเข้มข้นต่ำ ๆ

กว่าจะได้เป็นยา. ดรุกเกอร์ ไอลดอน กับ Mata เตอร์ ได้มากกับผลที่ได้ แต่ผู้บริหารของไซบากิกลับไม่ค่อยสนใจมาก ไอลดอนกับ Mata เตอร์ ใช้เวลาเกือบปีในการโน้มน้าวผู้บริหาร ให้อนุมัติการทดสอบต่อในสัตว์. แต่ผลทดสอบพิชิตยาในสุนัข ทำให้นักพิชิตยาค่อนข้างเป็นห่วง เรื่องความปลอดภัยในมนุษย์. หลังจากนั้นไม่นาน บริษัทไซบากิกิจการเข้ากับบริษัทชานโดซ รวมเป็น บริษัทโนوار์ทิส. ไอลดอนลาออก. โนوار์ทิสทดสอบยาอีกรอบ แต่นักพิชิตยาเกี้ยงไม่สนับสนุน การทดลองยาในมนุษย์.

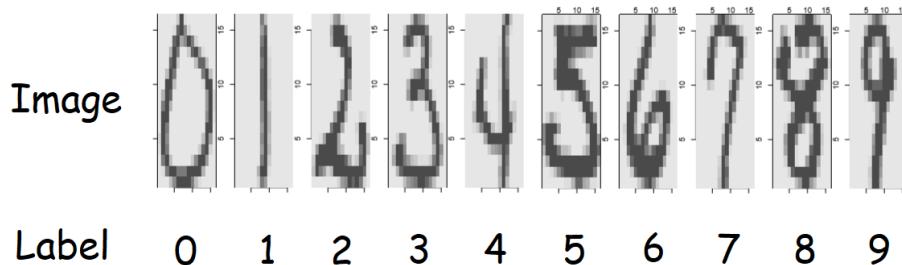
ดรุกเกอร์มองจากมุมที่ต่างไป. โอกาสครอบของคนไข้ที่ดรุกเกอร์เห็น มันรับรีมาก คนไข้รา ๆ 25 ถึง 50 เปอร์เซ็นต์ ตายภายในหนึ่งปี หลังจากพบว่าเป็นมะเร็ง และสิ่งที่ดรุกเกอร์ทำได้ ก็แค่ยื้อเวลาเท่านั้น ไม่สามารถรักษาได้เลย. ดรุกเกอร์คิดว่าพิชิตยา น่าจะพожัดการได้ โดยการติดตามผลที่ตัวคนไข้ และการปรับขนาดยา. ดรุกเกอร์ขอร้อง Mata เตอร์ และ Mata เตอร์ก็ยืนยันกับโนوار์ทิส ถึงความต้องการของยา จนในที่สุด ผู้บริหารยอมให้มีการทดสอบทางคลินิกในมนุษย์. การทดสอบเริ่มในปี ค.ศ. 1998 เกือบท้าปีหลังจากที่ดรุกเกอร์ได้ทดสอบยากับเซลล์มะเร็ง.

ทีมของดรุกเกอร์ทดสอบยา กับคนไข้ลูคีเมียเรื้อรังชนิดไม่อิลลอยด์จำนวนหนึ่ง โดยค่อนข้าง ฯ เพิ่มขนาดยา พร้อมกับติดตามอาการของโรคและผลข้างเคียง อย่างใกล้ชิด. ประสิทธิผลของยา วัดได้จากการลดจำนวนของเซลล์เม็ดเลือดขาว. ในคนปกติ เซลล์เม็ดเลือดขาวจะอยู่ที่ 4000 ถึง 6000 เซลล์ต่อเลือดหนึ่งมิลลิลิตร แต่ในผู้ป่วยลูคีเมียเรื้อรังชนิดไม่อิลลอยด์ เซลล์เม็ดเลือดขาวจะอยู่ที่ 100000 ถึง 500000 เซลล์ต่อเลือดหนึ่งมิลลิลิตร. ที่ยาปริมาณน้อย ๆ ทีมของดรุกเกอร์ไม่เห็นผลที่แตกต่าง แต่มีเพิ่มปริมาณยาขึ้น คนไข้บางคน เริ่มมีจำนวนเม็ดเลือดลดลงสูงมาก. พอนำมาเลือดของคนไข้ไปตรวจพบ ก็พบว่า สัดส่วนของเซลล์ที่มีพิลาเดลเพียโครโนไซม์ก็ลดลงด้วย. ยาทำงานได้ดี.

โนوار์ทิสสนับสนุนยาในขั้นตอนต่อมาอย่างเต็มที่. คนไข้ในโครงการถูกติดตามผลต่อไปอีกหลายเดือน. เก้าสิบเจ็ด เปอร์เซ็นต์ของคนไข้ที่ได้รับยาในขนาดสูงสุด มีจำนวนเม็ดเลือดขาวกลับสู่ระดับปกติ ภายในเวลาสี่สัปดาห์. เมื่อตรวจสอบบเซลล์จากคนไข้ ก็พบว่า คนไข้สามในสี่คน ไม่มีพิลาเดลเพียโครโนไซม์อีกแล้ว. ผลลัพธ์ดีเยี่ยม และดีที่สุด ในประวัติของการรักษามะเร็งด้วยการจัดยา. โนوار์ทิสยื่นจดทะเบียนยา ในชื่อ กลีเวค (Gleevec ชื่อสามัญ Imatinib) กับสำนักงานอาหารและยาของสหรัฐอเมริกา และภายใต้รับการรับรองในปี ค.ศ. 2001.

กลีเวคเปลี่ยนสถานการณ์ การรักษาลูคีเมียเรื้อรังชนิดไม่อิลลอยด์หน้ามือเป็นหลังมือ. โอกาสลดลงอย่างมาก (ยาวกว่า 8 ปี) เพิ่มขึ้นจากราว 45 เปอร์เซ็นต์ก่อนการรับรองกลีเวค ไปถึงเกือบ 90 เปอร์เซ็นต์ด้วยการใช้กลีเวค.

ปัจจุบัน เชื่อว่า ร่างกายมนุษย์ ประกอบด้วย เซลล์สองร้อยกว่าชนิด จากเซลล์ทั้งหมด จำนวนกว่าสามสิบเจ็ดล้านล้านเซลล์. จากยืนทั้งหมดรา ๆ สองหนึ่งยืนของมนุษย์ มีรา ๆ หนึ่งร้อยสี่สิบยืนที่มีกลไกพันธุ์ และอาจก่อให้เกิดมะเร็ง. ร้อยสี่สิบยืนเหล่านี้ เป็นส่วนหนึ่งในกระบวนการที่ควบคุมการเปลี่ยนสภาพ หรือการอยู่รอดของเซลล์. มะเร็งส่วนใหญ่จะเกี่ยวข้องกับการกลายพันธุ์ ส่องถึงแบดยืน จากหนึ่งร้อยสี่สิบยืน. การรักษาในส่วนนี้ ให้เราเข้าใจ และสามารถเชื่อมโยง การกลายพันธุ์ กับพฤติกรรมของมะเร็ง ไปจนถึงความสามารถทางวิธีรักษาตามการกลายพันธุ์นั้น ๆ ได้. ปัจจุบัน มีมากกว่าสามสิบชนิด ที่รักษามะเร็งตามการกลายพันธุ์ และก็ยังมีอีกมากที่อยู่ในกระบวนการวิจัย.



รูปที่ 1.2 ตัวอย่างรูปตัวเลขจากลายมือเขียน. แถวบนแสดงตัวอย่างข้อมูลนำเข้า ซึ่งเป็นภาพ. แถวล่างแสดงเฉลยของแต่ละภาพ ซึ่งเป็นฉลากของแต่ละรูปแบบ

เจเน็ต โรแลล เสียชีวิตในปีค.ศ. 2013 จากโรคมะเร็งรังไข่. ก่อนเสียชีวิต เธอได้ทำการนัดการผ่าตราช่องท้องเพื่อตรวจสภาพห้องท้องที่เธอเสียชีวิต เพื่อที่นักวิจัยจะได้ศึกษาโรคต่อไป.

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

---Sean B. Carroll

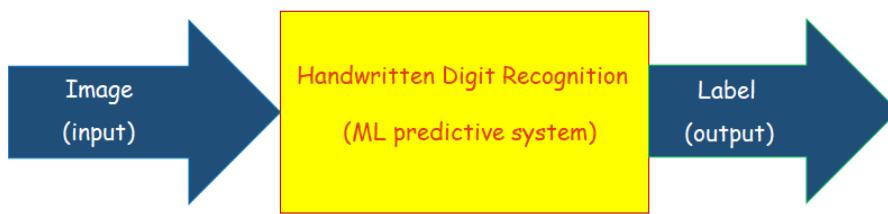
“สิ่งสำคัญที่สุดที่เราได้เรียนรู้เกี่ยวกับชีวิตมนุษย์ในระดับโมเลกุล คือ ทุก ๆ อย่างถูกควบคุม จัดระเบียบอย่างดี.”

—ฉอน บี คารอล

1.3 การรู้จำตัวเลขลายมือ

โปรแกรมรู้จำตัวเลขลายมือ^[115] เป็นตัวอย่างการรู้จำรูปแบบด้วยวิธีการเรียนรู้ของเครื่อง ที่นิยมอ้างถึงกันมาก เพราะภารกิจช่วยให้เข้าใจภาพรวมได้ดี และงานไม่ซับซ้อนเกินไป มีข้อมูลเข้าถึงได้ง่าย สามารถใช้เป็นตัวอย่างทดลองปฏิบัติได้. การรู้จำตัวเลขลายมือ (*handwritten digit recognition*) มีภารกิจ T คือ จากรูป (ข้อมูลสำรวจน์) ซึ่งคอมพิวเตอร์รอมองเห็นเป็นค่าความเข้มของพิกเซลต่าง ๆ แล้ว ให้โปรแกรม tahy ว่าภาพนั้นเป็นภาพแทนตัวเลขอะไร (ระบุประเภทรูปแบบ) โดยภาพของตัวเลข เป็นภาพลายมือเขียนตัวเลขต่าง ๆ จากเลข 0 ถึงเลข 9 ดังแสดงในรูปที่ 1.2. รูปตัวอย่างต่าง ๆ พร้อมเฉลย สามารถนำมาใช้ช่วยพัฒนาโปรแกรมได้ (ประสบการณ์ E). สมรรถนะ P วัดได้จากจำนวนรูปภาพที่ tally ได้ถูกต้อง.

รูปที่ 1.2 แถวบน แสดงตัวอย่างรูปภาพ ที่เป็นข้อมูลนำเข้า (หรืออินพุต input) ของโปรแกรมรู้จำตัวเลขลายมือ. แถวล่าง แสดงตัวอย่างฉลาก ของเฉลยสำหรับข้อมูลนำเข้าที่อยู่ด้านบน. ฉลาก (label) จะระบุประเภทของรูปแบบที่สนใจ ในกรณีนี้ มีสิบรูปแบบ. รูปแบบของเลขศูนย์ รูปแบบของเลขหนึ่ง รูปแบบของเลขสอง ไปจนถึง รูปแบบของเลขเก้า. เฉลย (ground truth) คือฉลากที่ถูกต้อง. เฉลย มีประโยชน์มาก โดยเฉพาะช่วยให้การวัดสมรรถนะทำได้ง่าย. สำหรับภาพได้ตาม หากฉลากที่ tally ตรงกับเฉลย ก็คือ tally ถูก และในทางตรงกันข้าม หากไม่ตรง ก็คือ tally ผิด.



รูปที่ 1.3 แผนภาพแสดงระบบรู้จำตัวเลขลายมือ โดยมีระบบประมวลผล (ที่ทำงานฉลาก โดยสร้างตามแนวทางของการเรียนรู้ของเครื่อง) รับข้อมูลนำเข้าเป็นภาพ และให้ข้อมูลนำออก ซึ่งเป็นฉลาก.

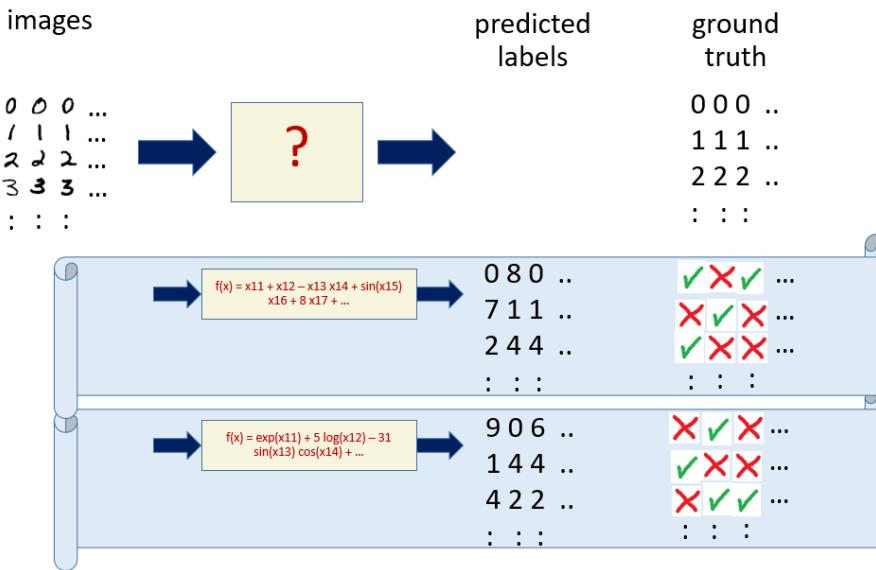
ฉลากที่ทายจากโปรแกรม บางครั้งอาจเรียกในชื่อที่ทั่วไปกว่า ว่า ข้อมูลนำออก (หรือเอาต์พุต output). จากรูปของระบบแล้ว ภาพ คือข้อมูลนำเข้า ระบบ(โปรแกรมการรู้จำตัวเลขลายมือ) รับข้อมูลนำเข้า ประมวลผล และให้ค่าข้อมูลนำออก ซึ่งคือฉลากของรูปแบบเลขออกมา. รูปที่ 1.3 แสดงแผนภาพโปรแกรมการรู้จำตัวเลขลายมือจากมุมมองระบบ.

จากรูปของนี้ ระบบประมวลผล f ทำหน้าที่แปลง ข้อมูลนำเข้า \mathbf{x} ที่เป็นภาพ ไปเป็นข้อมูลนำออก y ที่เป็นฉลาก. และ เพื่อให้เห็นภาพชัดเจน พิจารณาการรู้จำตัวเลขลายมือ ที่ออกแบบสำหรับ ชุดข้อมูลเออมนิสต์. ชุดข้อมูลเออมนิสต์^[115] (MNIST) เป็นข้อมูลขนาดใหญ่ของภาพพร้อมเฉลยของ ตัวเลขลายมือ เขียน ข้อมูลชุดนี้นิยมใช้ สำหรับทั้งศึกษาพัฒนาระบบประมวลผลภาพ และการเรียนรู้ ของเครื่อง โดยข้อมูลได้ปรับปรุงจากข้อมูลของสถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (National Institute of Standards and Technology) สหรัฐอเมริกา. ข้อมูลประกอบด้วย ภาพตัวเลขลายมือ เขียนจำนวน 70,000 ภาพ¹ พร้อมเฉลย แต่ละภาพมีขนาด 28×28 พิกเซล และเป็นภาพขาวดำสอง ระดับค่า (bi-level image) นั่นคือ แต่ละพิกเซลมีค่าเป็น 0 หรือ 1. ดังนั้น หากเขียน ระบบรู้จำ ตัวเลขลายมือเออมนิสต์นี้ เป็นฟังก์ชันคณิตศาสตร์ จะได้ $f : \mathbf{x} \mapsto y$ โดย $\mathbf{x} \in \{0, 1\}^{28 \times 28}$ และ $y \in \{0, 1, 2, \dots, 9\}$.

จากรูปของนี้ ปัญหาการสร้างระบบรู้จำตัวเลขลายมือ ถูกจำกัดกรอบลงมาเป็นการค้นหาฟังก์ชัน คณิตศาสตร์ f แทน. แนวทางการเรียนรู้ของเครื่อง ที่อาจทำได้คือ ใช้โปรแกรมค้นหาฟังก์ชัน f นี้. จากตัวอย่างข้อมูลภาพและเฉลยจำนวนมากที่มี โปรแกรมจะหาฟังก์ชันคณิตศาสตร์ที่สามารถแปลง จากรูปในตัวอย่างไปเป็นฉลากที่ถูกต้องได้มากที่สุด. รูปที่ 1.4 แสดงภาพตามแนวคิดนี้.

อย่างไรก็ตาม การค้นหาฟังก์ชันคณิตศาสตร์ได้ ๆ นั้นมีปริภูมิค้นหา (search space) ที่กว้าง ขวางมาก จนในทางปฏิบัติแล้ว วิธีนี้ทำงานไม่ได้เลย. วิธีแก้ปัญหาคือ แทนที่จะค้นหาฟังก์ชันคณิต- ศาสตร์ได้ ๆ แนวทางการเรียนรู้ของเครื่องที่ใช้งานได้ผล คือ จะเลือกฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์ (parametric model) ที่ พฤติกรรมการแปลงสามารถควบคุมได้ จากค่าของพารามิเตอร์ที่เลือกใช้ และ ใช้โปรแกรมค้นหาค่าของพารามิเตอร์แทน. นั่นคือ สมมติมีฟังก์ชันคณิตศาสตร์ $f(\mathbf{x}, \mathbf{w})$ ที่พฤติกรรม การแปลงค่าข้อมูลนำเข้า \mathbf{x} ไปเป็นข้อมูลนำออก y เปลี่ยนแปลงและควบคุมได้จากค่าพารามิเตอร์ (parameters) \mathbf{w} . ดังนั้นแทนที่จะให้โปรแกรมค้นหาสมการคณิตศาสตร์ได้ ๆ ที่เป็นได้ (ซึ่งมีจำนวนเกิน

¹ 60,000 ภาพสำหรับชุดฝึกหัด และ 10,000 ภาพสำหรับทดสอบ.



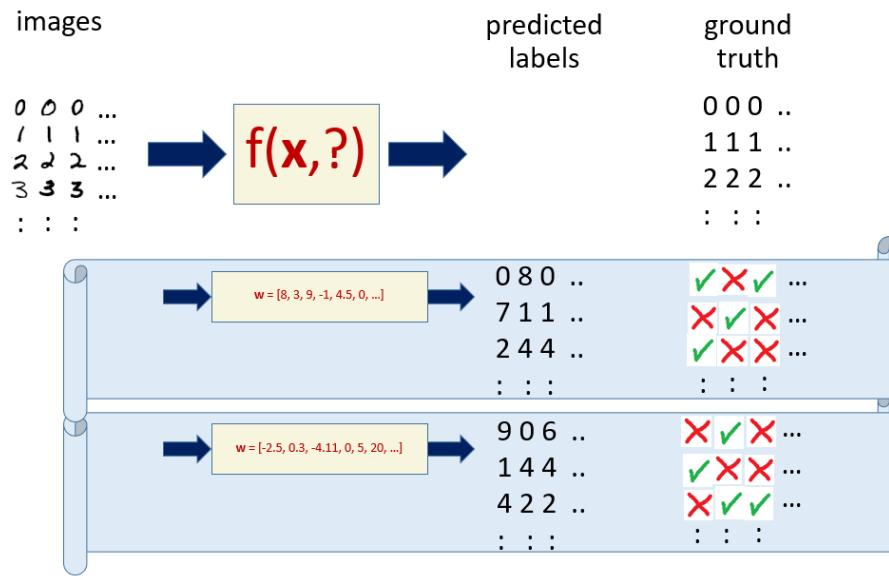
รูปที่ 1.4 แผนภาพการค้นหาฟังก์ชันรูจัตัวเลขหลายมีอ. ด้วยข้อมูลตัวอย่าง โปรแกรมค้นหาฟังก์ชันคณิตศาสตร์ ที่สามารถแปลงข้อมูลนำเข้า ไปเป็น ข้อมูลนำออกที่ตรงกับเฉลยมากที่สุด.

คณานับ) และการค้นหามีโอกาสสำเร็จน้อยมาก เปลี่ยนมาเป็น ให้โปรแกรมค้นหาค่าของพารามิเตอร์ w แทน จะช่วยลดขนาดของปริภูมิค้นหาลงมาหลัก และเพิ่มโอกาสสำเร็จขึ้นมาก. นอกจากนั้น หากฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์ ที่มักเรียกว่า **แบบจำลอง (model)** มีความสามารถในการปรับการแปลงมาก ๆ การเลือกค่าพารามิเตอร์ ก็สามารถจะให้ผลได้ใกล้เคียงกับการค้นหาฟังก์ชันคณิตศาสตร์ได้ ๆ ภายใต้บริบทของการกิจที่ทำงานอยู่. รูปที่ 1.5 แสดงแนวทางของการใช้ฟังก์ชันคณิตศาสตร์อิงพารามิเตอร์.

เนื่องจาก แบบจำลอง ทำหน้าที่แปลงจากค่าอินพุตไปหาค่าที่จะหมายสำหรับเอาต์พุต ดังนั้น จึงอาจมองว่า แบบจำลองหมายค่าเอาต์พุต จากค่าอินพุตได้. การรูจัรูปแบบ ก็อาจมองจากมุมนี้ได้ว่า คือ การหมาย (prediction ซึ่งบางครั้งเรียกว่า การอนุमาน inference) รูปแบบที่สนใจ (เอาต์พุต) จากข้อมูล (อินพุต). รายละเอียดของแบบจำลองสำหรับการรูจัตัวเลขหลายมีอเขียน และวิธีการหาค่าพารามิเตอร์ที่ทำงานได้ จะอธิบายโดยละเอียดในบทที่ 3.

1.4 ประเภทของการเรียนรู้ของเครื่อง

จากตัวอย่างการรูจัตัวเลขหลายมีอเขียน การวัดสมรรถนะสามารถทำได้ตรงมาตรงไป เพราะว่า รูปแบบของภาพตัวเลขหลายมีอเขียนมีเฉพาะ. การประยุกต์ใช้การเรียนรู้ของเครื่อง ไม่ได้จำกัดอยู่เฉพาะกับการกิจที่มีเฉพาะเท่านั้น. แต่การที่มีเฉพาะ ช่วยทำให้การวัดสมรรถนะสามารถทำได้อย่างตรงมาตรงไป. การกิจชนิดที่มีเฉพาะมาให้ด้วย จะเรียกว่า **การเรียนรู้แบบมีผู้สอน (supervised learning)**. การเรียนรู้แบบมีผู้สอนเอง ก็ยังอาจจำแนกออกได้เป็นหลายประเภท ส่วนใหญ่นิยมจำแนกตามลักษณะข้อมูลนำ



รูปที่ 1.5 แผนภาพการค้นหาฟังก์ชันรู้จำตัวเลขหลายมือ. ด้วยข้อมูลตัวอย่าง โปรแกรมค้นหาค่าพารามิเตอร์ของแบบจำลอง แทนการค้นหาฟังก์ชันคณิตศาสตร์.

ออก. หากข้อมูลนำออกเป็นการทายฉลาก หรือทายค่าวิยุต (*discrete value*) ที่มีจำนวนจำกัด นั่นคือ ข้อมูลนำออก $y \in \{\alpha_1, \alpha_2, \dots, \alpha_K\}$ เมื่อ K แทนจำนวนค่าวิยุตทั้งหมดที่เป็นไปได้ และ α_i แทนค่าวิยุตต่าง ๆ ($i = 1, \dots, K$) ดังเช่น การทายฉลากของตัวเลขหลายมือ $y \in \{0, 1, \dots, 9\}$ กลุ่มนี้จะเรียกว่า ภารกิจการจำแนกกลุ่ม (*classification*). แต่หากข้อมูลนำออกเป็นการทายค่าต่อเนื่อง (*continuous value*) นั่นคือ ข้อมูลนำออก $y \in \mathbb{R}$ ดังเช่น การทายค่าตั้งนีกิจการเติบโตทางเศรษฐกิจ ซึ่งอาจเป็น 3.2 หรือ 4.5 หรือ ค่าจำนวนจริงใด ๆ (ซึ่งคงไม่เกิน 20 และหวังว่าจะไม่เป็น 0 หรือติดลบ) กลุ่มนี้จะเรียกว่า ภารกิจการหาค่าถดถอย (*regression*). ค่าเฉลยสำหรับภารกิจการหาค่าถดถอย ก็จะเป็นค่าจำนวนจริงใด ๆ (ไม่ใช่ฉลาก แบบการรู้จำตัวเลขหลายมือ).

หากภารกิจที่ทำไม่มีเฉลยจริง ๆ เลย ภารกิจนี้จะเรียกว่า 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) ที่เหมาะสมกับเนื้อหาข้อความ, การลดมิติของข้อมูล (*dimension reduction*) ซึ่งคือ การลดจำนวนตัวแปรหรือส่วนประกอบของแต่ละจุดข้อมูลลง เพื่อให้การประมวลผลสามารถดำเนินการได้สะดวกรวดเร็วขึ้น, การอนุมานข้อมูลขึ้นใหม่ (*generative model*) ที่สามารถใช้สร้างข้อมูลขึ้นมาใหม่ในรูปแบบเดิม หรือเปลี่ยนรูปแบบใหม่ในบริบทเดิม ซึ่งนำไปสู่การซ้อม การ

สร้าง หรือการดัดแปลง ภาพ เพลง ข้อความ ไปจนถึงวิดีโอต่าง ๆ ซึ่งกำลังได้รับความสนใจอย่างมาก จากการศิลปะ ดนตรี บันเทิง และการออกแบบ แต่ก็เป็นส่วนที่สร้างความกังวลไม่น้อยให้กับวงการ สื่อสารมวลชน กว้างใหญ่ และการพิสูจน์หลักฐาน การเรียนรู้คุณลักษณะตัวแทน (*representation learning*[14]), การตรวจหารูปแบบใหม่ (*novelty detection*[152]) และการตรวจหารูปแบบผิดปกติ (*anomaly detection*[31]) เป็นต้น. ภารกิจที่ไม่มีเฉลยนี้ครอบคลุมกว้างขวางมาก ซึ่งอาจรวมไปถึง การหาค่าดีที่สุดด้วยวิธีการค้นหาเชิงคึกคักสำนึกร (optimization with heuristic search) ด้วย. การหาค่าดีที่สุดด้วยวิธีการค้นหาเชิงคึกคักสำนึกร คือการประยุกต์ใช้อย่างกว้างขวางมาก และมีรูปแบบ วิธีการที่หลากหลาย อาทิ วิธีซิมูเลเต็ดแอนนิลลิ่ง (*simulated annealing*[112]) และ จีโนติกอัลกอริทึม (*genetic algorithm*[216]) เป็นต้น.

นอกจากการเรียนรู้แบบมีผู้สอนที่มีเฉลยชัดเจน และการเรียนรู้แบบไม่มีผู้สอนที่ไม่มีเฉลยเลย ยังมี การกิจกรรมหลายประเภทที่ไม่อาจจัดอยู่ในสองกลุ่มนี้ง่ายๆ ได้ เช่น การเรียนรู้แบบกึ่งมีผู้ช่วยสอน (*semi-supervised Learning*) ที่เป็นลักษณะภารกิจที่มีเฉลย แต่ข้อมูลที่ได้ มีทั้งส่วนที่มีเฉลย และส่วนที่ไม่มีเฉลย และยังต้องการใช้ข้อมูลที่มีอยู่ให้คุ้มค่า โดยไม่ทิ้งข้อมูลที่ไม่มีเฉลยไปเปลย ฯ. การเรียนรู้การแนะนำลินค้า (*recommendation learning*[166, 191]) ที่สามารถใช้ผลการประเมินความพอใจจาก ตัวลูกค้าเอง กับสินค้าบางรายการ ประกอบกับ ผลประเมินจากลูกค้าคนอื่น ๆ เพื่อประเมินความชอบ ของลูกค้า กับสินค้ารายการที่ลูกค้าไม่ได้ประเมิน. การเรียนรู้การแนะนำลินค้า อาจมีลักษณะคล้าย ๆ การเรียนรู้คุณลักษณะตัวแทน ที่พยายามเรียนรู้คุณลักษณะภายในตัว ฯ ของสินค้าที่ลูกค้าชอบ แต่การ เรียนรู้การแนะนำลินค้า มีการใช้ค่าเฉลยของบางคนกับบางรายการ และไม่ได้มีค่าเฉลยของทุกคน ทุกรายการ เพื่อไปทำนายความพอใจของลูกคันทุกคน ในทุกรายการที่ไม่มีผลเฉลยได้. การเรียนรู้ แบบเสริมกำลัง (*reinforcement learning*) ที่เป็นภารกิจการตัดสินใจในแต่ละคาบเวลา ซึ่งอาจจะ สามารถเห็นผลระยะสั้นได้ (การเรียนรู้แบบเสริมกำลังแบบสังเกตได้สมบูรณ์ หรือ *fully observable reinforcement learning*) หรืออาจต้องประมาณผลระยะสั้นด้วย ซึ่งอาจจะประมาณผลระยะสั้นบาง ส่วน หรืออาจจะต้องประมาณผลระยะสั้นทั้งหมดเลย (การเรียนรู้แบบเสริมกำลังแบบสังเกตได้บางส่วน หรือ *partially observable reinforcement learning*) แต่เป้าหมายของภารกิจจริง ๆ คือการได้ผล ประโยชน์ระยะยาวที่ดี หรืออาจเป็นการหาสมดุลที่ดีระหว่างผลประโยชน์ระยะสั้น และผลประโยชน์ ระยะยาว ซึ่งแม้จะมีผลลัพธ์ระยะสั้นมาให้สังเกตได้ แต่การประเมินผลประโยชน์ระยะยาวก็ไม่ได้ตรง มาตรฐาน และไม่มีเฉลยจริง ๆ ของการตัดสินใจต่าง ๆ ให้ตรวจสอบ. การเรียนรู้แบบเสริมกำลังที่ดี จะต้องรักษาสมดุลระหว่างการเลือกการกระทำเพื่อที่จะได้ผลที่ดูเหมือนดีที่สุด กับการเลือกการกระ ทำการเพื่อเรียนรู้ผลจากการกระทำต่าง ๆ ในสถานะการณ์ต่าง ๆ. ประเด็นความสมดุลนี้เรียกว่า ประเด็นของ การใช้งานและการเรียนรู้ (*issue of exploitation and exploration*). ลักษณะเด่นชัดอีกอย่าง คือ การที่ระบบการเรียนรู้แบบเสริมกำลัง มีปฏิสัมพันธ์กับสิ่งแวดล้อม หรือกล่าวได้ว่า ผลของการกระทำที่ ระบบเลือกมีผลต่อประสบการณ์ที่ระบบจะเรียนรู้ (ดู [104] หรือ [194] สำหรับรายละเอียดเพิ่มเติม).

1.5 การเรียนรู้ของเครื่องและศาสตร์ที่เกี่ยวข้อง

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

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

แม้ว่าปัจจุบัน โดยเฉพาะในวงการธุรกิจมักใช้ คำว่าการเรียนรู้ของเครื่องและคำว่าปัญญาประดิษฐ์ แทนกัน. อย่างไรก็ตาม ปัญญาประดิษฐ์ เน้นที่เป้าหมาย แต่ไม่ได้กำหนดวิธีการ และวิธีการหลาย ๆ อย่างของปัญญาประดิษฐ์ ไม่ได้สามารถจัดเป็นการเรียนรู้ของเครื่อง ในขณะที่การเรียนรู้ของเครื่อง มีความหมายที่เน้นถึงแนวทางวิธีการที่จะทำการกิจที่ต้องการ. และแม้ศาสตร์และศิลป์ปัจจุบันของการเรียนรู้ของเครื่อง จะได้สร้างความตื่นตัวอย่างมากกับสังคม แต่ก็ยังไม่อาจนำปัญญาประดิษฐ์ไปสู่ศักยภาพสูงสุด ซึ่งคือ การสร้างสติปัญญา rationale ได้ โดยเฉพาะ เรื่องสามัญสำนึก (common sense) เรื่องการเข้าใจภาษาธรรมชาติ (natural language understanding) เรื่องการเข้าใจความหมายระดับสูง เข้าใจสิ่งที่เป็นนามธรรม เป็นต้น.

การทำเหมืองข้อมูล (data mining) เป็นกระบวนการค้นหารูปแบบจากฐานข้อมูลขนาดใหญ่ ซึ่งมีหลายแบบมุ่งที่คล้ายกับการเรียนรู้ของเครื่อง โดยเฉพาะหลาย ๆ วิธีการของการการทำเหมืองข้อมูล ก็เป็นวิธีการเดียวกับวิธีการที่ใช้ในศาสตร์การเรียนรู้ของเครื่อง. ในมุมมองหนึ่ง การทำเหมืองข้อมูล จะเน้นที่ รูปแบบที่จะได้มาจากการเรียนรู้ ซึ่งโดยส่วนใหญ่ ข้อมูลก็จะอยู่ในรูปของฐานข้อมูลแบบลัมพันธ์ (relational database) ในขณะที่การเรียนรู้ของเครื่องจะเน้นที่วิธีการ หรือมักเรียกว่า ขั้นตอนวิธี (algorithm) มากกว่า. อย่างไรก็ตาม ในหลาย ๆ ภารกิจ มันก็อาจยากที่จะวางแผนแบ่งที่ชัดเจนได้ แต่ ก็มีงานบางอย่างที่แสดงลักษณะเด่นของการทำเหมืองข้อมูล เช่น การหากฎความสัมพันธ์ (association rules) และงานบางอย่างที่แสดงลักษณะเด่นของการเรียนรู้ของเครื่อง เช่น การเรียนรู้แบบเสริมกำลัง.

อีกประเด็นหนึ่งที่อาจจะเป็นจุดต่างที่สำคัญ คือ ในขณะที่การเรียนรู้ของเครื่อง จะเน้นที่การค้นหารูปแบบโดยอัตโนมัติอย่างชัดเจน แต่การทำเหมืองข้อมูลนั้นอาจทำโดยอาศัยมนุษย์เป็นหลัก หรือใช้มนุษย์อยู่ในกระบวนการทำเหมืองข้อมูลอย่างมากได้. ตัวอย่างเช่น ในการหากฎความสัมพันธ์นั้น ขั้นตอนวิธีการหากฎความสัมพันธ์ อาจจะช่วยหาความสัมพันธ์ระหว่างสินค้าต่าง ๆ ที่ซื้อด้วยกันได้ จากความถี่ที่สินค้าเหล่านั้นปรากฏอยู่บ่อย ๆ ในรายการซื้อเดียวกัน. แต่หากจะใช้หาความสัมพันธ์ระหว่างคุณลักษณะของพนักงาน กับพฤติกรรมการทำงาน อาจจะต้องอาศัยมนุษย์ช่วยกลั่นกรองความลัมพันธ์ ที่ไม่เป็นสาระ (trivial association) ออก เช่น ความสัมพันธ์ที่พบว่าพนักงานที่ทำงานน้อยกว่าสาม

เดือนไม่เคยลากิจ ซึ่งเหตุผลจริง ๆ เป็นเพราะว่า เขายังไม่มีสิทธิ์ แต่หากสรุปผลไปผิดว่า พนักงานใหม่ ขยันกว่า เพราะไม่เคยลากิจเลย ซึ่งอาจทำให้เกิดการเข้าใจผิดได้ หรือ ความสัมพันธ์ที่พบว่าพนักงานที่ลากคลอดทั้งหมดเป็นผู้หญิง ซึ่งแม้เป็นความจริง แต่ก็ไม่ได้มีสาระประโยชน์อะไร จึงจำเป็นต้องอาศัยมนุษย์ช่วยกลั่นกรองรูปแบบความสัมพันธ์ที่พบ.

นอกจากปัญญาประดิษฐ์ และการทำเหมืองข้อมูลแล้ว ยังมีศาสตร์อื่นอีกที่มีความหมายทับซ้อน คุณมีเครือข่ายกับการเรียนรู้ของเครื่อง. **วิทยาการข้อมูล (data science)** รวมศาสตร์ต่าง ๆ เพื่อวิเคราะห์ ข้อมูล ทำความเข้าใจเรื่องราว และทำนายประเด็นที่สนใจ ไปจนถึงแสดงข้อมูล แสดงมุมมองและนำเสนอผลวิเคราะห์. ด้วยลักษณะของวิทยาการข้อมูล วิทยาการข้อมูลครอบคลุมเนื้อหาส่วนหนึ่งของสถิติศาสตร์, การเรียนรู้ของเครื่อง, การทำเหมืองข้อมูล, การจัดการฐานข้อมูล และการสร้างมโนภาพสำหรับข้อมูลและสารสนเทศ (*data and information visualization*) รวมถึงเทคโนโลยีต่าง ๆ ที่ใช้จัดการข้อมูลปริมาณมหาศาล เช่น แมปเรดิวซ์ (MapReduce).

หมายเหตุ การแบ่งแยกหรือจัดฉลากสำหรับศาสตร์ต่าง ๆ เหล่านี้ไม่ได้มีเส้นแบ่งที่ชัดเจน และในทางปฏิบัติ ก็ไม่ได้มีเส้นแบ่ง หรือไม่ได้มีข้อจำกัด หรือไม่ได้มีความจำเป็นใดที่ต้องแบ่งให้เด็ดขาด. ภารกิจที่ทำ เป็นสิ่งสำคัญที่สุด. นั่นหมายถึงว่า เทคนิคใด ๆ ก็ตามที่เป็นประโยชน์ ที่ใช้งานได้ ที่เหมาะสมกับงาน ถือว่าดีทั้งนั้น ไม่ว่ามันจะเรียกรือจัดเป็นศาสตร์ใด หรือแม้แต่มันจะเป็นแนวทางใหม่ที่อาจยากที่จะถูกจัดให้อยู่ภายใต้ศาสตร์ใดก็ตาม.

บางครั้ง การเรียนรู้ของเครื่อง ถูกสับสนกับการเรียนรู้เชิงลึก. **การเรียนรู้เชิงลึก (deep learning)** เป็น การเรียนรู้ของเครื่อง ที่เน้นการใช้แบบจำลอง ที่มีความสามารถในการแปลงข้อมูลสูง (model with high representative power) โดยใช้การประมวลผลเป็นลำดับชั้น เรียกว่า แบบจำลองเชิงลึก (deep model) หรือโครงข่ายเชิงลึก (deep network). ความสามารถของแบบจำลองเชิงลึก ได้มาจากการที่แบบจำลองมีโครงสร้างที่มีการคำนวณในลักษณะเป็นชั้น ๆ ลำดับชั้น. ผลจากชั้นหนึ่งส่งไปคำนวณต่อที่อีกชั้นหนึ่ง และทำการคำนวณเช่นนี้ต่อไปหลาย ๆ ชั้น (ที่มาของคำว่า ลึก). บท 5 อภิปรายการเรียนรู้เชิงลึก ในรายละเอียด.

นอกจากศาสตร์ต่าง ๆ ดังกล่าวแล้ว ประเด็นของข้อมูลหัว (big data) เป็นอีกหนึ่งเรื่องที่มักถูกสับสนกับการเรียนรู้ของเครื่อง. ข้อมูลหัว อ้างถึงชุดข้อมูลที่มีปริมาณข้อมูลขนาดใหญ่ และมีความหลากหลายของชนิดข้อมูล โดยลักษณะสำคัญของข้อมูลที่เป็นข้อมูลหัว คือ ข้อมูลมีปริมาณมาก ข้อมูลเพิ่มขึ้นอย่างรวดเร็ว และข้อมูลมีความหลากหลายมาก (ซึ่งมักถูกอ้างถึง โดยย่อว่า 3 Vs สำหรับ high volume, high velocity และ high variety). เมื่อเปรียบเทียบข้อมูลหัวกับการเรียนรู้ของเครื่อง กล่าวโดยง่าย คือ ในขณะที่ข้อมูลหัวเน้นที่ลักษณะและความท้าทายของการจัดการกับข้อมูลในเชิงปริมาณ การเรียนรู้ของเครื่องเน้นที่ภารกิจที่จะทำ โดยมักใช้ข้อมูลประกอบ เพื่อการบรรลุภารกิจ. การทำข้อมูลหัว อาจต้องการเพียงฮาร์ดแวร์ ระบบฐานข้อมูล รวมถึงโครงสร้างข้อมูลที่มีประสิทธิภาพขึ้น เพื่อรับความท้าทายเชิงปริมาณของข้อมูล. การทำข้อมูลหัว อาจจะใช้หรือไม่ใช้แนวทางการเรียนรู้ของเครื่องก็ได้ ขึ้นกับจุดประสงค์. การเรียนรู้ของเครื่องเอง เมื่อใช้งานกับข้อมูลที่มีลักษณะข้อมูลหัว

อาจต้องการเทคนิคและกลไกที่ช่วยจัดการความท้าทายเชิงปริมาณ และอาจต้องการขั้นตอนวิธีใหม่ ที่เหมาะสมกับกับปริมาณ ความเร็ว และความหลากหลายของข้อมูลหัก. หากเปรียบเทียบข้อมูลหักเป็นถนนรุกรังที่ยาวมาก ๆ การเรียนรู้ของเครื่องก็อาจเปรียบเป็นรถยนต์เบนซิน. บางครั้งก็ทำงานด้วยกัน บางครั้งก็ไม่. แต่เมื่อทำงานด้วยกัน หากเปลี่ยนเป็นยางสำหรับถนนรุกรัง เปลี่ยนช่วงล่างให้หนทางขึ้น และเตรียมน้ำมน้ำเชื้อเพลิงเพื่อไว้ให้เพียงพอ อาจจะช่วยให้ขับผ่านไปสู่เป้าหมายได้โดยสวัสดิภาพ.

เนื้อหาของตำราเล่มนี้ เน้นพื้นฐาน และศาสตร์และศิลป์ที่สำคัญของการเรียนรู้ของเครื่อง ซึ่งแม้ หลาย ๆ เรื่อง จะเป็นเนื้อหาของศาสตร์อื่น ๆ เช่นกัน แต่ตำนานี้ไม่ได้มีจุดประสงค์เพื่อ ครอบคลุม ปัญญาประดิษฐ์ การทำเหมืองข้อมูล หรือศาสตร์อื่น ๆ ที่เกี่ยวข้อง. ผู้อ่านที่สนใจศาสตร์ที่เกี่ยวข้อง เหล่านี้ สามารถศึกษาได้จากตำรา และแหล่งเรียนรู้เฉพาะของแต่ละศาสตร์.

1.6 อภิธานศัพท์

รูปแบบ (pattern): การจำแนกโครงสร้าง

การรู้จำรูปแบบ (pattern recognition): การทายค่าหรือระบุฉลากของรูปแบบ จากข้อมูลนำเข้า

การเรียนรู้ของเครื่อง (machine learning): ศาสตร์ของการทำให้คอมพิวเตอร์มีความสามารถที่จะ เรียนรู้ ที่จะทำนาย หรือตัดสินใจได้ โดยที่ไม่ต้องเขียนโปรแกรมวิธีการทำตรง ๆ

การรู้จำตัวเลขลายมือ (handwritten digit recognition): โปรแกรมทายภาพ ว่าภาพนั้นแทนตัว- เลขออะไร โดยภาพของตัวเลข เป็นภาพลายมือเขียนตัวเลขต่าง ๆ จากเลข 0 ถึงเลข 9

ข้อมูลนำเข้า หรืออินพุต (input): ตัวแปรต้น หรือข้อมูลที่โปรแกรมรับเข้า

ข้อมูลนำออก หรือเอาต์พุต (output): ตัวแปรตาม หรือค่าข้อมูลที่โปรแกรมต้องให้ออกมา

ฉลาก (label): ตัวแปรตาม หรือข้อมูล ที่ระบุประเภท หรือชื่อของรูปแบบที่สนใจ

เอมนิสต์ (MNIST): ข้อมูลขนาดใหญ่ของภาพพร้อมฉลากของตัวเลขลายมือเขียน ซึ่งนิยมใช้ทดสอบ ระบบการรู้จำตัวเลขลายมือ

แบบจำลอง (model): สมการคณิตศาสตร์ที่ใช้คำนวนค่าข้อมูลนำออก จากค่าข้อมูลนำเข้า และค่า พารามิเตอร์ที่เลือกใช้ ซึ่งข้อมูลนำออกนักเป็นค่าทำนายของสิ่งที่สนใจ

พารามิเตอร์ (parameter): ตัวแปรที่ค่าของมัน สามารถปรับเปลี่ยนการทำนายของแบบจำลอง โดย เปลี่ยนพฤติกรรมการแปลงค่าข้อมูลนำเข้าไปเป็นข้อมูลนำออก

การเรียนรู้แบบมีผู้สอน (supervised learning): ภารกิจการทำนายหรือตัดสินใจ ที่มีฉลากที่ถูกต้อง ให้

การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning): ภารกิจการทำนายหรือตัดสินใจ ที่ไม่มีฉลาก ที่ถูกต้องให้

การจำแนกกลุ่ม (classification): ภารกิจการทำนายฉลาก หรือทำนายค่าวิมุติที่มีจำนวนจำกัด

การหาค่าทดแทน (regression): ภารกิจการทำนายค่าที่เป็นจำนวนจริง

1.7 แบบฝึกหัด

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

“ออกเรือไปจากอ่าวที่ปลอดภัย
การใบไปรับลมสำเภา
ออกสำรวจ ออกผัน ออกค้นพบ.”
—マーク・吐温

เพื่อเป็นการทบทวนทักษะการเขียนโปรแกรม แบบฝึกหัดทบทวนการเขียนโปรแกรมทั่วไป. แม้ ตำรา จะอภิปรายเนื้อหา ศาสตร์การรู้ จำรู้ แบบและการเรียนรู้ของเครื่อง โดยทั่วไป แต่เพื่อให้ผู้อ่าน เข้าใจอย่างชัดเจน ตัวอย่างโปรแกรมที่ใช้จะแสดงด้วยภาษาไพธอน (เวอร์ชันสาม). แบบฝึกหัดเขียน โปรแกรมนี้ออกแบบมา เพื่อทบทวนทักษะการเขียนโปรแกรมด้วยภาษาไพธอน

แบบฝึกหัด 1.1

จะเขียนโปรแกรมเพื่อพิมพ์ข้อความต่อไปนี้ ออกมากที่หน้าจอ โดยให้มีการขึ้นบรรทัดตามที่แสดง

Bruce Lee:

Knowing is not enough, we must apply.
Willing is not enough, we must do.

คำใบ้ ลองคำสั่ง **print**

แบบฝึกหัด 1.2

จะเขียนโปรแกรมเพื่อรับตัวเลขจำนวนเต็มจากผู้ใช้ และพิมพ์ตัวเลขนั้น พร้อมค่ากำลังสองของมัน ดังตัวอย่าง

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 ค.ศ. 2000) ซัค โนแลนด์ รอดชีวิตจากเครื่องบินตก และติดอยู่ที่เกาะร้าง เขายังคงคำนวณหาโอกาส ที่ทีมค้นหาจะพบเขาที่เกาะร้าง ดังนี้ (1) เครื่องบินด้วย

ความเร็ว v ไมล์ต่อชั่วโมง. (2) เครื่องบินติดต่อกับหอควบคุมการบินไม่ได้ เป็นเวลา T ชั่วโมงก่อนจะตก. ซึ่งต้องการคำนวนพื้นที่ที่ทิมภัยต้องค้นหา.

จงเขียนโปรแกรม เพื่อคำนวนพื้นที่ค้นหา โดยรับความเร็วเครื่องบิน v ไมล์ต่อชั่วโมง และเวลา T ชั่วโมง จากที่ขาดการติดต่อจนถึงเครื่องตก. โปรแกรมรายงานออกมานเป็นพื้นที่ตารางไมล์ และเปรียบเทียบกับพื้นที่ของประเทศไทย โดยพื้นที่ประเทศไทย มีขนาดประมาณ 513120 ตารางกิโลเมตร หรือ 198120 ตารางไมล์.

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 เป็นผลการคำนวน

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

แบบฝึกหัด 1.4

จงเขียนฟังก์ชันคำนวนเวลาที่ลูกเทนนิสวิ่งจากหน้าไม้ของผู้เชิร์ฟไปถึงท้ายสนามเทนนิสฝั่งผู้รับ และคำนวนพลังงานที่ใช้ในการเชิร์ฟ ในหน่วยจูล (Joules ตัวย่อ J) และในหน่วยแคลอรี่ (calories ตัวย่อ cal) โดยฟังก์ชันรับ ค่าน้ำหนักของลูกбол m กรัม ค่าความเร็วสูงสุดของลูกบอล v ในหน่วย กิโลเมตรต่อชั่วโมง และความยาวของสนามเทนนิส d เมตร. สมมติว่าไม่มีแรงต้านทางอากาศ ไม่มีผลจากแรงดึงดูดของโลก ไม่มีผลจากการกระเดิงที่ผิวสนาม และคิดประมาณระยะทางเฉพาะในแนวราบ ทิศทางความยาวสนาม.

ตัวอย่างการเรียกใช้ฟังก์ชัน

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

เมื่อ 200 คือความเร็วสูงสุดของลูกбол ในหน่วยกิโลเมตรต่อชั่วโมง 23.8 คือความยาวสนาม ในหน่วย เมตร 58 คือน้ำหนักลูกбол ในหน่วยกรัม และ **serve** คือฟังก์ชันที่ใช้คำนวณ. ผลลัพธ์คือ

0.8568 s

89.51 J

21.39 cal

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

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

(8) แนวทางปฏิบัติที่ดีในการเขียนโปรแกรมไฟรอน คือ ส่วนของโปรแกรมหลักจะเขียนอยู่ในรูปแบบ

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

แบบฝึกหัด 1.5

บริษัทขนส่งแห่งหนึ่ง คิดค่าบริการซึ่งประกอบด้วย ค่าบริการส่ง (คิดตามพื้นที่) และค่าส่งของ (คิด ตามน้ำหนัก) โดย ค่าบริการส่ง คิด 50 บาท ถ้าส่งในเขตจังหวัดขอนแก่น และคิด 100 บาท ถ้าส่ง นอกเขตจังหวัดขอนแก่น. 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.

จงเขียนฟังก์ชันรับที่อยู่ และน้ำหนักของ และคำนวณค่าส่งของบริษัทแห่งนี้.

ตัวอย่างการเรียกใช้ฟังก์ชัน

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

เมื่อ "Khon Kaen" คือพื้นที่ส่ง (อยู่ในเขตจังหวัดขอนแก่น) 14 คือน้ำหนักของที่ต้องการส่ง และ พงกชัน delivery_kk ทำหน้าที่คำนวนค่าจัดส่ง. 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
```

จะเขียนโปรแกรมเพื่อคำนวณค่ารากกำลังสองเฉลี่ย (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 $\sqrt{\frac{10+2+0.4+3.8}{4}} = 5.445181356024793$.

คำใบ้ (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) มодูล **math** มีฟังก์ชัน **math.sqrt** เพื่อใช้คำนวณค่าราก. (3) ตัวอย่างรูปแบบไวยากรณ์ไฟรอนสำหรับการวนซ้ำ คือ

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

เมื่อ **num** เป็นจำนวนครั้งที่ต้องการวนซ้ำ และตัวแปร **i** เป็นตัวชี้ของการวนซ้ำ. (4) ไฟรอนมีวิธีจัดรูปแบบข้อมูลสายอักขระ (string) ได้หลายแบบ (4.1) ใช้ตัวดำเนินการ % เช่น "**value %d**"%8 ซึ่งจะแสดงผลเป็น **value 8** หรือ (4.2) ใช้เมธอด **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**. ทำนองเดียวกัน **x = x + 1** คือ การกำหนดค่าให้ตัวแปร **x** เป็นค่าจากการประเมินนิพจน์ **x + 1** ดังนั้นหากรันคำสั่ง **x = x + 1** นี้แล้ว ตัวแปร **x** จะมีค่าเพิ่มจากเดิมขึ้นหนึ่ง.

แบบฝึกหัด 1.7

จะเขียนฟังก์ชัน 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 โดยตัวเลขในแต่ละตำแหน่ง แทนจำนวนครั้งที่พบเหตุการณ์นั้น เช่น เหตุการณ์ที่ 1 ไม่พบเลย เหตุการณ์ที่ 2 พบ 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.

คำให้ (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) แต่ละค่าของลิสต์สามารถนำอกมาได้ โดยการใช้ดัชนี เช่น **count[2]** จะได้ค่า 20 อกมา (ดัชนีแรก เริ่มที่ 0). (3) ฟังก์ชัน **len** สามารถช่วยนับจำนวนรายการทั้งหมดในลิสต์ได้. (4) คำสั่ง **for** สามารถทำงานกับลิสต์ได้โดยตรง เช่น

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

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

แบบฝึกหัด 1.8

จะเขียนฟังก์ชันที่รับข้อความ และนับความถี่ของคำต่าง ๆ ในข้อความ และส่งผลการนับความถี่ออกมานะ.

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 ในข้อความ ของ `txt`. ผลลัพธ์คือ

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

ซึ่งอยู่ในรูปของไฟรอนดิกชันนารี (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']` 岀กมา. (3) เมท็อด `strip` ช่วยตัดช่องว่างรอบคำออกได้สะดวก. (4) ดิกชันนารีว่า สามารถสร้างได้ เช่น `w = {}` จะสร้างดิกชันนารีว่า ให้กับตัวแปร `w`. (5) การอ้างอิงรายการของดิกชันนารี จะใช้กุญแจตัวนี้ ซึ่งเป็นชื่อเดียวกับตัวนี้ของลิสต์ เพียงแต่ กุญแจตัวนี้ของดิกชันนารีสามารถใช้เป็นสายอักขระได้ เช่น `w['Evil'] = 1` เป็นการกำหนดค่า 1 ให้กับรายการที่มีกุญแจตัวนี้เป็น `'Evil'` ซึ่งหากยังไม่มีรายการของกุญแจนี้อยู่ ไฟรอนจะสร้างขึ้นมาใหม่ แต่หากมีอยู่แล้วค่า 1 ก็จะไปแทนที่ค่าเดิมของรายการนี้. กลไกนี้ทำให้ดิกชันนารีสะดวกมากกับการใช้นับความถี่คำในลักษณะ เช่นนี้. (6) เช่นเดียวกับตัวแปรเดียว รายการของตัวนี้สามารถใช้ในลักษณะการเปลี่ยนแปลงค่าได้ เช่น `w['Evil'] += 1` จะเป็นการเพิ่มค่าของรายการของกุญแจตัวนี้ `'Evil'` จากเดิม ขึ้นไปหนึ่ง.

แบบฝึกหัด 1.9

ถอดรหัสดีเอ็นเอ. ดีเอ็นเอประกอบด้วย ฐานนิวคลีโอไทด์ (nucleotide bases) สี่ชนิด ได้แก่ อซตีนีน (adenine ตัวย่อ A), ไซโตซีน (cytosine ตัวย่อ C), กัวานีน (guanine ตัวย่อ G) และไรมีน

(thymine ตัวย่อ T). ลำดับของฐานนิวคลีโอไทด์ต่าง ๆ จะเป็นข้อมูลที่เซลล์นำไปใช้ในกระบวนการสร้างโปรตีน. นั่นคือ ลำดับของฐานนิวคลีโอไทด์สามตัว จะบอกชนิดของกรดอะมิโน (amino acid) ที่จะเซลล์จะสร้างเพื่อไปประกอบเป็นโปรตีน (หรืออาจจะเป็นรหัส เพื่อบอกการจับของลำดับสายกรดอะมิโน). ชุดของฐานนิวคลีโอไทด์สามตัว จะเรียกว่า โคดอน (codon). โคดอน จะถูกอ่านตามลำดับ และจะไม่มีอ่านดีเอ็นเอซ้อนกัน เช่น 'AAGGGC' จะอ่านเป็นโคดอนสองชุด คือ 'AAG' และ 'GGC'.

จงเขียนฟังก์ชัน เพื่อแปลงจากลำดับดีเอ็นเอ ไปเป็นโปรตีน ซึ่งคือ สายของกรดอะมิโน โดย ฟังก์ชันรับไฟล์ตารางโคดอน ที่เป็นตารางการแปลงโคดอนไปเป็นกรดอะมิโน และรับไฟล์ลำดับดีเอ็นเอ แล้ว ตลอดลำดับดีเอ็นเอ ทีละสามฐาน และส่งออกผลที่แปลงออกมายังไฟล์

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.

ส่วน 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 เพื่อเข้าใจรูปแบบของข้อมูลที่เก็บ.
(2) ไฟรอนใช้ไวยากรณ์ดังนี้ในการเปิดอ่านไฟล์

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

โดย 'filename' แทนชื่อไฟล์ที่ต้องการเปิดอ่าน (ระบุด้วย 'r') และใช้ตัวแปร f เป็นตัวจัดการไฟล์ (file handle). เมื่อต้อง read ใช้อ่านเนื้อหาทั้งหมดของไฟล์ออกมายัง variable f.

- (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)]` เพื่อดึงโคดอนออกมามี `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 ครีบีชั้น จากตัวแรก	D	A
ตัวที่สาม เสียงสูงขึ้น 2 ครีบีชั้น จากตัวที่สอง	E	B
ตัวที่สี่ เสียงสูงขึ้น 1 ครีบีชั้น จากตัวที่สาม	F	C
ตัวที่ห้า เสียงสูงขึ้น 2 ครีบีชั้น จากตัวที่สี่	G	D
ตัวที่หก เสียงสูงขึ้น 2 ครีบีชั้น จากตัวที่ห้า	A	E
ตัวที่เจ็ด เสียงสูงขึ้น 2 ครีบีชั้น จากตัวที่หก	B	F# (2 ครีบีชั้น นั่นคือ E → F → F#)

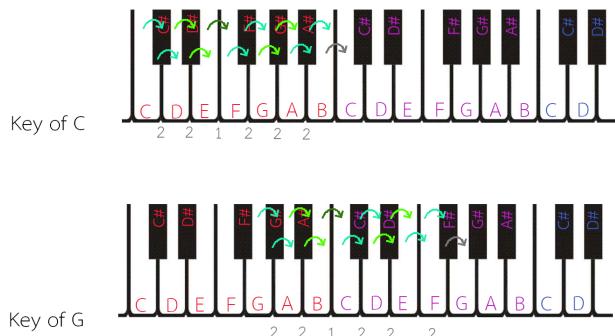
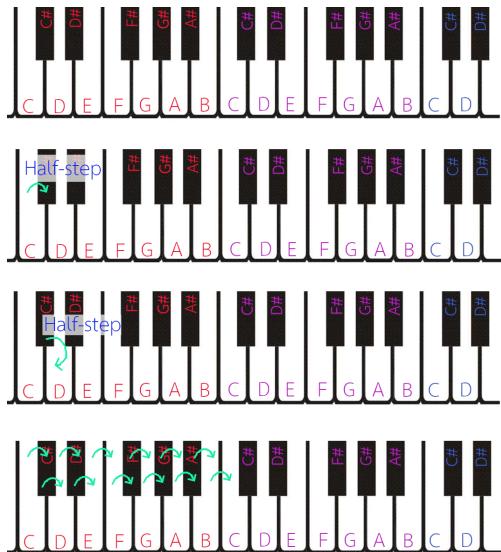
หมายเหตุ ครีบีชั้น (half-step) กล่าวโดยง่าย คือ ห่างกัน 1 ก้านดีดเปียโน (รวมก้านดีดทั้งสี่ขาว และดำ ดูรูปที่ 1.6 ประกอบ).

จะเขียนฟังก์ชัน `diatonic` ที่รับอาร์กิวเม้นต์ `scale_key` สำหรับกุญแจเสียง และให้ค่าโน้ตดนตรีในระดับเสียงเต็มรูปออกมายโดยใช้เลขจำนวนเต็มแทนโน้ตดนตรีต่าง ๆ ดังนี้ เลข 1 แทนโน้ต C, เลข 2 แทนโน้ต C#, เลข 3 แทนโน้ต D, เลข 4 แทนโน้ต D# เป็นต้น.

คำใบ้ มодูลो (modulo) หรือการหารเอาเศษ ซึ่งใช้ตัวดำเนินการ % อาจช่วยให้ทุกอย่างง่ายขึ้น ตัวอย่างผลการทำงาน

```
>>> 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)



รูปที่ 1.6 ระดับเสียงเต็มรูป. ภาพແລະບັນສຸດຂ້າຍ ແສດງກໍານົດເປີຍໂນ ພຣ້ອມໂນີຕົດນຕຣີ. ຝາພຂ້າຍແດວສອງ ແລະສາມ ແລະແດວ ຂໍ ແສດງລູກສຽບຮູບຮັບຮັບຕັບເສີຍຄຣິ່ງຂຶ້ນ. ສັງເກົດ E → F ແລະ B → C ເພີ່ມຮັບຮັບຕັບເສີຍແຄ່ຄຣິ່ງຂຶ້ນ (ໄມ້ມີກໍານົດດໍາຍໍ່ຕຽງກ່າງ). ຝາພບັນທາງຂວາ ແສດງກໍານົດເປີຍໂນ ພຣ້ອມໂນີຕົດນຕຣີ ແລະລູກສຽບແສດງການເພີ່ມຮັບຮັບຕັບເສີຍທີລະຄຣິ່ງຂຶ້ນ ຈາກກຸ່ມແຈເສີຍ C ເປົ້າຢືນເຫັນກັບພາພລ່າງທາງຂວາ ທີ່ແສດງກາຮາໂນີຕົດນຕຣີໃນຮັບຮັບຕັບເສີຍເຕີມຮູບ ເນື້ອໃຈ້ກຸ່ມແຈເສີຍ G.

บทที่ 2

พื้นฐาน

“Divide each difficulty into as many parts as is feasible and necessary to resolve it.”

---René Descartes

“แบกปัญหาออก เป็นส่วนย่อย ๆ เท่าที่จะทำได้และจำเป็นที่จะแก้มันได้”

—เรอเน เดการ์ต

ศาสตร์การรู้จำรูปแบบและการเรียนรู้ของเครื่อง อาศัยพื้นฐานจากหลาย ๆ ศาสตร์ การทำความเข้าใจศาสตร์นี้ และพัฒนาการ จำเป็นต้องอาศัยศาสตร์พื้นฐาน. บทนี้จะทบทวนศาสตร์พื้นฐานที่สำคัญคือ พีชคณิตเชิงเส้น ความน่าจะเป็น และการหาค่าดีที่สุด.

2.1 พีชคณิตเชิงเส้น

การรู้จำรูปแบบและการเรียนรู้ของเครื่อง เกี่ยวข้องกับข้อมูลและตัวแปรจำนวนมาก. พีชคณิตเชิงเส้น¹ มีเครื่องมือและทฤษฎีต่าง ๆ ที่ช่วยอำนวยความสะดวก ในการทำงานกับตัวแปรจำนวนมาก ดังนั้น จึงเป็น เป็นพื้นฐานที่สำคัญ

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

เวกเตอร์ (vector) หมายถึง ลำดับของตัวเลข. พีชคณิตเชิงเส้นมีเวกเตอร์สองชนิด คือ เวกเตอร์ วนวนอน และเวกเตอร์แนวตั้ง. เวกเตอร์วนวนอน แสดงด้วยลำดับในวนวนอน เช่น $[103.4, -28.6, 0, 9.99]$ เป็นเวกเตอร์วนวนอนที่เป็นลำดับของตัวเลขสี่ตัว. ตำราเล่มนี้ใช้เวกเตอร์ แนวตั้งเป็นหลัก. นั่นคือ หากกล่าวถึงเวกเตอร์โดยไม่ได้ระบุเป็นอื่นแล้ว จะหมายถึงเวกเตอร์แนวตั้ง ที่

¹เนื้อหาในหัวข้อนี้ได้รับอิทธิพลหลักจาก [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 ดังแสดงใน สมการ 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 ที่ด้านนีตำแหน่ง แถว i และส عم�� j . ด้านนีอาจเขียนโดยใช้ตัวห้อยโดยมีจุดภาคคั่น เช่น $A_{i,j}$ หรือจุดภาคอาจถูกละไว้ได้ เช่น A_{ij} ในกรณีที่ความหมายชัดเจน. เมตริกซ์สามารถถูกระบุขนาดมิติ ได้จาก สัญกรณ์ $\mathbf{A} \in \mathbb{R}^{m \times n}$.

นอกจากนี้ ศาสตร์การเรียนรู้ของเครื่อง นิยมใช้สัญกรณ์จุดคู่ ดังปฏิบัติใน [76]. นั่นคือ กำหนดให้ สัญกรณ์ $\mathbf{A}_{i,:}$ หมายถึง เมตริกซ์ย่ออย ที่ได้จากการนำส่วนประกอบที่แถว 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 เมตริกซ์ย่ออย ที่ได้จากการนำส่วนประกอบที่ส عم馬 j ของเมตริกซ์ \mathbf{A} เรียกว่า ส عم馬 j ของ \mathbf{A} .

การสลับเปลี่ยน (transpose) เป็นการดำเนินการจัดเรียงลำดับใหม่. สำหรับเวกเตอร์ การสลับเปลี่ยนของเวกเตอร์แนวอน จะได้เวกเตอร์แนวตั้ง และการสลับเปลี่ยนของเวกเตอร์แนวตั้ง จะได้ เวกเตอร์แนวอน และใช้สัญกรณ์ เช่น \mathbf{x}^T คือ การสลับเปลี่ยนของเวกเตอร์ \mathbf{x} เช่น จาก สมการ 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 การเปลี่ยนตำแหน่งของ ส่วนประกอบ โดย ส่วนประกอบที่ตำแหน่ง (i, j) จะถูกเปลี่ยนไปอยู่ตำแหน่ง (j, i) ซึ่งผลที่ได้เหมือน กับการสลับตำแหน่งรอบแนวทางและมุมของเมตริกซ์. สมการ 2.3 แสดงการสลับเปลี่ยนของเมตริกซ์

A เปรียบเทียบกับสมการ 2.2.

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

การสลับเปลี่ยนของเมทริกซ์ จะทำให้ขนาดมิติของเมทริกซ์เปลี่ยนไป เช่น จากตัวอย่าง ขนาดมิติของเมทริกซ์ \mathbf{A} คือ $m \times n$ แต่ ขนาดมิติของเมทริกซ์ \mathbf{A}^T คือ $n \times m$ (สังเกตจาก \mathbf{A}^T มี n แถว และ m สดมร्ग)

มิติ และลำดับชั้น. โครงสร้างของเวกเตอร์ มีหนึ่งมิติของลำดับ ในลักษณะที่ลำดับของข้อมูลดำเนินไปได้แนวเดียว. โครงสร้างของเมทริกซ์ มีสองมิติ ในลักษณะที่ลำดับของข้อมูลดำเนินไปได้สองแนว. ภาพขาวดำ² หรือภาพสเกลเทา (gray-scale image) ก็มีลักษณะโครงสร้างสองมิติของลำดับ นั่นคือ มีโครงสร้างสองมิติของลำดับค่าพิกเซล ตามลำดับแนวตั้ง และตามลำดับแนวนอน. การเขียนโปรแกรมจะใช้โครงสร้างข้อมูล เช่น อาร์เรย์ (array) ขนาดสองมิติ ในการแทนข้อมูลภาพสเกลเทานี้.

ข้อมูลอาจมีโครงสร้างมิติของลำดับที่ซับซ้อนมากได้. ภาพสี (color image) มีโครงสร้างสามมิติของลำดับค่าพิกเซล ตามลำดับแนวตั้ง ตามลำดับแนวนอน และตามชุดของช่องสี แดง เขียว น้ำเงิน. (แม้ช่องสีไม่ได้มีความสัมพันธ์ในเชิงลำดับในลักษณะที่ช่องสีแดง ไม่จำเป็นต้องมาก่อนช่องสีเขียว เป็นต้น แต่ข้อมูลของช่องสีต่าง ๆ แยกออกเป็นลักษณะอิสระของตัวเอง). การเขียนโปรแกรมจะใช้โครงสร้างข้อมูล เช่น อาร์เรย์ ขนาดสามมิติ ในการแทนข้อมูลภาพสี. ข้อมูลวิดีโอ (video) เป็นข้อมูล เป็นลักษณะโครงสร้างสี่มิติของลำดับค่าพิกเซล ตามลำดับแนวตั้ง ตามลำดับแนวนอน ตามชุดของช่องสี และตามลำดับเวลา. การเขียนโปรแกรมจะใช้โครงสร้างข้อมูล เช่น อาร์เรย์ ขนาดสี่มิติ ในการแทน ข้อมูลวิดีโอ และข้อมูลวิดีโอ ที่กล่าวถึงนี้ ยังไม่ได้รวมข้อมูลช่องเสียงด้วย ซึ่งหากเป็นช่องเสียงเดียว (monophonic sound channel) ก็ต้องการมิติของอาร์เรย์อีกหนึ่งมิติ หรือหากเป็นช่องเสียงสเตอโรไโอด (stereophonic sound channel) ก็ต้องการมิติของอาร์เรย์อีกสองมิติ หรือหากแยกช่องเสียงพูด ออกจากเสียงประกอบอื่น ๆ หรือมีข้อมูลคำบรรยายภาษาต่าง ๆ ประกอบ ก็ต้องการมิติของอาร์เรย์เพิ่มขึ้นอีก.

อย่างไรก็ตาม หากกล่าวถึง “มิติ” นั้นจะสังเกตว่า ภาพสเกลเทาสองมิติ เป็นข้อมูลโครงสร้างลำดับสองมิติ. ภาพสีสองมิติ เป็นข้อมูลโครงสร้างลำดับสามมิติ. วิดีโอ (ของภาพสองมิติไม่รวมข้อมูลเสียง) เป็นข้อมูลโครงสร้างลำดับสี่มิติ. นอกจากนั้น เมื่อกล่าวถึง ปริภูมิค่า (vector space) ซึ่ง ใช้บอก

²ภาพขาวดำ ในรูปแบบที่วีป หมายถึง ภาพสเกลเทา ที่ไม่สามารถแสดงค่าทึบสนิทได้แต่ขาว ไปจนถึงดำ. อย่างไรก็ตาม ในระบบข้อมูล มีข้อมูลภาพที่เป็นค่าทึบสนิท นั่นคือ ภาพสามารถแสดงสีได้แค่สองสี นั่นคือ แต่ละพิกเซล จะสามารถแสดงได้แค่สีขาว หรือสีดำ เท่านั้น ไม่สามารถแสดงระดับสีอื่น ๆ ระหว่างกลางได้. ดังนั้น เพื่อลดความสับสน ในที่นี้จะใช้คำว่า ภาพสเกลเทา แทนคำว่า ภาพขาวดำ.

ถึง ขนาดความเป็นไปได้ของลักษณะข้อมูล และ มิติของปริภูมิจะใช้บวกขนาด และความซับซ้อนของ ปริภูมินั้น ๆ เช่น ข้อมูลสเกลาร์ที่เป็นค่าจริง จะมีปริภูมิค่า เป็น \mathbb{R} เป็นปริภูมิค่าหนึ่งมิติ. แต่จะดู ข้อมูล อ้างถึงได้ด้วยตัวเลขตัวเดียว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาบนเส้น จำนวนจริง. ข้อมูลเวกเตอร์ที่มีส่วนประกอบเป็นค่าจริงสองค่า จะมีปริภูมิค่า เป็น \mathbb{R}^2 . แต่จะดู ข้อมูล อ้างถึงได้ด้วยตัวเลขสองตัว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาบนระนาบ จำนวนจริง. ข้อมูลเวกเตอร์ที่มีส่วนประกอบเป็นค่าจริงสี่ค่า จะมีปริภูมิค่า เป็น \mathbb{R}^4 . แต่จะดูข้อมูล อ้างถึงได้ด้วยตัวเลขสี่ตัว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาในปริภูมิขนาดสี่ มิติ. ข้อมูลเมทริกซ์ขนาด 2×2 จะมีปริภูมิค่า เป็น $\mathbb{R}^{2 \times 2}$. แต่จะดูข้อมูล อ้างถึงได้ด้วยตัวเลขสี่ตัว. การค้นหาจุดข้อมูลที่สนใจ ในปริภูมิ ทำได้โดยการค้นหาบนปริภูมิขนาดสี่มิติ เช่นเดียวกับ ข้อมูลเวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง. นั่นคือ เวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง มีความสามารถในการแทน ข้อมูลเทียบเท่ากับ เมทริกซ์ขนาด 2×2 เพียงแต่ ข้อมูลที่เก็บในเวกเตอร์ที่มีส่วนประกอบสี่ค่าจริง ไม่ ได้มีโครงสร้างลำดับ เมื่อนอกกับ ข้อมูลที่เก็บในเมทริกซ์ขนาด 2×2 และโครงสร้างลำดับเช่นนี้ โดย เนพาะหากเป็นโครงสร้างตามธรรมชาติของข้อมูล สามารถนำมาใช้ในประโยชน์ และช่วยในการรู้จำรูป แบบอย่างมีประสิทธิภาพได้ (ดังเช่น ที่จะได้อธิบายในบท 5 ต่อไป)

อย่างไรก็ตาม เพื่อลดความสับสน จากนี้ไป เมื่อกล่าวถึง “มิติ” จะมีการระบุอย่างชัดเจนว่า หมาย ถึง มิติในความหมายใด เช่น คำว่า มิติ ใช้ในความหมาย มุมมอง ซึ่งเป็นความหมายกว้าง ๆ ของมิติ และใช้ในความหมายของมิติโดยทั่วไป. คำว่า มิติปริภูมิค่า ใช้ในความหมายของ มิติของปริภูมิค่า และ คำว่า ลำดับชั้น (rank) ใช้ในความหมายของ มิติของโครงสร้างลำดับ. ตัวอย่าง ภาพสเกลเทาขนาด 600×800 พิกเซล เป็น ภาพสองมิติ ที่เป็นข้อมูลลำดับชั้นสอง (มีลำดับตามแนวตั้ง และตามแนว นอน) หรือสามารถแทนด้วย เมทริกซ์ ขนาดมิติ 600×800 และมีมิติปริภูมิค่า เป็น 480000. สัญ กรณ์ $\{0, \dots, 255\}^{600 \times 800}$ จะระบุชุดเจนทั้งจากมุมมองของลำดับชั้น และมิติปริภูมิค่า. นอกจาก นั้น สัญกรณ์นี้ยังระบุช่วงค่าที่เป็นไปได้ของข้อมูลด้วย ว่า แต่ละค่าเป็นจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 255.

เทนเซอร์ (tensor) หมายถึง โครงสร้างลำดับชั้นของตัวเลข. สเกลาร์ คือ เทนเซอร์ ลำดับชั้น ศูนย์ (rank-0 tensor). เวกเตอร์ คือ เทนเซอร์ ลำดับชั้นหนึ่ง (rank-1 tensor). เมทริกซ์ คือ เทนเซอร์ ลำดับชั้นสอง (rank-2 tensor). ข้อมูลที่แทนด้วยอาร์เรย์ขนาด n มิติ คือ เทนเซอร์ ลำดับชั้น n (rank- n tensor). ตัวเลขแต่ละตัวในเทนเซอร์ เป็น ส่วนประกอบของเทนเซอร์ และอ้างอิงได้โดยใช้ ดัชนี ตามลำดับชั้น.

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

$$\mathbf{A} = \left[\begin{bmatrix} [[1.2], [3]] & [[8], [6]] \end{bmatrix} \quad \begin{bmatrix} [[0.9], [-1]] & [[4], [1]] \end{bmatrix} \quad \begin{bmatrix} [[7], [5]] & [[0.1], [0]] \end{bmatrix} \right]$$

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 σ หรือ calc โดยไม่มีการ ทำแบบอักษรตัวหนา ไม่ว่าฟังก์ชันจะให้ค่าอ กมาเป็นสเกลาร์ หรือเมทริกซ์ หรือเทนเซอร์ 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 $f : \rho_1 \rightarrow \rho_2$ ระบุว่าฟังก์ชัน f รับค่าตัวแปรที่อยู่ในเซต ρ_1 เพื่อไปทำการคำนวณ และให้ผลลัพธ์การคำนวณออกมาเป็นค่าที่อยู่ในเซต ρ_2 . ตัวอย่างเช่น $g : \mathbb{R}^n \rightarrow \mathbb{R}$ จะบอกว่า ฟังก์ชัน g รับตัวแปรเข้าเป็นเวกเตอร์ที่มี n ส่วนประกอบ และจะให้ผลลัพธ์ออกมาเป็นสเกลาร์.

การคำนวณเมทริกซ์. การบวกลบเมทริกซ์กับเมทริกซ์ จะทำได้ ก็ต่อเมื่อ เมทริกซ์มีขนาดมิติเท่ากัน และผลลัพธ์คำนวณได้จากการค่าของส่วนประกอบทั้งสองที่ทำແเน่งเดียวกัน 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 จำนวนส่วนประกอบของเมทริกซ์ตัวหน้า เท่ากับ จำนวนแผลของเมทริกซ์ตัวหลัง และใช้สัญกรณ์ เช่น \mathbf{AB} หรือ $\mathbf{A} \cdot \mathbf{B}$ โดย เมทริกซ์ \mathbf{A} มีขนาดมิติ $m \times p$. เมทริกซ์ \mathbf{B} มีขนาดมิติ $p \times n$. ผลลัพธ์ $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ จะมีขนาดมิติ $m \times n$ และ $C_{ij} = \sum_k A_{ik} \cdot B_{kj}$.

นอกจากการคูณเมตริกซ์แล้ว การดำเนินการ **การคูณแบบตัวต่อตัว (element-wise product หรือ Hadamard product)** ก็มีการใช้อย่างกว้างขวาง. การคูณแบบตัวต่อตัว จะดำเนินการได้ ก็ต่อเมื่อ เมตริกซ์สองเมตริกซ์ที่จะคูณกัน ต้องมีขนาดมิติที่เท่ากัน และใช้สัญกรณ์ เช่น $\mathbf{A} \odot \mathbf{B}$ โดย เมตริกซ์ \mathbf{A} มีขนาดมิติ $m \times n$ เมตริกซ์ \mathbf{B} มีขนาดมิติ $m \times n$ และ ผลลัพธ์ $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$ จะมีขนาด มิติ $m \times n$ และ $C_{ij} = A_{ij} \cdot B_{ij}$. การคูณกันของเวกเตอร์ จะแสดงเป็นรูปแบบการดำเนินการเมตริกซ์ 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)

คุณสมบัติของการคูณเมทริกซ์. การคูณเมทริกซ์ 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 การคูณเมทริกซ์ไม่มีคุณสมบัติการสลับที่.

ระบบสมการ

เมทริกซ์ เวกเตอร์ และการดำเนินการต่าง ๆ ที่กล่าวมา นอกจากจะช่วยการอ้างถึง และการจัดการกับข้อมูลทำได้อย่างสะดวกแล้ว 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 และสามารถเขียนได้กระหัดรัด ด้วยสัญกรณ์ ของเมตริกซ์ และเวกเตอร์ ดังนี้

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

และกระหัดรัดขึ้นอีก เป็น

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2.7)$$

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) ที่นิยમใช้สัญญาณักษา \mathbf{I} (หรือ \mathbf{I}_n เมื่อต้องการระบุขนาด มิติ $n \times n$) เป็นเมตริกซ์ที่มีค่าตามแนวทะแยงมุมเป็นหนึ่ง และค่าอื่น ๆ เป็นศูนย์ เช่น เมตริกซ์เอกลักษณ์ ขนาดมิติ 3×3 คือ

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

เมตริกซ์เอกลักษณ์ มีคุณสมบัติสำคัญ คือ เมื่อคูณกับเมตริกซ์ใด หรือคูณกับเวกเตอร์ใด แล้วจะได้ เมตริกซ์นั้น หรือเวกเตอร์นั้น ตัวเดิม. นั่นคือ $\mathbf{A} \cdot \mathbf{I} = \mathbf{A}$.

เมตริกซ์ผกผัน (matrix inverse) คือ เมตริกซ์คู่คูณ ที่เมื่อคูณกับคู่ของมันแล้ว ผลลัพธ์จะได้เป็น เมตริกซ์เอกลักษณ์ ใช้สัญญาณักษาเป็นเมตริกซ์ที่มีตัวยกลบหนึ่ง เช่น \mathbf{A}^{-1} หมายถึง เมตริกซ์ผกผัน ที่ เป็นคู่ผกผันกับ \mathbf{A} นั่นคือ $\mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}$. จากตัวอย่าง เมตริกซ์ผกผัน 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) มีจำนวนแฉวเท่ากับจำนวนสตดมภ์ ซึ่งเรียกว่า เมทริกซ์จตุรัส (*square matrix*).

โดยที่ไปแล้ว ถ้าหากมีจำนวนสมการน้อยกว่า (จำนวนแฉวของเมทริกซ์ น้อยกว่า จำนวนสตดมภ์ หรือ เมทริกซ์ที่มีสัดส่วนเตี้ยกว่าง) คำตอบของระบบสมการจะมีได้หลายค่า เช่น ตัวอย่างระบบสมการ

$$\begin{aligned} x + y + z &= 6 \\ 2x + 2y - z &= 3 \end{aligned}$$

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 (จำนวนแฉวของเมทริกซ์ มากกว่า จำนวนสตดมภ์ หรือ เมทริกซ์ที่มีสัดส่วนสูงแอบ) จะจะไม่สามารถหาคำตอบของระบบสมการได้ 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 $x = 1, y = 2, z = 3$ ที่เป็นคำตอบของสามสมการ แรก 2.10 ถึง 2.12 ขัดแย้งกับสมการที่สี่ 2.13)

เมทริกซ์เอกฐาน. ถึงแม้เมทริกซ์จะเป็นเมทริกซ์จตุรัส 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

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

เมทริกซ์สัมประสิทธิ์ที่ได้จะเป็นเมทริกซ์เอกฐาน. สังเกต สมการแรก 2.14 กับสมการที่สอง 2.15 จะเห็นว่า สมการที่สอง มีค่าเท่ากับสมการแรกคูณสอง ซึ่งหมายความว่า ถึงแม้จะมีสองสมการ แต่ก็ให้ข้อมูลเทียบเท่ากับการมีแค่สมการเดียว 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 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$. การที่แผลดิของเมทริกซ์ เป็นผลรวมเชิงเส้นของแผลอื่น ๆ 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

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

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

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

2. ถ้ามีสอดคล้องที่มีค่าเท่ากัน ค่าดีเทอร์มิแนนต์จะเป็นศูนย์. นั่นคือ ถ้า $\mathbf{a}_i = \mathbf{a}_j$ 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

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

จากคุณสมบัติพื้นฐานทั่วไป ทฤษฎีการขยายปัจจัยร่วม (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 เป็นตัวเลขที่เลือกตั้งให้ค่าที่ ($k \in \{1, \dots, n\}$) และ ปัจจัยร่วม (cofactor) $C_{ij} = (-1)^{i+j} \cdot M_{ij}$. โดย M_{ij} เป็นค่าดีเทอร์มิแนนต์ของเมทริกซ์ย่อยจาก \mathbf{A} โดยการตัดแต่งแถวที่ i และตัดสอดคล้องที่ j ออก. ตัวอย่าง

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 3 \\ 8 & 2 & 7 \\ 0 & 5 & 9 \end{bmatrix}$$

เมื่อเลือกตั้ง $k = 1$ 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

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

$$\begin{vmatrix} 8 & 2 \\ 0 & 5 \end{vmatrix} = 8 \cdot |5| - 2 \cdot |0| = 40. \quad \text{ดังนั้น } \det \mathbf{A} = -17 - 4 \cdot 72 + 3 \cdot 40 = -185.$$

คุณสมบัติที่ตามมาของดีเทอร์มิแนนต์ มีหลายอย่าง แต่ที่สำคัญ คือ $\det \mathbf{A} = \det \mathbf{A}^T$ ดังนั้น การที่ระบบสมการมีสมการที่ไม่เป็นอิสระเชิงเส้นต่อกัน (例外ของเมทริกซ์ ไม่เป็นอิสระเชิงเส้นต่อกัน) จะทำให้ เมทริกซ์สัมประสิทธิ์ มีค่าดีเทอร์มิแนนต์เป็นศูนย์ ซึ่งหมายถึง เมทริกซ์เป็นเอกฐาน.

นอร์ม. ดีเทอร์มิแนนต์ บอกคุณสมบัติที่สำคัญของเมทริกซ์จัตุรัส. เวกเตอร์ก็ต้องการค่าบวกคุณ-สมบัติ. นอร์ม (norm) เป็นการวัดขนาดของเวกเตอร์. จำนวนส่วนประกอบของเวกเตอร์ อาจจะบวกขนาดมิติปริภูมิของเวกเตอร์ แต่นอร์มจะบวกขนาดของเวกเตอร์ 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 $p \in \mathbb{R}, p \geq 1$.

งานส่วนใหญ่เลือกใช้ L^2 นอร์ม ($p = 2$) หรือเรียกว่า ยูคลีเดียนนอร์ม (Euclidean norm) หรือ ระยะทางยูคลีเดียน (Euclidean distance) ที่ใช้สัญกรณ์ $\|\mathbf{x}\|_2$. ความนิยมของ L^2 นอร์ม ทำให้บ่อยครั้ง สัญกรณ์อาจลับตัวห้อยออก เป็น $\|\mathbf{x}\|$. นอกจากนั้น บางครั้ง เพื่อความสะดวกและลดการคำนวณ ค่า L^2 นอร์มกำลังสอง ได้แก่ $\|\mathbf{x}\|_2^2 = \sum_i x_i^2 = \mathbf{x}^T \cdot \mathbf{x}$ ก็นิยมใช้วัดขนาดเวกเตอร์

ตัวอย่างเช่น $\mathbf{v} = [0.5, -8, 11]^T$ มี L^2 นอร์มคือ $\|\mathbf{v}\| = \sqrt{\sum_i v_i^2} = \sqrt{0.25 + 64 + 121} = 13.61$. สังเกตว่า ส่วนประกอบที่มีค่าใกล้ ๆ ศูนย์ จะมีขนาดผลลัพธ์ลง เมื่อคำนวณด้วย L^2 นอร์ม (สังเกต ส่วนประกอบ 0.5 มีขนาดผลลัพธ์เป็น 0.25) ดังนั้น สำหรับบางภาระกิจที่ไม่ต้องการพฤติกรรมเช่นนี้ แต่ต้องการ พฤติกรรมที่ผลของส่วนประกอบมีอัตราส่วนคงที่ ซึ่งจะช่วยให้เห็นความแตกต่างระหว่างศูนย์ กับค่าที่ไม่เป็นศูนย์ได้ดีกว่า ภาระกิจเหล่านี้อาจเลือกใช้ 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 $\mathbf{u} \in \mathbb{R}^n$ ที่ซึ่งเป็นทางเดียวกับมัน และ

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

จากตัวอย่างข้างต้น เวกเตอร์ $\mathbf{v}_4 = [1, 1, 0]^T$ มีเวกเตอร์หนึ่งหน่วย $[0.7071, 0.7071, 0]^T$ ที่ซึ้งไปทางเดียวกับมัน และเวกเตอร์ $\mathbf{v}_5 = [0.7, 0.2, 0.1]^T$ มีเวกเตอร์หนึ่งหน่วย $[0.953, 0.272, 0.136]^T$ ที่ซึ้งไปทางเดียวกับมันนั้น.

ภาพฉายเชิงตั้งฉาก

การเปลี่ยนมุมมอง รวมถึงการแยกปัจจัยประกอบออก เป็นหนึ่งในวิธีที่ช่วยในการทำความเข้าใจเรื่อง รากต่าง ๆ รวมถึงการทำความเข้าใจกลุ่มของตัวแปรต่าง ๆ และทำความเข้าใจข้อมูล. การทำภาพฉายเชิงตั้งฉาก เป็นเสมือนการเปลี่ยนมุมมองวิธีหนึ่ง

รูปที่ 2.1 แสดงตัวอย่างการทำภาพฉายเชิงตั้งฉาก. เวกเตอร์ $\mathbf{x} = [3, 4]^T$ เมื่อฉายลงบนทิศทางของของเวกเตอร์ $\mathbf{u} = [1, 0]^T$ แล้ว จะมีขนาดเป็น 3 บันทิศทางของ \mathbf{u} และเมื่อฉายลงบน $\mathbf{v} = [0, 1]^T$ จะมีขนาดเป็น 4 บันทิศทางของ \mathbf{v} . ขนาดของการฉายเวกเตอร์ใด ๆ \mathbf{z} ลงบนทิศทางของเวกเตอร์หนึ่งหน่วย \mathbf{u} จะคำนวณได้จาก

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

เมื่อ z_u เป็นขนาดของเวกเตอร์ \mathbf{z} ที่ฉายลงบนทิศทางของเวกเตอร์หนึ่งหน่วย \mathbf{u} .

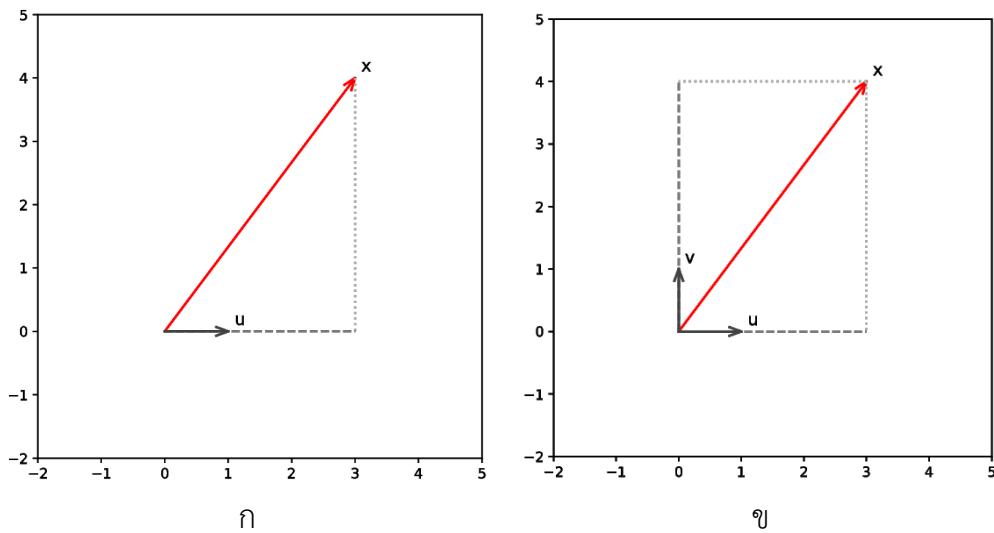
รูปที่ 2.2 แสดงตัวอย่างการทำภาพฉายเชิงตั้งฉากของเวกเตอร์เดิม $\mathbf{x} = [3, 4]^T$ แต่ฉายลงบนเวกเตอร์ $\mathbf{u} = [0.9578, 0.2873]^T$ และเวกเตอร์ $\mathbf{v} = [-0.2873, 0.9578]^T$. ดังนั้น ขนาดของ \mathbf{x} ที่ฉายลงบน \mathbf{u} จะเป็น $\mathbf{x}^T \mathbf{u} = 4.023$ และขนาดที่ฉายลงบน \mathbf{u} 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 ด้วยเมตริกซ์ แปลง (*transformation matrix*). นั่นคือ การแปลงเชิงเส้น $\mathbf{T}(\mathbf{x}) = \mathbf{Ax}$. การหาเวกเตอร์ตั้งฉาก อาจมองเป็นการแปลงเวกเตอร์ไปเป็นเวกเตอร์ที่ตั้งฉากกับตันฉบับ ซึ่งตัวอย่างข้างต้นนี้ใช้เมตริกซ์ แปลง

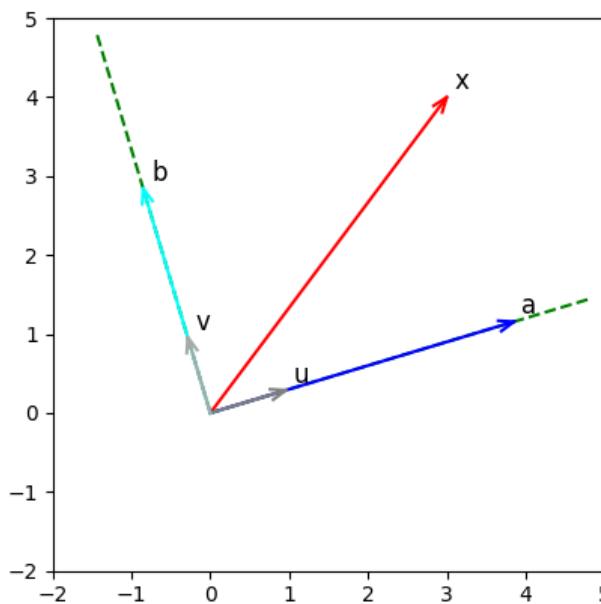
$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

และจากตัวอย่าง $\mathbf{u} = [0.9578, 0.2873]^T$ ซึ่งเมื่อทำการแปลงเชิงเส้นด้วยเมตริกซ์ \mathbf{A} ผลลัพธ์คือ เวกเตอร์ที่ตั้งฉากกับเวกเตอร์ตันฉบับ $\mathbf{Au} = [-0.2873, 0.9578]^T$.

เวกเตอร์ $\mathbf{x} \in \mathbb{R}^n$ เรียกว่า \mathbf{x} อยู่ในปริภูมิ \mathbb{R}^n นั่นคือ ถ้า \mathbb{R}^n เป็นเสมือนแผนที่ เวกเตอร์ \mathbf{x} ได้ ๆ ที่อยู่ในปริภูมิ ก็เปรียบเสมือนเป็นจุด ๆ หนึ่งบนแผนที่นั้น. จุดใด ๆ ในปริภูมิ \mathbb{R}^n จะสามารถอ้างอิงถึงได้โดยใช้ตัวเลข n ตัว (นั่นคือ n มิติปริภูมิค่า)



รูปที่ 2.1 การฉายภาพ. (ก) การฉายภาพของเวกเตอร์ $\mathbf{x} = [a, b]^T$ ลงบนเวกเตอร์ $\mathbf{u} = [1, 0]^T$ คือ การฉายภาพจากเวกเตอร์ที่สนใจ ในแนวตั้งจากไปลงบน 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 เมื่อฉาย บนเวกเตอร์ \mathbf{v} . ทั้ง \mathbf{u} และ \mathbf{v} เป็นเวกเตอร์หนึ่ง направย.



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

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

ตัวอย่าง เช่น จากรูปที่ 2.2 เวกเตอร์ \mathbf{x} อยู่ใน \mathbb{R}^2 และเวกเตอร์ \mathbf{u} และเวกเตอร์ \mathbf{v} และเวกเตอร์ \mathbf{a} และเวกเตอร์ \mathbf{b} ก็อยู่ในปริภูมิ \mathbb{R}^2 ด้วย. สมมติให้ ρ_1 เป็นปริภูมิย่อยของ \mathbb{R}^2 โดยมี เวกเตอร์ \mathbf{u} อยู่ใน ρ_1 แต่ไม่มีเวกเตอร์ \mathbf{v} ในปริภูมิย่อย ρ_1 . นั่นคือ ปริภูมิย่อย ρ_1 เป็นเซตปิด สำหรับ เวกเตอร์ในแนว \mathbf{u} เช่น \mathbf{u} และ \mathbf{a} อยู่ใน ρ_1 .

มองจากอีกมุมหนึ่ง ถ้ากำหนดให้ $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$ เป็นเซตของเวกเตอร์ใด ๆ ในปริภูมิ \mathbb{R}^n แล้ว ผลรวมเชิงเส้นทุกแบบของเวกเตอร์เหล่านี้ จะเป็นเซตของเวกเตอร์ ที่เรียกว่า การແພ່ທຳ (span) ของ เวกเตอร์ $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ และใช้สัญกรณ์³

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

และ ปริภูมิย่อยคือ การແພ່ທຳของเซตของเวกเตอร์. ตัวอย่างเช่น ปริภูมิย่อย ρ_1 ในตัวอย่างข้างต้น ก็ คือการແພ່ທຳของ $\{\mathbf{u}\}$ และปริภูมิ \mathbb{R}^2 ก็คือการແພ່ທຳของ $\{\mathbf{u}, \mathbf{v}\}$.

สังเกตว่าในขณะที่ การอ้างถึงตำแหน่งบนปริภูมิ \mathbb{R}^2 ต้องการตัวเลข 2 ตัว (สองมิติปริภูมิค่า) แต่การอ้างถึงตำแหน่งบนปริภูมิย่อย ρ_1 ต้องการตัวเลขแค่ตัวเดียว (ปริภูมิย่อย มีหนึ่งมิติปริภูมิค่า). จำนวนมิติปริภูมิค่า ของปริภูมิใด ๆ (รวมถึงปริภูมิย่อยใด ๆ) จะเท่ากับจำนวนของเวกเตอร์ที่เป็นอิสระ เชิงเส้นกัน ที่อยู่ในปริภูมิ. ตัวอย่างเช่น ปริภูมิ \mathbb{R}^2 มีสองเวกเตอร์ที่เป็นอิสระเชิงเส้น ไม่ว่าจะเลือก \mathbf{u} กับ \mathbf{v} หรือเลือกชุด $[1, 0]^T$ กับ $[1, 1]^T$ หรือเลือกชุด $[1, -1]^T$ กับ $[0, 1]^T$ หรือชุดอื่น ๆ ของเวกเตอร์สองตัวที่ตั้งฉากกัน. ดังนั้น \mathbb{R}^2 จึงมีสองมิติ(ปริภูมิค่า). ปริภูมิ ρ_1 มีแค่เวกเตอร์เดียวที่เป็นอิสระ เชิงเส้น ดังนั้น ρ_1 จึงมีหนึ่งมิติ(ปริภูมิค่า).

การแยกส่วนประกอบเชิงตั้งฉาก. ถ้ากำหนด ρ เป็นปริภูมิย่อยของ \mathbb{R}^n และกำหนดให้ ρ^\perp เป็นส่วนเดิมเต็มเชิงตั้งฉาก (orthogonal complement) ของ ρ โดย ρ^\perp ประกอบด้วยเวกเตอร์ทั้งหมด ที่ แต่ละเวกเตอร์ตั้งฉากกับทุกเวกเตอร์ใน ρ . นั่นคือ

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

เชต ρ^\perp เองก็เป็นปริภูมิย่อยของ \mathbb{R}^n ด้วย. เวกเตอร์ทั้งหมดจากทั้งสองเซต ρ และ ρ^\perp แห่งทั่ว \mathbb{R}^n . นั่นคือ เวกเตอร์ใด ๆ $\mathbf{x} \in \mathbb{R}^n$ สามารถแยกส่วนประกอบออกได้

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

³สัญกรณ์เซต $\{a_i : c(a_i)\}$ สำหรับ $i = 1, \dots, k$ หมายถึง เซตที่มีสมาชิกเป็น a_i ถ้า a_i ผ่านเงื่อนไขที่ระบุหลังเครื่องหมายจุด คือ นั่นคือ ถ้า a_i ผ่านเงื่อนไข $c(a_i)$ และ ค่า a_i จะเป็นสมาชิกของเซต แต่ ถ้า a_i ไม่ผ่านเงื่อนไข $c(a_i)$ และ ค่า a_i จะไม่เป็นสมาชิก ของเซต โดยพิจารณา a_i สำหรับ $i = 1, \dots, k$.

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

จากตัวอย่างในรูปที่ 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. หมายเหตุ วิธีการวิเคราะห์ล้วนประกอบหลัก (Principal Component Analysis ดู [15, §12.1] เพิ่มเติม) ซึ่งเป็นวิธีที่นิยมใช้เพื่อลดมิติปริภูมิของข้อมูล เพื่อช่วยการประมวลผล และเพื่อช่วยการนำเสนอภาพของข้อมูล ก็ใช้แนวคิดหลักจากการฉายภาพเชิงตั้งฉาก.

เวกเตอร์ลักษณะเฉพาะและค่าลักษณะเฉพาะ

การแปลงข้อมูลเชิงเส้นมีการใช้อย่างกว้างขวาง และเครื่องมือสำคัญที่นิยมใช้ในการวิเคราะห์การแปลงข้อมูลเชิงเส้น คือ เวกเตอร์ลักษณะเฉพาะ และ ค่าลักษณะเฉพาะ.

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

$$\mathbf{Av} = \lambda v \tag{2.26}$$

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{Av}) 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 มีอ คือ $\lambda_1 \cdot \mathbf{v}_1$. ดังนั้น λ_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} เป็นเมทริกซ์เอกลักษณ์. กรณีที่ $\mathbf{v} = \mathbf{0}$ เป็นกรณีที่ชัดเจนแต่ไม่น่าสนใจ (*trivial*). พิจารณากรณีที่ $\mathbf{v} \neq \mathbf{0}$ สำหรับกรณีนี้ $\det[\mathbf{A} - \lambda\mathbf{I}] = 0$ (เพราะว่า ถ้าดิเทอร์มิแนนต์นี้ไม่เท่ากับศูนย์ จะแก้สมการได้เป็น $\mathbf{v} = [\mathbf{A} - \lambda\mathbf{I}]^{-1}\mathbf{0} = \mathbf{0}$ ซึ่งจะเป็นกรณีแรก ที่ชัดเจนแต่ไม่น่าสนใจ)

จาก $\det[\mathbf{A} - \lambda\mathbf{I}] = 0$ แทนค่าและแก้สมการหาค่า λ แล้วใช้ค่า λ 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 λ 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 $\lambda_2 = 1.02$. นำค่าลักษณะเฉพาะแต่ละค่าไปหาเวกเตอร์ที่คู่กัน ได้แก่ สำหรับ $\lambda_1 = 11.28$ คำนวนหาเวกเตอร์ลักษณะเฉพาะ \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 $b = 0.41a$ ดังนั้นเวกเตอร์ลักษณะเฉพาะคือ $[1, 0.41]^T$ หรือ $[2, 0.82]^T$ หรือ $[3, 1.32]^T$ หรือ เวกเตอร์ใด ๆ ที่อยู่ในรูป $\alpha[1, 0.41]^T$ เมื่อ α เป็นจำนวนจริงใด ๆ ที่ $\alpha \neq 0$. สังเกตว่า เวกเตอร์ทั้งหมดที่ได้จะซึ่งเป็นพิธีทางเดียวกัน (หรือตรงกันข้ามก็ได้ แต่ไม่ซึ่งเป็นทางอื่น) แต่ต่างขนาดกันเท่านั้น.

เพื่อความสะดวก เวกเตอร์ลักษณะเฉพาะมักจะเลือกให้เป็นเวกเตอร์ขนาดหนึ่งหน่วย. ดังนั้น จาก เวกเตอร์ $[1, 0.41]^T$ 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 $\lambda_1 = 11.28$ และใน

ทำนองเดียวกัน ได้เวกเตอร์ลักษณะเฉพาะ $\mathbf{v}_2 = [0.38, -0.93]^T$ คู่กับค่าลักษณะเฉพาะ $\lambda_2 = 1.02$. หมายเหตุ ถึงแม้จะเลือกเวกเตอร์ลักษณะเฉพาะ ให้เป็นเวกเตอร์หนึ่งหน่วยแล้วก็ตาม แต่ เวกเตอร์ลักษณะเฉพาะก็ยังอาจเลือกได้หลายแบบอยู่ เช่น $\mathbf{v}_1 = [-0.93, -0.38]^T$ หรือ $\mathbf{v}_2 = [-0.38, 0.93]^T$ ที่อยู่ในแนวเดียวกัน มีขนาดเป็นหนึ่งเหมือนกัน เพียงแต่มีทิศตรงกันข้าม.

2.2 ความน่าจะเป็น

ปัจจัยสำคัญเรื่องหนึ่ง โดยเฉพาะกับงานการรู้จำรูปแบบและการเรียนรู้ของเครื่อง คือความไม่แน่นอน (*uncertainty*). ความไม่แน่นอนอาจจะมาจากการหลายสาเหตุ เช่น ความไม่เที่ยงของเครื่องมือ หรือ วิธีการวัด หรือวิธีการเก็บข้อมูล หรืออาจมาจากการสัญญาณรบกวน หรือมาจากการขาดของข้อมูลที่จำกัด หรือแม้แต่ธรรมชาติความหลากหลายและความแปรปรวนของข้อมูลเอง. ทฤษฎีความน่าจะเป็น (*probability theory*) เป็นแนวทางหนึ่งที่ให้กรอบวิธีการสำหรับการวัด และการจัดการกับความไม่แน่นอน. ดังนั้น ทฤษฎีความน่าจะเป็น⁴ จึงเป็นพื้นฐานที่สำคัญสำหรับการเรียนรู้ของเครื่องและการรู้จำรูปแบบ.

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 $\{\text{'ร'}, \text{'ท'}, \text{'ณ'}, \text{'ต'}, \text{'ด'}\}$ 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 Ω มักใช้แทนเซตของค่าที่เป็นไปได้ทั้งหมด.

⁴เนื้อหาในส่วนนี้ได้รับอิทธิพลหลักจาก [15] [61] [133] และ [80].

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 $a \in A$ แล้ว $a \in A \cup B$ และถ้า $b \in B$ แล้ว $b \in A \cup B$ 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 $c \in A \setminus B$ แล้ว $c \in A$ และ $c \notin B$ และในทางกลับกัน ถ้า $a \in A$ และ $a \notin B$ แล้ว $a \in A \setminus B$.

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

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 จากตัวอย่าง⁵ หากสมมติว่า การทดลองทำซ้ำ ๆ เป็นจำนวน N ครั้ง โดยให้สภาพแวดล้อมเหมือนเดิมมากที่สุด. กำหนดให้ A เป็นเหตุการณ์ที่สนใจ โดย A อาจจะเกิดขึ้นหรือไม่เกิดก็ได้ ในแต่ละซ้ำ. หากสิ่งที่พบคือ เมื่อจำนวนทำซ้ำ N ใหญ่ขึ้น อัตราส่วนของจำนวนครั้ง

⁵ ตัวอย่างนี้อธิบายตามมุมมองแบบความถี่นิยม (*frequentist*). ความน่าจะเป็น อาจถูกมองได้ตามมุมมองแบบความถี่นิยม หรือตามมุมมองแบบเบสต์ (*bayesian*). มุมมองแบบความถี่นิยม จะมองความน่าจะเป็น เป็นความถี่จากการทำซ้ำต่อซ้ำอีก แต่�ุมมองแบบเบสต์ จะมองความน่าจะเป็น เป็นการประมาณโอกาสที่เหตุการณ์จะเกิด โดยเน้นที่รرمชาติของโอกาสเอง ไม่ได้มองเป็นความถี่จากการทำซ้ำ. มุมมองแบบเบสต์จะกว้างกว่า มุมมองแบบความถี่นิยมที่คำอธิบายจะเข้าได้ยากกับเหตุการณ์ที่ทำซ้ำไม่ได้ เช่น การประมาณอายุของดาว ที่ไม่เกี่ยวกับการทำซ้ำ แต่เป็นการประมาณ ที่เมื่อมีหลักฐาน หรือข้อมูลเข้ามาใหม่ ความน่าจะเป็นของอายุของดาว ก็สามารถจะถูกประเมินเป็นค่าใหม่ได้.

ที่จะเกิด A ในแต่ละซ้ำ จะมีค่าเข้าใกล้ค่า ๆ หนึ่งมากขึ้น. ค่าที่อัตราส่วนเข้าใกล้มากขึ้นเมื่อจำนวนซ้ำมากขึ้น คือค่าความน่าจะเป็นของ A .

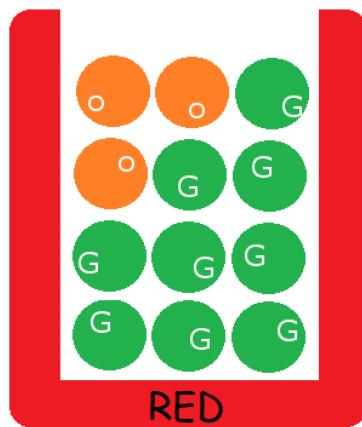
ขยายความคือ หากกำหนดให้ $N(A)$ แทนจำนวนครั้งที่จะเกิดเหตุการณ์ A ในการทำซ้ำทั้งหมด N ครั้ง อัตราส่วน $\frac{N(A)}{N}$ จะค่อย ๆ ลู่เข้าสู่ค่าค่าหนึ่ง เมื่อ N เพิ่มขึ้น. ค่าที่อัตราส่วนลู่เข้า จะเรียกว่า ความน่าจะเป็นที่เหตุการณ์ A จะเกิดขึ้น. ค่าความน่าจะเป็นของเหตุการณ์ A แทนด้วยสัญลักษณ์ $\Pr(A)$ และความน่าจะเป็นจะมีค่าอยู่ระหว่าง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

ตัวอย่างเช่น ลังใส่ลูกบอลสีต่าง ๆ ดังแสดงในรูปที่ 2.3 หากสุ่มหยิบลูกบอล ออกมายก ลัง 1 ลูก และกำหนดให้ A เป็นเหตุการณ์ที่หยิบได้ลูกบอลสีเขียว. สมมติทำการทดลอง(สุ่มหยิบ)ซ้ำ $N = 10$ ผลการจำลองเหตุการณ์ (*simulation*) ดังแสดงในรูปที่ 2.4 ซึ่งบอกได้ว่า อัตราส่วนที่หยิบได้ลูกบอลสีเขียว เป็น $\frac{N(A)}{N} = \frac{8}{10} = 0.8$. หากเพิ่มจำนวนการทำซ้ำ N จาก 10 เป็น 100 แล้วเพิ่มเป็น 1000 แล้วเป็น 10000 แล้วทำต่อ ๆ ไป จะเริ่มเห็นว่าอัตราส่วน $\frac{N(A)}{N}$ ลู่เข้าสู่ค่าค่าหนึ่ง ดังแสดงในตารางที่ 2.2. เมื่อนำค่าต่าง ๆ ไปplotกราฟ จะได้ดังรูปที่ 2.5 ซึ่งจะเห็นว่าค่าที่ อัตราส่วน $\frac{N(A)}{N}$ ลู่เข้าหาคือ 0.75. นั่นคือ ความน่าจะเป็นของการสุ่มหยิบได้ลูกบอลสีเขียว $\Pr(A) = 0.75$. มองจากอีกมุมหนึ่ง ในลังมีลูกบอล 12 ลูก และเป็นลูกบอลสีเขียวอยู่ 9 ลูก หากสุ่มหยิบอย่างยุติธรรม (แต่ละลูกมีโอกาสสูญเสียเท่า ๆ กัน) โอกาสที่จะหยิบได้ลูกเขียวจะเป็น $\frac{9}{12} = 0.75$ ซึ่งค่าที่คำนวนนี้สอดคล้องกับค่าความน่าจะเป็นที่ได้จากการจำลองเหตุการณ์ข้างต้น.

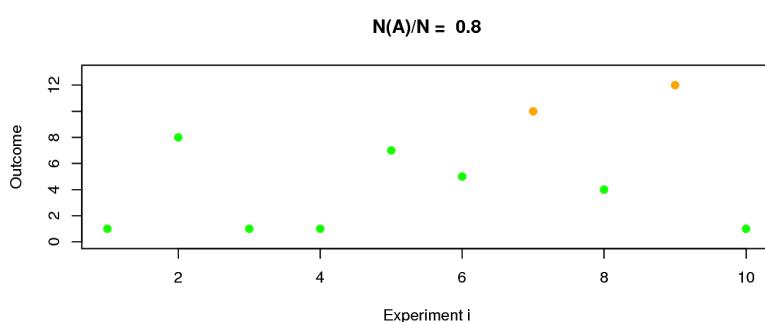
มุมมองของความน่าจะเป็น. เหตุการณ์อาจมองได้ในหลาย ๆ ระดับ เช่น เหตุการณ์ระดับล่าง ได้แก่ เหตุการณ์ที่หยิบได้ลูกบอลลูกที่หนึ่ง a_1 (ซึ่งเป็นสีเขียว) ไปจนถึง เหตุการณ์ที่หยิบได้ลูกบอลลูกที่สิบสอง a_{12} (ซึ่งเป็นสีส้ม) หรือเหตุการณ์ระดับที่สูงขึ้น ได้แก่ เหตุการณ์ที่หยิบได้ลูกบอลสีเขียว A เหตุการณ์ที่หยิบได้ลูกบอลสีส้ม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

ตารางที่ 2.1 ภาษาเฉพาะที่ใช้ในเรื่องเซตกับเรื่องความน่าจะเป็น

สัญลักษณ์ทั่วไป	ภาษาเฉพาะในเรื่องเซต	ภาษาเฉพาะในเรื่องความน่าจะเป็น
Ω	กลุ่มของค่าที่เป็นไปได้ทั้งหมด	ปริภูมิตัวอย่าง หรือผลลัพธ์ทั้งหมดที่เป็นไปได้
$a \in \Omega$	ค่าหนึ่งที่เป็นไปได้	รูปแบบของผลลัพธ์ของเรื่องที่สนใจ หรือเหตุการณ์พื้นฐาน
$A \subset \Omega$	เซตย่อยของ Ω	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



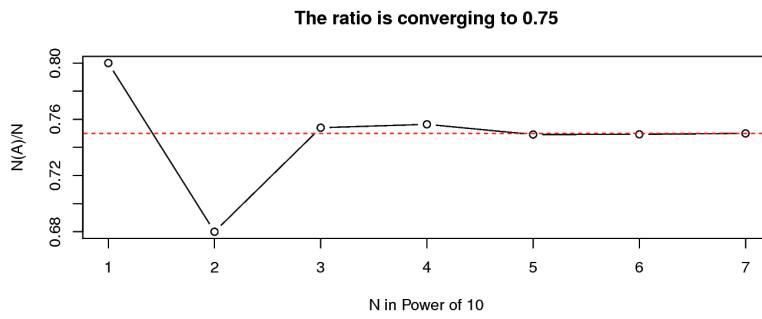
รูปที่ 2.3 ตัวอย่างความน่าจะเป็น ลังใส่ลูกบอล ซึ่งมีลูกบอลอยู่ภายใน 12 ลูก เป็นลูกบอลสีส้มสามลูก และที่เหลือเป็นสีเขียว



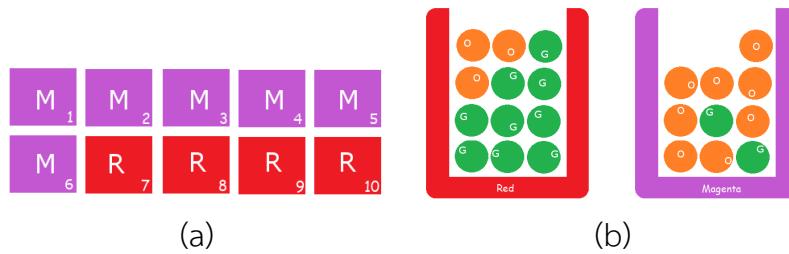
รูปที่ 2.4 ผลจากการจำลองเหตุการณ์สุ่มหยิบลูกบอล 10 ครั้ง จากกล่องลูกบอลที่แสดงในรูปที่ 2.3. ลูกที่ 1–9 สีเขียว ลูกที่ 10–12 สีส้ม. จากการสุ่มทำ 10 ครั้ง มีครั้งที่ 7 และ 9 ที่หยิบได้ลูกบอลสีส้ม. ดังนั้น อัตราส่วนจำนวนครั้งที่หยิบได้ลูกบอลสีเขียว คือ 0.8 (ระบุที่ด้านบนของภาพ)

ตารางที่ 2.2 อัตราส่วนของการสุ่มหยิบได้ลูกบอลสีเขียว 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 2.5 การลุ่เข้าของอัตราส่วนการหยิบได้สีเขียว. อัตราส่วน $\frac{N(A)}{N}$ ลุ่เข้าหา $\Pr(A) = 0.75$ เมื่อ N เพิ่มขึ้น (แสดงด้วยเส้นประสีแดง)



รูปที่ 2.6 ตัวอย่างความน่าจะเป็นแบบมีเงื่อนไข ลังบรรจุลูกบอลสี. ภาพ (a) แสดงจำนวนลังสีฟ้ากับลังสีแดง (มีลังสีแดงอยู่ 4 ลัง ที่เหลือเป็นสีฟ้า). ภาพ (b) แสดงลูกบอลสีภายในลัง โดย ลังซ้ายสีแดงมีลูกบอลสีส้มอยู่ 3 ลูก ที่เหลือสีเขียว และลังขวาสีฟ้ามีลูกบอลสีเขียวอยู่ 2 ลูก ที่เหลือสีส้ม

คุณสมบัติของความน่าจะเป็น. ความน่าจะเป็นมีคุณสมบัติที่น่าสนใจหลายอย่าง เช่น ความน่าจะเป็นที่จะเกิดผลลัพธ์จากกลุ่มผลลัพธ์ทั้งหมดทุกแบบที่เป็นไปได้ คือ ต้องพบรณ์อน ความน่าจะเป็นมีค่าสูงสุด. นั่นคือ $\Pr(\Omega) = 1$. ความน่าจะเป็นที่จะพบรณ์ที่เป็นไปไม่ได้ คือ ต้องไม่พบรณ์อน 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 $A \cap B = \emptyset$ แล้วจะเรียกว่า 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 $B_i \cap B_j = \emptyset$ สำหรับทุก ๆ ค่าของ $i \neq j$ ตั้งแต่ $1, \dots, n$ และ กฎของความน่าจะเป็นรวม (law of total probability) กล่าว

ว่า

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

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

ตัวแปรสุ่ม. เพื่อความสะดวก เหตุการณ์อาจถูกระบุด้วย ตัวแปรสุ่ม เช่น จากตัวอย่าง เหตุการณ์ที่ หยิบได้ลูกบอลสีเขียว และเหตุการณ์ที่หยิบได้ลูกบอลสีฟ้า จะสามารถถูกอ้างถึงได้สะดวก และซัดเจน กว่า ถ้ากำหนด ตัวแปรสุ่ม B แทนสีของลูกบอลที่ถูกสุ่มหยิบขึ้นมา. เหตุการณ์ที่หยิบได้ลูกบอลสีเขียว สามารถ เขียนเป็น $B = 0$ เมื่อ 0 แทนสีเขียว และ เหตุการณ์ที่หยิบได้ลูกบอลสีฟ้า สามารถ เขียนเป็น $B = 1$ เมื่อ 1 แทนสีฟ้า. **ตัวแปรสุ่ม (random variable)** อาจถูกนิยามว่า เป็นตัวแปรที่ค่าของมัน ขึ้นกับผลลัพธ์ของเรื่องที่สนใจ เมื่อเรื่องที่สนใจเป็นกระบวนการที่มีความไม่แน่นอนอยู่.

หมายเหตุ ตัวแปรสุ่มเป็นการแทนเหตุการณ์ด้วยค่าตัวเลข โดยสำหรับรرمชาติของเหตุการณ์ที่ ไม่ได้เป็นตัวเลข การใช้งานตัวแปรสุ่มนี้อาจทำได้โดยการกำหนดความหมายให้กับตัวเลข เช่น 0 แทนสีเขียว และ 1 แทนสีฟ้า. แต่หลาย ๆ ครั้งเพื่อความสะดวกและซัดเจน อาจมีการใช้สัญลักษณ์ เช่น ‘ σ ’ 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 ที่ตัวแปรสุ่ม มีค่าจนถึงค่า x ที่กำหนด ดังนั้นฟังก์ชันการแจกแจงนี้ บางครั้งอาจเรียก **ฟังก์ชันการแจกแจงสะสม (cumulative distribution function คำย่อ cdf)**.

ตัวแปรสุ่มอาจมีได้หลายแบบขึ้นกับลักษณะของค่าของมัน ซึ่งค่าของมันก็คือลักษณะของผลลัพธ์ที่ เป็นไปได้. **ตัวแปรสุ่มวิถุ (discrete random variable)** คือตัวแปรสุ่มที่ค่าของมัน อยู่ในเซตจำกัด (*finite set*) หรืออยู่ในเซตไม่จำกัดแต่นับได้ (*countably infinite set*). ตัวอย่าง ตัวแปรสุ่มสีของ ลูกบอล B นี้เป็นตัวแปรสุ่มวิถุ เนื่องจากค่าของมันมาจากเซตจำกัด ได้แก่ $\{0, 1\}$ (มีจำนวนสมาชิก น้อยกว่าค่าอนันต์ ∞). ตัวแปรสุ่มจำนวนต้นทุเรียน ก็เป็นตัวแปรสุ่มวิถุ เนื่องจากค่าของมันมาจากเซต ไม่จำกัดแต่นับได้ ได้แก่ $\{0, 1, 2, 3, \dots\}$. แต่ตัวแปรสุ่มปริมาณน้ำในอ่างเก็บน้ำ ไม่ใช่ตัวแปรสุ่มวิถุ เพราะว่า ค่าของมันมาจาก \mathbb{R}^+ ตัวแปรสุ่มปริมาณน้ำในอ่างเก็บน้ำ จะเป็นตัวแปรสุ่มต่อเนื่อง. หัวข้อ 2.2 ยกไปรายตัวแปรสุ่มต่อเนื่องเพิ่มเติม.

ตัวแปรสุ่มวิถุ 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 ค่าคาดหมายสามารถคำนวณได้จาก

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

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

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

ความน่าจะเป็นร่วม. เมื่อใช้ตัวแปรสุ่มอธิบายเหตุการณ์ที่เกี่ยวข้องกับตัวแปรสุ่มตั้งแต่สองตัวขึ้นไป ความน่าจะเป็นที่ใช้จะเรียกว่า 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 สำหรับตัวแปรสุ่มวิญญาณ $\text{cov}[X, Y] = \sum_x \sum_y (x - E[X])(y - E[Y]) \cdot \Pr(X = x, Y = y)$.

ความน่าจะเป็นแบบมีเงื่อนไข

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

จากตัวอย่างของรูปที่ 2.3 พิจารณาตัวอย่างที่คราวนี้ มีลังอยู่ 10 ลัง ซึ่ง เป็นลังสีแดง 4 ลัง และ เป็นลังสีบานเย็น 6 ลัง ดังรูปที่ 2.6. ถ้าสุ่มยกมาหนึ่งลัง โอกาสที่จะเป็นลังที่ 7 คือ $1/10$ แต่ถ้าเห็นว่า ลังที่สุ่มมาเป็นสีแดง โอกาสที่จะเป็นลังที่ 7 คือ $1/4$ เพราะว่า มีแค่ลังที่ 7 ถึงลังที่ 10 ที่เป็นสีแดงอยู่แค่ 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 **กฎรวม** (sum rule). สังเกตว่า การใช้กฎผลรวม จะลดตัวแปรสุ่มลงไป ซึ่ง การทำเช่นนี้ จึงอาจถูกเรียกว่า การถลายปั๊จจัย (marginalization).

กฎผลคูณสามารถใช้ต่อเนื่องกัน ในลักษณะถูกโฉم

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

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

กฎของเบส์ (Bayes' rule หรือ Bayes' theorem) คือ

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

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

จากความสัมพันธ์ที่ได้จากกฎของเบส์ การอนุมานค่าที่สนใจจากข้อมูล มักจะเรียกชื่อพจน์ต่าง ๆ ใน สมการ 2.36 เพื่อความสะดวก ดังนี้ ถ้ากำหนดให้ตัวแปรสุ่ม Y แทนเป้าหมายของการอนุมาน และ ตัวแปรสุ่ม 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 ฟังก์ชันควรจะเป็น (likelihood function คำย่อ likelihood). ดังนั้น จากกฎของเบส์ และชื่อพจน์ต่าง ๆ อาจสรุปความสัมพันธ์ได้เป็น $\text{posterior} \propto \text{likelihood} \cdot \text{prior}$.

ตัวอย่างการคำนวณ. กลับมาที่รูปที่ 2.6 อีกครั้ง คราวนี้จะสุมเลือกลัง และพอดีลังแล้วก็จะสุมหยิบลูกบลอกรถมา. โอกาสที่จะสุมได้ลังแดงเป็น $\frac{4}{10}$ หรือความน่าจะเป็นที่จะได้ลังสีแดง $\Pr(C = 'r') = 0.4$ โดย $C = 'r'$ แทนเหตุการณ์ที่จะได้ลังสีแดง. ในทำนองเดียวกัน ความน่าจะเป็นที่จะได้ลังสีบานเย็น $\Pr(C = 'm') = 0.6$.

ถ้ารู้ว่าเป็นลังสีแดง เมื่อสุมหยิบลูกบลอกรถมา โอกาสที่จะหยิบได้ลูกบลสีเขียว คือ $\frac{9}{12} = 0.75$ หรือ เที่ยวนเป็นสัญกรณ์ ได้ว่า $\Pr(B = 'g' | C = 'r') = 0.75$ โดย $B = 'g'$ แทนเหตุการณ์ที่จะหยิบได้ลูกบลสีเขียว. ทำนองเดียวกัน ก็จะได้ความน่าจะเป็นแบบนี้เช่นนี้อีก ๑ ดังนี้

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

สังเกตว่า ความน่าจะเป็นที่จะหยิบได้ลูกบลสีเขียวเมื่อรู้ว่าลังสีแดง $\Pr(B = 'g' | C = 'r')$ ไม่เหมือนกับความน่าจะเป็นที่จะหยิบลูกบลสีเขียวและลุ่มได้ลังสีแดง $\Pr(B = 'g', C = 'r')$. สำหรับ $\Pr(B = 'g' | C = 'r') = 0.75$ นั้นไม่ต้องสนใจเลยว่าโอกาสที่จะได้ลังสีแดงเป็นเท่าไร. ในขณะที่ $\Pr(B = 'g', C = 'r')$ จะประกอบด้วยโอกาสที่จะได้ลังสีแดง $\Pr(C = 'r') = 0.4$ และโอกาสที่จะหยิบได้ลูกบลสีเขียวจากลังนั้น $\Pr(B = 'g' | C = 'r') = 0.75$ 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 ดังแสดงในตารางที่ 2.3.

ทบทวน (1) ผลรวมของความน่าจะเป็นของทุก ๆ เหตุการณ์เป็น ๑. 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 เท่ากับผลรวมของความน่าจะเป็นของเหตุการณ์ X และ Y สำหรับทุก ๆ ความเป็นไปได้ของ Y ดังเช่น

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

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

ธรรมชาตินี้คือกฎผลบวก (สมการ 2.34).

จากกฎของการบวก จะได้ ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีเขียว และความน่าจะเป็นที่จะได้ลูกบอลสีส้ม (โดยไม่นับใจสีของลัง)

$$\Pr(B = 'g') = \Pr(C = 'r', B = 'g') + \Pr(C = 'm', B = 'g')$$

$$= 0.3 + 0.12 = 0.42$$

$$\Pr(B = 'o') = \Pr(C = 'r', B = 'o') + \Pr(C = 'm', B = 'o')$$

$$= 0.1 + 0.48 = 0.58.$$

และความน่าจะเป็นของลังถ้าหากรู้สีของลูกบอลที่สุ่มหยิบออกจากหนึ่งลูก

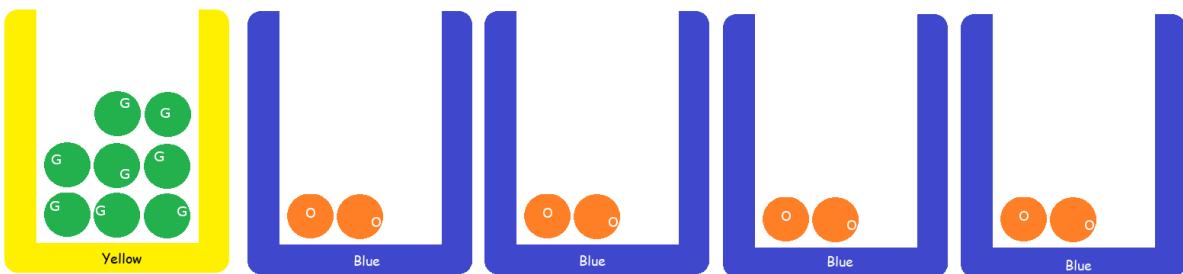
$$\begin{aligned} \Pr(C = 'r' | B = 'g') &= \frac{\Pr(B = 'g' | C = 'r') \cdot \Pr(C = 'r')}{\Pr(B = 'g')} \\ &= \frac{(0.75)(0.4)}{0.42} = 0.71. \end{aligned}$$

ความน่าจะเป็นแบบมีเงื่อนไข และทฤษฎีของเบล์ ช่วยให้สามารถหาค่าความน่าจะเป็นที่สนใจได้ จากค่าของความน่าจะเป็นอื่นที่ประเมินความน่าจะเป็นได้ง่ายกว่า เช่น $\Pr(B|C)$ จะประเมินได้ง่าย เพราะว่า ลูกบอลอยู่ในลัง ดังนั้นจะนับได้ง่ายว่า ในลังแต่ละสี มีลูกบอลสีไหนจำนวนเท่าไร ต่อจำนวนลูกบอลทั้งหมดในลัง. $\Pr(C)$ ก็ประเมินได้ง่าย แต่ $\Pr(C|B)$ ประเมินตรง ๆ ได้ยาก เพราะลูกบอลแต่สีกระจายไปทุก ๆ ลัง.

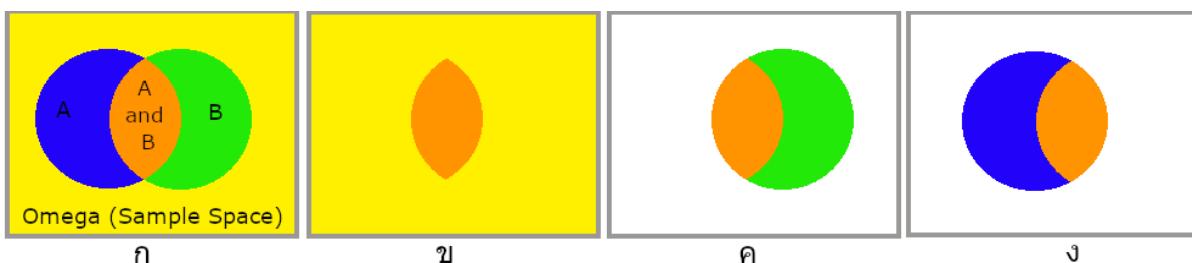
ตารางที่ 2.3 สรุปค่าความน่าจะเป็นร่วม ของตัวอย่างการสุ่มลังและลูกบอล

ลัง C	ลูกบอล B	
	เขียว 'g'	ส้ม 'o'
แดง 'r'	0.3	0.1
บานเย็น 'm'	0.12	0.48

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



รูปที่ 2.7 ตัวอย่างเพิ่มเติมความน่าจะเป็นแบบมีเงื่อนไข เน้นความต่างระหว่างสุ่มเลือกลังแล้วสุ่มเลือกลูกบอล (ในภาพ) เปรียบเทียบกับเหลกบอลทั้งหมดมาร่วมกัน แล้วสุ่มเลือกลูกบอล (ไม่มีภาพ)

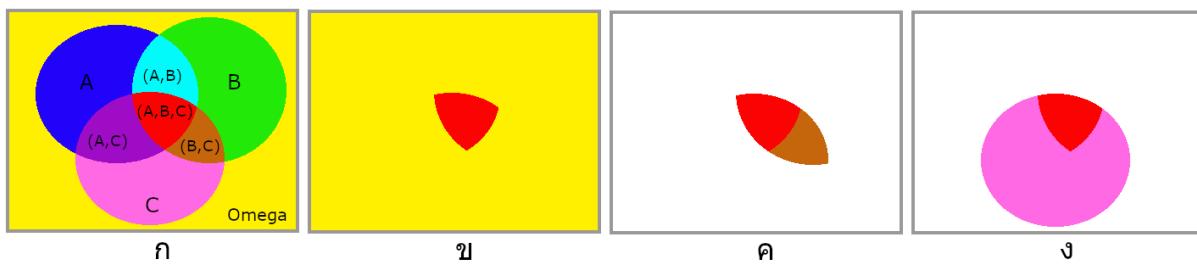


รูปที่ 2.8 ความเกี่ยวเนื่องของสองเหตุการณ์. ภาพ ก แสดงเหตุการณ์ A ด้วยวงกลมซ้าย และเหตุการณ์ B ด้วยวงกลมขวา พื้นที่ที่ทั้งสองกันคือ เหตุการณ์ร่วม $A \cap B$ พื้นที่ทั้งหมดในกรอบคือผลลัพธ์ทุก ๆ แบบที่เป็นไปได้ (Ω หรือปริภูมิตัวอย่าง). ภาพ ข ความน่าจะเป็น $\Pr(A \cap B)$ มองโอกาสเกิดเหตุการณ์ร่วม $A \cap B$ จากบริบทของทุก ๆ ผลลัพธ์ที่เป็นไปได้. ภาพ ค ความน่าจะเป็น $\Pr(A|B)$ มองโอกาสเกิดเหตุการณ์ร่วม $A \cap B$ จากบริบทของเหตุการณ์ B . ภาพ จ ความน่าจะเป็น $\Pr(B|A)$ มองโอกาสเกิดเหตุการณ์ร่วม $A \cap B$ จากบริบทของเหตุการณ์ A .

สองกรณีนี้ จะเห็นต่างกันชัดเจนมาก ถ้าพิจารณากรณี เช่น ตัวอย่างในรูปที่ 2.7 มีลังสีเหลืองแค่ 1 ลัง มีลังสีฟ้า 4 ลัง แต่ลังสีเหลืองมีลูกบอล 8 ลูก ที่ทั้งหมดสีเขียว และลังสีฟ้ามีลูกบอล 2 ลูก ที่ทั้งหมด สีส้ม. เมื่อคิดความน่าจะเป็นแล้วจะพบว่า กรณีนี้ ถ้าเหลกบอลทั้งหมดออกจากลัง แล้วสุ่มหยิบโอกาส ที่จะได้สีเขียวเป็น $8/16 = 1/2$ แต่ถ้าสุ่มเลือกลังก่อน ลังสีเหลืองมีโอกาสแค่ $1/5$ แล้วโอกาสได้ลูกบอลสีเขียวเป็น $4/5$ แต่โอกาสได้ลูกบอลสีเขียวเป็น 0 ดังนั้นโอกาสได้ลูกบอลสีเขียวจะเป็นแค่ $(1/5) \cdot 1 + (4/5) \cdot 0 = 1/5$.

ภาพความเกี่ยวเนื่องของเหตุการณ์ และความน่าจะเป็นร่วม และความน่าจะเป็นแบบมีเงื่อนไข แสดงในรูปที่ 2.8 สำหรับสองเหตุการณ์ และรูปที่ 2.9 สำหรับสามเหตุการณ์.

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. ความหมายก็คือ ถ้าเหตุการณ์ A และเหตุการณ์ B เป็นอิสระต่อกันแล้ว การรู้หรือไม่รู้ข้อมูลของ B ก็ไม่ได้เปลี่ยนการประมาณค่าของ A .



รูปที่ 2.9 ภาพแสดงความเกี่ยวเนื่องของเหตุการณ์สามเหตุการณ์ A เหตุการณ์ B เหตุการณ์ C ด้วยวงกลม พื้นที่ที่ทับซ้อนกันແນะเหตุการณ์ร่วม ฉลาก เช่น (A, B) ระบุเหตุการณ์ร่วม $A \cap B$ และ (A, B, C) ระบุเหตุการณ์ร่วม $A \cap B \cap C$. พื้นที่ทั้งหมดในกรอบคือผลลัพธ์ทุก ๆ แบบที่เป็นไปได้ (Ω). 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 .

ตัวอย่างการใช้งานความน่าจะเป็นแบบมีเงื่อนไข

ปัญหาการตรวจเต้านมด้วยภาพเอ็กซเรย์. สมมติว่า ผู้หญิงอายุสี่สิบปีคนหนึ่ง ไปทำการตรวจเต้านมด้วยภาพเอ็กซเรย์ (mammogram) และผลตรวจเป็นบวก (positive ซึ่งหมายถึง เครื่องตรวจบอกว่า เป็นมะเร็ง) โอกาสที่ผู้หญิงคนนี้จะเป็นมะเร็งจริง ๆ เป็นเท่าไร

สมมติว่าข้อมูลประกอบ คือ วิธีการตรวจเต้านมด้วยเอ็กซเรย์มีค่าความไว (sensitivity) ที่ 80% ซึ่งหมายความว่า 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).

ความเข้าใจโดยอย่างหนึ่งที่พบบ่อย คือ การสรุปว่า ผู้หญิงคนนั้นมีโอกาสเป็นมะเร็ง 80% ซึ่งผิดเพระการสรุปนี้ ไม่ได้คำนึงถึงความน่าจะเป็นก่อน นั่นคือ โอกาสที่ผู้หญิงอายุสี่สิบปี จะเป็นมะเร็งเต้านม ซึ่งจากสถิติ⁶ คือ 17%. นั่นคือ $\Pr(C = 1) = 0.17$.

นอกจากนั้น การสรุปยังต้องการข้อมูลว่า วิธีการตรวจนี้ผลบวกผิด (false positive หรือลัญญาณหลอก false alarm) เป็นเท่าไร ถ้าวิธีการตรวจนี้ผลบวกผิด 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%.

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

ปัญหามอนตี้霍ล. ปัญหามอนตี้霍ล (Monty Hall Problem) เป็นสถานการณ์การตัดสินใจของผู้เข้าแข่งขันเกมส์โชว์มอนตี้霍ล. ผู้เข้าแข่งขันต้องเลือกเปิดประตูหนึ่งในสามประตู. มีประตูหนึ่งที่ซ่อนรางวัลใหญ่ไว้. อีกสองประตูมีแต่ของปลอกใจ. ผู้เข้าแข่งขันจะได้อะไรก็ตามที่อยู่หลังประตูกลับบ้าน. แต่หลังจากผู้เข้าแข่งขันเลือกประตูไปแล้ว แทนที่พิธีกรจะเปิดประตูนั้นออกทันที. พิธีกรจะเดินไปที่ประตู และเลือกเปิดประตูหนึ่ง ที่ผู้เข้าแข่งขันไม่ได้เลือก. ประตูที่พิธีกรเปิด จะไม่มีรางวัลอยู่ และเสนอโอกาสให้ผู้เข้าแข่งขันเปลี่ยนใจ ไปเลือกประตูที่เหลืออยู่. 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.

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. โอกาสที่รางวัลจะอยู่ประตูไหน มีเท่า ๆ กัน. นั่นคือ $\Pr(R = 1) = \Pr(R = 2) = \Pr(R = 3) = 1/3$ และ เพราะประตูที่มีรางวัลเป็นอิสระกับประตูที่ผู้แข่งขันเลือก ดังนั้น $\Pr(R) = \Pr(R|C)$. นั่นคือ $\Pr(R = 1|C = 1) = \Pr(R = 2|C = 1) = \Pr(R = 3|C = 1) = 1/3$.

แต่พิธีกรต้องไม่เปิดประตูที่ผู้แข่งขันเลือก หรือไม่เปิดประตูที่มีรางวัล ดังนั้น

$\Pr(H = 2|C = 1, R = 1) = 1/2$ เพราะ พิธีกรเลือกเปิดประตูที่สองหรือที่สามก็ได้

$\Pr(H = 2|C = 1, R = 2) = 0$ เพราะ พิธีกรเปิดประตูที่มีรางวัลไม่ได้

$\Pr(H = 2|C = 1, R = 3) = 1$ เพราะ พิธีกรเปิดประตูที่สองได้เท่านั้น

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 ถ้าผู้แข่งขันเปลี่ยนประตู จะมากกว่า โอกาส ถ้าผู้แข่งขันไม่เปลี่ยน (เพราะว่า สมการ 2.38 และ 2.39 มีตัวหารเท่ากัน).

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

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

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

ตัวแปรสุ่มต่อเนื่อง

ตัวแปรสุ่ม X จะเรียกว่า เป็นตัวแปรสุ่มต่อเนื่อง (continuous random variable) ก็ต่อเมื่อ พังก์ชัน การแจกแจงของ 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 นี้จะ เรียกว่า พังก์ชันความหนาแน่นความน่าจะเป็น (probability density function) บางครั้งอาจเรียก พังก์ชันความหนาแน่น (density function คำย่อ pdf) ของ X .

สิ่งที่มักสับสน. ตัวแปรสุ่มต่อเนื่องมีคุณสมบัติหลายอย่างที่มักถูกเข้าใจผิด. ทั้งตัวแปรสุ่มต่อเนื่อง และตัวแปรสุ่มวิชุต ใช้บรรยายเหตุการณ์ ซึ่งเหตุการณ์จะสามารถนำไปหาค่าความน่าจะเป็นได้. ความ น่าจะเป็น ยังมีคุณสมบัติเหมือนเดิม ไม่ว่า จะเป็น ความน่าจะเป็นของเหตุการณ์ที่บรรยายด้วยตัวแปร

ลุ่มวิยุต หรือด้วยตัวแปรสุ่มต่อเนื่อง. นั่นคือ ความน่าจะเป็น มีค่าระหว่างศูนย์ถึงหนึ่ง และผลรวมของ ความน่าจะเป็นทั้งหมดเป็นหนึ่ง.

แต่ตัวแปรสุ่มวิยุตและตัวแปรสุ่มต่อเนื่องมีคุณสมบัติหลาย ๆ อย่างต่างกัน. หากกำหนดให้ D เป็นตัวแปรสุ่มวิยุต และ C เป็นตัวแปรสุ่มต่อเนื่อง สำหรับตัวแปรสุ่มวิยุต ความน่าจะเป็นของแต่ละ ผลลัพธ์ คือค่าฟังก์ชันมวลความน่าจะเป็นของค่าผลลัพธ์นั้น นั่นคือ $\Pr(D = d) = \text{pmf}(d)$. แต่ สำหรับตัวแปรสุ่มต่อเนื่อง ความน่าจะเป็นของแต่ละผลลัพธ์เป็นศูนย์เสมอ ไม่ว่าค่านั้นจะเป็นเท่าไร นั่น คือ $\Pr(C = c) = 0$.

แม้ว่า ความน่าจะเป็นของแต่ละค่าเป็นศูนย์ แต่ความน่าจะเป็นของช่วงค่าสามารถหาได้. วิธี ประเมินความน่าจะเป็น ในกรณีตัวแปรสุ่มต่อเนื่อง จะใช้ฟังก์ชันการแจกแจง $F(c) = \Pr(C \leq c)$ และความน่าจะเป็น $\Pr(C > c) = 1 - F(c)$. เมื่อต้องการประเมินความน่าจะเป็นเป็นช่วง ก็ สามารถทำได้โดย $\Pr(c_0 < C \leq c_1) = F(c_1) - F(c_0)$. หากต้องการประเมินความน่าจะ เป็นบริเวณรอบ ๆ ค่าใดก็สามารถทำได้โดย $\Pr(c - \varepsilon < C \leq c + \varepsilon) = F(c + \varepsilon) - F(c - \varepsilon)$ เมื่อ ε ระบุรูปแบบของบริเวณรอบ ๆ. ข้อควรระวัง ถ้า ε มีค่าน้อยมาก ๆ แล้ว $\Pr(c - \varepsilon < C \leq c + \varepsilon)$ จะใกล้กับศูนย์ (ความน่าจะเป็นของค่าจุดจุดหนึ่งของตัวแปรสุ่มต่อเนื่องเป็นศูนย์).

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

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

ตารางที่ 2.4 สรุปคุณสมบัติของตัวแปรสุ่มต่อเนื่อง ที่มักถูกเข้าใจผิด เปรียบเทียบกับคุณสมบัติ ของตัวแปรสุ่มวิยุตในประเด็นเดียวกัน.

ตารางที่ 2.4 คุณสมบัติที่มักสับสนของตัวแปรสุ่มต่อเนื่อง

ประเด็น	ตัวแปรสุ่มวิยุต 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	$F(d) = \Pr(D \leq d)$	$F(c) = \Pr(C \leq c)$
	$F(d) = \sum_{u \leq d} \text{pmf}(u)$	$F(c) = \int_{-\infty}^c \text{pdf}(u)du$
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 อธิบายการแจกแจงของตัวแปรลุ่มต่อเนื่อง 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 μ และ σ^2 เป็นพารามิเตอร์ของแบบจำลอง⁷. ฟังก์ชันการแจกแจงของการแจกแจงเกาส์เชียน ไม่มีรูปแบบปิด (closed form ซึ่งในคณิตศาสตร์ หมายถึง นิพจน์ที่สามารถเขียนโดยใช้การคำนวณพื้นฐานได้) และฟังก์ชันการแจกแจง มักเขียนในรูป

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

เมื่อ

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

รูปที่ 2.10 แสดงความสัมพันธ์ระหว่างพารามิเตอร์ μ กับ σ และผลต่อค่าความหนาแน่นความน่าจะเป็นของการแจกแจงเกาส์เชียน. ค่า μ จะควบคุมตำแหน่งที่มีความหนาแน่นสูงสุด. ค่า σ ควบคุมการแผ่. สังเกตว่า ความหนาแน่นความน่าจะเป็น มีค่าเกินหนึ่งได้ (ภาพล่างที่สองจากซ้าย $\sigma = 0.2$).

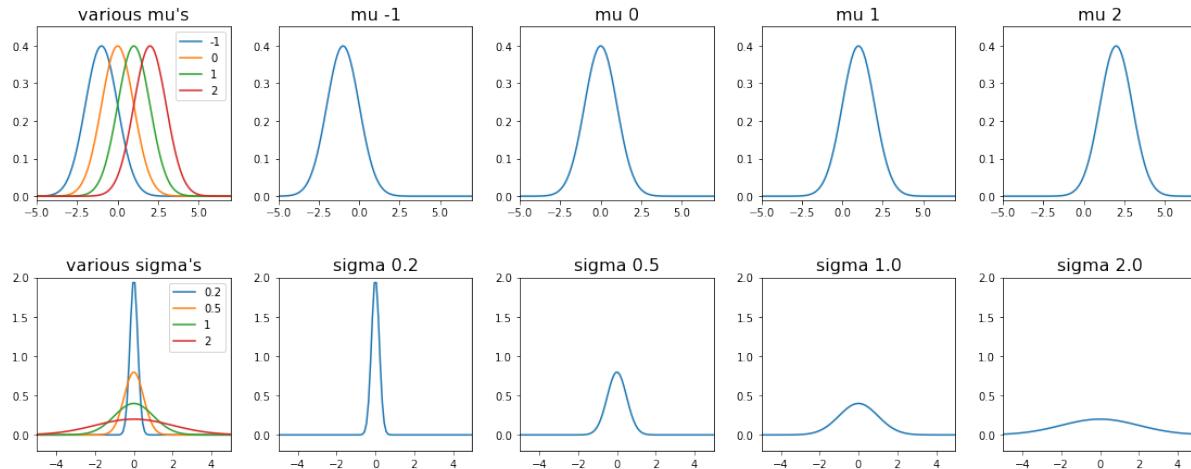
รูปที่ 2.11 แสดงความหนาแน่นความน่าจะเป็น (pdf) และการแจกแจงความน่าจะเป็น (cdf). สังเกตว่า การแจกแจงความน่าจะเป็น จะเป็นฟังก์ชันเพิ่ม (increasing function) เพราะว่า การแจกแจงความน่าจะเป็น $F(x) = \Pr(X \leq x)$ ดังนั้น ที่ค่ามากขึ้น ความน่าจะเป็นจะไม่มีทางน้อยลง และที่อนันต์ $F(\infty) = 1$. การแจกแจงความน่าจะเป็น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

2.3 การหาค่าดีที่สุด

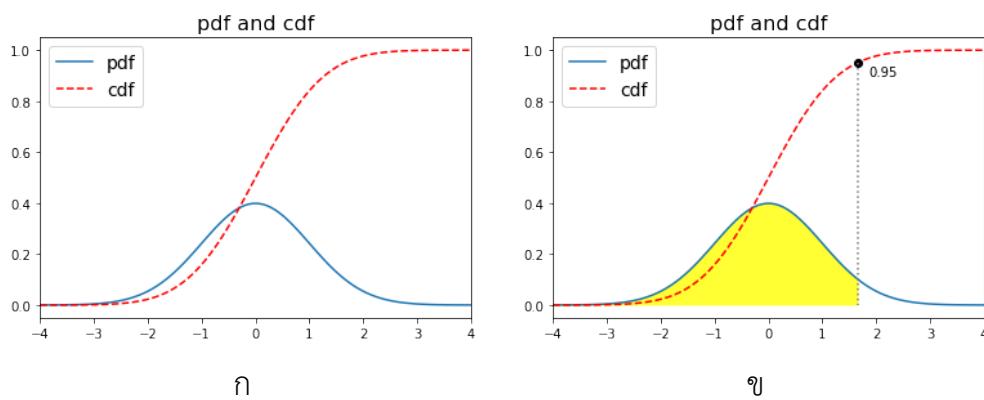
การรู้จักรูปแบบ และการเรียนรู้ของเครื่อง ถูกสร้างบนพื้นฐานของศาสตร์การหาค่าดีที่สุด⁸ การรู้จักรูปแบบ ต้องการค้นหารูปแบบที่สนใจอ่อนมา และต้องการให้ผลการค้นหานั้นผิดพลาดน้อยที่สุด. การเรียนรู้ของเครื่อง ต้องการที่จะทำภาระกิจที่ได้รับมอบหมาย ให้ได้สมรรถนะสูงสุด จากประสบการณ์ที่มี.

⁷ ในที่นี้ การแจกแจงเกาส์เชียน ถูกมองเป็นแบบจำลองที่ใช้ทำนายความน่าจะเป็นของ X .

⁸ เนื้อหาในหัวข้อนี้ได้รับอิทธิพลหลักจาก [39] และ [61, App. B]



รูปที่ 2.10 ความหนาแน่นความน่าจะเป็นของการแจกแจงเกาส์เชียน ที่ค่าพารามิเตอร์ต่าง ๆ ภาพในแถบบน แสดงผลของค่า μ ต่าง ๆ ตั้งแต่ -1 ถึง 2 โดย ค่า μ จะบออยู่ข้างบนแต่ละภาพ. ภาพข่ายสุด แสดงผลของค่า μ ต่าง ๆ ในภาพเดียวกันเพื่อการเปรียบเทียบได้ชัดเจน. ภาพในแถบล่าง จัดเรียงในลักษณะเดียวกัน แต่เป็น ผลของค่า σ ต่าง ๆ ตั้งแต่ 0.2 ถึง 2 . สังเกตว่า ความหนาแน่นความน่าจะเป็น มีค่ามากกว่าศูนย์เสมอ แต่อาจมีค่ามากกว่าหนึ่งได้ เช่นแสดงในภาพล่างที่สองจากข้างซ้าย.



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

การหาค่าดีที่สุด (optimization) คือ การหาค่าของปัจจัย (แทนด้วยตัวแปร) ที่มีผลให้เป้าหมาย (แทนด้วยฟังก์ชันของตัวแปร) มีค่าน้อยที่สุด (หรือมีค่ามากที่สุด ขึ้นกับเป้าหมายที่ต้องการ). ปัจจัยที่ต้องการเลือก เรียกว่า **ตัวแปรตัดสินใจ (decision variable)** และฟังก์ชันแทนเป้าหมาย ซึ่งประมาณความสัมพันธ์ระหว่างค่าของตัวแปรตัดสินใจและเป้าหมายที่ต้องการ เรียกว่า **ฟังก์ชันจุดประสงค์ (objective function)**.

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 การหาเส้นทางขับรถจากขอนแก่นไปร้อยเอ็ด ที่ใช้เวลาเดินทางน้อยที่สุด (ตัวแปรตัดสินใจเลือกเส้นทาง และฟังก์ชันจุดประสงค์ ประมาณเวลาเดินทาง) การหาทำเลตั้งเสาสัญญาณวิทยุ ที่ใช้บประมาณรวมน้อยที่สุด (ตัวแปรตัดสินใจเลือกตำแหน่งที่ตั้งเสาสัญญาณ และฟังก์ชันจุดประสงค์ ประมาณงบประมาณรวม) การหารูปแบบการจัดรูปร่างของโปรดีน ที่ใช้พลังงานน้อยที่สุด (ตัวแปรตัดสินใจเลือกรูปร่างของโปรดีน 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 สัญกรณ์อาจระบุเซตของค่าตัวแปรตัดสินใจที่ใช้ค้นหา เช่น $\min_{x \in \mathbb{R}} h(x)$ ซึ่งระบุว่า ค่าของตัวแปรตัดสินใจสามารถเป็นจำนวนจริงได ๆ หรือ $\min_{x \in \mathbb{R}^2} h(x)$ ระบุว่า ค่าของตัวแปรตัดสินใจเป็นเวกเตอร์ที่มีสองส่วนประกอบจำนวนจริง.

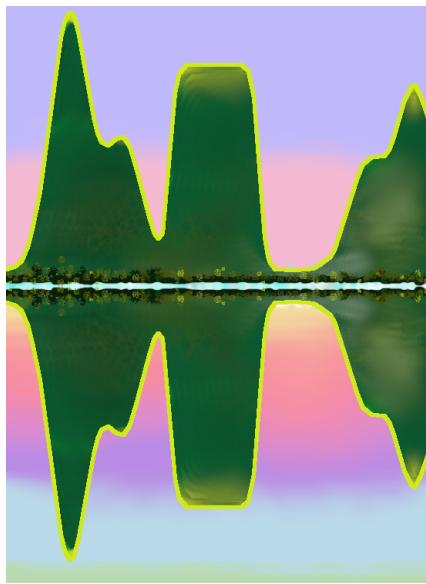
ปัญหาค่าน้อยที่สุดและปัญหาค่ามากที่สุด จริง ๆ แล้ว เป็นเสมือนเรื่องเดียวกันที่มองจากคนละมุม. ปัญหาค่าน้อยที่สุดและปัญหาค่ามากที่สุด สามารถแปลงไปมาระหว่างกันได้. นั่นคือ การหาตัวแปรตัดสินใจ x ที่ทำให้ฟังก์ชันจุดประสงค์ $h(x)$ มีค่ามากที่สุด จะเทียบเท่ากับการหาค่า x ที่ทำให้ $-h(x)$ มีค่าน้อยที่สุด. นั่นคือ $\max_x h(x) \equiv \min_x -h(x)$. รูปที่ 2.12 แสดงภาพเปรียบเทียบค่าฟังก์ชัน $h(x)$ และ $-h(x)$ ที่เปรียบเสมือนภพภูเขา และเจาของภพภูเขาที่สะท้อนน้ำ โดยค่าของฟังก์ชันจะพลิกกลับรอบ ๆ ค่าศูนย์ (ค่าวากเปลี่ยนเป็นลบ ค่าลบเปลี่ยนเป็นบวก ค่าศูนย์อยู่ที่เดิม ค่าวากมากอยู่สูงจะเปลี่ยนเป็นค่าลบมากอยู่ต่ำ เป็นต้น).

ดังนั้นเพื่อความสะดวก တารานี้จะอ้างถึง ปัญหาค่าน้อยที่สุด แทนปัญหาการหาค่าดีที่สุด โดยเฉพาะ เมื่อกิจกรรมวิธีการที่ใช้แก้ปัญหา ซึ่งเมื่อปัญหาทั้งสองแบบเทียบเท่ากัน วิธีต่าง ๆ ที่แก้ปัญหาค่าน้อยที่สุดได้ ก็สามารถใช้แก้ปัญหาค่ามากที่สุดได้เช่นกัน.

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 ว่า ปัญหาเป็นปัญหาค่าน้อยที่สุด (เช่น ฟังก์ชันค่าใช้จ่าย, ฟังก์ชันความสูญเสีย และ ฟังก์ชันพลังงาน) หรือปัญหาเป็นปัญหาค่ามากที่สุด (เช่น ฟังก์ชันผลประโยชน์ และ ฟังก์ชันคุณค่า). ชื่อเหล่านี้ มีการใช้อ้างถึงความต่างตามศาสตร์ ตามสาขาวิชา และตามงานประยุกต์ใช้งานด้านต่าง ๆ เช่น เครชชูศาสตร์ มักใช้ฟังก์ชันผลประโยชน์, ศาสตร์การวิจัยปฏิบัติการ (operation research) มักพบคำว่า ฟังก์ชันค่าใช้จ่าย. ศาสตร์การเรียนรู้ของเครื่อง มักเลือกใช้คำว่า ฟังก์ชันความสูญเสีย. ส่วนคำว่า ฟังก์ชันพลังงาน อาจพบเห็นได้บ้าง ในงานทางด้านการประมวลผลภาพ.

ผลลัพธ์จากการหาค่าดีที่สุด คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยที่สุด. ค่าที่ได้นี้ เรียกว่า ค่าทำให้น้อยที่สุด (minimizer) และนิยมใช้สัญลักษณ์เป็นตัวแปรตัดสินใจตามด้วย ตัวยกที่เป็นดาว เช่น x^* เพื่อระบุว่า กำลังกล่าวถึง ค่าทำให้น้อยที่สุดที่นำมาเสร็จแล้ว ไม่ใช่ x ที่เป็นตัวแปรตัดสินใจ ที่อาจใช้ค่าใด ๆ ก็ได้. หมายเหตุ ค่าทำให้น้อยที่สุด โดยทั่วไปจะไม่ใช่ค่าที่น้อยที่สุด. รูปที่ 2.13 แสดงแก่นอนแทนค่าของตัวแปรตัดสินใจ 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 .

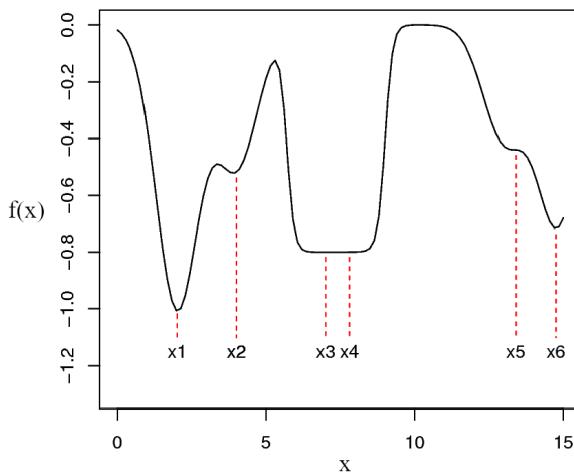
สัญกรณ์ $\min_x f(x)$ นี้ใช้เพื่อระบุเป้าหมายและตัวแปรที่เกี่ยวข้องเท่านั้น. หากต้องการระบุความสัมพันธ์ในสมการ อาจใช้สัญกรณ์ เช่น $v = \arg \min_x f(x)$ เพื่อระบุว่า ค่า $v = x^*$ ที่หาได้จากการแก้ปัญหา $\min_x f(x)$. หากต้องการระบุค่าฟังก์ชันจุดประสงค์ที่น้อยที่สุด อาจระบุด้วยสัญกรณ์ เช่น $f(x^*)$ หรือ สัญลักษณ์ เช่น f_{\min} เป็นต้น.



รูปที่ 2.12 ปัญหาค่ามากที่สุดกับ ปัญหาค่าน้อยที่สุด เป็นเรื่องเดียวกันที่มองจากคลุมมุน. การหาค่า x (เปรียบเหมือนตำแหน่งตามแนวอน) ของ $h(x)$ ที่มากที่สุด (เปรียบเหมือนยอดเขา) เป็นเรื่องเดียวกับ การหาค่า x ที่ทำให้ $-h(x)$ มีค่าน้อยที่สุด (ยอดเขาสูงเท่าไร เงาของยอดเขาจึงต่ำลงมากเท่านั้น แต่ตำแหน่งตามแนวอนเป็นที่เดิม).

ค่าทำให้น้อยที่สุดท้องถิ่น และค่าทำให้น้อยที่สุดทั่วหมด. จากรูปที่ 2.13 สังเกตว่า เม็ด 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 ที่ทำให้ฟังก์ชันจุดประยงค์มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประยงค์จากบริเวณรอบ ๆ. กล่าวอีกอย่างได้ว่า ค่าทำให้น้อยที่สุดท้องถิ่น คือ ค่าที่ทำให้ฟังก์ชันจุดประยงค์มีค่าน้อยที่สุดในท้องถิ่น (ไม่มีใครในละแวกที่น้อยเกิน). ค่า $x_1, x_2, x_3, x_4, x_5, x_6$ ต่างก็ทำให้ค่าฟังก์ชันจุดประยงค์น้อยกว่าหรือเท่ากับค่าจากบริเวณรอบ ๆ แต่ค่า x_1 นอกจากจะทำให้ $f(x_1)$ มีค่าน้อยกว่าค่าจากบริเวณรอบ ๆ แล้ว (ซึ่งทำให้ x_1 เป็นค่าทำให้น้อยที่สุดท้องถิ่น) ค่า $f(x_1)$ ยังน้อยที่สุดทุกที่ด้วย. ค่าตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประยงค์มีค่าน้อยกว่าหรือเท่ากับ ฟังก์ชันจุดประยงค์ของค่าตัวแปรทุกตัวที่เป็นไปได้ เรียกว่า ค่าทำให้น้อยที่สุดทั่วหมด (global minimizer). กล่าวอีกอย่างได้ว่า ค่าทำให้น้อยที่สุดทั่วหมด คือ ค่าที่ทำให้ฟังก์ชันจุดประยงค์มีค่าน้อยที่สุดทั่วหมดทุกที่ (ไม่มีใครในหล้าที่น้อยเกิน). ค่าทำให้น้อยที่สุดทั่วหมด จะเป็นค่าทำให้น้อยที่สุดท้องถิ่นด้วยเสมอ. แต่ค่าทำให้น้อยที่สุดท้องถิ่น อาจไม่ใช่ค่าทำให้น้อยที่สุดทั่วหมด. สถานการณ์ที่ การแก้ปัญหาค่าดีที่สุด แล้วได้ค่าทำให้น้อยที่สุดท้องถิ่น แต่ไม่ใช่ค่าทำให้น้อยที่สุดทั่วหมด มักถูกอ้างถึงว่าเป็น สถานการณ์ที่ดีที่สุดท้องถิ่น (local optimum).

สังเกตว่า บริเวณ x_3 และ x_4 จะเป็นที่สมิือนที่ราบ ซึ่งนอกจาก x_3 และ x_4 ค่าบริเวณนั้นก็จะให้ฟังก์ชันจุดประยงค์ที่เท่ากัน ค่า x บริเวณที่ราบนั้น ก็จะเรียกว่าเป็น ค่าทำให้น้อยที่สุดท้องถิ่นได้ทั้งหมด เพราะว่า รอบ ๆ ข้างไม่มีใครทำให้ฟังก์ชันจุดประยงค์น้อยเกินได้. ค่า x_5 ก็เป็นค่าทำให้น้อยที่สุดท้องถิ่น แต่เป็นลักษณะที่เรียกว่า จุดอานม้า (saddle point).



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.

การแก้ปัญหาด้วยวิธีลงเกรเดียนต์

ศาสตร์การหาค่าดีที่สุดนั้นกว้างขวาง และมีการประยุกต์ใช้ที่หลากหลาย. สำหรับการประยุกต์ใช้กับงานการรู้จำรูปแบบ และการเรียนรู้ของเครื่อง ลักษณะปัญหา มักจะถูกตีกรอบอกรอบมาให้ตัวแปรตัดสินใจ $v \in \mathbb{R}^n$ เมื่อ n เป็นจำนวนเต็มมีค่าตั้งแต่หนึ่งขึ้นไป และฟังก์ชันจุดประสงค์ g เป็นฟังก์ชันที่สามารถหาอนุพันธ์ได้ (*differentiable function*). การมีฟังก์ชันจุดประสงค์ที่สามารถหาอนุพันธ์ได้ ช่วยให้สามารถใช้ขั้นตอนวิธีแก้ปัญหาค่าน้อยที่สุด ที่มีประสิทธิภาพได้.

วิธีลงเกรเดียนต์ (gradient descent algorithm) เป็นขั้นตอนวิธี (*algorithm*) สำหรับปัญหาค่าน้อยที่สุด. วิธีลงเกรเดียนต์เป็นขั้นตอนวิธีที่เรียบง่าย และใช้ได้ผลดี โดยเฉพาะกับปัญหาขนาดไม่ใหญ่มาก. แนวคิดของวิธีลงเกรเดียนต์ คือ การใช้ค่าเกรเดียนต์ ช่วยในการหาค่าของตัวแปรตัดสินใจ โดย การเริ่มต้นด้วย ค่าของตัวแปรตัดสินใจ ค่าหนึ่ง และคำนวณค่าเกรเดียนต์ ณ จุดนั้นของกราฟ แล้วใช้ค่าเกรเดียนต์ที่ได้ บอกทิศทางในการปรับค่าของตัวแปรตัดสินใจ ว่าควรปรับเพิ่มหรือลด มากน้อยเท่าไร ดำเนินการปรับค่าตัวแปรตัดสินใจ และวนทำไปเรื่อย ๆ จนพบจุดที่เป็นค่าทำให้น้อยที่สุด.

เกรเดียนต์ ซึ่งเป็นอนุพันธ์ของฟังก์ชันจุดประสงค์ต่อตัวแปรตัดสินใจ จะบอกอัตราการเปลี่ยนค่าของฟังก์ชันจุดประสงค์ เมื่อตัวแปรตัดสินใจเพิ่มค่าขึ้น. ดังนั้นหากเกรเดียนต์เป็นบวกและมีค่ามาก นั่นหมายถึง ถ้าเพิ่มค่าตัวแปรตัดสินใจขึ้น แล้วฟังก์ชันจุดประสงค์จะมีค่าเพิ่มขึ้นมาก. หากค่าของเกรเดียนต์เป็นบวกแต่มีขนาดเล็ก การเพิ่มค่าตัวแปรตัดสินใจขึ้น จะไปเพิ่มค่าฟังก์ชันจุดประสงค์ขึ้นแต่ไม่มาก และหากเกรเดียนต์เป็นลบ เมื่อเพิ่มค่าตัวแปรตัดสินใจขึ้น ค่าฟังก์ชันจุดประสงค์จะลดลง. ดังนั้นเกรเดียนต์จึงสามารถใช้เป็นسمீனเนื่องจาก ที่บอกทิศทางที่จะปรับค่าตัวแปรตัดสินใจ เพื่อลดค่าฟังก์ชันจุดประสงค์ลงไปเรื่อย ๆ จนไปถึงค่าฟังก์ชันจุดประสงค์ที่น้อยที่สุดได้. สมการ 2.46 แสดงการ

คำนวณค่าตัวแปรตัดสินใจ ด้วยวิธีลงเกรเดียนต์

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

เมื่อ ตัวแปร $\boldsymbol{v}^{(k+1)}$ เป็นค่าใหม่ของตัวแปรตัดสินใจ (ที่ได้จากการคำนวณครั้งที่ $k + 1$) และ ตัวแปร $\boldsymbol{v}^{(k)}$ เป็นค่าเดิมของตัวแปรตัดสินใจ (ที่ได้จากการคำนวณครั้งที่ k) และ เกรเดียนต์ $\nabla g(\boldsymbol{v}^{(k)})$ คือ ค่าเกรเดียนต์ของฟังก์ชันจุดประสงค์ ที่ค่าเดิมของตัวแปรตัดสินใจ (ค่าตัวแปรตัดสินใจที่ได้จากการคำนวณครั้งที่ k) และค่าสเกลาร์ $\alpha > 0$ เรียกว่า ขนาดก้าว (step size) เป็นค่าที่ใช้ควบคุมอัตราเร็วในการปรับค่าของตัวแปรตัดสินใจ.

แนวคิดนี้ อุปมาเหมือน คนที่อยู่บ้านยอดเขาสูง ที่มองลงมาจนมองอะไรไม่เห็น และต้องการกลับบ้านที่อยู่พื้นล่าง. เปรียบฟังก์ชันจุดประสงค์ เป็นเหมือนระดับความสูงของพื้นที่ ณ จุดที่ยืนอยู่ และตัวแปรตัดสินใจเป็นเหมือนตำแหน่งเส้นรุ้งเส้นวาง (latitude-longitude location) ณ จุดที่ยืนอยู่ ตัวแปร $\boldsymbol{v}^{(k)}$ ก็เสมือนตำแหน่งที่ยืนอยู่ปัจจุบัน และเกรเดียนต์ $\nabla g(\boldsymbol{v}^{(k)})$ ก็เหมือนความชัน ณ ตำแหน่งที่ยืนปัจจุบัน ที่บอกว่าทิศทางไหนที่รู้สึกว่าพื้นขึ้นขึ้น. ถ้าต้องการกลับบ้าน หรือไปตำแหน่งที่ระดับความสูงที่ต่ำที่สุด วิธีคือ ขยับจากจุดที่ยืนปัจจุบันไปจุดใหม่ โดยขยับไปในทิศทางลง (ซึ่งคือ ทิศทางตรงข้ามกับทิศที่พื้นขึ้นขึ้น ได้แก่ ทิศ $-\nabla g(\boldsymbol{v}^{(k)})$) โดยต้องค่อย ๆ เดิน ค่อย ๆ ก้าวเล็ก ๆ เพราะถ้าก้าวยาวเกินไป อาจก้าวข้ามทางเดินลงเข้าแคบ ๆ ที่จะกลับบ้านได้ และขนาดก้าว α ก็เป็นค่าที่ใช้คุณไม่ให้ก้าวยาวเกินไป หลังจากก้าวไปแล้ว ตอนนี้ ตำแหน่งที่ยืนก็จะเปลี่ยนใหม่เป็น $\boldsymbol{v}^{(k+1)}$ และก็ขยับแบบนี้อีก ทำซ้ำเรื่อย ๆ จนกลับถึงบ้าน.

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

- เริ่มต้นด้วยการเลือกจุดเริ่มต้น สมมติเลือก $x^{(0)} = 6.5$ และเลือกใช้ค่าขนาดก้าว สมมติเลือก เป็น $\alpha = 0.5$. เกรเดียนต์สามารถวิเคราะห์การคำนวณได้

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

- 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

$$\begin{aligned}x^{(3)} &= 6.1202 - (0.5) \cdot \nabla f(6.1202) = 5.8009 \\x^{(4)} &= 5.8009 - (0.5) \cdot \nabla f(5.8009) = 5.3792\end{aligned}$$

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 ธรรมชาติ⁹ ของค่าทำให้น้อยที่สุด ซึ่งคือ ณ จุดค่าทำให้น้อยที่สุด เกรเดียนต์จะมีค่าเป็นศูนย์¹⁰ ดังนั้น ถึงแม้จะคำนวณต่อ ค่าของตัวแปรตัดสินใจก็จะยังคงติดอยู่ที่ค่าทำให้น้อยที่สุด.

การใช้งานวิธีลงเกรเดียนต์. วิธีลงเกรเดียนต์มีอภิธานพารามิเตอร์เป็นค่าขนาดก้าว. อภิธานพารามิเตอร์ (meta-parameter หรือ hyper-parameter) หมายถึงปัจจัยระดับสูงของวิธีคำนวณแต่ละวิธี ที่ผู้ใช้ต้องเลือกให้เหมาะสม. สำหรับวิธีลงเกรเดียนต์ ผู้ใช้ต้องเลือกค่าของขนาดก้าว. ถ้าเลือกใช้ขนาดก้าวที่มีค่าเล็กพอก วิธีลงเกรเดียนต์ จะรับประทานการลูปเข้าได้¹¹. การลูปเข้า (convergence) หมายถึง พฤติกรรมที่ดีของผลการคำนวณ สำหรับการคำนวณลักษณะการวนซ้ำขั้นตอน (iterative refinement computation) นั่นคือ ค่าของผลการคำนวณของแต่ละรอบคำนวณ มีการเปลี่ยนแปลงน้อยลง เมื่อรอบการคำนวณเพิ่มมากขึ้น อาจกล่าวได้ว่า การลูปเข้าคือพฤติกรรมที่ ผลลัพธ์จากการคำนวณเปลี่ยนแปลงค่าเข้าหา(ลูปเข้าหา)ค่าใดค่าหนึ่ง ในลักษณะคงที่ เมื่อรอบการคำนวณเพิ่มมากขึ้น.

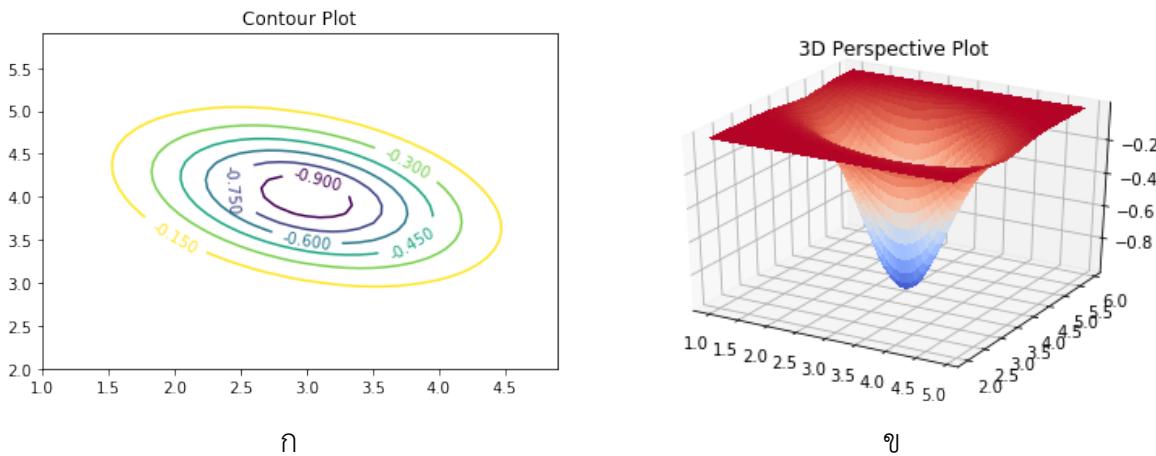
ขนาดก้าวที่ใหญ่เกินไป อาจทำให้การคำนวณล้มเหลวได้ แต่ขนาดก้าวที่เล็กเกินไป อาจทำให้ต้องทำการคำนวณหลายรอบมาก ๆ ซึ่งมีผลให้ใช้เวลาในการคำนวณนาน (ดูแบบฝึกหัด 2.22 ประกอบ). ในทางปฏิบัติ เพื่อให้ไม่เสียเวลาคำนวณมากเกินไป เงื่อนไขการจบการคำนวณ (terminating condition) มักถูกใช้ประกอบ. เงื่อนไขการจบที่นิยมใช้กับวิธีลงเกรเดียนต์ 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.

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

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

¹¹วิธีลงเกรเดียนต์ รับประทานการลูปเข้าสู่ค่าทำให้น้อยที่สุดทั้งถาวร เมื่อเลือกใช้ขนาดก้าวที่มีค่าเล็กพอก แต่ไม่ได้รับประทานว่า จะเจอค่าทำให้น้อยที่สุดทั่วหมด หรือไม่ได้รับประทานว่า จะไม่ไปติดอยู่จุดไหนมา เป็นต้น.



รูปที่ 2.14 (ซ้าย) ภาพคอนทัวร์ และ (ขวา) ภาพสามมิติของฟังก์ชัน $g(\mathbf{v}) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$

เมื่อไหความคลาดเคลื่อนยินยอม กำหนดค่าความคลาดเคลื่อนที่ยอมรับได้ ซึ่งอาจเลือกใช้เงื่อนไข ขนาดของเกรเดียนต์ $\epsilon = \|\nabla g^{(k+1)}\|$. แบบฝึกหัด 2.17 แสดงตัวอย่างโปรแกรมวิธีลงเกรเดียนต์ ที่ใช้เงื่อนไขการจบทุกๆ จำนวนรอบสูงสุด. แบบฝึกหัด 2.18 แสดงตัวอย่างโปรแกรมวิธีลงเกรเดียนต์ที่ใช้เงื่อนไขคลาดเคลื่อนยินยอม.

วิธีลงเกรเดียนต์ประยุกต์ใช้ได้ง่าย ต้องการแค่เกรเดียนต์ และค่าเริ่มต้น. ค่าเริ่มต้นก่อนการคำนวณของตัวแปรตัดสินใจ อาจเป็นค่าใดก็ได้¹² แต่โดยทั่วไป นิยมการกำหนดค่าเริ่มต้น (initialization) ให้กับตัวแปรตัดสินใจ ด้วยค่าที่สุ่มขึ้นมา. (ดูแบบฝึกหัด 2.24)

ปัญหาที่ตัวแปรตัดสินใจมีหลาย ๆ ตัว สามารถใช้วิธีลงเกรเดียนต์ได้ โดยการจัดหลาย ๆ ตัวแปรตัดสินใจ เข้ามาร่วมกันเป็นเวกเตอร์ตัวแปรตัดสินใจเวกเตอร์เดียว และวิธีลงเกรเดียนต์เตรียมการมาสำหรับกรณีนี้อยู่แล้ว (สมการ 2.46). สังเกตว่า ตัวแปรตัดสินใจเขียนเป็นเวกเตอร์ และอัตราการเปลี่ยน ก๊าซเกรเดียนต์ แต่การนำวิธีลงเกรเดียนต์ ไปเขียนโปรแกรมเพื่อทำงานกับเวกเตอร์ จะต้องระวังเรื่อง ของตัวแปรเป็นพิเศษ. (ดูแบบฝึกหัด 2.21 ประกอบ) นอกจากนั้น เพื่อช่วยให้สามารถตรวจสอบความ ก้าวหน้า ในการหาค่าทำให้น้อยที่สุดได้อย่างมีประสิทธิภาพ ควรตรวจสอบค่าฟังก์ชันจุดประสงค์ทุก ๆ รอบคำนวณ. ตัวอย่างต่อไปนี้ แสดงการคำนวณของวิธีลงเกรเดียนต์ เมื่อตัวแปรตัดสินใจมีสองค่า.

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

¹²วิธีลงเกรเดียนต์ทบทวนต่อค่าเริ่มต้นต่าง ๆ ในแต่ละคราว (ถ้าปัญหาไม่ยากเกินไป) วิธีลงเกรเดียนต์จะสามารถหาค่าทำให้น้อยที่สุดได้ แม้เลือกค่าเริ่มต้นต่างกัน แต่การเลือกค่าเริ่มต้นที่ดี จะช่วยให้วิธีลงเกรเดียนต์ทำงานได้เร็วขึ้น และสำหรับหลาย ๆ กรณี ค่าเริ่มต้นต่างกันอาจนำไปสู่ค่าทำให้น้อยที่สุดท่องถี่นคนละตัว (กรณีปัญหาหลายภาวะ multi-modal problem ดูแบบฝึกหัด 2.23 เพิ่มเติม) หรือสำหรับบางกรณี ค่าเริ่มต้นบางค่า อาจทำให้วิธีลงเกรเดียนต์ไม่สามารถทำงานได้เลย. ดูแบบฝึกหัด 2.23 ประกอบ.

- เลือกอภิมานพารามิเตอร์ ได้แก่ ค่าขนาดก้าว $\alpha = 0.01$
- เตรียมฟังก์ชันคำนวนเกรเดียนต์

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

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

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

- ปรับค่าตัวแปรตัดสินใจ ตามสมการ 2.46

การคำนวนครั้งแรก ($k = 1$)

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

การคำนวนที่สอง ($k = 2$)

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

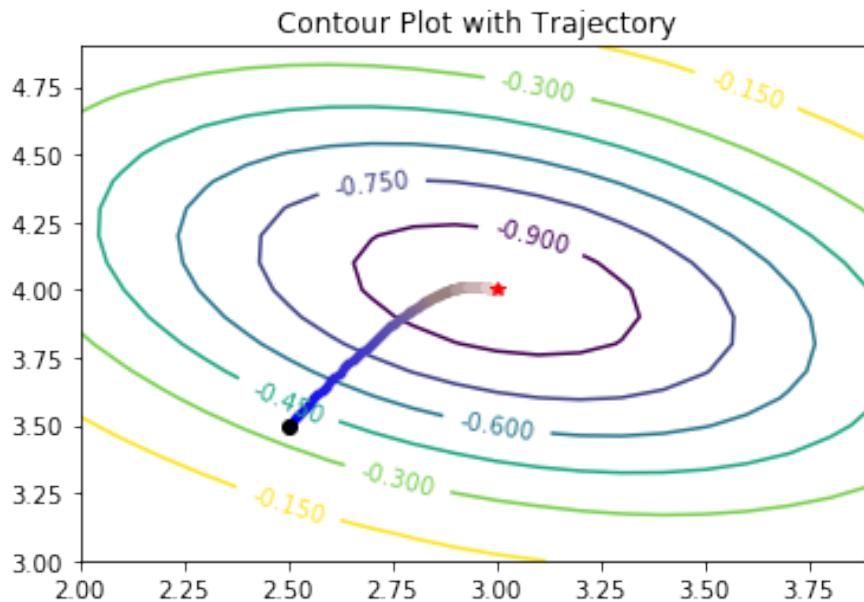
การคำนวนที่สาม ($k = 3$)

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

- และเมื่อดำเนินการคำนวนต่อไป (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 จนถึงแทบไม่ปรับเลยในรอบคำนวนหลัง ๆ นี้คือ \mathbf{v} ลู่เข้าหา $[3, 4]^T$. รูปที่ 2.15 แสดงการปรับค่าตัวแปรตัดสินใจ ในรูปเส้นทางบนภาพคอนทัวร์¹³ ของฟังก์ชันจุดประสงค์. รูปที่ 2.16 แสดง ความก้าวหน้า (progress) ของการดำเนินการหาค่าทำให้น้อยที่สุด.

#

¹³รูปเส้นทางบนภาพคอนทัวร์ เช่นนี้ สามารถแสดงได้เฉพาะกรณีที่ตัวแปรตัดสินใจ $\mathbf{v} \in \mathbb{R}^2$ หากตัวแปรตัดสินใจอยู่ในมิติปริภูมิที่ใหญ่ขึ้น การนำเสนอตัวยภาพจะทำได้ยากมาก.

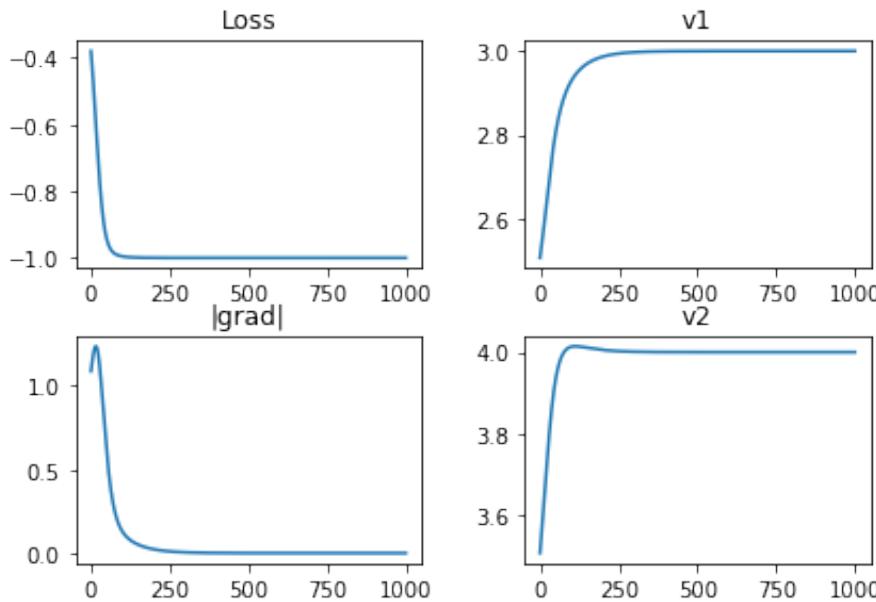


รูปที่ 2.15 เส้นทางการหาค่าทำให้น้อยที่สุด จุดวงกลมสีดำแสดงจุดเริ่มต้น (ตำแหน่ง $[2.5, 3.5]^T$) และจุดดาวสีแดงแสดงจุดสุดท้าย (ตำแหน่ง $[3, 4]^T$). เส้นทางระหว่างจุดทั้งสอง คือค่าต่าง ๆ ที่ตัวแปรตัดสินใจถูกปรับตั้งแต่รอบคำนวณแรก ๆ ไปจนถึงรอบคำนวณท้าย ๆ.

จากรูปที่ 2.16 ค่าของตัวแปรตัดลินใจ v ทั้งสองค่า จะปรับเข้าหา $[3, 4]^T$ นั่นคือคำตอบลู๊เข้า. เมื่อถูกที่จุดลู๊เข้า ดังที่ได้อธิบายมาแล้ว ขนาดของเกรเดียนต์จะเป็นศูนย์ที่จุดดีที่สุด (*optimal point*) ซึ่งเป็นจุดที่ตัวแปรตัดลินใจปรับค่าไปอยู่ที่ค่าทำให้น้อยที่สุด. ขนาดของเกรเดียนต์จึงนิยมใช้เป็นเงื่อนไขในการจบโปรแกรม เพื่อไม่ต้องคำนวนมากรอบเกินไป เช่นกรณีนี้ จะเห็นว่าทำการคำนวนแค่ราว ๆ 500 รอบก็เห็นการลู๊เข้าชัดเจนแล้ว. เมื่อถูกที่ฟังก์ชันจุดประสงค์ (ภาพ Loss บนซ้าย) ค่าฟังก์ชันจุดประสงค์จะน้อยลงเรื่อย ๆ ถ้าเลือกค่าขนาดก้าวไม่ใหญ่เกินไป. พฤติกรรมการทำงานของวิธีลงเกรเดียนต์จะมีลักษณะเช่นนี้ คือ ค่าฟังก์ชันจุดประสงค์จะลดลง เมื่อรอบคำนวนเพิ่มขึ้น. การดูค่าฟังก์ชันจุดประสงค์ต่อรอบคำนวนเช่นนี้ สะดวก เพราะสามารถตรวจสอบได้ ไม่ว่าตัวแปรตัดสินใจจะมีจำนวนเท่าใด และถูกใช้เป็นแนวทางปฏิบัติ เพื่อติดตามความก้าวหน้า และวิเคราะห์การทำงานของการแก้ปัญหาค่าน้อยที่สุด ว่าดำเนินการได้ดีมากน้อยเพียงใด.

วิธีลงเกรเดียนต์:

“ไม่ว่าเราเริ่มต้นอยู่ที่ไหน ถ้าเรายืบไปทิศทางที่ดีขึ้นเรื่อย ๆ
เราจะไปถึงจุดที่ดีที่สุดได้ เพียงต้องทำเรื่อย ๆ และไม่รีบเกินไป”



รูปที่ 2.16 ความก้าวหน้าในการหาค่าทำให้น้อยที่สุด. ภาพบนซ้าย (Loss) แสดงค่าฟังก์ชันจุดประสงค์ต่อรอบการคำนวณ. ภาพล่างซ้าย ($|grad|$) แสดงขนาดเกรเดียนต์ต่อรอบการคำนวณ. ขนาดเกรเดียนต์นิยมใช้เป็นเงื่อนไขการจบเพื่อลดเวลาคำนวณลง. ภาพขวาแสดงค่าตัวแปรตัดสินใจต่อรอบการคำนวณ ภาพบนแสดง v_1 และภาพล่างแสดง v_2 .

การหาค่าดีที่สุดแบบมีข้อจำกัด

การหาค่าดีที่สุดที่ได้อภิปรายมา ค่าของตัวแปรตัดสินใจเป็นค่าจำนวนจริงใด ๆ ซึ่งเขียนเป็นสัญกรณ์ทั่วๆ ไป $\min_{\mathbf{v}} g(\mathbf{v})$ โดย $\mathbf{v} \in \mathbb{R}^n$ เมื่อ n เป็นจำนวนเต็มที่มีค่าตั้งแต่หนึ่งเป็นตันไป การหาค่าดีที่สุดแบบนี้ไม่ได้มีเงื่อนไขจำกัดค่าของตัวแปรตัดสินใจ (นอกจากเป็นจำนวนจริง) และการหาค่าดีที่สุดแบบนี้ จะเรียกว่า **การหาค่าดีที่สุดแบบไม่มีข้อจำกัด (unconstrained optimization)**.

การหาค่าดีที่สุด สำหรับกรณีที่มีข้อจำกัดของค่าของตัวแปรตัดสินใจ จะเรียกว่า **การหาค่าดีที่สุดแบบมีข้อจำกัด (constrained optimization)** และจะใช้สัญกรณ์ เพื่อระบุข้อจำกัดอย่างชัดเจน เช่น

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

เมื่อ $\mathbf{v} \in \mathbb{R}^n$ เป็นตัวแปรตัดสินใจ ฟังก์ชัน $g : \mathbb{R}^n \rightarrow \mathbb{R}$ เป็นฟังก์ชันจุดประสงค์ และเซต Ω เป็นเซตย่อของ \mathbb{R}^n ที่ระบุค่าของตัวแปรตัดสินใจในช่วงที่สนใจ หรือกลุ่มค่าที่ยอมรับได้. ค่าคำตอบของ \mathbf{v} ที่ได้จะต้องอยู่ในเซตย้อนนี้. เซต Ω นี้ เรียกว่า **เขตข้อจำกัด (constraint set)** หรือ **เซตที่เป็นไปได้ feasible set**. กรณีที่ $\Omega = \mathbb{R}^n$ ปัญหานี้จะเป็น การหาค่าดีที่สุดแบบไม่มีข้อจำกัด. สัญกรณ์ 2.47 อาจเขียนย่อเป็น $\min_{\mathbf{v}} g(\mathbf{v}) \text{ s.t. } \mathbf{v} \in \Omega$.

จากตัวอย่าง การหาอุณหภูมิบ่อมทุเรียน ถ้ากำหนดว่า อุณหภูมิบ่อมต้องอยู่ในช่วง 30 ถึง 80 องศาเซลเซียส (เพราะว่าเตาบ่อมที่มี ทำอุณหภูมิได้ในช่วงแค่นั้น) ปัญหานี้จะเป็นการหาค่าดีที่สุดแบบมีข้อ

จำกัด และอาจเขียนเป็น สัญกรณ์ \max_t sugar(t) s.t. $30 \leq t \leq 80$ เมื่อ t แทนอุณหภูมิบ่ในหน่วยองศาเซลเซียส พังก์ชัน sugar ประเมินระดับน้ำตาลที่ได้จากการบ่มด้วยอุณหภูมิ t และข้อจำกัด $30 \leq t \leq 80$ ระบุค่าของ t ที่ให้ได้ว่า อุ่นในช่วง 30 ถึง 80. (ปัญหาค่าน้ำกที่สุดนี้ เทียบเท่าปัญหาค่าน้อยที่สุด $\min_t -\text{sugar}(t)$ s.t. $30 \leq t \leq 80$)

ค่าทำให้น้อยที่สุดท้องถิ่น และค่าทำให้มากที่สุดทั่วหมด. ในกรณีที่มีข้อจำกัด ค่าของตัวแปรตัดสินใจพิจารณาจากค่าที่เป็นไปได้ (feasible) เท่านั้น. ค่าทำให้น้อยที่สุดท้องถิ่น คือ ค่าของตัวแปรตัดสินใจ ที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประสงค์ของค่าที่เป็นไปได้บริเวณรอบ ๆ. นั่นคือ ถ้ากำหนดให้ $g : \mathbb{R}^n \rightarrow \mathbb{R}$ เป็นฟังก์ชันค่าจริง ที่นิยามสำหรับเขต $\Omega \subset \mathbb{R}^n$ และ จุด $\mathbf{v}^* \in \Omega$ เป็นค่าทำให้น้อยที่สุดท้องถิ่น ของฟังก์ชัน g บนเขต Ω 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 ที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยกว่าหรือเท่ากับค่าฟังก์ชันจุดประสงค์ของค่าที่เป็นไปได้อีน ๆ ทุกค่า. นั่นคือ จุด $\mathbf{v}^* \in \Omega$ จะเป็นค่าทำให้น้อยที่สุดทั่วหมดของฟังก์ชัน g บนเขต Ω 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 หรือยากที่จะคำนวณ ในหลาย ๆ สถานการณ์. งานการเรียนรู้ของเครื่อง มักนิยมใช้แนวทางการลงโทษมากกว่า.

แนวทางการลงโทษ (penalty approach) จะใช้การลงโทษ เมื่อค่าตัวแปรตัดสินใจไม่อุ่นในเขตข้อจำกัด 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).

พารามิเตอร์ลงโทษ บอยครั้งมักถูกเรียกว่า ลากรานจ์พารามิเตอร์ (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. $x \geq 0$. รูปที่ 2.17 แสดงฟังก์ชันจุดประสงค์ และแสดงส่วนที่ไม่ผ่านเส้นในด้วยแรเงาสีเทา. รูปแสดงให้เห็นว่า จุดที่ทำให้ค่าฟังก์ชันจุดประสงค์ น้อยที่สุด มีค่าเป็นลบ แต่จุดนี้จะเมิดข้อจำกัด 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 วิธีลงเกรเดียนต์ ได้) และฟังก์ชัน P ลงโทษค่าที่จะเมิดข้อจำกัดตามปริมาณที่จะเมิด. รูปที่ 2.18 แสดงค่าของฟังก์ชันลงโทษในสมการ 2.50 และเมื่อนำฟังก์ชันลงโทษนี้ ไปใช้ร่วมกับฟังก์ชันจุดประสงค์ ฟังก์ชันที่ปรับใหม่ จะเปลี่ยนพฤติกรรมไป ดังแสดงในรูปที่ 2.19.

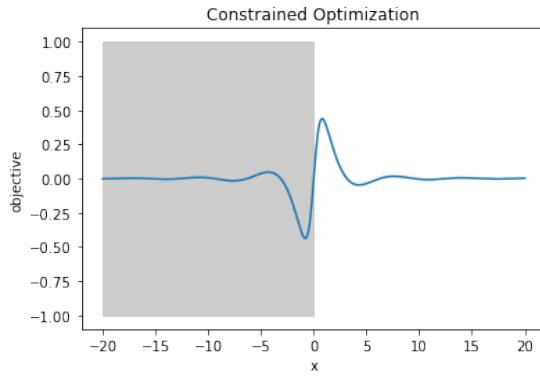
วิธีใช้การลงโทษนี้ ก็คือ การเปลี่ยนจากปัญหาแบบมีข้อจำกัด กลับมาเป็นปัญหาแบบไม่มีข้อจำกัด โดย การดัดแปลงฟังก์ชันจุดประสงค์ใหม่ เพื่อให้คำตอบอยู่ในช่วงค่าที่เป็นไปได้. ดังนั้น ขั้นตอนการหาคำตอบต่อไป ก็สามารถดำเนินการได้ โดยใช้วิธีแก้ปัญหา แบบเดียวกับปัญหาแบบไม่มีข้อจำกัด.

รูปที่ 2.19 แสดงให้เห็นว่า เมื่อ ค่าของลากกรานจ์พารามิเตอร์ใหญ่มากพอ ค่าหน่วยที่สุดของฟังก์ชันจุดประสงค์ใหม่ จะถูกบังคับให้อยู่ในช่วงค่าที่เป็นไปได้. กลไกการทำงานของการลงโทษ คือ การลงโทษจะทำให้ คำตอบของปัญหาค่าน้อยที่สุดใหม่ ที่รวมการลงโทษเข้าไป จะเป็น¹⁴ คำตอบของปัญหาค่าน้อยที่สุดแบบมีข้อจำกัดเดิม เมื่อลากกรานจ์พารามิเตอร์มีค่าใหญ่มากพอ.

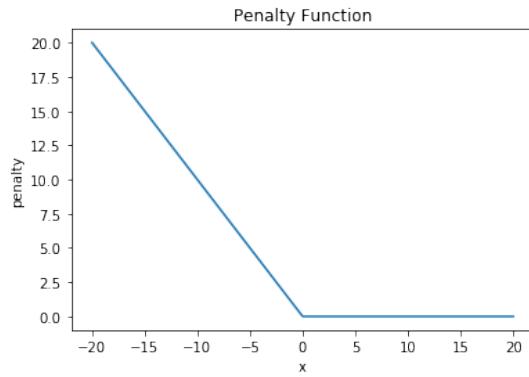
ในทางปฏิบัติ การแก้ปัญหาด้วยวิธีลงโทษ จะเลือกค่าของลากกรานจ์พารามิเตอร์มาค่านึง และดำเนินการแก้ปัญหา แล้วตรวจสอบค่าคำตอบที่ได้ ว่าอยู่ภายใต้ข้อจำกัดหรือไม่ ถ้าคำตอบที่ได้ อยู่ภายใต้ข้อจำกัด นี่ก็คือ คำตอบของปัญหาที่ใช้ได้. แต่ถ้าคำตอบที่ได้ ไม่อยู่ภายใต้ข้อจำกัด จะต้องเพิ่มค่าของลากกรานจ์พารามิเตอร์ขึ้น และดำเนินการแก้ปัญหาอีก และทำเช่นนี้คือ ค่อย ๆ เพิ่มค่าลากกรานจ์พารามิเตอร์ขึ้น จนกว่าจะได้คำตอบที่อยู่ภายใต้ข้อจำกัด.

เกร็ดความรู้ สติปัญญาของลิง รายการโนนว่า ออกอากาศสารคดีสติปัญญาของลิงไม่มีหาง เรื่อง “Ape Genius” ทางสถานีโทรทัศน์พีบีเอส ของสหรัฐอเมริกา ในปี ค.ศ. 2008. รายการนำเสนอองค์การศึกษาสติปัญญาของลิงชิมแปนซีและลิงโบโนโน่หลาย ๆ งาน (ลิงชิมแปนซีและลิงโบโน่ มีพัฒนรุ่นธรรมต่างจากมนุษย์แค่ประมาณ 1.2%. มนุษย์แต่ละคนมีพัฒนรุ่นธรรมแตกต่างกันประมาณ 0.1%) รายการ พยายามจะตอบคำถาม ที่ถามว่า ลักษณะของสติปัญญาแห่งมุ่งได้ที่ต่างกัน และทำให้ชิมแปนซีและโบโน่ไม่สามารถพัฒนาขึ้นมาสร้างอารยธรรม เช่นเดียวกับที่มนุษย์ทำได้.

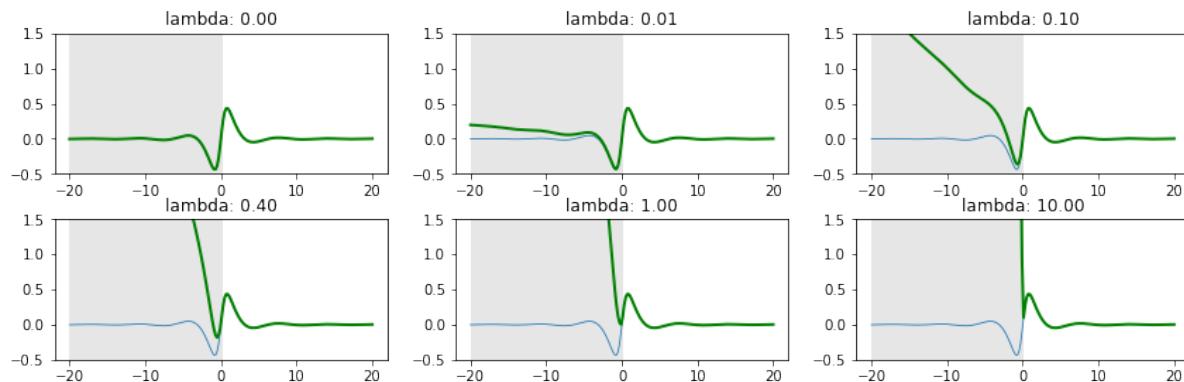
¹⁴กำหนดให้ $\mathbf{v}^{(k)}$ เป็นคำตอบของปัญหาที่มีการลงโทษด้วยค่า λ_k และฟังก์ชันจุดประสงค์เดิม g เป็นฟังก์ชันต่อเนื่อง และ $\lambda_k \rightarrow \infty$ เมื่อ $k \rightarrow \infty$ แล้ว ลิมิต (*limit*) ของลำดับที่ถูกเข้า $\{\mathbf{v}^{(k)}\}$ จะเป็นคำตอบของปัญหาแบบมีข้อจำกัดเดิม. ดู [39] สำหรับรายละเอียด.



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



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



รูปที่ 2.19 ตัวอย่างฟังก์ชันถูกุ้ยเลี้ยงที่รวมการลงโทษเข้าไป (เส้นหนาสีเขียว) ที่ใช้ลักษณะพารามิเตอร์ค่าต่าง ๆ (ตั้งแต่ 0 ถึง 10 ตามที่ระบุในแต่ละภาพ) เปรียบเทียบกับฟังก์ชันเดิม (เส้นบางสีฟ้า). จากรูป สำหรับตัวอย่างนี้ ค่าลักษณะพารามิเตอร์ 1.00 (ภาพกลาง แก้วล่าง) เพียงพอที่จะเปลี่ยนค่าที่ทำให้น้อยที่สุดไปสู่จุดที่อยู่ภายนอกข้อจำกัด.

ความสามารถในการสร้างและใช้เครื่องมือ. มีหลักฐานชัดเจนว่า ลิงชิมเป็นซึ่มีการสร้างและใช้เครื่องมือ เช่น เจน กูด ดอล พบลิงชิมเป็นซีในแทนซาเนีย ใช้กิ่งไม้ในการล้อมดามากิน และ จิต พริตซ์ พบลิงชิมเป็นซีในปาโภกอลี ในประเทศไทย เชนกัล ที่สร้างหอกจากกิ่งไม้ และใช้เป็นเครื่องมือในการล่าหาอาหาร

ความสามารถในการทำงานร่วมกัน. มีหลักฐานหลายอย่างของพฤติกรรมการอุกล่าร่วมกันของลิงชิมเป็นซี และสถาบันวิจัยลิงใหญ่เมืองทาง (Great Ape Research Institute) ของญี่ปุ่น พบว่า ลิงชิมเป็นซีมีความสามารถในการทำงานร่วมกัน มีความสามารถในการขอความช่วยเหลือ และก็สามารถให้ความช่วยเหลือมนุษย์ได้เวลาที่ถูกร้องขอ

ความสามารถในการแก้ปัญหา. การศึกษาหนึ่งทดลอง โดยใส่เมล็ดถั่วไว้ในหลอดยาวย ที่ลิงไม่สามารถจะล้วงเข้าไป หยิบได้ และตัวหลอด ก็ยังติดกับกรงแน่น จนลิงไม่สามารถขยับได้. ลิงใช้เวลาพักหนึ่ง ก่อนจะพบวิธีแก้ปัญหา. มันไปปีบ่อน้ำในกรง อมน้ำแล้วมาพ่นใส่หลอด แล้วอาหารก็หลอยขึ้นบนน้ำ มันเติมน้ำเข้าไป จนอาหารลอยอยู่ในระดับที่เอื้อมถึงได้ สิ่งนี้ แสดงถึงความสามารถในการแก้ปัญหาของลิง.

ความสามารถในการเลียนแบบ. ทีมของนักจิตวิทยาแอนดรู วิทเทนต้องการทดสอบความสามารถในการเลียนแบบของลิง. ทีมสร้างเครื่องกลไกที่ลิงจะต้องทำสองขั้นตอน ได้แก่ หมุนจานให้พอดีซ่อง และโยกคันโยก เพื่อจะได้กินอาหาร และนำเครื่องไปทดสอบกับตัวลิง. ลิงไม่สามารถจะหาวิธีทำนี้ได้เอง. แต่ทีมงานค่อย ๆ สอนลิงขั้นมาตัวหนึ่ง จากนั้นลองให้ลิงตัวอื่นดูลิงตัวนี้ทำงาน แล้วสังเกตว่าลิงตัวอื่น ๆ ก็สามารถเลียนแบบ เพื่อทำงานสองขั้นตอนนี้ได้อย่างง่ายดาย.

ความสามารถทางตัวเลข. เทหழูโร มัตซูชิวะแห่งมหาวิทยาลัยเกียวโต นำเสนอผลการทดสอบลิงชิมเป็นซีชื่อว่า ที่แสดงความสามารถทางตัวเลข ในการเข้าใจความหมายของตัวเลขของบิก และยังสามารถรู้ลำดับของตัวเลขได้

ความสามารถทางภาษาและการสื่อสาร. ลิงโบโนะชื่อคานซี เรียนรู้ภาษาอังกฤษได้เอง โดยไม่ได้ถูกสอนโดยตรง และชู ชาเวจ-รัมบava แสดงให้เห็นว่าคานซีเข้าใจภาษาอังกฤษและสามารถทำความสั่งได้อย่างถูกต้อง

ความสามารถที่ลิงไม่มี ความสามารถที่กล่าวมาข้างต้น เป็นความสามารถที่พบหลักฐานในลิงชิมเป็นซีหรือโบโนะ. แต่ความสามารถที่ลิงชิมเป็นซีหรือโบโนะไม่มี และเป็นปัจจัยสำคัญที่ทำให้ลิงไม่สามารถพัฒนาอารยธรรมขึ้นมาได้ เช่นว่า คือความสามารถด้านอารมณ์. ลิงชิมเป็นซีมีปัญหาที่เห็นได้ชัดเจน คือปัญหาด้านอารมณ์ ทั้งการแก่งแย่งชิงตีกัน ความรุนแรง และที่สำคัญคือ การควบคุมอารมณ์ตัวเอง.

ความสามารถในการควบคุมตัวเอง. การทดลองของแซลลี่ บอยเซนมหาวิทยาลัยรัฐโอไฮโอ แสดงให้เห็นโดยให้ลิงเลือกงานอาหารระหว่างงาน ส่องงาน ที่มีขั้นตอนอยู่ไม่เท่ากัน แต่งานที่ลิงเอื้อมมือไปหา จะเป็นงานที่จะนำไปให้กับลิงอีกด้วย. ถ้าเป็นขนมที่อยู่บนงาน ลิงไม่สามารถจะอดใจและเอื้อมไปที่งานที่น้อยกว่าได้ มันจะเอื้อมไปที่งานที่มันเห็นอาหารมากกว่า ตลอด. แต่พอแซลลี่ บอยเซนเปลี่ยนจากการที่เอาขนมวางไว้ในงานให้เห็น กลับใช้ตัวเลขซึ่งลิงเข้าใจความหมาย วางไว้แทน. ลิงสามารถเรียนรู้ที่จะเอื้อมไปที่งานที่ตัวเลขน้อยกว่าได้. การทดลองนี้แสดงให้เห็นว่า ลิงชิมเป็นซีมีปัญหาในการควบคุมอารมณ์ของตัวมันเอง. เวลาที่มันเห็นอาหารอยู่ มันไม่สามารถควบคุมตัวเพื่อเลือกทางเลือกที่ดีกว่าได้ แต่พอตัดแรงกระตุนทางอารมณ์ออก (ใช้ตัวเลขวางแทนอาหารจริง) มันสามารถเลือกทางเลือกที่ดีกว่าได้.

นอกจากการขาดความสามารถในการควบคุมตนเองแล้ว ปัจจัยสำคัญอีกสองอย่างที่รายการสรุปว่า เป็นอุปสรรคที่ทำให้สติปัญญาของลิงไม่อาจสะสม สร้างเสริมไปสู่การพัฒนาในระดับเดียวกับมนุษย์ได้ ก็คือ ความสามารถในการเรียนรู้โดยรับการถ่ายทอดจากคนอื่น (หรือลิงตัวอื่น) และความสามารถในการสอน. แม้เด็กอาจไม่ได้แสดงความสามารถในการแก้ปัญหาได้เท่ากับลิงชิมเป็นซี แต่เด็ก ๆ แสดงความสามารถที่สามารถเรียนรู้จากสิ่งที่ถูกสอนได้ดีกว่า สุนัขเองก็ยังมีความสามารถในการเรียนจากการสอนของมนุษย์ได้ดีกว่าลิง.

นอกจากความสามารถในการเรียนจากการถ่ายทอด ความเต็มใจที่ถ่ายทอด หรือความเต็มใจที่จะสอน ก็เป็นส่วนประกอบสำคัญที่ทำให้การถ่ายทอดความรู้เกิดขึ้นได้ และลิงชิมเป็นซีไม่ทั้งสององค์ประกอบนี้. อารยธรรมของมนุษย์สร้างโดยการส่งผ่านความรู้และปัญญาจากรุ่นสู่รุ่น. แม้ลิงสามารถเรียนรู้จากลิงตัวอื่นได้โดยการเลียนแบบ แต่การเรียนรู้โดยการ

เลียนแบบนั้นมักจะซ้ำและตื้นเขิน. บางครั้งยังอาจมีการสูญหายไป จากการเปลี่ยนรุ่นของลิงอีกด้วย. ลิงรุ่นเก่าตายไป ลิงรุ่นใหม่อาจไม่ได้เรียนรู้สิ่งที่ลิงรุ่นเก่ารู้แล้ว หลาย ๆ อย่างที่ลิงรุ่นเก่ารู้แล้ว เช่นวิธีการใช้เครื่องมือ อาจหายไปจากลิงรุ่นใหม่ และอาจใช้เวลาอีกนานกว่าที่ลิงรุ่นใหม่จะพบรู้ใช้เครื่องมืออีกรึเปล่า.

การควบคุมตัวเอง การเรียนรู้จากการถ่ายทอด และความเต็มใจที่จะสอน เป็นคุณสมบัติที่แยกมนุษย์ออกจากลิง และเป็นพื้นฐานอารยธรรมของมนุษย์.

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

--Robina Courtin

“... แบบเดียวกับที่นักท่องมีสองปีก
เราต้องมีทั้งปัญญาและเมตตา”

—โรบิน่า เคอร์ทิน

2.4 อภิธานศัพท์

มิติ (dimension): มุ่งมอง หรือหมายถึง มิติปริภูมิค่า ที่เป็นจำนวนตัวเลขที่ใช้ เพื่อรับ��ดีาแห่งของค่าที่สนใจในปริภูมิค่า เช่น เวกเตอร์ $\mathbf{v} \in \mathbb{R}^{12}$ จะอยู่ใน 12 มิติปริภูมิค่า หรือหมายถึงลำดับชั้น เช่น เทนเซอร์ $\mathbf{V} \in \mathbb{R}^{2 \times 3 \times 16}$ จะมี 3 ลำดับชั้น.

เทนเซอร์ (tensor): โครงสร้างลำดับชั้นของตัวเลข.

นอร์ม (norm): ขนาดของเวกเตอร์.

ภาพฉายเชิงตั้งฉาก (orthogonal projection): การแปลงค่าที่สนใจ ลงไปบนทิศทางใหม่ โดยการคูณกับเวกเตอร์หนึ่งของทิศทางใหม่.

การแยกส่วนประกอบเชิงตั้งฉาก (orthogonal decomposition): การแยกส่วนประกอบของค่าที่สนใจออกเป็นส่วน ที่แต่ละส่วนอยู่ในปริภูมิย่อยที่ตั้งฉากกัน โดยปริภูมิย่อยทั้งหมดที่ได้จากการแยกส่วนประกอบ จะແພ່ທົ່ວປະກຸມໄດ້.

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

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

ความน่าจะเป็น (probability): ค่าประเมินโอกาสที่จะเกิดเหตุการณ์ที่สนใจ.

ผลลัพธ์ (outcome): ผลเฉลย หรือความจริงที่เกิดขึ้น สิ่งที่เกิดขึ้น หรือสิ่งที่ประจักษ์ภายหลัง.

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): ตัวแปรสุ่มที่ฟังก์ชันการแจกแจงของมันสามารถเขียนได้ในรูป $F(x) = \int_{-\infty}^x f(u)du$ เมื่อ $f : \mathbb{R} \rightarrow [0, \infty)$ เป็นฟังก์ชันความหนาแน่น.

ฟังก์ชันความน่าจะเป็น (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): ความน่าจะเป็นของตัวแปรสุ่มที่ต้องการอนุมาน หลังจากรู้ข้อมูลประกอบแล้ว.

ฟังก์ชันควรจะเป็น (likelihood function): ฟังก์ชันของค่าของเงื่อนไข ของความน่าจะเป็นแบบมีเงื่อนไข.

การสลายปัจจัย (marginalization): การใช้กฎผลรวม เพื่อลดจำนวนตัวแปรสุ่มลงจากการความน่าจะเป็นที่พิจารณา.

การหาค่าดีที่สุด (optimization): การหาค่าของตัวแปรตัดสินใจ เพื่อให้ได้ค่าเป้าหมายดีที่สุด.

ตัวแปรตัดสินใจ (decision variable): ตัวแปรที่ต้องการหาค่าในการหาค่าดีที่สุด.

ฟังก์ชันจุดประสงค์ (objective function): ฟังก์ชันที่ใช้ประมาณค่าเป้าหมายในการหาค่าดีที่สุด หรือบางครั้งอาจเรียกว่า ฟังก์ชันสูญเสีย.

ปัญหาค่าน้อยที่สุด (minimization problem): การหาค่าดีที่สุดที่ต้องการให้ฟังก์ชันจุดประสงค์มีค่าน้อยที่สุด.

ค่าทำให้น้อยที่สุด (minimizer): ค่าของตัวแปรตัดสินใจ ที่ทำให้ค่าฟังก์ชันจุดประสงค์น้อยที่สุด.

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

ວິທີລົງເກຣດີຍົນຕໍ່ (gradient descent algorithm): ຊັ້ນຕອນວິທີນີ້ໃນກາຮແກ້ປ່ມຫາຄ່ານ້ອຍທີ່ສຸດ ໂດຍ
ກາຮຄໍານວນແບບວນໜ້າ ແລະ ອາສີຍຄ່າເກຣດີຍົນຕໍ່ຂອງຝຶກໜັນຈຸດປະສົງເກີດໄດ້ກັບຕົວແປຣຕັດສິນ
ໄຈ.

ຂາດກ້າວ (step size): ຄ່າສເກລ່າຮ່າທີ່ໃຊ້ຄຸບຄຸມຄວາມເຮົວໃນກາຮປັບຄ່າຕົວແປຣຕັດສິນໄຈ ຂອງຊັ້ນຕອນ
ວິທີເກີດໄດ້ປ່ມຫາຄ່ານ້ອຍທີ່ສຸດ ເຊັ່ນ ວິທີລົງເກຣດີຍົນຕໍ່.

ເຈື່ອນໄຂກາຮຈບ (terminating condition): ເຈື່ອນໄຂທີ່ໃຊ້ຫຼຸດກາຮຄໍານວນ.

ກາຮກໍານົດຄ່າເຮີມຕັນ (initialization): ກາຮກໍານົດຄ່າເຮີມຕັນໃຫ້ກັບຕົວແປຣ.

ກາຮສູ່ເຂົາ (convergence): ພລລັພົກຈາກວິທີກາຮຄໍານວນແບບວນໜ້າທີ່ຄ່າຂອງພລລັພົກເຂົາໄກລັກ່າ່ ຖ້າ ມີ
ມາກີ້ນເຮືອຍ ທ່ານີ້ຈຳນວນຮອບຄໍານວນເພີ່ມເຂົ້ນ.

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

“วิธีการมีเป็นล้านและมากกว่าแต่หลักการมีไม่มาก บุคคลผู้ยึดในหลักการสามารถเลือกวิธีการได้อย่างดี. บุคคลผู้ลองแท้วิธีการ ละเลยหลักการ ย่อมแน่นอนว่าจะเดือดร้อน”

—ราล์ฟ วัลโด 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 หรือคำตอบมีหลายชุด. สังเกต แต่ละแผล เป็นอิสระเชิงเส้นกัน แต่สอดม Märkvärd ของ x และ y) ไม่เป็นอิสระเชิงเส้นกัน จากมุ่งมองของสารสนเทศที่ได้รับ ถ้าการไม่เป็นอิสระเชิงเส้นกัน ของแผล หมายถึง สมการที่ได้ไม่ได้ให้สารสนเทศเพียงพอ สมการหนึ่งได้มาจากการแปลงเชิงเส้นของสมการอื่น แล้ว การไม่เป็นอิสระเชิงเส้นกันของสอดม Märkvärd จะหมายถึงอะไร อภิปราย

แบบฝึกหัด 2.2

ความแปรปรวน (variance) เป็นค่าประเมินการแจกแจงของข้อมูลจากค่าเฉลี่ย. ความแปรปรวน อาจถูกสับสนกับเอนโทรปี. 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) เป็นฟังก์ชันทางเดียว (monotonic function) และการทำลับ ช่วยให้ได้ความสัมพันธ์ที่ $h(x)$ ค่าน้อย เมื่อ $\Pr(X = x)$ ค่ามาก และ $h(x)$ ค่ามาก เมื่อ $\Pr(X = x)$ ค่าน้อย นอกจากนั้น ยังช่วยให้ $h(x)$ มีค่ามากกว่าหรือ

เท่ากับศูนย์ด้วย. ค่าเฉลี่ย หรือค่าคาดหมายของปริมาณสารสนเทศ

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

จะเรียกว่า เอนโโทรปีสารสนเทศ.

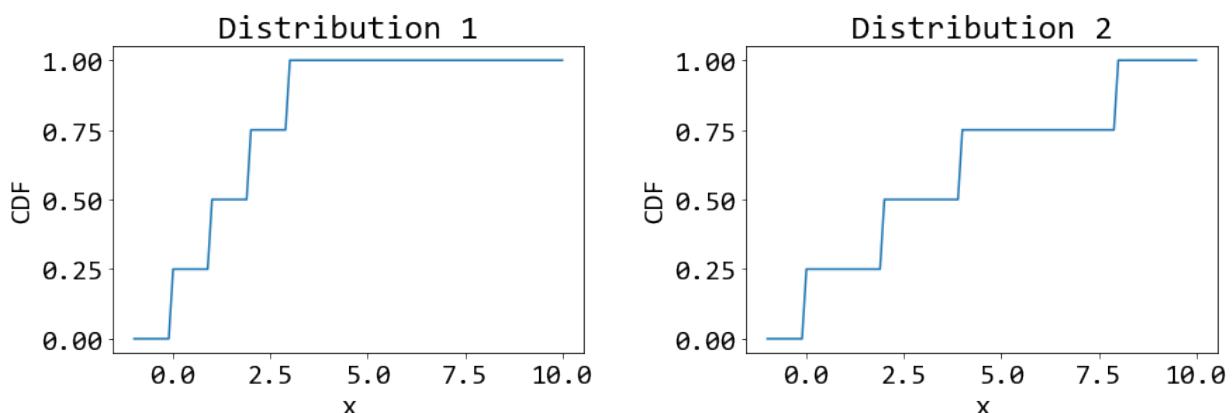
จงคำนวณค่าความแปรปรวนและค่าเอนโโทรปี ของข้อมูลต่อไปนี้. ข้อมูลชุดที่ 1 มีความน่าจะเป็นดังนี้

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

และข้อมูลชุดที่ 2 มีความน่าจะเป็นดังนี้

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

รูปที่ 2.20 แสดงการแจกแจงของข้อมูลทั้งสองชุด. การแจกแจง $F(x) = \Pr(X \leq x) = \sum_{v \leq x} \Pr(X = v)$. สังเกตผลลัพธ์ของค่าความแปรปรวนและค่าเอนโโทรปีที่คำนวณได้. สรุปและอภิปรายค่าความแปรปรวน และค่าเอนโโทรปีของข้อมูลสองชุดนี้ พร้อมอภิปรายความต่างกันของค่าความแปรปรวน และค่าเอนโโทรปี โดยทั่วไป และยกตัวอย่างลักษณะของข้อมูลที่ทำให้เห็นความต่างชัดเจนประกอบ.



รูปที่ 2.20 ตัวอย่างแสดงความต่างของค่าแปรปรวนและเอนโโทรปี. 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 และเลือกค่าของ Lagrange multiplier อย่างไร ยกตัวอย่างค่าที่ทำงานได้ ดูแบบฝึกหัด 2.25 สำหรับผลจากการแก้ปัญหา ด้วยโปรแกรม.

คำใบ้ พจน์แรกในฟังก์ชันจุดประสงค์ได้แก่ $0.2 \log(1 + e^{-v-3})$ คือ ฟังก์ชันบวกก้อน (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 ที่พิ่งกล่าวไป 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 ฟังก์ชัน gauss เซียน (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 และเลือกค่า Lagrange multiplier อย่างไร ยกตัวอย่างค่าที่ทำงานได้

คำใบ้ อนุพันธ์ของฟังก์ชันต่อเนื่องเป็นช่วง สามารถหาได้ในแต่ละช่วง.

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

แบบฝึกหัดการเขียนโปรแกรม

แบบฝึกหัดไพธอนแก่น

แบบฝึกหัดต่อไปนี้ ทบทวนการเขียนโปรแกรมด้วยภาษาไพธอนพื้นฐาน ยังไม่แนะนำให้เรียกใช้โมดูลอื่น ณ 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 ว่าขนาดของเมทริกซ์สามารถทำการคูณเมทริกซ์ได้ ทำการคูณเมทริกซ์ และรีเทิร์นผลลัพธ์ออกมาในรูปลิสต์ ที่มีโครงสร้างลักษณะเดียวกัน

แบบฝึกหัด 2.7

การหาเมทริกซ์ผกผัน ด้วยวิธีเกลล์จอร์ден (*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

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

- ดำเนินปฏิบัติการเชิงเส้นกับแถวของ \mathbf{A}' จนส่วนหน้ากล้ายเป็นเมทริกซ์เอกลักษณ์ ดังเช่น

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

รายละเอียดการดำเนินปฏิบัติการ คือ (1) ที่แล้ว i เพื่อทำให้ส่วนประกอบแนวทะแยงมุมเป็นหนึ่ง ดำเนินการ $r_{ij} = a_{ij}/a_{ii}$ สำหรับ $j = 1, \dots, 2n$ เมื่อ r_{ij} คือค่าส่วนประกอบภายหลัง

การคำนวณที่ตำแหน่ง (i, j) . (2) ที่แล้วอีน ๆ และ $k \neq i$ เพื่อทำให้ส่วนประกอบ ตำแหน่ง (k, i) นอกแนวทางเดิมเป็นศูนย์ ดำเนินการ $s_{kj} = a_{kj} - r_{ij} \cdot a_{ki}$ สำหรับ $j = 1, \dots, 2n$ เมื่อ s_{ij} คือค่าส่วนประกอบภายหลังการคำนวณที่ตำแหน่ง (i, j) . (3) ดำเนินการเช่นนี้ โดยทำทุกแถว จากแถว $i = 1, \dots, n$ แล้วผลลัพธ์จะได้ \mathbf{B}' .

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

จากโปรแกรมตัวอย่างในรายการ 2.1 จะเขียนโปรแกรมเพื่อน เพื่อหาเมตริกซ์ผกผันด้วยวิธีเกาส์จอร์เดน และทดสอบผลลัพธ์ที่ได้ โดยใช้การคูณเมตริกซ์ จากแบบฝึกหัด 2.6 เล้าศึกษาโปรแกรม และอภิปรายข้อจำกัด และกรณีที่อาจเสี่ยงทำให้โปรแกรมรันผิดพลาด พิรุณเสนอการปรับปรุงแก้ไข. คำให้ การหารด้วยศูนย์จะทำให้โปรแกรมล้มได้.

รายการ 2.1: ตัวอย่างโปรแกรม วิธีหารด้วยวิธีเกาส์จอร์เดน

```

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)

```

แบบฝึกหัดไพธอนด้วยนัมไพ

เนื่องจากโปรแกรมการรูปจำลองแบบ ใช้การคำนวณเชิงเลข (*numerical computation*) อย่างมาก ดังนั้น การใช้มอดูลการคำนวณเชิงเลขประกอบจึงมีประโยชน์มาก เพื่อช่วยลดภาระ และเพื่อจะได้ทุ่มเทไปที่ศาสตร์ของการรูปจำลองแบบ และการเรียนรู้ของเครื่องได้เต็มที่มากขึ้น. หัวข้อนี้ แนะนำมอดูลคำนวณเชิงเลข นัมไพ (*Numpy*) และแมทพล็อตลิบ (*Matplotlib*) เพื่อช่วยการคำนวณเชิงเลข และการนำไปสร้างภาพให้เห็น (*visualization*)

นอกจากมอดูลนัมไพ และแมทพล็อตลิบ มีมอดูลที่มีประโยชน์ และนิยมใช้ สำหรับงานรูปจำลองแบบ อีกมาก ซึ่ง ตำนานี้ จะได้แนะนำมอดูลอื่น ๆ อีก ตามเนื้อหาที่เหมาะสมต่อไป

ตัวอย่างโปรแกรมในแบบฝึกหัดต่าง ๆ จากนี้ จะนำเข้ามอดูล *numpy* และ *matplotlib* ด้วยคำสั่ง

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

ซึ่งจะนำเข้ามอดูล `numpy` และ `matplotlib.pyplot`. มอดูล `pyplot` เป็นมอดูลย่อยเพื่อใช้ในการวาดกราฟ. การนำเข้านั้น ได้ตั้งชื่อใหม่ให้กับมอดูลทั้งสองเพื่อความสะดวก เป็น `np` และ `plt`.

แบบฝึกหัด 2.8

จงสร้างตัวแปร `a1` เป็น `np.array` จาก `list` เช่น `[2, 4, 8, 9]` และเรียนรู้คำสั่งต่อไปนี้ คำสั่ง `len(a1)` คำสั่ง `a1.shape` คำสั่ง `type(a1)` คำสั่ง `a1.tolist()` และสร้าง `a2` เป็น `np.array` จาก `list` สองมิติ เช่น `[[2, 4], [8, 9]]` และเรียนรู้คำสั่งแบบเดียวกัน รวมทั้งทดลอง และอภิปรายผลของคำสั่งตัวต่อๆ กัน (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 ดัดแปลง 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()`
- ทดลองและเปรียบเทียบ 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 และเปรียบเทียบผลกับแบบฝึกหัด 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

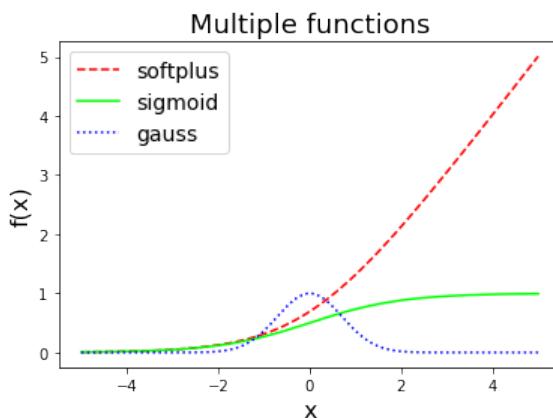
จงหาค่าและเวกเตอร์ลักษณะเฉพาะของเมทริกซ์ในแบบฝึกหัด 2.10.

คำใบ้ พังก์ชัน `numpy.linalg.eig` เรียกใช้ เลแพค (LAPACK) ซึ่งเป็นมอดูลที่ได้รับความเชื่อถือในการคำนวณค่าและเวกเตอร์ลักษณะเฉพาะ. เลแพค ใช้วิธีคำนวณด้วยการแยกตัวประกอบค่าเอกฐาน (*singular value decomposition*) ที่มีประสิทธิภาพมากกว่า ตัวอย่างวิธีคำนวณที่อภิปรายในหัวข้อ 2.1.

แบบฝึกหัด 2.12

จงเขียนโปรแกรม เพื่อวาดกราฟของพังก์ชันบวกอ่อน $f(x) = \log(1 + e^x)$ พังก์ชันซิกมอยด์ $f(x) = 1/(1 + e^{-x})$ และพังก์ชันเก้าอี้ยน $f(x) = e^{-x^2}$ โดย วาดกราฟทั้งสามให้อยู่ในสภาพเดียวกัน ดังแสดงในรูปที่ 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 กราฟแสดงพฤติกรรมของฟังก์ชันบวกก้อน ฟังก์ชันซิกมอยด์ และฟังก์ชันเกาส์เซียน

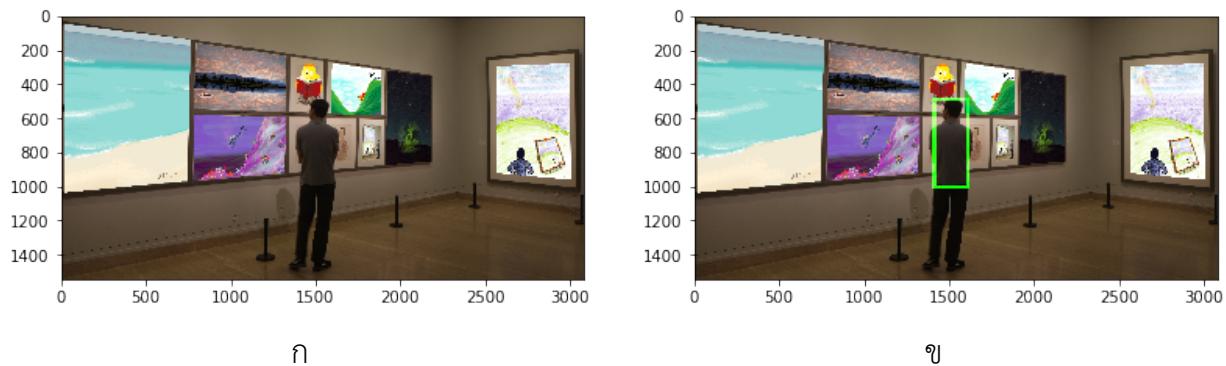
แบบฝึกหัด 2.13

จงเขียนโปรแกรม โดยใช้มอdule `matplotlib.pyplot` เพื่อนำเข้าภาพ (ซึ่งอาจจะเป็นไฟล์นามสกุล `.png`) แล้วนำมาแสดง ดังตัวอย่างในรูปที่ 2.22 (ภาพ ก).

การดำเนินการนี้ ต้องอ่านภาพเข้ามาก่อน และวิธีคือการแสดงภาพออกหน้าจอ. คำสั่ง เช่น `img = plt.imread('t.png')` อ่านภาพจากไฟล์ชื่อ '`t.png`'. ตรวจสอบสัดส่วนและลักษณะข้อมูล เช่น ใช้คำสั่ง `print(img.shape)` ใช้คำสั่ง `print(img)`.

สังเกต `plt.imread` อ่านภาพ ออกแบบมาเป็นข้อมูลที่มีสัดส่วน 4 ลำดับชั้น สำหรับ ช่องสีแดง ช่องสีเขียว ช่องสีน้ำเงิน และช่องอัลฟ่า (*alpha channel*) และแต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1. การแสดงภาพทำได้ ด้วยคำสั่ง เช่น `plt.imshow(img)`.

หลังจากนั้นทดลองเปลี่ยนค่าพิกเซลของภาพ แล้วสังเกตผล. ตัวอย่างในรูปที่ 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 ตามแนวนอน ในช่องสีที่หนึ่ง (ช่องสีเขียว) โดยเปลี่ยนค่าเป็นหนึ่ง (ค่ามากที่สุด). เมื่อเปลี่ยนค่าเสร็จแล้ว และนำข้อมูลมาแสดงดูใหม่ จะเห็นเส้นสีเขียวปรากฏขึ้นมา (ภาพ ข มีการเปลี่ยนค่าลักษณะนี้สีครึ้ง สำหรับ สีเส้นที่เห็นแสดงเป็นกรอบ). หมายเหตุ ถ้ากำหนดค่าใหม่ให้กับพิกเซลแล้ว ภาพนั้นจะมีตำแหน่งนั้นคือ เส้นสีเขียวจะไม่สามารถถูกลบออกได้ (แต่ถูกเปลี่ยนทับได้). หากต้องการเก็บภาพตัวฉบับไว้ ให้ทำสำเนาไว้ก่อนที่จะมีการตัดแปลงภาพ เช่น `marked = img.copy()` และดำเนินการเปลี่ยนค่าพิกเซล ที่สำเนาแทน.



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

แบบฝึกหัด 2.14

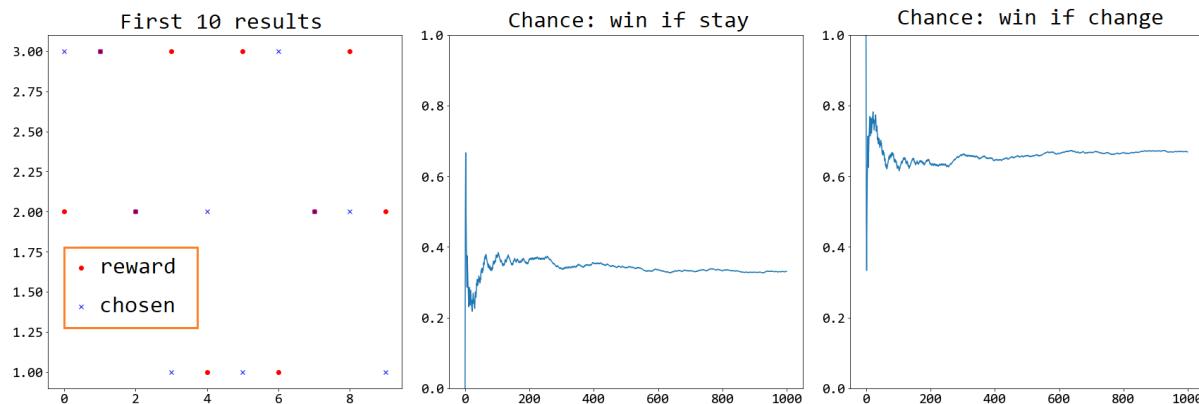
จากค่าฟังก์ชัน $g(v) = -e^{-53-v_1^2-2v_2^2-v_1v_2+10v_1+19v_2}$ จะเขียนโปรแกรม เพื่อวัดภาพคอนทัวร์ และภาพสามทัศนวิติ ดังรูปที่ 2.14.

คำໃບ คำสั่ง `plt.contour` จะวัดภาพคอนทัวร์ จากค่าพิกัดสามแกนวิติ ดังนี้เพื่อจาวดภาพคอนทัวร์ 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

จากปัญหามอนตีสอล ในหัวข้อ 2.2 จะเขียนโปรแกรมเพื่อจำลองสถานการณ์ และแสดงให้เห็นว่า โอกาสที่ผู้เข้าแข่งขันจะได้รางวัล หากเลือกที่จะเปลี่ยนประตุเป็น $2/3$ และโอกาสที่ผู้เข้าแข่งขันจะได้รางวัล หากเลือกประตุเดิมเป็น $1/3$.

อภิปรายโปรแกรม และผลการทดลอง พิร้อนออกแบบวิธีนำเสนอผล ให้ประจักษ์ชัดเจน. รูปที่ 2.23 แสดงตัวอย่างวิธีนำเสนอผล ซึ่งจากรูปจะเห็นว่า เมื่อจำนวนช้ำมากขึ้น โอกาสที่จะชนะเมื่อเลือกประตุเดิม (ภาคลาง) จะประมาณ 0.33 หรือ $1/3$ ในขณะที่ถ้าเปลี่ยนไปประตุใหม่ โอกาสจะประมาณ 0.67 หรือ $2/3$.



รูปที่ 2.23 ผลลัพธ์จากการจำลองสถานการณ์ปัญหามอนตี้霍ล. ภาพซ้าย แสดงผลลัพธ์ 10 ครั้งแรก. สัญลักษณ์ ‘○’ แสดงแทนรางวัล สัญลักษณ์ ‘x’ แสดงแทนประตูที่ผู้แข่งขันเลือก เช่น ครั้งแรกสุด (ครั้งที่ 0) ผู้แข่งขันเลือกประตูที่สาม แต่รางวัลอยู่ประตูที่สอง. ประตูที่พิธีกรเปิดไม่ได้แสดงในภาพ. ภาพกลาง แกนต์แสดงอัตราส่วนสะสมของจำนวนครั้งที่ผู้แข่งขันชนะได้รางวัล เมื่อเลือกประตูเดิม. ส่วนแกนนอนแสดงจำนวนครั้งที่จำลองข้า. ภาพขวาแสดงอัตราส่วนสะสมของจำนวนครั้งที่ผู้แข่งขันชนะได้รางวัล เมื่อตัดสินใจเปลี่ยนไปประตูใหม่.

คำໃບ คำสั่ง `np.random.choice` ใช้สุ่มเลือกค่าจากลิสต์. 朗วลดอาจสุ่มให้อยู่ประตูไหน ก็ได้. ประตูที่ผู้เข้าแข่งขันเลือก ก็อาจอยู่ประตูไหนก็ได้. แต่ประตูที่พิธีกรเลือกเปิด จะเป็นประตูที่มีรางวัลไม่ได้ หรือจะเป็นประตูที่ผู้แข่งขันเลือกก็ไม่ได้. การเลือกประตูของพิธีกร ต้องทำหลังจากการเลือกประตู朗วลด และการเลือกประตูของผู้แข่งขัน.

แบบฝึกหัด 2.16

ผลการคำนวณค่าสถิติจากข้อมูลที่จำกัด จะมีความไม่แน่นอนอยู่¹⁵ ในแท่งที่ว่า หากสุ่มกลุ่มข้อมูล ออกมากใหม่ แล้วผลการคำนวณอาจจะเปลี่ยนไป.

หากคำนวณค่าเฉลี่ยของกลุ่มข้อมูลที่สุ่มมาจากการแจกแจงเดียวกัน และนำค่าเฉลี่ยของแต่ละการสุ่มมาวิเคราะห์ จะพบว่า[76] ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย

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

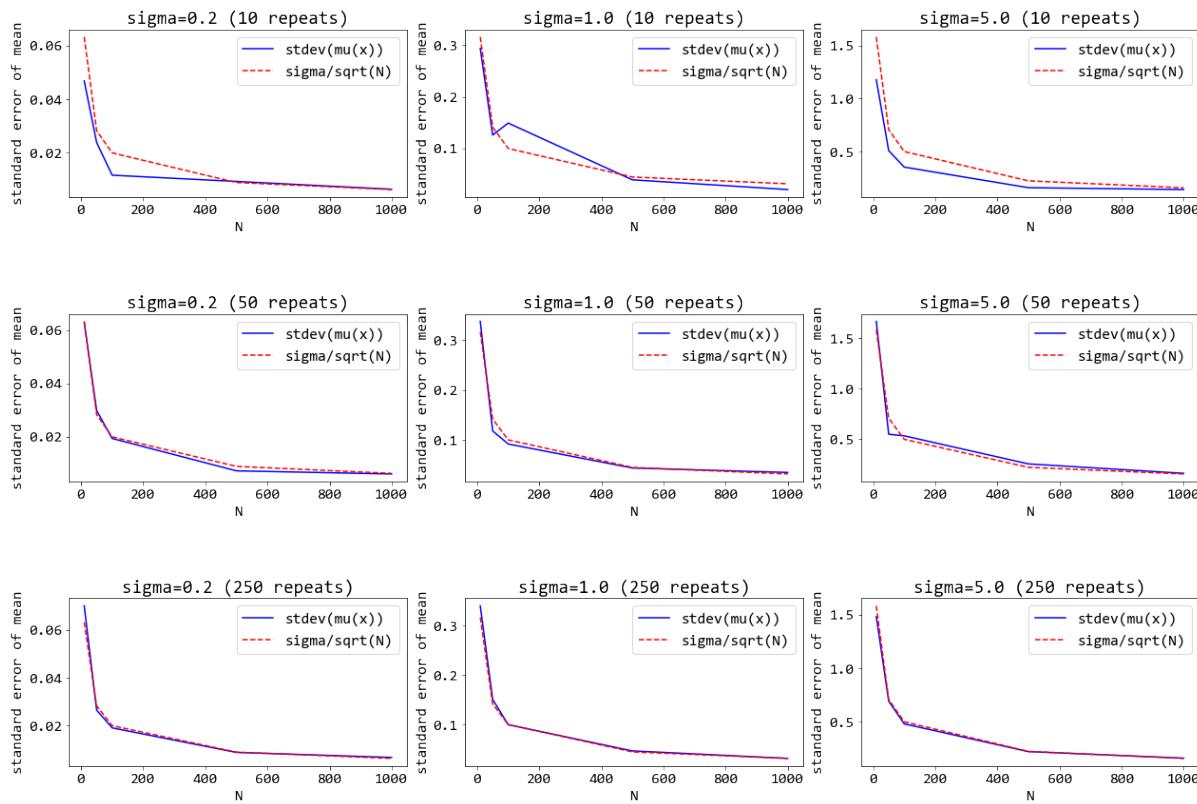
เมื่อ μ_N คือค่าเฉลี่ยที่ของกลุ่มขนาด N และ x_n คือจุดข้อมูลในกลุ่มที่สุ่มขึ้นมา และ σ คือ ค่าเบี่ยงเบนมาตรฐานของการแจกแจง.

จากทฤษฎีข้างต้น ยิ่งกลุ่มข้อมูลมีขนาดใหญ่ (N มีค่ามาก) ค่าเฉลี่ยที่คำนวณได้จะยิ่งมีความแม่นยำใกล้เคียงกับค่าเฉลี่ยจริงของการแจกแจงเท่านั้น.

จงเขียนโปรแกรม เพื่อแสดงในเห็นว่า สมการ 2.61 เป็นจริง.

¹⁵เนื้อหาในแบบฝึกหัดนี้ ได้รับแรงบันดาลใจและอิทธิพลหลัก ๆ จาก [76].

ตัวอย่างการนำเสนอ แสดงในรูปที่ 2.24. (ไม่จำเป็นต้องทำตามตัวอย่างนี้.) ผลที่ได้ มาจากการจำลอง (*simulation*) โดย กำหนดการแจกแจงแบบเกาส์เซียน ที่มีค่าเฉลี่ยเป็นศูนย์ และค่าความแปรปรวน σ^2 และจำลองการสุ่มกลุ่มจำนวน N ขึ้นมา. ตัวอย่างนี้ ทดลองค่า σ สามค่าได้แก่ 0.2, 1.0, และ 5.0 และจำลองด้วย ภาพซ้าย กลาง และขวา ตามลำดับ.



รูปที่ 2.24 ความสัมพันธ์ของความคลาดเคลื่อนกับการคำนวณกับขนาดข้อมูล. แต่ละภาพ แสดง ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ที่คำนวณจากกลุ่มข้อมูลที่สุ่มขึ้นมา (เส้นทึบ สีน้ำเงิน) กับค่าที่คำนวณทางทฤษฎี (เส้นประ สีแดง). แกนนอน แสดง จำนวนจุดข้อมูลในกลุ่ม. ภาพต่าง ๆ ทางซ้าย แสดงผล เมื่อการแจกแจงของข้อมูลมี $\sigma = 0.2$ ภาพต่าง ๆ กลาง แสดง $\sigma = 1$ และภาพต่าง ๆ ทางขวา $\sigma = 5$. ภาพต่าง ๆ ในแนวนอน แสดงผลเมื่อทำซ้ำ 10 ครั้ง และกลาง ทำซ้ำ 50 ครั้ง และล่าง ทำซ้ำ 250 ครั้ง.

จากนั้น คำนวณค่า μ_N ของกลุ่ม โดยทำซ้ำ 10 ครั้ง (ผลแสดงในภาพต่าง ๆ แนวนอน) ทำซ้ำ 50 ครั้ง (กลาง) และทำซ้ำ 250 ครั้ง (ล่าง). 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 เส้นทึบสีน้ำเงิน และสัญกรณ์ $stdev(mu(x))$ ในภาพ) เป็นค่าที่คำนวณจากข้อมูล ส่วน σ/N เป็นค่าที่คำนวณจากทฤษฎี (แทนด้วย เส้นประสีแดง และสัญกรณ์ $sigma/sqrt(N)$ ในภาพ). ในแต่ละภาพ แกนตั้ง แสดงค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย และแกนนอน แสดงค่าขนาดของกลุ่ม N ซึ่งในตัวอย่างใช้ 10, 50, 100, 500, และ 1000.

ผลที่ได้ แสดงในเห็นว่า (1) เมื่อ N ขนาดใหญ่ขึ้น ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ยมีค่าเล็กลง. (2) ถ้า σ ของการแจกแจง มีค่าใหญ่ ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ก็จะมีค่าใหญ่ตามสัดส่วน. (3) ค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ย ที่ได้จากการคำนวณกับข้อมูล เป็นไปตามทฤษฎี และจะเห็นความสอดคล้องได้ชัดเจนมากขึ้น เมื่อจำนวนช้ำเพิ่มขึ้น.

หมายเหตุ ในทางปฏิบัติ เราไม่รู้ค่า σ ของการแจกแจง. ดังนั้น การประเมินความคลาดเคลื่อนของการคำนวณ เมื่อกลุ่มข้อมูลมีขนาดจำกัด จะไม่สามารถทำได้สะดวกเช่นนี้. ในทางปฏิบัติ เราจะสามารถประเมินความคลาดเคลื่อน ต่อขนาดข้อมูลได้อย่างไร. หรือ ที่น่าสนใจกว่าคือคำถามว่า ทำอย่างไร เราถึงจะสามารถประเมิน ได้ว่าภาระกิจที่กำลังทำอยู่ ต้องการจำนวนข้อมูลเท่าไร. อภิปรายพร้อมให้เหตุผลประกอบ.

แบบฝึกหัด 2.17

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

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

และหลังจากมีเกรเดียนต์แล้ว โปรแกรมวิธีลงเกรเดียนต์ก็สามารถทำงานได้ ดังแสดงในรายการ 2.2. โปรแกรมเลือกใช้ค่าขนาดก้าว **step_size** เป็น 0.5 (ตัวแปร α ในสมการ 2.46) และใช้เงื่อนไขจบเป็นจำนวนรอบสูงสุด ซึ่งในที่นี้ใช้เป็น 8 รอบ ที่ระบุลงไปให้ลูปเลย ด้วย **for i in range(8)** ค่าเริ่มต้นของตัวแปรตัดสินใจ เลือกกำหนดให้เป็น 6.5 ในบรรทัดที่สอง และคำสั่งในบรรทัดที่สี่คือการคำนวณสมการ 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 นิยมเพิ่มเงื่อนไขการจบ เพื่อให้สามารถเลือกจำนวนรอบสูงสุดไว้มาก ๆ ก่อน โดยไม่ต้องกังวลว่า จะเสียเวลา ran โดยไม่จำเป็น (ดูแบบฝึกหัด 2.19 ประกอบ)

จากแบบฝึกหัด 2.17 โปรแกรมวิธีลิงเกรเดียนต์ง่าย ๆ ที่แสดงในรายการ 2.2 สามารถเพิ่มเงื่อนไขการจบทิ้งไปได้ ดังแสดงในรายการ 2.3. เนื่องจากจำนวนรอบสูงสุด ควบคุมได้โดยผ่านตัวแปร **Nmax**. โปรแกรมตัวอย่างนี้ ให้เงื่อนไขความคลาดเคลื่อนยินยอม วัดค่าผ่าน **eps** โดย ถ้า **eps** น้อยกว่าหรือเท่ากับค่าที่ยินยอมได้ ก็จะทำการคำนวณทันที. ค่าที่ยินยอมได้ (*tolerance*) กำหนดผ่าน **tol**.

รายการ 2.3: ตัวอย่างโปรแกรม วิธีลิงเกรเดียนต์ที่มีเงื่อนไขการจบทิ้งความคลาดเคลื่อนที่ยินยอมได้

```

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 มีการคำนวณที่ซ้ำซ้อนมาก ในแต่ละรอบ เช่น **grad(x)** มีการคำนวณค่าเดียวกันถึงสามครั้ง. 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 ... แทนโค๊ดต่าง ๆ ที่เหลือ ไม่ได้หมายถึงการพิมพ์จุดสามครั้งเข้าไปในโปรแกรม. ตัวแปร **gradx** ใช้เก็บค่าเกรเดียนต์ที่คำนวณไว้แล้ว. ค่าเกรเดียนต์ที่ **x = x - step_size*gradx** จะเป็นคงคละค่ากับที่แสดงด้วยคำสั่ง **print** และที่ใช้กำหนดค่า **eps** ดังนั้นตำแหน่งของ **gradx = grad(x)** จึงต้องอยู่หลังการปรับค่า **x** และเพื่อให้รอบคำนวณแรกสามารถทำได้ จึงต้องมีคำสั่ง **gradx = grad(x)** อยู่ก่อนเข้าลูปด้วย.

จากโปรแกรมในรายการ 2.3 จะทดลองรัน และเปรียบเทียบผล แล้วทดลองเปลี่ยนค่า **tol** เป็นค่าอื่น ๆ เช่น 0, 1e-5, 1e-3.

คำตามเพิ่มเติม. ทำไมเงื่อนไขความคลาดเคลื่อน นั้นคือ $\text{eps} \leq \text{tol}$ ถึงเขียนด้วยการเปรียบเทียบน้อยกว่าหรือเท่ากับ ทดลองเปลี่ยนเป็นการเปรียบเทียบน้อยกว่า ได้แก่ $\text{eps} < \text{tol}$ และทดลองค่า **tol** ต่าง ๆ อีกรังส์ สังเกตและอภิปรายผล

แบบฝึกหัด 2.19

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

อภิปรายถึงประเด็นปัญหาที่จะเกิด เมื่อนำโปรแกรมนี้ไปรันในทางปฏิบัติ แล้วทำการทดลองใหม่ โดยรันโปรแกรมในรายการ 2.3 และอภิปรายข้อดีของการใช้งานในการจับ ที่มีในโปรแกรม 2.3.

แบบฝึกหัด 2.20

เงื่อนไขความคลาดเคลื่อนยินยอม จะกำหนดค่าความคลาดเคลื่อนที่ยอมรับได้ ซึ่งอาจทำได้หลายวิธี เช่น ใช้ค่าคลาดเคลื่อนของตัวแปร $\epsilon_v = \|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|$, หรือใช้ค่าคลาดเคลื่อนของเกรเดียนต์ $\epsilon_\nabla = \|\nabla g(\mathbf{v}^{(k+1)}) - \nabla g(\mathbf{v}^{(k)})\|$ หรือของฟังก์ชันจุดประสงค์ $\epsilon_g = |g(\mathbf{v}^{(k+1)}) - g(\mathbf{v}^{(k)})|$ หรือใช้ค่าความคลาดเคลื่อนข้างต้นผสมกัน. 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 วิธีลงเกรเดียนต์ต้องการฟังก์ชันคำนวณค่าเกรเดียนต์ 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 \mathbf{u} แทนเวกเตอร์ $\mathbf{u} = [u_1, u_2]^T$. ค่าส่วนประกอบ u_1 และ u_2 เข้าถึงได้โดย $\mathbf{u}[0, 0]$ และ $\mathbf{u}[1, 0]$ ตามลำดับ. นอกจากนี้ 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 วิธีลิงเกรเดียนต์ เมื่อตัวแปรเป็นเวกเตอร์.

รายการ 2.4: ตัวอย่างโปรแกรม วิธีลิงเกรเดียนต์ 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('Terminate at i={} with {}'.format(i,eps))
17        break
18
19    print('{:}: v= [{:.3f} {:.3f}]; grad = [{:.4f} {:.4f}]'.format(i,v[0,0],v[1,0],gradv[0,0],gradv[1,0]))
20
21 print('{:}: v= [{}, {}]; g = {}'.format(i,v[0,0],v[1,0],loss))
```

สังเกตว่า โปรแกรมบันทึกค่าฟังก์ชันจุดประสงค์ หรือเรียกว่า “ \rightarrow ” ว่า ค่าสูญเสีย (`loss`) ของทุกรอบการคำนวณไว้ใน `losses`. ค่าสูญเสียต่อรอบคำนวณ สามารถนำมาใช้ เพื่อตรวจสอบการทำงานแก้ปัญหาค่าน้อยที่สุดได้ ดังแสดงในภาพข้างบนของรูปที่ 2.16. ภาพอื่น ๆ ในรูปที่ 2.16 ก็

จะสามารถทำได้ในแบบเดียวกัน เพียงแต่ต้องบันทึกค่านั้น ๆ ขณะรันด้วย ซึ่งในที่นี้ ขอละไว้เพื่อไม่ให้โปรแกรมดูซับซ้อนเกินไป.

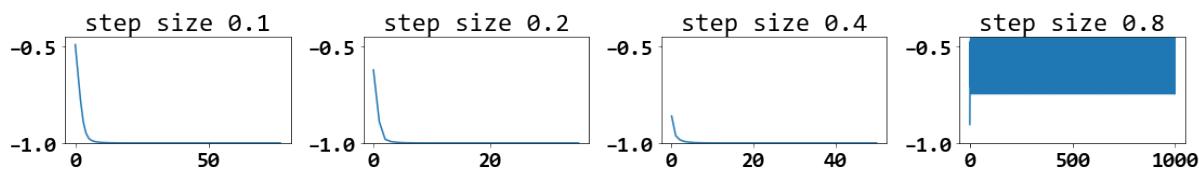
ลองเปรียบเทียบฟังก์ชัน **grad** กับการคำนวณหาค่าเกรเดียนต์ด้วยมือ และเปรียบเทียบโปรแกรมในรายการ [2.4](#) เปรียบเทียบกับรายการ [2.3](#) และอภิปรายจุดแตกต่าง

ทดลองรันโปรแกรมในรายการ [2.4](#) แล้วแก้ไขค่า **step_size** รัน และตรวจสอบผลที่ได้ กับการคำนวณของตัวอย่างในหัวข้อ [2.3](#).

แบบฝึกหัด 2.22

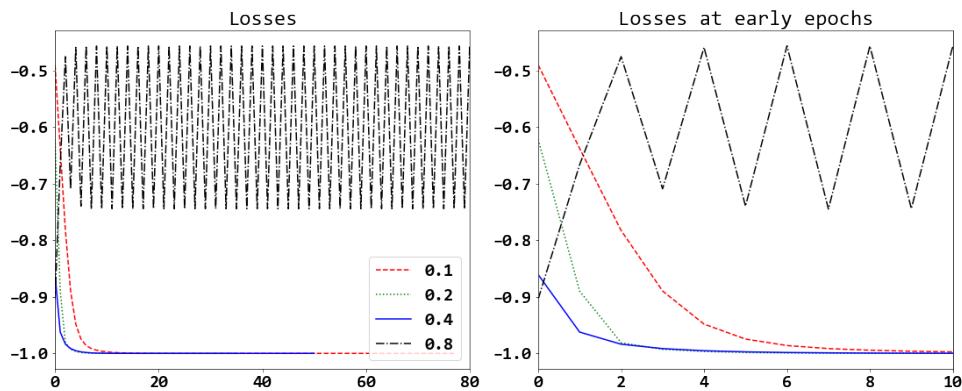
จากปัญหาในแบบฝึกหัด [2.21](#) ทดลอง เปลี่ยนค่าขนาดก้าว (ตัวแปร **step_size**) ต่าง ๆ เช่น $0.1, 0.2, 0.4, 0.8$ แล้วสังเกตผลและอภิปราย และเตรียมหลักฐานเพื่อประกอบการอภิปราย เช่น รูปที่ [2.25](#) รูปที่ [2.26](#) และรูปที่ [2.27](#). ทดลองค่าอภิมานพารามิเตอร์อื่น ๆ เพื่อยืนยันข้อสรุปที่ได้อภิปราย.

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

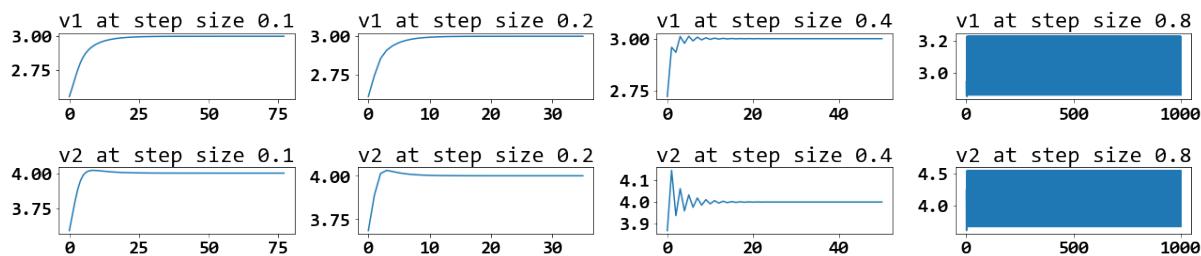


รูปที่ 2.25 ตัวอย่างแสดงผลการทำงานวิธีลิงเกรเดียนต์ เมื่อใช้ค่าขนาดก้าวต่าง ๆ.

รูปที่ [2.26](#) แสดงความก้าวหน้าของการแก้ปัญหา เมื่อใช้ค่าขนาดก้าวต่าง ๆ ในภาพเดียวกัน โดยค่าขนาดก้าวที่ใช้ระบุด้วยสัญลักษณ์ ดังแสดงในกรอบคำอธิบายลัญลักษณ์ (*legend*). ภาพช้ายังแสดง ในช่วง 80 รอบคำนวณแรก (เมื่อใช้ขนาดก้าวบางค่า การคำนวณจบก่อน 80 รอบ). ภาพขวาแสดง ในช่วง 10 รอบคำนวณแรก เพื่อให้เห็นชัดเจนว่าที่ค่าขนาดก้าวเท่าใดถึงเข้าสู่ค่าตอบได้เร็วที่สุด (ค่าสูญเสียลดลงต่ำสุด ในรอบคำนวณที่น้อยที่สุด หมายถึงการถึงเข้าสู่ค่าตอบได้เร็วที่สุด). ตัวอย่างนี้ จะเห็นว่า ขนาดก้าว 0.4 ช่วยให้วิธีลิงเกรเดียนต์ถึงเข้าเร็วที่สุด และขนาดก้าวที่น้อยลง มีผลให้ถึงเข้าช้าลง. แต่หากใช้ขนาดก้าวที่ใหญ่เกินไป (เช่น 0.8 ในตัวอย่างนี้) อาจทำให้วิธีลิงเกรเดียนต์ไม่สามารถถึงเข้าสู่ค่าตอบได้. รูปที่ [2.27](#) แสดงค่าของตัวแปรตัดสินใจ v_1 และ v_2 สังเกตว่า เมื่อใช้ขนาดก้าว $0.1, 0.2, 0.4$ ค่าของตัวแปรตัดสินใจถึงเข้าสู่ค่า 3 และ 4 ตามลำดับ โดย เมื่อใช้ขนาดก้าว 0.4 ค่าของ v_1 และ v_2 มีการส่ายอยู่บ้างก่อนส่ายน้อยลงจนถึงเข้าสู่ค่าตอบ. แต่เมื่อใช้ขนาดก้าว 0.8 ค่าของ v_1 และ v_2 แสดงการส่ายต่อเนื่องไปจนครบ 1000 รอบฝึกโดยไม่มีแนวโน้มจะถึงเข้า ขนาดก้าว 0.8 เป็นขนาดก้าวที่ใหญ่เกินไปอย่างชัดเจน. อภิปราย พฤติกรรมนี้ พร้อมว่าด้วยทางการหาค่าทำให้น้อยที่สุด (เช่นรูปที่ [2.15](#)) เพื่อประกอบการอภิปราย.



รูปที่ 2.26 ผลการทำงานวิธีลิงเกรเดียนต์ ในรอบคำนวณต้น ๆ เมื่อใช้ค่าขนาดก้าวต่าง ๆ เมื่อคาดรวมกัน.



รูปที่ 2.27 ผลลัพธ์จากวิธีลิงเกรเดียนต์ เมื่อใช้ค่าขนาดก้าวต่าง ๆ.

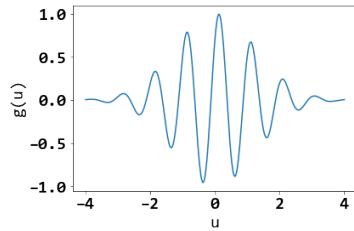
หมายเหตุ อย่างไรก็ตาม ค่าขนาดก้าวนี้ไม่จำเป็นต้องใช้เป็นค่าเดียวกันตลอดทุกรอบการคำนวณ อาจเปลี่ยนขนาดได้ตามความเหมาะสม เช่น อาจปรับให้ขนาดเล็กลง เมื่อจำนวนรอบคำนวณมาก ๆ ได้ หรืออาจใช้กลไกการปรับที่ซับซ้อนขึ้นได้ เช่น วิธีลิงเกรเดียนต์ชันที่สุด (steepest gradient descent method ดู [39] เพิ่มเติม สำหรับผู้ที่สนใจ).

แบบฝึกหัด 2.23

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

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 ทำไม่ถึงเป็นเช่นนั้น?

รูปที่ 2.28 แสดงค่าฟังก์ชันจุดประสงค์ $g(u)$. อภิปรายการใช้งานวิธีลิงเกรเดียนต์กับปัญหาลักษณะนี้ โดยเฉพาะในทางปฏิบัติ ที่มักไม่สามารถวาดกราฟของฟังก์ชันจุดประสงค์ได้.



รูปที่ 2.28 ตัวอย่างปัญหาหลายภาวะ. 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 ปัญหาค่าน้อยที่สุดที่มีค่าทำน้อยที่สุดท่องถิ่นหลายที่ และวิธีลงเกรเดียนต์จะพบคำตอบที่ใกล้ที่สุด ที่ทิศทางเกรเดียนต์ของจุดเริ่มต้นซึ่งไป เมื่อใช้ขนาดกว้างเล็กพอก.

แบบฝึกหัด 2.24

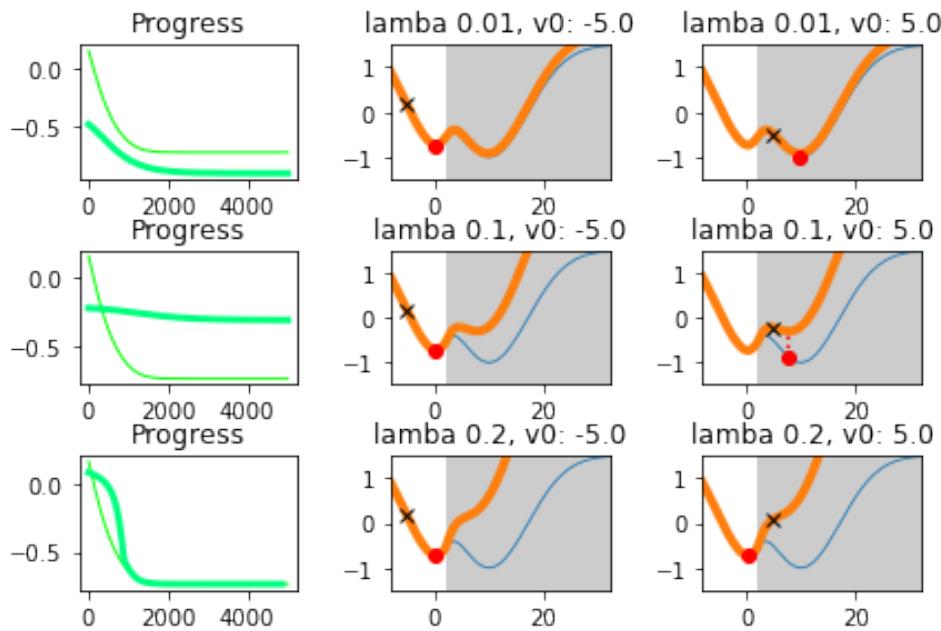
จงแก้ปัญหานี้ในแบบฝึกหัด 2.23 โดยใช้การกำหนดค่าเริ่มต้นด้วยวิธีการสุ่ม แล้วทดลอง เพิ่มจำนวนทำซ้ำให้มากขึ้นเท่าตัว. สังเกตผลลัพธ์ที่ได้ อภิปรายว่า การกำหนดค่าเริ่มต้นด้วยวิธีการสุ่ม จะช่วยบรรเทาปัญหานี้ของการติดอยู่ในสถานการณ์ที่ดีที่สุดท่องถิ่นได้อย่างไร

คำใบ้ ลองคำสั่งสุ่มค่า เช่น `np.random.uniform`, หรือ `np.random.normal`, หรือ `np.random.rndn`.

แบบฝึกหัด 2.25

จากแบบฝึกหัด 2.3 จงเขียนโปรแกรม เพื่อแก้ปัญหานี้แบบมีข้อจำกัด และเปรียบเทียบผลที่ได้ กับผลที่แสดงในรูปที่ 2.29. ผลที่แสดงในรูปที่ 2.29 ได้จากการคำนวณวิธีลงเกรเดียนต์ โดยใช้ รอบคำนวณ สูงสุด เป็น 5000 และใช้ค่าขนาดกว้างเป็น 0.02.

รูปที่ 2.29 แสดงให้เห็นว่า เมื่อเลือกใช้ค่า λ ขนาดใหญ่มากพอ ไม่ว่าจะเลือกใช้ค่าเริ่มต้นที่ไหน คำตอบที่ได้จะเป็นค่าที่ใช้ได้ (feasible) ดังที่เห็นในแควรล่างสุด ภาพกลางและขวา.



รูปที่ 2.29 ตัวอย่างผลการแก้ปัญหาค่าน้อยที่สุดแบบมีข้อจำกัด เมื่อใช้วิธีการลงโทษ. ตัวอย่างการทำงานของวิธีลงเกรเดียนต์ ในแบบฝึกหัด 2.25. แต่ละแถวแสดงผล เมื่อเลือก λ เป็น 0.01, 0.1, 0.2 ตามลำดับ. ภาพในส่วนแรก แสดงความก้าวหน้า (ค่าฟังก์ชันจุดประยงค์ต่อรอบปีก) โดยเส้นบางสีเขียว แสดงความก้าวหน้า เมื่อค่าเริ่มต้นเป็น -5 และ เส้นหนาสีน้ำเงินเขียว แสดงความก้าวหน้า เมื่อค่าเริ่มต้นเป็น 5 . กราฟทั้งหมดลุ้นเข้าดี ยกเว้นที่ $\lambda = 0.1$ และค่าเริ่มต้นเป็น 5 (กราฟกลางเส้นหนาสีน้ำเงินเขียว) ที่ดูเหมือนกำลังลดลง แต่อาจต้องการรอบคำนวนเพิ่ม. ภาพในส่วนที่สองและที่สาม แสดง ค่าฟังก์ชันจุดประยงค์ดั้งเดิม (เส้นบางสีฟ้า) ค่าฟังก์ชันจุดประยงค์ที่ถูกลงโทษ (เส้นหนาสีส้ม) ค่าเริ่มต้นของตัวแปร ตัดสินใจ (กาบบาทสีดำ) และค่าสุดท้ายของการคำนวน (จุดกลมสีแดง) พื้นที่สีเทาแสดงบริเวณที่ไม่มีค่าของข้อจำกัด คำตอบได้ ที่อยู่ในบริเวณนี้ถือว่าใช้ไม่ได้ โดยภาพในส่วนที่สอง แสดงผลเมื่อค่าเริ่มต้นเป็น -5 และภาพในส่วนที่สาม แสดงผลเมื่อ ค่าเริ่มต้นเป็น 5 .

แบบฝึกหัด 2.26

ทฤษฎีบทカラูชคุนทัคเกอร์ (Karush-Kuhn-Tucker theorem คำย่อ KKT)¹⁶ กล่าวถึงเงื่อนไข สำหรับค่าทำให้น้อยที่สุดของปัญหา

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

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

ทฤษฎีบทカラูชคุนทัคเกอร์ กล่าวว่า หากกำหนดให้ฟังก์ชัน f , \mathbf{h} , และ \mathbf{g} เป็นฟังก์ชันที่สามารถ หาอนุพันธ์ได้อย่างต่อเนื่อง (*continuously differentiable*) ซึ่งระบุด้วยสัญกรณ์ $f, \mathbf{h}, \mathbf{g} \in \mathcal{C}^1$ และ

¹⁶ เนื้อหาในส่วนนี้ เรียบเรียงจาก [39].

กำหนดให้ \mathbf{x}^* เป็นจุดปกติ และเป็นค่าทำให้น้อยที่สุดท้องถิ่นของปัญหา $\min f(\mathbf{x})$ s.t. $\mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ และจะต้องมี $\boldsymbol{\alpha} \in \mathbb{R}^m$ และ $\boldsymbol{\beta} \in \mathbb{R}^p$ โดยที่

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

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

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

การประยุกต์ใช้ทฤษฎีบทكارูชคุณทักษะ ก็จะใช้เงื่อนไขทั้งสามนี้ ประกอบกับอีกสองเงื่อนไขข้อ จำกัดเดิม ได้แก่ $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ และ $\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}$ เพื่อค้นค่า \mathbf{x} 's ต่าง ๆ ที่มีโอกาสเป็นค่าทำให้น้อยที่สุด \mathbf{x}^* .

หมายเหตุ จุดปกติ (regular point) หมายถึง ค่า \mathbf{x}^* ที่สอดคล้องกับข้อ จำกัดทั้งหมด และ มีเกรเดียนต์ของข้อ จำกัดที่ทำงานเป็นอิสระเชิงเส้นกัน. นั่นคือ ค่า \mathbf{x}^* จะเป็นจุดปกติ เมื่อ เงื่อนไข ดังเดิม $h_1(\mathbf{x}^*) = 0, \dots, h_m(\mathbf{x}^*) = 0$ และเงื่อนไขดังเดิม $g_1(\mathbf{x}^*) \leq 0, \dots, g_p(\mathbf{x}^*) \leq 0$ และ เกรเดียนต์เวกเตอร์ $\nabla h_i(\mathbf{x}^*), \nabla g_j(\mathbf{x}^*)$ สำหรับ $i = 1, \dots, m$ และ $j \in J(\mathbf{x}^*)$ เป็นอิสระเชิงเส้น ต่อกัน โดย เชตของดัชนีข้อ จำกัดที่ทำงาน $J(\mathbf{x}^*) \equiv \{j : g_j(\mathbf{x}^*) = 0\}$.

ข้อ จำกัดแบบภาวะไม่เท่ากัน $g_j(\mathbf{x}) \leq 0$ จะเรียกว่า ทำงาน (active) ที่ \mathbf{x}^* ถ้า $g_j(\mathbf{x}^*) = 0$ และข้อ จำกัด จะเรียกว่า ไม่ทำงาน (inactive) ที่ \mathbf{x}^* ถ้า $g_j(\mathbf{x}^*) < 0$.

ความหมายของทฤษฎีบทكارูชคุณทักษะ ก็คือ ด้วยเงื่อนไข $\beta_j \geq 0$ และข้อ จำกัด $g_j(\mathbf{x}^*) \leq 0$ ทำให้เงื่อนไข $\boldsymbol{\beta}^T \mathbf{g}(\mathbf{x}^*) = \beta_1 g_1(\mathbf{x}^*) + \dots + \beta_p g_p(\mathbf{x}^*)$ สามารถอนุมานได้ว่า ถ้า $g_j(\mathbf{x}^*) < 0$ และ $\beta_j = 0$ แต่ถ้า $g_j(\mathbf{x}^*) = 0$ และ β_j อาจจะมีค่าเป็นบวกก็ได้ หรือเป็นศูนย์ก็ได้.

จงวิเคราะห์ค่า \mathbf{x}^* ด้วยเงื่อนไขจากทฤษฎีบทكارูชคุณทักษะ 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 $f(\mathbf{x}) = 1.5(x_1 + c_1)^2 + 1.5(x_2 + c_2)^2$ 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 เพื่อแสดงผลเช่นรูปที่ 2.30.

ตัวอย่างการตรวจสอบเงื่อนไขการณ์ทักษะ พิจารณาสถานการณ์ 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 $\beta_2 \leq 0$. เงื่อนไขสอง $\nabla f(\mathbf{x}) + \beta_1 \nabla g_1(\mathbf{x}) + \beta_2 \nabla g_2(\mathbf{x}) = 0$. นั่นคือ

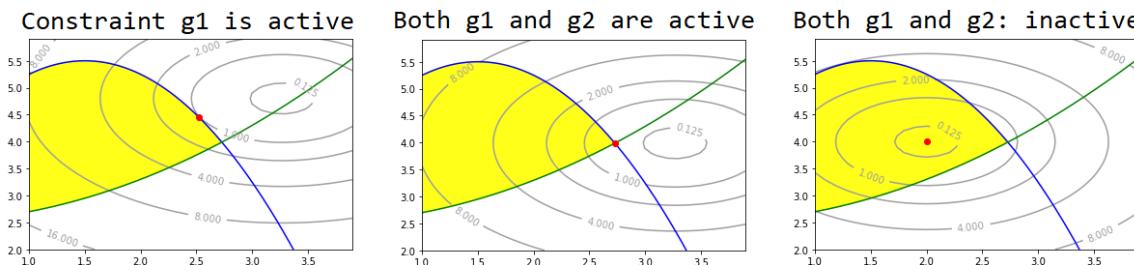
$$\begin{bmatrix} 3(x_1 - 3.27) + 2\beta_1(x_1 - 1.5) + 0.4\beta_2 x_1 \\ 3(x_2 - 4.8) + \beta_1 - \beta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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 β_1 และ β_2 เข้าไปในเงื่อนไขที่สองแล้วจะได้ว่า $\mathbf{x}_a = [3.27, 4.8]^T$ แต่เมื่อตรวจสอบเงื่อนไขจำกัด $g_1(\mathbf{x}_a) = 2.4329$ ซึ่งเมิดข้อจำกัด $g_1(\mathbf{x}) \leq 0$. ดังนั้น \mathbf{x}_a ไม่ใช่คำตอบ.

กรณีที่สอง (b) ถ้า $\beta_1 = 0$. เมื่อแทนค่า β_1 เข้าไปในเงื่อนไขที่สองและเงื่อนไขที่สาม จะได้สามสมการ ซึ่งสามารถแก้สมการเพื่อหาค่า x_1, x_2, β_2 ออกมายได้ หลังจากแก้สมการแล้ว จะได้ $\mathbf{x}_b = [3.348, 4.742]^T$ และ $\beta_2 = -0.175$ ซึ่งค่า $\beta_2 < 0$ ละเมิดเงื่อนไขแรก. ดังนั้น \mathbf{x}_b ไม่ใช่คำตอบ.

กรณีที่สาม (c) ถ้า $\beta_2 = 0$. เมื่อแทนค่า β_2 เข้าไปในเงื่อนไขที่สองและเงื่อนไขที่สาม จะได้สามสมการ ซึ่งสามารถแก้สมการเพื่อหาค่า x_1, x_2, β_1 ออกมายได้ หลังจากแก้สมการแล้ว จะได้ $\mathbf{x}_c = [2.529, 4.440]^T$ และ $\beta_1 = 1.079$ ซึ่งค่า $\beta_1 > 0$ แสดงคล้องกับเงื่อนไขแรก และเมื่อตรวจสอบ $g_1(\mathbf{x}_c) = 0$ และ $g_2(\mathbf{x}_c) = -0.661$ ซึ่งก็แสดงคล้องกับข้อจำกัด $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. ดังนั้น \mathbf{x}_c สามารถเป็นคำตอบได้.



รูปที่ 2.30 ตัวอย่างปัญหาค่าน้อยที่สุดแบบมีข้อจำกัดกรณีต่าง ๆ (เงื่อนไขครุชคุนทักเกอร์). 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 $\mathbf{x} \in \mathbb{R}^2$ แสดงด้วยแกนนอนแทน x_1 และแกนตั้งแทน x_2 . สำหรับแต่ละภาพ ค่าฟังก์ชันจุดประสงค์ f แสดงด้วยวัดภาพคอนทัวร์ (เส้นระดับสีเทา). เส้นสีน้ำเงิน แสดงขอบเขตของข้อจำกัด g_1 (เส้นแทน $g_1(\mathbf{x}) = 0$). บริเวณค่าที่แสดงคล้องกับข้อจำกัด g_1 อยู่ด้านล่างของเส้นสีน้ำเงิน. เส้นสีเขียว แสดงขอบเขตของข้อจำกัด g_2 (เส้นแทน $g_2(\mathbf{x}) = 0$). บริเวณค่าที่แสดงคล้องกับข้อจำกัด g_2 อยู่ด้านบนของเส้นสีเขียว. พื้นที่แรเงาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดของ g_1 และ g_2). จุดสีแดงคือ \mathbf{x}^* ที่ถูกต้องสำหรับแต่ละกรณี.

หมายเหตุ การแก้สมการด้วยมือ เป็นการฝึกทักษะที่ดี. แต่อย่างไรก็ตาม ไฟรอนมีเครื่องมือที่สามารถในการช่วยแก้สมการลักษณะแบบนี้. 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,
```

$$3*(x_2 - 4.8) - \text{beta2}, \\ \text{beta2}*(-x_2 + 0.2*x_1^2 + 2.5)])$$

ในทางปฏิบัติ การแก้ปัญหาแบบมีข้อจำกัด อาจสะอาดกว่าที่จะเลือกใช้วิธีการลงโทษ แต่ทฤษฎีบท かるูชคุณทั้งสอง ช่วยให้ความเข้าใจเกี่ยวกับค่าตอบของปัญหา ซึ่งในหลาย ๆ กรณี ได้นำไปสู่วิธีการแก้ ปัญหาที่มีประสิทธิภาพมาก. (หัวข้อ 4.2 อธิบายแบบจำลองจำแนกค่าทวิภาค ที่การพัฒนาใช้ประโยชน์ จากการทฤษฎีบท かるูชคุณทั้งสอง)

แบบฝึกหัด 2.27

จงเขียนโปรแกรม เพื่อแก้ปัญหาในแบบฝึกหัด 2.26 โดยใช้วิธีการลงโทษ.

คำใบ้ วิธีการลงโทษต้องการฟังก์ชันลงโทษ. ตัวอย่างเช่น ฟังก์ชันลงโทษ $P_1(\mathbf{x})$ สำหรับข้อจำกัด $g_1(\mathbf{x}) \leq 0$ อาจกำหนดเป็น $P_1(\mathbf{x}) = \delta(g_1(\mathbf{x})) \cdot g_1(\mathbf{x})$ เมื่อ δ 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 $(1 - \delta(-g_1))$. การใช้กลไก ลักษณะนี้จะให้ค่าเป็นหนึ่ง (ลงโทษ) เมื่อ $g1$ มากกว่าศูนย์ และให้ค่าเป็นศูนย์ (ไม่มีการลงโทษ) เมื่อ $g1$ น้อยกว่าหรือเท่ากับศูนย์. เงื่อนไขที่ขอบ (ที่ $g1$ เท่ากับศูนย์) จะไม่มีเกรดีเรียนต์. เมื่อเปรียบเทียบ กับ ($g1 \geq 0$) ซึ่งเทียบเท่า $\delta(g_1)$ ผลลัพธ์อาจต่างกันเพียงเล็กน้อย แต่การรวมเงื่อนไขขอบที่ ถูกต้องอาจช่วยให้การทำงานของวิธีลงเกรดีเรียนต์มีเสถียรภาพมากขึ้น.

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

แบบฝึกหัด 2.28

หลาย ๆ สถานการณ์พบว่า ปัญหาการหาค่าดีที่สุดแบบมีเงื่อนไข จะมีคู่ปัญหาของมัน. และใน กรณีนี้ ปัญหาการหาค่าดีที่สุดแบบมีเงื่อนไขดังเดิม จะเรียกว่า **ปัญหาปฐม (primal problem)** ส่วน ปัญหาที่เป็นคู่ของปัญหาปฐม จะเรียกว่า **ปัญหาคู่ (dual problem)**.

สำหรับตัวอย่างของภาวะคู่กัน (duality) พิจารณาปัญหาเชิงเส้นที่เขียนในรูป

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

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].

จากวิธีลAGRANGE (Lagrange method ศึกษาได้จาก [39]) ปัญหาเชิงเส้นแบบมีข้อจำกัดข้างต้น สามารถเขียนในรูปปัญหาที่ไม่มีข้อจำกัดได้เป็น

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

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 $\mathbf{Ax} < \mathbf{b}$ จะทำให้พจน์ $-\boldsymbol{\lambda}_1^T (\mathbf{Ax} - \mathbf{b})$ มีค่าเป็นบวก และเมื่อประกอบกับกลไกของลAGRANGE พารามิเตอร์ $\boldsymbol{\lambda}_1 \geq \mathbf{0}$ ที่หาก $\boldsymbol{\lambda}_1$ มีขนาดใหญ่มากพอ จะทำให้ค่าจุดประสงค์รวมมากขึ้น และส่งผลต่อเนื่องทำให้การค้นหาค่า \mathbf{x} จะต้องปรับค่า \mathbf{x} และส่งผลเป็นการแก้ไขการลดเมิดดังกล่าว.

วิธีของลAGRANGE จะต้องเลือกลAGRANGE PARALLEL ให้เหมาะสม นั่นคือมีค่าใหญ่มากพอที่ข้อจำกัดจะไม่ถูกลดเมิด. แต่การเลือกลAGRANGE PARALLEL ที่มีค่าใหญ่มากเกินไป จะไปขัดขวางการค้นหาค่าที่ดีที่สุด (ผลลัพธ์ที่ได้ จะไม่ลดเมิดข้อจำกัด แต่จะไม่ใช้ค่าที่ดีที่สุดที่เป็นไปได้¹⁷). ดังนั้น การเลือกขนาดของลAGRANGE PARALLEL ที่ใหญ่ที่สุด ที่จะไม่ทำร้ายจุดประสงค์เดิมของปัญหาปัญหา.

เพื่อความสะดวก กำหนดให้ฟังก์ชันจุดประสงค์รวม

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

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

ดังนั้นเกรเดียนต์

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

¹⁷วิธีลAGRANGE จะต่างจากวิธีการลงโทษ โดยวิธีลAGRANGE ใช้ลAGRANGE PARALLEL และต้องเลือกค่าให้เหมาะสม. แต่วิธีการลงโทษ ใช้ฟังก์ชันการลงโทษ ซึ่งจะลงโทษเฉพาะตอนที่ลดเมิดข้อจำกัด ดังนั้นการเลือกค่าน้ำหนักในการลงโทษจึงผ่อนคลายกว่า. นั่นคือ สำหรับวิธีการลงโทษ เพียงเลือกค่าน้ำหนักให้มีค่าใหญ่มากพอเท่านั้น ไม่ต้องห่วงว่ามากเกินไปจะไปรบกวนฟังก์ชันจุดประสงค์หลัก. แต่ความสะดวกนี้ ก็จะแลกมาด้วยการเลือกใช้ฟังก์ชันลงโทษให้เหมาะสม และประสิทธิภาพการทำงาน.

และเมื่อพิจารณา ณ จุดดีที่สุด¹⁸ นั่นคือ ที่ $\nabla_{\mathbf{x}} L = \mathbf{0}$ และแก้สมการจะได้ $\boldsymbol{\lambda}_2^T = \mathbf{c}^T - \boldsymbol{\lambda}_1^T \mathbf{A}$. ดังนั้น ค่าฟังก์ชันจุดประสังค์รวม 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 เป็นค่าฟังก์ชันจุดประสังค์รวม ที่ค่าขึ้นกับ \mathbf{x} แต่ L' เป็นค่าฟังก์ชันจุดประสังค์รวม ที่ได้เลือก \mathbf{x} ให้ดีที่สุดแล้ว และ (2) L' ไม่ใช่ฟังก์ชันของ \mathbf{x} แต่เป็นฟังก์ชันของ $\boldsymbol{\lambda}_1$. การมองจากปัญหาจากมุมมองของ $\boldsymbol{\lambda}_1$ จะทำให้ได้ปัญหาคู่ ซึ่งเขียนได้เป็น

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

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

หากเปรียบเทียบ ปัญหาปฐมเป็นเสมือนการหาค่า \mathbf{x} ที่ทำให้จุดประสังค์เดิมเล็กที่สุด แต่การดำเนินการให้จุดประสังค์เดิม f มีขนาดเล็ก ถูกควบคุมด้วยข้อจำกัดดังเดิมต่าง ๆ. ดังนั้น จุดประสังค์เดิมจะเล็กได้เท่าที่ข้อจำกัดเดิมอนุญาต. ในขณะที่ปัญหาคู่ มองจากอีกด้านของมุม มองจากจุดที่ปรับ \mathbf{x} ได้ดีที่สุดแล้ว แต่ต้องการคุณไม่ให้ลดเมิดข้อจำกัด. ปัญหาคู่ จึงเสมือนการหาค่า $\boldsymbol{\lambda}_1$ ที่ทำให้จุดประสังค์รวม L' (ซึ่งรวมข้อจำกัดเดิม และปรับ \mathbf{x} ดีที่สุดแล้ว) มีค่ามากที่สุด เพื่อรักษาข้อจำกัดเดิมต่าง ๆ ไว้ แต่การดำเนินการให้ L' ใหญ่ ถูกควบคุมไม่ให้มากเกินไปจนรบกวนจุดประสังค์ดังเดิม.

เมื่อแก้ปัญหาคู่เสร็จ คำตอบจะได้ $\boldsymbol{\lambda}_1^*$ และทำให้สามารถคำนวณ $\boldsymbol{\lambda}_2^* = \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda}_1^*$. ด้วยความเชื่อมโยงและทฤษฎีบทカラชคุนท์เกอร์ 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 $\lambda_{1i} > 0$ และ $\lambda_{2i} < 0$ สำหรับ i^{th} ทำงาน นั่นคือ $\mathbf{A}_{i,:} \cdot \mathbf{x} = b_i$. ถ้า $\lambda_{2i} > 0$ แปลว่า $x_i = 0$. ค่าของ \mathbf{x}^* สามารถวิเคราะห์ได้จากสมการที่ได้เหล่านี้.

ตัวอย่างปัญหาเชิงเส้นข้างล่าง

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

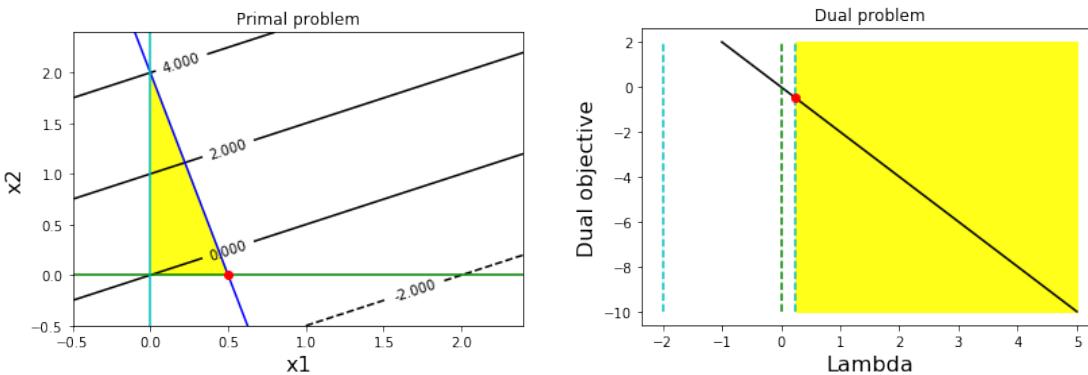
ซึ่งอยู่ในปฐมรูป (primal form). เปรียบเทียบกับแก้ปัญหา

¹⁸ การวิเคราะห์นี้เทียบเท่าทฤษฎีบทカラชคุนท์เกอร์ โดยเฉพาะเงื่อนไขที่สอง.

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

ซึ่งเป็นรูปคู่' (dual form) ของปัญหาข้างต้น.

เมื่อแก้ปัญหาคู่เสร็จ 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 $\mathbf{x}^* = [0.5, 0]^T$. รูปที่ 2.31 แสดงภาพของภาวะคู่กันในตัวอย่างนี้.



รูปที่ 2.31 ตัวอย่างภาวะคู่กัน. ภาพช้าย แสดงปัญหาปฐม (ปัญหาค่าน้อยที่สุด) ด้วยค่าฟังก์ชันจุดประสังค์ในปริภูมิของตัวแปร \mathbf{x} . ค่าฟังก์ชันจุดประสังค์ $f(\mathbf{x}) = 2x_2 - x_1$ แสดงด้วยเส้นภาพคอนทัวร์. 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 $x_1 = 0$). พื้นที่เรางาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดทั้งสาม). จุดสีแดงคือ \mathbf{x}^* . ภาพขวา แสดงปัญหาคู่ (ปัญหาค่ามากที่สุด) ด้วยแกนตั้งเป็นค่าฟังก์ชันจุดประสังค์ของปัญหาคู่ และแกนนอนแสดงค่า λ . เส้นสีดำที่บ คือ ค่าฟังก์ชันจุดประสังค์ของปัญหาคู่ $L'(\lambda_1) = -2\lambda_1$. เส้นประ แสดงขอบเขตของข้อจำกัด $\lambda_1 \geq 0$ (เส้นสีเขียว) และข้อจำกัด $-4\lambda_1 \leq -1 \equiv \lambda_1 \geq 0.25$ และ $-\lambda_1 \leq 2 \equiv \lambda_1 \leq 2$ (ทั้งคู่แสดงด้วยเส้นสีฟ้าเขียว). พื้นที่เรางาสีเหลือง แสดงบริเวณค่าที่ยอมรับได้ (ผ่านข้อจำกัดทั้งสาม). จุดสีแดง (ในทั้งสองภาพ) แทนค่าตอบที่ถูกต้อง. นั่นคือ ปัญหาปฐม $x_1^* = 0.5$, $x_2^* = 0$ และปัญหาคู่ $\lambda_1^* = 0.25$.

จากปัญหาเชิงเส้นข้างล่าง จงแปลงเป็นรูปคู่ แก้ปัญหาทั้งในรูปปฐม และรูปคู่ และตรวจสอบค่าตอบ.

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

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

การคำนวณเชิงเลข

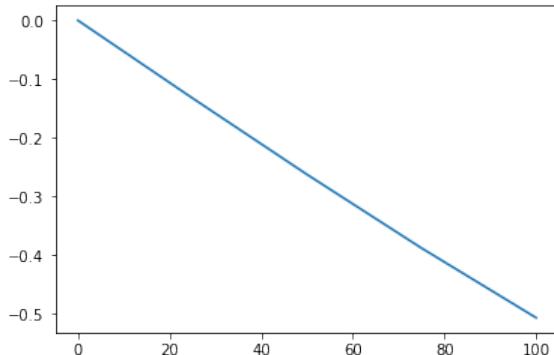
การเขียนโปรแกรมคำนวณเชิงเลข มีปัจจัยด้านข้อจำกัดที่ต้องคำนึงถึง. แบบฝึกหัดต่อไปนี้ แนะนำบางประเด็นที่ควรคำนึงถึง เวลาคำนวณทางคณิตศาสตร์มาเขียนโปรแกรม.

แบบฝึกหัด 2.29

โปรแกรมข้างล่างนี้ใช้วาดรูปที่ 2.32.

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

จงวิเคราะห์และอธิบายว่า ทำไม่รูปที่ได้ไม่เห็นเป็นรูปโค้งขึ้นลง เช่น รูปของค่าฟังก์ชันไซน์ที่คุณเคย



รูปที่ 2.32 ผลการวาดกราฟฟังก์ชันไซน์ ที่ดูต่างจากความคาดหวัง. ผลจากคำสั่ง `plt.plot(x, np.sin(x))` จากแบบฝึกหัด 2.29.

แบบฝึกหัด 2.30

จากโปรแกรมและผลการรันดังแสดงข้างล่างนี้ จงอภิปรายว่าทำไม่มี 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]

ทำไมจึงมีผลบางค่าที่เป็น **False** ทั้ง ๆ ที่ $\frac{x}{7} \cdot 7$ มีค่าเท่ากับ x ? จวิเคราะห์และอธิบายผลของ $x == 7*(x/7)$ กับ $x == 7*x/7$ ประกอบพร้อมอภิปรายประเด็นที่ได้เรียนรู้นี้ กับสถานการณ์ที่อาจจะเกิดขึ้น.

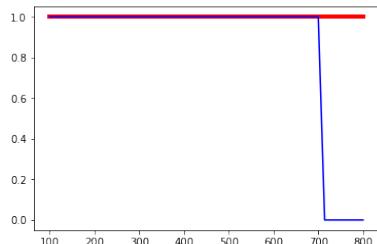
แบบฝึกหัด 2.31

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 ซึ่งทั้ง x/x และ $x/\log(\exp(x))$ ่มีค่าเท่ากับ 1 เมื่อ $x > 0$. ดังนั้น ผลลัพธ์น่าจะเห็นเส้นตรงแนวโนนที่ค่าหนึ่ง เท่าเดิมคงที่ตลอดช่วง. แต่ผลที่ได้เป็นดังแสดงในรูปที่ 2.33. จสืบกรณีนี้ อธิบายสิ่งที่เกิดขึ้น และอภิปรายผลที่อาจเกิดขึ้นในทางปฏิบัติ จากประเด็นที่ได้เรียนรู้.

คำใบ้ ตรวจดูค่า `np.exp(x)` ที่ค่า x ต่าง ๆ และตรวจสอบเรื่อง IEEE754 จากอินเตอร์เน็ต.



รูปที่ 2.33 ผลจากการวาดกราฟ ซึ่งกราฟที่ได้ดูแลกจากที่คาด. ผลจากการวาดกราฟ x/x (เส้นสีแดง) 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)
```

จงทดสอบฟังก์ชันนี้ ด้วยค่า v ต่าง ๆ เช่น `softmax(np.array([1, 2, 5]))` (ลองผสมค่าหลาย ๆ แบบ ทั้งค่าบวก ค่าลบ และศูนย์) อภิปรายการพฤติกรรมของฟังก์ชันซอฟต์แมกซ์.

แล้วลองทดสอบอีกครั้งด้วยค่าขนาดใหญ่ เช่น `softmax(np.array([1000, 2000, 5000]))` สังเกตผลลัพธ์ที่ได้ อภิปรายถึงปัญหาและสาเหตุ พร้อมเสนอวิธีแก้ปัญหา.

คำใบ้ ปัญหาอยู่ที่ไหน วิธีแก้อาจใช้คณิตศาสตร์ไปช่วยบรรเทาสาเหตุ. (หัวข้อ 3.7 อภิปรายวิธีเขียนโปรแกรม ฟังก์ชันซอฟต์แมกซ์ที่ทนทาน (*robust*) สำหรับการใช้งานในทางปฏิบัติ.)

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 การปรับเส้นโค้งด้วยฟังก์ชันพหุนาม ในหัวข้อ 3.1 อภิรายตัวอย่างง่าย ๆ ที่เป็นแนวทางหลัก และส่วนท่อนหลักการที่สำคัญของการเรียนรู้ของเครื่อง. หัวข้อ 3.2 อภิรายพื้นฐาน หลักการและประเด็นสำคัญของศาสตร์การเรียนรู้ของเครื่อง. หัวข้อ 3.3 อภิรายโครงข่ายประสาทเทียม ซึ่งเป็นแบบจำลองที่สำคัญ ใช้งานได้กว้างขวาง และเป็นหนึ่งในศาสตร์และศิลป์ของการเรียนรู้ของเครื่อง. หัวข้อ 3.4 อภิรายการประยุกต์ใช้งานของโครงข่ายประสาทเทียม. หัวข้อ 3.5 อภิรายคำแนะนำทั้งสำหรับการใช้งานโครงข่ายประสาทเทียม และการใช้งานการเรียนรู้ของเครื่องโดยทั่วไป.

3.1 การปรับเส้นโค้งด้วยฟังก์ชันพหุนาม

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

พิจารณากรณีที่ทั้งอินพุตและเอาต์พุตเป็นมิติเดียว นั่นคือ คำถาม $x \in \mathbb{R}$ และ $y \in \mathbb{R}$. หากต้องการจะคำนวณค่า y ที่สัมพันธ์กับค่า x โดยที่มีตัวอย่างข้อมูลเป็นคู่ ๆ ของ (x, y) ได้แก่ (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) ทั้งหมดจำนวน n คู่. ตัวแปรต้น x เป็นค่าที่ถูกมาเพื่อหา y ที่เป็นตัวแปรตาม หรือค่าที่อยากรู้คำตอบไป. แต่ละคู่ (x_i, y_i) อาจเรียกว่า จุดข้อมูล (datapoint).

รูปที่ 3.1 แสดงตัวอย่างจุดข้อมูล 10 จุด. ตำแหน่งของแต่ละจุดข้อมูลในภาพ ระบุจากค่า x ตามแกนนอน และค่า y ตามแกนตั้ง. ตัวแปรต้น x อาจเรียก อินพุต หรือข้อมูลนำเข้า (input) และตัวแปร

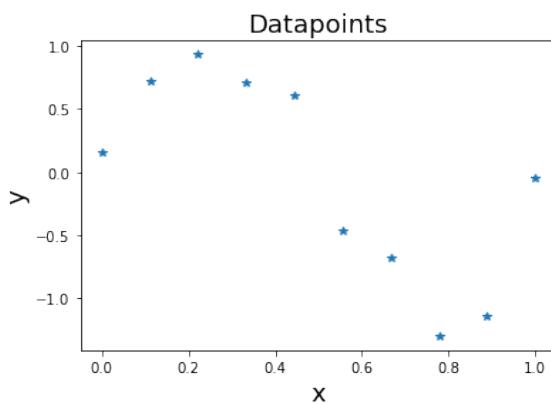
ตาม y อาจเรียก เอ้าต์พุต หรือข้อมูลนำออก (output). จากตัวอย่างในภาพ จุดข้อมูลแรกสุด มีค่า $x = 0$ ค่า $y = 0.16$.

เป้าหมายของตัวอย่างนี้คือ การคำนวณค่าประมาณเอ้าต์พุต y ของค่าอินพุต x ที่ส่งสัญญาโดยอินพุต x อาจจะเป็นค่าเดิม หรืออาจจะเป็นค่าใหม่ที่ไม่เคยเห็นมาก่อน. แนวทางคือ การใช้แบบจำลอง ซึ่งเป็นการคำนวณทางคณิตศาสตร์ ที่เป็นฟังก์ชันของตัวแปร x และใช้ค่าที่ฟังก์ชันคำนวณได้ ทายเป็นค่า y . แบบจำลองที่จะเลือกใช้สำหรับตัวอย่างนี้ 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. ก่อนที่จะสามารถนำฟังก์ชันพหุนามไปใช้คำนวณค่า y จากค่า x ที่ถูกกำหนด ต้องกำหนดระดับขั้น m และค่าของพารามิเตอร์ \mathbf{w} ให้เรียบร้อยก่อน.

ตัวอย่างเช่น หากเลือก ระดับขั้น $m = 2$ และค่าของพารามิเตอร์ $\mathbf{w} = [0.7, -0.65, 1]^T$ สำหรับแบบจำลอง f_1 นั้นคือ ฟังก์ชันพหุนาม $y = f_1(x) = 0.7 - 0.65x + x^2$ และที่ $x = 0.5$ 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 $y = f_2(x) = 13x - 39x^2 + 32x^3$ และที่ $x = 0.5$ จะคำนวณค่า y เป็น 0.75 ซึ่งแตกต่างจากผลคำนวณจาก f_1 .

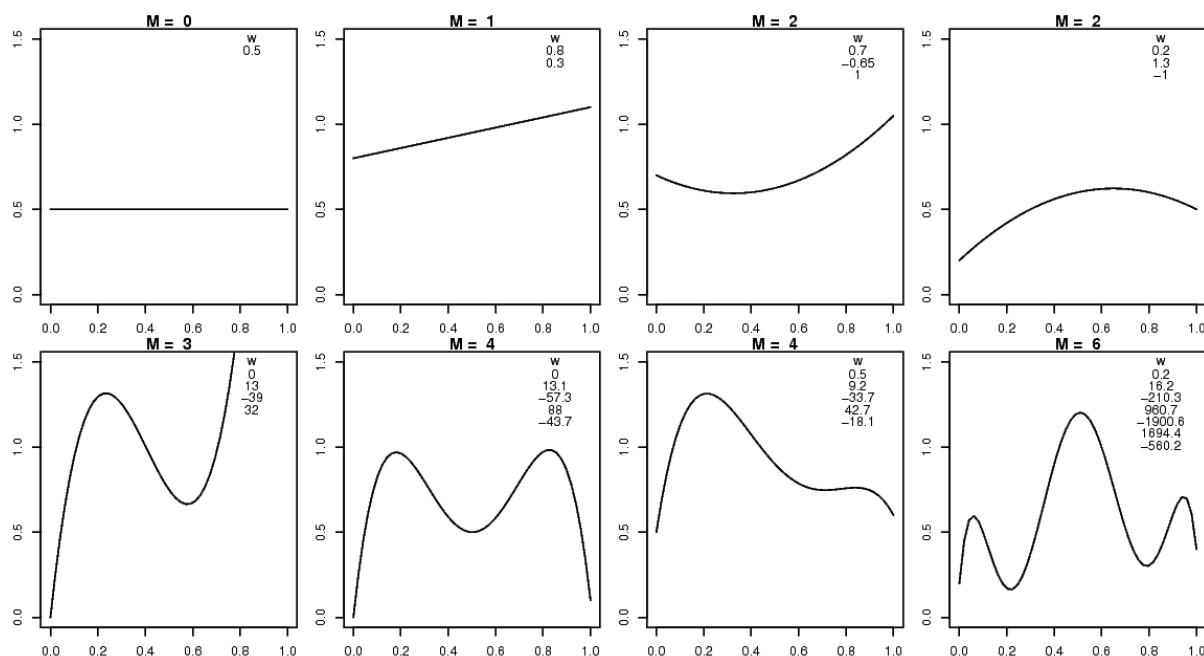


รูปที่ 3.1 ตัวอย่างจุดข้อมูล $(0, 0.160), (0.111, 0.724), (0.222, 0.931), (0.333, 0.712), (0.444, 0.610), (0.556, -0.460), (0.667, -0.684), (0.778, -1.299), (0.889, -1.147)$, และ $(1, -0.045)$ รวม 10 จุด.

ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ จะให้ผลการทำนายต่างกัน. รูปที่ 3.2 แสดงพฤติกรรมการทำนาย ของฟังก์ชันพหุนาม เมื่อเลือกใช้ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ. พฤติกรรมการทำนาย หมายถึง ความสัมพันธ์ระหว่างอินพุตกับเอ้าต์พุต. การปรับเปลี่ยนค่า ใช้ประโยชน์จากการที่ พฤติกรรม

การทำนายเปลี่ยนตามระดับขั้นและค่าพารามิเตอร์. ดังนั้น การปรับเลี้นโค้ง ทำได้โดยปรับระดับขั้น และค่าพารามิเตอร์ของสมการ เพื่อปรับเส้นโค้งให้ได้ผลการทำนายที่ดีขึ้น.

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



รูปที่ 3.2 ฟังก์ชันพหุนาม เมื่อเลือกใช้ระดับขั้นและค่าพารามิเตอร์ต่าง ๆ. ระดับขั้น M แสดงเหนือแต่ละภาพ และค่าพารามิเตอร์ แสดงภายใต้แต่ละภาพ.

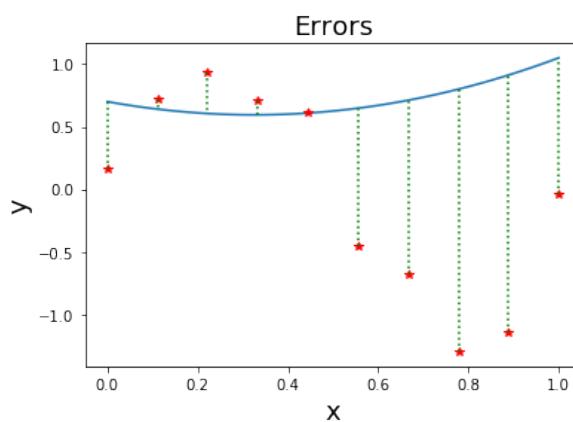
การฝึกแบบจำลอง. การฝึกแบบจำลอง คือการทำค่าพารามิเตอร์ของแบบจำลอง เพื่อให้แบบจำลองทำนายได้ถูกต้องมากที่สุด หรือกล่าวอีกอย่างคือ เพื่อให้แบบจำลองทำนายผิดน้อยที่สุด. การวัดว่า แบบจำลองทำนายได้ผิดมากน้อยเท่าใด สามารถใช้แนวทางของวิธีกำลังสองน้อยที่สุด (least square method) ได้. วิธีกำลังสองน้อยที่สุด วัดว่าแบบจำลองทำนายได้ผิดมากน้อยเท่าใด จาก ผลต่างกำลังสอง ระหว่างค่าเอาร์พูตที่ทำนายกับค่าเอาร์พูตริง. นั่นคือ $E_n = (\hat{y}_n - y_n)^2$ เมื่อ E_n คือความผิดพลาด (error) ของการทำนายจุดข้อมูลที่ n^{th} ซึ่งวัดจากผลต่างกำลังสองระหว่าง ค่าเอาร์พูตที่ทำนาย \hat{y}_n สำหรับจุดข้อมูลที่ n^{th} และค่าเอาร์พูตริง y_n ของจุดข้อมูลที่ n^{th} . ค่า y_n ที่ได้จากข้อมูล อาจเรียกว่า **เอาร์พูตจริง (ground truth)** หรือ **ค่าเฉลย**. รูปที่ 3.3 แสดงผลต่างระหว่างค่าที่ทำนายและค่า

ເຄາຕີພຸດຈິງ. 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).

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



รูปที่ 3.3 ความผิดพลาดของการทำนายที่จุดข้อมูลต่าง ๆ. จุดดาวสีแดง แสดงจุดข้อมูล. เส้นทึบสีฟ้า แทนพอดีกรรมการทำนาย แบบจำลอง. ความผิดพลาดวัดจากที่ค่า x เดียวกัน ค่าที่แบบจำลองทำนายห่างจากค่า y ของจุดข้อมูลเท่าไร ในภาพ เน้นความห่างนี้ด้วยเส้นปรสีเขียว.

ตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง. สำหรับตัวอย่างข้อมูลในรูปที่ 3.1 สมมติระดับขั้นที่เลือกคือ ระดับขั้นหนึ่ง ($m = 1$) นั่นคือ $\hat{y} = w_0 + w_1x$. ในการฝึกแบบจำลอง ซึ่งคือการหา 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 $E_n = (w_0 + w_1x_n - y_n)^2$.

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

$$\frac{\partial \frac{1}{2} \sum_{n=1}^N \{w_0 + w_1x_n - y_n\}^2}{\partial w_0} = 0, \quad (3.3)$$

$$\frac{\partial \frac{1}{2} \sum_{n=1}^N \{w_0 + w_1x_n - y_n\}^2}{\partial w_1} = 0 \quad (3.4)$$

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

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

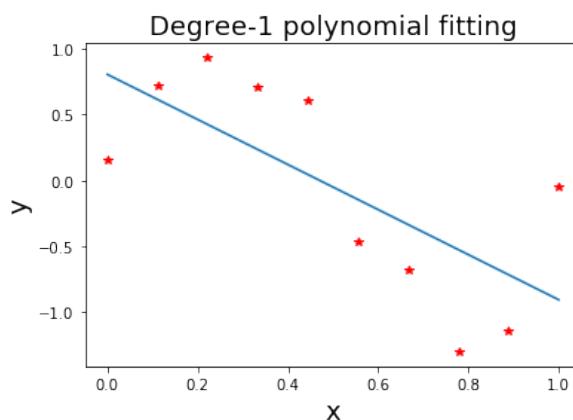
$$w_0 \sum_{n=1}^N \{x_n\} + w_1 \sum_{n=1}^N \{x_n^2\} - \sum_{n=1}^N \{y_n \cdot x_n\} = 0 \quad (3.8)$$

ซึ่งเมื่อจัดรูปสมการ 3.7 และ 3.8 ให้อยู่ในรูปเมทริกซ์จะได้

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

เมื่อนำค่าจุดข้อมูลในรูปที่ 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 $\sum_{n=1}^N y_n x_n = -1.992$. เมื่อแก้สมการแล้วจะได้ $[w_0, w_1]^T = [0.805, -1.710]^T$. นั่นคือ แบบจำลอง $\hat{y} = 0.805 - 1.71x$ พฤติกรรมของแบบจำลองนี้แสดงดังในรูปที่ 3.4.

การใช้งาน หรือการทำนายด้วยแบบจำลอง $\hat{y} = 0.805 - 1.71x$ คือ การคำนวณโดยแทนค่า x ที่สามลงไป 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 พฤติกรรมของแบบจำลองพื้นฐานระดับชั้นหนึ่งที่ฝึกแล้ว (เส้นทึบสีฟ้า) กับจุดข้อมูลที่ฝึก (ดาวสีแดง).

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

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

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

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

สมองส่วนล่างนี้รวมส่วนบนของไขสันหลัง ก้านสมอง (*brain stem*) และ เชเรเบลัม (*cerebellum*). สมองส่วนล่างจะควบคุมการทำงานที่เป็นพื้นฐานของการดำเนินชีพ เช่น การหายใจ และการเต้นของหัวใจ. เชเรเบลัมช่วยประสานงานเรื่องการเคลื่อนไหวและการเรียนรู้ของการเคลื่อนไหวที่เกิดจากการฝึกทำซ้ำ เช่น การเล่นเปียโนหรือการตีลูกเทนนิส จะอาศัยการทำงานของเชเรเบลัมช่วย.

สมองส่วนกลางอยู่ด้านบนของก้านสมอง ทำหน้าที่เกี่ยวกับการควบคุมการตอบสนองแบบฉับพลัน และเป็นส่วนหนึ่งในระบบการควบคุมการเคลื่อนไหวของดวงตาและการเคลื่อนไหวโดยสมัครใจอีน ๆ. สมองส่วนกลางนี้มีส่วนที่ทำงานประมวลผลภาพอยู่ด้วย. สภาพเห็นทั้งหมด (*blindsight*) เป็นสภาพของผู้พิการทางสายตา ที่การพิการเกิดจากส่วนประมวลผลภาพหลักที่เปลือกสมองส่วนการเห็น (*visual cortex* ซึ่งจัดอยู่ในสมองส่วนบน) ไม่สามารถทำหน้าที่ได้ แต่ดวงตาและส่วนอื่น ๆ ในระบบการมองเห็น รวมถึงส่วนประมวลผลภาพของสมองส่วนกลางยังดีอยู่. สภาพเห็นนี้ ตัวผู้พิการจะไม่รับรู้ถึงการมองเห็น แต่เมื่อมีการทดลอง โดยบังคับให้ผู้มีสภาพเห็นทั้งหมดบรรยายรูปร่างหรือตำแหน่งของวัตถุด้วยการเดา ผู้มีสภาพเห็นทั้งหมดจะบรรยายได้ถูกต้องทั้งรูปร่าง ตำแหน่ง และการเคลื่อนไหว ซึ่งความถูกต้องแม่นยำที่ได้สูงมากเกินกว่าที่จะได้มาจากการคาดเดา. คำอธิบายสภาพนี้ก็คือ สมองกลับไปใช้ผลการประมวลภาพจากสมองส่วนกลาง ซึ่งแม้จะไม่มีความสามารถในการประมวลผลได้ดีเท่ากับเปลือกสมองส่วนการเห็น แต่ก็ช่วยให้เกิดการมองเห็นได้จิตสำนึกนี้เกิดขึ้นได้.

เนื่องจากระบบประมวลผลภาพในสมองมีทั้งที่สมองส่วนกลางและบริเวณเปลือกสมองส่วนการเห็นในสมองส่วนบน ทฤษฎีวิวัฒนาการเชื่อว่า การประมวลภาพที่สมองส่วนกลางเป็นวิวัฒนาการในช่วงก่อน (สัตว์หลายชนิด เช่น กบ) ใช้การประมวลภาพที่สมองส่วนกลางเป็นหลัก) และเปลือกสมองส่วนการเห็นเป็นวิวัฒนาการในช่วงต่อมา. ผู้เชี่ยวชาญด้านประสาทวิทยาเดวิด ลินเดน (ผู้เชี่ยวนหนังสือ *Accidental Mind*[121]) ได้อธิบายเพิ่มเติมในการสนทนาส่วนตัวว่า หากการประมวลผลภาพของสมองส่วนกลางเสียหาย แต่ส่วนอื่น ๆ ในระบบการมองเห็นยังดีอยู่ รวมถึงเปลือกสมองส่วนการเห็นก็ยังดีอยู่ ผู้ป่วยจะรับรู้ถึงการมองเห็นได้ แต่พบว่าผู้ป่วยจะมีการตอบสนองการประสาทงานระหว่างมือและตา (*hand-eye coordination*) ที่ช้าลงอย่างชัดเจน.

สมองส่วนบนเป็นส่วนที่ใหญ่ที่สุดในสมองส่วน. สมองส่วนบนประกอบด้วยเชเรบรัม (*cerebrum*) และส่วนสมองใน (*the inner brain*). หมายเหตุ เชเรบรัม (ของสมองส่วนบน) มาจากภาษาลาติน แปลตรงตัวว่า สมอง ขณะที่ เชเรเบลัม (ของสมองส่วนล่าง) มาจากภาษาลาติน ซึ่งแปลตรงตัวว่า สมองน้อย. เชเรบรัมคือภาพของสมองที่คนทั่วไปจะนึกถึงเมื่อกล่าวถึงสมอง. เชเรบรัม ทำหน้าที่หลักในการรับรู้ ความจำ การวางแผน การคิด การจินตนาการ รวมถึงศีลธรรม นิสัย และบุคคลิกภาพ. เมื่อมองจากด้านบน เชเรบรัมดูเหมือนจะแบ่งได้เป็นซีกซ้ายและซีกขวา โดยมีดูเหมือนมีร่องแบ่งสมองสองซีกนี้ออกจากกัน. สมองทั้งสองซีกเชื่อมต่อกันผ่านเส้นใยประสาทเรียกว่า คอร์ปัส คาโลซัม (*corpus callosum*). สมอง

ทั้งสองซีกนี้ทำงานร่วมกัน แต่สมองซีกซ้ายจะควบคุมการทำงานของร่างกายซีกขวา และสมองซีกขวาจะควบคุมการทำงานของร่างกายซีกซ้าย โดยสมองซีกซ้ายจะเด่นด้านการทำงานเกี่ยวกับภาษา การวิเคราะห์รายละเอียด และทักษะเชิงรูปธรรม ในขณะที่สมองซีกขวาจะเด่นด้านการอ่านภาพรวม และทักษะเชิงนามธรรม. การทำงานไขว้ระหว่างซีกสมองกับร่างกายนั้น แม้จะยังไม่มีคำอธิบายว่าเหตุใดกลไกของร่างกายจึงเป็นเช่นนั้น แต่ข้อเท็จจริงคือสัญญาณจากสมองซีกหนึ่งจะไปบังคับร่างกายอีกซีกหนึ่ง ดังนั้น หากสมองซีกหนึ่งเสียหาย ร่างกายอีกซีกหนึ่งจะได้รับผลกระทบ เช่น ผู้ป่วยโรคหลอดเลือดสมองเมื่อเกิดสมองซีกขวาเสียหาย จะส่งผลให้ผู้ป่วยเป็นอัมพาตในซีกซ้ายของร่างกาย.

การศึกษาที่น่าสนใจเกี่ยวกับสมองซีกซ้ายและขวา หล่ายกรณีได้มาจากการศึกษาผู้ป่วยโรคลมชักรุนแรง ที่แพทย์ต้องตัดคดอร์ปัส คาโลซัมเพื่อลดความรุนแรงของอาการลมซักไม่ให้แพร่ขยายข้ามซีกสมองได้. หนึ่งในตัวอย่าง[204] คือ การศึกษาที่นำผู้ป่วยที่ผ่านการตัดการเชื่อมต่อระหว่างสมองซีกซ้ายและขวาออกจากกัน มาใส่ค้อนแทกเลนส์พิเศษเพื่อแยกการมองเห็นระหว่างตาซ้ายและตาขวาออกจากกัน. ตาซ้ายและร่างการศึกษาเชื่อมโยงกับสมองซีกขวา ตาขวาและร่างการศึกษาเชื่อมโยงกับสมองซีกซ้าย. เมื่อให้ตาขวารับภาพของเท้าของໄก์ และให้ตาซ้ายรับภาพของบ้านที่ถูกหินะท่ำวม พ้อมสั่งให้ผู้ทดลองซึ่งเลือกภาพที่เกี่ยวข้องด้วยมือซ้ายและขวา ผู้ทดลองซึ่งมือขวาไปที่ภาพตัวแม่ໄก์ และซึ่มือซ้ายไปที่ภาพพลวั่ง ผู้ทดลอง อธิบายถึงเท้าໄก์ได้ แต่ไม่สามารถอธิบายภาพของบ้านที่ถูกหินะท่ำวมได้ และเมื่อให้ผู้ทดลองอธิบายเหตุผลที่ซึ่งเลือกภาพแม่ໄก์ และพลวั่ง สมองส่วนซ้าย ซึ่งไม่ได้รับรู้ภาพของบ้านที่ถูกหินะท่ำวม ก็พยายามอธิบายไปว่า เท้าໄก์เกี่ยวข้องกับแม่ໄก์ และพลวั่งเกี่ยวข้องคือเป็นเครื่องมือตักมูลໄก์. กรณีนี้ ผู้เขียนชี้ว่าสมองซีกซ้ายซึ่งมีความสามารถทางภาษา แต่ไม่ได้รับภาพที่สมองซีกขวาเห็น ไม่ได้รับรู้ถึงภาพบ้านหินะท่ำวม แต่สมองซีกขวา เมนจ์รับรู้ภาพของบ้านหินะท่ำวมและยังบังคับมือซ้ายไปซึ่งพลวั่ง ซึ่งเป็นสิ่งที่มักจะเชื่อมโยงกับภาพหินะท่ำวม ในกลุ่มคนที่คุ้นเคยกับสภาพหินะท่ำวม แต่สมองซีกขวาไม่มีความสามารถทางภาษา จึงไม่สามารถอธิบายออกมาเป็นคำพูดได้.

เซเรบรัมแต่ละชีกยังสามารถแบ่งเป็นส่วนย่อย ๆ ลงไปได้อีก ซึ่งแต่ละส่วนของเซเรบรัมมักจะเรียกว่ากลีบ(lobe). เซเรบรัมมีกลีบหลัก ๆ เช่น กลีบทหน้า (frontal lobe), กลีบข้าง (parietal lobe), กลีบท้ายทอย (occipital lobe), และกลีบมัน (temporal lobe). สมองกลีบทหน้าจะอยู่บริเวณหลังหัวผากของเราระและทำหน้าที่เกี่ยวกับ การวางแผน การจินตนาการ ถึงอนาคต การใช้เหตุผล การควบคุมดัวเอง บุคคลิกภาพ และ ศีลธรรม. ลีบเข้าไปท้าย ๆ กลีบจะเป็นบริเวณที่ทำหน้าที่เกี่ยวกับการควบคุมการเคลื่อนไหว. ในกลีบทหน้าของสมองชีกซ้ายจะมีบริเวณโบร์ก้า (Broca's area) ซึ่งเป็นส่วนที่ทำหน้าที่เกี่ยวกับการใช้ภาษา.

สมองกลีบข้างซึ่งอยู่ถัดจากกลีบหน้าเข้ามา (บริเวณใต้กลังกระหม่อม) ทำหน้าที่เกี่ยวกับรสนิลิน สัมผัส รวมถึงการรับรู้การเคลื่อนไหวของร่างกาย ความสามารถในการอ่านหนังสือและการคิดคำนวณตัวเลข ก็เกี่ยวข้องกับสมองกลีบข้าง. สมองกลีบท้ายทอยอยู่ถัดจากกลีบข้างไปทางหน้าหลัง (บริเวณท้ายทอย) ทำหน้าที่หลักเกี่ยวกับการมองเห็น. เป็นส่วนของส่วนการเห็น ซึ่งเป็นส่วนประมวลผลการมองเห็นหลักก็อยู่ในบริเวณกลีบท้ายทอย. สมองกลีบขมับจะอยู่ใต้กลีบหน้าและกลีบข้าง ซึ่งมีเม็ดเทียบกับภายนอกแล้วจะอยู่บริเวณขมับ. สมองกลีบขมับทำหน้าที่หลักเกี่ยวกับการประมวลผลเสียงต่าง ๆ และมีหน้าที่ช่วยในการรวมความจำและความรับรู้ต่าง ๆ ทั้งภาพ เสียง กลิ่น และ สัมผัส เข้าด้วยกัน.

ที่ผิวนอกของเชrebัรัมจะเป็นชั้นของเนื้อยื่อที่หนาประมาณ 2 ถึง 4 มิลลิเมตร ซึ่งเรียกว่า เชrebอลคอร์เท็กซ์ (cerebral cortex). การประมวลผลของสมองส่วนใหญ่ใช้กันว่าเกิดขึ้นภายในเนื้อยื่อส่วนนี้ เนื้อยื่อส่วนนี้จะมีสีเข้มกว่า เนื้อยื่อส่วนด้านใน และมักถูกอ้างถึงในชื่อของเนื้อเทา (gray matter) เปรียบเทียบกับเนื้อขาว (white matter) ซึ่งอยู่ภายนอกใน เนื้อเทาจะประกอบด้วยเซลล์ประสาท หลอดเลือดฟอย และ เซลล์เกลีย. เซลล์ประสาทในเนื้อเทาจะมีไขมันที่เป็นฉนวนน้อยกว่าเซลล์ประสาทในเนื้อขาว จึงทำให้สีของเนื้อยื่อโดยรวมดูเข้มกว่า. (ดูรายละเอียดของเซลล์ประสาท ในเกร็ดความรู้เซลล์ประสาท.) เนื่องจากเชrebอลคอร์เท็กซ์เป็นผิวของสมอง รอยหยักของสมองจะช่วยเพิ่มพื้นที่ผิวและปริมาณของเนื้อเทาซึ่งสัมพันธ์กับปริมาณของข้อมูลที่สมองสามารถประมวลผลได้.

ส่วนสมองในเป็นอีกบริเวณในสมองส่วนบน ส่วนสมองในนี้จะเชื่อมต่อไปสันหลังเข้ากับเรเบรัม ส่วนสมองในทำหน้าที่เกี่ยวกับอารมณ์ มีส่วนในการเปลี่ยนแปลงการรับรู้และการตอบสนองไปตามสถานะของอารมณ์ในขณะนั้น ๆ มีส่วนช่วย

เริ่มการเคลื่อนไหวต่าง ๆ ที่เราทำโดยเราไม่ต้องคิดถึงการเคลื่อนไหวเหล่านั้น และมีส่วนสำคัญในกระบวนการสร้างความจำ ส่วนประกอบต่าง ๆ ของส่วนสมองในนี้จะมีเป็นคู่ ๆ ทางซ้ายและขวา โดยมีส่วนประกอบที่สำคัญ เช่น ไฮป์โรลาแมส (hypothalamus) รารามัส (thalamus) bachol แกงเกลีย (basal ganglia) อะมิกดาลา (amygdala) และ hippocampus (hippocampus). ไฮป์โรลาแมสเป็นสมองศูนย์กลางการจัดการอารมณ์ รามัสช่วยจัดการข้อมูลที่ผ่านไปมาระหว่างเซเรบรมและไขสันหลัง bachol แกงเกลียช่วยการเริ่มและประสานงานการเคลื่อนไหวต่าง ๆ โรคพาร์กินสันซึ่งผู้ป่วยจะมีอาการที่เด่นชัดคือมีปัญหาในการเคลื่อนไหว เช่น อาการสั่น เดินหรือเคลื่อนไหวได้ช้า เป็นโรคที่เกี่ยวพันกับเซลล์ประสาทที่เสื่อมต่อกับbachol แกงเกลียนนี้ อะมิกดาลาทำหน้าที่เกี่ยวกับอารมณ์ ความกลัว ความก้าวร้าว และ ความจำที่เกี่ยวข้องกับอารมณ์ความรู้สึก มีงานศึกษาที่พบความเกี่ยวข้องกันระหว่างขนาดของอะมิกดาลาของบุคคลกับความสัมพันธ์ทางสังคมของบุคคลนั้น อีปีโปแคมปัสทำหน้าที่จัดส่งความจำใหม่ไปเก็บในตำแหน่งที่เหมาะสมในเซเรบรม และค้นหาความจำที่ต้องการจากเซเรบรม ผู้ป่วยที่สูญเสียอีปีโปแคมปัสจะสูญเสียความสามารถในการสร้างความทรงจำใหม่

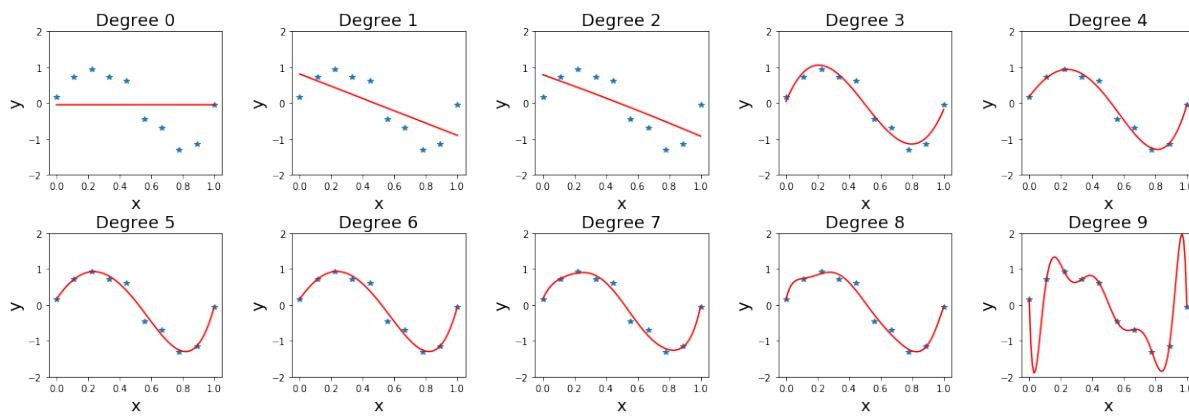
3.2 คุณสมบัติความทั่วไปและการเลือกแบบจำลอง

หัวข้อ 3.1 แสดงตัวอย่างของการปรับเส้นโค้ง ด้วยฟังก์ชันพหุนามระดับขั้นหนึ่ง ระดับขั้นของฟังก์ชันพหุนาม เป็นอภิมานพารามิเตอร์ ของแบบจำลองพหุนาม การสร้างแบบจำลองสามารถใช้ระดับขั้นได้ ก็ได้ แต่การเลือกระดับขั้นที่เหมาะสม เพื่อได้แบบจำลองทำงานยที่ดี จะได้อภิรายในหัวข้อนี้ รูปที่ 3.5 แสดงพฤติกรรมการทำงานที่ระดับขั้นต่าง ๆ.

จากรูปที่ 3.5 สังเกตว่า ระดับขั้นที่สูงขึ้นช่วยให้แบบจำลองยืดหยุ่นมากขึ้น และสามารถปรับตัวเข้าหากุศลข้อมูลได้ง่ายขึ้น และที่ระดับขั้นสูงมาก ๆ เช่น ที่ระดับขั้นเก้า ฟังก์ชันพหุนามสามารถปรับเข้าหากุศลข้อมูลได้ใกล้มาก ๆ แต่ที่ระดับขั้นเก้า พฤติกรรมทำงาน ระหว่างจุดข้อมูลมีการเปลี่ยนแปลงรุนแรงมาก.

การสร้างแบบจำลองทำงาน ต้องการแบบจำลองทำงานที่มีคุณสมบัติความทั่วไป คุณสมบัติที่แบบจำลองสามารถทำงานได้ดี แม้กับข้อมูลที่ไม่เคยเห็นมาก่อน จะเรียกว่า **คุณสมบัติความทั่วไป (generalization)**. ปกติแล้ว ข้อมูลจะมีสัญญาณrgb ที่ประกอบเข้ามาด้วย แบบจำลองที่ดีควรจะจับสารสนเทศที่สำคัญของข้อมูล เมื่อแบบจำลองปรับตัวเข้ากับข้อมูลที่ใช้ฝึก มากเกินไป แบบจำลองอาจจะจับสัญญาณrgb เข้าไปปนกับสารสนเทศที่สำคัญ ซึ่งจะส่งผลให้แบบจำลองสามารถทำงานข้อมูลที่ใช้ฝึกได้อย่างแม่นยำ แต่อ้าจะไม่สามารถทำงานข้อมูลใหม่ได้ดี.

ตัวอย่างข้อมูลที่แสดงนี้ จริง ๆ แล้วสร้างมาจากความสัมพันธ์ $y = \sin(2\pi x) + \epsilon$ เมื่อสัญญาณrgb ϵ สุ่มขึ้นมากจากการแจกแจงเกาส์เซียน ที่มีค่าเฉลี่ยเป็น 0 และค่าเบี่ยงเบนมาตรฐานเป็น 0.3. นั่นคือ $\epsilon \sim \mathcal{N}(0, 0.3)$. รูปที่ 3.6 แสดงพฤติกรรมการทำงาน เปรียบเทียบกับสารสนเทศที่สำคัญของข้อมูล (แสดงด้วยเส้นประสีดำ). แบบจำลองที่ดี คือแบบจำลองที่สามารถประมาณสารสนเทศที่สำคัญของข้อมูลได้ แต่ในทางปฏิบัติ การสร้างแบบจำลอง ไม่ได้รู้สารสนเทศที่สำคัญ เพราะหากว่า ก็สามารถสร้างแบบจำลอง จากสารสนเทศที่สำคัญที่รู้นั้นได้โดยตรง. ดังนั้น การใช้งานแบบจำลอง



รูปที่ 3.5 พฤติกรรมการทำนายของแบบจำลองพหุนามที่ระดับขั้นต่าง ๆ. เส้นกราฟสีแดง แสดงผลการทำนายของพังก์ชันพหุนาม. สัญลักษณ์ดาวสีฟ้า แทนจุดข้อมูล.

การทำนายในทางปฏิบัติ ต้องการกลไก หรือกระบวนการที่จะวนตรวจสอบว่า แบบจำลองยังมีคุณสมบัติความทว่าไปดีอยู่หรือไม่.

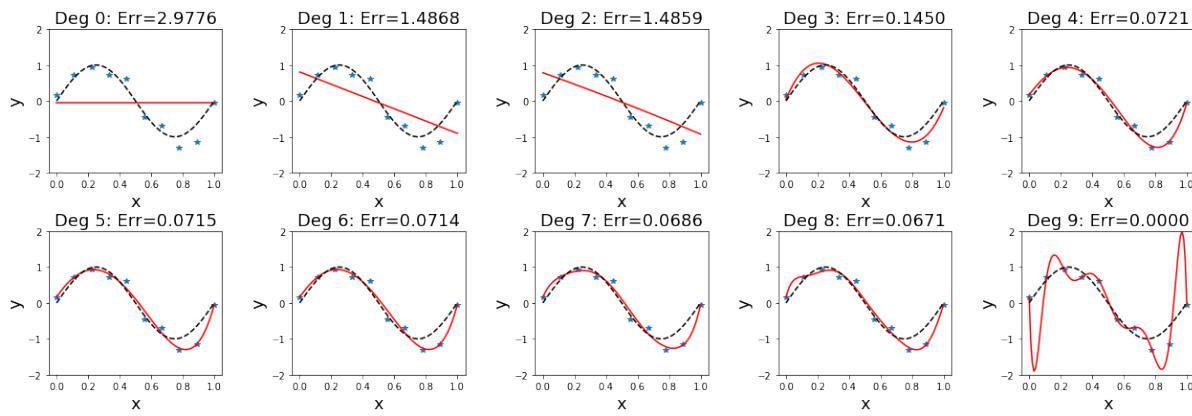
กระบวนการที่ใช้ตรวจสอบ คุณสมบัติความทว่าไป ที่ตรงมาตรงไปที่สุด คือการใช้ข้อมูลทดสอบ. **ข้อมูลทดสอบ (test data)** คือข้อมูลอีกชุด ที่ไม่ได้ถูกใช้ในการกระบวนการฝึก เป็นข้อมูลที่แบบจำลองไม่เคยเห็นเลย จะใช้เพื่อทดสอบแบบจำลองเท่านั้น. เพื่อให้สามารถจำแนกได้ชัดเจน ว่ากำลังพูดถึงข้อมูลสำหรับจุดประสงค์ใดอยู่ ข้อมูลที่ใช้ฝึกแบบจำลอง จะเรียกว่า **ข้อมูลฝึก (training data)**.

จากตัวอย่างการปรับเส้นโค้งข้างต้น สมมติมีข้อมูลทดสอบ 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. รูปที่ 3.7 ภาพ ก แสดงจุดข้อมูลฝึก (10 จุด) และจุดข้อมูลทดสอบ (5 จุด). ภาพ ข แสดงค่าผิดพลาดของแบบจำลองที่ระดับขั้นต่าง ๆ เมื่อทดสอบกับชุดข้อมูลฝึก และชุดข้อมูลทดสอบ.

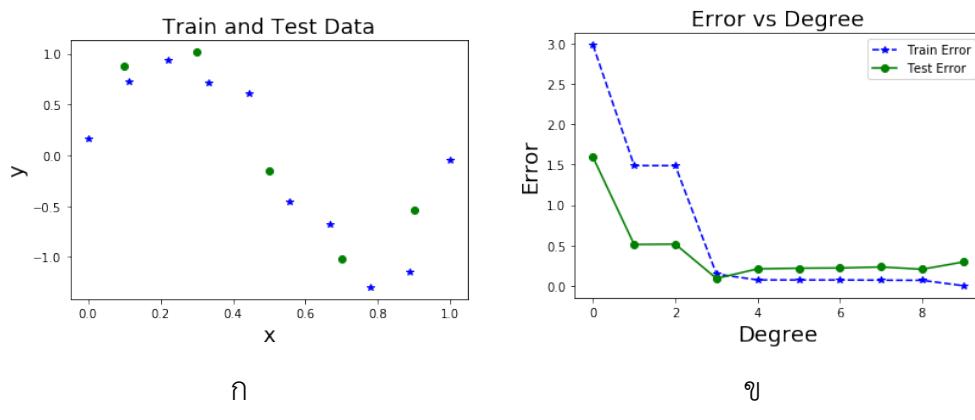
จาก ภาพ ข สังเกตว่า ค่าผิดพลาดเมื่อทดสอบกับชุดฝึก ซึ่งค่านี้มักเรียกว่า **ค่าผิดพลาดชุดฝึก (training error)** มีค่าลดลงเรื่อย ๆ เมื่อระดับขั้นเพิ่มขึ้น. แต่ค่าผิดพลาดเมื่อทดสอบกับชุดทดสอบ ซึ่งค่านี้มักเรียกว่า **ค่าผิดพลาดชุดทดสอบ (test error)** มีค่าลดลงจนต่ำสุดที่ ในตัวอย่างนี้ เป็นระดับขั้นสาม และค่ากลับเพิ่มขึ้น เมื่อระดับขั้นเพิ่มขึ้นหลังจากนั้น. ณ จุดที่ค่าผิดพลาดชุดฝึกต่ำลง แต่ค่าผิดพลาดชุดทดสอบกลับสูงขึ้น เป็นสัญญาณบ่งชี้ว่า แบบจำลองเริ่มเสียคุณสมบัติความทว่าไป ซึ่งมักเรียกว่า แบบจำลองเกิดการโอเวอร์ฟิต (overfitting).

ผลจากค่าผิดพลาดชุดทดสอบ ปั้งชี้ว่าแบบจำลองพหุนามระดับขั้นสี่ขึ้นไป เริ่มเกิดโอเวอร์ฟิต และแบบจำลองที่มีคุณสมบัติความทว่าไปดีที่สุดในการทดสอบนี้ คือ แบบจำลองพหุนามระดับขั้นสาม. หากสังเกต รูปที่ 3.6 จะเห็นว่า ระดับขั้นสามให้การประมาณธรรมชาติจริงของข้อมูลดีที่สุด (พฤติกรรมของแบบจำลอง เส้นทึบแดง มีลักษณะใกล้เคียงกับธรรมชาติจริง เส้นประสีดำ มากกว่าระดับขั้นอื่น ๆ).

การโอเวอร์ฟิตของแบบจำลอง สัมพันธ์โดยตรงกับข้อมูล โดยเฉพาะกับจำนวนจุดข้อมูล. พังก์ชันพหุนามที่มีระดับขั้นสูง เรียกว่า เป็นแบบจำลองที่มี **ความซับซ้อน (complexity)** สูง. แบบจำ-



รูปที่ 3.6 พฤติกรรมการทำงานของแบบจำลองที่ระดับขั้นต่าง ๆ ที่แสดงความสัมพันธ์จริงด้วยเส้นประสีดีประกอบ. ความสัมพันธ์จริง คือความสัมพันธ์ระหว่าง x และ y ที่ใช้ในกระบวนการสร้างจุดข้อมูลสำหรับตัวอย่าง. กระบวนการสร้างจุดข้อมูลสำหรับตัวอย่างนี้ ใช้ความสัมพันธ์จริง $y = \sin(2\pi x)$ ประกอบกับสัญญาณรบกวน ϵ . นั่นคือ จุดข้อมูลสร้างจาก $y = \sin(2\pi x) + \epsilon$. ค่าผิดพลาด (ระบุด้วยคำว่า Err) เหนือแต่ละค่าคือ ค่าผิดพลาดของแต่ละแบบจำลอง โดยคิดจาก 10 จุดข้อมูลฝึก.



รูปที่ 3.7 ค่าผิดพลาดชุดฝึกกับค่าผิดพลาดชุดทดสอบ. ภาพ ก แสดงข้อมูลฝึก (ดาวสีฟ้า) และข้อมูลทดสอบ (วงกลมสีเขียว). ภาพ ข แสดงค่าผิดพลาดของแบบจำลอง เมื่อทดสอบกับชุดข้อมูลฝึกและชุดข้อมูลทดสอบ.

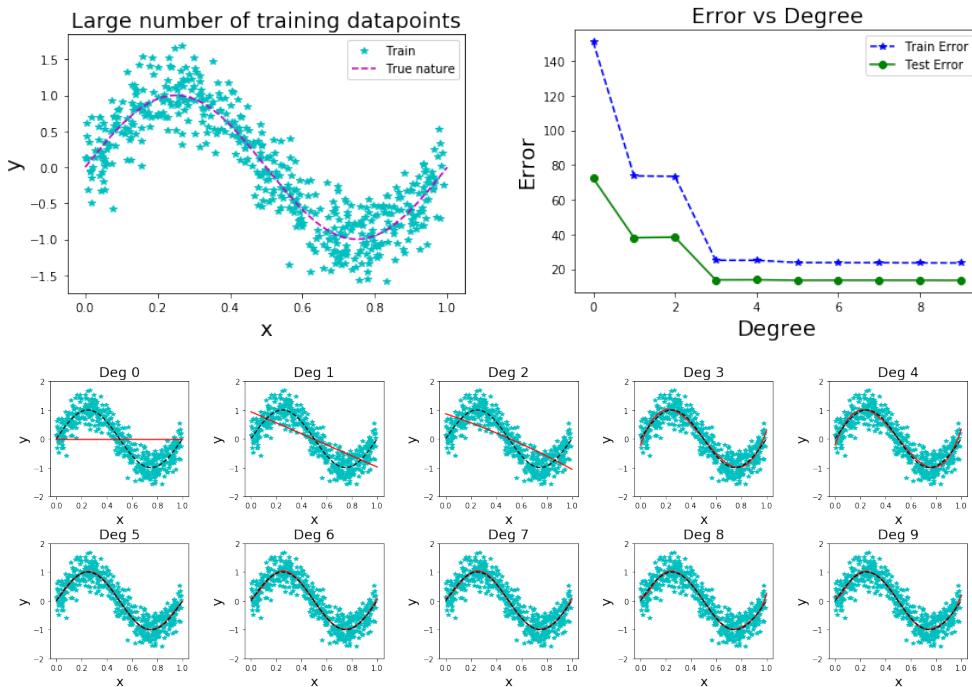
ลองที่มีความซับซ้อนสูง มีความยืดหยุ่นมาก. จำนวนจุดข้อมูลที่มีมากพอ จะช่วยให้เห็นสารสนเทศที่สำคัญ จากสัญญาณรบกวนที่ปนมาได้ชัดเจนขึ้น และช่วยการฝึกแบบจำลองที่มีความซับซ้อนสูง ให้มีคุณสมบัติความทั่วไปดีขึ้นได้.

รูปที่ 3.8 แสดงให้เห็นว่า ข้อมูลจำนวนมาก สามารถช่วยลดปัญหาโอเวอร์ฟิตได้. ภาพบนซ้ายแสดง ค่าผิดพลาดชุดทดสอบลดลงจนถึงค่อนข้างคงที่หลังจากระดับขั้นที่สาม.

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

อย่างไรก็ตาม นอกจากการเพิ่มจำนวนจุดข้อมูล ยังมีกลไกอื่น ๆ อีกที่สามารถช่วยแก้หรือบรรเทา

ปัญหาโอเวอร์ฟิต เมื่อใช้แบบจำลองที่มีความซับซ้อนสูงได้ เช่น การใช้แนวทางเบย์เชียน (Bayesian) หรือ การทำเรกูลาริซ์ชัน.



รูปที่ 3.8 แบบจำลองพหุนามเมื่อใช้จำนวนจุดข้อมูลมากขึ้น. ภาพบนซ้าย แสดงจุดข้อมูลจำนวนมาก สร้างจาก $y = \sin(2\pi x) + \epsilon$ โดยสารสนเทศที่สำคัญวัดด้วยเส้นประสานเย็น. จุดข้อมูล 500 จุดถูกสร้างขึ้นมาเป็นข้อมูลฝึก (จุดสิน้ำเงินเขียว). จุดข้อมูล 250 จุดถูกสร้างขึ้นมาเป็นข้อมูลทดสอบ (ไม่ได้แสดงในภาพ). ภาพบนขวา แสดงค่าผิดพลาดดูฝึก (เส้นประสานสีเขียว) และชุดทดสอบ (เส้นทึบสีเขียว). ด้วยจุดข้อมูลจำนวนมาก ไม่มีสัญญาณบ่งบอกถึงการโอเวอร์ฟิตให้เห็น. ภาพล่าง แสดงพฤติกรรมการทำงานของฟังก์ชันพหุนามที่ระดับขั้นต่าง ๆ (ระบุเหนือภาพ ด้วยคำย่อ เช่น Deg 0 สำหรับ ระดับขั้น 0 หรือ degree 0).

อีกประเดิ่นหนึ่งที่นำเสนอใน สองเกตระดับค่าผิดพลาดที่แสดงในรูปที่ 3.8 เปรียบเทียบกับที่แสดงในรูปที่ 3.7 ระดับค่าที่แสดงในรูปที่ 3.7 (ภาพซ้าย) มีค่าค่อนข้างต่ำ (แกน y สูงสุดประมาณ 3.0) ในขณะที่ ระดับค่าที่แสดงในรูปที่ 3.8 (ภาพซ้ายบน) มีค่าสูงมาก (แกน y สูงสุดเกิน 140). นั่นเป็นเพราะ ทั้งสองภาพวัดค่าผิดพลาดด้วย ผลรวมค่าผิดพลาด $E = 0.5 \sum_n E_n$. โดยปกติแล้ว เมื่อจำนวนจุดข้อมูลมากขึ้น ผลรวมค่าผิดพลาดจะมากขึ้นตามไปด้วย. ในทางปฏิบัติ การประเมินผล มักรายงานผลด้วยค่าความแม่นยำ (accuracy) ซึ่งอาจวัดด้วย ค่าเฉลี่ยความผิดพลาดกำลังสอง (mean square error คำย่อ MSE) หรือ รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (root mean square error คำย่อ RMSE) ที่ให้ผลคงเส้นคงความมากกว่าผลรวมค่าผิดพลาด. ค่าเฉลี่ยความผิดพลาดกำลังสอง คำนวณจาก $MSE = \frac{1}{N} \sum_{n=1}^N E_n$ 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

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N E_n}.$$

การทำเรกูลาริゼชัน. การทำเรกูลาริเซชัน (regularization) เป็นวิธีหนึ่งที่นิยมใช้เพื่อช่วยลดปัญหาการโอเวอร์พิท. แนวทางหนึ่ง คือ การทำค่าน้ำหนักเลื่อน (weight decay) โดย การใส่พจน์เสริมใน การลงให้ λ เป็นพื้นที่น้ำหนักเพิ่มมากขึ้น เพื่อจะถ่วงดูลงมาให้พารามิเตอร์ของแบบจำลองมีค่าให้ต่ำลง. สมการ 3.10 แสดงพื้นที่น้ำหนักเพิ่มมากขึ้น ที่ประกอบด้วยพจน์ค่าผิดพลาดและพจน์ค่าน้ำหนักเลื่อม ซึ่งเป็นพจน์แรกและพจน์ที่สองทางขวามือ ตามลำดับ. นั่นคือ พื้นที่น้ำหนักเพิ่มมากขึ้น

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

เมื่อ $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$ และ พารามิเตอร์ λ ควบคุมสมดุลย์ระหว่างอิทธิพลของค่าผิดพลาดจากการทำงาน และอิทธิพลจากพจน์ค่าน้ำหนักเลื่อม. จากมุมมองของการหาค่าน้อยที่สุด พารามิเตอร์ λ อาจถูกเรียกว่าเป็น ลากرانจ์พารามิเตอร์. บีชอบ[15] ชี้ว่า ป้อยครั้งที่ พจน์ค่าน้ำหนักเลื่อม จะไม่รวม w_0 . หรือ ถ้ามี w_0 ก็อาจจะมีลากرانจ์พารามิเตอร์เฉพาะของตัวเอง.

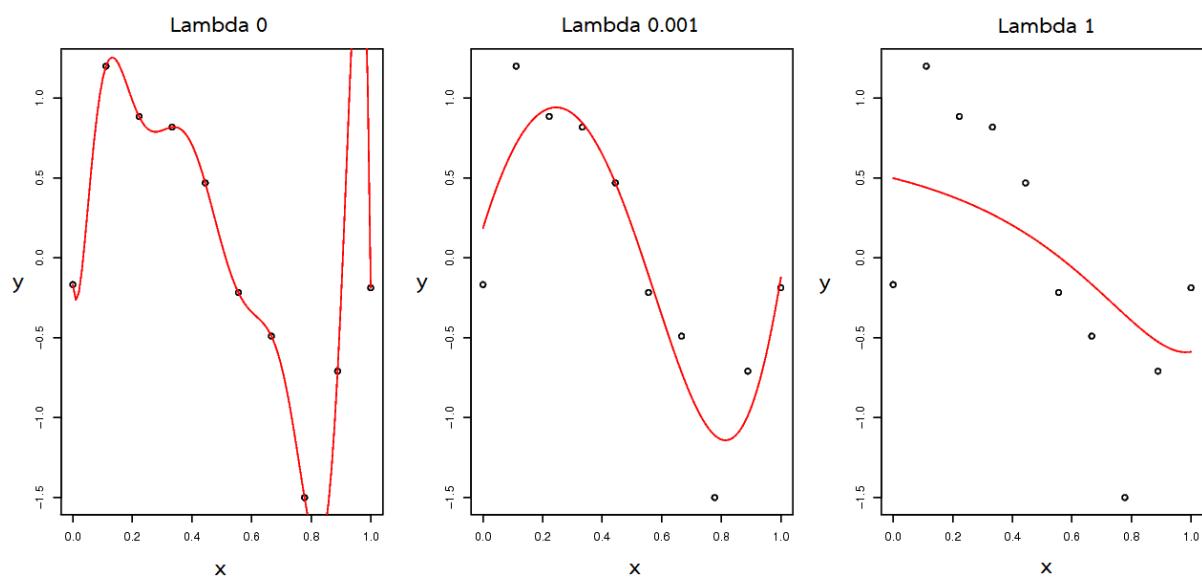
รูปที่ 3.9 แสดงผลจากการทำเรกูลาริเซชันด้วยการใช้ค่าลากرانจ์ต่าง ๆ. ภาพช้ายสุด $\lambda = 0$ เทียบเท่ากับการไม่ได้ใช้วิธีค่าน้ำหนักเลื่อม. การโอเวอร์พิทเห็นได้ชัดในกรณีนี้. ภาพกลาง แสดงค่าลากرانจ์ที่เหมาะสม ค่าลากرانจ์พารามิเตอร์ที่เหมาะสม จะช่วยบังคับแบบจำลองที่มีความซับซ้อนสูง ให้ทำตัวเสมือนมีความซับซ้อนต่ำลง. ค่าประมาณจากแบบจำลอง (แสดงด้วยเส้นทึบสีแดง) มีลักษณะใกล้เคียงกับ $\sin(2\pi x)$ ที่ใช้สร้างจุดข้อมูล. แต่ถ้าหากใช้ค่าลากرانจ์ค่าให้ต่ำลงไป ก็อาจทำให้เกิดการอันตรายพิเศษได้ ดังแสดงในภาพขวาสุด.

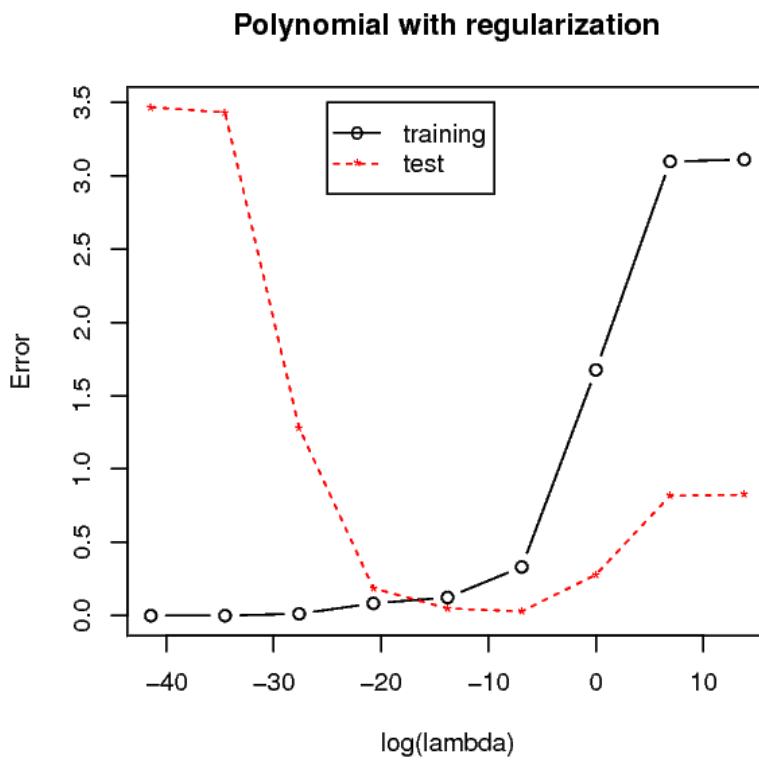
ตารางที่ 3.1 แสดงให้เห็นว่า ถ้าใช้ค่า λ ให้ต่ำลง ก็จะช่วยลดค่าลากرانจ์ต่ำลง แต่ถ้าใช้ค่า λ ให้ต่ำลง ก็จะทำให้ค่าพารามิเตอร์น้อยลง. รูปที่ 3.10 แสดงผลค่าผิดพลาดของแบบจำลองพหุนามระดับขั้นสูง กับการทำค่าน้ำหนักเลื่อมที่ลากرانจ์ค่าต่าง ๆ เมื่อประเมินกับข้อมูลชุดฝึกหัดและชุดทดสอบ. สังเกตว่าค่าผิดพลาดของแบบจำลอง เมื่อประเมินกับชุดทดสอบ ค่าผิดพลาดของแบบจำลองจะน้อยลง เมื่อใช้ค่าลากرانจ์ค่าน้อย ๆ (ให้ผลลัพธ์กับการทำพื้นที่น้ำหนามระดับขั้นสูง). ส่วนเมื่อประเมินกับชุดทดสอบ ค่าผิดพลาดของแบบจำลองจะลดลงต่ำสุดที่ค่าลากرانจ์ราว ๆ 0.001 หรือ $\log(\lambda) \approx -6.91$.

สำหรับการประเมินแบบจำลอง ปัจจัยสำคัญ คือ แบบจำลองสามารถทำงานข้อมูลที่ไม่เคยเห็นมาก่อนได้ดี หรือ แบบจำลองมีคุณสมบัติความทั่วไป. ดังนั้น เพื่อเลือกความซับซ้อนของแบบจำลอง เช่น การเลือกระดับขั้นของพหุนาม หรือการเลือกค่าลากرانจ์ของการทำค่าน้ำหนักเลื่อม จึงควรทำการวัดคุณสมบัติความทั่วไปของแบบจำลองที่ความซับซ้อนต่าง ๆ กัน. วิธีที่ง่ายและตรงไปตรงมาที่สุด ก็คือ การแบ่งข้อมูลออกเป็น 2 ชุด ได้แก่ ข้อมูลชุดฝึก ที่ใช้ฝึกแบบจำลอง นั่นคือใช้ค่าของพารามิเตอร์ \mathbf{w} และชุดตรวจสอบ ที่ใช้เลือกความซับซ้อนของแบบจำลอง เช่น M หรือ λ .

ตารางที่ 3.1 ค่าพารามิเตอร์ของแบบจำลองพหุนาม กับการทำค่าสำนักเสื่อมที่ลากرانจ์ค่าต่าง 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

รูปที่ 3.9 พหุนามระดับขั้นเก้า กับการทำค่าสำนักเสื่อม ด้วยลากرانจ์ค่าต่าง ๆ. ภาพซ้าย แสดงการโอเวอร์ฟิต ($\lambda = 0$). ภาพกลาง และดงแบบจำลองที่เหมาะสม ($\lambda = 0.001$). ภาพขวา แสดงการอันเดอร์ฟิต ($\lambda = 1$). จุดวงกลม คือจุดข้อมูลฝึก และเส้นทึบสีแดง แสดงค่าที่แบบจำลองทำนาย.



รูปที่ 3.10 การทำน้ำหนักเสื่อมด้วยลากرانจ์ค่าต่าง ๆ ประเมินด้วยข้อมูลชุดฝึกหัด (เส้นทึบสีดำ) กับ ชุดทดสอบ (เส้นประสีแดง). แกนตั้ง แสดงค่าผิดพลาด. แกนนอน แสดงค่าของ $\log \lambda$.

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

ครอสвалиเดชั่น. ถ้าข้อมูลมีจำนวนมาก การแบ่งบางส่วนของข้อมูลมาเป็นชุดตรวจสอบนั้น ไม่ได้ดูว่ามีปัญหาอะไร แต่หากข้อมูลมีปริมาณจำกัด ควรจะจัดการสถานการณ์อย่างไร เมื่อการฝึกแบบจำลองให้ดีต้องการข้อมูลจำนวนมาก แต่คุณภาพของการตรวจสอบเลือกความซับซ้อน และการทดสอบ ก็ต้องการข้อมูลจำนวนมากเช่นกัน. การแบ่งส่วนข้อมูลที่มีปริมาณน้อยอยู่แล้ว ยิ่งจะทำให้แต่ละส่วนมีปริมาณน้อยลงไปอีก. วิธีหนึ่งที่ออกแบบมาเพื่อบรรเทาปัญหานี้ คือ การทำครอสвалиเดชั่น (cross-validation). แนวคิดคือ การสุ่มและใช้ผลเฉลี่ย โดยทำการฝึกแบบจำลองและการตรวจสอบหลาย ๆ ครั้ง แต่ละครั้ง แบ่งข้อมูลต่าง ๆ กันไป แล้วเอานำผลลัพธ์ที่ได้มาเฉลี่ยกัน เพื่อสรุปแบบจำลองที่มีคุณสมบัติความทั่วไปดีที่สุด. การดำเนินการ จะแบ่งข้อมูลออกเป็น K ส่วน แต่ละครั้งจะเลือกส่วนหนึ่งมาเป็นชุดตรวจสอบ และใช้ส่วนที่เหลือ ($K - 1$ ส่วน) สำหรับฝึกแบบจำลอง. เนื่องจาก การแบ่งข้อมูลเป็น K ส่วน วิธีนี้ มักถูกเรียกว่า วิธีครอสвалиเดชั่น K พับ (K-fold cross-validation).

วิธีครอสвалиเดชั่น K พับทำการฝึกและตรวจสอบ K ครั้ง ที่แต่ละครั้งจะเลือกส่วนที่ทำการตรวจ



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

สอบแต่กต่างกัน. เมื่อทำจนครบทุกส่วนแล้ว จึงนำผลประเมินจากแต่ละครั้ง รวม K ค่า มาหาค่าเฉลี่ย เป็นค่าประเมินกรอสวัลเดชั้น (*cross-validation evaluation*) ของแบบจำลอง. ค่าประเมินกรอสวัลเดชั้นนี้ สามารถใช้เปรียบเทียบกับแบบจำลองอื่น (หรือแบบจำลองเดียวกันแต่ความซับซ้อนอื่น) เพื่อหาแบบจำลอง(หรือความซับซ้อน)ที่ดีที่สุด.

รูปที่ 3.11 แสดงแผนภาพการแบ่งข้อมูลสำหรับวิธีกรอสวัลเดชั้น 5 พับ ($K = 5$) และการจัดสรรข้อมูลสำหรับการฝึก และการตรวจสอบในแต่ละครั้ง. การฝึกและตรวจสอบแต่ละครั้ง จะเรียกว่า วัลเดชั้นรัน (*validation run*). ในภาพแสดง 5 วัลเดชั้นรัน ที่รันแรก (Run 1) ฝึกแบบจำลองด้วยข้อมูล 4 ส่วนแรก และนำแบบจำลองที่ฝึกแล้ว ไปตรวจสอบกับข้อมูลส่วนหลังสุด (แรงงานสีเข้มในรูป). รันที่สอง ฝึกแบบจำลองด้วยข้อมูลส่วนอื่นยกเว้นส่วนที่ 4 (แรงงาน) แล้วตรวจสอบกับส่วนที่ 4 ที่กันออกไว้. ทำเช่นนี้จนครบ 5 รัน แล้วนำผลที่ได้มาเฉลี่ย.

ด้วยวิธีนี้ แต่ละรันจะฝึกแบบจำลองด้วยข้อมูลขนาด $K - 1$ ส่วนจากทั้งหมด K ส่วน และผลค่าผิดพลาดจากการตรวจสอบ เป็นค่าเฉลี่ยของค่าผิดพลาดที่ได้จากทุกส่วนของข้อมูล. วิธีกรอสวัลเดชั้น ทำให้สนใจว่ามีข้อมูลมากขึ้น ทั้งการฝึกและการตรวจสอบ.

แม้วิธีกรอสวัลเดชั้น จะบรรเทาปัญหาของขนาดข้อมูลที่จำกัด และเป็นวิธีที่ใช้ข้อมูลได้อย่างคุ้มค่า แต่ข้อเสียของวิธีกรอสวัลเดชั้นคือ การที่ต้องทำการรันทั้งหมด K ครั้ง. ถ้าเลือกค่า K ใหญ่ ก็จะเบริยบเสื่อมได้ใช้ข้อมูลปริมาณมากในการฝึก แต่ข้อเสีย คือเท่ากับเพิ่มจำนวนรันด้วย โดยเฉพาะถ้าการรันแต่ละครั้งใช้เวลามาก. กล่าวอีกนัยก็คือ วิธีกรอสวัลเดชั้น ใช้การคำนวณที่เพิ่มขึ้น เพื่อบรรเทาปัญหาข้อมูลปริมาณน้อย. ดังนั้น หากถ้าการรันแต่ละครั้งใช้การคำนวณมากอยู่แล้ว แนวทางของวิธีกรอสวัลเดชั้นอาจจะไม่เหมาะสม.

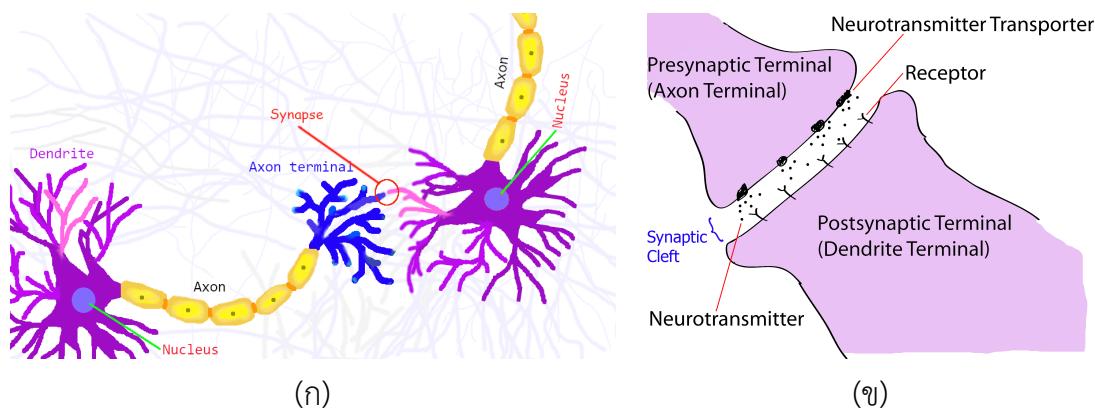
สมองมนุษย์ประกอบด้วยเซลล์ชนิดต่าง ๆ มากมาย เช่น เส้นเลือด เซลล์เกลีย เซลล์ประสาท เส้นเลือดทำหน้าที่รับส่งอากาศ น้ำ อาหาร เซลล์เกลีย(neuroglia) ทำหน้าที่สนับสนุนต่าง ๆ รวมถึงการรักษาภาวะhomeostasis เพื่อให้ภายในสมองมีสภาพที่เหมาะสม เช่น การควบคุมระดับความเข้มข้นของโซเดียมและแคลเซียมในอ่อน เซลล์ประสาททำหน้าที่หลักของสมอง ได้แก่ การควบคุมระบบการทำงานต่าง ๆ ในร่างกายให้เป็นปกติ รวมไปถึง การให้ความสามารถในการจำ การเรียนรู้ การคิด การรับรู้ และการตอบสนอง.

สมองมนุษย์มีเซลล์ประสาทอยู่ประมาณแสนล้านเซลล์ เซลล์ประสาทเองก็มีอยู่หลายประเภท แต่โครงสร้างพื้นฐาน มีลักษณะคล้าย ๆ กัน นั่นคือ เซลล์ประสาthatแต่ละเซลล์ มีไซปรัสที่เพื่อรับสัญญาณเข้าสู่เซลล์ เรียกว่า เดนไดร็ต(dendrite) สัญญาณต่าง ๆ ทั้งสัญญาณกระตุ้นและสัญญาณบันยังที่เข้าสู่เซลล์ จะถูกนำมารวมกันที่นิวเคลียส และผลรวมของสัญญาณที่รับเข้ามา จะเป็นตัวตัดสินว่า เซลล์ประสาทนั้นจะอยู่ในสถานะถูกกระตุ้นหรือไม่ ถ้าเซลล์ประสาทอยู่ในสถานะถูกกระตุ้น มันจะส่งสัญญาณออกไปให้กับเซลล์ประสาทอื่น ๆ ที่รับสัญญาณจากมัน โดยส่งออกผ่านไซปรัส ออกไปสัญญาณ เรียกว่า แอกซอน(axon) จุดต่อระหว่างแอกซอนของเซลล์ประสาทตัวหนึ่งกับเดนไดร็ตของเซลล์ประสาท อีกเซลล์หนึ่ง เป็นจุดประสานประสาทที่เรียกว่า ไซแนปส์ (synapse) แนวคิดพื้นฐานนี้เองที่ โอลเซนแบล็อก (หัวข้อ 3.3) นำไปสร้างแบบจำลองเพอร์เซปตรอน (ดูรูปที่ 3.13 และ 3.12 ประกอบ) เมื่อเปรียบเทียบเพอร์เซปตรอน (รูปที่ 3.14) กับเซลล์ประสาท ผลคุณของอินพุตกับค่าน้ำหนักของเพอร์เซปตรอน (เช่น x_1w_1 และ x_2w_2) เทียบได้กับ ความแรงของสัญญาณประสาท แต่ละสัญญาณที่รับเข้ามาผ่านไซแนปส์ แล้วเดินทางเข้าสู่นิวเคลียสของเซลล์ประสาท เพื่อไปรวมกับความแรงของสัญญาณประสาทที่รับเข้ามาผ่านไซแนปส์จุดอื่น ๆ ความแรงของสัญญาณประสาท แต่ละสัญญาณที่รับเข้ามาผ่านไซแนปส์ จะขึ้นอยู่กับ สัญญาณที่ส่งมา (เปรียบเทียบกับ x_i) และ ความแข็งแรงในการเข้ามือต่อสัญญาณของไซแนปส์ (เปรียบเทียบกับ w_i).

โดยเฉลี่ยแล้ว เซลล์ประสาทแต่ละเซลล์จะมีไซแนปส์ประมาณห้าพันจุด ซึ่งนั่นคือเมื่อรวมแล้ว ในสมองมนุษย์หนึ่ง คนจะมีการเข้ามือต่อประสาทอยู่ราว ๆ ห้าร้อยล้านล้านไซแนปส์ การรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทมีหลายกลไก เช่น กลไกทางเคมี (ผ่านสารสื่อประสาท) กลไกทางไฟฟ้า และ กลไกเชิงภูมิคุ้มกัน แต่กลไกหลักของการส่งสัญญาณประสาทคือกลไกทางเคมี ซึ่งคือการรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทโดยดำเนินการผ่านสารสื่อประสาท (neurotransmitter) เซลล์ประสาทที่ส่งสัญญาณจะปล่อยสารสื่อประสาทออกมานั้น ผ่านโปรตีนที่ทำหน้าที่ส่งสารสื่อประสาท โปรตีนส่งสาร เรียกว่า ทรานสปอร์เตอร์ (neurotransmitter transporter) และเซลล์ประสาทที่รับสัญญาณ จะรับสารสื่อประสาทเหล่านั้น ด้วยโปรตีนที่ทำหน้าที่รับสารสื่อประสาท โปรตีนรับสาร เรียกว่า รีเซปเตอร์ (receptor).

เมื่อรีเซปเตอร์ได้รับสารสื่อประสาท นั่นคือ โครงสร้างของสารสื่อประสาทจับกับโครงสร้างของรีเซปเตอร์ แล้วทำให้กลไกของรีเซปเตอร์เปิดทำงาน โมเลกุลที่จับกับรีเซปเตอร์ จะเรียกว่า ลิกเคนต์ (ligand) กลไกของการจับตัวระหว่างรีเซปเตอร์กับลิกเคนต์นี้ จะเป็นกลไกในลักษณะแม่กุญแจกับลูกกุญแจ (lock and key) นั่นคือ โครงสร้างของรีเซปเตอร์แต่ละชนิดจะจับตัวได้เฉพาะกับลิกเคนต์ที่มีโครงสร้างที่เข้ากันได้เท่านั้น เช่น สารสื่อประสาทอาเซติลโคลีน (acetylcholine) ซึ่งเป็นสารสื่อประสาทที่เซลล์ประสาทใช้คิดต่อกระตุ้นเซลล์กล้ามเนื้อ จะจับกับรีเซปเตอร์สำหรับอาเซติลโคลีนได้เท่านั้น และ รีเซปเตอร์สำหรับสารสื่อประสาทตัวอื่น ก็ไม่อาจจับกับอาเซติลโคลีนได้เช่นกัน การเข้าใจกลไกการทำงานลักษณะนี้ ช่วยให้เภสัชศาสตร์สามารถออกแบบตัวยาที่เฉพาะเจาะจงกับสารสื่อประสาทเฉพาะตัวได้ เช่น ยาต้านอาการเกร็ง ฟลูโอดเซตทีน (Fluoxetine) ที่เฉพาะเจาะจงกับสารสื่อประสาทเซอโรโนนิน (Serotonin).

หมายเหตุ เซลล์ประสาทแต่ละชนิดจะมีลักษณะเฉพาะตัวต่างกันและจะทำงานกับสารสื่อประสาทเฉพาะชนิด เช่น เซลล์ประสาทเซอโรโนนิน (serotonin neurons) ที่อยู่บริเวณดอร์ราฟิโนเคลียส (dorsal raphe nucleus) ของก้านสมอง จะทำงานกับสารสื่อประสาทเซอโรโนนิน[119] เซลล์ประสาทซีเอ๑พีรามิดอล (CA1 pyramidal neurons) ที่อยู่บริเวณซีเอ๑ (CA1) ของอิบปีโนแคมปัส จะทำงานกับสารสื่อประสาทกลูตาเมต(Glutamate)[138] เซลล์ประสาทมิดเบรน ໂດพามีนอยิก (midbrain dopaminergic neurons) ที่อยู่หลาย ๆ บริเวณรวมถึง พื้นที่เวนทรอลเทกโนนทอล (ventral



รูปที่ 3.12 ภาพแสดงเซลล์ประสาทเชื่อมต่อสัญญาณกันผ่านไชแนปซ์ โดยสัญญาณที่สื่อสารกันนั้นทำได้โดยผ่านกลไกของสารสื่อประสาท. ภาพ ก แสดงเซลล์ทางซ้ายมีส่งสัญญาณผ่านไชแนปซ์ ไปสู่เซลล์ทางขวาเมื่อ. ภาพ ข แสดงภาพขยายส่วนของไชแนปซ์ ซึ่งส่วนปลายของแอกซอฟอน (*presynaptic terminal*) จะส่งสารสื่อประสาทออกมามา และส่วนปลายของเดนไดร็ต (*postsynaptic terminal*) จะรับสารสื่อประสาท.

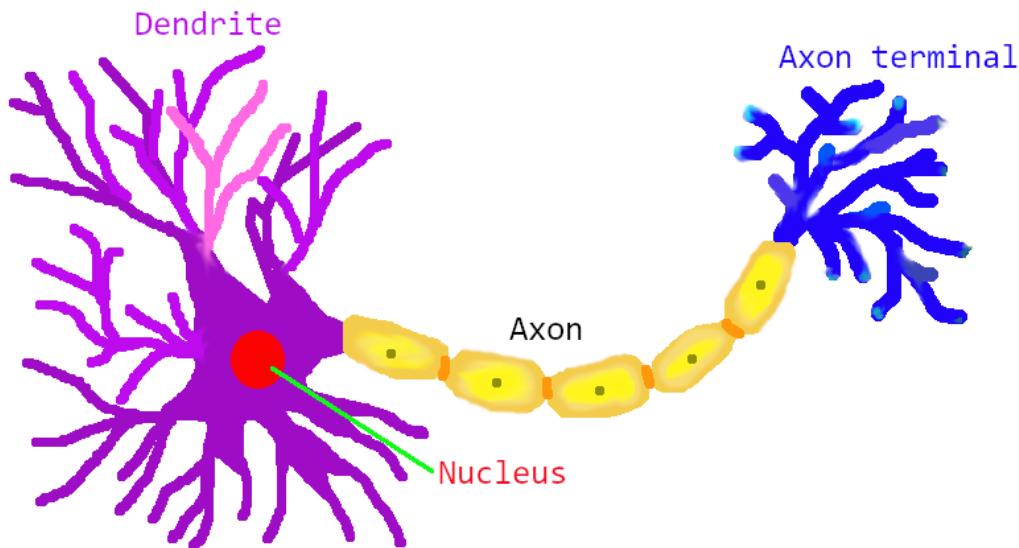
tegmental area) ในสมองส่วนกลาง จะทำงานกับสารสื่อประสาทดोพามีน (*Dopamine*)^[138]. เซลล์ประสาทบางชนิดทำงานกับสารสื่อประสาทมากกว่าหนึ่งชนิด เช่น เซลล์ประสาทนามกลาง (medium spiny neurons) ที่อยู่บริเวณ bazol แกงเกลี้ย ส่งสัญญาโนอกผ่านกาบา (*GABA*) แต่สามารถรับสัญญาณผ่านสารสื่อประสาทหลายชนิดรวมถึงกลูตาเมตและโดพามีน.

3.3 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (*artificial neural network* คำย่อ ANN) เป็นแบบจำลองที่นำมาย ที่ใช้ลักษณะการคำนวณง่าย ๆ คล้าย ๆ กัน จำนวนมาก ที่เมื่อนำมารวมกันแล้ว ให้ผลโดยรวม เป็นแบบจำลองที่นำมายที่มีความสามารถสูง.

โครงข่ายประสาทเทียม มีอยู่หลายชนิด หนึ่งในชนิดที่สำคัญและได้รับความนิยมอย่างมาก คือ เพอร์เซปตรอนหลายชั้น. เพอร์เซปตรอนหลายชั้น (*multi-layer perceptron* คำย่อ MLP) ที่รวมการคำนวณของหน่วยคำนวณย่อย ที่เรียกว่า เพอร์เซปตรอน หลาย ๆ หน่วย เข้าด้วยกัน ในลักษณะเป็นชั้น ๆ.

หน่วยคำนวณย่อย เพอร์เซปตรอน (*perceptron*) ถูกพัฒนาโดยการเลียนแบบเซลล์ประสาทของสัมภาระ. เซลล์ประสาทของสัมภาระมีอยู่หลายชนิด แต่มีลักษณะทั่ว ๆ ไป ดังแสดงในรูปที่ 3.13. แต่ละเซลล์รับสัญญาณกราฟตุ้นจากเซลล์อื่น ๆ ผ่านเดนไดร็ต (*dendrite*) เมื่อผลรวมของสัญญาณกราฟตุ้นมากพอเซลล์จะเข้าสู่สถานะภูมิกราฟตุ้น และส่งสัญญาโนอกผ่านแอกซอฟอน (*axon*) ไปให้เซลล์อื่น ๆ ต่อไป. ความแรงของสัญญาณกราฟตุ้นที่รับมาจากแต่ละเซลล์ก็ต่าง ๆ กันไป ขึ้นกับการเชื่อมต่อ ซึ่งความแข็งแรงของการเชื่อมต่อ ก็มีการปรับเปลี่ยนตามการใช้งาน.



รูปที่ 3.13 รูปแสดงโครงสร้างของเซลล์ประสาททั่ว ๆ ไป เด่นไดรต์ ทำหน้าที่สืบสานอินพุตของเซลล์ และ ออกชอน ทำหน้าที่ส่งเมื่อนกับเอ้าต์พุตของเซลล์. เมื่อสัญญาณกระตุ้นจากอินพุตร่วมกันมากพอ เซลล์จะเข้าสู่สถานะถูกกระตุ้น และส่งการกระตุ้นออกต่อไป ผ่านออกชอน.

แฟรงค์ โรเซนแบล็ท (Frank Rosenblatt) ออกแบบ สร้าง และได้สาธิตการทำงานของเพอร์เซปตรอน ที่สร้างด้วยวงจรไฟฟ้า เพื่อจำลองการทำงานของเซลล์ประสาท ในปี 1957. รูปที่ 3.14 แสดงโครงสร้างแนวคิดของเพอร์เซปตรอน.

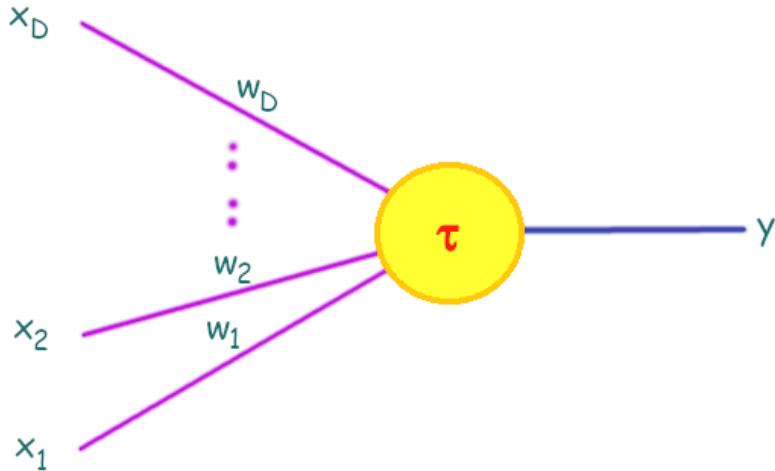
การคำนวณของเพอร์เซปตรอน ดำเนินการ โดย การนำอินพุตแต่ละตัว ไปคูณกับค่าน้ำหนักของ อินพุตนั้น ๆ และนำค่าผลคูณทั้งหมดมาบวกกัน. แล้วหากผลรวมมีค่ามากพอ นั่นคือ มีค่าเท่ากับหรือมากกว่าค่าระดับกระตุ้น เพอร์เซปตรอนจะอยู่ในสถานะถูกกระตุ้น (ให้อาต์พุตเป็น 1) แต่หากผลรวม มีค่าน้อยกว่าค่าระดับกระตุ้น เพอร์เซปตรอนจะอยู่ในสถานะไม่ถูกกระตุ้น (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 และพารามิเตอร์ τ คือ ค่าระดับกระตุ้น โดยผลลัพธ์ $y = 1$ แทนสถานะการถูกกระตุ้น และ $y = 0$ แทนสถานะไม่ถูกกระตุ้น. บางครั้ง อินพุต x_1, \dots, x_D อาจถูกมองรวม คือมองเป็น อินพุต $\mathbf{x} = [x_1, \dots, x_D]^T$ โดยแต่ละตัว หรือแต่ละส่วนประกอบ x_i จะเรียกเป็น มิติ (dimension) หรือคุณลักษณะ (feature) ของอินพุต.

เพื่อความสะดวก นิยาม ไบอส (bias) เป็น $b = -\tau$ และเพอร์เซปตรอนสามารถเขียนได้เป็น

$$y = \begin{cases} 0 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b < 0, \\ 1 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b \geq 0. \end{cases} \quad (3.12)$$



รูปที่ 3.14 แผนผังแสดงโครงสร้างของเพอร์เซปตรอน ที่เอาต์พุต y แสดงสถานะของหน่วยประสาทเทียม โดย หน่วยประสาทเทียมจะอยู่ในสถานะสูกระดับต้น เมื่อผลรวมสัญญาณกระดับต้น ซึ่งคือ $w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_D \cdot x_D$ มีค่ามากพอ (มากถึงหรือเกินค่าระดับกระดับต้น τ). ค่า x_1, \dots, x_D เป็นอินพุตต่าง ๆ ของเพอร์เซปตรอน. ค่า w_1, \dots, w_D เป็นค่าน้ำหนัก แสดงความแข็งแรงของการเชื่อมต่อ กับแต่ละอินพุต.

และเมื่อมองในมุมที่กว้างขึ้น สมการ 3.12 สามารถเขียนได้เป็น

$$y = h \left(\sum_{i=1}^D w_i x_i + b \right) \quad (3.13)$$

เมื่อ h เป็นฟังก์ชันกระดับต้น (activation function) ซึ่งอาจนิยามเป็น ฟังก์ชันจำกัดแข็ง (hard limit function) หรือบางครั้งอาจเรียก ฟังก์ชันขั้นบันไดหนึ่งหน่วย unit step function) ได้แก่

$$h(a) = \begin{cases} 0 & \text{เมื่อ } a < 0, \\ 1 & \text{เมื่อ } a \geq 0. \end{cases} \quad (3.14)$$

ในตอนนี้ งานของโรเชนแบลททำให้วิเคราะห์โดยเฉพาะอย่างยิ่งการปัญญาประดิษฐ์ตั้นเต็มมาก ที่แนวทางนี้ อาจเป็นโอกาสที่มนุษย์จะสามารถสร้างเครื่องจักร ที่สามารถเลียนแบบการทำงานของสมองมนุษย์ได้ และเป้าหมายของปัญญาประดิษฐ์ และความฝันของวิทยาการคอมพิวเตอร์อาจจะสำเร็จได้ เกิดการคาดการณ์ถึงศักยภาพ ความสามารถต่าง ๆ ที่เครื่องคอมพิวเตอร์จะสามารถทำได้. แต่ความฝันและความหวังก็ล้มสถาายนไป หลังจาก มาเรียน มินสกี้ (Marvin Minsky) และ เชมาร์ ปาเปิต (Seymour Papert) ได้ร่วมกันเขียนหนังสือเพอร์เซปตรอนส์[128] ที่วิเคราะห์โครงสร้างและการทำงานของเพอร์เซปตรอน. ประเด็นสำคัญของหนังสือ คือ มินสกี้และปาเปิตถก และวิจารณ์ว่า เพอร์เซปตรอนนี้สามารถทำได้แต่ง่าย ๆ เช่นหากเป็นงานการจำแนกประเภท ก็สามารถทำงานได้ กับปัญหาที่สามารถแบ่งได้ด้วยเส้นแบ่งตัดสินใจเชิงเส้นเท่านั้น ไม่สามารถทำงานที่ซับซ้อนกว่านั้นได้.

พร้อมทั้งยังยกตัวอย่าง การทำงานของตรรกะ เอ็กซ์ออร์ (XOR หรือ exclusive OR) ที่เพอร์เซปตรอน ไม่สามารถเลียนแบบได้. ตารางที่ 3.2 แสดงพฤติกรรมของตรรกะเอ็กซ์ออร์.

ตารางที่ 3.2 ตรรกะเอ็กซ์ออร์ ที่อ้างว่าเพอร์เซปตรอน ไม่สามารถทำงานได้. ตรรกะเอ็กซ์ออร์ เป็นตรรกะ ที่ผลลัพธ์ไม่สามารถใช้สืบสานต่อตัวเดียวกันได้.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

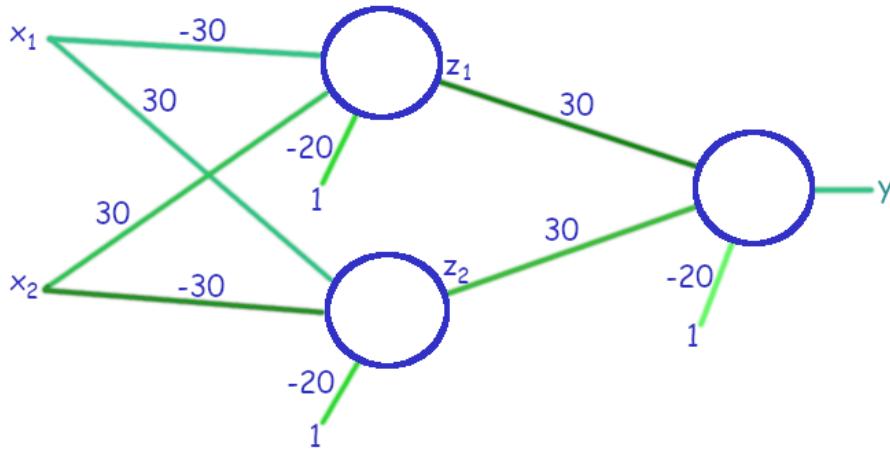
ผลจากคำวิจารณ์ของมินสกี้และปาเปิต นอกจากจะทำให้เพอร์เซปตรอนเสื่อมความสนใจแล้ว ยังทำให้เทคนิคทางด้านโครงข่ายประสาทเทียมทั้งหมด รวมไปถึงสาขาวิชาปัญญาประดิษฐ์ เสียความนิยม และเสื่อมความสนใจไปในช่วงหลายปีต่อจากนั้น จนเรียกว่า ช่วงเวลานั้นเป็น หน้าหนาวของปัญญาประดิษฐ์ (AI Winter). นอกจากคำวิจารณ์ของมินสกี้และปาเปิต เหตุผลอื่น ๆ ที่มีส่วนทำให้โครงข่ายประสาทเทียมเสียความนิยมไป ได้แก่ (1) การกำหนดโครงสร้างการเชื่อมต่อของโครงข่ายประสาทเทียม ที่ตอนนั้นยังใหม่มาก และยังไม่มีแนวทางที่ชัดเจน¹ และ (2) การใช้งานโครงข่ายประสาทเทียม จะต้องเลือกค่าน้ำหนัก และค่าไบอสให้ถูกต้อง ซึ่ง ณ ช่วงเวลานั้น ยังไม่มีวิธีการที่มีประสิทธิภาพในการเลือกค่าน้ำหนัก และค่าไบอส. (ยังไม่มีแม้แต่ หลักในการเลือกจำนวนเพอร์เซปตรอนที่เหมาะสมกับการใช้งาน.)

จนกระทั่งรัฐศาสตร์ให้หลัง งานของเวอร์โนบ[214]และโดยเฉพาะอย่างยิ่งงานของรูเมลาร์ต, ชินตัน และวิลเลียม[170] ที่อุปมาแสดงให้เห็นถึงประสิทธิผลของโครงข่ายประสาทเทียม พร้อมนำเสนอวิธีที่มีประสิทธิภาพ ในการหาค่าน้ำหนัก และค่าไบอส ซึ่งงานเหล่านี้ ได้ช่วยฟื้นฟูความนิยมของโครงข่ายประสาทเทียมกลับมาใหม่.

เมื่อจะกล่าวไปแล้ว สิ่งที่มินสกี้กับปาเปิตวิจารณ์ว่า เพอร์เซปตรอนทำงานได้แต่งานง่าย ๆ ก็ไม่ได้ผิดชาติ. เพียงแต่ว่า มินสกี้กับปาเปิตสรุปความเห็น จากการวิเคราะห์การทำงานของเพอร์เซปตรอนที่เป็นโครงข่ายชั้นเดียว. การทำงานของโครงข่ายสมองมนุษย์ไม่ได้เป็นชั้นเดียว ในลักษณะเดียวกัน โครงข่ายประสาทเทียมที่มีประสิทธิผล จะต้องมีโครงสร้างมากกว่าหนึ่งชั้น. นั่นก็คือ ที่มาของพัฒนาการต่อมา ได้แก่ เพอร์เซปตรอนหลายชั้น ซึ่งซึ่งได้เน้นย้ำ ถึงการใช้โครงข่ายต่อเชื่อมกันในลักษณะหลายชั้น ของหน่วยคำนวณแบบเพอร์เซปตรอน.

รูปที่ 3.15 แสดงเพอร์เซปตรอนสองชั้น (two-layer perceptron) ที่สามารถเลียนแบบการทำงานของตรรกะเอ็กซ์ออร์ได้ โดยใช้เพอร์เซปตรอนสามตัว ต่อ กันในลักษณะสองชั้นคำนวณ. เพอร์เซปตรอน

¹ สถาปัตยกรรมที่ต้องน่วยคำนวณอยู่เป็นลักษณะของชั้นคำนวณ ไม่ได้มาโดยธรรมชาติ แต่มาจากการออกแบบภายหลัง.



รูปที่ 3.15 ตัวอย่างโครงข่ายเพอร์เซปตรอนสองชั้น ที่ทำตระกูลเอ็กซ์อร์. เพอร์เซปตรอนสามหน่วยต่อในลักษณะชั้นคำนวณ โดย เพอร์เซปตรอนสองหน่วย รับอินพุต x_1 และ x_2 และคำนวณเอาต์พุตอกมา เป็น z_1 (หน่วยบน) และ z_2 (หน่วยล่าง). เอาต์พุต z_1 และ z_2 จากสองหน่วย ถูกใช้เป็นอินพุตของเพอร์เซปตรอนตัวสุดท้าย (ตัวขวาสุด) และเอาต์พุตของเพอร์เซปตรอนตัวสุดท้าย y ถูกใช้เป็นเอาต์พุตของโครงข่าย. ค่าน้ำหนักระหว่างการเชื่อมต่อ เทียบปริมาณสัมแสดงการเชื่อมต่อ. ในภาพ ค่าไปอัลส ถูกแสดงด้วยค่าน้ำหนักของอินพุตค่าคงที่หนึ่ง.

แต่ละตัว อาจถูกเรียกว่า **โนนด (node)** หรือ **หน่วยคำนวณ (unit)**. ผลลัพธ์จากโนนดในชั้นคำนวณแรก และส่งไปเป็นอินพุตให้กับโนนดในชั้นคำนวณที่สอง. การจัดโครงสร้างเป็นลักษณะชั้นคำนวณ (layer) แบบนี้ ช่วยให้โครงข่ายประสาทเทียมสามารถทำงานที่ซับซ้อนได้. เอาต์พุตของเพอร์เซปตรอนสองโนนดในชั้นแรก ซึ่งคือ $z_1 = h(-30x_1 + 30x_2 - 20)$ และ $z_2 = h(30x_1 - 30x_2 - 20)$ ทำงานที่เป็นอินพุตของเพอร์เซปตรอนตัวที่อยู่ชั้นที่สอง. เอาต์พุตของเพอร์เซปตรอนชั้นที่สอง ซึ่งเป็นชั้นสุดท้าย ที่จะใช้เป็นเอาต์พุตของทั้งโครงข่าย คำนวณด้วย $y = h(30z_1 + 30z_2 - 20)$.

ตารางที่ 3.3 แจ้งແลงการทำงาน โดย $a_1^{(1)}$ และ $a_2^{(1)}$ เป็นตัวกระตุ้น (ผลกระทบของสัญญาณกระตุ้น) ของโนนดตัวบน และของตัวล่างในชั้นที่หนึ่ง ตามลำดับ และ $a^{(2)}$ เป็นของโนนดในชั้นที่สอง. สังเกตว่า ค่าน้ำหนักและไปอัลส สามารถเปลี่ยนไปใช้ค่าอื่นได้ โดยที่การทำงานยังคงเดิมได้ เช่น อาจใช้ค่า $20, -20, -10$ แทน $30, -30, -20$ ในรูปที่ 3.15 ได้ โดยพฤติกรรมการทำงานยังคงเดิม. นี้เป็นลักษณะอย่างหนึ่งของโครงข่ายประสาทเทียม ที่ ค่าน้ำหนักและไปอัลสที่ดีที่สุด มีได้หลายชุด.

การคำนวณของเพอร์เซปตรอนหลายชั้น 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 $\mathbf{x} = [x_1, \dots, x_D]^T$ เป็นอินพุตของแบบ

ตารางที่ 3.3 ตัวอย่างการทำงานของเพอร์เซปตรอน 2 ชั้นในรูปที่ 3.15. แต่ละแกร แสดงแต่ละกรณี ตั้งแต่ (x_1, x_2) เป็น $(0, 0), (0, 1), (1, 0)$ และ $(1, 1)$. สมมุติ แสดงค่าของตัวแปรต่าง ๆ ดังระบุที่หัวสมมุติ ได้แก่ อินพุต x_1 และ x_2 , ค่าการกระตุ้น a_1 และ a_2 , ผลลัพธ์การกระตุ้น z_1 และ z_2 และเอาต์พุต 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	$-30(1) + 30(1) - 20 = -20$	$h(-20) = 0$	$30(1) - 30(1) - 20 = -20$	$h(-20) = 0$	$30(0) + 30(0) - 20 = -20$	$h(-20) = 0$

จำลอง 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} มีโนนดจำนวน M_q โนนด. พารามิเตอร์ $w_{ji}^{(q)}$ เป็นค่าน้ำหนักสำหรับการเข้มต่อของโนนด j^{th} ในชั้น q^{th} กับโนนด i^{th} ในชั้นก่อนหน้า. พารามิเตอร์ $b_j^{(q)}$ เป็นค่าไบอสของโนนด j^{th} ในชั้น q^{th} . พงกชั้น h เป็นพงกชั้นกระตุ้น.

ตัวแปร $z_j^{(q)}$ เป็นเอาต์พุตของโนนด j^{th} ในชั้น q^{th} สำหรับ $q = 1, \dots, L$ โดยสมการ 3.15 นิยามเอาต์พุตของชั้น 0^{th} เป็นอินพุตของแบบจำลอง เมื่อความกระหัดรัด.

เอาต์พุตของแบบจำลองคือ เอาต์พุตของโนนดต่าง ๆ ในชั้นสุดท้าย (ดังระบุในสมการ 3.18) ดังนั้น $M_L = K$ โดย K เป็นจำนวนมิติของเอาต์พุตที่ต้องการ. จำนวนชั้นคำนวณและจำนวนโนนดในแต่ละชั้น เป็นอภิธานพารามิเตอร์ของแบบจำลองเพอร์เซปตรอนหลายชั้น.

สมการ 3.15 ถึง 3.18 แสดงการคำนวณของเพอร์เซปตรอน ในรูปตัวแปรสเกลาร์. การคำนวณของเพอร์เซปตรอน สามารถเขียนได้กระชับกว่า โดยเขียนในรูปเกกเตอร์และเมทริกซ์ ดังสมการ 3.19 และ 3.20 ซึ่ง การจัดรูปในลักษณะนี้จะเรียกว่า เวคเตอร์ไซชัน (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 $\mathbf{z}^{(q)} = [z_1^{(q)}, z_2^{(q)}, \dots, z_{M_q}^{(q)}]^T$ คือ ผล-

ลัพธ์การกระตุ้นในชั้นคำนวณ q^{th} โดยกำหนดให้ อินพุตเสริมจากโหนดชั้นสูง $\mathbf{z}^{(0)} = \mathbf{x} = [x_1, \dots, x_D]^T$ 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 $\mathbf{b}^{(m)} \in \mathbb{R}^{M_q}$ คือค่าน้ำหนักและค่าไบอสของชั้นคำนวณ q^{th} และ $M_0 = D$ และ $M_L = K$. พิงก์ชัน h เป็นพิงก์ชันกระตุ้น ซึ่งสำหรับพิงก์ชันกระตุ้น $h : \mathbb{R} \mapsto \mathbb{R}$ นิยาม สัญกรณ์ เช่น $h(\mathbf{a})$ เป็นปฏิบัติการที่มีลักษณะการคำนวณเชิงตัวต่อตัว (*element-wise*) เมื่อใช้กับ เวกเตอร์ เมทริกซ์ หรือテンเซอร์ (ยกเว้นจะระบุเป็นอย่างอื่น). นั่นคือ เช่น $h([a_1, \dots, a_m]^T) = [h(a_1), \dots, h(a_m)]^T$.

สังเกตว่า การคำนวณจะดำเนินการในลักษณะคล้าย ๆ กัน เป็นชั้นคำนวณ อินพุตของระบบถูก ป้อนให้กับชั้นคำนวณแรก เมื่อคำนวณผลจากชั้นหนึ่งเสร็จแล้ว ผลลัพธ์จะถูกใช้ป้อนเป็นอินพุตให้กับ ชั้นคำนวณต่อไป และดำเนินการเช่นนี้ต่อไปจนถึงชั้นคำนวณสุดท้าย ผลลัพธ์ของชั้นคำนวณสุดท้าย จะใช้เป็นเอาต์พุตของระบบ. เนื่องจากการคำนวณมีลักษณะที่คำนวณเป็นชั้น ๆ ผ่านไปทิศทางเดียว เพอร์เซปตรอนหลายชั้น บางครั้งอาจถูกเรียกว่า **โครงข่ายแพร่กระจายไปข้างหน้า (feedforward network)**. ชั้นคำนวณทั้งหมดที่อยู่ก่อนชั้นสุดท้าย นั่นคือ ชั้น $q = 1, \dots, L - 1$ จะเรียกว่า **ชั้นซ่อน (hidden layer)** เนื่องจากเอาต์พุตของชั้นคำนวณเหล่านี้ ไม่ได้ใช้เป็นเอาต์พุตสุดท้ายของแบบ จำลอง และโหนดต่าง ๆ ที่อยู่ในชั้นซ่อน จะเรียกว่า **โหนดซ่อน (hidden node)** หรือ **หน่วยซ่อน (hidden unit)**. การเลือกอภิมานพารามิเตอร์ของเพอร์เซปตรอนหลายชั้น บางครั้งนิยม อ้างถึง จำนวนชั้นซ่อน เช่นในตัวอย่างรูปที่ 3.14 อาจอ้างถึงเป็น โครงข่ายประสาทเทียม หนึ่งชั้นซ่อน ที่มีสอง หน่วยซ่อน. หมายเหตุ ถึงแม่โครงข่ายประสาทเทียม จะมีหลายชนิด แต่เพอร์เซปตรอนหลายชั้น เป็น ชนิดแรก และเป็นชนิดที่รู้จักกันอย่างกว้างขวาง บ่อยครั้งที่ คำว่า เพอร์เซปตรอนหลายชั้น, โครงข่าย ประสาทเทียม หรือโครงข่ายแพร่กระจายไปข้างหน้า ถูกใช้แทนกัน.

ปัจจุบันโครงข่ายประสาทเทียม เป็นแบบจำลองได้ถูกนำไปใช้อย่างกว้างขวาง ในงานหลาย ๆ ลักษณะ เช่น การหาค่าคาดถอย การจำแนกประเภท การประมาณพิงก์ชัน. นอกจากนี้ มีการศึกษา โครงข่ายประสาทเทียมในทางทฤษฎี และพิสูจน์ว่าโครงข่ายประสาทเทียม เป็นตัวประมาณค่าสากล (*universal approximator*)[47, 91] ซึ่งความหมายคือ โครงข่ายประสาทเทียม สามารถแทนพิงก์ชัน ได้ ๆ ก็ได้ ที่ความละเอียดตามที่ต้องการ หากมีจำนวนหน่วยคำนวณมากเพียงพอ. ตามทฤษฎีแล้ว แค่ โครงข่ายประสาทเทียมแบบสองชั้น ก็เป็นตัวประมาณค่าสากลได้แล้ว แต่การใช้โครงข่ายประสาทเทียม แบบลีก มีประโยชน์หลายอย่าง ดังที่จะอภิปรายในบทที่ 5.

การฝึกโครงข่ายประสาทเทียม

กลับมาที่เรื่องการหาค่าน้ำหนักและค่าไบอสที่เหมาะสม อย่างที่เห็นจากตัวอย่าง การที่โครงข่ายประสาทเทียมจะสามารถทำงานได้ตามที่ต้องการนั้น นอกจากโครงข่ายจะต้องมีโครงสร้างที่รองรับได้แล้ว (โครงสร้างเป็นลักษณะชั้น ๆ ต่อกันที่เหมาะสม และมีจำนวนหน่วยคำนวณเพียงพอ) ค่าน้ำหนักและค่าไบอสต่าง ๆ จะต้องมีค่าที่เหมาะสมด้วย.

ค่า น้ำหนัก และค่า ไบอัส ของโครงข่ายประสาทเทียม ก็คือ พารามิเตอร์ต่าง ๆ ของแบบจำลอง. ดังนั้น การหาค่า น้ำหนัก และค่า ไบอัส ก็สามารถทำได้ในลักษณะเดียวกับ การหาค่าพารามิเตอร์ของฟังก์ชันพหุนามที่ได้อธิบายในหัวข้อ 3.1. นั่นคือ การใช้วิธีของการหาค่าดีที่สุด และเช่นเดียวกัน การหาค่าน้ำหนักและค่าไบอัส จะเรียกว่า การฝึก.

อย่างไรก็ตาม เปรียบเทียบการฝึกแบบจำลองพหุนาม กับการฝึกโครงข่ายประสาทเทียม มีประเด็นที่น่าสนใจ คือ อนุพันธ์ของฟังก์ชันจุดประสงค์ต่อค่าพารามิเตอร์. การหาอนุพันธ์ เป็นกลไกสำคัญ สำหรับวิธีของการหาค่าดีที่สุดอย่างมีประสิทธิภาพ. เพอร์เซปตรอนใช้ฟังก์ชันจำกัดแข็ง เป็นฟังก์ชัน กระตุ้น แต่เนื่องจาก ฟังก์ชันจำกัดแข็ง (สมการ 3.14) เป็นฟังก์ชันที่มีค่าไม่ต่อเนื่อง (ที่ $a = 0$) ทำให้ ไม่สามารถหาค่าอนุพันธ์ของฟังก์ชันจำกัดแข็ง และส่งผลให้การหาค่าน้ำหนักที่เหมาะสมของเพอร์เซปตรอนทำได้ยาก.

ฟังก์ชันจำกัดแข็ง แม้จะเลียนแบบการทำงานของเซลล์ประสาท แต่ลักษณะทางคณิตศาสตร์ของมัน เป็นอุปสรรคสำคัญ ต่อการฝึกโครงข่ายประสาทเทียม. การสร้างแบบจำลองคณิตศาสตร์ เพื่อทำความเข้าใจเซลล์ประสาทชีวภาพ จัดอยู่ในขอบข่ายของประสาทวิทยาเชิงคำนวณ (*computational neuroscience*) ซึ่งเป็นสาขาเฉพาะ และอยู่นอกเหนือจากขอบเขตของหนังสือเล่มนี้. แม้กระนั้น ฟังก์ชันจำกัดแข็ง ก็ไม่ได้อธิบายการทำงานของเซลล์ประสาทชีวภาพได้เที่ยงตรงซึ่งที่เดียว นอกจากนั้น สิ่งที่ต้องการจริง ๆ ในมุมมองทางวิศวกรรม ก็คือเครื่องมือที่ใช้งานได้ เช่น แบบจำลองที่มีความสามารถในการทำนายที่ดี.

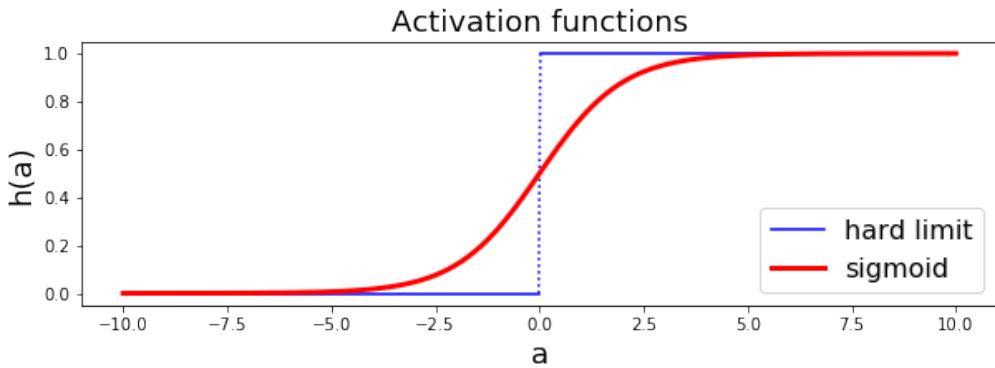
การฝึกโครงข่ายประสาทเทียมที่ใช้ฟังก์ชันจำกัดแข็งจะทำได้ยาก และไม่สามารถทำได้อย่างมีประสิทธิภาพ. เมื่อปัญหาอยู่ที่ฟังก์ชันจำกัดแข็ง วิธีแก้ก็แก้ที่ฟังก์ชันจำกัดแข็ง. สิ่งที่รูเมลาร์ต, อินตัน และวิลเลียม[170] เสนอคือ ใช้ฟังก์ชันซิกมอยด์ (*sigmoid function* หรือบางครั้งเรียก *logistic function* หรือ *logistic sigmoid function*) เป็นฟังก์ชันกระตุ้นแทนฟังก์ชันจำกัดแข็ง. สมการ 3.21 แสดงการคำนวณฟังก์ชันซิกมอยด์

$$\text{sigmoid}(a) = \frac{1}{1 + \exp(-a)}. \quad (3.21)$$

ฟังก์ชันซิกมอยด์ เป็นฟังก์ชันค่าต่อเนื่อง ดังนั้นจึงสามารถหาอนุพันธ์ได้. รูปที่ 3.16 แสดงค่าผลลัพธ์ การกระตุ้นจากฟังก์ชันจำกัดแข็งเปรียบเทียบกับฟังก์ชันซิกมอยด์.

อย่างไรก็ตามแม้ ชื่อของเพอร์เซปตรอนจะเชื่อมโยงกับฟังก์ชันจำกัดแข็ง แต่ในทางปฏิบัติแล้ว โดยทั่วไป ชื่อเพอร์เซปตรอนหลายชั้น ก็มักอ้างถึงโครงข่ายประสาทเทียม ที่ใช้ฟังก์ชันซิกมอยด์ เป็นฟังก์ชัน กระตุ้น. นอกจากฟังก์ชันซิกมอยด์แล้ว ฟังก์ชันไฮเปอร์บolic แทนเจนต์ (*hyperbolic tangent function*) ซึ่งคำนวณโดย $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ ก็นิยมใช้เป็นฟังก์ชันกระตุ้นของโครงข่ายประสาทเทียม.

อีกประเด็น ฟังก์ชันกระตุ้น h ในสมการ 3.17 หรือ 3.20 ไม่จำเป็นต้องใช้เหมือนกันทุก ๆ ชั้น คำนวณ. นั่นคือ สมการ 3.20 อาจเขียนใหม่ เพื่อเน้นประเด็นนี้ได้เป็น $\mathbf{z}^{(q)} = h_q(\mathbf{a}^{(q)})$ เมื่อ h_q เป็น



รูปที่ 3.16 ภาพเปรียบเทียบฟังก์ชันจำกัดแข็ง (สันบางสีฟ้า) กับฟังก์ชันซิกมอยด์ (สัน涵สีแดง).

ฟังก์ชันกระตุนของชั้น q^{th} . ในทางปฏิบัติแล้ว ฟังก์ชันกระตุนของชั้นคำนวณสุดท้าย มักจะต่างจาก ฟังก์ชันกระตุนของชั้โน่น ๆ. ชั้นคำนวณสุดท้าย มักถูกเรียกว่า ชั้นเอาต์พุต (output layer) เป็นชั้น ที่จะเตรียมค่าของเอาต์พุต ให้อยู่ในรูปแบบที่ใกล้เคียง กับรูปแบบเอาต์พุตที่เหมาะสมมากที่สุด. ฟังก์ชันกระตุนของชั้นเอาต์พุต เรียกสั้น ๆ ว่า ฟังก์ชันกระตุนเอาต์พุต (output activation function) จะถูกเลือกใช้ตามภารกิจ และลักษณะค่าเอาต์พุตที่ต้องการ. ตัวอย่างเช่น การหาค่า ณดถอย ซึ่งต้องการเอาต์พุต $y \in \mathbb{R}$. ฟังก์ชันกระตุนเอาต์พุต นิยมใช้ฟังก์ชันเอกลักษณ์ (identity function). นั่นคือ $h(a) = a$.

เมื่อเลือกใช้ฟังก์ชันกระตุนเป็นฟังก์ชันต่อเนื่องแล้ว การฝึกโครงข่ายประสาทเทียม ก็สามารถทำได้อย่างมีประสิทธิภาพ โดยอาศัยค่าอนุพันธ์. เช่นเดียวกับแบบจำลองท่านายอื่น ๆ การฝึกโครงข่าย ประสาทเทียม คือ การหา $\Theta^* = \arg \min_{\Theta} E$ เมื่อ E คือ ฟังก์ชันค่าผิดพลาดที่เป็นจุดประสงค์ และ Θ คือพารามิเตอร์ของแบบจำลอง ซึ่งสำหรับโครงข่ายประสาทเทียม จากสมการ 3.16 ก็คือ $\Theta = \{w_{ji}^{(q)}, b_j^{(q)}\}$ โดย $q = 1, \dots, L; j = 1, \dots, M_q$; และ $i = 1, \dots, M_{q-1}$.

สำหรับการหาค่าณดถอย (regression) ซึ่งคือการทำนายเอาต์พุตที่ค่าเป็นจำนวนจริง ฟังก์ชันค่า ผิดพลาด E สามารถนิยามด้วย ค่าเฉลี่ยค่าผิดพลาดกำลังสอง ดังแสดงในสมการ 3.22 สำหรับข้อมูล จำนวน 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 $\hat{\mathbf{y}}$ หาก บริบทชัดเจน คือค่าที่แบบจำลองทำนายสำหรับจุดข้อมูลที่ n^{th} เมื่อใช้ค่าพารามิเตอร์เป็น Θ . ค่าของ $\hat{\mathbf{y}} = \mathbf{z}^{(L)}$ คำนวณได้จากสมการ 3.19 และ 3.20 โดยให้ $\mathbf{z}^{(0)} = \mathbf{x}_n$ เมื่อ \mathbf{x}_n เป็นค่าตัวแปรต้นของจุด ข้อมูลที่ 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

$$\nabla_{\Theta} E_n = \begin{bmatrix} \frac{\partial E_n}{\partial w_{1,1}^{(1)}} & \cdots & \frac{\partial E_n}{\partial w_{M_1,M_0}^{(1)}} & \frac{\partial E_n}{\partial b_1^{(1)}} & \cdots & \frac{\partial E_n}{\partial b_{M_1}^{(1)}} & \cdots & \frac{\partial E_n}{\partial w_{M_L,M_{L-1}}^{(L)}} & \cdots & \frac{\partial E_n}{\partial b_{M_L}^{(L)}} \end{bmatrix}^T \quad (3.24)$$

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 $\frac{\partial E_n}{\partial w_{ji}^{(q)}} = \frac{\partial E_n}{\partial a_j^{(q)}}$. $\frac{\partial a_j^{(q)}}{\partial w_{ji}^{(q)}}$ เพื่อความกระชับ สัญกรณ์ที่แสดง อาจจะตัวยก (q) ในกรณีที่บริบทชัดเจน 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 $\delta_j^{(q)} = \frac{\partial E_n}{\partial a_j^{(q)}}$ ที่ชื่นเอาร์พูต $q = L$ แทนค่า E_n จากสมการ 3.23 และสำหรับการหาค่า ณดถอย พังก์ชันกราฟตุ้นเอาร์พูตใช้พังก์ชันเอกลักษณ์ นั่นคือ $[\hat{y}_1, \dots, \hat{y}_K]^T = [a_1^{(L)}, \dots, a_K^{(L)}]^T$ จะได้ว่า $\delta_k^{(L)} = \frac{\partial E_n}{\partial a_k^{(L)}} = \frac{\partial \frac{1}{2} \sum_m (\hat{y}_m - y_m)^2}{\partial a_k^{(L)}}$ และหลังจากหาอนุพันธ์จะได้

$$\delta_k^{(L)} = \hat{y}_k - y_k. \quad (3.29)$$

ที่ชั้นซ่อน $q < L$ ตัวแปรต้น $a_j^{(q)}$ จะส่งผลต่อ E_n ผ่านโนหนดต่าง ๆ ในชั้นถัดไป (รูปที่ 3.17) ดังนั้น จากกฎลูกโซ่ของการหาอนุพันธ์

$$\delta_j^{(q)} = \frac{\partial E_n}{\partial a_j^{(q)}} = \sum_m \frac{\partial E_n}{\partial a_m^{(q+1)}} \cdot \frac{\partial a_m^{(q+1)}}{\partial a_j^{(q)}}. \quad (3.30)$$

พจน์หน้า $\frac{\partial E_n}{\partial a_m^{(q+1)}} = \delta_m^{(q+1)}$ มาจากชั้นถัดไป. พจน์หลังคำนวนได้โดยเขียน $a_m^{(q+1)}$ จากสมการ 3.16 และ 3.17 และหาอนุพันธ์ ซึ่งผลลัพธ์คือ $\frac{\partial a_m^{(q+1)}}{\partial a_j^{(q)}} = w_{mj}^{(q+1)} \frac{\partial h(a_j^{(q)})}{\partial a_j^{(q)}}$. เมื่อแทนทั้งสองพจน์นี้ในสมการ 3.30 จะได้ว่า สำหรับชั้นซ่อน ($q < L$) แล้ว

$$\delta_j^{(q)} = h'(a_j^{(q)}) \cdot \sum_m \delta_m^{(q+1)} \cdot w_{mj}^{(q+1)} \quad (3.31)$$

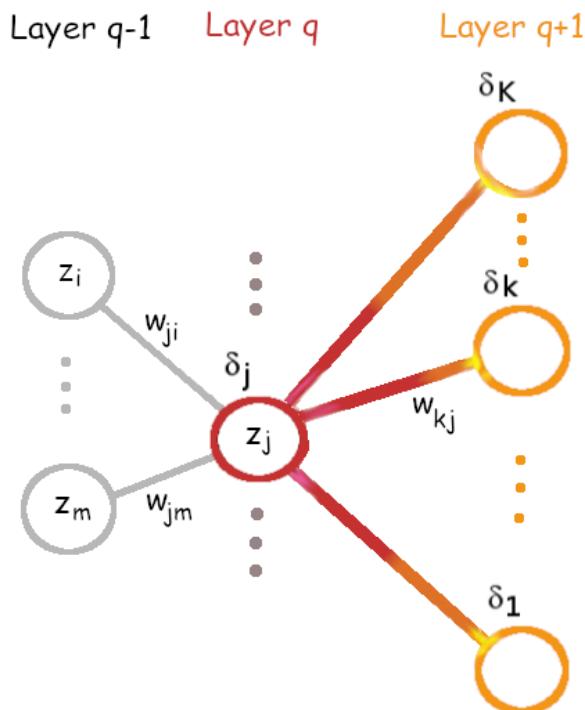
เมื่อ $h'(a_j^{(q)}) = \frac{\partial h(a_j^{(q)})}{\partial a_j^{(q)}}$ คืออนุพันธ์ของฟังก์ชันกระต้น ที่ค่า $a_j^{(q)}$. หมายเหตุ $h'(a_j^{(q)})$ เป็นอนุพันธ์ของฟังก์ชันกระต้นของชั้น q^{th} และบางครั้ง อาจใช้สัญกรณ์ $h_q'(a_j^{(q)}) = \frac{\partial h_q(a_j^{(q)})}{\partial a_j^{(q)}}$ ในกรณีที่อาจสับสน.

ทั้งหมดที่อภิปรายมา สรุปได้ว่า เกรเดียนต์ของโครงข่ายประสาทเทียม $\nabla_{\Theta} E_n$ สามารถหาได้โดยกระบวนการดังนี้

1. ทำการคำนวนไปข้างหน้า (forward propagation หรือ forward pass) โดยคำนวนสมการ 3.15 ถึง 3.18. สิ่งที่ได้คือ ค่าที่ทำนาย \hat{y} กับ ค่าตัวแปรต้นและผลลัพธ์การกระต้นต่าง ๆ $a^{(q)}$ และ $z^{(q)}$ สำหรับ $q = 1, \dots, L$.
2. คำนวนค่า $\delta_k^{(L)}$ สำหรับทุก ๆ โนหนดເອົາຕົກ $k = 1, \dots, K$ ตามสมการ 3.29. สิ่งที่ได้คือ $\boldsymbol{\delta}^{(L)}$.
3. คำนวนค่า $\delta_j^{(q)}$ สำหรับทุก ๆ โนหนด ในทุก ๆ ชั้นซ่อน $j = 1, \dots, M_q; q = 1, \dots, L-1$ ตามสมการ 3.31. สิ่งที่ได้คือ $\boldsymbol{\delta}^{(q < L)}$.
4. คำนวนค่าอนุพันธ์ที่ต้องการ 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 $\nabla_{\Theta} E_n$ ที่ได้สามารถนำไปใช้กับขั้นตอนวิธีการหาค่าที่สุด (optimization algorithm) เช่น วิธีลงเกรเดียนต์ที่อภิปรายในหัวข้อ 2.3 ได้. กระบวนการหาเกรเดียนต์ $\nabla_{\Theta} E_n$ ที่ได้อภิปรายมานี้คือ วิธีการแพร่กระจายย้อนกลับ (error backpropagation หรือ backpropagation) คณของ

ทีรูเมลาร์ต [170] เสนอในปี ค.ศ. 1986 และเป็นการค้นพบที่พื้นพื้นความสนใจในโครงข่ายประสาทเทียม กลับมา หลังจากกว่าทศวรรษของหน้าหน่าวของปัญญาประดิษฐ์.



รูปที่ 3.17 ผลจากโหนดที่ส่งผ่านโหนดอื่น ๆ ในชั้นคำนวนถัดไป. ภาพแสดงการเชื่อมต่อ และเน้นผลของโหนด j^{th} ในชั้น q^{th} ที่มีต่อค่าผลผลิต โดยผ่านโหนดต่าง ๆ ในชั้น $(q+1)^{th}$. ค่า a_j (ไม่ได้แสดงในภาพ) ส่งผลผ่าน z_j ซึ่งส่งผลต่อโหนดต่าง ๆ ในชั้นถัดไปผ่านค่าน้ำหนักของการเชื่อมต่อ.

เกี่ยวกับความรู้จิตและการเรียนรู้ (เรียบเรียงจาก [217], [1], และ [154])

“ทุกสิ่งเริ่มที่จิต นำโดยจิต และสร้างโดยจิต” ธรรมบท[1]

จิตคือสภาวะเชิงการรับรู้และเชิงสติปัญญา ซึ่งรวมถึง สติรู้ตัว การรับรู้สัมผัส ความรู้สึก อารมณ์ การคิด การตัดสินใจ การจำจำ การรับประสารณ์ การเรียนรู้ และการตอบสนองต่อสภาวะแวดล้อม. ส่วนประกอบของจิตนี้ อาจจัดออกได้เป็น 4 หมวด. หมวดหนึ่ง วิญญาณ (vijnana) ซึ่งคือสติรู้ตัว (consciousness). หมวดสอง สัญญา (samjana) ซึ่งคือการรับรู้ (perception) ผ่านสัมผัสทางการมองเห็น สัมผัสทางการได้ยิน สัมผัสทางการได้กลิ่น สัมผัสทางการรับรส สัมผัสทางกาย และสัมผัสมากจากจิตเอง. จิตเองก็มีสัมผัสเช่นกัน ดังตัวอย่างของอาการแขนขาลวง (phantom limbs) ที่เกิดในผู้ที่เสียแขนขาไป แต่ภายหลังยังรู้สึกคันหรือเจ็บที่แขนหรือขาที่ไม่มีอยู่. แม้ว่าสาเหตุของอาการนี้ยังไม่มีคำอธิบายที่ทางการแพทย์ยอมรับร่วมกันอย่างกว้างขวาง แต่นายแพทย์วิลาyanur รามาฉันทรัน (Vilayanur S. Ramachandran) อธิบายว่า สัมผัสที่รู้สึกนั้นมาจากประสาทส่วนที่เคยทำงานกับแขนหรือขาส่วนนั้นขาดสัญญาณที่เคยได้รับ และอาจส่งสัญญาณออกมากหัว ที่ไม่ได้รับสัญญาณรับรู้จริง ๆ. จากทฤษฎีนี้ นายแพทย์รามาฉันทรันเสนอวิธีการบำบัดอาการแขนขาลวง โดยออกแบบกระบวนการให้ผู้มีอาการได้ฝึกประสาทรับรู้ใหม่ ซึ่งพบว่าได้ผลดีมาก. หากมองจากมุมมองทางวิชวกรรม สัญญาณที่ขาดหายไปจากแขนขาที่เสียไปนั้น อาจให้ผลในลักษณะคล้ายการที่วงศิริรับอินพุตมาจากข้าวปลายที่ปล่อยลอยอยู่ ซึ่งข้าว

ปลายที่ปล่อยลอยอาจรับสัญญาณรบกวนเข้ามาแทนได้ การฝึกประสาทรับรู้ใหม่ ก็อาจคล้ายการต่อขึ้นปลายนั้นเข้ากับสายสัญญาณเดิมอีน เป็นไม่ให้มายลอยที่จะรับสัญญาณรบกวนเข้ามา หมวดสาม เวทนา (vedana) ซึ่งคือความรู้สึก (feeling) อารมณ์ ความชอบ ความไม่ชอบ ความวางแผน และ หมวดสี่ สังขาร (sankhara) ซึ่งคือการคิดและกระบวนการเชิงสติ ปัญญาอื่น ๆ (mental activity) ได้แก่ การตัดสินใจ การตอบสนองต่อสภาวะแวดล้อม การจดจำ การรับประสบการณ์ และ การเรียนรู้.

นักประสาทวิทยาด้านสติปัญญาชั้นนำ รีเบกก้า แซก [177] เชื่อว่า รูปแบบของกิจกรรมทางไฟฟ้าของเซลล์ประสาท เกี่ยวข้องโดยตรงกับจิตของเรา เพียงแต่ว่างการวิทยาศาสตร์ยังไม่รู้อะไรมากเกี่ยวกับจิตและความสัมพันธ์ระหว่างรูปแบบ ของกิจกรรมทางไฟฟ้าและจิต

จากมุมมองของกิจกรรมทางไฟฟ้า การเรียนรู้ก็เหมือนการปรับเปลี่ยนวงจรหรือเปลี่ยนการเชื่อมต่อภายในวงจร ซึ่ง ส่งผลให้เกิดการปรับเปลี่ยนพฤติกรรมของกิจกรรมทางไฟฟ้า. สภาพพลาสติกของระบบประสาท (synaptic plasticity) คือ ความสามารถของระบบประสาทที่สามารถเพิ่มหรือลดความแข็งแรงของการเชื่อมต่อสัญญาณประสาทระหว่างเซลล์ได้. สภาพพลาสติกของระบบประสาทนี้เชื่อว่าเป็นคุณสมบัติที่อยู่เบื้องหลังความสามารถในการจำและการเรียนรู้ของสมอง. กลไกนี้เปรียบเทียบได้กับการปรับค่าน้ำหนักหรือการฝึกโครงข่ายประสาทเทียม แต่ประเด็นหนึ่งที่ต่างกันก็คือ การเปลี่ยนค่าน้ำหนักของโครงข่ายประสาทเทียมจะทำเฉพาะในขั้นตอนการฝึกโครงข่ายประสาทเทียม และค่าน้ำหนักที่ได้แล้วจะถูก ตรึงให้คงค่าเหล่านั้นไว้คงที่ขณะใช้งาน. แต่ระบบประสาท(ทางชีวภาพ)จะเปลี่ยนแปลงตัวเองตลอดเวลา เปลี่ยนขณะเรียนรู้ เปเลี่ยนขณะคิด เปเลี่ยนขณะทำงานต่าง ๆ เปเลี่ยนขณะไม่ได้ทำงาน เปเลี่ยนขณะเล่น เปเลี่ยนขณะทำสิ่งที่เป็นโภช เนื่อง เมื่อสิ่งที่เรา คาดหวังไม่ได้ดังใจ และเราไม่ชอบใจ ถ้าเราเลือกที่จะกรร สมองจะเรียนรู้การตอบสนองนั้น และ เมื่อเราทำแบบนั้นบ่อย ๆ เราจะกลายเป็นคนที่กรรง่าย หรือกล่าวอย่างชัดเจนก็คือ เราฝึกสมองของเราให้เก่งที่จะอยู่ในสภาวะอารมณ์กรรนั้นเอง.

กลไกเบื้องหลังการส่งสัญญาณของระบบประสาท กล่าวโดยคร่าว ๆ ก็คือ เมื่อสัญญาณจากนิวเคลียสของเซลล์ประสาทเดินทางถึงปลายแอகซอนซึ่งเป็นปลายสำหรับส่งสัญญาณออก สัญญาณซึ่งถ่ายทอดมาในรูปความต่างศักดิ์จะทำให้ห้องไอออนที่ควบคุมด้วยแรงดันไฟฟ้า (voltage-gated ion channel) ของปลายแอกซอนเปิดออก ทำให้แคลเซียมไอออน (calcium ion สัญลักษณ์ Ca^{2+}) ซึ่งอยู่ในของเหลวรอบ ๆ เซลล์ ไหลเข้าสู่ปลายแอกซอน. แคลเซียมไอออนที่เข้าสู่ปลายแอกซอนจะทำปฏิกิริยากับโปรตีนและเอนไซม์ภายในแอกซอน ซึ่งส่งผลให้เกิดการปล่อยสารสื่อประสาทออกม. สารสื่อประสาทที่ออกม. จะมีบางโมเลกุลที่ได้จับกับรีเซปเตอร์ที่ปลายเดนไดร์ต ซึ่งปลายเดนไดร์ตคือปลายประสาทที่นำสัญญาณเข้าสู่เซลล์ประสาทด้วยที่จะรับสัญญาณ. เมื่อรีเซปเตอร์จับกับสารสื่อประสาทแล้ว การจับตัวกันทำให้รีเซปเตอร์เปลี่ยนโครงสร้าง ซึ่งจะเปิดช่องให้ไอออนบางไหหลเข้าสู่เดนไดร์ต เมื่อไอออนบางไหหลเข้าสู่เดนไดร์ตจะทำให้ความต่างศักดิ์ของเดนไดร์ตจุดนั้นเปลี่ยนไป ซึ่งความต่างศักดินี้เองเป็นสัญญาณที่จะถ่ายทอดต่อไปยังนิวเคลียสของเซลล์ประสาท.

สารสื่อประสาทเป็นกลไกหลักในการช่วยส่งสัญญาณประสาทข้ามเซลล์ประสาท แต่ตัวสารสื่อประสาทเองไม่ได้ถูกส่งเข้าไปในเดนไดร์ตของเซลล์ประสาทด้วยรับ. มันทำหน้าที่สเมื่อนช่วยเปิดประตูให้ไอออนบางไหเข้าไปในปลายเดนไดร์ตดังที่ได้อธิบายไปข้างต้น. หลังจากสารสื่อประสาทจับกับรีเซปเตอร์ได้สักพัก มันจะหลุดออกม. สารสื่อประสาททั้งที่พึงหลุดมาจากการจับกับรีเซปเตอร์และที่ยังไม่ได้จับกับรีเซปเตอร์เลยจะถูกกำจัดออกไปโดยกลไกหลาย ๆ ชนิด เช่น การทำลายทิ้งหรือ การที่ปลายแอกซอนดึงสารสื่อประสาทเหล่านี้กลับเข้าไปภายในเพื่อนำกลับไปใช้ใหม่.

การเรียนรู้หรือการปรับความแข็งแรงของการเชื่อมต่อสัญญาณระหว่างเซลล์ประสาท เกี่ยวข้องกับการปรับความสามารถในการรับส่งสัญญาณประสาทระหว่างเซลล์. เมื่อกล่าวถึงสัญญาณประสาทโดยละเอียดขึ้น สัญญาณประสาทจะถูกส่งในหลายรูปแบบ ขณะที่สัญญาณประสาทส่งผ่านเส้นทางจากนิวเคลียสของเซลล์ประสาทด้วยตัวหนึ่งไปสู่นิวเคลียสของเซลล์ประสาทอีกด้วยหนึ่ง มีการเปลี่ยนรูปแบบอย่างครั้ง. เมื่อเซลล์ประสาทอยู่ในสถานะถูกกระตุ้น นิวเคลียสของเซลล์จะส่งสัญญาณออกม. ในรูปความถี่ของพัลส์. นั่นคือ นิวเคลียสของเซลล์ประสาทจะส่งสัญญาณในลักษณะพัลส์ (pulse) ซึ่งมักเรียกว่าศักยะงาน (action potential) เช่น ความต่างศักย์ภายในเซลล์ประสาทจะมีค่าประมาณ -70 มิลลิโวลต์เมื่อ

เทียบกับจุดภายนอกเซลล์ แต่เมื่อมีศักยะงานเกิดขึ้น ค่าความต่างศักย์นี้เพิ่มขึ้นอย่างรวดเร็วจากค่าพักตัวที่ประมาณ -70 มิลลิโวลต์ ไปสูงสุดที่ประมาณ 40 มิลลิโวลต์ หลังจากนั้นจะลดค่าลงอย่างรวดเร็วมาที่ราว ๆ -90 มิลลิโวลต์ และกลับมาจนที่ค่าพักตัวที่ประมาณ -70 มิลลิโวลต์ โดยที่ศักยะงานแต่ละลูกจะบานานประมาณ 4 มิลลิวินาที. ความถี่หรือจำนวนศักยะงานต่อวินาที จะขึ้นกับความแรงของการกระตุ้น เช่น เซลล์โอล์แฟกตอร์คortex เท็กซ์ไฟรามิดอล (*Olfactory Cortex pyramidal cell*) ที่ทำงานเกี่ยวกับการรับรู้กลิ่น จะส่งศักยะงานอ่อนๆ ที่ความถี่ประมาณ 0.8 ถึง 2.0 ลูกต่อวินาที ขณะที่ปกติ แต่จะส่งศักยะงานความถี่ประมาณ 4 ถึง 11 ลูกต่อวินาที ขณะที่เรากำลังดมอะไรอยู่[138]. สัญญาณในรูปความถี่นี้ได้มีการศึกษามาตั้งแต่ ค.ศ. 1926 ที่ เอเดรียนและโซตเตอมัน[2] รายงานการศึกษาการกระตุ้นและสัญญาณไฟฟ้าเซลล์ประสาท โดยใช้ตุ้มน้ำหนักดึงกล้ามเนื้อของกบและวัดสัญญาณไฟฟ้าจากเนื้อเยื่อประสาทของมัน และพบความสัมพันธ์ที่ชัดเจน ระหว่างค่าน้ำหนักที่ยืดกล้ามเนื้อออกร (ตัวแทนของปริมาณความแรงของการกระตุ้น) กับความถี่ของศักยะงานที่เกิดขึ้น. ศักยะงานนี้คือสัญญาณที่ถูกส่งออกจากนิวเคลียลส์ผ่านไปเป็นปลายแทกซอน สัญญาณที่ส่งผ่านออกจากปลายแทกซอนของเซลล์ตัวส่งเข้าไปสู่ปลายเดนไดร์ตของเซลล์ตัวรับจะส่งผ่านกลไกของสารสื่อประสาท และปลายเดนไดร์ตรับสัญญาณประสาทเข้ามาในรูประดับความต่างศักย์ระหว่างภัยในปลายเดนไดร์ตและภายนอก. รูปแบบที่เปลี่ยนไปของสัญญาณประสาทจากนิวเคลียลส์ของตัวส่งไปจนถึงปลายแทกซอน (ในรูปศักยะงาน) ผ่านไซแนปซ์ (ในรูปสารสื่อประสาท) และรับเข้าสู่ปลายเดนไดร์ตไปจนถึงส่งเข้าไปสู่นิวเคลียลส์ของตัวรับ (ในรูประดับความแรงและความต่างศักย์) รูปแบบเหล่านี้ เกี่ยวข้องสัมพันธ์กับทฤษฎีที่ใช้อธิบายกระบวนการเรียนรู้ของเซลล์ประสาท.

ทฤษฎีที่อธิบายการปรับความแข็งแรงของการเชื่อมต่อสัญญาณระหว่างเซลล์ประสาท กล่าวถึง กลไกหลัก 2 กลไก. นั่นคือ การเพิ่มความแข็งแรงเชิงประสาทระยะยาว (*long-term synaptic potentiation* คำย่อ LTP) และ การลดถอยความแข็งแรงเชิงประสาทระยะยาว (*long-term synaptic depression* คำย่อ LTD). จากงานศึกษาการเชื่อมต่อของเซลล์ในยีปีเพคปัส โดยเฉพาะ ไซแนปซ์ที่เชื่อมต่อระหว่างเซลล์ประสาทในบริเวณซีเอ3 (CA3) ที่ส่งแทกซอน ซึ่งเรียกว่า แซฟเฟอร์คอลเลอร์ (Schaffer collaterals) ไปเชื่อมต่อกับเซลล์ประสาทในบริเวณซีเอ1 สรุปว่า เมื่อมีการกระตุ้นด้วยสัญญาณประสาทความถี่สูงผ่านไซแนปซ์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดร์ตของไซแนปส์นั้นจะมีค่าเพิ่มขึ้นมาก และค่านั้นจะคงอยู่เป็นเวลานาน (หลายนาทีหรือหลายวัน หลังจากการกระตุ้นนั้น) โดยค่าความต่างศักย์ที่ปลายเดนไดร์ตที่เชื่อมต่อไซแนปซ์อื่นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การเพิ่มความแข็งแรงเชิงประสาทระยะยาว. ประเด็นสำคัญคือ (1) มีการเปลี่ยนแปลงความแข็งแรงของไซแนปส์ระยะยาวเกิดขึ้น (2) ผลการเปลี่ยนแปลงความแข็งแรงขึ้นกับความถี่ (3) ผลการเปลี่ยนแปลงเกิดขึ้นเฉพาะตัวของไซแนปซ์.

ในทางกลับกัน เมื่อมีการกระตุ้นตัวยังสัญญาณประสาท ความถี่ต่ำผ่านไซแนปซ์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดร์ตของไซแนปส์นั้นจะมีค่าลดลงมาก และค่านั้นก็คงอยู่เป็นเวลานาน โดยค่าความต่างศักย์ที่ปลายเดนไดร์ตที่เชื่อมต่อไซแนปส์อื่นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การลดถอยความแข็งแรงเชิงประสาทระยะยาว. ประเด็นสำคัญคือ (1) มีการเปลี่ยนแปลงความแข็งแรงของไซแนปส์ระยะยาวเกิดขึ้น (2) ผลการเปลี่ยนแปลงความแข็งแรงขึ้นกับความถี่ (3) ผลการเปลี่ยนแปลงเกิดขึ้นเฉพาะตัวของไซแนปซ์.

กลไกเบื้องหลังนั้นอธิบายว่า ไซแนปส์ของแซฟเฟอร์คอลเลอร์ ทำงานผ่านสารสื่อประสาทกลูตาเมท และปลายประสาทของเซลล์ตัวรับที่ซีเอ1 มีรีเซปเตอร์ที่ทำงานกับกลูตาเมทอยู่ 2 ชนิด คือ แอมປารีเซปเตอร์ (*AMPA receptor*) และ เอนเอมดีเอรีเซปเตอร์ (*NMDA receptor*). เมื่อปลายประสาทของเซลล์ที่ซีเอ3 จะส่งสัญญาณกระตุ้นผ่านไซแนปซ์ มันจะปล่อยสารสื่อประสาท กลูตาเมท ออกมาน. เมื่อกลูตาเมทจับกับแอมປารีเซปเตอร์ แอมປารีเซปเตอร์จะเปิดออก และด้วยคุณสมบัติของแอมປารีเซปเตอร์ โฉเดียมไอออนจะไหลเข้าสู่ปลายเดนไดร์ตตัวรับที่ซีเอ1 แคลเซียมไอออนบางส่วนก็อาจไหลเข้าได้บ้างแต่ไม่มาก. แอมປารีเซปเตอร์ที่เปิดรับโฉเดียมไอออนแล้วชักพักก็จะปิดลง. เมื่อกลูตาเมทจับกับเอนเอมดีเอรีเซปเตอร์ เอนเอมดีเอรีเซปเตอร์จะเปิดออกนาน กดต่อเนื่องดี เอรีเซปเตอร์จะมีแมgnีเซียมไอออนปิดช่องอยู่ ทำให้ไอออนยังไม่สามารถไหลผ่านได้. หลังจากโฉเดียมไอออนผ่านแอมປารีเซปเตอร์เข้าสู่เซลล์ซีเอ1 สักพักบริเวณโฉเดียมไอออนจะถูกสูบออกโดยกลไกของโฉเดียม-โพแทสเซียมปั๊ม (*sodium-potassium pump*) ซึ่งเป็นเอนไซม์ที่ทำงานรักษาสมดุลของเซลล์.

การเพิ่มความแข็งแรงเชิงประสาทระยะยาว หากการกระตุ้น มีความถี่สูงพอที่โซเดียมไอออนที่ไหลเข้ามาจะสะสมได้ (ความถี่สูงพอที่จะสะสมโซเดียมไอออน ที่เหลือจากการสูบออกของโซเดียม-โพแทสเซียมปั๊ม) ปริมาณโซเดียมไอออนที่เพิ่มขึ้นมาก จะเพิ่มระดับแรงดันไฟฟ้าสถิตย์ขึ้น และเมื่อไฟฟ้าสถิตย์มากพอ มันจะสร้างแรงที่จะผลักแมกนีเซียมไอออน ที่ปิดเอนเอมดีเอรีเซปเตอร์ออกไปได้ เมื่อแมกนีเซียมหลุดออกไป แคลเซียมไอออนก็สามารถผ่านเข้ามาทางเอนเอมดีเอรีเซปเตอร์ได้ แคลเซียมไอออนที่ไหลเข้าปริมาณมากจะช่วยเพิ่มค่าความต่างศักย์ที่ได้รับที่ปลาย денไดร์ตี้น์ไป นอกจากนั้นแคลเซียมไอออนปริมาณมากจะจับกับโปรตีนคีแนส (Protein kinases) ซึ่งส่งผลให้เกิดการสร้างแอมปารีเซปเตอร์และติดตั้งแอมปารีเซปเตอร์ที่สร้างใหม่ เข้าไปที่ปลายเชื่อมประสาท ทำให้จำนวนแอมปารีเซปเตอร์เพิ่มขึ้น ผลการเพิ่มแอมปารีเซปเตอร์จากกระบวนการนี้ จะคงอยู่เพียงแค่เวลาสั้น ๆ ไม่กี่ชั่วโมงเท่านั้น แต่หากมีแคลเซียมไอออนไหลเข้ามาในปริมาณมากเป็นระยะเวลานานพอ (มีการกระตุ้นด้วยสัญญาณประสาทความถี่สูงเป็นระยะเวลานาน) จะส่งผลต่อเนื่องไปจนทำให้เกิดการเพิ่มข้อจำกัดการลอกหรือหัลตีเย็นเอ (transcription factor) ซึ่งส่งผลต่อการแสดงออกของยีน (gene expression) และทำให้เกิดการสร้างโปรตีนที่ทำให้เกิดหั้งแอมปารีเซปเตอร์ใหม่เพิ่มขึ้น และ โกรหแฟคเตอร์ (growth factor) ที่จะไปทำให้เกิดการสร้างไซแนปส์ใหม่เพิ่มขึ้น ซึ่งผลจากการนี้จะยาวนานและคงทนมาก.

การลดถอยความแข็งแรงเชิงประสาทระยะยาว สำหรับการกระตุ้นที่มีความถี่ต่ำ จะส่งผลให้ปริมาณของแคลเซียมไอออนอยู่ในระดับต่ำ ปริมาณของแคลเซียมไอออนที่อยู่ในระดับต่ำจะไปกระตุ้นการทำงานของเอนไซม์ฟอสฟेटเตส (phosphatase) ซึ่งส่งผลไปลดจำนวนแอมปารีเซปเตอร์ที่ทำงานได้ลง.

การเพิ่มความแข็งแรงเชิงประสาทระยะยาว และ การลดถอยความแข็งแรงเชิงประสาทระยะยาว ต่างก็มีปัจจัยมาจากการปริมาณแคลเซียมไอออนในปลายไซแนปส์ตัวรับ โดย ปริมาณแคลเซียมไอออนในระดับสูงจะทำให้เกิดการเพิ่มความแข็งแรงเชิงประสาทระยะยาว และ ปริมาณแคลเซียมไอออนในระดับต่ำจะทำให้เกิดการลดถอยความแข็งแรงเชิงประสาทระยะยาว. ระดับที่เป็นขีดแบ่งระหว่างการเพิ่มและการลดอยู่นี้ เป็นจุดที่เปลี่ยนแปลงตามสภาพของเซลล์ในลักษณะที่ช่วยรักษาสมดุลย์ (ทฤษฎีบีซีเอ็ม BCM theory[16]) ได้แก่ การเปลี่ยนแปลงในลักษณะการบ้อนกลับเชิงลบ (negative feedback) เช่น เมื่อไซแนปส์อยู่ในภาวะการเพิ่มความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้จะสูงขึ้น เพื่อช่วยลดความเสี่ยงของการเพิ่มมหาศาลลง และ เมื่อไซแนปส์อยู่ในภาวะการลดถอยความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้น่าจะลดลงเพื่อช่วยลดความเสี่ยงของการลดถอยจนสูญเสียการเชื่อมต่อ.

จากสภาพพลาสติกของระบบประสาทและทฤษฎีบีซีเอ็ม เราอาจกล่าวได้ว่า การเรียนรู้ที่มีประสิทธิภาพคือ การเรียนรู้ต่อเนื่อง สลับกับการหยุดพักผ่อน เพื่อให้เกิดการกระตุ้นประสาทต่อเนื่องยาวนานเพียงพอ ที่จะทำให้เกิดการสร้างการเชื่อมต่อใหม่ และ หยุดพักเพื่อให้ระดับขีดแบ่งปรับตัวลงมา ทำให้การสร้างการเชื่อมต่อใหม่ทำได้ง่ายขึ้น เพราะ การเรียนรู้ต่อเนื่องเป็นเวลานานเกินไป จะเพิ่มระดับขีดแบ่งซึ่งจะทำให้การเข้าสู่ภาวะการเพิ่มความแข็งแรงเชิงประสาทระยะยาวทำได้ยากขึ้น.

“Never go to excess, but
let moderation be your guide.”

---Marcus Tullius Cicero

“ทำสิ่งใดอย่างมากเกินไป
ให้ความพอดีเป็นตัวชี้ทาง.”

—มาร์คัส ทูลลิอุส ซิเชโน

การฝึกแบบหมู่กับการฝึกแบบออนไลน์

ค่าเกรเดียนต์ที่คำนวณได้จากการแพร่กระจายย้อนกลับ สามารถนำมาช่วยการฝึกโครงข่ายประสาทเทียมได้. แต่สมการเกรเดียนต์ของโครงข่ายประสาทเทียม ต่างจากสมการเกรเดียนต์ของฟังก์ชันพหุนาม. คำตอบของ เกรเดียนต์ฟังก์ชันพหุนามเป็นศูนย์ สามารถแก้สมการ เพื่อคำนวณหาค่าพารามิเตอร์ของ ฟังก์ชันพหุนามได้ทันที (คำตอบ สามารถเขียนอยู่ในรูปแบบบิดทางคณิตศาสตร์ได้). แต่สำหรับโครงข่ายประสาทเทียม คำตอบ ไม่สามารถหาได้โดยตรงจากการแก้สมการคณิตศาสตร์ และต้องใช้ขั้นตอนวิธีมาช่วย. ขั้นตอนวิธีการหาค่าดีที่สุด เช่น วิธีลิงเกรเดียนต์ สามารถนำมาช่วยแก้ปัญหานี้ได้.

การนำขั้นตอนวิธีการหาค่าดีที่สุดไปใช้กับการฝึกแบบจำลอง มีประเด็นที่น่าสนใจคือ จังหวะการปรับค่าพารามิเตอร์ ควรทำบ่อยขนาดไหน. ตัวอย่างเช่น วิธีลิงเกรเดียนต์ ปรับค่าพารามิเตอร์ด้วยสมการ 2.46 ที่นำมาเขียนในพจน์ของโครงข่ายประสาทเทียม

$$\Theta^{(i)} = \Theta^{(i-1)} - \alpha \cdot \nabla_{\Theta} E(\Theta^{(i-1)}) \quad (3.32)$$

เมื่อ $\Theta^{(i)}$ คือค่าพารามิเตอร์ของแบบจำลอง ที่ได้จากการคำนวณรอบที่ i^{th} . ค่าสเกลาร์ α คือขนาดก้าว ซึ่งสำหรับการฝึกแบบจำลอง นิยมเรียกว่า อัตราเรียนรู้ (learning rate). เวกเตอร์ $\nabla E_{\Theta}(\Theta^{(i-1)})$ เป็นค่าเกรเดียนต์ ณ ค่าพารามิเตอร์ก่อนการคำนวณรอบที่ i^{th} . การคำนวณแต่ละรอบ จะเรียกว่า สมัย (epoch).

หากต้องการตรวจสอบว่า จะได้การปรับค่าน้ำหนัก ตามสมการ 3.32. การปรับค่าพารามิเตอร์ ที่ปรับทีเดียว ในแต่ละสมัย โดย การปรับ ใช้ค่าเฉลี่ยของเกรเดียนต์ที่คิดจากจุดข้อมูลฝึกทุกจุด. การปรับค่าน้ำหนักในลักษณะนี้ จะเรียกว่า การฝึกแบบหมู่ (batch training).

แต่จากสมการ 3.22 นั้นเท่ากับ $\nabla_{\Theta} E = \frac{1}{N} \sum_n \nabla_{\Theta} E_n$ เมื่อ N คือจำนวนจุดข้อมูล ในชุดข้อมูลฝึก ดังนั้น

$$\Theta^{(i)} = \Theta^{(i-1)} - \alpha \cdot \frac{1}{N} \sum_{n=1}^N \nabla_{\Theta} E_n(\Theta^{(i-1)}). \quad (3.33)$$

หากตัดแปลงเล็กน้อย โดยการปรับค่าพารามิเตอร์ให้ถี่ขึ้น คือการปรับค่าพารามิเตอร์ แต่ละครั้ง สำหรับแต่ละจุดข้อมูล ดังแสดงในสมการ 3.34 และจะปรับค่าพารามิเตอร์ N ครั้งในแต่ละสมัย. การปรับค่าพารามิเตอร์ในลักษณะเช่นนี้ จะเรียกว่า การฝึกแบบออนไลน์ (online training).

$$\Theta^{(i)} = \Theta^{(i-1)} - \frac{\alpha}{N} \cdot \nabla_{\Theta} E_n(\Theta^{(i-1)}). \quad (3.34)$$

การฝึกแบบออนไลน์ จะสามารถรับข้อมูลเข้ามาฝึกเพิ่มได้ตลอด บางครั้งจึงอาจถูกเรียกว่า การฝึกในโหมดเพิ่ม (*training in an incremental mode*).

โดยทั่วไป การฝึกแบบหมู่จะทำได้เร็วกว่า แต่ก็ต้องการหน่วยความจำมากกว่าการฝึกแบบออนไลน์. สำหรับคุณภาพการฝึก เฮียกิน[83] ได้อภิปรายถึงข้อดีข้อเสียว่า การฝึกแบบหมู่ จะให้ประสิทธิภาพค่าเกรเดียนต์ที่แม่นยำกว่า ดังนั้น เมื่อทำงานกับวิธีลิงเกรเดียนต์ แล้วจะได้ค่าพารามิเตอร์ที่ดีกว่า.

นอกจากนั้น การฝึกแบบหมู่ มีธรรมชาติเข้ากับการคำนวณแบบขนานมากกว่า และสามารถนำการประมวลผลแบบขนาน (parallel processing) มาช่วยการประมวลผลได้ง่ายกว่า เมื่อมองจากมุมมองทางสถิติศาสตร์ การฝึกแบบหมู่ เป็นรูปแบบการอนุมานทางสถิติ (statistical inference) อย่างหนึ่ง ดังนั้นการฝึกแบบหมู่ จะเหมาะสมกับงานการหาค่าคาดถอย. แต่ข้อเสียของการฝึกแบบหมู่ ก็คือความต้องการใช้หน่วยความจำปริมาณมาก (เพราะว่าใช้ข้อมูลทั้งหมดที่เดียว ใน การคำนวณแต่ละสมัย).

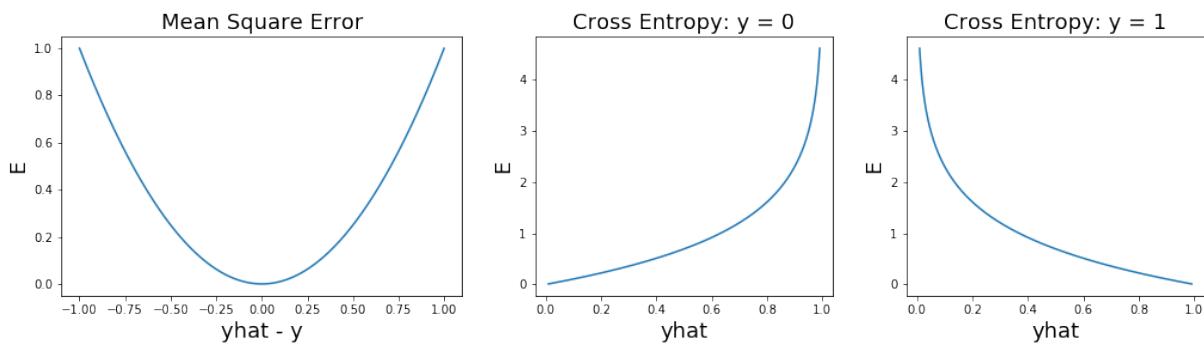
ข้อดีของการฝึกแบบออนไลน์ เมื่อทำร่วมกับการสลับลำดับข้อมูลในแต่ละสมัย จะช่วยลดความเสี่ยงในการเข้าไปติดอยู่ที่ค่าทำให้น้อยที่สุดทั้งนั้นได้ และเมื่อเทียบกับการฝึกแบบหมู่ การฝึกแบบออนไลน์ ต้องการใช้หน่วยความจำปริมาณน้อยกว่า. นอกจากนี้ในทางปฏิบัติแล้วพบว่า การฝึกแบบออนไลน์ สามารถใช้ประโยชน์จากข้อมูลที่ซ้ำซ้อนกันได้มากกว่าการฝึกแบบหมู่. การฝึกแบบออนไลน์ยังอาจช่วยให้การติดตามการเปลี่ยนแปลงเล็ก ๆ ในข้อมูล ทำได้สะดวกขึ้น โดยเฉพาะอย่างยิ่ง สำหรับข้อมูลที่มีลักษณะไม่คงที่ (nonstationary). แม้ว่าการฝึกแบบออนไลน์ จะทำงานได้ช้ากว่า แต่การฝึกแบบออนไลน์ ก็มีการใช้อย่างแพร่หลาย โดยเฉพาะกับงานการจำแนกรูปแบบ ส่วนหนึ่ง เพราะว่า การฝึกแบบออนไลน์ สามารถขยายขึ้นไปทำงานกับข้อมูลขนาดใหญ่ได้ง่ายกว่า. อย่างไรก็ตาม พัฒนาการต่อมา นิยมใช้การสมการฝึกแบบหมู่ และการฝึกแบบออนไลน์ ที่เรียกว่า การฝึกแบบหมู่เล็ก ดังที่จะอภิปรายในบท 5.

การกำหนดค่าเริ่มต้นในการฝึก. ขั้นตอนวิธีในการฝึกโครงข่ายประสาทเทียม ใช้การปรับปรุงแก้ไขค่าพารามิเตอร์ให้ดีขึ้นเรื่อย ๆ ในแต่ละสมัยฝึก. สมัยฝึกแรกสุด ต้องการค่าเริ่มต้นของพารามิเตอร์. การกำหนดค่าเริ่มต้นของพารามิเตอร์ ที่มักเรียกว่า การกำหนดค่าน้ำหนักเริ่มต้น (weight initialization) วิธีที่นิยมในการกำหนดค่าเริ่มต้น คือการกำหนดค่าเริ่มต้นด้วยการสุ่ม (แบบฝึกหัด 3.12). นอกจากนี้ มีเทคนิคจำนวนมาก สำหรับวิธีการกำหนดค่าเริ่มต้นที่มีประสิทธิภาพ เช่น วิธีเหนี่ยวนวิดโดร์ (Nguyen-Widrow weight initialization[141]) หัวข้อ 5.4 อภิปรายวิธีการกำหนดค่าเริ่มต้นที่มีประสิทธิภาพโดยเฉพาะสำหรับโครงข่ายประสาทเทียมแบบลึก.

โครงข่ายประสาทเทียมสำหรับการจำแนกกลุ่ม

สมการ 3.23 แสดงค่าผิดพลาดกำลังสอง ซึ่งนิยมใช้เป็นฟังก์ชันจุดประสงค์ สำหรับงานการหาค่าคาดถอย. การรู้จักรูปแบบ ที่บ่อยครั้ง มักจะถูกตีกรอบปัญหาเป็นการจำแนกกลุ่ม มีสองภาระกิจหลัก ๆ คือ การจำแนกค่าทวิภาค และการจำแนกกลุ่ม .

การจำแนกค่าทวิภาค. การจำแนกค่าทวิภาค เป็นภาระกิจที่ต้องการทำนายผลลัพธ์ ซึ่งมีรูปแบบที่เป็นไปได้สองแบบ. รูปแบบที่เป็นไปได้ทั้งสอง นั้นมักจะถูกเลือกให้แทนด้วย 1 และ 0 เช่น ปัญหาการทายผลการตรวจข้อมูลเอ็กซ์เต้านมของมวลเนื้อ (แบบฝึกหัด 3.15) ที่ผลทายมีอยู่สองอย่างคือ เนื้อร้าย หรือ ไม่ใช่นื้อร้าย.



รูปที่ 3.18 พฤติกรรมของฟังก์ชันสูญเสีย. ภาพซ้าย แสดงพฤติกรรมฟังก์ชันสูญเสียแบบค่าผิดพลาดกำลังสอง. แกนนอนแสดงผลต่างระหว่างค่าท่านายและค่าเฉลย แกนตั้งแสดงค่าผิดพลาดกำลังสอง. ภาพกลางแสดง ค่าฟังก์ชันสูญเสียแบบクロสเอนไทรปี เมื่อเฉลยมีค่าเป็น 0. แกนนอนแสดงค่าท่านาย. หากค่าท่านายถูกต้อง ค่าสูญเสียจะเป็น 0 แต่หากท่านายผิด ค่าสูญเสียจะสูงมาก และหากท่านายผิดเต็มที่ นั่นคือ $\hat{y} = 1$ ค่าสูญเสียจะเป็นอนันต์ (ไม่สามารถแสดงในภาพ). ภาพขวาแสดง ค่าฟังก์ชันสูญเสียแบบクロสเอนไทรปี เมื่อเฉลยมีค่าเป็น 1. เช่นเดียวกับภาพกลาง ค่าสูญเสียจะสูงมาก เมื่อทายผิด.

การจำแนกค่าทวิภาค (binary classification) ซึ่งต้องการเอาต์พุต $y \in \{0, 1\}$ นิยมใช้ฟังก์ชันกราฟต้นเอาต์พุตเป็นฟังก์ชันซิกโนอยด์. นั่นคือ $h(a) = 1/(1 + e^{-a})$ ซึ่งจะทำให้ค่าที่ท่านาย $\hat{y} \in (0, 1)$. เพื่อให้การฝึกแบบจำลองทำได้มีประสิทธิภาพ ฟังก์ชันจุดประสงค์ สำหรับการจำแนกค่าทวิภาค นิยมใช้ ฟังก์ชันสูญเสียクロสเอนไทรปี (cross entropy loss). ฟังก์ชันสูญเสียクロสเอนไทรปี สำหรับการจำแนกค่าทวิภาค 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 $\hat{y}(\mathbf{x}_n, \Theta)$ คือผลทาย สำหรับจุดข้อมูลที่ n^{th} (ที่มีตัวแปรต้นเป็น \mathbf{x}_n) และใช้ค่าพารามิเตอร์ Θ . สมการ 3.35 มักถูกเขียนย่อเป็น

$$E_n = -y_n \cdot \log(\hat{y}_n) - (1 - y_n) \cdot \log(1 - \hat{y}_n). \quad (3.36)$$

รูปที่ 3.18 แสดงพฤติกรรมของฟังก์ชันสูญเสียชนิดค่าผิดพลาดกำลังสอง เปรียบเทียบกับค่าของฟังก์ชันสูญเสียชนิดクロสเอนไทรปี ในกรณีที่เฉลยเป็น 0 และเฉลยเป็น 1.

การจำแนกกลุ่ม. การจำแนกกลุ่ม เป็นภาระกิจที่ต้องการท่านายกลุ่มของรูปแบบ ตัวอย่างเช่น การรู้จำตัวเลขลายมือ (แบบฝึกหัด 3.16) ที่ต้องการระบุกลุ่มของภาพตัวเลขที่เขียนด้วยลายมือ ว่าอยู่กลุ่มตัวเลขใด ระหว่าง 0 ถึง 9.

การจำแนกกลุ่ม (classification หรือ multi-class classification) ซึ่งต้องการเอาต์พุตที่ระบุคลาสของกลุ่ม ซึ่งฉลาก นิยมกำหนดด้วยรหัสหนึ่งร้อน (one-hot coding) หรือบางครั้งอาจเรียกว่า

one-of-K coding). รหัสหนึ่งร้อน จะใช้ตัวเลข K ตัว ในการระบุกลุ่ม K กลุ่ม. ตำแหน่งของเลขแต่ละตัว แทนกลุ่มที่สนใจ แต่ละกลุ่ม. หากจุดข้อมูล อยู่ในกลุ่มใด ตัวเลขที่อยู่ตำแหน่งของกลุ่มนั้นจะเป็นหนึ่ง และตัวเลขอื่น ๆ จะเป็นศูนย์. นั่นคือ เอาร์พุตในรหัสหนึ่งร้อน $\mathbf{y} = [y_1, \dots, y_K]^T$ โดย $y_k \in \{0, 1\}$ และ $\sum_k y_k = 1$. การจำแนกกลุ่ม นิยมใช้ฟังก์ชันกระตุนเอาร์พุต เป็นฟังก์ชันซอฟต์แมกซ์ (softmax function) ซึ่งคำนวณโดย สมการ 3.37.

$$h(\mathbf{a}) = \left[\frac{e^{a_1}}{\sum_{k=1}^K e^{a_k}} \quad \cdots \quad \frac{e^{a_K}}{\sum_{k=1}^K e^{a_k}} \right]^T \quad (3.37)$$

สมการ 3.37 มักเขียนย่อเป็น

$$h(a_i) = \frac{e^{a_i}}{\sum_{k=1}^K e^{a_k}}. \quad (3.38)$$

ฟังก์ชันซอฟต์แมกซ์ ช่วยให้ค่าที่ทำนาย $\hat{\mathbf{y}} = h(\mathbf{a})$ 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 ฟังก์ชันสูญเสียครอสแอนไทรบี ที่นิยามดังสมการ 3.39 มักถูกเลือกใช้เป็น ฟังก์ชันจุดประสงค์สำหรับการจำแนกกลุ่ม.

$$E_n = -\log (\hat{y}_c(\mathbf{x}_n, \Theta)) \quad (3.39)$$

เมื่อ c คือตัวชี้ของกลุ่มที่เฉลย นั่นคือ $y_c(n) = 1$. สมการ 3.39 มักถูกเขียนเป็น

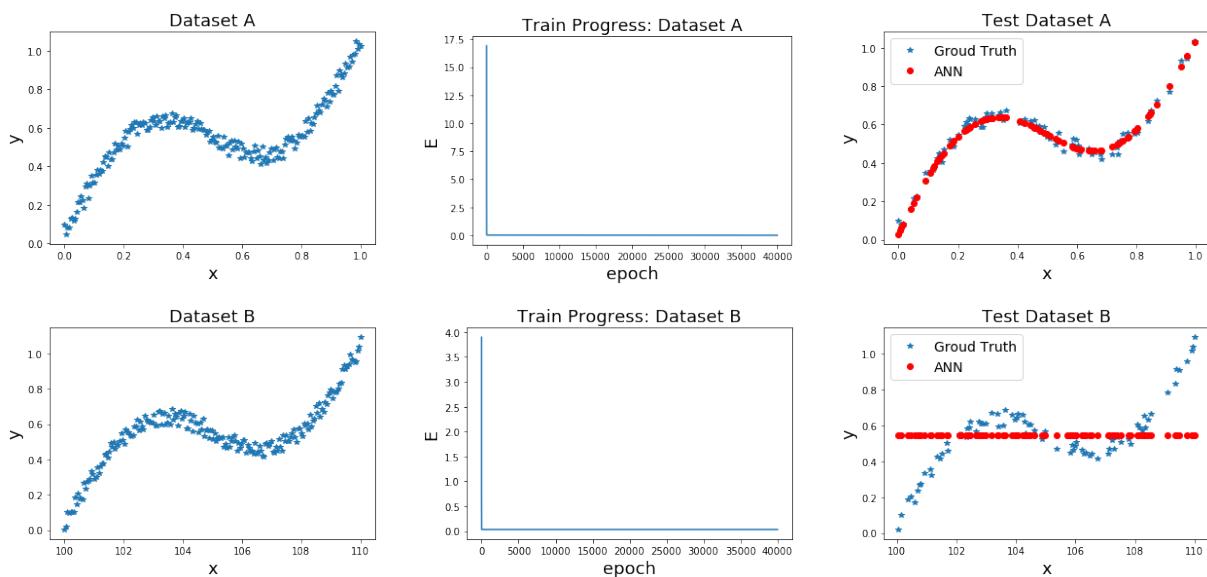
$$E_n = -\sum_{k=1}^K y_k(n) \cdot \log (\hat{y}_k(\mathbf{x}_n, \Theta)) \quad (3.40)$$

เมื่อ K คือจำนวนของกลุ่มทั้งหมด.

ฟังก์ชันสูญเสียครอสแอนไทรบีในสมการ 3.35 และ 3.39 ต่างมาจากการพื้นฐานเดียวกัน แต่ pragmatically ต่างกัน. เพื่อลดความสับสน สมการ 3.35 จะเรียกว่า ฟังก์ชันสูญเสียครอสแอนไทรบีสำหรับการจำแนกค่าทวิภาค และอาจเรียกย่อเป็น ครอสแอนไทรบีทวิภาค (binary cross entropy). สมการ 3.39 เป็นครอสแอนไทรบีสำหรับการจำแนกกลุ่ม และอาจเรียกย่อเป็น ครอสแอนไทรบีพหุกลุ่ม (multi-class cross entropy หรือ categorical cross entropy).

3.4 การประยุกต์ใช้โครงข่ายประสาทเทียม

หัวข้อ 3.3 อภิปรายกลไกที่จำเป็นสำหรับการทำงานของโครงข่ายประสาทเทียม. การประยุกต์ใช้โครงข่ายประสาทเทียม ในทางปฏิบัติ มีประเด็นอื่น ๆ ที่ต้องพิจารณาประกอบ เพื่อให้การฝึก และการใช้งาน



รูปที่ 3.19 โครงข่ายประสาทเทียมกับขนาดของอินพุต. การใช้โครงข่ายประสาทเทียมกับข้อมูลชุดเอ แสดงในแถบบน และ กับข้อมูลชุดบี แสดงในแถบล่าง. จุดข้อมูลต่าง ๆ แสดงในภาพชี้. การดำเนินการกับข้อมูลทั้งสองชุดทำเหมือนกันทุก ประการ ได้แก่ แบ่งข้อมูลออกเป็นชุดฝึกและชุดทดสอบ ฝึกแบบจำลองกับข้อมูลชุดฝึก (ความก้าวหน้าในการฝึก บ่งชี้จาก ค่าผิดพลาดต่อสมัยฝึก แสดงในภาพกลาง) และทดสอบกับข้อมูลชุดทดสอบ. ข้อมูลชุดเอ ได้ค่าผิดพลาดกำลังสองเป็น 0.0007 สำหรับทั้งชุดฝึกและชุดทดสอบ. ข้อมูลชุดบี ได้ค่าผิดพลาดกำลังสองเป็น 0.0338 สำหรับชุดฝึก และ 0.0369 สำหรับชุดทดสอบ. ผลการทำนายของแบบจำลอง แสดงในภาพขวา.

โครงข่ายประสาทเทียมทำงานได้ดี. ประเด็นหนึ่งที่สำคัญ และค่อนข้างจะทั่วไป ไม่ได้เจาะจงกับภาระ กิจไดเฉพาะ คือ การทำนอร์มอลайซ์ (normalization) สำหรับอินพุต. การทำนอร์มอลายซ์สำหรับ อินพุต คือ การปรับขนาดอินพุต เพื่อช่วยให้การฝึก และการทำงานของโครงข่ายประสาทเทียมทำได้ ง่ายขึ้น.

ก่อนอภิปรายวิธีการทำนอร์มอลายซ์อินพุต พิจารณาผลการฝึก และทดสอบโครงข่ายประสาทเทียม ของข้อมูลสองชุด ซึ่งลักษณะข้อมูล ความก้าวหน้าในการฝึก และผลการทำนายกับข้อมูลทดสอบ แสดง ในรูปที่ 3.19. ผลการทดสอบ (ระบุในคำบรรยายภาพ) และภาพในรูปที่ 3.19 แสดงให้เห็นชัดเจน ว่า แบบจำลองทำงานได้ดีกับข้อมูลชุดเอ (dataset A) แต่ทำงานได้แย่มากกับข้อมูลชุดบี (dataset B). ข้อมูลทั้งสองชุด มีลักษณะความสัมพันธ์ระหว่างตัวแปรต้น x และตัวแปรตาม y ในแบบเดียวกัน. ความต่างระหว่างข้อมูลชุดเอและชุดบี มีอย่างเดียวคือ ขนาดของอินพุตของข้อมูล. ขนาดของอินพุต ของข้อมูลชุดเอ อยู่ในช่วง 0 ถึง 1 ในขณะที่ขนาดของอินพุตของข้อมูลชุดบี อยู่ในช่วง 100 ถึง 110.

พิจารณาลักษณะของการแพร่กระจายข้อมูล โดยเฉพาะสมการ 3.31 จะเห็นว่า $h'(a_j^{(q)})$ มีผล โดยตรงกับเกรเดียนต์. ค่า $h'(a_j^{(q)})$ คืออนุพันธ์ของฟังก์ชันกระตุ้น ซึ่งซิกมอยด์เป็นฟังก์ชันกระตุ้น ที่ใช้. อนุพันธ์ของซิกมอยด์ (แสดงในรูปที่ 3.20) จะมีค่าน้อยมาก ๆ เมื่อตัวกระตุ้นมีค่าใหญ่ ๆ (เช่น มากกว่าห้า หรือน้อยกว่าห้า) ซึ่งจะขับค่าของฟังก์ชันซิกมอยด์ไปที่ปลาย² (ค่าใกล้ ๆ หนึ่ง หรือค่า

²บางครั้งช่วงค่าซิกมอยด์ในyanที่มีค่าใกล้ ๆ หนึ่ง หรือค่าใกล้ ๆ ศูนย์ มักถูกเรียกว่า ช่วงอิมเมต้า (saturation region) เนื่องจาก

ใกล้ ๆ ศูนย์). ค่าอินพุตที่มีขนาดใหญ่ จะทำให้ตัวกระตุ้นมีค่าใหญ่ และจะส่งผลให้อัตราพัฒนาของชิกโมย์ดี มีค่าน้อย ซึ่งส่งผลต่อให้เกรเดียนต์มีค่าน้อย และสุดท้าย ส่งผลให้การฝึกแบบจำลองทำได้ช้า.

การทำอร์มอลайเซอร์อินพุต จึงเป็นการปรับขนาดของอินพุตให้อยู่ในช่วงที่การฝึกแบบจำลองทำได้ง่าย จึงสำคัญอย่างมากในทางปฏิบัติ. การเรียนรู้สำหรับสิงมีชีวิตก็เช่นเดียวกัน หากสภาพแวดล้อม เหมาะกับการเรียนรู้ สิงมีชีวิตก็จะเรียนรู้ได้เร็วขึ้น เรียนรู้ได้ดีขึ้น.

การทำอร์มอลายเซอร์สามารถทำได้หลายวิธี สองวิธีที่วิธีการปรับสูช่วงที่กำหนด และวิธีการปรับสูค่าสถิติที่กำหนด. วิธีการปรับสูช่วงที่กำหนด จะปรับขนาดของอินพุต ให้ค่าอยู่ใน ช่วงที่กำหนด เช่น $[-1, 1]$ หรือ $[0, 1]$. หากช่วงที่ต้องการคือ $[x'_{\min}, x'_{\max}]$ ค่าอินพุตที่ผ่านการทำอร์มอลายเซอร์ หรือเรียกว่า นอร์มอลายเซอร์อินพุต (*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} คือค่าน้อยที่สุดกับค่ามากที่สุดของอินพุตเดิม.

วิธีการปรับสูค่าสถิติที่กำหนด นิยมปรับค่าอินพุต เพื่อให้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน เป็น 0 กับ 1 ตามลำดับ. ดังนั้น นอร์มอลายเซอร์อินพุต x' สามารถคำนวณได้จาก

$$x' = \frac{x - \bar{x}}{\sigma_x} \quad (3.42)$$

เมื่อ \bar{x} และ σ_x คือค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน ของค่าอินพุตเดิม.

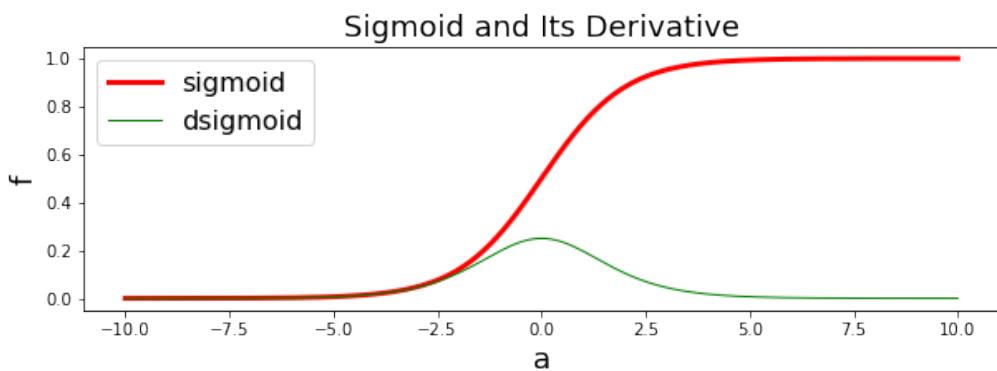
หมายเหตุ สำหรับอินพุตขนาดหลายมิติ $\mathbf{x} \in \mathbb{R}^D$ เมื่อ D เป็นจำนวนมิติของปริภูมิอินพุต. การ ทำการทำอร์มอลายเซอร์ ยังช่วยรักษาสมดุลของขนาดของอินพุตในมิติต่าง ๆ กันด้วย. การทำการทำอร์มอลายเซอร์ จะทำ แต่ละมิติ เช่น กรณีสองมิติ $\mathbf{x} = [x_1, x_2]^T$ และต้องการทำอร์มอลายเซอร์ ให้อยู่ในช่วง $[0, 1]$ ทั้งคู่ จะ ดำเนินการโดย

$$\mathbf{x}' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} (x_1 - x_{1\min}) / (x_{1\max} - x_{1\min}) \\ (x_2 - x_{2\min}) / (x_{2\max} - x_{2\min}) \end{bmatrix}$$

เมื่อ x'_1 และ x'_2 คือค่าอร์มอลายเซอร์อินพุต ของมิติที่หนึ่งและมิติที่สองตามลำดับ และ $x_{1\min}$ กับ $x_{1\max}$ คือค่าน้อยที่สุดกับค่ามากที่สุดของอินพุตเดิมในมิติที่หนึ่ง และ $x_{2\min}$ กับ $x_{2\max}$ คือค่าน้อยที่สุดกับค่ามากที่สุดของอินพุตเดิมในมิติที่สอง.

การหยุดก่อนกำหนด. ประเด็นของการโอลีฟิต และคุณสมบัติความทวีไป เป็นประเด็นสำคัญสำหรับการใช้งานแบบจำลองทำนาย ที่รวมถึงโครงข่ายประสาทเทียม. การเลือกใช้แบบจำลองที่มีความ

ในย่านเหล่านั้น ค่าฟังก์ชันชิกโมย์มีการเปลี่ยนแปลงน้อยมาก เมื่อเทียบกับค่าตัวกระตุ้น (อัตราพัฒนาเร็ว).

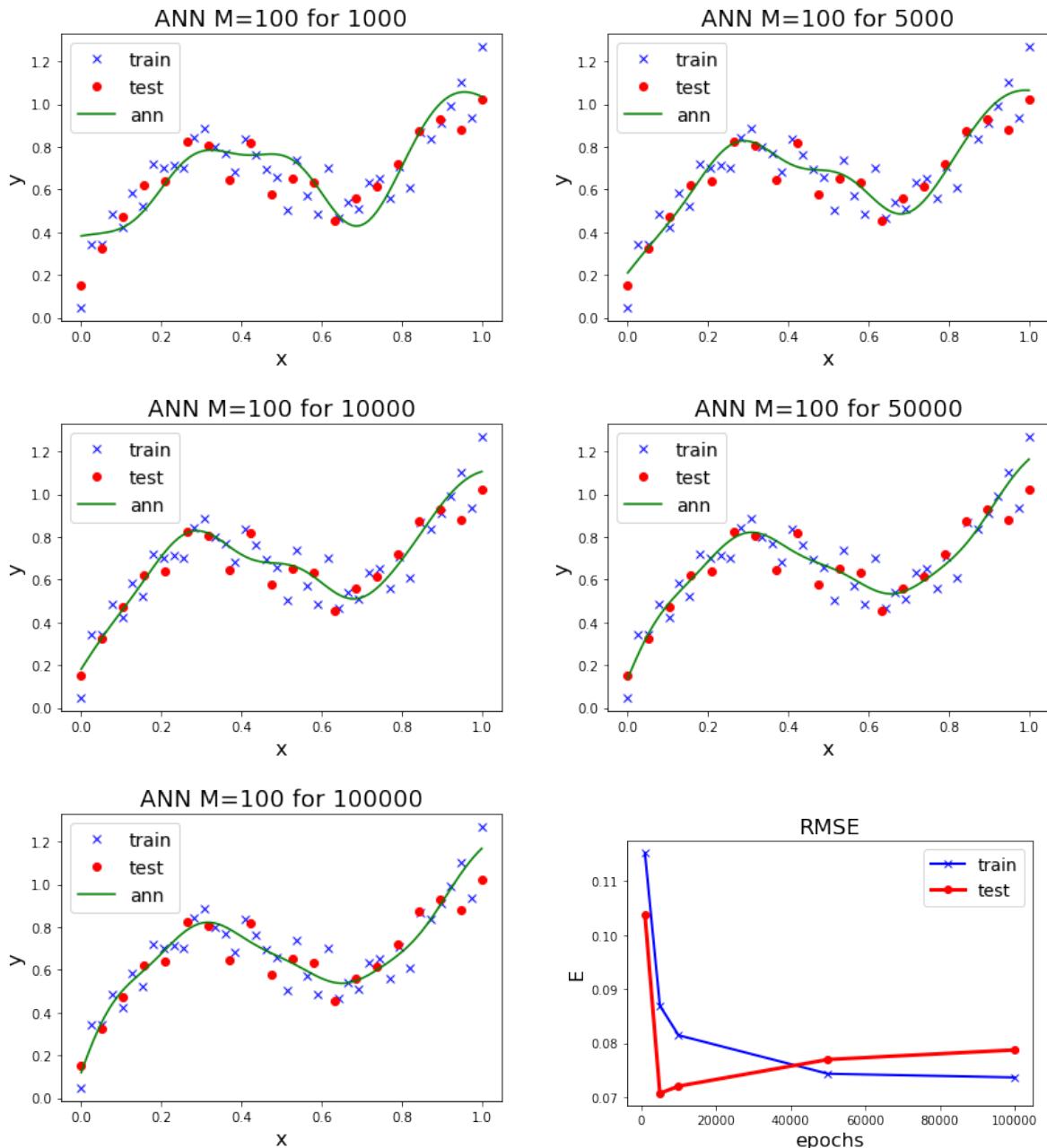


รูปที่ 3.20 พังก์ชันซิกมอยด์ (เส้นสีแดง) และอนุพันธ์ (เส้นสีเขียว).

ยึดหยุ่นสูง ๆ เช่น การใช้โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยซ่อนมาก ๆ หรือการใช้โครงข่ายประสาทเทียมที่มีชั้นลึก ๆ ก็เสียงที่จะเกิดการโอเวอร์ฟิตได้.

รูปที่ 3.21 แสดงผลการฝึกโครงข่ายสองชั้นขนาด 100 หน่วยซ่อน กับข้อมูลฝึก 40 จุดข้อมูล. ระหว่าง 5000 ถึง 10000 สมัย ให้ผลการทำนายที่ดี และการฝึกต่อเพิ่มไป นอกจากเสียเวลาเพิ่ม แล้วยังทำให้แบบจำลองโอเวอร์ฟิตอีกด้วย ดังเห็นได้จากค่าผิดพลาดกับข้อมูลทดสอบที่เพิ่มขึ้น (ภาพล่างสุดซ้าย) และเวลาที่ใช้ยังเป็นราوا 9 ถึง 18 เท่าอีกด้วย เวลาที่ใช้ฝึกจะบุในคำบรรยายภาพ. หมายเหตุเวลาที่ระบุ แสดงเป็นเวลาอ่อนร์มอลайซ์ด (normalized time). นั่นคือ เวลาที่แสดงเป็นอัตราส่วนโดยใช้เวลาที่อ้างอิงเป็นตัวหาร. เวลาที่รายงานในรูปที่ 3.21 ใช้เวลาที่ฝึกแบบจำลอง 5000 สมัยเป็นเวลาอ้างอิง. การรายงานเวลา ด้วยเวลาอ่อนร์มอลายซ์ด แทนเวลาสมบูรณ์ ช่วยบอกความรวมของเวลาการทำงาน โดยลดความจำเป็นในการรายงานรายละเอียด ของฮาร์ดแวร์และระบบที่ใช้ทดสอบบล.

การฝึกโครงข่ายประสาทเทียมใช้เวลานาน การฝึกโดยการใช้สมัยฝึกมาก จะเกิดโอเวอร์ฟิต นอกจากจะได้แบบจำลองที่ไม่มีคุณสมบัติความทั่วไปแล้ว ยังเสียเวลาฝึกด้วย. การปฏิบัติที่นิยม ก็คือ การหยุดก่อนกำหนด (early stopping). การหยุดก่อนกำหนด คือ การใช้เงื่อนไขเพื่อหยุดการฝึก โดยเงื่อนไข คือเมื่อค่าผิดพลาดของข้อมูลทดสอบสูงขึ้นกว่าเดิม ให้หยุดการฝึก. ค่าผิดพลาดของข้อมูลทดสอบสูงขึ้นกว่าเดิม เป็นสัญญาณของการโอเวอร์ฟิต และเพื่อให้ไม่มีการใช้ข้อมูลทดสอบสุดท้าย ในกระบวนการฝึกจริง ๆ ข้อมูลทดสอบ เพื่อใช้สำหรับการหยุดก่อนกำหนดจะแบ่งออกมากจากข้อมูลชุดฝึก เรียกว่า ข้อมูลตรวจสอบ (validation data). ข้อมูลตรวจสอบ เป็นข้อมูลอีกชุดที่แยกออก มา. ข้อมูลตรวจสอบ ไม่ใช่ในการฝึก (ไม่ใช่ในการคำนวณเพื่อปรับค่าน้ำหนัก) และไม่ใช่ในการทดสอบสุดท้าย (ไม่ใช่ในการทดสอบ เพื่อรายงานผลการทำงานแบบจำลอง). ข้อมูลตรวจสอบ เป็นข้อมูลที่ใช้ในกระบวนการฝึก แต่ไม่ได้ใช้ในการฝึก ข้อมูลตรวจสอบ จะใช้เพื่อเลือกแบบจำลอง หรือใช้เพื่อเลือกความซับซ้อนของแบบจำลอง (เช่น จำนวนหน่วยซ่อน) หรือใช้เพื่อการหยุดก่อนกำหนด เป็นต้น.



รูปที่ 3.21 การฝึกที่จำนวนสมัยต่าง ๆ ผลการฝึกโครงข่ายสองชั้นขนาดหน่วยช่อง 100 หน่วย ที่จำนวนสมัยฝึก 1000, 5000, 10000, 50000 และ 100000 รอบ. ห้าภาพแรก แสดงจุดข้อมูลฝึก (ภาคบาทสีฟ้า) จุดข้อมูลทดสอบ (วงกลมสีแดง) ค่าทำนายจากแบบจำลอง (เส้นสีเขียว) ซึ่งภาพระบุจำนวนหน่วยช่อง และจำนวนสมัยที่ได้ทำการฝึกไป. ภาพสุดท้าย (ล่างสุดช้าย) 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.

3.5 คำแนะนำสำหรับการใช้แบบจำลองทำนาย

การประเมินผล เป็นกลไกที่สำคัญมากสำหรับการใช้แบบจำลองทำนาย. แต่หากผลที่ประเมินได้ไม่น่าพอใจ และต้องการปรับปรุงให้ได้ผลการทำนายดีขึ้น มีทางเลือกต่าง ๆ มากมาย เช่น การเพิ่มจำนวนข้อมูลที่จะใช้สำหรับการฝึก หรือการคัดเลือกตัวแปรต้นของข้อมูล โดยเลือกเฉพาะคุณลักษณะที่สำคัญนั่นคือเลือกเฉพาะมิติ บางมิติที่สำคัญ มาใช้เป็นอินพุตสำหรับแบบจำลอง หรือการเพิ่มคุณลักษณะใหม่เข้าไปในอินพุตของแบบจำลอง หรือการเพิ่มความซับซ้อนของแบบจำลองขึ้น หรือการลดความซับซ้อนของแบบจำลองลง.

ทางเลือกที่หลากหลาย อาจทำให้สับสนได้ และการปรับปรุงแบบจำลอง ควรจะลองทำอะไรก่อน อะไรหลัง ซึ่งการลองทำแต่ละอย่าง อาจใช้เวลา多く และยังอาจเพิ่มงบประมาณด้วย เช่น การออดไปหาข้อมูลมาเพิ่ม (เพิ่มจำนวนจุดข้อมูล) หรือการเพิ่มลักษณะสำคัญนิดใหม่เข้าไป (เพิ่มมิติใหม่สำหรับอินพุต). ผู้เชี่ยวชาญศาสตร์การเรียนรู้ของเครื่อง แอนดรูว์ อิง[139] แนะนำว่า ก่อนตัดสินใจเลือกทดลองปรับปรุงด้วยวิธีใด ควรทำการทดลองง่าย ๆ และใช้เส้นโค้งเรียนรู้ (learning curve) เป็นตัวชี้แนะ. การใช้เส้นโค้งเรียนรู้ มีพื้นฐานมาจาก การศึกษาเรื่องคุณภาพการทำนาย ที่สัมพันธ์กับความลำเอียง (*bias*³) และความแปรปรวน (*variance*).

ความลำเอียงกับความแปรปรวน

พิจารณารูปที่ 3.5 ที่ใช้ฟังก์ชันพหุนามทำนายข้อมูล. ภาพ x ใน รูปที่ 3.7 แสดงค่าผิดพลาดที่ได้ต่อระดับขึ้น. ระดับขึ้น บอกความซับซ้อนของแบบจำลองฟังก์ชันพหุนาม. ที่ความซับซ้อนของแบบจำลองที่เหมาะสม ค่าผิดพลาดของชุดข้อมูลทดสอบ จะมีค่าต่ำที่สุด.

ฟังก์ชันพหุนามระดับขึ้นเก้าก้าว เอเวอร์ฟิตข้อมูลอย่างชัดเจน. ส่วนฟังก์ชันพหุนามระดับขั้นศูนย์ ระดับขั้นหนึ่ง และระดับขั้นสองอันเดอร์ฟิตของข้อมูลอย่างชัดเจน. การอันเดอร์ฟิต (underfit) หมายถึง การที่ความซับซ้อนของแบบจำลอง ไม่เพียงพอที่จะประมาณความสัมพันธ์ของข้อมูลได้. ช่วงที่ความซับซ้อนของแบบจำลองน้อยเกินไป มีจุดสังเกตสำคัญคือ ค่าผิดพลาดจะสูงกับทั้งข้อมูลชุดทดสอบ และชุดฝึกหัด. เมื่อความซับซ้อนของแบบจำลองไม่พอ แบบจำลองจะไม่ยืดหยุ่นพอที่ปรับตัว เพื่อลดค่าผิดพลาดกับข้อมูลชุดฝึกหัด. กรณีที่แบบจำลองทำงานได้ไม่ดี เนื่องจากความซับซ้อนของแบบจำลองน้อยเกินไปนี้ จะเรียกว่า กรณีที่มีความลำเอียงสูง (*high bias*) ซึ่งสืบถึงการอันเดอร์ฟิต.

ส่วนกรณีที่ความซับซ้อนของแบบจำลองมากเกินไป ซึ่งคือกรณีการเอเวอร์ฟิตมีจุดสังเกตที่สำคัญคือ ค่าผิดพลาดกับชุดฝึกจะต่ำมาก แต่ค่าผิดพลาดกับชุดทดสอบจะสูง. แบบจำลองที่มีความซับซ้อนมาก จะสามารถปรับพฤติกรรมการทำนาย เพื่อลดค่าผิดพลาดกับข้อมูลชุดฝึกหัดลงได้ดี แต่หากความซับซ้อนมากเกินไป จะนำไปลดค่าผิดพลาดกับข้อมูลชุดฝึกหัดที่เกิดจากสัญญาณรบกวนด้วย แบบจำลอง

³ในบริบทของพหุติกรรมโดยรวมของแบบจำลอง ไม่ใช่บริบทของพารามิเตอร์ของโครงข่ายประสาทเทียม.

จะทำนายข้อมูลชุดทดสอบได้ไม่ดี. กรณีนี้จะเรียกว่า เป็นกรณีที่มีความแปรปรวนสูง (*high variance*) ซึ่งสื่อถึงการโอเวอร์ฟิต.

กล่าวอีกอย่างหนึ่ง ความลำเอียงสูง หมายถึงแบบจำลองของไม่ยึดหยุ่นมากพอ ส่วนความแปรปรวนสูงหมายถึงแบบจำลองของยึดหยุ่นมากเกินไป. แบบจำลองที่ดีที่สุด ที่เราจะทำได้ ก็คือ แบบจำลองที่ลดทั้งความลำเอียงและความแปรปรวนให้มีค่าต่ำ. แต่ไม่ว่าแบบจำลองไหนก็ตาม เราทำดีที่สุดได้แค่ดีที่สุด จะฝืนข้อจำกัดของธรรมชาติไม่ได้.

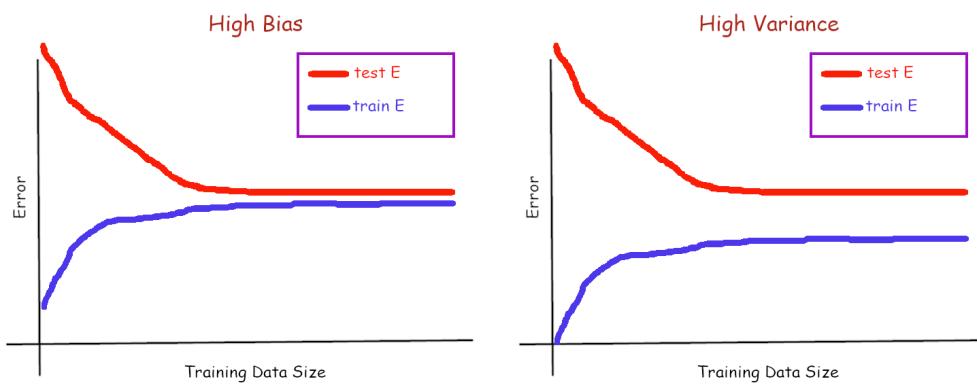
ทฤษฎีที่กล่าวถึงข้อจำกัดของการทำแบบจำลองนี้คือ ทวิบตของความลำเอียงกับความแปรปรวน (*bias/variance dilemma*). ทวิบตของความลำเอียงกับความแปรปรวน เสนอโดย เจมัน และคณะ[70] จากการศึกษาความสามารถ และขีดจำกัดของการทำแบบจำลอง. ทวิบตของความลำเอียงกับความแปรปรวน สรุปว่า เราสามารถทำแบบจำลองให้ดีที่สุดได้ โดยลดทั้งค่าความลำเอียงและความแปรปรวนให้ต่ำ. แต่ค่าผิดพลาดก็มีขีดจำกัดหนึ่ง (ขึ้นกับธรรมชาติของปัญหา) ที่เราไม่สามารถลดค่าผิดพลาดลงไปให้ต่ำกว่านั้นได้. การที่เราพยายามจะลดค่าผิดพลาดให้ต่ำกว่านั้น โดยการลดส่วนหนึ่ง ก็จะไปเพิ่มอีกส่วน เช่น หากพยายามลดค่าผิดพลาดจากความลำเอียงมากเกินไป ก็จะทำให้ส่วนที่เกิดจากความแปรปรวนเพิ่ม และในทางกลับกัน หากพยายามลดค่าผิดพลาดจากความแปรปรวนมากเกินไป ก็จะทำให้ส่วนที่เกิดจากความลำเอียงเพิ่ม.

เส้นโค้งเรียนรู้

เส้นโค้งเรียนรู้ (*learning curve*) เป็นเครื่องมือช่วยแสดงความสัมพันธ์ระหว่างความพยายามที่ใช้ไปในการฝึกแบบจำลอง กับผลการทำงานของแบบจำลอง. เส้นโค้งเรียนรู้ สามารถใช้เพื่อช่วยให้เข้าใจว่า แบบจำลองที่ใช้อยู่ มีความเสี่ยงที่จะมีความลำเอียงสูง หรือเสี่ยงที่จะมีความแปรปรวนสูง. เส้นโค้งเรียนรู้ สร้างโดย การตรวจสอบผลการฝึกแบบจำลองที่จำนวนจุดข้อมูลฝึกต่าง ๆ แล้วนำค่าผิดพลาดเฉลี่ยกับชุดฝึก และค่าผิดพลาดเฉลี่ยกับชุดทดสอบ มาวาดกราฟ ดังแสดงในรูปที่ 3.22.

การวัดค่าผิดพลาดกับชุดฝึก วัดเฉพาะกับจุดข้อมูลที่ใช้ฝึก เช่น หากฝึกแบบจำลองด้วย 10 จุดข้อมูล ก็หาค่าผิดพลาดของการทำนายค่า 10 จุดข้อมูลนี้. ดังนั้น เมื่อจำนวนจุดข้อมูลฝึกน้อย แบบจำลองมีเจตนาที่ต้องทำนาย ก็มีโอกาสที่จะทำค่าผิดพลาดของชุดฝึกน้อยด้วย. เมื่อเพิ่มจำนวนจุดข้อมูลฝึกขึ้น ค่าผิดพลาดของชุดฝึกก็จะเพิ่มขึ้น และลู่เข้าสู่ค่า ๆ หนึ่ง. ในขณะเดียวกัน เมื่อจำนวนจุดข้อมูลฝึกเพิ่มขึ้น ค่าผิดพลาดของชุดทดสอบจะลดลงจนลู่เข้าสู่ค่า ๆ หนึ่ง.

ในกรณีที่ แบบจำลองมีความเสี่ยงจากความลำเอียงสูง ถ้าข้อมูลทดสอบ และข้อมูลฝึกมีปริมาณเพียงพอ พฤติกรรมการทำนายของแบบจำลอง จะให้ผลในลักษณะเดียวกัน และทำให้ค่าผิดพลาดจากชุดฝึก และจากชุดทดสอบ มีค่าใกล้เคียงกัน. แต่กรณีที่แบบจำลองมีความแปรปรวนสูง นั่นคือ แบบจำลองมีความยึดหยุ่นมากเกินไป หากจนพอที่จะปรับตัวเข้ากับสัญญาณrgbในข้อมูลฝึกได้. ผลคือแบบจำลองสามารถลดค่าผิดพลาดของชุดฝึกได้ต่ำ และทำให้ผลต่างระหว่างค่าผิดพลาดจากชุดฝึก



รูปที่ 3.22 ภาพวาดเส้นโค้งเรียนรู้. ภาพซ้าย แสดงกรณีแบบจำลองมีความล้าเอียงสูง. เมื่อจำนวนจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดกับชุดฝึก และค่าผิดพลาดกับชุดทดสอบ มีค่าใกล้เคียงกัน. ภาพขวา แสดงกรณีแบบจำลองมีความแปรปรวนสูง. เมื่อจำนวนของจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดกับชุดฝึก และค่าผิดพลาดกับชุดทดสอบ มีค่าต่างกัน.

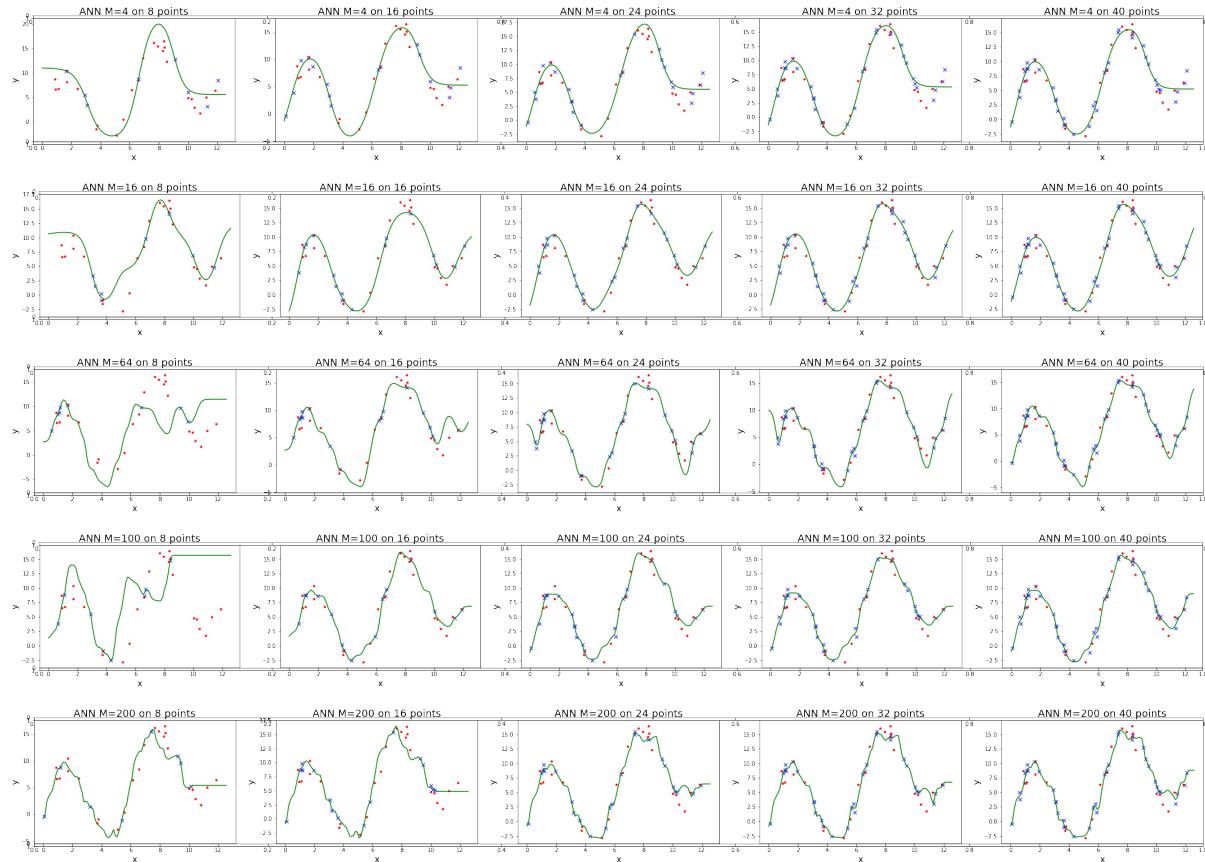
และจากชุดทดสอบมีค่าต่างกัน. ดังนั้นความต่างระหว่างค่าผิดพลาดจากข้อมูลทดสอบและจากข้อมูลฝึกจึงสามารถใช้บ่งชี้สถานการณ์ค่าต่อไปได้.

รูปที่ 3.22 เป็นภาพวาดเพื่อให้เห็นภาพรวม แต่ ในสถานการณ์จริง เส้นโค้งเรียนรู้ที่ได้ อาจจะมีสัญญาณรบกวนมาก หรือแม้แต่สเกลของการวัดกราฟ อาจทำให้ต้องใช้ความระมัดระวัง ในการอ่านผลจากเส้นโค้งเรียนรู้. รูปที่ 3.23 แสดงกระบวนการทดสอบแบบจำลอง เพื่อวัดเส้นโค้งเรียนรู้. รูปที่ 3.24 แสดงเส้นโค้งเรียนรู้ที่ได้จาก แบบจำลองที่มีความซับซ้อนต่าง ๆ กัน เพื่อให้เห็นตัวอย่างลักษณะของเส้นโค้งเรียนรู้ในกรณีต่าง ๆ.

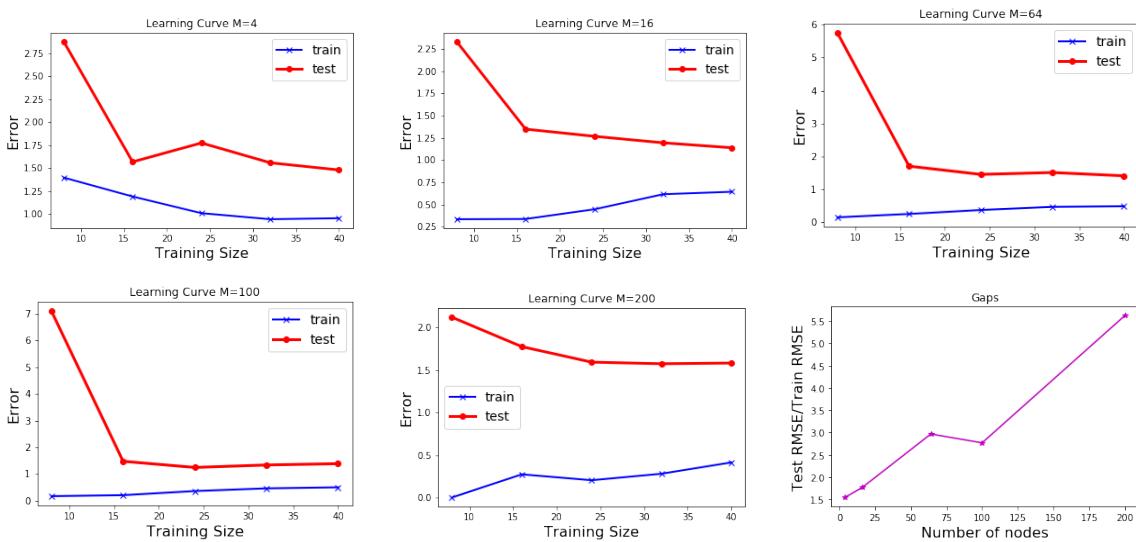
จากรูปที่ 3.23 และคำบรรยาย แบบจำลองโครงข่ายประสาทเทียมสองชั้น ขนาด 4 หน่วยช่อง อันเดอร์พิตข้อมูล และเส้นโค้งเรียนรู้ ก็ถูกเข้าหากัน โดยอัตราความต่างระหว่างค่าผิดพลาดจากชุดทดสอบและชุดฝึกค่อนข้างต่ำ (ประมาณ 1.55). แบบจำลองโครงข่ายประสาทเทียมสองชั้นขนาด 200 หน่วยช่อง โอเวอร์พิตข้อมูลอย่างชัดเจน (รูปที่ 3.23) และเส้นโค้งเรียนรู้ก็ถูกเข้าห่างกันชัดเจน โดยอัตราความต่างระหว่างค่าผิดพลาดจากชุดทดสอบและชุดฝึกสูง (ประมาณ 5.62 รูปที่ 3.24).

หลังจากพูดแล้วว่า สถานการณ์อยู่ในกรณีความล้าเอียง หรือความแปรปรวนสูง แอนดรูว์ อิง[139] แนะนำให้พิจารณาทางเลือก ดังนี้

- เก็บข้อมูลมาเพิ่มสำหรับการฝึก. การเพิ่มจำนวนข้อมูลในการฝึก จะช่วยในกรณีความแปรปรวนสูง.
- เลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต. การเลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต. การลดมิติของอินพุตลง ก็จะช่วยในกรณีความแปรปรวนสูง โดยเฉพาะ สำหรับโครงข่ายประสาทเทียม. ตัวอย่าง เช่น โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยช่องเท่าเดิม การลดมิติของอินพุตลง เท่ากับลดจำนวนพารามิเตอร์ลง.



รูปที่ 3.23 ตัวอย่างพัฒนาระบบจำลองที่ความซับซ้อนต่าง ๆ ตัวอย่างกระบวนการทดสอบแบบจำลอง เพื่อวัดเลี้น์โค้งเรียนรู้ สำหรับโครงข่ายประสาทเทียมสองชั้น ขนาด 4 ขนาด 16 ขนาด 64 ขนาด 100 และขนาด 200 หน่วยซ่อน จากแควนลลงล่าง ตามลำดับ. แต่ละaccoแสดง 5 ภาพ ที่แต่ละภาพเป็นการฝึกกับข้อมูลฝึกขนาด 8 จุด ขนาด 16 จุด ขนาด 24 จุด ขนาด 32 จุด และขนาด 40 จุด จากข้อมูลฝึกขนาด 8 ตามลำดับ. แต่ละภาพใช้สถาปัตยกรรมฟีดฟอร์เวดเดิร์ฟเพื่อแสดงจุดข้อมูลฝึก และใช้วงกลมสีแดงแสดงจุดข้อมูลทดสอบ เส้นสีเขียวแสดงค่าที่แบบจำลองที่ฝึกเสร็จแล้วทำนาย. ซึ่งแต่ละภาพระบุความซับซ้อนของแบบจำลองด้วยจำนวนหน่วยซ่อน และระบุจำนวนจุดข้อมูลที่ใช้ฝึก. ตัวอย่างนี้ เพื่อแสดงในเห็นกระบวนการดำเนินการอย่างชัดเจน และเพื่อเปรียบเทียบผลของเส้นโค้งเรียนรู้. ในสถานการณ์จริง การใช้เส้นโค้งเรียนรู้ ก็เพื่อหลีกเลี่ยง การที่จะต้องทดลองโดยตรงกับหลาย ๆ แบบจำลองเช่นนี้. ค่ารากสองของค่าเฉลี่ยค่าผิดพลาดกำลังสองของโครงข่ายสองชั้นกับข้อมูลทดสอบ 40 จุด คือ 1.48, 1.14, 1.40, 1.39, และ 1.58 เมื่อใช้หน่วยซ่อน 4, 16, 64, 100, และ 200 หน่วยตามลำดับ.



รูปที่ 3.24 ตัวอย่างเส้นโค้งเรียนรู้ จากแบบจำลองขนาด 4, 16, 64, 100, และ 200 หน่วยช่อง ตามที่ระบุในชื่อแต่ละภาพ. ภาพล่างขวา แสดงความต่างระหว่างค่าผิดพลาดจากข้อมูลฝึกกับค่าผิดพลาดจากข้อมูลทดสอบ (แกนตั้ง) และความซับซ้อนของแบบจำลอง (แกนนอน). ความต่างแสดงในรูปแบบอัตราส่วน นั่นคือ $E_{\text{test}}/E_{\text{train}}$ เมื่อ E_{test} คือรากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากข้อมูลทดสอบ และ E_{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.

- เพิ่มลักษณะที่สำคัญใหม่เข้าไปในอินพุต. การเพิ่มลักษณะที่สำคัญใหม่เข้าไปในอินพุต การเพิ่มนิตริของอินพุตขึ้น โดยทั่วไปแล้ว จะช่วยในกรณีความจำเอียงสูง.
- เพิ่มความซับซ้อนของแบบจำลองขึ้น. การเพิ่มความซับซ้อนของแบบจำลอง เช่น การเพิ่มจำนวนหน่วยช่อง จะช่วยในกรณีความจำเอียงสูง.
- ลดความซับซ้อนของแบบจำลองลง. การลดความซับซ้อนของแบบจำลอง เช่น การลดจำนวนหน่วยช่อง การทำเรกุลาราizer หรือ การทำการหยุดก่อนกำหนด จะช่วยในกรณีความแปรปรวนสูง.

ตารางที่ 3.4 สรุปทางเลือกที่แอนดรูว์ อิง[139] แนะนำ สำหรับกรณีความจำเอียงสูง และกรณีความแปรปรวนสูง. สุดท้าย หากลองวิธีทั่ว ๆ ไปดังนี้แล้ว ผลยังไม่น่าพอใจ อาจจะลองวิเคราะห์ และตรวจสอบผลที่ขึ้นตอนวิธีทำผิด ดูว่าผิดลักษณะไหน อย่างไร. เพื่อว่า อาจจะพบคุณสมบัติเฉพาะบางอย่างที่ผลมักจะผิด เช่น หากเป็นการจำแนกตัวเลขจากภาพ ขึ้นตอนวิธีที่ใช้ อาจจะจำแนกเลข 2 เป็นเลข 4 บ่อย ๆ ซึ่งเมื่อดูภาพของเลขที่จำแนกผิด ก็อาจพบว่า มีรูปแบบการเขียนเลข 2 แบบหนึ่ง ที่มักจะจำแนกผิด. หากพบรูปแบบเฉพาะนั้น อาจเพิ่มการจัดการพิเศษเฉพาะสำหรับรูปแบบนั้นได้.

ตารางที่ 3.4 สรุปคำแนะนำสำหรับปรับปรุงแบบจำลอง ในการนีความลำเอียงสูงและความแปรปรวนสูง.

ทางเลือก	กรณี	
	ความลำเอียงสูง	ความแปรปรวนสูง
เพิ่มรอบฝึก	แนะนำ	
เพิ่มจำนวนหน่วยอย่างไร	แนะนำ	
ทำเรกูล่าไรซ์		แนะนำ
ทำหยุดก่อนกำหนด		แนะนำ
ลดมิติของอินพุตลง		แนะนำ
เพิ่มมิติของอินพุตขึ้น	แนะนำ	
เพิ่มจำนวนจุดข้อมูลฝึก		แนะนำ

3.6 อภิรานศัพท์

การปรับเส้นโค้ง (curve fitting): การปรับพหุติกร姆ของแบบจำลองทำนาย โดยการเปลี่ยนค่าของ พารามิเตอร์ เพื่อให้พหุติกร姆ทำนายสอดคล้องกับข้อมูลที่มี.

จุดข้อมูล (datapoint): ตัวอย่างข้อมูลที่มีค่าของตัวแปรต้น x และตัวแปรตาม y ที่เป็นคู่กัน.

พหุกัณฑ์พหุนาม (polynomial function): พหุกัณฑ์ที่แสดงความสัมพันธ์ระหว่างตัวแปรต้น x และ ตัวแปรตาม y ด้วยสมการพหุนาม $y = f(x, w) = \sum_{m=0}^M w_m x^m$ เมื่อ พารามิเตอร์ของ พหุกัณฑ์ $w = [w_0, w_1, \dots, w_M]^T$.

การฝึก (training): การฝึก หรือการเรียนรู้ คือการปรับค่าพารามิเตอร์ของแบบจำลอง เพื่อให้แบบ จำลองมีพหุติกร姆การทำนายสอดคล้องกับข้อมูล.

ค่าเฉลย (ground truth): ค่าของตัวแปรตาม y จากข้อมูล ที่คู่กับตัวแปรต้น x ที่สนใจ.

คุณสมบัติความทั่วไป (generalization): ความสามารถของแบบจำลองทำนาย ที่สามารถทำนาย ข้อมูลที่ไม่เคยเห็นได้ดี.

ข้อมูลฝึก (training data): ข้อมูลที่ใช้ในกระบวนการการฝึกแบบจำลอง

ข้อมูลทดสอบ (test data): ข้อมูลที่ใช้ในทดสอบแบบจำลอง.

การโอเวอร์ฟิต (overfitting): การที่แบบจำลองสามารถทำนายข้อมูลฝึกได้ดี แต่ทำนายข้อมูลใหม่ที่ ไม่เคยเห็นได้ไม่ดี นั่นคือ การที่แบบจำลองไม่มีคุณสมบัติความทั่วไป.

ความซับซ้อนของแบบจำลอง (model complexity): การยึดหยุ่นของแบบจำลอง อาจบ่งชี้ได้จาก จำนวนพารามิเตอร์ของแบบจำลอง.

โครงข่ายประสาทเทียม (artificial neural network): แบบจำลองคำนวณ ที่ทำการคำนวณ โดยใช้ หน่วยคำนวณอย่อยหลากรูปแบบ ที่แต่ละหน่วยทำการคำนวณในแบบคล้าย ๆ กัน.

เพอร์เซปตรอนหลายชั้น (multi-layer perceptron): โครงข่ายประสาทเทียม ที่หน่วยคำนวณต่าง

ๆ ต่อ กัน เป็น โครง ข่าย ใน ลักษณะ ชั้น คำนวณ.

โนนด หรือ หน่วย คำนวณ (node หรือ unit): การคำนวณย่อยของโครงข่ายประสาทเทียม ซึ่งมักเป็น การคำนวณง่าย ๆ ไม่ซับซ้อน.

ชั้น คำนวณ (layer): กลุ่มของโนนด ในโครงข่ายประสาทเทียม ที่จัดโครงสร้างเป็นลักษณะชั้น คำนวณ โดย กลุ่มของโนนด ใน ชั้น คำนวณเดียวกัน จะ รับ อินพุต จาก (กลุ่มของโนนด ใน) ชั้น คำนวณ ก่อน หน้า หรือ จะ ส่ง เอ้าต์พุต ออก ไป ให้ (กลุ่มของโนนด ใน) ชั้น คำนวณ ถัด ไป หรือ ทั้งสองอย่าง.

ชั้นซ่อน (hidden layer): ชั้น คำนวณ ที่ จะ ส่ง เอ้าต์พุต ออก ไป ให้ (กลุ่มของโนนด ใน) ชั้น คำนวณ ถัด ไป โดย เอ้าต์พุต ของ ชั้น ซ่อน จะ ไม่ ใช่ เอ้าต์พุต สุด ท้าย ของ โครงข่าย.

หน่วยซ่อน (hidden node หรือ hidden unit): โนนด ใน ชั้น ซ่อน.

ค่าน้ำหนัก และ ค่าไบอส (weights and biases): ค่า พารามิเตอร์ ต่าง ๆ ของ โครงข่าย ประสาท เทียม.

ฟังก์ชัน กระตุ้น (activation function): ฟังก์ชัน คำนวณ ของ โนนด.

ชั้น เอ้าต์พุต (output layer): ชั้น คำนวณ ชั้น สุด ท้าย ที่ เอ้าต์พุต ของ ชั้น จะ เป็น เอ้าต์พุต ของ โครงข่าย.

การ แพร่ กระจาย ย้อน กลับ (backpropagation หรือ error backpropagation): ขั้นตอน วิธี คำนวณ หา ค่า เกรเดียนต์ เพื่อ ปรับ ค่าน้ำหนัก สำหรับ โครงข่าย ประสาท เทียม.

การ ฝึก แบบ หมู่ (batch training): การ ฝึก ที่ ใช้ ข้อมูล ฝึก ทั้งหมด ที่ เดียว นั่น คือ การ หา ค่า เกรเดียนต์ และ ปรับ ค่า พารามิเตอร์ ทำ ได้ ยาว ใน แต่ ละ สมัย ฝึก.

การ ฝึก แบบ ออนไลน์ (online training): การ ฝึก ที่ ใช้ ข้อมูล ฝึก ที่ ละ หนึ่ง จุด ข้อมูล นั่น คือ การ ปรับ ค่า พารามิเตอร์ จะ ทำ หลาย ๆ ครั้ง แต่ ละ ครั้ง สำหรับ แต่ ละ จุด ข้อมูล และ จะ ทำ จน กว่า จะ ครบ ทุก จุด ข้อมูล ใน แต่ ละ สมัย ฝึก.

สมัย (epoch): รอบ การ ปรับ ค่า พารามิเตอร์ โดย แต่ ละ รอบ จะ นับ เมื่อ มี การ ใช้ ข้อมูล ฝึก ครบ ทุก จุด.

อัตราเรียนรู้ (learning rate): ขนาด ก้าว ที่ เป็น ค่า สเกลาร์ เพื่อ ควบคุม ความเร็ว ใน การ ฝึก แบบ จำ ลอง เป็น ค่า ที่ ใช้ กับ ขั้นตอน วิธี การ หา ค่า ดี ที่ สุด ที่ อยู่ บี หลัง การ ฝึก.

การ กำหนด ค่าน้ำหนัก เริ่ม ต้น (weight initialization): การ กำหนด ค่าน้ำหนัก และ ไบอส ให้ กับ โครงข่าย ประสาท เทียม ก่อน การ ฝึก.

การ จำ แนก ค่า ทวิภาค (binary classification): ภาระ กิจ การ ทำ นาย ที่ ผล การ ทำ นาย มี ได้ ส่อง แบบ.

ฟังก์ชัน สูญเสีย เอน โตรピー (cross entropy loss): ฟังก์ชัน จุด ประ สงค์ สำหรับ การ จำ แนก ค่า ทวิภาค หรือ การ จำ แนก กลุ่ม.

รหัส หนึ่ง ร้อน (one-hot coding หรือ one-of-K coding): รูปแบบ แทน ข้อมูล ที่ มี ลักษณะ เป็น กลุ่ม โดย รหัส จะ มี จำนวน ส่วน ประ กอบ ที่ เท่า กับ จำนวน กลุ่ม ทั้งหมด และ ตำแหน่ง ของ แต่ ละ ส่วน ประ กอบ แทน ฉลาก ของ กลุ่ม แต่ ละ กลุ่ม. รหัส จะ ระบุ ฉลาก ของ กลุ่ม โดย กำหนด ให้ ส่วน ประ กอบ ที่ อยู่ ตำแหน่ง ฉลาก นั้น มี ค่า เป็น หนึ่ง และ ส่วน ประ กอบ อื่น ๆ มี ค่า เป็น ศูนย์.

ซอฟต์แมกซ์ (softmax): ฟังก์ชัน คำนวณ เพื่อ ควบคุม ให้ เอ้าต์พุต ให้อยู่ ใน รูปแบบ ที่ สามารถ เปรียบ เทียบ ได้ กับ รหัส หนึ่ง ร้อน.

ກາຮານອ່ຽນໄລ້ອືນພຸດ (input normalization): ກາຮປະບົບນາດຂອງອືນພຸດທັງໝົດ.

ກາຮຢຸດກ່ອນກຳຫນດ (early stopping): ກາຮທຳເງື່ອນໄຂຈົບກາຮຟຶກ ໂດຍໃຊ້ຂໍ້ອູນລວມຕວຈສອບ.

ຂໍ້ອູນລວມຕວຈສອບ (validation data): ຜຸດຂໍ້ອູນລ ເພື່ອສະໜິມກະບວນກາຮເຕີຣີມແບບຈຳລອງ ອາຈໃຊ້ໜ່ວຍ
ກະບວນກາຮຟຶກ ແຕ່ໄໝໄດ້ໃຊ້ຟຶກແບບຈຳລອງໂດຍຕຽງ.

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 ไม่ใช่คนที่ไม่มีรู้สึกกลัว แต่เป็นคนที่อยู่เหนือความกลัวนั้น.”

—เนลสัน แมนเดลา

แบบฝึกหัด 3.1

จากตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง ในหัวข้อ 3.1 จะเขียนรูปสมการในลักษณะเดียวกับสมการ 3.9 สำหรับแบบจำลองพหุนามระดับขั้นได ๆ m . คำใบ้ ลองทำสำหรับระดับขั้นสองหรือระดับขั้นสามก่อน.

แบบฝึกหัด 3.2

จะแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันซิกมอยด์ 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

จะแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันไฮเปอร์บolicแทนเจนต์ $\tanh(a) = (e^a - e^{-a}) / (e^a + e^{-a})$ 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

จะแสดงให้เห็นว่าอนุพันธ์ของฟังก์ชันเรเดียลเบชิส (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) การหาค่าผลอย ใช้ฟังก์ชันเอกลักษณ์ ซึ่งคือ $\hat{y}_k = a_k^{(L)}$
และฟังก์ชันจุดประสงค์ค่าผิดพลาดกำลังสอง คือ $E = \frac{1}{2} \sum_k (\hat{y}_k - y_k)^2$.
- (ข) การจำแนกค่าทวิภาค ใช้ฟังก์ชันซิกโนยด ซึ่งคือ $\hat{y}_k = \frac{1}{1 + \exp(-a_k^{(L)})}$
และฟังก์ชันจุดประสงค์クロสโอนไทรปี $E = - \sum_k \{y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)\}$.
- (ค) การจำแนกกลุ่ม ใช้ฟังก์ชันซอฟต์แมกซ์ ซึ่งคือ $\hat{y}_k = \frac{\exp(a_k^{(L)})}{\sum_{j=1}^K \exp(a_j^{(L)})}$
และฟังก์ชันจุดประสงค์クロสโอนไทรปี $E = - \sum_j y_j \log(\hat{y}_j)$.

แบบฝึกหัด 3.6

การทำเรกูลาร័ไซซ์ กล่าวง่าย ๆ คือการควบคุมพฤติกรรมการทำนายของแบบจำลอง เพื่อช่วยลดความเสี่ยงการโอเวอร์พិត โดยยังคงความซับซ้อนของแบบจำลองไว้. โครงข่ายประสาทเทียมสามารถทำเรกูลาร័ไซซ์ได้ ดังเช่น ฟังก์ชันจุดประสงค์ในสมการ 3.23 สามารถถูกตัดแปลงเป็น

$$\text{loss}_n = E_n + \frac{\lambda}{2} \sum_q \sum_j \sum_i w_{ji}^2(q) \quad (3.46)$$

เมื่อ $w_{ji}(q) \equiv w_{ji}^{(q)}$ แทนค่าน้ำหนักในชั้น q^{th} ของแบบจำลอง. หมายเหตุ สัญลักษณ์ $w_{ji}(q)$ ใช้แทน $w_{ji}^{(q)}$ เพื่อลดความรุกราน. 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 การทำเรกูลาร័ไซซ์ไม่รวมค่าไบอส.

แบบฝึกหัดเขียนโปรแกรม

แบบฝึกหัด 3.7

จากตัวอย่างการฝึกแบบจำลองพหุนามระดับขั้นหนึ่ง ในหัวข้อ 3.1 โปรแกรมในรายการ 3.3 แสดงตัวอย่างการปรับเส้นโดยด้วยฟังก์ชันพหุนามระดับขั้นหนึ่ง. สามารถทั้งแรกเป็นการเตรียมข้อมูล. บรรทัดที่สี่ เป็นการฝึกแบบจำลอง ซึ่งเรียกว่าโปรแกรม `train_poly1` ที่แสดงในรายการ 3.2. หลังจาก

ฝึกแบบจำลองเรียบร้อย แบบจำลองที่ฝึกเสร็จ (แบบจำลองที่เลือก พร้อมค่าพารามิเตอร์ที่นำมาได้) จะสามารถนำไปใช้งาน ซึ่งคือการทำนายคำตอบ จากค่าที่ถูกต้องได้ บรรทัดสุดท้าย แสดงตัวอย่างที่ทำนายค่า y สำหรับค่า $x = 5$ ซึ่งทำโดยเรียกใช้ฟังก์ชัน `fmodel` ที่โปรแกรมแสดงในรายการ 3.1.

จะทำความเข้าใจโปรแกรมเหล่านี้ ทดสอบโปรแกรม และเปรียบเทียบผลกับตัวอย่างในหัวข้อ 3.1.

รายการ 3.1: ตัวอย่างฟังก์ชันพื้นฐาน

```

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,
2     0.889,1]
3 DY =[0.160,0.724,0.931,0.712,0.610,-0.460,-0.684,-1.299,
4     -1.147,-0.045]
5 X, Y = np.array(DX), np.array(DY)
6 wo = train_poly1(X, Y); print('trained w =\n', wo)
7 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 จงเขียนโปรแกรม เพื่อศึกษาคุณสมบัติความทว่าไประของแบบจำลอง (หัวข้อ 3.2) โดยการสร้างข้อมูล $y = \sin(2\pi x) + \epsilon$ เมื่อ $\epsilon \sim \mathcal{N}(0, 0.3)$ โดยสร้างข้อมูลขึ้นมา 10 จุดข้อมูล สำหรับการฝึก และ 5 จุดข้อมูลสำหรับการทดสอบ. ให้ x อยู่ในช่วง 0 ถึง 1. พร้อมเขียนโปรแกรม เพื่อวัดกราฟดังรูปที่ 3.6 และ 3.7.

คำໃບ ดูคำสั่ง `np.linspace` และ `np.random.normal`.

แบบฝึกหัด 3.10

รายการ 3.4 แสดงโปรแกรมคำนวนโครงข่ายประสาทเทียม. โปรแกรมคำนวนตามสมการ 3.19 และ 3.20. โดยรับ จำนวนชั้นคำนวน ผ่าน `net_params['layers']`. ค่าของพารามิเตอร์ ต่าง ๆ ที่รับผ่าน `net_params` เช่น ค่าใบอัลชันที่หนึ่ง ผ่าน `net_params['bias1']` ค่าน้ำหนักชั้นที่หนึ่ง ผ่าน `net_params['weight1']` โดยตัวเลขตามหลังชื่อระบุชั้นของพารามิเตอร์. ค่าใบอัลชัน เป็นเวกเตอร์ที่มีจำนวนส่วนประกอบเท่ากับจำนวนโนนดในชั้นคำนวน. ค่าน้ำหนัก เป็นเมทริกซ์ขนาด $M_q \times M_{q-1}$ เมื่อ M_q คือจำนวนโนนดของชั้นคำนวน และ M_{q-1} คือจำนวนโนนดของชั้นคำนวนก่อนหน้า. เพื่อความสะดวกในการเขียนโปรแกรม อินพุตถูกกำหนด เป็นแมติริกซ์ X ที่รับเข้าต้องอยู่ในรูปเมทริกซ์ ขนาด $D \times N$ เมื่อ D เป็นจำนวนมิติของอินพุต และ N เป็นจำนวนจุดข้อมูล. โครงข่ายประสาทเทียมจะให้อาตพุตอกมา ในรูปเมทริกซ์ ขนาด $K \times N$ เมื่อ K คือจำนวนมิติของอาตพุตที่ต้องการ.

การกำหนดจำนวนโนนดในแต่ละชั้นคำนวน ทำทางอ้อมผ่านการกำหนดขนาดของค่าใบอัลชัน และขนาดของค่าน้ำหนัก. นอกจากจำนวนชั้นคำนวน ค่าใบอัลชัน และค่าน้ำหนักแล้ว พังก์ชันกระตุนสามารถกำหนดได้ในแต่ละชั้นคำนวน เช่น หากกำหนดพังก์ชันกระตุนของชั้นคำนวนที่หนึ่ง เป็นพังก์ชันจำกัดแข็ง อาจทำโดย การกำหนดค่า `net_params['act1']` ให้เป็น `hardlimit` เมื่อ `hardlimit` ว่างลงพังก์ชันจำกัดแข็งที่โปรแกรมแสดงในรายการ 3.5.

ข้อสังเกต พังก์ชัน `hardlimit` ไม่ได้เขียนโดยใช้คำสั่ง `if`. อะไรคือข้อดีข้อเสีย ของการเขียนโปรแกรมในแบบรายการ 3.5 เปรียบเทียบกับการเขียนโดยใช้คำสั่ง `if`

รายการ 3.4: โปรแกรมคำนวนโครงข่ายประสาทเทียม

```

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: โปรแกรมคำนวณฟังก์ชันจำกัดແ豨້ງ

```

1 def hardlimit(a):
2     return 1*(a > 0)

```

การใช้งานโปรแกรมคำนวณโครงข่ายประสาทเทียม สามารถทำได้ เช่น หากทำการคำนวณตระกูล เอ็กซ์อร์ ในรูปที่ 3.15 การคำนวณสามารถทำได้ดังแสดงในรายการ 3.6. ตัวแปร **net** กำหนดจำนวนชั้นคำนวณ ค่าน้ำหนัก ค่าเบ้อส์ และฟังก์ชันกระตุน.

จะศึกษาโปรแกรมเหล่านี้ ทดลองรัน และปรับแต่งโครงข่ายประสาทเทียม โดยใช้ค่าน้ำหนักและค่าเบ้อส์อื่น หรือปรับแต่งเป็นโครงสร้างอื่น สังเกตผล และสรุป. หมายเหตุ โครงข่ายในรูปที่ 3.15 เป็นโครงข่ายสองชั้น แต่การเรียกใช้โปรแกรม **mlp** กำหนด '**layers**': 3 ซึ่งในโปรแกรม **mlp** นับอินพุตเป็นชั้นคำนวณที่ศูนย์เข้าไปด้วย โดยชั้นคำนวณที่ศูนย์ไม่มีการคำนวณ (เช่น $Z = X$ และเริ่มคำนวณลูปจากด้านที่หนึ่ง ดูรายการ 3.4 ประกอบ).

รายการ 3.6: ตัวอย่างการปรับแต่งโปรแกรมคำนวณโครงข่ายประสาทสำหรับรูปที่ 3.15

```

1 net = {'layers': 3, 'bias1': np.array([[ -20], [-20]]),
2        'weight1': np.array([[ -30,  30],[30,  -30]]),
3        'bias2': np.array([[ -20]]),
4        'weight2': np.array([[30,  30]]),
5        'act1': hardlimit, 'act2': hardlimit}
6
7 x = np.array([[0, 0, 1, 1], [0, 1, 0, 1]])
8 y = mlp(net, x)
9 print(x[0,:])
10 print(x[1,:])
11 print(y[0,:])

```

แบบฝึกหัด 3.11

รายการ 3.7 แสดงโปรแกรมฝึกโครงข่ายประสาทเทียม **train_mlp** ที่ใช้วิธีแพร์กระจายย้อนกลับ คำนวณค่าเกรเดียนต์ และใช้วิธีลงชั้นที่สุด เพื่อปรับค่าพารามิตเตอร์. การปรับค่าพารามิตเตอร์ใช้

ข้อมูลฝึกทุกจุดที่เดียว และการปรับทำครั้งเดียวในแต่ละสมัยฝึก นี้คือ การฝึกแบบหนึ่ง. โปรแกรม **train_mlp** รับแบบจำลอง (พร้อมค่าพารามิเตอร์เริ่มต้น) ผ่านอาร์กิวเม้นต์ **net_params** ซึ่งเป็นไฟลอนดิกชันนารีที่มีกุญแจต่าง ๆ ดังอภิรายในแบบฝึกหัด 3.10. อาร์กิวเม้นต์ **trainX** และ **trainY** เป็นตัวแปรต้น และตัวแปรตามของข้อมูลฝึก ที่อยู่ในรูป $D \times N$ และ $K \times N$ ตามลำดับ เมื่อ D, K , และ N เป็นจำนวนมิติของอินพุต จำนวนมิติของเอาต์พุต และจำนวนจุดข้อมูล ตามลำดับ. อาร์กิวเม้นต์ **loss** อ้างถึงฟังก์ชันสูญเสีย ที่จะใช้เป็นฟังก์ชันจุดประสงค์. อาร์กิวเม้นต์ **lr** แทนอัตราการเรียนรู้. อาร์กิวเม้นต์ **epochs** แทนจำนวนสมัยที่จะฝึก.

โปรแกรม **train_mlp** รีเทิร์นไฟลอนดิกชันนารี **net_params** ที่แทนโครงสร้างแบบจำลอง ซึ่งค่าพารามิเตอร์ได้ถูกปรับเปลี่ยนไปแล้ว จากการฝึก และรีเทิร์นไฟลอลิสต์ **train_losses** ที่บันทึกค่าเฉลี่ยค่าผิดพลาดกำลังสองของแบบจำลองหลังฝึกแต่ละสมัย. สังเกต ขั้นตอนที่สามของการแพร่กระจายย้อนกลับ ในโปรแกรมใช้ **delta[i - 1] = dsigmoid(Z[i - 1]) * sumdw** ซึ่งเรียกใช้อันพันธ์ของซิกมอยด์ ดังนั้น หากขั้นซ่อนใช้ฟังก์ชันgradeตุนอื่น นอกจากซิกมอยด์ จะต้องแก้ไขโปรแกรมที่ส่วนนี้. ในเบื้องต้นนี้ โปรแกรมใช้คำสั่ง **assert act_f == sigmoid** เพื่อป้องกัน ความพลังเหลือที่อาจเกิดขึ้น.

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             w = net_params['weight%d'%i]
31             b = net_params['bias%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             w = net_params['weight%d'%i]      # w: Mnnext x M
44             b = net_params['bias%d'%i]        # b: Mnnext x 1
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())
53         dEb[i] = np.dot(delta[i], np.ones((N, 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         w = net_params['weight%d'%i]

```

```

63     b = net_params['bias%d'%i]
64
65     w -= step_size * dEw[i]
66     b -= step_size * dEb[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 อนุพันธ์ของซิกมอยด์รับอาร์กิวเม้นต์ที่เป็นผลลัพธ์จากการกระตุ้นแล้ว Z ไม่ใช่ผลรวมการกระตุ้น A . (ดูแบบฝึกหัด 3.2 และอภิปรายถึงข้อดีข้อเสียของการเขียนโปรแกรมให้รับอาร์กิวเม้นต์เป็น Z เปรียบเทียบกับการเขียนโปรแกรมให้รับอาร์กิวเม้นต์เป็น A . คำใบ้ โปรแกรมที่มีประสิทธิภาพ ควรลดการคำนวณที่ซ้ำซ้อน.)

รายการ 3.9 แสดงโปรแกรมคำนวณฟังก์ชันเอกลักษณ์ ซึ่งทำหน้าที่ เป็นจุดอ้างอิง เพื่อให้โปรแกรม `mlp` และ `train_mlp` รวมถึงการปรับแต่งโครงข่ายประสาทเทียม ผ่านไพรอนดิกชันนารี `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 แสดงโปรแกรมกำหนดค่า w_{init} สุ่มค่าไปอัสและค่าน้ำหนัก จากการแจกแจงเกาส์เซียน ซึ่งค่าดิฟอลต์คือค่าเฉลี่ยเป็น 0 และค่าเบี่ยงเบนมาตรฐานเป็น 1. โปรแกรมรับอาร์กิวเม้นต์ `Ms` เป็นลิสต์ของเลขที่ระบุจำนวนโนหนดในแต่ละชั้น โดยเริ่มจากจำนวนโนหนดในชั้นอินพุต (ซึ่งคือจำนวนมิติของอินพุต) และตามด้วยจำนวนโนหนดในชั้นคำนวณที่หนึ่ง จนถึงจำนวนโนหนดในชั้นเอ้าต์พุต (ซึ่งเท่ากับจำนวนมิติของเอ้าต์พุตสุดท้าย). โปรแกรมรีเทิร์น

ไฟล์นี้แสดงตัวอย่างการใช้ Python ในการสร้างโครงข่ายประสาทเทียมที่สามารถคำนวณค่าผิดพลาดสำหรับตัวอย่างที่กำหนด ให้ได้โดยใช้ฟังก์ชัน `w_initn` และ `train_mlp` ที่เราได้บรรยายไว้ในบทที่ 3.

รายการ 3.10: โปรแกรมคำนวณค่าผิดพลาดสำหรับตัวอย่างที่กำหนด

```

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 แสดงโปรแกรมคำนวณค่าผิดพลาดสำหรับตัวอย่างที่กำหนด ซึ่งใช้แบบจำลองที่ฝึกแล้วใน

รายการ 3.11: โปรแกรมคำนวณค่าผิดพลาดสำหรับตัวอย่างที่กำหนด

```

1 def sqr_error(yhat, y):
2     assert yhat.shape == y.shape
3     return (yhat - y)**2 # output: K x N

```

จากโปรแกรมต่อไปนี้แสดง การสร้าง การฝึก และใช้แบบจำลองที่ฝึกแล้วในการทำนาย. สมมติ ข้อมูลจำนวน 200 จุดข้อมูล ทั้งตัวแปรต้นและตัวแปรตามมีมิติเดียว 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))

```

สมมติหากเราต้องการใช้โครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยชั้น โครงข่ายประสาทเทียมสามารถสร้าง 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**. การฝึกทุกครั้งควรตรวจสอบความก้าวหน้าในการฝึกว่าดำเนินไปได้ด้วยดี เช่น ทำ **plt.plot(losses)** เพื่อดูว่ากราฟลู่ลงจนราบดีแล้ว. หลังจากฝึกเสร็จแล้ว แบบจำลอง **tnet** สามารถนำไปใช้ทำนายได้ เช่น **Yp = mlp(tnet, x)** เมื่อ **x** เป็นตัวแปรต้นที่ถูก แล้วผลลัพธ์การทำนายคือ **Yp**.

จึงศึกษาโปรแกรมเหล่านี้ ทดลองสร้าง ทดลองฝึก และทดลองใช้แบบจำลองทำนาย รวมไปถึงประเมิน เช่น ลองวัดค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง และลองวัดผลการทำนาย เปรียบเทียบกับข้อมูล อภิราย และสรุปสิ่งที่ได้เรียนรู้.

แบบฝึกหัด 3.12

การกำหนดค่าน้ำหนักเริ่มต้น มีผลอย่างมากต่อการฝึกโครงข่ายประสาทเทียม. จงสร้างข้อมูล (ดูแบบฝึกหัด 3.11) แบ่งข้อมูลออกเป็นข้อมูลฝึก และข้อมูลทดสอบ. จากนั้น เลือกโครงข่ายประสาทเทียมที่เหมาะสม แล้วทดลองฝึกโครงข่ายประสาทเทียม และวัดผลการทำนายกับข้อมูลทดสอบ. ทดลองซ้ำทั้งหมดไม่น้อยกว่า 40 ช้ำ ซึ่งในแต่ละช้ำ ทำการกำหนดค่าน้ำหนักเริ่มต้นใหม่ทุกครั้ง นอกจากนี้ให้ทำอย่างอื่นเหมือนเดิม. การกำหนดค่าน้ำหนักเริ่มต้น ให้ใช้วิธีการสุ่ม (รายการ 3.10) สังเกตผลการทำงานจากการทดลองซ้ำ อภิรายผล และทำการทดสอบเช่นนี้อีก แต่ใช้วิธีเงื่อนวิดโดยร่ว (รายการ 3.12) ในการกำหนดค่าน้ำหนักเริ่มต้น เปรียบเทียบผลที่ได้ อภิราย และสรุปสิ่งที่ได้เรียนรู้.

หมายเหตุ การแบ่งข้อมูล อาจทำได้ดังนี้ เมื่อ **x** และ **y** เป็นตัวแปรต้นและตัวแปรตามของข้อมูลซึ่งมีจำนวน **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 การใช้แบบจำลองทำนายข้อมูลทดสอบ และวัดผลการทำทดสอบด้วยค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาดกำลังสอง (**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)))
```

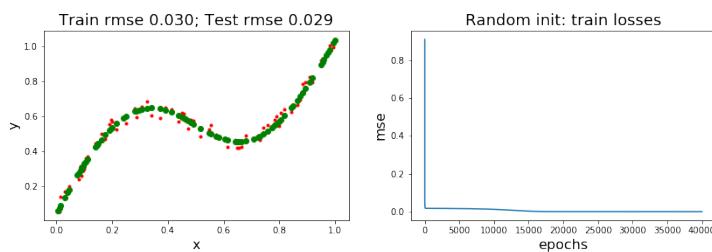
รายการ 3.12: โปรแกรมกำหนดค่าน้ำหนักเริ่มต้นตามแนวทางหنجีนวิดโดร์ [141]

```

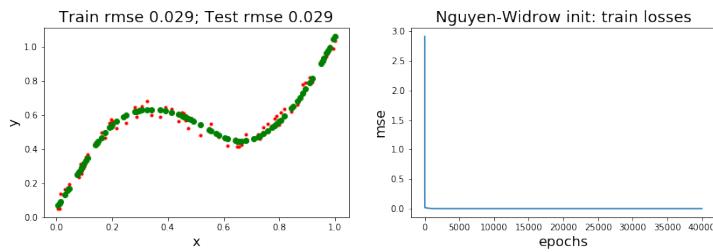
1 def w_initngw(Ms, maxoff=(1.4, 0)):
2     assert len(Ms) >= 2
3     num_layers = len(Ms)
4     params = {'layers': num_layers}
5     scale = maxoff[0]
6     offset = maxoff[1]
7     for i, m in enumerate(Ms[1:], start=1):
8         mprev = Ms[i-1]
9         wmax = scale*m**(1/mprev)
10        wi_ = np.random.rand(m, mprev) * 2 - 1
11        denomi = np.sqrt(np.sum(wi_**2,
12                           axis=1).reshape((-1,1)))
13        wi = wmax * wi_ /denomi
14        bi = wmax * np.linspace(-1, 1, m) \
15                  * np.sign(np.random.rand(m)) + offset
16        params['bias%d'%i] = bi.reshape((m,1))
17        params['weight%d'%i] = wi
18
19    return params

```

รูปที่ 3.25 และ 3.26 แสดงตัวอย่างผลจากการทดลอง ซึ่งเลือกผลที่ดีที่สุด (ค่าผิดพลาดทดสอบ ต่ำสุด) จากการทดลองซ้ำ 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยการสุ่ม และด้วยวิธีหنجีนวิดโดร์ ตามที่ระบุในคำบรรยายรูป.



รูปที่ 3.25 ผลเมื่อสุ่มกำหนดค่าน้ำหนักเริ่มต้น. ตัวอย่างผลการทำนายที่ดีที่สุด จากการทดลอง 40 ครั้ง เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยการสุ่ม.



รูปที่ 3.26 ผลเมื่อกำหนดค่า α ให้เท่ากับ 0.029 ตัวอย่างผลการทำนายที่ดีที่สุด จากการทดลอง 40 ครั้ง เมื่อกำหนดค่า α ให้เท่ากับ 0.029 ตัวอย่างวิธีเหจี้ยนวิดโดร์.

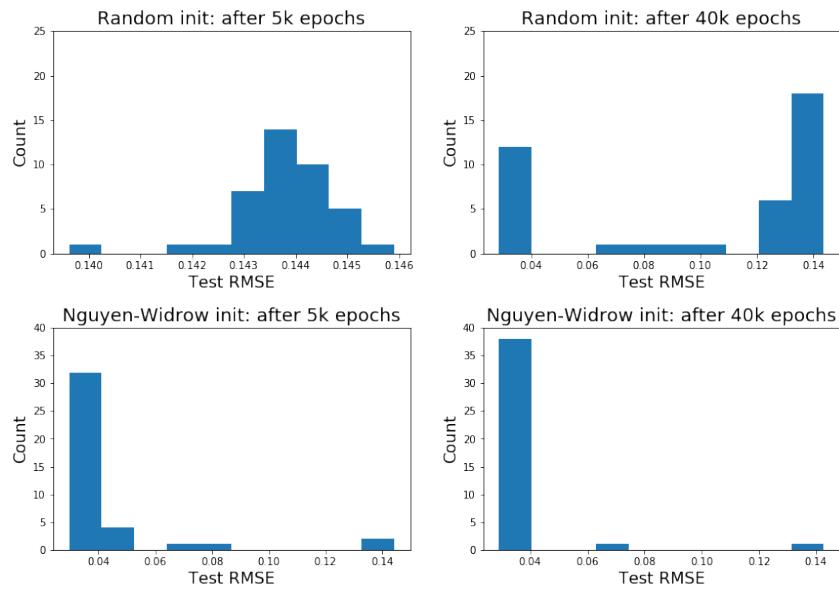
รูปที่ 3.27 และ 3.28 แสดงตัวอย่างวิธีการนำเสนอผลศึกษา. ตารางที่ 3.5 แสดงตัวอย่างค่าสถิติจากการศึกษา. สังเกตจากผลในตัวอย่าง ผลการทำงานของแบบจำลองที่ฝึกแล้ว เมื่อใช้การสุ่มกำหนดค่าเริ่มต้น จะมีความหลากหลายค่อนข้างมาก. เมื่อเปรียบเทียบกับวิธีเหจี้ยนวิดโดร์ (1) ผลลัพธ์ที่ดีที่สุด เมื่อกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม หากฝึกนานพอ ไม่ได้ต่างจาก ผลลัพธ์ที่ดีที่สุด เมื่อกำหนดค่าเริ่มต้นด้วยวิธีเหจี้ยนวิดโดร์. แต่ (2) การกำหนดค่าเริ่มต้นด้วยวิธีเหจี้ยนวิดโดร์ สามารถช่วยให้ได้แบบจำลองที่ดี โดยใช้จำนวนสมัยฝึกที่น้อยกว่า การกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม. (3) ผลลัพธ์โดยเฉลี่ย ของ การกำหนดค่าเริ่มต้นด้วยวิธีเหจี้ยนวิดโดร์ ดีกว่า ผลลัพธ์โดยเฉลี่ย เมื่อกำหนดค่าเริ่มต้นด้วยวิธีสุ่ม. (4) โอกาสที่จะได้แบบจำลองที่ดี เมื่อกำหนดค่าเริ่มต้นด้วยวิธีเหจี้ยนวิดโดร์ 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. เปรียบเทียบผลที่ได้ทำการทดลอง เอง กับผลตัวอย่างที่นำเสนอในรูปที่ 3.27 และตารางที่ 3.5 อภิปรายผลที่ได้ กับข้อสังเกตที่ตั้งไว้นี้ รวมถึงอภิปรายถึงแนวทางปฏิบัติ เมื่อทำแบบจำลองโครงข่ายประสาทเทียม.

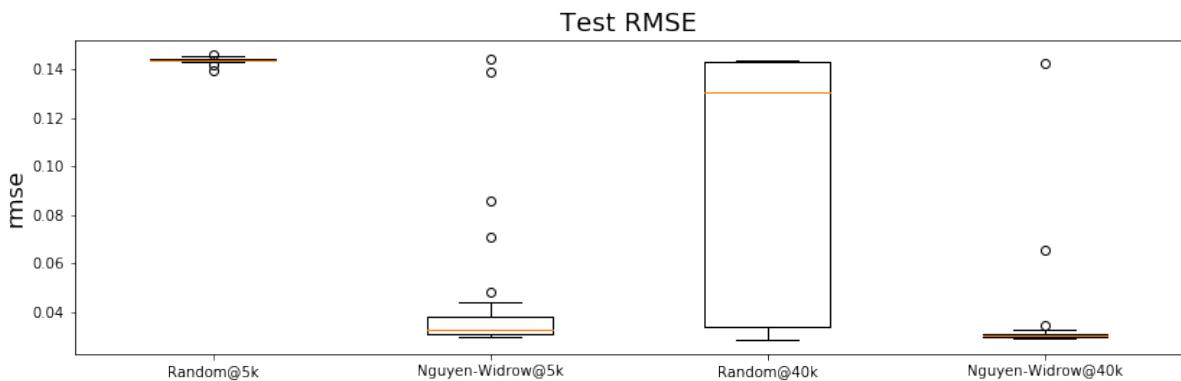
ความสำคัญของการทำซ้ำ. เกี่ยวกับการทำซ้ำ การทำแบบจำลองด้วยโครงข่ายประสาทเทียม นิยมทำซ้ำหลายครั้ง (หากทรัพยากรอำนวย) เนื่องจาก ผลค่อนข้างจะหลากหลาย ดังที่เห็นจากรูปที่ 3.27 โดยเฉพาะอย่างยิ่งเมื่อใช้วิธีการกำหนดค่า α ให้เท่ากับ 0.029 ตัวอย่างเช่น รูปที่ 3.29 แสดงความสำคัญของการทำซ้ำ ซึ่งช่วยให้มีโอกาสมากพอ ที่จะพบแบบจำลองที่ดี หรืออย่างน้อย ก็ช่วยให้ได้เห็นความสามารถโดยเฉลี่ยของแบบจำลองและการฝึกที่ใช้.

โปรแกรมข้างล่างนี้ เป็นตัวอย่างคำสั่งส่วนหนึ่ง (ไม่ใช่ทั้งหมด) ที่ใช้สร้างรูปที่ 3.29.

```
cave = []
cstd = []
cbest = []
```



รูปที่ 3.27 ผลการทำนาย ที่ได้จากการทดลองขึ้น 40 ครั้ง เมื่อใช้วิธีกำหนดค่าน้ำหนักเริ่มต้นแบบต่าง ๆ. สองภาพในแถวบน แสดงอิสโน้ตแกรมของผลที่ได้ เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยวิธีการสุ่ม. สองภาพในแถวล่าง แสดงอิสโน้ตแกรมของผลที่ได้ เมื่อกำหนดค่าน้ำหนักเริ่มต้น ด้วยวิธีเหจីយនិវិត្តទូរគ័រ. ภาพทางซ้าย แสดงผลหลังจากการฝึกไป 5000 សម្រាប់. ภาพทางขวา แสดงผลหลังจากการฝึกไป 40000 សម្រាប់.



รูปที่ 3.28 แผนภูมิกล่อง แสดงผลการทำนาย ที่ได้จากการทดลองขึ้น 40 ครั้ง สำหรับกรณีต่าง ๆ ดังระบุใน/gen/nอน ซึ่ง Random หมายถึง เมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยการสุ่ม. Nguyen-Widrow หมายถึง เมื่อกำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีเหជីយនិវិត្តទូរគ័រ. คำต่อท้าย @5k หมายถึง วัดผลของแบบจำลองที่ได้ทำการฝึกไป 5000 សម្រាប់. คำต่อท้าย @40k หมายถึง วัดผลของแบบจำลองที่ได้ทำการฝึกไป 40000 សម្រាប់.

ตารางที่ 3.5 ค่าสถิติของผลการฝึกแบบจำลอง จากการทดลองซ้ำ 40 ครั้ง เมื่อกำหนดค่า λ นักเรียนต้นตัวอย่างวิธีการสุ่ม และด้วยวิธีเหจิยันวิดโดร์.

การกำหนดค่า λ นักเรียนต้น	ค่าสถิติ	ผลจากแบบจำลองที่ผ่านการฝึก	
		5000 สมัย	40000 สมัย
วิธีสุ่ม	ค่าน้อยที่สุด	0.140	0.029
	ค่าเฉลี่ย	0.144	0.101
	ค่ามากที่สุด	0.146	0.144
วิธีเหจิยันวิดโดร์	ค่าน้อยที่สุด	0.030	0.029
	ค่าเฉลี่ย	0.041	0.034
	ค่ามากที่สุด	0.144	0.143

```

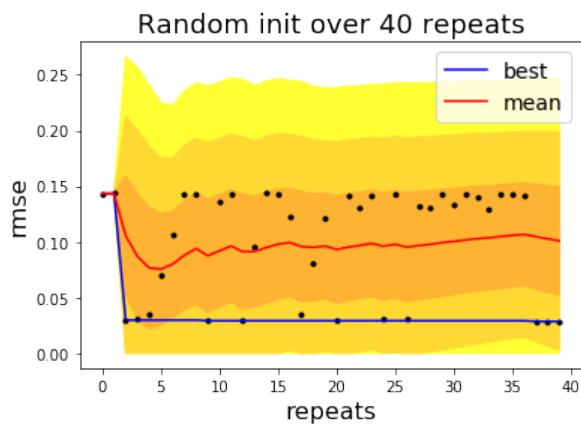
for i in range(40):
    cave.append(np.mean(trmses[::(i+1)]))
    cstd.append(np.std(trmses[::(i+1)]))
    cbest.append(np.min(trmses[::(i+1)]))

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 $trmses$ แทนไฟรอนลิสต์ที่เก็บค่าผิดพลาดทดสอบของแต่ละซ้ำไว้ ทั้งหมด 40 ชั้น. หมายเหตุ ค่าสถิติที่ใช้ทำ แบบความเชื่อมั่น (*confidence intervals*) ในรูป คือ $\mu \pm \sigma$ (แสดงในโปรแกรมตัวอย่าง) และ $\mu \pm 2\sigma$ กับ $\mu \pm 3\sigma$ (ไม่ได้แสดงในโปรแกรมตัวอย่าง). สังเกตว่า แนวทางปฏิบัติ ที่ทำซ้ำหลาย ๆ ครั้ง และเลือกแบบจำลองที่ทำงานได้ดีที่สุด นั้น เป็นเสมือนการหาค่าดีที่สุด อ่อน ๆ (*soft optimization*) ที่ช่วยบรรเทา การติดกับสถานการณ์ที่ดีที่สุดท่องถิน ของขั้นตอนวิธีหาค่าดีที่สุดลงได้บาง.



รูปที่ 3.29 ความสำคัญของการทำซ้ำ เมื่อสุ่มกำหนดค่าเริ่มต้น. ค่าผิดพลาดทดสอบ จากการทำซ้ำ 40 ครั้ง. จุดสีดำ แสดง ค่าผิดพลาดทดสอบ ของแต่ละซ้ำ. ตำแหน่งตามแกนนอนของจุดสีดำ คือดัชนีของการทดลองซ้ำ. ในขณะที่ ผลของแต่ละซ้ำ ก็แตกต่างกันไป แต่ค่าเฉลี่ยของผลลัพธ์ที่ได้ ที่จำนวนซ้ำต่าง ๆ ที่แสดงด้วยเส้นสีแดง จะค่อนข้างนิ่ง เมื่อจำนวนซ้ำมากขึ้น. แกนนอน แสดงจำนวนซ้ำ. แถบสีส้มแก่ ส้มอ่อน และเหลืองแสดงความเชื่อมั่น ของผลที่จะได้ หรือ การประเมินโอกาส ของผลที่จะได้ โดยไม่จากโอกาส sama กับโอกาสเดียวกัน ตามลำดับ. แถบความเชื่อมั่น ประเมินคร่าว ๆ จากค่าสถิติ ที่ได้คำนวณ จากผลลัพธ์ของแต่ละซ้ำ. เส้นสีนำเงิน แสดงค่าดีที่สุด ที่ได้จากการทำซ้ำ ที่จำนวนซ้ำต่าง ๆ.

แบบฝึกหัด 3.13

จากรูปที่ 3.19 จงสร้างข้อมูลชุดบี 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 ϵ สุ่มจากการแจกแจง เอกกรูป. สร้างข้อมูลขึ้นมา 500 จุดข้อมูลสำหรับการฝึก และ 250 จุดข้อมูลสำหรับการทดสอบ. ให้ x อยู่ในช่วง 100 ถึง 110. ทดลองใช้โครงข่ายประสาทเทียมสองชั้น ที่มีจำนวนหน่วยซ่อน 8 หน่วย เลือกอภิมานพารามิเตอร์ เพื่อให้การฝึกสามารถทำงานได้ดี. เปรียบเทียบ (ก) การไม่ทำนอร์มอลайซ์ กับ (ข) การทำนอร์มอลายซ์ให้อยู่ในช่วง $[0, 1]$ และ (ค) การทำนอร์มอลายซ์ให้อยู่ในช่วง $[-1, 1]$ และ (ง) การทำนอร์มอลายซ์ให้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน เป็น 0 และ 1 ตามลำดับ. เปรียบเทียบ อภิปรายผล และสรุป พร้อมเขียนโปรแกรมเพื่อวาดกราฟนำเสนอผลสรุป.

คำใบ้ ดูคำสั่ง `np.random.rand`. การทดลองกรณีง่ายก่อน จะช่วยให้การเลือกค่าอภิมานพารามิเตอร์สะดวกขึ้น. กรณีง่าย หมายถึง กรณีที่น่าจะช่วยให้การฝึกแบบจำลองทำได้ง่ายกว่า.

หมายเหตุ รายการ 3.13 แสดงโปรแกรมสำหรับการนอร์มอลายซ์อินพุตแบบกำหนดช่วง. ศึกษา โปรแกรม ทดลองใช้ อภิปราย และเขียนโปรแกรมสำหรับการนอร์มอลายซ์อินพุตแบบค่าสถิติ ทดสอบ แล้วนำโปรแกรมนอร์มอลายซ์อินพุตทั้งสองแบบ ไปทำการทดลอง.

รายการ 3.13: โปรแกรมนอร์มอลายซ์อินพุต

```

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] ดาวน์โหลดได้ที่

<http://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics>

การกิจ คือการประมาณค่าแรงต้านที่เหลือค้าง (residuary resistance) ของเรือยอชต์ขณะแล่น ซึ่งเป็นปัญหาการหาค่าถดถอย. ข้อมูลชุดนี้มี 308 ระเบียน นั่นคือ มี 308 จุดข้อมูล และแต่ละจุด ข้อมูลจะมี 7 เขตข้อมูล (7 ลักษณะสำคัญ). ตามมุ่มนองของฐานข้อมูล (database) จุดข้อมูล จะเรียก ระเบียน (record) และคุณลักษณะสำคัญต่าง ๆ ในระเบียน จะเรียก เขตข้อมูล (field).

เขตข้อมูลที่หนึ่ง ตำแหน่งตามแนวยาวเรือของศูนย์กลางการลอยตัว (longitudinal position of the center of buoyancy) เป็นลักษณะการกระจายน้ำหนักตามแนวยาว ที่ช่วยอธิบายว่า น้ำหนัก กระจายอยู่หน้าลำ กลางลำ หรือท้ายลำอย่างไร. เขตข้อมูลที่สอง ค่าสัมประสิทธิปริซึ่ม (prismatic coefficient) เป็นลักษณะที่ช่วยอธิบายรูปทรงของห้องเรือ โดยวัดจากอัตราส่วนของปริมาตรที่อยู่ใต้ น้ำของห้องเรือ เปรียบเทียบกับปริมาตรของรูปทรงปริซึ่มที่มีความยาวเท่ากัน และพื้นที่หน้าตัดเท่ากับ พื้นที่หน้าตัดที่กว้างที่สุดของห้องเรือ. เขตข้อมูลที่สาม อัตราส่วนความยาวเรือกับการกระจัด (length-displacement ratio) เป็นลักษณะที่ช่วยบ่งชี้ถึงความหนักของเรือ เมื่อเทียบกับความยาว เรือที่หนัก จะมีค่านี้มาก เรือที่เบาจะมีค่านี้น้อย. เขตข้อมูลที่สี่ อัตราส่วนความกว้างเรือกับระดับจน้ำ (beam-draught ratio) เป็นลักษณะทรงท้องเรือที่วัดจาก อัตราส่วนความกว้างเรือ กับความกว้างของส่วน ที่กว้างที่สุดของเรือในแนวระดับน้ำ. เขตข้อมูลที่ห้า อัตราส่วนความยาวกับความกว้างเรือ (length-beam ratio). เขตข้อมูลที่หก ตัวเลขฟรูด (Froude number) เป็นค่าที่บอกความต้านทานของการ ที่วัดถูกเคลื่อนที่ในน้ำ. เขตข้อมูลทั้งหกนี้ บรรยายลักษณะรูปร่าง และการกระจายน้ำหนักของห้องเรือ และเขตข้อมูลเหล่านี้จะใช้เป็นอินพุตของแบบจำลอง.

เขตข้อมูลที่เจ็ด (Residuary resistance per unit weight of displacement) เป็นค่าความต้านทานเหลือค้าง ซึ่งเป็นแรงต้านสำคัญที่เกิดกับเรือ และเกี่ยวพันกับลักษณะต่าง ๆ ของเรือ (ที่บรรยาย

ด้วยหากเขตข้อมูลแรก). การประมาณค่าความต้านทานเหลือค้างได้แม่นยำ จะช่วยในการประเมินสมมติฐานของเรือได้ดี รวมถึงช่วยเป็นข้อมูลประกอบ สำหรับการออกแบบเรือด้วย เช่น การเลือกรูปทรงและขนาดของห้องห้องเรือ การกำหนดน้ำหนักบรรทุกของเรือ การเลือกขนาดของเครื่องยนต์ที่เหมาะสม. (ดูคำอธิบายเพิ่มเติมจากคลังข้อมูลยูซีโอ และอรทิโกสาและคณะ[144].)

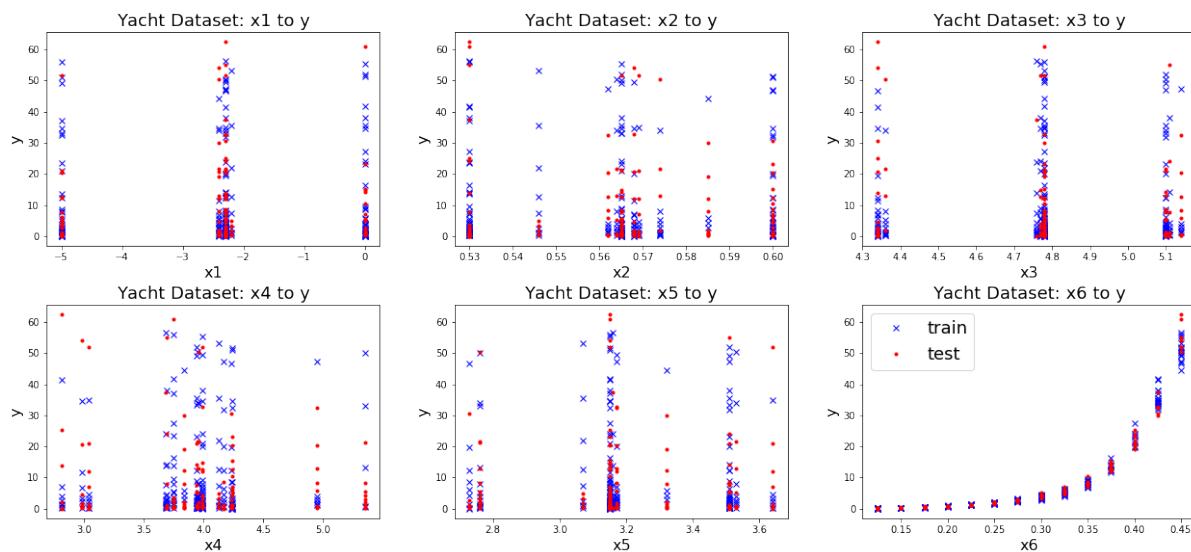
ข้อมูลนี้ 308 ตัวอย่าง และสามารถอ่านข้อมูลเข้าได้ดังเช่นไฟล์ข้อความทั่วไป ดังแสดงในโปรแกรมตัวอย่างข้างล่าง เมื่อไฟล์ข้อมูลถูกดาวน์โหลดมาในชื่อ `yacht_hydrodynamics.data`.

```
with open('yacht_hydrodynamics.data', 'r') as f:
    yacht = f.read()
```

ข้อมูลที่นำเข้ามาจะอยู่ในรูปข้อความ เพื่อประสิทธิภาพในการประมวลผล ควรจัดข้อมูลเข้าในรูปนัมไฟอาร์เรย์ก่อน 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 308×7 . รูปที่ 3.30 แสดงความสัมพันธ์ระหว่างเขตข้อมูลที่หนึ่งถึงหก (ทีละเขตข้อมูล) กับเขตข้อมูลที่เจ็ด (ที่เป็นตัวแปรตาม) หลังจากแบ่งข้อมูลแล้ว. ในตัวอย่างนี้ ข้อมูล 185 จุดข้อมูล (ประมาณ 60%) ถูกแบ่งเป็นชุดฝึก (`trainx` และ `trainy`) และที่เหลือเป็นชุดทดสอบ (`testx` และ `testy`).



รูปที่ 3.30 ชุดข้อมูลเรือยอชต์ หลังจากแบ่งข้อมูลแล้ว. แต่ละภาพแสดงความสัมพันธ์ของแต่ละเขตข้อมูลของตัวแปรต้น กับตัวแปรตาม. ภาคบาทสีน้ำเงิน แทนจุดข้อมูลที่ถูกแบ่งไปชุดฝึก. จุดสีแดง แทนจุดข้อมูลที่ถูกแบ่งไปชุดทดสอบ.

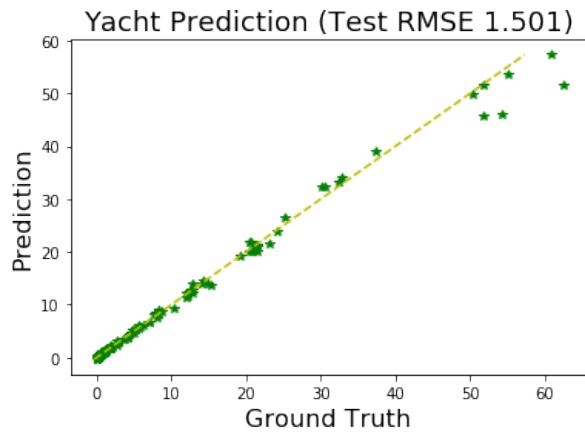
จากรูปที่ 3.30 สังเกตว่า อินพุตมิติต่าง ๆ มีขนาดแตกต่างกันพอสมควร เช่น x_1 (ภาพซ้ายบน) มีขนาดตั้งแต่ -5 ถึง 0 ในขณะที่ x_6 (ภาพขวาล่าง) มีขนาดไม่เกิน 0.5 . เพื่อช่วยให้การฝึกแบบจำลองทำได้ง่ายขึ้น สถานการณ์เช่นนี้ ควรnorrmalize ให้เป็นอินพุต (หัวข้อ 3.4).

จะสร้างแบบจำลองโครงข่ายประสาทเทียม เพื่อประมาณค่าความต้านทานเหลือค้าง จากคุณลักษณะทั้งหกของเรือ โดยใช้ชุดข้อมูลเรือยอชต์ ในการฝึก และการทดสอบ โดยแบ่งข้อมูลให้เรียบร้อย ทำนองร์มอลайเซอินพุต อภิปรายว่า กรณีนี้การนอร์มอลายเซนิดได (ระหว่างนอร์มอลายเซเข้าสู่ช่วงที่จำกัด กับนอร์มอลายเซเข้าสู่ค่าสถิติที่กำหนด) เหมาะสมมากกว่ากัน พร้อมเหตุผลประกอบ. หลังจากฝึกและทดสอบเสร็จ นำเสนอผลให้ชัดเจน. รูปที่ 3.31 แสดงตัวอย่างการนำเสนอผล.

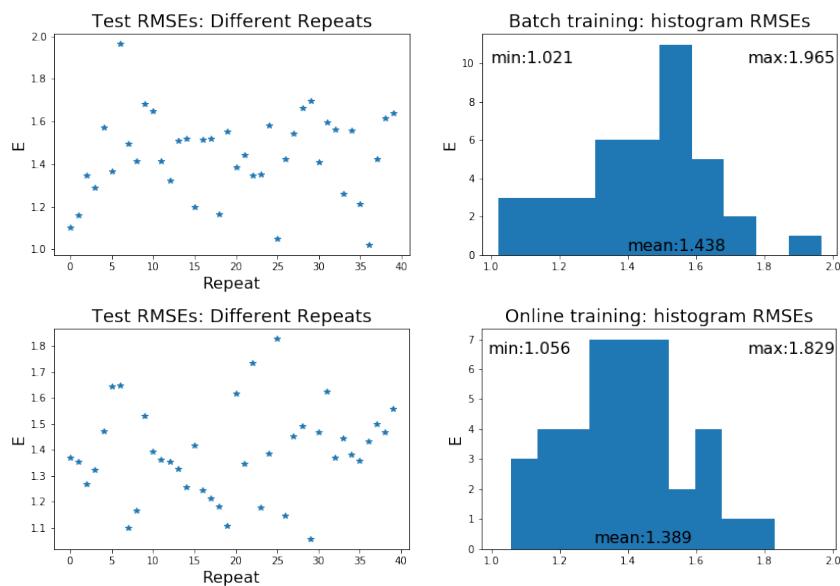
การฝึกด้วยโปรแกรม ดังรายการ 3.7 ซึ่งใช้ข้อมูลฝึกทั้งหมดในการปรับค่าพารามิเตอร์ที่เดียว เป็นการฝึกแบบหนู่ (หัวข้อ 3.3). เมื่อสามารถทำแบบจำลองประมาณค่าความต้านทานเหลือค้างได ดีพอกับแล้ว ศึกษาความแปรปรวนของผล ด้วยการทำซ้ำ และเปรียบเทียบผลกับการฝึกแบบออนไลน์.

รายการ 3.14 แสดงตัวอย่างโปรแกรมฝึกโครงข่ายประสาทเทียม โดยใช้วิธีการฝึกแบบออนไลน์ (เปรียบเทียบกับรายการ 3.7). รูปที่ 3.32 แสดงผลความแม่นยำ ที่ได้จากการฝึกแบบหนู่ เปรียบเทียบ กับแบบออนไลน์ จากการทดลองซ้ำ 40 ครั้ง.

ในกรณี ดังผลที่แสดงในรูปที่ 3.32 ความแม่นยำที่ได้ ไม่ได้แตกต่างกันเท่าไร แต่เวลาในการฝึกต่างกันมาก เช่นในการทดลองตัวอย่าง การฝึกแบบออนไลน์ใช้เวลาเป็น 36 เท่าของเวลาฝึกแบบหนู่.



รูปที่ 3.31 ตัวอย่างการนำเสนอผลการทำนายชุดข้อมูลเรือยacht. จุดข้อมูลทดสอบ ที่ทำแทนงangenonแทนด้วยค่าเฉลย และทำแทนงangenตั้งแทนด้วยค่าที่แบบจำลองทำนาย. เสน่ประ แสดงตำแหน่งที่ค่าเฉลยและค่าทำนายเท่ากัน. ดังนั้น จุดที่อยู่ไกล้เสนประแสดงถึงความแม่นยำในการทำนาย. ผลตัวอย่างนี้ ได้จากโครงข่ายประสาทเทียมสองชั้นขนาด 16 หน่วยช่อง. การฝึกใช้อัตราเรียนรู้ 0.1 ฝึก 5000 สมัย และกำหนดค่าน้ำหนักเริ่มต้นด้วยวิธีเหงี่ยนวิดโดยรุ่ว.



รูปที่ 3.32 ชุดข้อมูลเรือยacht เปรียบเทียบการฝึกแบบหมุ่ และการฝึกแบบออนไลน์. ผลความแม่นยำ ที่ได้จากการฝึกแบบหมุ่ เปรียบเทียบกับแบบออนไลน์ จากการทำลองซ้ำ 40 ครั้ง. ภาพในแถบนี้ แสดงผลเมื่อฝึกแบบหมุ่. ภาพในแถบล่าง แสดงผลเมื่อฝึกแบบออนไลน์

รายการ 3.14: โปรแกรมฝึกโครงข่ายประสาทเทียมแบบออนไลน์

```

1 def train_online(net_params,trainX,trainY,loss,lr,epochs):
2     num_layers = net_params['layers']
3     last_layer = num_layers-1
4     out_act = 'act%d'%last_layer

```

```

5     _, N = trainX.shape
6     A = {}
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!
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/dy
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

```

แบบฝึกหัด 3.15

ข้อมูลภาพเอ็กซเรย์เต้านมของมวลเนื้อ (Mammographic Mass dataset) จากคลังข้อมูลยูซีไอ สามารถดาวน์โหลดได้ที่

<http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass>

เอกสารและคณะ[62] ศึกษาการนำผลการตรวจภาพเอ็กซเรย์เต้านม ของร่องรอยมวลเนื้อ (mammographic mass lesion) ว่าเป็นเนื้อดี (benign) หรือเนื้อร้าย (malignant) จากค่าลักษณะสำคัญต่าง ๆ ของของภาพ ประกอบกับอายุของผู้ป่วย. วิธีตรวจภาพเอ็กซเรย์เต้านม (mammography) เป็นวิธีที่มีประสิทธิผลมากในการตรวจมะเร็งทรวงอก [62]. ข้อมูลชุดนี้ประกอบด้วย ค่าการประเมินใบแรตส์ (BI-RADS assessment เป็นค่าเชิงเลขลำดับ ordinal values), อายุของผู้ป่วย (เลขจำนวนเต็ม), รูปทรงของมวลเนื้อ (mass shape ซึ่งถูกแทนด้วย 1 สำหรับทรงกลม round, 2 สำหรับทรงรี oval, 3 สำหรับทรงกลีบย้อย lobular, 4 สำหรับทรงที่ผิดแปลง irregular), ลักษณะขอบของมวลเนื้อ (mass margin ซึ่งถูกแทนด้วย 1 สำหรับเขตรอบซัดเจน circumscribed, 2 สำหรับขอบเขตเป็นกลีบย้อย ฯ microlobulated, 3 สำหรับขอบเขตคลุมเครือ obscured, 4 สำหรับขอบเขตยากจะระบุ ill-defined, 5 สำหรับขอบเขตเป็นลักษณะหนามหรือปุ่ม spiculated), ความหนาแน่นของมวลเนื้อ (mass density ซึ่งถูกแทนด้วย 1 สำหรับความหนาแน่นสูง high, 2 สำหรับความหนาแน่นกลาง ฯ iso, 3 สำหรับความหนาแน่นต่ำ low, 4 สำหรับมวลเนื้อมีไขมันอยู่ fat-containing) และความร้ายแรง (severity ซึ่งมีสองค่า 0 สำหรับเนื้อดี หรือ 1 สำหรับเนื้อร้าย). ค่าความร้ายแรง คือค่าเป้าหมาย ที่ต้องการทำนาย. การสามารถทำนายค่าความร้ายแรงได้อย่างแม่นยำจะช่วยให้แพทย์และผู้ป่วยสามารถตัดสินใจได้ดีขึ้นว่าควรจะทำการตัดเนื้อจากบริเวณที่สงสัยออกตรวจเพื่อยืนยันผลหรือไม่.

ข้อมูลชุดนี้มี 961 ระเบียน เฉลยหรือผลการตรวจจริง ระบุ 516 ระเบียนที่ผลเป็นเนื้อดี (ค่าความร้ายแรง เป็นศูนย์) และ 445 ระเบียนที่ผลเป็นเนื้อร้าย (ค่าความร้ายแรง เป็นหนึ่ง). ข้อมูลชุดนี้มีค่าบางค่าของเขตข้อมูลที่ไม่ครบ (missing attribute values) ได้แก่ ค่าการประเมินใบแรตส์ขาดไป 2 ค่า, อายุขาดไป 5 ค่า, รูปทรงของมวลเนื้อขาดไป 31 ค่า, ลักษณะขอบของมวลเนื้อขาดไป 48 ค่า และ

ความหนาแน่นของมวลเนื้อหาดไป 76 ค่า. ส่วนความร้ายแรงมีค่าครบทุกระเบียน.

จงทำแบบจำลองโครงข่ายประสาทเทียมเพื่อทำนายค่าความร้ายแรง จากลักษณะสำคัญของข้อมูลภาพเอ็กซเรย์เต้านม เลือกแบบจำลอง ฝึก ทดสอบ วัดผล รายงานผลที่ได้ อกบราย และสรุป. ภาระกิจนี้ เป็นการจำแนกค่าทวิภาค ที่เอาต์พุต คือความร้ายแรง ซึ่งมีค่าเป็นหนึ่งหรือศูนย์. การจัดการกับข้อมูลขาดหาย สามารถดำเนินการได้หลายแนวทาง. แนวทางหนึ่ง ซึ่งเป็นแนวทางที่ง่ายที่สุด คือตัดระเบียนที่มีข้อมูลขาดหายทิ้งทั้งทั้งระเบียน.

คำสั่งข้างล่าง แสดงตัวอย่างการอ่านไฟล์ข้อมูล `mammographic_masses.data` และเตรียมข้อมูล

```
with open('mammographic_masses.data', 'r') as f:
    mammo_text = f.read()

mammo1 = []
mammo_miss = []
for line in mammo_text.split('\n'):
    row = []
    complete = True
    for field in line.split(','):
        try:
            val = int(field)
        except:
            complete = False
            val = None
        row.append(val)
    if complete:
        mammo1.append(row)
    else:
        mammo_miss.append(row)
# end for line
mammo1 = np.array(mammo1)
```

ซึ่งผลลัพธ์จะคือ `mammo1` ซึ่งเป็นแมทริเซอร์ขนาด 830×6 ซึ่งทุกແຕວ (ทุกระเบียนข้อมูล) ครบถ้วนสมบูรณ์ ไม่มีข้อมูลขาดหาย. ในขณะที่ `mammo_miss` เป็นลิสต์ของระเบียนข้อมูลที่เหลือ และทุกระเบียนใน `mammo_miss` มีข้อมูลที่ขาดหายไป (ดูย่อหน้า การจัดการกับข้อมูลขาดหาย).

ข้อมูลของ `mammo1` แสดงในรูปที่ 3.33, 3.34 และ 3.35. สังเกตว่า ข้อมูลภาพเอ็กซเรย์เต้านมนี้ อินพุตมิติแรก (รูปที่ 3.33 และคำอธิบายข้อมูล ที่สามารถดาวน์โหลดได้พร้อมไฟล์ข้อมูล) เป็นค่าเชิงเลขลำดับ ซึ่งสามารถคิดเสมอว่าเป็นค่าตัวเลขได้. อินพุตมิติที่สอง เป็นจำนวนเต็มแสดงอายุ ซึ่งเป็นตัวเลขจริง ๆ. แต่อินพุตมิติที่สามถึงห้า (รูปที่ 3.35) เป็นค่าแทนซื้อ (*nominal values*) ซึ่งไม่ได้มี

ความหมายเป็นปริมาณตามขนาดใหญ่เล็กของตัวเลขจริง ๆ ตัวเลขทำหน้าที่ แค่เป็นดัชนีเพื่ออ้างถึงชื่อเท่านั้น.

รูปที่ 3.36 แสดงผลลัพธ์ของการทำนาย ด้วยแผนภูมิกอล์ง จากตัวอย่างผลการทำนายของ แบบจำลองที่ทำโดยไม่สนใจความหมายของอินพุต (กล่องซ้าย ระบุด้วยชื่อ **Naive**) และแบบจำลองที่ทำการเข้ารหัสอินพุตให้เหมาะสมกับความหมาย (กล่องขวา ระบุด้วยชื่อ **Coding**). แบบจำลองที่ทำโดยไม่สนใจความหมายของอินพุต นั่นคือ ใช้ค่าตัวเลขของข้อมูลใส่เป็นอินพุตให้กับแบบจำลองโดยตรง เช่น ในตัวอย่างข้างล่าง ที่เพียงทำนอร์มอลайซ์อินพุต

```
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** เป็นดัชนีที่สุ่มเลือก เพื่อใช้เป็นข้อมูลฝึก และข้อมูลทดสอบ ตามลำดับ. ตัวอย่างผลที่แสดงในแบบฝึกหัดนี้ ได้จากการใช้ข้อมูลจำนวน 300 จุดข้อมูลเป็นข้อมูลทดสอบ และใช้ที่ข้อมูลที่เหลือเป็นข้อมูลฝึก (ดูตัวอย่างวิธีแบ่งข้อมูล จากแบบฝึกหัด 3.12). โปรแกรม **normalize1**, **w_initn**, **sigmoid** และ **train_mlp** ได้อธิบายไปแล้ว ดังรายการ 3.13, 3.10, 3.8 และ 3.7 ตามลำดับ. ส่วนโปรแกรม **binaries_cross_entropy** เป็นฟังก์ชันสัญญาณสองทางแบบทวิภาค ที่นิยมใช้กับงานจำแนกค่าทวิภาค แสดงในรายการ 3.15.

รายการ 3.15: ฟังก์ชันสัญญาณสองทางแบบทวิภาค

```
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 Y_p เป็นค่าที่ทำนายจากแบบจำลอง ซึ่งมีค่าอยู่ในช่วง $[0, 1]$ และเพื่อบังคับให้ผลตัดสินใจเป็น 0 หรือ 1 จึงใช้โปรแกรม **cutoff** ช่วย. ผลประเมินความแม่นยำของแบบจำลองจำแนกทวิภาค อาจวัดด้วยค่าความแม่นยำ **accuracy** ที่มีค่าระหว่างศูนย์ถึงหนึ่ง และค่าใกล้หนึ่งหมายถึงความแม่นยำสูง. โปรแกรม **cutoff** เขียนด้วยคำสั่งดังนี้

```
def cutoff(a, tau=0.5):
    return (a > tau)*1
```

ผลลัพธ์ที่แสดงในรูปที่ 3.36 ได้มาจากการทดลองซ้ำ 40 ครั้ง. แบบจำลอง **naive** สร้างโดยไม่สนใจความหมายของข้อมูล ดังได้อธิบายไป. ส่วนแบบจำลอง **coding** สร้างโดยแปลงอินพุตเป็นรหัสตามความหมายของลักษณะสำคัญ แต่ละอย่าง โดยเฉพาะลักษณะสำคัญที่สามถึงห้า ซึ่งเป็นค่าแทนซึ่งได้แก่ รูปทรงของมวลเนื้อ ลักษณะขอบของมวลเนื้อ และความหนาแน่นของมวลเนื้อ. แบบจำลอง **coding** เข้ารหัสลักษณะทั้งสาม ด้วยรหัสหนึ่งร้อน โดยคำสั่ง **trainxc, cparams = mammo_coding(trainx)**. นอกจากลักษณะสำคัญทั้งสาม ลักษณะสำคัญที่หนึ่งถูกเข้ารหัสเป็นระดับค่า และลักษณะสำคัญที่สอง ซึ่งเป็นอายุ ถูกนอร์มอลайซ์ด้วยค่าสถิติ.

โปรแกรม **mammo_coding** สามารถเขียนด้วยคำสั่งดังนี้

```
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** เป็นโปรแกรมที่ทำนอร์มอลายซ์อินพุต (ดูแบบฝึกหัด 3.13) และโปรแกรม **coding**, **onehot_cbook** และ **level_cbook** เขียนดังนี้

```

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 แต่ช่วยให้ผลการทำงานดีขึ้นอย่างชัดเจน ดังตัวอย่างผลลัพธ์ที่แสดงในรูปที่ 3.36. รูปที่ 3.36 รายงานผลด้วยค่าความแม่นยำในช่วง 0 ถึง 1. บางครั้ง ค่าความแม่นยำนิยมรายงานเป็นเปอร์เซ็นต์ เช่น ค่าความแม่นยำเฉลี่ยเป็น 80.9% และ 82.3% สำหรับแบบจำลอง **naive** และแบบจำลอง **coding**.

นอกจากการรายงานผลด้วยค่าความแม่นยำแล้ว เมทริกซ์ความสับสน (confusion matrix) เป็นการแจกแจงผลการทำงานออกเป็นสี่ชนิด คือ จำนวนบวกจริง (true positive คำย่อ TP), จำนวนบวกเท็จ (false positive คำย่อ FP), จำนวนลบจริง (true negative คำย่อ TN) และจำนวนลบเท็จ (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. ลักษณะการแจกแจงเช่นนี้ ทำให้ค่าความแม่นยำ สามารถสะท้อนคุณภาพการทำนายจริงของแบบจำลองได้ดี. แต่หากการแจกแจงไม่สมดุลอย่างมาก เช่น สมมติอัตราส่วนคนเป็นมะเร็งตับอ่อนต่อประชากรมีค่าน้อยกว่า 1% เพียงแต่แบบจำลองทำนายผลเป็น 0 คือไม่เป็นมะเร็ง สำหรับทุก ๆ ตัวอย่างที่เข้ามาทดสอบแบบจำลองนั้นก็มีโอกาสสูญถึง 99% (ความแม่นยำ เป็น 0.99) นั่นคือ เท่ากับทายว่าไม่มีใครเป็นมะเร็งตับอ่อนเลย ซึ่งแม้จะมีโอกาสทายถูกสูงมาก ๆ แต่มันไม่ได้มีประโยชน์ต่อการช่วยระบุกลุ่มเสี่ยงเลย.

ตั้งนั้นแทนที่จะใช้แค่ค่าความแม่นยำ ค่าความเที่ยงตรง (สมการ 3.48) และค่าการระลึกกลับ (สมการ 3.49) จะช่วยสะท้อนคุณภาพการทำนายได้ดีกว่า โดยเฉพาะสำหรับข้อมูลที่มีการแจกแจงข้อมูลไม่สมดุล. การตรวจสอบค่าความเที่ยงตรง และค่าการระลึกกลับ จะช่วยบอกได้ว่าแบบจำลองทำนายได้ดีอย่างสมดุลกับผลทำนาย. สำหรับงานจำแนกค่าทวิภาค ค่าความเที่ยงตรง P สามารถคำนวณได้จาก

$$P = \frac{TP}{TP + FP} \quad (3.48)$$

และค่าการระลึกกลับ R สามารถคำนวณได้จาก

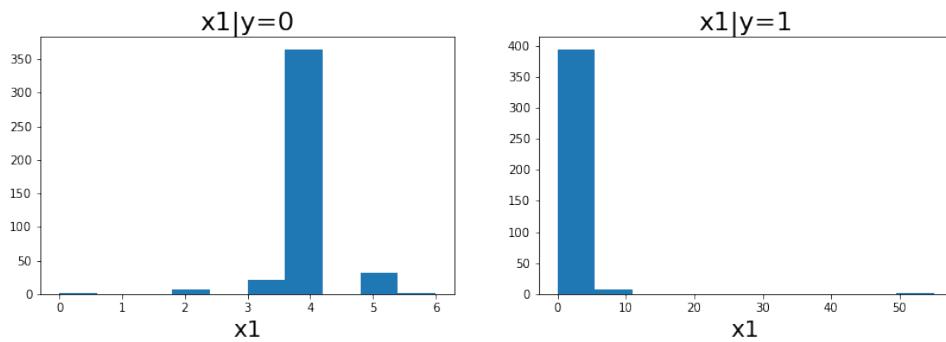
$$R = \frac{TP}{TP + FN} \quad (3.49)$$

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 แต่อีกแบบจำลองหนึ่ง มีค่าความเที่ยงตรงต่ำ แต่ค่าการระลึกกลับสูง. เพื่อความสะดวก ค่าคะแนนเอฟ (F Score หรือ บางครั้งเรียก $F1$ Score) ซึ่งเป็นค่าเฉลี่ยเชิงเรขาคณิต มักใช้ในการสรุป ค่าความเที่ยงตรง และค่าการระลึกกลับ ให้เป็นตัวเลขเดียว. ค่าคะแนนเอฟ สัญกรณ์ F สามารถคำนวณได้จาก

$$F = 2 \cdot \frac{P \cdot R}{P + R} \quad (3.50)$$

จากเมทริกซ์ความสัมสัมในตัวอย่าง ค่าคะแนนเอฟ จะเท่ากับ 0.836.



รูปที่ 3.33 ข้อมูลภาพเอ็กซเรย์เต้านม ลักษณะสำคัญมิติแรก ค่าประเมินไบแรตส์ สำหรับกรณีก้อนเนื้อเป็นเนื้อดี (ภาพชั้ย) และเนื้อร้าย (ภาพขาว).

สุดท้าย ย้อนกลับมาทบทวนเรื่องผลลัพธ์สุดท้าย จากหัวข้อ 2.2 ตัวอย่าง ปัญหาการตรวจเต้านมด้วยภาพเอ็กซเรย์. ตัวอย่างนั้น ยกไปยารการคำนวณหา โอกาสที่จะเป็นมะเร็ง เมื่อผลการตรวจนะบุว่าเป็น. นั่นคือ หา $\Pr(C = 1|M = 1)$ เมื่อ $M = 1$ หมายถึงผลการตรวจนะบุว่าเป็นมะเร็ง และ $C = 1$ หมายถึงการเป็นมะเร็งจริง ๆ.

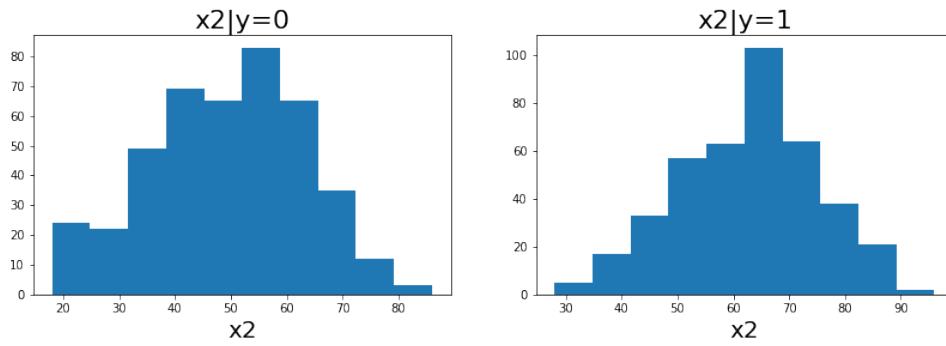
ข้อมูลประกอบ คือ 17% ของผู้หญิงอายุเกิน 40 ปีเป็นมะเร็งเต้านม นั่นคือ $\Pr(C = 1) = 0.17$. ค่า $\Pr(M = 1|C = 1)$ สามารถประมาณได้จากค่าในเมทริกซ์ความสัมสัม $\Pr(M = 1|C = 1) \approx R$ เมื่อ R คือค่าระลึกกลับ ซึ่งในตัวอย่างนี้เป็น 0.839 และค่า $\Pr(M = 1|C = 0)$ ประเมินได้จาก $\Pr(M = 1|C = 0) \approx \frac{FP}{FP+TN}$ ซึ่งในตัวอย่างนี้เป็น $\frac{26}{26+119} = 0.179$.

เมื่อรวมหลักฐานทุกอย่างเข้าด้วยกัน โดยใช้กฎของเบเยร์ จะได้ว่า

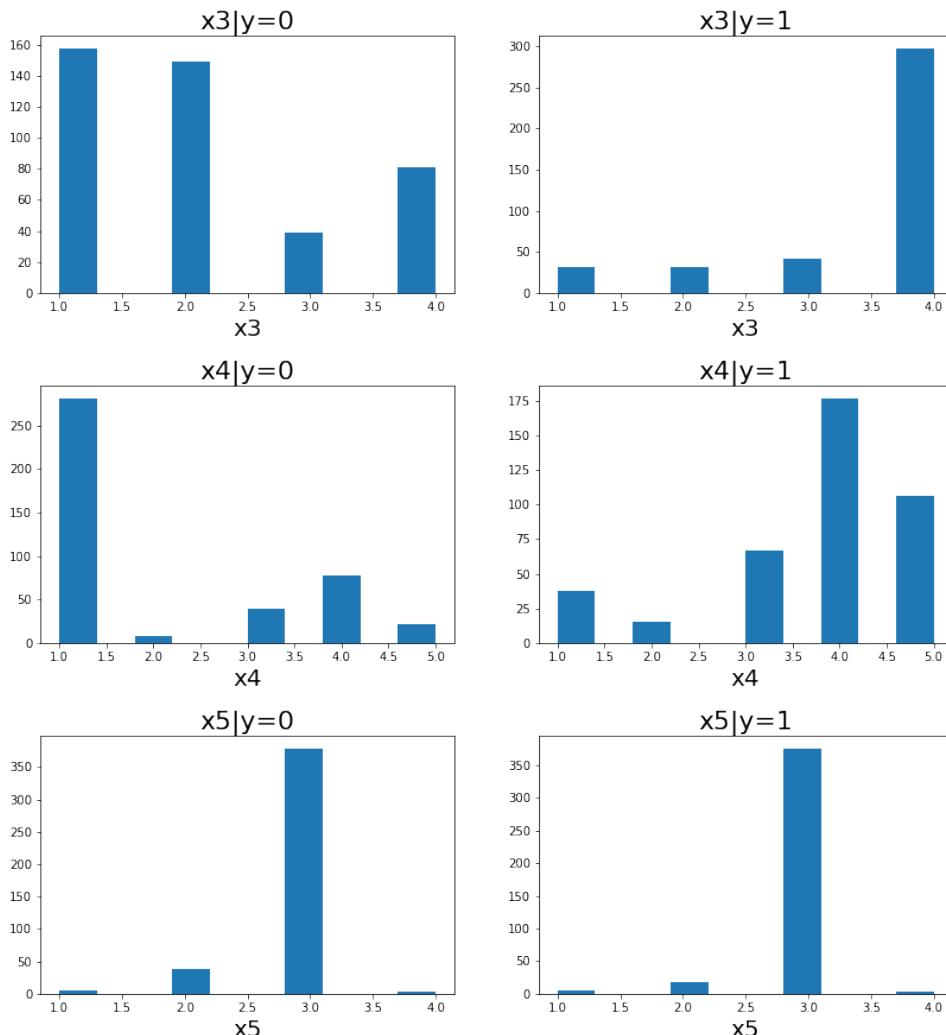
$$\begin{aligned}\Pr(C = 1|M = 1) &= \frac{\Pr(M = 1|C = 1)\Pr(C = 1)}{\Pr(M = 1|C = 0)\Pr(C = 0) + \Pr(M = 1|C = 1)\Pr(C = 1)} \\ &= \frac{0.839 \cdot 0.17}{0.179 \cdot 0.83 + 0.839 \cdot 0.17} = 0.4898.\end{aligned}$$

ดังนั้น ผลตรวจภาพเอ็กซเรย์เต้านม จึงเป็นเสมือนแค่การตรวจเบื้องต้น เพื่อประกอบการตัดสินใจ ตัดขั้นเนื้อไปตรวจ (*biopsy*).

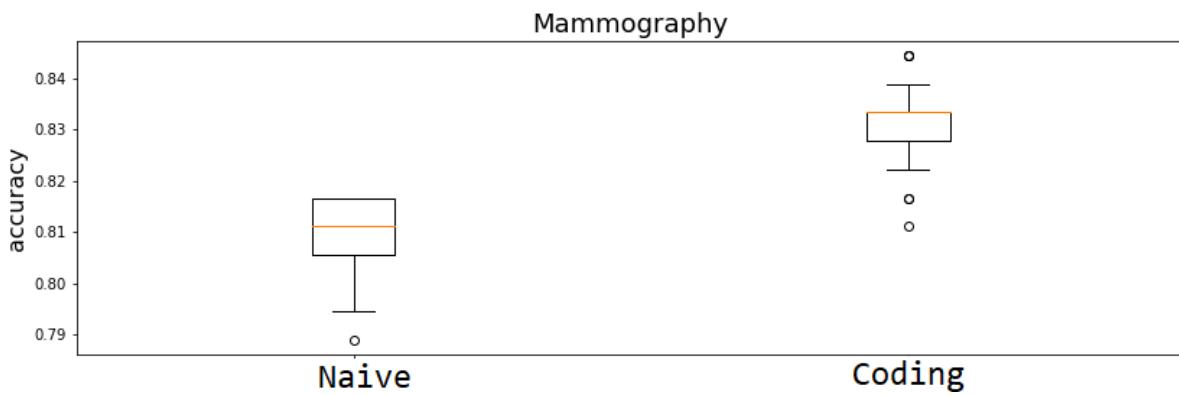
การจัดการกับข้อมูลขาดหาย. แบบฝึกหัด 3.15 แนะนำวิธีตัดระเบียนที่มีข้อมูลขาดหายทิ้งไป ซึ่งเป็นหนึ่งในแนวทางที่นิยม และดำเนินการได้ง่าย. นอกจากแนวทางตัดระเบียนทิ้ง ยังมีแนวทางอื่น ๆ อีก เช่น วิธีการแทนค่าข้อมูลที่ขาดหายไป ด้วยค่าเฉลี่ย สำหรับมิติหรือเขตข้อมูลที่เป็นค่าต่อเนื่อง (continuous-value field) หรือแทนด้วยค่าที่พบบ่อยที่สุด ในกรณีที่เขตข้อมูลเป็นค่าแทนชื่อ ฉลาก หรือหมวดหมู่ (categorical field). อีกวิธีที่นิยม คือ วิธีการแทนค่าที่หายไป ด้วยค่าทุกค่าที่เป็นไปได้. วิธีนี้ ระเบียนที่ข้อมูลขาดหายไป จะถูกแทนด้วยระเบียนใหม่หลาย ๆ ระเบียน โดยค่าข้อมูลต่าง ๆ ของระเบียนใหม่ จะเหมือนระเบียนเดิม ยกเว้นข้อมูลที่หายไป จะถูกแทนด้วยค่าหนึ่ง ในกลุ่มค่าที่เป็นไป



รูปที่ 3.34 ข้อมูลภาพเอ็กซเรย์เต้านม ลักษณะสำคัญมิติที่สอง อายุ สำหรับกรณีก้อนเนื้อเป็นเนื้อดี (ภาพซ้าย) และเนื้อร้าย (ภาพขวา).



รูปที่ 3.35 ข้อมูลภาพเอ็กซเรย์เต้านม ลักษณะสำคัญมิติที่สามถึงห้า รูปทรงมวลก้อนเนื้อ ลักษณะมวลก้อนเนื้อ และความหนาแน่นของมวลเนื้อ ซึ่งลักษณะสำคัญเหล่านี้เป็นค่าแทนชื่อ. ภาพซ้าย แสดงค่ากรณีก้อนเนื้อเป็นเนื้อดี. ภาพขวา แสดงค่ากรณีก้อนเนื้อเป็นเนื้อร้าย.



รูปที่ 3.36 ผลการทำนายข้อมูลภาพเอ็กซ์เรย์ต้านม ของแบบจำลองฝึกด้วยข้อมูลโดยไม่สนใจความหมาย **naive** (กล่องซ้าย) เปรียบเทียบกับผลของแบบจำลองที่ฝึกด้วยข้อมูลที่เข้ารหัสตามความหมาย **coding** (กล่องขวา). แกนนอนแสดงค่าความแม่นยำ.

ได. จำนวนระเบียนใหม่นี้จะเท่ากับจำนวนค่าที่เป็นไปได้ทั้งหมดของเขตข้อมูลที่ข้อมูลขาดหายไป. ดูกรีซีมาลาบุสเสและสู[81]เพิ่มเติม สำหรับวิธีต่าง ๆ ในการจัดการข้อมูลที่ขาดหายไป. นอกจากนั้น การเรียนรู้แบบกึ่งมือช่วยสอน[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 การทดลองตัวอย่างเลือกแทนค่าอายุเป็นช่วง ๆ ช่วงละหกปี แทนการแทนทุกค่าที่เป็นไปได้ ซึ่งหากทำทุกค่า ในกรณีนี้ อาจเพิ่มจำนวนระเบียนขึ้นมหาศาลโดยไม่จำเป็น.

รายการ 3.16: โปรแกรมช่วยวิธีแทนค่าขาดหายด้วยทุกค่าที่เป็นไปได้.

```
1 def fix_row(row, fix_vals):
2     rows = [row]
3     i = 0
4     while i < len(rows):
```

```

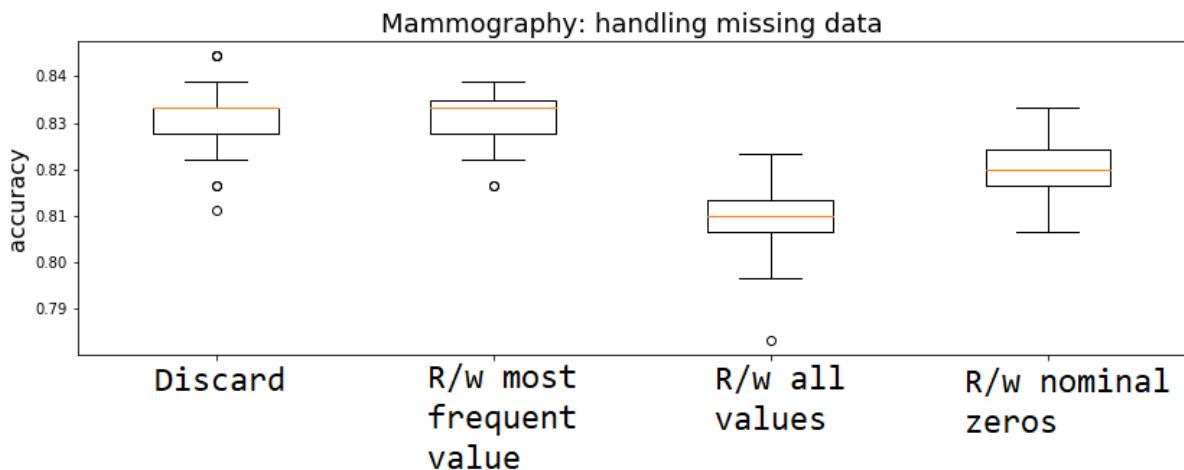
5      flag_next = True
6      for j, c in enumerate(rows[i]):
7          if c is None:
8              for v in fix_vals[j]:
9                  newrow = rows[i].copy()
10                 newrow[j] = v
11                 rows.append(newrow)
12             del rows[i]
13             flag_next = False
14             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 ตัวอย่างผลการคำนายนของชุดข้อมูลภาพอีกชุดเรียลเต้่านม เมื่อใช้วิธีจัดการกับข้อมูลขาดหายแบบต่าง ๆ.

วิธี	ค่าความแม่นยำ	
	ค่าเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน
ตัดระเบียนไม่สมบูรณ์ออก	0.8314	0.0073
แทนด้วยค่าเฉลี่ย หรือค่าพับบ่อຍที่สุด	0.8308	0.0063
แทนด้วยทุกค่าที่เป็นไปได้	0.8093	0.0070
แทนด้วยรหัสศูนย์	0.8203	0.0061

รูปที่ 3.37 และตารางที่ 3.6 แสดงตัวอย่างผลเปรียบเทียบ วิธีจัดการข้อมูลขาดหายแบบต่าง ๆ. นอกจากวิธีทั่วไป ดังอภิรายข้างต้น ในรูป ยังเสนอผลจากวิธีแทนข้อมูลค่าแทนซึ่งที่ขาดหาย ด้วยรหัสศูนย์. เนื่องจากข้อมูลภาพอีกชุดเรียลเต้่านม มีเขตข้อมูลหลายเขต ที่เป็นชนิดค่าแทนซึ่ง ซึ่งในตัวอย่าง ของแบบฝึกหัด 3.15 ใช้การเข้ารหัสหนึ่งร้อน (ดูโปรแกรม **coding** และ **onehot_cbook**). รหัสหนึ่งร้อน จะใช้ค่าทวีภาพหลายค่า แทนซึ่ง โดยมีแค่หนึ่งค่าที่ตำแหน่งสำหรับซึ่งนั้น ๆ ที่จะมีค่าเป็นหนึ่ง เพื่อรับซึ่งอักษรนั้น และค่าอื่น ๆ ในรหัสจะเป็นศูนย์ เช่น อักษรหนึ่ง แทนด้วย $[1, 0, 0, 0]^T$ และอักษรสอห แทนด้วย $[0, 1, 0, 0]^T$ สำหรับกรณีที่มีสีอักษร. ข้อมูลที่ขาดหาย อาจสามารถแทนเป็นรหัสศูนย์ โดยไม่มีหนึ่งที่ตำแหน่งใดในรหัสเลย เช่น $[0, 0, 0, 0]^T$. นี่คือ แนวคิดของวิธีแทนข้อมูลชนิดค่าแทนซึ่งที่ขาดหายด้วยรหัสศูนย์ (ระบุด้วย R/w nominal zeros ในภาพ).



รูปที่ 3.37 แผนภูมิกล่อง แสดงตัวอย่างผลเปรียบเทียบ วิธีการจัดการกับข้อมูลขาดหายแบบต่าง ๆ ผลได้จากการทดลอง จำนวน 40 ครั้ง. แกนตั้งแทนค่าความแม่นยำ. กล่องชี้ㄚยสุด Discard แสดงผลที่ดี เมื่อใช้วิธีตัดระเบียนที่ข้อมูลไม่สมบูรณ์ ทิ้ง. กล่องที่สองจากชี้ㄚย R/w most frequent value และผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ด้วยค่าเฉลี่ย หรือ ค่าที่พบบ่อยที่สุด. กล่องที่สาม R/w all values และผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ด้วยทุกค่าที่เป็นไปได้. กล่องที่สี่ R/w nominal zeros และผลเมื่อใช้วิธีแทนข้อมูลขาดหาย ที่เป็นค่าแทนซึ่ง ด้วยรหัสศูนย์.

แบบฝึกหัด 3.16

ข้อมูลเอมนิสต์ [116] เป็นชุดข้อมูลขนาดใหญ่ของภาพตัวเลขลายมือเขียน พร้อมเวลา. ชุดข้อมูลประกอบด้วย 60000 ตัวอย่าง สำหรับข้อมูลฝึก และ 10000 ตัวอย่าง สำหรับข้อมูลทดสอบ. แต่ละ ตัวอย่าง ตัวเลขในภาพถูกปรับขนาด และปรับจุดศูนย์กลาง และอยู่ในภาพสเกลเทา (gray scale) ขนาด 28×28 พิกเซล. คำอธิบาย และข้อมูลเอมนิสต์ สามารถหาได้ที่ <http://yann.lecun.com/exdb/mnist/>

จงทำแบบจำลองโครงข่ายประสาทเทียม เพื่อรู้จำภาพตัวเลขลายมือเขียน. นั่นคือ สร้างแบบจำลองที่ tally ฉลากตัวเลข จากราฟตัวเลขลายมือเขียน เลือกแบบจำลอง ฝึก ทดสอบ วัดผล รายงานผล อกีประย และสรุป.

ข้อมูลเอมนิสต์อยู่ในไฟล์ชนิดไบนาเรี่ย (binary file) และเก็บด้วยรูปแบบโอนเดียนใหญ่ (big endian). ไฟล์ฉลาก เช่น **train-labels.idx1-ubyte** มีส่วนหัว 8 ไบต์ ซึ่งเป็นตัวเลขขนาดสามสิบสองบิตสองตัว และจึงตามด้วยข้อมูลขนาด 60000 ไบต์ แต่ละไบต์เป็นตัวเลขแทนฉลากของข้อมูล แต่ละระเบียน. ไฟล์ข้อมูลภาพ เช่น **train-images.idx3-ubyte** มีส่วนหัว 16 ไบต์ ซึ่ง เป็นตัวเลขขนาดสามสิบสองบิตสี่ตัว และจึงตามด้วยข้อมูลที่แต่ละไบต์เป็นตัวเลขแทนค่าความเข้มของ พิกเซล (0 ถึง 255) เรียงกันต่อ กันไป 784 ไบต์ต่อภาพ จำนวน 60000 ภาพ รวมเป็นข้อมูลภาพ ทั้งหมด 47040000 ไบต์.

ตัวอย่างคำสั่งข้างล่างนำเข้าข้อมูลฉลากของชุดฝึก

```
import struct
```

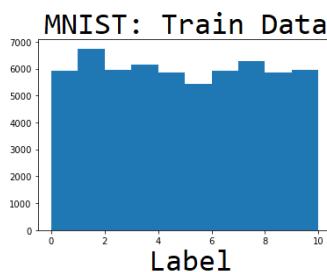
```

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))

```

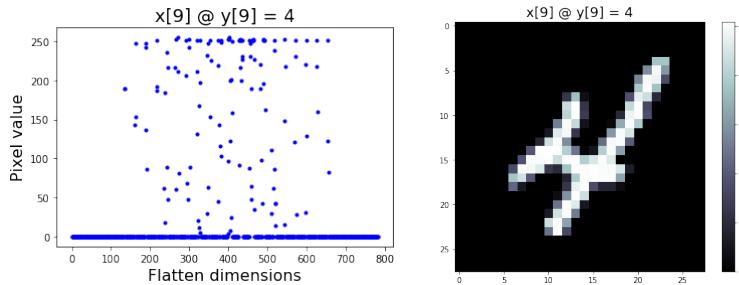
คำสั่งข้างต้นใช้คำสั่ง `struct.unpack_from(">I", bi)` เพื่ออ่านตัวเลข (32 บิต) ออกมา สำหรับตรวจสอบความถูกต้อง และใช้คำสั่ง `struct.unpack_from(">B", bi)` เพื่ออ่านข้อมูลตัวเลข (8 บิต) ออกมาเก็บรวมใน `trainy`. รูปที่ 3.38 แสดงการแจกแจงของข้อมูลเอมนิสต์.



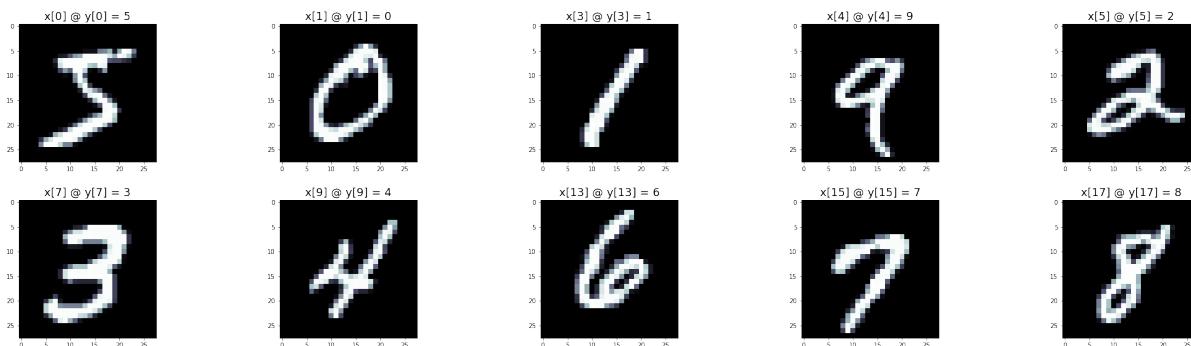
รูปที่ 3.38 การแจกแจงของข้อมูลเอมนิสต์ชุดฝึก. จำนวนจุดข้อมูลของแต่ละฉลากแสดงในแกนต์. ฉลาก (0 ถึง 9) แสดงในแกนนอน.

ในลักษณะเดียวกัน ข้อมูลภาพ เช่น `train-images.idx3-ubyte` ก็สามารถนำเข้าเป็นตัวแปร `trainx` (นามเพ้อาร์เรย์ สัดส่วน $(784, 60000)$). ข้อมูลอินพุตแต่ละระเบียน จะมี 784 ค่าลักษณะ. แต่ละค่าลักษณะ เป็นเลขจำนวนเต็มตั้งแต่ 0 ถึง 255. ตัวอย่างของอินพุตแต่ละระเบียน แสดงในรูปที่ 3.39 ภาพช้าย. เมื่อนำค่าลักษณะ จะจัดเรียงใหม่เป็นสองลำดับชั้นมิติ และนำไป

วัดลงในภาพสองมิติ จะได้ดังแสดงในภาพขวา ซึ่งจัดด้วยคำสั่ง `plt.imshow(trainx[:, 9].reshape((28, 28)), cmap=plt.cm.bone)` ซึ่งเลือกข้อมูลภาพลำดับที่เก้ามาแสดง. ตัวอย่างภาพตัวเลขต่าง ๆ ของข้อมูลเอนนิสต์ แสดงดังรูปที่ 3.40.



รูปที่ 3.39 อินพุตของเอนนิสต์. ภาพซ้าย แสดงโดยไม่มีโครงสร้างมิติ. ค่าของอินพุตแสดงเรียงไปทั้ง 784 ค่า. ด้านขวาของค่าแสดงตามแกนนอน และแกนตั้งแสดงค่าความเข้มของพิกเซล. ภาพขวา แสดงโดยจัดโครงสร้างมิติเป็นสองลำดับชั้น (อินพุต 784 ค่าถูกจัดเรียงเป็น 28×28) และค่าพิกเซลแทนด้วยระดับสีเทาต่าง ๆ ดังแสดงในแบบแผนที่ลีทางขวาของภาพ. บนแต่ละภาพ แสดงลำดับของภาพและฉลากเฉลย. ตัวอย่างนี้ ได้จากข้อมูลลำดับที่เก้า ซึ่งฉลากเฉลยระบุว่าเป็นเลขสี่.



รูปที่ 3.40 ตัวอย่างภาพตัวเลขข้อมูลเอนนิสต์. บนแต่ละภาพ แสดงตัวเลข และฉลากเฉลย.

ภาระกิจการรู้จำตัวเลขลายมือ เป็นงานการจำแนกประเภท ที่เอาร์พุตมีค่าที่เป็นไปได้ 10 ฉลาก. คำสั่งข้างล่าง แสดงตัวอย่างการสร้างโครงข่ายประสาทเทียมเป็นแบบจำลองสำหรับงานรู้จำตัวเลขลายมือ โดยใช้ข้อมูลเอนนิสต์ในการฝึก.

```
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** เป็นข้อมูลอบรมนิสต์สำหรับฝึก ที่นำเข้ามา. คำสั่ง **trainxn = trainx/255** ปรับขนาดของอินพุตให้อยู่ในช่วงศูนย์ถึงหนึ่ง. ส่วนคำสั่งด้านมา ปรับเอาต์พุตให้อยู่ในรหัสหนึ่งร้อน (โปรแกรม **coding** และ **onehot_cbook** ดูแบบฝึกหัด 3.15). ตัวอย่างนี้ใช้โครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยชั้น โดยโครงข่ายรับอินพุตขนาด 784 มิติ และให้อเอต์พุตออกขนาด 10 มิติ. ค่าน้ำหนักเริ่มต้น กำหนดด้วยวิธีเหงี่ยนวิดโดร์ (รายการ 3.12). พังก์ชันซอฟต์แมกซ์ และพังก์ชันสูญเสียครอสแอลโตรปี ถูกเลือกใช้สำหรับการกิจกรรมจำแนกลุ่ม (รายการ 3.17 และ 3.18). ตัวอย่างนี้ฝึก 300 สมัย โดยใช้อัตราเรียนรู้เป็น 1.

สังเกต โปรแกรมซอฟต์แมกซ์ เขียนโดยอาศัยคุณสมบัติคณิตศาสตร์

$$\frac{e^{a_k}}{\sum_i e^{a_i}} = \frac{e^{a_k - a_{\max}}}{\sum_i e^{a_i - a_{\max}}}$$

เมื่อ a_{\max} คือค่าส่วนประกอบของเวกเตอร์ที่มีค่ามากที่สุด. (ทดลองรันพังก์ชันซอฟต์แมกซ์ โดยใช้ค่า **va** ต่าง ๆ. ทดลองค่าใหญ่ ๆ ด้วย เช่น **va = np.array([[800],[500],[100]])** สังเกตผล และเปรียบเทียบกับผลจากโปรแกรมที่แสดงในแบบฝึกหัด 2.32 และอภิราย.) พังก์ชันสูญเสียครอสแอลโตรปี (รายการ 3.18) ซึ่งคือ $-\log \hat{y}_k$ สำหรับ ทุก ๆ ค่า k ที่ทำให้ $y_k = 1$ ก็เขียนด้วย การคำนวน

$$-\log \left(\sum_k y_k \cdot \hat{y}_k \right)$$

เมื่อ \hat{y}_k คือเอาต์พุตจากแบบจำลองที่ผ่านซอฟต์แมกซ์ออกแบบ สำหรับกลุ่มที่ k^{th} และเอาต์พุตเฉลย y_k เป็นส่วนประกอบของลากในรหัสหนึ่งร้อน $\mathbf{y} = [y_1, \dots, y_K]$ เมื่อ K เป็นจำนวนกลุ่ม. นั่นคือ $y_k \in \{0, 1\}$ และ $\sum_{k=1}^K y_k = 1$. (อภิราย การเขียนโปรแกรมพังก์ชันสูญเสียครอสแอลโตรปี 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 $1 \times N$. ดังนั้นการคำนวนค่าสูญเสียไม่สามารถเขียนในรูปเวกเตอร์ได้. โปรแกรมในรายการ 3.18 จึงเขียนด้วย **-np.log(np.sum(y*yhat, axis=0))**.

รายการ 3.17: พังก์ชันซอฟต์แมกซ์

```
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: พัฒนาสูญเสียครอสแอนโตรปี

```

1 def cross_entropy(yhat, y):
2     assert yhat.shape == y.shape
3     eps = 1e-323
4     return -np.log(np.sum(y*yhat,
5                           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 ที่อยู่ในรูปประมาณรหัสหนึ่งร้อน ส่วน **Yc** คือฉลากที่ทาย โดย เลือกฉลากที่มีส่วนประกอบในรหัสหนึ่งร้อน มีค่าสูงสุด เป็นฉลากที่ทาย.

ตารางที่ 3.7 แสดง เมทริกซ์สับสน ของผลทดสอบตัวอย่างที่ได้. ตัวเลขตามแนวทะแยงมุม คือจำนวนที่ทายถูก ในแต่ละประเภท. จากเมทริกซ์สับสนในตัวอย่าง ช่วยให้การวิเคราะห์ความผิดพลาด ทำได้สะเดກขึ้น เช่น จากเมทริกซ์ ภาพตัวเลขที่ทายผิดมากที่สุด คือภาพเลขเก้า ที่ถูกทายเป็นเลขสี่สิบ 81 ครั้ง และในทางกลับกัน ก็มีผิดไป 48 ครั้ง. รูปที่ 3.41 แสดงตัวอย่างรูปที่สับสนระหว่างภาพเลขสี่ และภาพเลขเก้า.

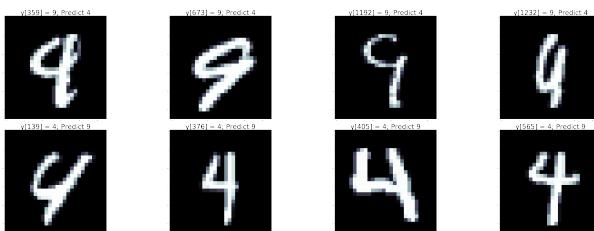
แบบฝึกหัด 3.17

การทำนายการจับตัวกันของโปรตีนและโมเลกุลขนาดเล็ก (*protein-ligand binding prediction*) เป็นภารกิจที่สำคัญในกระบวนการค้นหายา โดยเฉพาะการออกแบบยา (ดูเกร็ดความรู้การค้นหายา). แบบฝึกหัดนี้ ได้แรงบรรดาลใจจากการศึกษาของชานเชชและคณะ[175]. คณะของชานเชช ใช้ข้อมูลจากฐานข้อมูลดียูดี (*DUD: A Directory of Useful Decoys* <http://dud.docking.org/>) ที่รวบรวมข้อมูลของลิแกนต์ และตัวหลอก ของโปรตีนต่าง ๆ ซึ่งลิแกนต์ (*ligand*) คือโมเลกุลที่จับตัวกับโปรตีนที่สนใจ ส่วนตัวหลอก (*decoy*) คือโมเลกุลที่ไม่จับตัวกับโปรตีนที่สนใจ.

ตารางที่ 3.7 เมทริกซ์สับสน แสดงผลการทำนาย โดยแยกตามประเภท ทั้งประเภททิไทย (แสดงตามแท่ง) และประเภทเจริญ ที่ระบุด้วยฉลากเฉลย (แสดงตามสมดุล).

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. ภาพในแถวบน เลขเก้าที่ถูกไทยเป็นเลขสี่ และภาพในแถวล่าง เลขสี่ที่ถูกไทยเป็นเลขเก้า.

จะเลือกโปรตีนเป้าหมายจากฐานข้อมูลดียดี และสร้างแบบจำลองการทำนายการจับตัวกันของโปรตีน เป้าหมาย กับโมเลกุลขนาดเล็ก โดยจะสร้างเป็นแบบจำลองเฉพาะสำหรับโปรตีนนั้น และใช้คุณลักษณะ ต่าง ๆ ของโมเลกุลขนาดเล็ก เพื่อทำนายว่าโมเลกุลจะสามารถจับกับโปรตีนเป้าหมายได้หรือไม่. ทดสอบ วิเคราะห์ ภูมิประยุผล และสรุป. หมายเหตุ การสร้างแบบจำลองทั่วไปที่สามารถทำนายการจับตัวระหว่างโมเลกุลกับโปรตีนได ๆ มีความท้าทายมาก และคู่ควรกับโครงการวิจัยระยะยาว (งานวิจัยของ คณะของชานเชช[175] เอง ก็เป็นการสร้างแบบจำลองเฉพาะกับแต่ละโปรตีน). ดังนั้น เพื่อให้เหมาะสม กับเนื้อหา ระดับความยาก และเวลา แบบฝึกหัดนี้จำกัดปัญหาเป็นการสร้างแบบจำลองเฉพาะโปรตีน ก่อน.

การกิจนี้ มีผลทำนายเป็นสองสถานะ คือ จับตัวกัน หรือไม่จับตัวกัน. ดังนั้น ภาระกิจนี้ควรวางแผนเป็นงานการจำแนกค่าทิวภาค. อินพุตของแบบจำลองเป็นคุณลักษณะของโมเลกุล ซึ่งคณะของ ชานเชช[175] ใช้ไลบรารีเคมอย [27] ช่วยในการจัดเตรียมคุณลักษณะของโมเลกุล จาก ข้อมูลรูปแบบโมลส่อง (Tripos's mol-2 format) ที่ได้จากฐานข้อมูลดียดี. แต่ตัวอย่างที่จะแสดงต่อไป

นี้ เลือกที่จะจัดเตรียมคุณลักษณะต่าง ๆ ของโมเลกุลเอง โดยเลือกทำเฉพาะคุณลักษณะง่าย ๆ ที่ไม่ซับซ้อนจนเกินไป. ผู้อ่านอาจทดลองไลบรารีเคมีไฟ หรือไลบรารีที่เกี่ยวข้องอื่น ๆ เช่น อาร์ดีคิต (RDKit <http://www.rdkit.org>) หากสนใจ.

ฐานข้อมูลดียูดี มีข้อมูลของโปรตีนสำคัญ ๆ อยู่หลายตัว (<dud.docking.org/r2/>) เช่น แอนจิโอเท็นซินคอนเวิร์ตติง เอ็นไซม์ (Angiotensin-converting enzyme), อะเซติลโคลีน เอสเตอเรส (Acetylcholine esterase) รวมถึง ไทโรซีนคิเนส查ร์ค (Tyrosine kinase SRC) และ เอชเอมจี โคลอเรตตักเตส (Hydroxymethylglutaryl-CoA reductase). ตัวอย่างต่อไปนี้ เลือกเป้าหมายเป็นโปรตีนไทรอซีนคิเนส查ร์ค ซึ่งเป็นเอนไซม์ที่เกี่ยวข้องกับโรคมะเร็งเนื้อยื่นเกี่ยวพัน (sarcoma). รายการ 3.19 แสดงโปรแกรมสำหรับนำเข้าข้อมูล โดยโปรแกรมรับชื่อไฟล์ข้อมูล (พร้อมเส้นทาง) ด้วยอาร์กิวเมนต์ `cpath` และรีเทิร์นลิสต์ของดิกชันนารีอุปกรณ์. ลิแกนต์และตัวหลอก ถูกโหลดได้ด้วยคำสั่ง เช่น

```
ligands = load_compounds('databases/dud_decoys2006/src_decoys.mol2')
decoys = load_compounds('databases/dud_ligands2006/src_ligands.mol2')
```

สำหรับโปรตีนไทรอซีนคิเนส查ร์คนี้ ข้อมูลลิแกนต์มีอยู่ 159 โมเลกุล และข้อมูลตัวหลอกมีอยู่ 6319 โมเลกุล. โมเลกุลแต่ละตัว จะมีข้อมูลอยู่สามชนิดคือ ข้อมูลที่ว่าเป็นโมเลกุล ข้อมูลของอะตอมต่าง ๆ ในโมเลกุล และข้อมูลของพันธะที่เข้มมะตอมต่าง ๆ ซึ่งสามารถเข้าถึง ได้ด้วยคำสั่งเช่น `ligands[0]['MOLECULE']` หรือ `ligands[0]['ATOM']` หรือ `ligands[0]['BOND']` สำหรับ ข้อมูลต่าง ๆ ของลิแกนต์โมเลกุลแรก (ลำดับที่ศูนย์). รูปแบบไฟล์ข้อมูลไม่สองและคำอธิบาย สามารถศึกษาเพิ่มเติมได้จากเอกสารประกอบ Tripos Mol2 SYBYL 7.1 (Mid-2005) ที่สามารถค้นหาได้จากอินเตอร์เน็ต.

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'
7         field_count = 0
8         compounds = []
9         hasData = False
10        d = {}
11        for line in ctxt.split('\n'):
12
13            if line[:9] == '@<TRIPOS>':
14                field_count = 0 # start new field
15                if hasData:
```

```
16             d[field_name] = v
17             field_name = line[9:]
18
19         if field_name == 'MOLECULE' and hasData:
20             compounds.append(d)
21             d = {}
22             hasData = True
23     else:
24         if len(line) > 0:
25             if field_name == 'MOLECULE':
26                 if field_count == 0:
27                     v = {}
28                     v['name'] = line
29                 elif field_count == 1:
30                     molecule_info = line.split()
31                     v['num_atom'] = molecule_info[0]
32                     if len(molecule_info) > 1:
33                         v['num_bonds'] = molecule_info[1]
34                 elif field_count == 2:
35                     v['mol_type'] = line
36                 elif field_count == 3:
37                     v['charge_type'] = line
38             elif field_name == 'ATOM':
39                 if field_count == 0:
40                     v = []
41                 atom_info = line.split()
42                 if len(atom_info) > 5:
43                     row = [int(atom_info[0]), # id
44                             atom_info[1], # name
45                             float(atom_info[2]), # x
46                             float(atom_info[3]), # y
47                             float(atom_info[4]), # z
48                             *atom_info[5:]] # type & 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 จำนวนพันธะเดียว จำนวนพันธะคู่ จำนวนพันธะสาม จำนวนพันธะเอไมด์ และ จำนวนพันธะอะโรมาติก ดังโปรแกรม **compound_feat1** ในรายการ 3.20. ข้อมูลถูกแกนต์และตัวหลอก (ตัวแปร **xlig** และ **xdec** ตามลำดับ) เตรียมได้ดังตัวอย่าง คำสั่ง

```

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 แล้วรวมข้อมูลิกเกนต์และตัวหลอกเข้าด้วยกัน

```

_, 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:]]

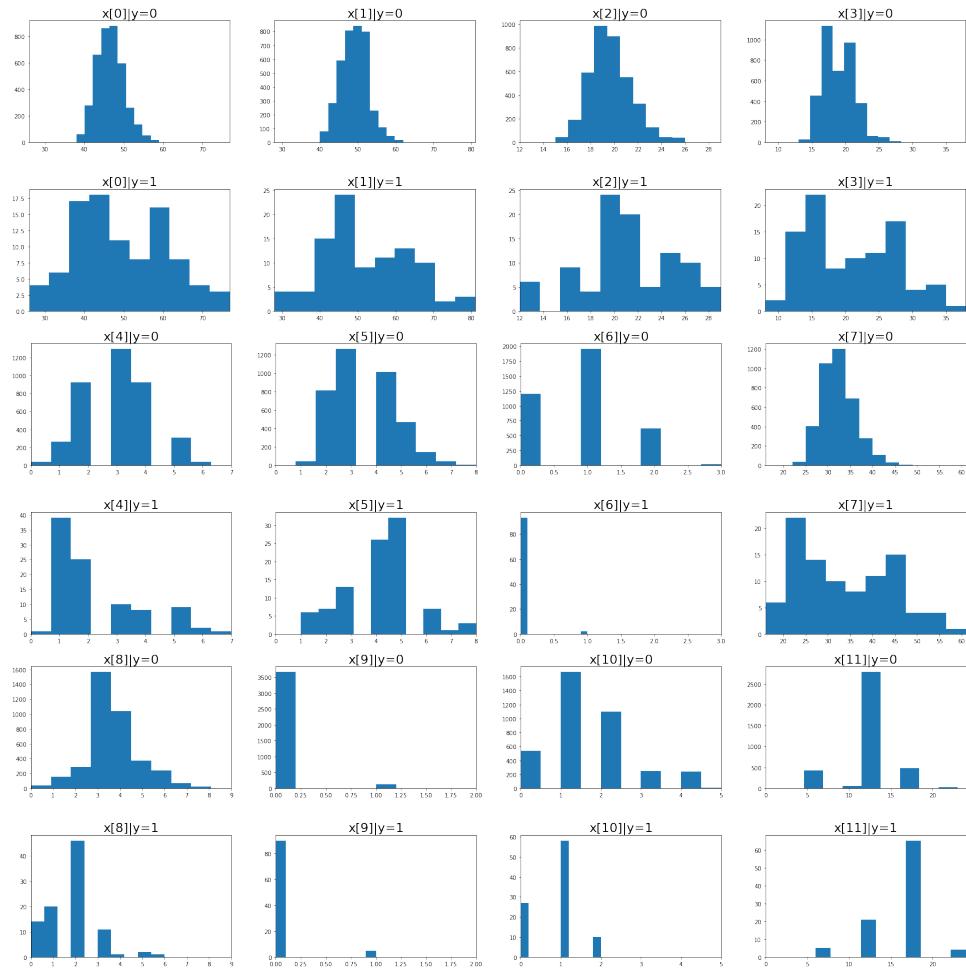
# Combine ligands and decoys
_, N1 = trainx_lig.shape
_, N0 = trainx_dec.shape
trainx = np.hstack((trainx_lig, trainx_dec))
trainy = np.hstack((np.ones((1, N1)), np.zeros((1, N0)))))

_, 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 แสดงการแยกจัดของข้อมูลฝึก. ค่าอินพุตมีช่วงค่อนข้างกว้าง คำสั่งข้างล่างแสดงตัวอย่างการทำรอมอ่อล์อินพุต เตรียมแบบจำลองโครงข่ายประสาทเทียมสองชั้นขนาด 8 หน่วยซ่อนและฝึก 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)
```



รูปที่ 3.42 การแจกแจงของลักษณะสำคัญเชิงเลข ทั้ง 12 ลักษณะสำคัญ ($x[0]$ ถึง $x[11]$) ของข้อมูลไม่เลกูลสารประกอบทั้งลิแกนต์ ($y=1$) และตัวหลอก ($y=0$).

ตัวอย่างคำสั่งข้างล่างทำการทดสอบผลการทำนาย

```
testxn, _ = normalize2(testx, normpars)
Yp = mlp(trained_net, testxn)
Yc = cutoff(Yp)
accuracy = np.mean(Yc == testy)
```

ผลลัพธ์ของตัวอย่าง แสดงค่าความแม่นยำอกรมาที่ 97.5%. หมายเหตุ ผลลัพธ์ที่ทดลองแต่ละครั้งอาจแสดงค่าที่ต่างกันไปเนื่องจากผลของการสุ่ม ซึ่งอยู่ในกระบวนการแบ่งข้อมูล และการกำหนดค่าน้ำหนักเริ่มต้น. ดังนั้น หากต้องการศึกษาปัจจัยที่เกี่ยวข้องอย่างสมบูรณ์ ควรทำการทดลองซ้ำ โดยให้จำนวนทำซ้ำมากพอ⁴ เพื่อยืนยันผลว่าความต่างของผลลัพธ์เป็นผลมาจากการปัจจัยที่ต่างกันจริง ๆ ไม่ใช่มา

⁴ ประเดิมเมื่อจำนวนซ้ำ มีหลักการอยู่ว่า จำนวนซ้ำต้องมากพอ ที่หลักการทางสถิติ เช่น การทดสอบนัยสำคัญ (significance test) สามารถยืนยันความต่างได้ หากความต่างมีจริง. แต่หากการทดสอบนัยสำคัญ ไม่สามารถยืนยันความต่างได้ อาจหมายความได้ว่า (1) ผลที่เปรียบเทียบกันนั้นไม่ได้ต่างกันจริง ๆ ความต่างที่สังเกตเป็นเพียงความแปรปรวนของข้อมูล หรือ (2) ผลที่เปรียบเทียบ

จากความแปรปรวนของข้อมูลหรือความแปรปรวนจากการกระบวนการสุ่ม. แต่เพื่อความกระชับ ตัวอย่างนี้ไม่ได้ทำซ้ำ.

ค่าความแม่นยำที่ได้ แม้จะดูดีมาก แต่เมื่อพิจารณาทริกซ์ความสัมสูนที่ได้ (ดังแสดงข้างล่าง) แล้วจะพบว่าแบบจำลองนี้ล้มเหลว เพราะมันไม่ระบุสารประกอบใดที่อาจจับตัวกับเป้าหมายเลย.

ผลจริง

		1	0
ผลท่านาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	0	จำนวนลบเท็จ
	0	64	2528

สังเกตว่า เมื่อสัดส่วนจำนวนข้อมูลต่างกันมาก แบบจำลองเพียงทำนายว่า ไม่จับตัวกับเป้าหมาย กับทุก ๆ สารประกอบ ก็สามารถจะได้ค่าความแม่นยำที่สูงมากได้. แต่เมื่อพิจารณาค่าความเที่ยงตรง และค่าการระลึกกลับ ซึ่งเป็น 0/0 และ 0 ตามลำดับ จะพบว่า ความเที่ยงตรง และการระลึกกลับ สะท้อนความล้มเหลวของแบบจำลองทำนายได้ชัดเจนมาก.

ก่อนจะอภิรายเรื่องวิธีจัดการกับปัญหาลักษณะจำนวนข้อมูลไม่สมดุล พิจารณาค่าเออร์พุตที่ได้ จากแบบจำลอง สำหรับกรณีของลิแกนต์และตัวหลอก. รูปที่ 3.43 แสดงให้เห็นว่า ค่าเออร์พุตที่มาก ที่สุด มีค่าอยู่แค่ประมาณ 0.2. ค่าเออร์พุตที่ได้จะถูกตัดสินใจสุดท้ายด้วย โปรแกรม **cutoff** (ดูแบบฝึกหัด 3.15) ที่ค่าดีฟอลต์คือตัดทายหนึ่งที่ 0.5. ดังนั้น ค่าได ๆ ที่น้อยกว่า 0.5 จะถูกตัดสินใจ แต่หากค่าอยู่ในช่วง 0.2-0.5 ที่เหลือจะถูกตัดสินใจโดย 0.5 ที่ตัดสินใจนี้ ทำให้ตัวหัวใจของแบบจำลอง คือค่าเออร์พุต ที่ได้สำหรับลิแกนต์ เปรียบเทียบกับตัวหลอก จะพบว่า แม้ทั้งคู่จะมีค่าต่ำ แต่ค่าเออร์พุตสำหรับลิแกนต์ส่วนใหญ่ ก็มากกว่าค่าเออร์พุตสำหรับตัวหลอก ค่อนข้างชัดเจน. ดังนั้น ความล้มเหลวของการทำนายนี้ อาจบรรเทาได้เพียงแค่การปรับระดับค่าขีดแบ่ง (*threshold*) ลง.

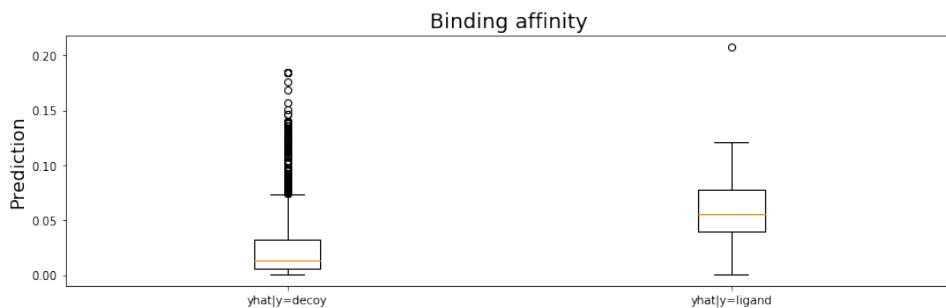
พฤติกรรมการทำนายของแบบจำลองจำแนกค่าทิวภาค สามารถถูกปรับแต่งได้จากการปรับระดับค่าขีดแบ่ง. รูปที่ 3.44 แสดงค่าคงแคนเนอฟ เมื่อใช้ระดับค่าขีดแบ่งต่าง ๆ. หมายเหตุ เพื่อลดความยุ่งยากจากการณ์ 0/0 ตัวหารของค่าความเที่ยงตรง และค่าคงแคนเนอฟ คำนวนด้วยคำสั่งดังต่อไปนี้

อาจต่างกันจริง ๆ เพียงแต่ด้วยจำนวนข้อมูลหรือจำนวนช้าที่มี ไม่สามารถยืนยันได. นั่นหมายความว่า หากเลือกจำนวนช้าแล้ว การทดสอบนัยสำคัญสามารถยืนยันความต่างได้ แปลว่าจำนวนช้าที่เลือกนั้นเพียงพอ. แต่หากเลือกจำนวนช้าแล้ว การทดสอบนัยสำคัญไม่สามารถยืนยันความต่างได้ อาจแปลว่า (1) จำนวนช้าที่เลือกนั้นไม่เพียงพอ ควรเพิ่มจำนวนช้า หรืออาจแปลว่า (2) ผลที่เปรียบเทียบไม่ได้ต่างกัน. ดังนั้น ในทางปฏิบัติ หากการทดสอบนัยสำคัญ ยังไม่สามารถยืนยันความต่างได้ ผู้ทดลองอาจเลือกเพิ่มจำนวนช้า หากผู้ทดลองเชื่อว่าเป็นกรณีแรก หรือผู้ทดลองอาจเลือกจากการทดลอง และสรุปว่าการทดสอบนัยสำคัญไม่สามารถยืนยันความต่างได้ ที่ความมั่นใจที่ระบุ เมื่อใช้จำนวนช้าที่เลือก. สังเกตว่า การทดสอบนัยสำคัญ สามารถยืนยันความต่างได้ แต่ไม่สามารถยืนยันความเหมือน (หรือความไม่ต่าง). นั่นคือ หากการทดสอบนัยสำคัญยืนยันว่าผลต่างกันจริง หมายถึงผลต่างกันจริง ๆ. แต่หากการทดสอบนัยสำคัญไม่สามารถยืนยันความต่าง อาจแปลว่าหลักฐานไม่พอ หรืออาจแปลว่าผลไม่ต่างกัน.

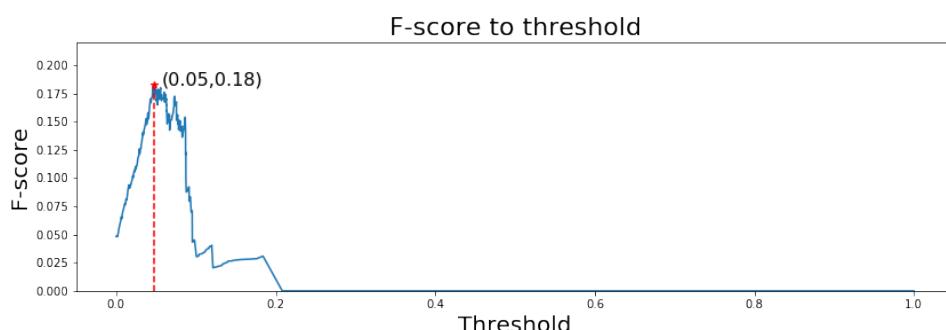
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP} + 1e-12}$$

$$\text{fScore} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall} + 1e-12)}$$

เมื่อ **TP**, **FP**, และ **Recall** เป็นจำนวนบวกจริง, จำนวนบวกเท็จ, และค่าการระลึกกลับ ตามลำดับ. ค่า **1e-12** เป็นค่าน้อย ๆ ที่เพิ่มเข้าไป ซึ่งจะเปลี่ยนกรณี 0/0 เป็น 0 และไม่รบกวนกรณีอื่นๆมาก.



รูปที่ 3.43 แผนภูมิกล่องแสดงตัวอย่างผลจากแบบจำลองที่ทำนายการจับตัวกับโปรตีน สำหรับลิแกนต์และตัวหลอก.



รูปที่ 3.44 ค่าคะแนนเอฟของการทำนายการจับตัว เมื่อใช้ระดับค่าขีดแบ่งต่าง ๆ.

จากรูปที่ 3.44 ค่าคะแนนเอฟจะสูงสุด เมื่อเลือกใช้ระดับค่าขีดแบ่งประมาณ 0.05 และเมื่อเลือก ระดับค่าขีดแบ่งที่ประมาณ 0.05 แล้วจะได้ผลทดสอบดังเมทริกซ์ความสัมสุน

		ผลจริง	
		1	0
ผลทำนาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	จำนวนลบเท็จ	จำนวนลบจริง

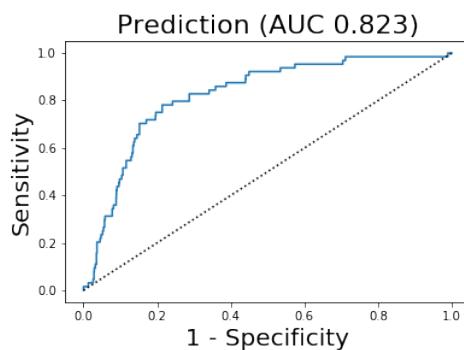
45 384
19 2144

และได้ค่าความเที่ยงตรง 0.105 ค่าการระลึกกลับ 0.703 และค่าคะแนนเอฟ 0.183. ผลลัพธ์ที่ได้ แม้จะยังแย่ แต่ก็ดีขึ้นกว่าเดิมมาก. นอกจากนี้ ระดับค่าขีดแบ่ง ก็สามารถเลือกปรับให้เหมาะสมกับ

ความชอบส่วนบุคคล หรือให้เหมาะสมกับสถานการณ์ได้ เช่น บางภาระกิจอาจเลือกบวกเท็จ (เช่น หากทรัพยากรเพียงพอ ได้ตัวหลอกเกินมา ดีกว่าขาดลิแกนต์ไป) ในขณะที่บางภาระกิจอาจเลือกลบเท็จ ดีกว่าบวกเท็จ (เช่น เมื่อทรัพยากรจำกัดมาก ๆ ตกลิแกนต์ไปบ้าง ดีกว่าได้ตัวหลอกมา และเปลืองค่าใช้จ่ายในขั้นตอนการพัฒนาอย่างต่อไปเป็นต้น). เนื่องจากการเลือกระดับค่าขีดแบ่ง มีผลต่อการทำนายมาก และยังอาจถูกปรับให้เหมาะสมกับความชอบส่วนบุคคลได้ การประเมินแบบจำลอง บางครั้ง จึงนิยมใช้กราฟอาร์โอซี (Receiver Operating Characteristic คำย่อ ROC) และพื้นที่ใต้เส้นโค้ง (Area Under Curve คำย่อ AUC).

กราฟอาร์โอซี หมายถึง กราฟแสดงผลการทำนาย โดยอาจเลือกใช้ดัชนีวัดได้หลายแบบ เช่น อาจใช้กราฟระหว่างค่าความเที่ยงตรงกับค่าการระลึกกลับ (precision-recall plot) หรืออาจใช้กราฟระหว่างค่าอัตราการตรวจจับได้กับอัตราลัญญาณหลอก (detection-rate-to-false-alarm-rate plot). อัตราการตรวจจับได้ อาจเรียกว่าค่าความไว (sensitivity หรือ true positive rate) $S_1 = \text{Recall} = TP / (TP + FN)$ เมื่อ TP และ FN คือจำนวนบวกจริง และจำนวนลบเท็จ ตามลำดับ. อัตราลัญญาณหลอก (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 คือจำนวนลบจริง และจำนวนบวกเท็จ ตามลำดับ.

รูปที่ 3.45 แสดงกราฟระหว่างค่าความไวกับอัตราลัญญาณหลอก จากผลตัวอย่าง. จุดต่าง ๆ บนเส้นกราฟค่านวนโดยการปรับระดับค่าขีดแบ่งจากน้อยที่สุดไปมากที่สุด และประเมินผลการทำนาย สำหรับแต่ละระดับค่าขีดแบ่ง. พื้นที่ใต้เส้นโค้ง ก็คือพื้นที่ใต้กราฟอาร์โอซีที่เลือกใช้. รูปที่ 3.45 แสดงค่าพื้นที่ใต้เส้นโค้ง กำกับไว้เหนือภาพ. ค่าพื้นที่ใต้เส้นโค้งที่ใกล้หนึ่ง แสดงถึงคุณภาพการทำนายที่ดีของแบบจำลอง.



รูปที่ 3.45 กราฟอาร์โอซีการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีน. ภาพแสดง กราฟระหว่างค่าความไว (Sensitivity) กับอัตราลัญญาณหลอก ($1 - \text{Specificity}$) ของตัวอย่างผลการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีน troponin C ในคนไข้的心臟. ค่าพื้นที่ใต้เส้นโค้ง (AUC) แสดงเหนือภาพ.

สำหรับการกิจกรรมจำแนกค่าทวิภาค หรือการจำแนกกลุ่ม เมื่อจำนวนจุดข้อมูลของแต่ละกลุ่มข้อมูลต่างกันมาก จะเกิดปัญหาลักษณะจำนวนข้อมูลไม่สมดุล (unbalanced data) ขึ้น. วิธีจัดการปัญหา

สัดส่วนจำนวนข้อมูลไม่สมดุลในชุดข้อมูลฝึก สามารถทำได้หลายวิธี^[34] เช่น วิธีการสุ่มเกิน (over sampling), วิธีการสุ่มขาด (under sampling), วิธีปรับฟังก์ชันจุดประสงค์. วิธีการสุ่มเกิน ใช้การสุ่มแบบคืนที่ (sampling with replacement) เพื่อเพิ่มจุดข้อมูลของกลุ่มน้อยขึ้นมาให้ใกล้เคียงกับกลุ่มใหญ่. วิธีการสุ่มขาด ใช้การสุ่มเลือกบางส่วนของข้อมูลจากกลุ่มใหญ่ เพื่อให้ข้อมูลของที่ใช้ฝึกของกลุ่มใหญ่มีจำนวนใกล้เคียงกับจำนวนของกลุ่มน้อย. วิธีปรับฟังก์ชันจุดประสงค์ ปรับการคำนวนค่าฟังก์ชันจุดประสงค์ โดยให้น้ำหนักความสำคัญกับการทำนายกลุ่มน้อยมากขึ้น (หรือลดน้ำหนักความสำคัญของการทำนายกลุ่มใหญ่ลง หรือทำทั้งสองทาง) เพื่อชดเชยกับจำนวนข้อมูลที่ต่างกัน.

ตัวอย่างที่จะแสดงต่อไปนี้ ใช้วิธีการสุ่มเกิน 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).

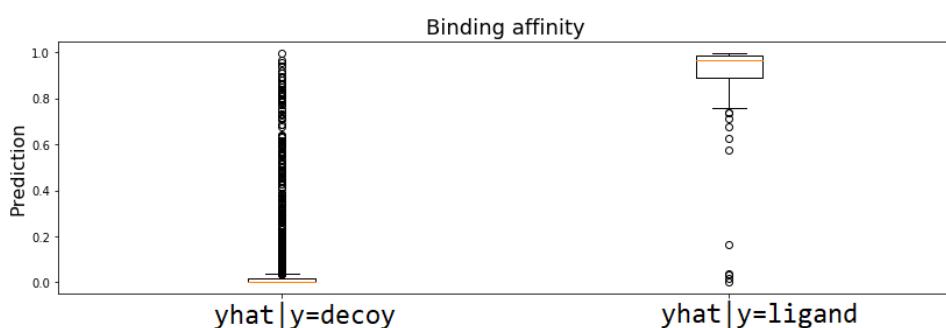
หลังจากการน้อมอไลซ์อินพุต ฝึกแบบจำลอง⁵ และทดสอบแล้ว ผลที่ได้แสดงดังเมทริกซ์ความสับสน

⁵ ตัวอย่างนี้ ฝึกโครงข่ายประสาทเทียมสองชั้น ขนาด 8 หน่วยช่อน 10000 สมัย ด้วยอัตราเรียนรู้ 0.1. การฝึกครั้งนี้ ใช้จำนวนสมัยฝึกมากกว่า จำนวนสมัยของการฝึกกับข้อมูลที่ไม่มีการจัดการข้อมูลไม่สมดุล เนื่องจาก การฝึกควรที่จะนัดการฝึกสมบูรณ์ หรือค่อนข้างสมบูรณ์ โดยพิจารณาจากความก้าวหน้าของการฝึก (**train_losses**). จากความก้าวหน้าของการฝึกที่ได้ กรณีการสุ่มเกิน ไม่สามารถฝึกแค่ 500 สมัยได้ (เพราะการฝึกดูยังห่างความสมบูรณ์อยู่มาก) แต่กรณีการไม่ทำอะไร สามารถฝึก 10000 สมัยได้. อย่างไร ก็ตาม ผู้เขียนเห็นว่า การทดลองดังผลที่นำเสนอในรูปข้างต้น เปิดโอกาสให้เห็นความเสี่ยงจากการพึ่งพาความแม่นยำเพียงอย่างเดียว รวมถึงความสำคัญของการตรวจสอบผลที่ได้ ซึ่งอาจจะเป็นประโยชน์มากกว่า. การทดลองโดยใช้จำนวนสมัยฝึกพอ ๆ กันสามารถทำได้ และผู้เขียนพบว่า ได้ผลลัพธ์ในทิศทางเดียวกัน เพียงแต่ผลต่างอาจไม่เด่นชัดเท่าที่นำเสนอในตัวอย่างนี้.

ผลจริง

		1	0
ผลท่านาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	จำนวนลบเท็จ	จำนวนลบจริง
		59	87
		5	2441

ค่าความเที่ยงตรง 0.404 ค่าการระลึกกลับ 0.922 และค่าคณเอนเอฟ 0.562 ซึ่งปรับปรุงขึ้นมาก. ผลประเมินนี้ได้จากการตัดสินผลทำนายด้วยระดับค่าชีดแบ่ง 0.5. ในลักษณะเดียวกัน รูปที่ 3.46 แสดงแผนภูมิกล่องของค่าเอาร์พุตจากแบบจำลอง สำหรับข้อมูลกลุ่มน้อย และกลุ่มใหญ่. สังเกตว่า เอาร์พุตจากแบบจำลอง มีช่วงค่าแยกกันชัดเจนมากระหว่างข้อมูลกลุ่มน้อย (ค่าใกล้หนึ่ง) และข้อมูล กลุ่มใหญ่ (ค่าใกล้ศูนย์) แม้จะมีค่าผิดปกติบ้าง. ในทางสถิติ ค่าผิดปกติ (outliers) หมายถึง ค่าของจุด ข้อมูลจำนวนน้อย ที่มีค่าต่างจากค่าของจุดข้อมูลอื่น ๆ ในกลุ่มอย่างมาก.



รูปที่ 3.46 ผลหลังแก้ข้อมูลไม่สมดุล. แผนภูมิกล่องของค่าเอาร์พุตจากแบบจำลอง สำหรับข้อมูลกลุ่มน้อย (**decoy**) และ ข้อมูลกลุ่มน้อย (**ligand**) เมื่อใช้วิธีการสุมเกิน เพื่อจัดการปัญหาจำนวนข้อมูลไม่สมดุล.

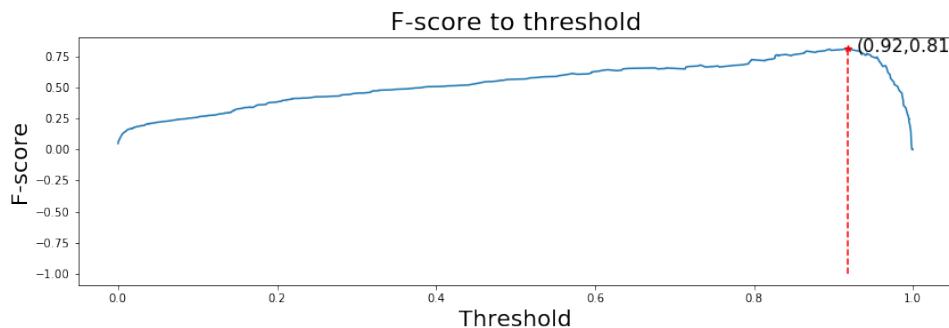
จากรูปที่ 3.46 การตัดสินผลทำนาย อาจสามารถปรับปรุงได้ง่าย ๆ ด้วยการเปลี่ยนระดับค่าชีด แบ่ง. พิจารณาที่ 3.47 ซึ่งแสดงค่าคณเอนเอฟ ที่ระดับค่าชีดแบ่งต่าง ๆ และเมื่อเปลี่ยนระดับค่าชีด แบ่งเป็นประมาณ 0.92 จะได้เมตริกซ์ความสัมสูน

ผลจริง

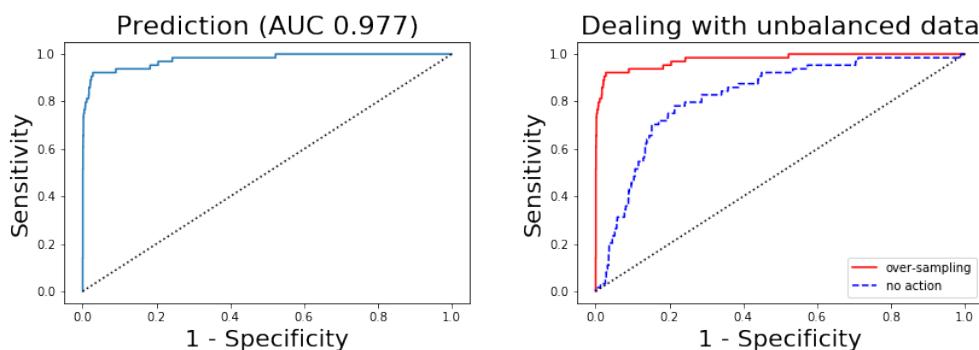
		1	0
ผลท่านาย	1	จำนวนบวกจริง	จำนวนบวกเท็จ
	0	จำนวนลบเท็จ	จำนวนลบจริง
		47	5
		17	2523

ค่าความเที่ยงตรง 0.904 ค่าการระลึกกลับ 0.734 และค่าคณเอนเอฟ 0.810 ซึ่งโดยทั่วไปแล้ว ค่า คณเอนเอฟขนาดนี้ ลือว่าแบบจำลองสามารถทำงานได้ดีพอสมควร. รูปที่ 3.48 แสดงแสดงกราฟ

ระหว่างค่าความไวกับอัตราลัญญาณหลอก พร้อมค่าพื้นที่ได้เส้นโค้ง เมื่อใช้วิธีสุมเกิน (ภาพซ้าย) และเปรียบเทียบกับการไม่ทำอะไรเลย (ภาพขวา). จากตัวอย่างข้างต้น วิธีสุมเกินสามารถช่วยปรับปรุงคุณภาพของการเตรียมแบบจำลองทำนาย ในกรณีจำนวนข้อมูลไม่สมดุลได้อย่างชัดเจน.



รูปที่ 3.47 ค่าคะแนนเอฟ ที่ระดับค่าขีดแบ่งต่าง ๆ เมื่อใช้วิธีการสุมเกิน เพื่อจัดการปัญหาจำนวนข้อมูลไม่สมดุล.



รูปที่ 3.48 ผลการทำนายการจับตัวของโมเลกุล หลังปรับปรุงด้วยวิธีสุมเกิน. ภาพซ้าย แสดงกราฟระหว่างค่าความไวกับอัตราลัญญาณหลอก ของตัวอย่างการทำนายการจับตัวของโมเลกุลขนาดเล็กกับโปรตีนไฮโรเซ็นคิโนซาร์ค หลังปรับปรุงข้อมูลไม่สมดุลด้วยวิธีสุมเกิน และภาพขวา แสดงกราฟเปรียบเทียบระหว่างการทำไรเรเลยกับปัญหาข้อมูลไม่สมดุล (no action) กับการใช้วิธีสุมเกิน (over-sampling).

เกร็ดความรู้การค้นหายา (เรียบเรียงจาก [29] และ [66] และ [217]) ยา โดยทั่วไปคือ โมเลกุลที่กระตุ้นหรือยับยั้งการทำงานของชีวโมเลกุล เช่น โปรตีน ซึ่งส่งผลทางการรักษาโรคกับผู้ป่วย. แนวทางในการค้นหายาแบบดั้งเดิม อาจจะเริ่มด้วยการหาส่วนผสมออกฤทธิ์ (*active ingredient*) จากตัวรับยาตั้งเดิม เช่น ยาเรเซอร์พีน. รีเซอร์พีน (Reserpine) เป็นยาสำหรับบำบัดอาการความดันสูง ที่สะกัดจากรากของต้นระย່อมน้อย (*Rauvolfia serpentina* หรือชื่อสามัญ Indian snakeroot) ที่อยู่ในตัวรับยาอายุรเวทของอินเดียแต่โบราณ. หรืออาจจะเริ่มด้วยการค้นหาสารประกอบต่าง ๆ ที่ส่งผลที่ต้องการ จากการทดลองกับสัตว์ที่ป่วยเป็นโรค หรือจากการทดลองในหลอดทดลองกับเซลล์ที่เป็นโรค. สารประกอบที่พบจากการค้นหาเบื้องต้น จะเรียกว่า สารประกอบหลัก. สารประกอบหลัก (*lead compounds*) คือ สารประกอบที่จากการทดลองแล้วพบว่าจะช่วยรักษาโรคได้ แต่โครงสร้างทางเคมีอาจจะยังไม่ดีเท่าไร. จากนั้น สารประกอบหลักต่าง ๆ ที่

ได้ จะถูกดัดแปลงทางเคมี เพื่อปรับปรุง การออกฤทธิ์ (potency) และสมรรถนะการเลือก (selectivity) รวมถึงปรับปรุง คุณสมบัติทางเภสัชจลนศาสตร์อื่น ๆ ให้เหมาะสมที่จะเป็นยา และสามารถดำเนินการทดสอบกับสัตว์ทดลอง และทดสอบทางคลินิกได้ต่อไป. แนวทางในการค้นหาแบบดั้งเดิมนี้ เริ่มจากการค้นหาสารประกอบหลักโดยสังเกตผลที่ได้โดยตรง และเมื่อพบราระประกอบหลักต่าง ๆ แล้วจึงค่อยศึกษาใกล้การทำงาน และชีวโมโนเลกุลต่าง ๆ ที่เกี่ยวข้องกับการทำงานของสารประกอบเหล่านั้น และนำความรู้ความเข้าใจที่ได้กลับมาปรับปรุงโครงสร้างของสารประกอบหลัก เพื่อให้ได้สารประกอบที่มีคุณสมบัติทางยาที่ดีมากขึ้น. แต่แนวทางการพัฒนาฯ เช่น กลีเวค ที่อภิรายไปในเกรดความรู้ รูปแบบของลูคีเมียและยารักษา ดำเนินการกลับกัน คือ เริ่มจากการเข้าใจกลไกของโรค และวิถีที่เกี่ยวข้อง (biological pathway). จากนั้น เลือกชีวโมโนเลกุลในวิถีที่เกี่ยวข้องกับโรค เป็นเป้าหมาย. แล้วจึงออกแบบเบลิกแคนต์ หรือโมโนเลกุลของสารประกอบที่จะเข้าไปจับกับชีวโมโนเลกุลเป้าหมาย เพื่อปรับการทำงานของโมโนเลกุลเป้าหมายในทางรักษาบรรเทาโรค. แนวทางหลังนี้ อาจเรียกว่า การค้นหา ยาแบบย้อนกลับ (reverse drug discovery) หรือ การค้นหาโดยกำหนดเป้าหมาย (target-based drug discovery).

การค้นหาโดยกำหนดเป้าหมาย โรคหัวใจ คอเลสเตรอรอล และกลุ่มยาสแตติน. แนวทางการค้นหาโดยกำหนดเป้าหมาย เริ่มที่การเข้าใจกลไกของการทำงานของร่างกายและกลไกของโรค หรือเข้าใจเหตุก่อน. จากนั้นเลือกเป้าหมายที่อยู่ในกลไก ซึ่งอาจเป็นโปรตีน หรือดีเอ็นเอ หรืออาร์เอ็นเอ แล้วจึงหาลิแกนต์ ซึ่งคือโมโนเลกุลจะเข้าไปจับกับเป้าหมาย และเปลี่ยนการทำงานของเป้าหมาย ในทางที่จะช่วยแก้ไขกลไกที่เป็นเหตุของโรค. การค้นพบกลุ่มยาสแตติน (Statins) เป็นตัวอย่างหนึ่งของการค้นพบยาโดยกำหนดเป้าหมาย.

หลังสงครามโลกครั้งที่สองส่งบ ยุโรปหลังสงครามลำบากมาก ขาดแคลนแพทย์ทุกอย่าง แต่ อันเชล กีส (Ancel Keys) นักผจญภัยและนักวิทยาศาสตร์ จากมินนิโซตา สหรัฐอเมริกา พบรัฐิติที่น่าสนใจ คือ สถิติการตายด้วยโรคหัวใจในยุโรปหลังสงครามลดลงอย่างมาก ขณะที่สถิติในอเมริกาสูงมาก. กีสสังสัย และศึกษาว่าอะไรเป็นปัจจัยต่อการตายด้วยโรคหัวใจ นอกจากนั้น ระหว่างท่องเที่ยว กีสพบว่า ชาวประมงในเนปีล อิตาลี มีระดับคอเลสเตรอรอลในกระแสเลือดต่ำกว่าระดับคอเลสเตรอรอลของนักธุรกิจอเมริกันมาก ๆ. จากข้อมูลที่เห็น กีสเชื่อเลยว่า คนรายกินอาหารที่อุดมด้วยไขมัน และกีหัวใจ วายมากกว่า. แต่ตอนนั้น ส่วนใหญ่ไม่ได้เชื่อแบบกีส.

ถึงแม้ว่าก่อนหน้านั้น มีงานศึกษาที่พบร่วม หลอดเลือดแดงใหญ่จากเนื้อเยื่อผู้ป่วยโรคท่อเลือดแดงและหลอดเลือดแดง แข็ง มีคอเลสเตรอรอลมากกว่าที่เนื้อเยื่อปกติมี ถึงกว่า 5 เท่า และถ้าให้สัตว์กินคอเลสเตรอรอลมาก ๆ แล้วมันจะป่วยเป็นโรคภาวะไขมันในเลือดสูง และโรคท่อเลือดแดงและหลอดเลือดแดงแข็ง แต่คนส่วนใหญ่ก็ยังไม่ค่อยมั่นใจเท่าไรว่า อาหารระดับคอเลสเตรอรอลในกระแสเลือด และโรคหัวใจ มันเกี่ยวข้องกัน. ดังนั้น กีสและเพื่อนนักวิจัย ได้ร่วมกันทำโครงการวิจัยระดับนานาชาติ เพื่อศึกษาปัจจัยความเสี่ยงต่อการหัวใจวาย โดยครอบคลุมกลุ่มตัวอย่างมากกว่า 12,000 คน จากที่ตั่ง ๆ ของโลก ยุโรป แอฟริกา อิตาลี กรีก ฟินแลนด์ เนเธอร์แลนด์ ญี่ปุ่น และสหรัฐอเมริกา ซึ่งแต่ละที่มีวัฒนธรรมอาหารการกินที่แตกต่างกันมาก. ผลการศึกษาพบ ความสัมพันธ์ระหว่างอาหารที่กินกับระดับคอเลสเตรอรอลในกระแสเลือด และยืนยันว่า ระดับคอเลสเตรอรอลในกระแสเลือดเป็นปัจจัยเสี่ยงหลักต่อการเป็นโรคหัวใจ คนที่มีระดับคอเลสเตรอรอลในกระแสเลือดสูงกว่า 260 มิลลิกรัมต่อลิตร จะมีโอกาสที่จะหัวใจวายเป็นห้าเท่าของคนที่มีระดับคอเลสเตรอรอลในกระแสเลือดต่ำกว่า 200 มิลลิกรัมต่อลิตร.

สิ่งหนึ่งที่ควรระลึก คือ เช่นเดียวกับหลาย ๆ อย่างในธรรมชาติและชีวิต ไม่มีอะไรที่ดีหรือชั่วด้วยสมบูรณ์. คอเลสเตรอรอลไม่ใช่สิ่งชั่วร้าย น่ารังเกียจ ที่ต้องกำจัดออกไปให้สิ้นเชิง ถอน根ถอนโคน. คอเลสเตรอรอลเป็นสิ่งที่จำเป็นกับชีวิต ร่างกายเราต้องการคอเลสเตรอรอล คอเลสเตรอรอลเป็นส่วนประกอบสำคัญของเยื่อหุ้มเซลล์ในเซลล์ของสัตว์ทุกชนิด (รวมถึงเซลล์ของเราร้าว). 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.”

--Paracelsus

“ทุกสิ่งล้วนเป็นพิษ ไม่มีสิ่งใดปราศจากพิษ
ปริมาณเท่านั้นที่จะทำให้มันเป็นพิษ.”

—พาราเซลซัส

ช่วงปี ค.ศ. 1969-1970 ระหว่างที่นายแพทย์โจเซฟ โกลด์สไตน์ (Joe Goldstein) ทำงานที่โรงพยาบาลของสถาบันหัวใจแห่งชาติ (National Heart Institute) ในเมืองเบресดา รัฐแแมริแลนด์ สหรัฐอเมริกา โกลด์สไตน์ได้คุยกับคนไข้เด็กสองคนที่ป่วยเป็นโรคภาวะไขมันในเลือดสูงทางพันธุกรรม (familial hypercholesterolemia คำย่อ FH). เด็กทั้งสองเป็นพี่น้องกัน อายุแค่หกขวบกับแปดขวบเท่านั้น แต่มีค่าコレสเตอรอลในเลือดอยู่ในระดับสูงมาก คืออยู่ในช่วง 800 มิลลิกรัมต่อเดซิลิตร.

โกลด์สไตน์สนใจกรณีนี้มาก และศึกษากรณีร่วมกับไมเคิล บราน์ (Michael Brown). ตอนนั้นในการแพทย์รู้อยู่แล้วว่า ร่างกายมีการสังเคราะห์คอเลสเตอรอล และการสังเคราะห์คอเลสเตอรอลเป็นกลไกการควบคุมแบบป้อนกลับ นั่นคือ ถ้าให้อาหารที่มีค่าコレสเตอรอลสูงกับสุนัข ร่างกายของสุนัขจะหยุดการสังเคราะห์คอเลสเตอรอลลง. ความรู้นี้ทำให้โกลด์สไตน์และบราน์สงสัยว่า เด็กทั้งสองอาจจะมีการผิดปกติในกลไกการควบคุมแบบป้อนกลับนี้.

ขณะที่เพื่อน ๆ ของโกลด์สไตน์และบราน์ ส่วนใหญ่ศึกษาเรื่องมะเร็ง หรือประสาทวิทยา หรือเรื่องอื่น ๆ ที่อยู่ในกระแส แต่ทั้งโกลด์สไตน์และบราน์ ตัดสินใจที่จะศึกษาเรื่องกลไกควบคุมคอเลสเตอรอลอย่างจริงจัง ถึงแม้เพื่อน ๆ ของเขาก็จะชอบล้อเลียนว่า “มันก็แค่ก้อนหุ่ย ๆ ไร้ประโยชน์” โกลด์สไตน์และบราน์ ได้ทำงานร่วมกันอย่างเป็นทางการ หลังจากทั้งคู่พยายามศึกษาการแพทย์ตะวันตกเฉียงใต้มหาวิทยาลัยเท็กซัส. ระหว่างสองปีที่ทั้งคู่มุ่งมั่นทำงานหนัก ปริศนาเกลิกควบคุมคอเลสเตอรอลก็เฉลย.

โกลด์สไตน์และบราน์ เริ่มสืบจากวิถีการสังเคราะห์คอเลสเตอรอลที่วงการแพทย์ตอนนั้นรู้ดีอยู่แล้ว. ทั้งคู่มุ่งความสนใจที่อัตราการสังเคราะห์คอเลสเตอรอล ซึ่งจะขึ้นกับเอนไซม์ในขั้นแรกของวิตามินซี ชื่อ เอชเออเจ โคเอ รีดักเตส (HMG-CoA reductase หรือ 3-hydroxy-3-methyl-glutaryl-coenzyme A reductase) ซึ่งจะเรียกว่า “รีดักเตส”. ถ้ารีดักเตสทำงานมาก คอเลสเตอรอลจะถูกสังเคราะห์ออกมาก.

การทำงานของรีดักเตส จะอยู่ที่ตับ เพราะฉะนั้น โกลด์สไตน์และบราน์ไม่สามารถศึกษาการทำงานของรีดักเตสโดยตรงได้. ทั้งคู่ตัดสินใจ ศึกษาการทำงานของรีดักเตสจากเซลล์ที่ตัดและนำมาระบายน้ำเพาะเลี้ยงไว้แทน. เซลล์ที่เพาะเลี้ยงในหลอดทดลอง ต้องการสารอาหารที่จะป้อนให้ในรูปซีรัม (serum ซึ่งเป็นน้ำเลือดที่ไม่มีเม็ดเลือด). โกลด์สไตน์และบราน์ สังเกตว่าการทำงานของรีดักเตสถูกควบคุมจากอะไรบางอย่างในซีรัม คือ พอยาชีรัม การทำงานของรีดักเตสลดลง แต่พอเอา ya-chi-rum ออก การทำงานของรีดักเตสเพิ่มขึ้นเป็นสิบเท่า. โกลด์สไตน์และบราน์สงสัย และค้นหาว่าอะไรในซีรัมที่ควบคุมการทำงานของรีดักเตส จนพบว่า ไขมันโปรตีนเบา (low-density lipoprotein คำย่อ LDL) เป็นตัวยับยั้ง (inhibitor) การทำงานของรีดักเตส.

โกลด์สไตน์และบราน์มีสมมติฐานว่า ผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม ที่ร่างกายสร้างคอเลสเตอรอลมากเกินไป อาจเพราะมีการกลยุพัณฑุ์ของยินของรีดักเตส ที่ทำให้มีการสร้างรีดักเตสที่ผิดปกติและไม่ตอบสนองต่อไขมันโปรตีนเบา. ทั้งคู่ทำการทดลอง และพบว่า เซลล์จากผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม มีการทำงานของรีดักเตสมากกว่าเซลล์ปกติ สีสันถึงหากสิบเท่า และไขมันโปรตีนเบาไม่มีผลต่อการทำงานของรีดักเตส. แต่การทดลองต่อมาของทั้งคู่ กลับไม่พบความผิดปกติในตัวเอนไซม์รีดักเตสของผู้ป่วย ซึ่งชี้ว่า สมมติฐานรีดักเตสผิดปกติไม่ถูกต้อง.

ไขมันโปรตีนเบาจับตัวได้กับเซลล์ปกติ แต่ไม่ทำให้เซลล์ผู้ป่วย. รีดักเตสของเซลล์ผู้ป่วยไม่ได้ผิดปกติ. ดังนั้น น่าจะต้องมีอะไรระหว่างกลาง ที่เป็นปัจจัย. ไขมันโปรตีนเบา จะประกอบไปด้วยโปรตีน ที่เรียกว่า ลิปอโปรตีน และไขมัน ซึ่งรวมถึงคอเลสเตอรอล. โกลด์สไตน์และบราน์ ทดลองป้อนเฉพาะคอเลสเตอรอล โดยไม่มีลิปอโปรตีน และพบว่า คอเลสเตอรอลยับยั้งการทำงานของรีดักเตสอย่างชัดเจน ทั้งในเซลล์ปกติและเซลล์ผู้ป่วย. นั่นคือ รีดักเตสของผู้ป่วยทำงานได้ปกติ ถูกควบคุมด้วยคอเลสเตอรอลได้เหมือนกับรีดักเตสปกติ แต่ถูกควบคุมไม่ได้ถ้าคอเลสเตอรอลอยู่ในรูปไขมันโปรตีนเบา.

ไขมันโปรตีนเบาจับตัวได้กับเซลล์ปกติ แต่ไม่จับกับเซลล์ของผู้ป่วย. เซลล์ปกติมีรีเชปเตอร์สำหรับจับตัวกับไขมันโปรตีนเบา แต่เซลล์ของผู้ป่วยไม่มี. โกลด์สไตน์และบราน์ ศึกษาใกล้ๆ และพบว่า ลิปอโปรตีนของไขมันโปรตีนเบา นำคอเลสเตอรอลไปให้เซลล์ โดยตัวลิปอโปรตีนจะจับตัวกับรีเชปเตอร์ไขมันโปรตีนเบา (LDL receptors) และคอเลสเตอรอล

จะถูกแยกออกจากโปรตีนตอนที่เข้าไปอยู่ในเซลล์ ซึ่งคอลเลสเตรอรอลจะสามารถเข้าควบคุมการทำงานของรีดักเตสได้.

นั่นคือ ในเซลล์ปกติ ไขมันโปรตีนเบา (ซึ่งมีคอลเลสเตรอรอลอยู่) จับกับรีเซปเตอร์ไขมันโปรตีนเบา และส่งผลยับยั้งการทำงานของรีดักเตส. แต่ในเซลล์ของผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพัณฑุรกรรม ไขมันโปรตีนเบา (ซึ่งมีคอลเลสเตรอรอลอยู่) ไม่สามารถส่งคอลเลสเตรอรอลเข้าไปในเซลล์ได้ การทำงานของรีดักเตสมีถูกยับยั้ง และส่งผลให้มีการสังเคราะห์คอลเลสเตรอรอลอ่อนมาก มากกว่าในเซลล์ปกติหลักๆ เท่านั้น.

ช่วงเวลาใกล้เคียงกัน อา基ระ เอนโดะ (Akira Endo) ที่ขณะนั้นทำงานกับบริษัทยาชันเคียว ในโตเกียว ญี่ปุ่น พยายามค้นหาสารประกอบเพื่อยับยั้งการทำงานของรีดักเตส. เอนโดะมีประสบการณ์จากการงานก่อนหน้า ที่เขียนพับเอนไซม์จากราเพื่อย่อยเนื้อผลไม้ที่ป่นมาในไวน์และเหล้าผลไม้. เอนโดะรู้เรื่องของราบานชนิด ที่มีโมเลกุลออกอสเตรอรอล (ergosterol) เป็นส่วนประกอบสำคัญของเยื่อหุ้มเซลล์ แทนที่จะเป็นคอลเลสเตรอรอล เขายังคิดว่า ราบานชนิดน่าจะมีสารประกอบที่ยับยั้งกระบวนการสังเคราะห์คอลเลสเตรอรอลได้.

เอนโดะกับทีมงานค้นหาสารประกอบที่อยากได้ โดยค้นหาจากราปะมาน 6000 ชนิด และทดสอบดูว่า น้ำจากการแต่ละชนิด จะยับยั้งการทำงานของรีดักเตสได้หรือไม่ จากการค้นหาอยู่สองปี เอนโดะกับทีมงาน พบรากอฤทธิ์สกัดจากราสองชนิดที่ยับยั้งการทำงานของรีดักเตสได้ ตัวหนึ่งได้มาจาก ไฟเรียม อัลติมัม (*Pythium ultimum*) ซึ่งกล้ายเป็นยาปฏิชีวนะที่รู้จักกันอยู่แล้ว ชื่อ ซิตринิน (citrinin). ซิตринินยับยั้งการทำงานของรีดักเตสได้ แต่เป็นพิษมาก. อีกด้านหนึ่งได้จากรา เพนนิซิลเลียม ซิตринัม (*Penicillium citrinum*) ซึ่งมาจากสกุลเดียวกับราที่ใช้สกัดยาเพนนิซิลลิน.

สำหรับศึกษาและพัฒนาความเป็นยาต่อ เอนโดะ กับทีมงานต้องเพาะ เลี้ยง เพนนิซิลเลียมซิตринัมมากถึง 600 ลิตร เพื่อที่จะสกัดสารประกอบมาได้ปริมาณ 23 มิลลิกรัม และพบว่า โมเลกุล ML-236B ซึ่งภายหลังคือ คอมแพคติน (Compactin หรือชื่ออื่น เมวาสแตติน Mevastatin) เป็นสารออกฤทธิ์. เอนโดะกับทีมงานเผยแพร่การค้นพบนี้[\[63\]](#) และพัฒนาคอมแพคตินต่อเพื่อเป็นยา ซึ่งคือการทดลองในสัตว์. การทดลองในหนู แม้ว่าไม่พบผลเป็นพิษ แต่คอมแพคตินไม่ช่วยลดคอลเลสเตรอรอลในกระเส้นเลือดของหนูเลย ไม่ว่าจะให้ยาอยู่เจ็ดวัน หรือใช้ขนาดยาสูงอยู่ถึงห้าสัปดาห์.

ผิดหวัง แต่เอนโดะยังไม่ยอมแพ้. จากการทดลองที่ผ่านๆ มา เอนโดะสังสัยว่า ที่คอมแพคตินไม่เป็นผลกับหนู อาจเป็นเพราะ ร่างกายของหนูมีกลไกควบคุมคอลเลสเตรอรอลที่ต่างไป และเอนโดะจึงเริ่มการทดลองใหม่ในไก่ ซึ่งได้ผลดีมาก และผลในลิงและผลในสุนัข ก็พบการลดลงของคอลเลสเตรอรอลอย่างเด่นชัด. โอกาสของคอมแพคตินเริ่มสดใส และชันเคียวที่ทำการสนับสนุนอย่างเต็มที่. แต่ นักพิชวิทยาเห็นความผิดปกติในเซลล์ตับของหนูที่ให้คอมแพคตินที่ขนาดยาสูงมาก สุดท้ายหลังจากได้รับอยู่ห้าวัน เดือน ชันเคียวที่ตัดสินใจดำเนินการทดลองทางคลินิก. แต่แล้ว ชันเคียว ก็สั่งหยุดการพัฒนาคอมแพคตินทันที หลังจาก นักพิชวิทยาของบริษัททรงสัญญา สนับที่ให้คอมแพคตินที่ขนาดยาสูงติดต่อกันสองปี จะมีเนื้องอกในลำไส้.

ในช่วงนั้น บริษัทยาต่าง ๆ รู้เรื่องการพัฒนาคอมแพคตินของชันเคียว. รอย วาเจโลส (Roy Vagelos) หัวหน้าฝ่ายวิจัยของบริษัทเมอร์ค อยากรจะเปลี่ยนวิธีการค้นหาฯ จากเดิมที่การค้นหาสารประกอบทำด้วยการทดลองกับเซลล์หรือจุลทรรศน์ วาเจโลสพยายามเปลี่ยนเป็นการทดลองกับโมเลกุลเป้าหมาย.

จากการของโกลด์สไตน์และบราร์น และการค้นพบคอมแพคตินของเอนโดะ วาเจโลสเห็นโอกาสที่จะได้ลองวิธีใหม่นี้. วาเจโลสและทีมงานที่เมอร์ค ค้นหาแบบคอมแพคตินจากการชนิดอื่น ๆ และสุดท้าย พบรากอฤทธิ์อัสเพอร์จิลลัส เทโรเรียส (*Aspergillus terreus*) ซึ่งภายหลังคือ โลวาสแตติน (Lovastatin). แต่หลังจากที่เมอร์ครู้ช่าวันเคียวยกเลิกการพัฒนาคอมแพคตินทันที หลังจาก นักพิชวิทยาของบริษัททรงสัญญา สนับที่ให้คอมแพคตินที่ขนาดยาสูงติดต่อกันสองปี จะมีเนื้องอกในลำไส้.

โกลด์สไตน์และบราร์นเองก็รู้เรื่องงานของเอนโดะ. ทั้งคู่สนใจ ติดต่อกับเอนโดะ และได้ตัวอย่างคอมแพคตินมาทดลอง ซึ่งผลการทดลอง นอกจากแสดงในเห็นว่า เมื่อใช้คอมแพคติน การทำงานของรีดักเตสลดลงชัดเจนแล้ว. สิ่งที่โกลด์สไตน์และบราร์นพบร่วมกันคือ เซลล์สร้างรีดักเตสเพิ่มขึ้น.

ในขณะที่การสังเคราะห์คอลเลสเตรอรอล ถูกควบคุมด้วยการทำงานของรีดักเตส การสังเคราะห์รีดักเตสเองก็ถูกควบคุมยับยั้งด้วยปริมาณคอลเลสเตรอรอล. ผลการทดลองที่โกลด์สไตน์และบราร์นพบร่วมกัน แสดงให้เห็นถึง อิทธิพลภาครเสริม (*double-*

negative effect) ในการควบคุมปริมาณคอเลสเทอรอล. นั่นคือ การยับยั้งการทำงานของรีดักเตส ส่งผลให้มีมีคอเลสเทอรอลผลิต เมื่อไม่มีคอเลสเทอรอล ก็ไม่มีอิทธิพลต่อการสังเคราะห์รีดักเตส ดังนั้นปริมาณรีดักเตสจะเพิ่มขึ้น. หมายเหตุ แม้ปริมาณของรีดักเตสเพิ่มขึ้น แต่รีดักเตสไม่ได้ทำงาน.

โกลเดิลส์ไตน์และบราร์น์ ได้ใจมากกับการค้นพบนี้ เพราะว่า งานวิจัยก่อนหน้านี้ทำให้ทั้งคู่รู้ว่า การสังเคราะห์รีดักเตสและรีเซบ/เตอร์ไซมันโปรตีนเบาถูกควบคุมไปพร้อม ๆ กัน ดังนั้น การเห็นการสังเคราะห์รีดักเตสเพิ่ม ก็อาจหมายถึงการสังเคราะห์รีเซบ/เตอร์ไซมันโปรตีนเบาเพิ่มด้วย. การเพิ่มรีเซบ/เตอร์ไซมันโปรตีนเบา ก็น่าจะทำให้เซลล์สามารถดึงไขมันโปรตีนเบาจากกระแสเลือดเข้าเซลล์ได้มากขึ้น และลดระดับคอเลสเทอรอลในกระแสเลือด ที่เป็นสาเหตุของการหัวใจวาย.

นั่นคือ โกลเดิลส์ไตน์และบราร์น์ วางแผนติดธุรานว่า สำหรับผู้ป่วยโรคภาวะไขมันในเลือดสูงทางพันธุกรรม รีเซบ/เตอร์ไซมันโปรตีนเบามีจำนวนน้อย ทำให้คอเลสเทอรอลในกระแสเลือดมีปริมาณมาก. แต่เมื่อใช้คอมแพคตินแล้ว การทำงานของรีดักเตสลดลง การสังเคราะห์คอเลสเทอรอลในเซลล์ลดลง การสังเคราะห์รีดักเตสและรีเซบ/เตอร์ไซมันโปรตีนเบาเพิ่มขึ้น. รีเซบ/เตอร์ไซมันโปรตีนเบามีจำนวนเพิ่มขึ้น สามารถรับไขมันโปรตีนเบาจากกระแสเลือดเข้ามาในเซลล์ได้เพิ่มขึ้น ช่วยให้ภายในเซลล์มีคอเลสเทอรอลใช้งาน และขณะเดียวกันก็ช่วยให้คอเลสเทอรอลในกระแสเลือดมีปริมาณลดลง.

ทั้งคู่ทดสอบสมมติฐาน โดยขอตัวอย่างโลวัสแตตินมาจากเมอร์ค และทดลองกับสุนัข. ผลคือ ทั้งรีเซบ/เตอร์ไซมันโปรตีนเบามีจำนวนเพิ่มขึ้น และคอเลสเทอรอลในกระแสเลือดมีปริมาณลดลง. ทั้งคู่มุ่งเน้นไปที่ผลการทำงาน แต่จะพยายามให้ผู้ป่วยจากใหม่ ในเมื่อทั้งชันคีวยและเมอร์คกีรังับการพัฒนา เนื่องจากกลัวความเสี่ยงของการเกิดเนื้องอกในลำไส้. โกลเดิลส์ไตน์และบราร์น์ตัดสินใจเปลี่ยนเพื่อบริกากับเอนโด. ตอนนั้นเอนโดได้ทำงานให้ชันคีวยแล้ว เอนโดใช้ยาไปทำงานที่มหาวิทยาลัยเกษตรและเทคโนโลยีโตเกียว. เอนโดใช้ เชื่อว่านักพิษวิทยาอาจจะตีความผลที่เห็นในสุนัขผิด และคิดว่าสิ่งที่นักพิษวิทยาเห็นในลำไส้ อาจจะไม่ใช่เนื้องอก อาจจะเป็นยาที่ไม่ย่อยมากกว่า เพราะว่า การทดลองใช้ขนาดยาที่สูงมาก ซึ่งมากกว่าที่จะใช้ในคนถึงร้อยเท่า.

ค่อนข้างมั่นใจกับยา และด้วยโลวัสแตตินที่ได้มา โกลเดิลส์ไตน์และบราร์น์ร่วมกับเพื่อนอีส่องคน ทดสอบยา กับผู้ป่วยภาวะไขมันในเลือดสูงทางพันธุกรรมจำนวนหกคน และผลที่ได้ คือรีเซบ/เตอร์ไซมันโปรตีนเบามีจำนวนเพิ่มขึ้น และคอเลสเทอรอลในกระแสเลือดลดลงประมาณ 27%. ผลที่ได้นี้ ช่วยให้เมอร์คตัดสินใจกลับมาพัฒนาโลวัสแตตินต่อ. แต่ผู้บริหารของเมอร์ค ก็ยังกังวลกับความเสี่ยงจากการเนื้องอกอยู่. เพื่อทำประดีเรื่องเนื้องอกให้ชัดเจน และโอกาสในการใช้ยา กับผู้ป่วยภาวะไขมันในเลือดสูงทั่วไป อีเดเวิร์ด สโคลนิก (Edward Skolnick) หัวหน้าฝ่ายวิจัยพื้นฐานของเมอร์ค ตั้งทีมงานเฉพาะขึ้นมา เพื่อศึกษาผลทางพิษวิทยาให้สมบูรณ์. สโคลนิกปรึกษากับโกลเดิลส์ไตน์และบราร์น์ และโกลเดิลส์ไตน์และบราร์น์ได้แนะนำวิธีการทดสอบ เพื่อระบุว่า สิ่งที่เห็นในสัตว์ทดลองว่าเป็นผลจากยาจริง ๆ หรือว่าแค่มาจากการทดสอบตัวยานวดยา สูงมาก ซึ่งสามารถป้องกันได้ง่าย ๆ. ทีมงานนักวิจัยของสโคลนิกทดลอง และไม่พบผลร้ายจากยา. สโคลนิกลองออก และเมอร์ค มั่นใจในความปลอดภัยของยา.

ผลจากการทดสอบอยู่สองปี ยืนยันว่า โลวัสแตตินช่วยลดคอเลสเทอรอลในกระแสเลือดได้กว่า 20% เมอร์คยื่นจดทะเบียนยา และได้เริ่มขายโลวัสแตตินในปี ค.ศ. 1987. ทั้งโลวัสแตติน และคอมแพคติน รวมไปถึงยาที่พัฒนาขึ้นมาภายหลังตัวอื่น ๆ ในกลุ่มนี้ จะเรียกว่า กลุ่มยาสแตติน (Statins). เพื่อการติดตามผลการใช้ยา เมอร์คสนับสนุนการศึกษาห้าปี กับผู้ป่วยระดับคอเลสเทอรอลในกระแสเลือดสูง จำนวน 4,444 คน ที่ใช้ยาชีวภาพเดติน (ซึ่งเป็นยาในกลุ่มสแตติน ที่พัฒนาขึ้นมาภายหลัง) และพบว่า ยาช่วยลดอัตราการตายจากหัวใจวายของผู้ป่วยลง 42%. ปัจจุบัน มีผู้ใช้ยาในกลุ่มสแตตินมากกว่า 5 ล้านคนทั่วโลก และอัตราการตายจากหัวใจวายของชาวอเมริกันลดลงเกือบหนึ่งครั้ง (นับจากที่ อันเชล กี๊ส พบอันตรายจากคอเลสเทอรอล).

อุตสาหกรรมยา การค้นหาและพัฒนา. โรนัล คริสโตเฟอร์ (Ronald Christopher) จากบริษัทยาอาเรينا บรรยายเรื่อง การเลือกสารประกอบและการศึกษา ก่อนคลินิก[66] ว่า การค้นหายาเป็นกิจกรรมที่อัตราการล้มเหลวสูงมาก ประมาณหนึ่งในพัน นั่นคือ จากขั้นตอนแรก ๆ อาจมีสารประกอบที่สนใจอยู่ประมาณห้าพันถึงหกพันตัว สุดท้ายจะเหลือแค่

ประมาณสิบตัวที่ผ่านกระบวนการไปจนถึงการทดสอบทางคลินิกกับมนุษย์ได้. ระยะเวลาในการค้นหายา โดยเฉลี่ย จะประมาณสิบสองปี และค่าใช้จ่ายในการพัฒนายาแต่ละตัว ประมาณ 1.3 พันล้านдолลาร์หรือประมาณสี่แสนล้านบาท ต่อการค้นหาและพัฒนายาที่จะได้รับการขึ้นทะเบียนหนึ่งตัว. (หมายเหตุ คณะของดิมาสี[55] ประมาณตัวเลขอยู่ที่ 2.87 พันล้านдолลาร์ แต่ประสาตและมายานาคอดี[155] ประมาณค่าใช้จ่ายอยู่ที่ 648 ล้านдолลาร์ซึ่งต่ำกว่ามาก. อย่างไรก็ตาม 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> ซึ่งวิจารณ์วิธีประเมินของประสาตและมายานาคอดี โดยเฉพาะเรื่องที่ประสาตและมายานาคอดี ไม่ได้รวมค่าใช้จ่ายของความล้มเหลวในกระบวนการค้นหายาเข้าไปด้วย. เออร์เปอร์วิจารณ์ว่า ผลสรุปของประสาตและมายานาคอดี เป็นลักษณะของความลำเอียงไปทางคนที่รอด *survivorship bias*. ความลำเอียงไปทางคนที่รอด หมายถึง การวิเคราะห์ที่ใช้ผลสรุปแทนภาพรวมทั้งหมด แต่ใช้ข้อมูลเฉพาะจากกลุ่มข้อมูลที่ทำได้ดีหรือกลุ่มผู้รอด. ไม่ว่าจะอย่างไร กิจกรรมการค้นหาพัฒนายาเป็นกิจกรรมที่ลงทุนมหาศาล อาศัยเครื่องมือขั้นสูงและทักษะกับความทุ่มเทอย่างยิ่งของบุคคลากรที่เกี่ยวข้อง.)

رونัลด์ คริสโตเฟอร์ อธิบายการเลือกสารประกอบมาเป็นยาว่า มีเกณฑ์ในการพิจารณาอย่าง ๆ หลาย เช่น คุณสมบัติทางเภสัชวิทยา ได้แก่ การออกฤทธิ์ที่ดี สมรรถนะการเลือกที่สูง (สารประกอบจับตัวกับเป้าหมายติกว่าจับตัวกับชีวะโนเมเลกุล อื่นในร่างกาย มากกว่าพันเท่า) ประสิทธิผลที่ดีในการทดลองกับสัตว์ (แสดงให้เห็นว่ามันได้ผล). นอกจากนั้น กีังพิจารณาเรื่อง เมแทบอลิซึมของยา และคุณสมบัติทางเภสัชจุนศาสตร์ (ร้ายกาจตอบสนองต่อยาอย่างไร) รวมถึง การปฏิสัมพันธ์ระหว่างยา กับยา (drug-drug interactions) ว่ายาตัวใหม่นี้จะไม่ไปกวนยาอื่นที่ผู้ป่วยใช้อยู่ และปัจจัยด้านความปลอดภัย เช่น ผลการศึกษาด้านความปลอดภัยในทางที่ดีทั้งการศึกษาในหลอดทดลอง และในสัตว์ทดลอง.

คริสโตเฟอร์ยกตัวอย่างประสบการณ์การพัฒนา Yao Lokalibdin สำหรับบำบัดโรคเบาหวาน. โรคเบาหวาน เป็นภาวะที่ร่างกายมีน้ำตาลในเลือดสูง. ระดับน้ำตาลในเลือด (blood glucose) ถูกควบคุมด้วยอินซูลิน (insulin). การปล่อยอินซูลินถูกควบคุมด้วยอินซีรีโนอินคริตินส์ (incretins[109]) การควบคุมอินคริตินส์ถูกควบคุมด้วยดีพีพีสี (Dipeptidyl peptidase-4 คำย่อ DPP-4). การควบคุมในร่างกายมีอยู่สองแบบหลัก ๆ ได้แก่ การควบคุมเชิงบวก คือการสนับสนุนหรือกระตุ้น และควบคุมเชิงลบ คือการลดหรือยับยั้ง.

อินซูลินควบคุมระดับน้ำตาลในเลือดในเชิงลบ. อินคริตินส์ควบคุมอินซูลินในเชิงบวก แต่ดีพีพีสีควบคุมอินคริตินส์ในเชิงลบ. นั่นคือ หากอินซูลินเพิ่มขึ้น ระดับน้ำตาลในเลือดจะลดลง. หากอินคริตินส์เพิ่มขึ้น อินซูลินจะเพิ่มขึ้น. แต่หากดีพีพีสีทำงาน อินคริตินส์จะลดลง ส่งผลให้อินซูลินลดลง ส่งผลให้ระดับน้ำตาลในเลือดเพิ่มขึ้น. ยาบางตัว เช่น เอ็กซีนาไทด์ (Exenatide) เลือกเป้าหมายเป็นรีเซปเตอร์ของอินคริตินส์. ตัวยาจะไปจับกับรีเซปเตอร์ของอินคริตินส์ เพื่อส่งผลเหมือนการเพิ่มของอินคริตินส์. เอ็กซีนาไทด์ เป็นยาชนิดและมีผลข้างเคียงค่อนข้างมาก. ทีมงานของคริสโตเฟอร์ เลือกเป้าหมายเป็นดีพีพีสี และต้องการหาโมเลกุลที่ยับยั้งดีพีพีสี (DPP4 inhibitor). การยับยั้งดีพีพีสี เท่ากับเพิ่มการทำงานของอินคริตินส์ อินคริตินส์ทำงานมากขึ้นจะไปเพิ่มอินซูลิน อินซูลินเพิ่มขึ้นจะไปลดระดับน้ำตาลในเลือด.

ตอนนั้น มียาที่ยับยั้งดีพีพีส์ในตลาดอยู่หลายตัวแล้ว เช่น วิลดากลิปทิน. ทีมงานของคริสโตเฟอร์ ศึกษาโครงสร้างทางเคมีของดีพีพีสี ซึ่งเป็นงานที่มีขั้นตอนที่ซับซ้อนมาก ตั้งแต่การโครงสร้างดีพีพีสี และทำกระบวนการต่าง ๆ ที่จะทำให้ดีพีพีสีตกลง และถ่ายภาพดีพีพีสีที่ตกผลึกด้วยเอ็กซ์เรย์ ซึ่งต้องใช้เครื่องซินโครตรอน. ภาพถ่ายที่ได้จะเป็นภาพของรูปแบบการกระเจิงของเอ็กซ์เรย์ ซึ่งต้องใช้นักชีววิทยาโครงสร้างอ่าน ตีความ และแปลงอุอกมาเป็นแบบจำลองคอมพิวเตอร์ของโครงสร้างเคมีสามมิติ ที่นักเคมีสามารถใช้วิเคราะห์ได้ต่อไป. นักเคมีในทีมงานของคริสโตเฟอร์ ดูโครงสร้างดีพีพีสี และการเข้าอุ้จับตัวกับวิลดากลิปทิน และพบว่า ในการจับตัวกันของวิลดากลิปทินและดีพีพีสี มีการจับด้วยพันธะโควาเลนท์อยู่. พันธะโควาเลนท์ทำให้โมเลกุลยาจับตัวกับดีพีพีสีแน่น. การจับดีพีพีสีแน่นเกินไป อาจก่อให้เกิดผลข้างเคียงต่อระบบภูมิคุ้มกัน. ทีมงานของคริสโตเฟอร์ ต้องการจะพัฒนายาใหม่ที่จับดีพีพีสี โดยไม่มีพันธะโควาเลนท์.

การค้นหาออกแบบและพัฒนาของทีมของคริสโตเฟอร์ ทำโดยอาศัยโครงสร้างโมเลกุล. ในการค้นยาออกแบบยาโดยทั่วไป จะมีสองแนวทางหลัก ๆ คือ อาศัยลิแกนต์ (ligand-based) หรืออาศัยโครงสร้าง (structure-based). วิธีอาศัย

ลิแกนต์ ไม่จำเป็นต้องรู้โครงสร้างทางเคมีสามมิติของเป้าหมาย แต่ต้องรู้จักบางลิแกนต์ของเป้าหมาย แล้วสร้างแบบจำลองทำนายการจับตัว และค้นหาโมเลกุลที่อาจเป็นยาได้จากแบบจำลองทำนาย (ที่สร้างโดยอาศัยข้อมูลลิแกนต์เหล่านั้น). วิธีอาศัยโครงสร้าง ต้องรู้โครงสร้างของโมเลกุลเป้าหมาย. ยา มักเป็นโมเลกุลขนาดเล็ก แต่เป้าหมาย เช่น โปรตีน เป็นโมเลกุลขนาดใหญ่. การหาโครงสร้างสามมิติของโมเลกุลขนาดใหญ่เป็นเรื่องซับซ้อนและใช้ทักษะสูง แต่หากได้โครงสร้างสามมิติของโมเลกุลเป้าหมายมาแล้ว นักเคมีจะดูโครงสร้างของตัวแทนง่ายๆ จับตัวในโปรตีนเป้าหมาย แล้วจึงพิจารณาหาลิแกนต์ โดยอาจเริ่มจากส่วนเล็ก ๆ ของโครงสร้างทางเคมีที่ต้องการ แล้วค่อยค้นหาโมเลกุลของลิแกนต์ตามนั้น หรืออาจจะค้นหาจากสารประกอบที่มีอยู่ในฐานข้อมูล ด้วยวิธีการกลั่นกรองสมீอ่อน หรืออาจจะออกแบบโดยโครงสร้างของลิแกนต์ขึ้นมาใหม่เลยก็ได้.

วิธีการกลั่นกรองสมீอ่อน (virtual screening[98] คำย่อ vs) เป็นการใช้คอมพิวเตอร์เข้ามาช่วยค้นหาโมเลกุลต่าง ๆ จากฐานข้อมูล เพื่อหาโมเลกุลที่มีโอกาสสูงในการนำมายาต่อไป. โมเลกุลต่าง ๆ ที่อาจเป็นยาได้ มีจำนวนมหาศาล. การทดสอบแต่ละโมเลกุลกับเป้าหมายในหลอดทดลองมี代价ใช้จ่ายสูง. การใช้คอมพิวเตอร์ช่วยกลั่นกรองเลือกโมเลกุลต่าง ๆ ก่อน แล้วค่อยเลือกทดสอบโมเลกุลที่ผ่านการกลั่นกรองขึ้นต้นกับเป้าหมายในหลอดทดลอง จะช่วยลดค่าใช้จ่าย เวลา และทรัพยากรบุคคลในการพัฒนา yal ได้มาก. ในทางปฏิบัติ วิธีการกลั่นกรองสมீอ่อนก็ไม่ได้ค้นหากับทุกโมเลกุลที่เป็นไปได้ แต่อาจจะเลือกจากฐานข้อมูลของยาที่ได้มีการทดสอบแล้ว อาจเลือกจากรายการของสารประกอบที่มีอยู่ในคลังของบริษัทแล้ว อาจเลือกจากฐานข้อมูลจากผู้ขาย เป็นต้น โดยอาจลำดับความสำคัญของการค้นหา จาบที่มีการทดสอบแล้ว ต่อด้วยสารต่าง ๆ ที่มีอยู่คลัง แล้วไปสารต่าง ๆ ที่สามารถจัดชั้นให้ จนสุดท้ายถึงค้นหาสารต่าง ๆ ที่จะต้องสังเคราะห์ขึ้นใหม่. หากพบโมเลกุลจากฐานข้อมูลยาที่มีการทดสอบแล้ว จะช่วยลดค่าใช้จ่ายในการศึกษาหลาย ๆ อย่างที่มีผลการศึกษาอยู่แล้ว. สารที่มีอยู่แล้วในคลังก็จัดหาได้ง่ายกว่า และสารที่สามารถหาซื้อได้ ก็สะดวกและมักเสียค่าใช้จ่ายน้อยกว่าการสังเคราะห์สารขึ้นมาเองใหม่.

วิธีการกลั่นกรองสมீอ่อน อาจใช้การทำนายการเข้าอ้อ' (docking) ซึ่ง จะทำนายรูปร่างและทิศทางการวางตัวของโมเลกุล เมื่อโมเลกุลจับตัวกับเป้าหมาย ซึ่งผลการทำนายนี้อาจใช้ประกอบ เพื่อทำนายอัตราการจับตัวกัน (binding affinity). อัตราการจับตัวกัน เป็นโอกาสของการจับตัวกันระหว่างลิแกนต์กับเป้าหมาย. แบบจำลองที่ทำนายอัตราการจับตัวกัน จะเรียกว่า ฟังก์ชันคะแนน (scoring function). ฟังก์ชันคะแนน อาจทำนายโอกาสของการจับตัว ด้วยพลังงานรวมของการจับตัวกัน โดยคำนวณจากทฤษฎีสนามแรง ซึ่งอาศัยรูปร่างและทิศทางการวางตัวของโมเลกุลที่จับตัวกัน. ค่าพลังงานที่ต่ำกว่า หมายถึงผลการจับตัวที่มีเสถียรภาพมากกว่า และโอกาสที่มากกว่าของการจับตัวกัน. การใช้แบบจำลองการเรียนรู้ของเครื่องเป็นอีกแนวทางหนึ่งที่สามารถนำมาใช้สร้างฟังก์ชันคะแนนได้. หมายเหตุ ตัวอย่างในแบบฝึกหัด 3.17 เป็นแค่การทำนายการจับตัวกัน ไม่ได้มีการประเมินโอกาสจับตัวกันของมาเป็นตัวเลข ซึ่งฐานข้อมูลดีดูดีไม่มีข้อมูลน้อย. แต่หากมีข้อมูลอัตราการจับตัวกัน ซึ่งมักวัดเป็นเปอร์เซ็นต์การจับตัวต่อความเข้มข้นของสารละลายของลิแกนต์ (%binding per molar concentration) ก็สามารถนำมาสร้างเป็นแบบจำลองได้ โดยการทำนายลักษณะนี้เป็นการทำนายค่าต่อเนื่อง และเหมาะสมที่จะวัดกรอบเป็นแบบจำลองการหาค่าติดถอย.

การค้นหาและพัฒนาฯ มักดำเนินการในลักษณะการวนวนทวนกลั่นกรอง นั่นคือ เป็นลักษณะวนค้นหา ปรับปรุง และสลับกันไป จนกว่าจะได้ลิแกนต์ที่มีลักษณะความเป็นยาสูงอกรถ.

หลังจากได้ลิแกนต์ที่ผ่านรอบแรกมา ที่มีพัฒนาจะปรับปรุงโมเลกุล โดยอาจจะค้นหาโมเลกุลอื่น ๆ ที่ใกล้เคียงกับลิแกนต์เหล่านั้น หรืออาจจะปรับโครงสร้างบางส่วนของลิแกนต์เหล่านั้น เพื่อให้มีคุณสมบัติทางยาต่าง ๆ ดีขึ้น. คุณสมบัติทางยาต่าง ๆ ที่ปรับปรุง ได้แก่ อัตราการจับตัวกับเป้าหมาย ($IC_{50} \leq 100$ นาโนโมลาร์ ซึ่งค่า IC_{50} วัดจากความเข้มข้นของลิแกนต์ที่สามารถจับกับเป้าหมายได้ครึ่งหนึ่ง), สมรรถนะการเลือก (ลิแกนต์มีการจับตัวกับเป้าหมายได้ดีกว่าจับตัวกับโมเลกุลอื่นที่คล้ายเป้าหมายหนึ่งพันเท่า), รวมถึงการออกฤทธิ์ของยา คุณสมบัติเมแทบอลิซึมของยา คุณสมบัติทางเภสัชจุนศาสตร์และเภสัชพลศาสตร์ เป็นต้น. ในขั้นตอนการพัฒนาฯ อาจมีการใช้แบบจำลองทำนาย เช่น ความล้มพ้นร์ เชิงปริมาณระหว่างโครงสร้างและกิจกรรม (Quantitative Structure-Activity Relationship[25, 130] คำย่อ QSAR) เข้ามาช่วย. ความล้มพ้นร์เชิงปริมาณระหว่างโครงสร้างและกิจกรรม เป็นการทำนายกิจกรรมหรือคุณสมบัติของโมเลกุล จาก

โครงสร้างของโมเลกุล ซึ่งกิจกรรมที่ทำนาย อาจเป็น ผลกับเป้าหมายหลังการจับตัว (ว่าเป็น ผลทำการ agonism ที่ทำให้เป้าหมายทำงานมากขึ้น หรือผลต่อต้าน antagonism ที่ลดการทำงานของเป้าหมายลง), ชีวปริมาณออกฤทธิ์ (bioavailability), การละลายในน้ำ (solubility), สมรรถนะการเลือก, การออกฤทธิ์ เป็นต้น. ความสัมพันธ์เชิงปริมาณระหว่างโครงสร้างและกิจกรรม อาจสร้างจากพื้นฐานทางฟิสิกส์และเคมี หรืออาจจะสร้างตามแนวทางการเรียนรู้ของเครื่องโดยอาศัยข้อมูลกีด้วย คล้ายกับแบบฝึกหัด 3.17 โครงสร้างทางเคมีจะถูกแปลงเป็นลักษณะสำคัญเชิงเลข ที่อาจเรียกว่า ตัวบอก (descriptor) เพื่อให้สามารถใช้คำนวณในแบบจำลองได้.

หลังจากที่มีงานของคริสโตเฟอร์ทำงานอย่างหนัก กระบวนการค้นหาออกแบบและพัฒนาโมเลกุลเสริจสิน ที่มีงานได้ อลอกลิบติน (Aloglibitin). อลอกลิบตินเข้าอยู่ในตัวกับดีพีพีสีได้ และไม่มีพันธะที่เป็นพันธะโควาเลนท์. อลอกลิบตินจับตัวกับดีพีพีสีสักพักแล้วก็หลุด และกลับไปจับตัวใหม่ สลับกันไป เปิดโอกาสให้ดีพีพีสีเป็นอิสระเป็นพัก ๆ ลดความเสี่ยงของผลข้างเคียงต่อระบบภูมิคุ้มกัน.

ที่มีงานทดสอบอลอกลิบตินในหลอดทดลอง และพบว่าอโลกลิบตินมีการออกฤทธิ์ที่ดี ($IC_{50} = 6.9$ ซึ่ง IC_{50} สำหรับการออกฤทธิ์ของตัวยับยั้ง วัดจากความเข้มข้นของสารละลายยา ที่เพียงพอที่จะยับยั้งการทำงานของเป้าหมายได้ 50%. ตั้งนั้นตัวเลขน้อยกว่า หมายถึงการออกฤทธิ์ที่ดีกว่า). ยาตัวอื่นในกลุ่มเดียวกันที่อยู่ในตลาด มีการออกฤทธิ์วัดด้วย IC_{50} เป็น 23.8 และ 12.1 ซึ่งทั้งหมดจัดว่ามีการออกฤทธิ์ที่ดี.

ผลการทดสอบสมรรถนะการเลือกของอลอกลิบตินก็ออกมาดี. อลอกลิบติน มีอัตราการจับตัวกับดีพีพีสี ดีกว่าการจับตัวกับโปรตีนที่ใกล้เคียงมากกว่าหนึ่งแสนเท่า สำหรับโปรตีนที่ใกล้เคียงแต่ละตัว ซึ่งได้แก่ DPP-2, DPP-8, DPP-9, FAP, PREP, และ Tryptase.

การทดสอบกับสัตว์ทดลอง แสดงผลที่ดีเยี่ยมกัน. ผลในลิง แสดง (1) ความเข้มข้นของยาในกระแสเลือดหลังจากรับยาเข้าไป โดยความเข้มข้นเพิ่มจนไปถึงจุดสูงสุด ใช้เวลาประมาณหนึ่งชั่วโมง หลังจากนั้นค่อย ๆ ลดลง และความเข้มข้นของยาในกระแสเลือดเป็นไปตามขนาดยาที่ได้ ซึ่งนี้เป็นการดูการตอบสนองของร่างกายต่อยา ซึ่งผลเป็นไปตามที่ทีมงานคาด และ (2) เปรอร์เซ็นต์การยับยั้งการทำงานของดีพีพีสี หลังรับยาเข้าไป ซึ่งเปอร์เซ็นต์ยับยั้งสูงสุดอยู่ที่ประมาณ 90% หลังจากรับยาไปราว ๆ ส่องถึงสามชั่วโมง แต่โดยรวมเปอร์เซ็นต์ยับยั้งค่อนข้างคงที่ต่อเวลาในรอบยี่สิบสี่ชั่วโมง และยังค่อนข้างคงที่ต่อขนาดยาที่ทดสอบด้วย (2mg/kg, 10mg/kg, และ 30 mg/kg) ซึ่งทีมงานก็พอใจ.

ผลทดสอบกับหนูทดลองที่มีกลุ่มควบคุม (กลุ่มของหนูที่ไม่เป็นโรค) และกลุ่มที่หนูเป็นโรคเบาหวาน (Neonatally streptozotocin-induced diabetic rats คำย่อ N-STZ-1.5 rats) ก็ได้ผลที่ดี. ผลกับการทำงานของดีพีพีสี แสดงการลดลงของเปอร์เซ็นต์การทำงานของดีพีพีสีตามขนาดยาที่ใช้. ผลกับปริมาณของอินคริตินส์ แสดงการเพิ่มขึ้นของอินคริตินต่อเวลา ตามขนาดยาที่ใช้. ผลกับกลูโคสในกระแสเลือดต่อเวลา แสดงการลดลงของกลูโคสต่อเวลา ตามขนาดยาที่ใช้.

นอกจากการทดสอบการลดระดับน้ำตาลในเลือดแล้ว สำหรับโรคเบาหวาน ยังต้องทดสอบด้วยว่าจะไม่ทำให้เกิดอาการน้ำตาลต่ำ (hypoglycemia). นั่นคือ ต้องทดสอบว่า ยาจะไม่ปลดน้ำตาลในเลือดมากเกินไป. จากการทดลองกับหนูที่อดอาหาร และกินอลอกลิบตินเข้าไปที่ขนาดยาสูง (30mg/kg และ 100mg/kg) เปรียบเทียบกับกลุ่มควบคุม และเปรียบเทียบกับกลุ่มที่กินนาทีไกลินเดน์ ยาซึ่งทีมงานรู้ว่าให้ผลด้านนี้เมดี ทีมงานไม่พบว่า อลอกลิบตินทำให้ระดับน้ำตาลลดต่ำเกินไปหรือระดับอินซูลินสูงเกินไป.

การทดลองต่าง ๆ ข้างต้นเป็นการให้ยาครั้งเดียว การศึกษาสภาพทนทาน (robustness) ของยา จะทดสอบโดยให้ยาติดต่อกันเป็นเวลานาน ซึ่งผลที่ได้ แสดงการลดลงของการทำงานของดีพีพีสีตามขนาดยา การเพิ่มขึ้นของอินคริตินส์ตามขนาดยา เมื่อใช้ยาติดต่อกัน สำหรับหนูทดลองหลาย ๆ ชนิด. ผลสภาพทนทานเป็นไปด้วยดี.

การศึกษาเมแทบอลิซึมของยา พบร่วยว่าถูกย่อยสลายด้วยเอนไซม์ CYP-2D6 และ CYP-3A4 และยาไม่ผลกระแทกต่อการยับยั้งหรือกระตุ้นเอนไซม์ทั้งสองน้อยมาก. นอกจากนั้น ผลศึกษาแสดงการจับของยา กับโปรตีนในน้ำเลือดต่ำมาก และไม่พบปฏิกิริยาสัมพันธ์กับยาตัวอื่น เมื่อทดสอบกับยาเบาหวานตัวอื่น ๆ. ทีมงานลองใจกับผลทดสอบนี้.

การทดสอบ เพื่อวัดข้อมูลเกสัจลนาศาสตร์ และช่วงปริมาณออกฤทธิ์เป็นที่น่าพอใจ (ได้ค่า F เป็น 87% ในลิงที่ให้อโลกลิบตินขนาด 10mg/kg หนึ่งครั้ง ซึ่งค่า F เป็นสัดส่วนของยาที่ดูดซึมเข้าไปในกระแสเลือดต่ออยาที่กินเข้าไป). ค่าครึ่งอายุที่ 5.7 ชั่งโมง (ในลิง และค่าครึ่งอายุ เป็นดัชนีที่บอกเวลาที่ยาลดความเข้มข้นเหลือครึ่งหนึ่งจากความเข้มข้นสูงสุด) ซึ่งคริสโตเฟอร์รู้สึกว่า่น้อยไปนิดและอยากได้ที่แ配ดชั่งโมง เพื่อที่ผู้ป่วยจะสามารถกินยาครั้งเดียวต่อวันได้ แต่ก็หวังว่าค่าครึ่งอายุในมนุษย์จะมากขึ้นกว่านี้. เส้นทางการขับถ่ายยาออกจากร่างกายคือผ่านปัสสาวะและอุจจาระ.

การทดลองด้านความปลอดภัยของยา ซึ่งทีมงานของคริสโตเฟอร์ต้องทำการศึกษามากกว่าสิบการทดสอบ และใช้งบประมาณไปหลักสิบล้านдолลาร์ ผลที่ได้สรุปว่า ไม่พบความเป็นพิษต่อระบบประสาทกลาง ไม่พบความเป็นพิษต่อหัวใจ และหลอดเลือด ไม่พบความเป็นพิษกับระบบปอดและทางเดินหายใจ ไม่พบความพิษทางพันธุกรรม ไม่พบความเป็นพิษเมื่อใช้ต่อเนื่อง (กับยาขนาด 200mg/kg ในสุนัข โดยให้ติดต่อกันทุกวันเป็นเวลา 9 เดือน) ซึ่งจากผลที่ได้ทีมงานสามารถนำไปคำนวณความปลอดภัยในมนุษย์ ซึ่งผลที่ได้อือว่าปลอดภัยมาก.

หลังจากได้ผลที่น่าพอใจกับสัตว์ทดลองแล้ว ยานี้จะมีโอกาสได้ทดสอบทางคลินิกกับมนุษย์ได้ต่อไป ก่อนที่จะได้รับการพิจารณาเพื่อการรับรองขึ้นทะเบียนยา.

“Whatever you do may seem insignificant to you, but it is most important that you do it.”

---Mohandas Karamchand Gandhi

“อะไรก็ตามที่เราทำ มันอาจดูเล็กน้อยไม่สำคัญ แต่มันสำคัญที่เราทำมัน.”

—มหาตมา คานธี

แบบฝึกหัด 3.18

จากเรื่องการหยุดก่อนกำหนด ในหัวข้อ 3.4 จะเขียนโปรแกรม เพื่อสร้างข้อมูล และฝึกโครงข่ายประสาทเทียมสองชั้น ขนาด 100 หน่วยย่อย และทดลอง และเปรียบเทียบผลการทดลอง กับผลที่แสดงในรูปที่ 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 พร้อมแบ่งข้อมูลส่วนหนึ่งออกมาจากข้อมูลฝึก มาเป็นข้อมูลตรวจสอบ ทดลองฝึกแบบจำลอง ที่มีการหยุดก่อนกำหนด เปรียบเทียบผล สรุป และอภิปราย.

หมายเหตุ ตัวอย่างในรูปที่ 3.21 ใช้การกำหนดค่าหนักเริ่มต้น ที่ดัดแปลงมาจากวิธีเรียนวิดโดร์ เพื่อให้เห็นภาพชัดเจนขึ้น และลดเวลาในการฝึก. การใช้การกำหนดค่าหนักเริ่มต้นจากการสุ่มค่า อาจต้องทำการฝึกนานกว่าจำนวนสมัยฝึกที่เห็นในรูปที่ 3.21.

แบบฝึกหัด 3.19

จากเรื่องเล่นโค้งเรียนรู้ ในหัวข้อ 3.5 จะเขียนโปรแกรม เพื่อทดลองสร้างเส้นโค้งเรียนรู้. จงสร้างข้อมูลขึ้นมา 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 หน่วยช่อง, (ข) โครงข่ายประสาทเทียมสองชั้น ที่มี 8 หน่วยช่อง และ (ค) โครงข่ายประสาทเทียมสองชั้น ที่มี 64 หน่วยช่อง โดยดำเนินการฝึกและทดสอบแบบจำลอง 5 ครั้ง ครั้งแรกใช้

ข้อมูลฝึก 8 จุด ต่อมามาใช้ 16, 24, 32, และ 40 จุดตามลำดับ. วัดค่ารากที่สองของค่าเฉลี่ยค่าผิดพลาด กำลังสองของทั้งข้อมูลทดสอบ และข้อมูลฝึก. นำเสนอดูตัวอย่างรูปที่ 3.23 และ 3.24) อภิปรายผลที่ได้ และสรุป.

หมายเหตุ ให้ฝึกโครงข่ายประสาทเทียมให้สมบูรณ์ดีทุกครั้ง อาจใช้วิธีเกรียนวิดโดร์ ช่วยในการกำหนดค่าน้ำหนักเริ่มต้น เพื่อช่วยลดเวลาในการฝึกลงได้.

แบบฝึกหัด 3.20

จากวิธีคำนวณค่าเกรเดียนต์เชิงเลข ด้วย

$$\nabla_{\theta} E(\theta) = \begin{bmatrix} \frac{\partial E}{\partial \theta_1} \\ \frac{\partial E}{\partial \theta_2} \\ \vdots \\ \frac{\partial E}{\partial \theta_M} \end{bmatrix} \text{ และ } \frac{\partial E}{\partial \theta_i} \approx \frac{E\left(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i + \varepsilon \\ \theta_{i+1} \\ \vdots \end{bmatrix}\right) - E\left(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i \\ \theta_{i+1} \\ \vdots \end{bmatrix}\right)}{\varepsilon} \quad (3.51)$$

เมื่อ $E(\theta)$ คือฟังก์ชันจุดประสงค์ และ $\theta = \{\theta_1, \dots, \theta_M\}$ คือพารามิเตอร์ของแบบจำลอง ที่มีจำนวนพารามิเตอร์ทั้งหมด M ตัว.

จงตรวจสอบค่าเกรเดียนต์ที่คำนวณจากวิธีแพร่กระจายย้อนกลับ เปรียบเทียบกับค่าเกรเดียนต์ที่คำนวณจากการเชิงเลข โดยให้ใช้ค่า ε เป็น 1, 0.01 และ 0.0001 ตามลำดับ. จงสรุปและอภิปรายผล.

รายการ 3.22 แสดงโปรแกรมคำนวณเกรเดียนต์เชิงเลขแบบทางเดียว (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 นิยามดังที่ได้อภิปราย. ผลลัพธ์ `dE` เป็นไฟรอนดิกชันนารี ที่เก็บเกรเดียนต์ของค่าน้ำหนักและไบอัส ที่เรียกใช้ได้ เช่น `dE['weight1']` และ `dE['bias1']` คือ เกรเดียนต์ของค่าน้ำหนักและไบอัสของขั้นคำนวณที่หนึ่ง.

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                ngrad['weight%d'%i] = gradw/epsilon
25
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 อุเшиบะ

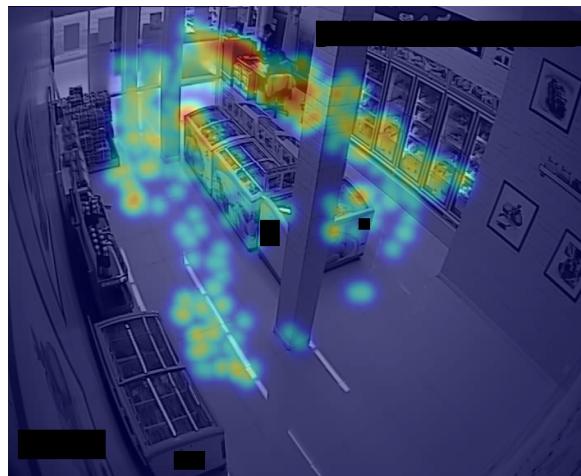
บทนี้ นำเสนอตัวอย่างการนำวิธีการเรียนรู้ของเครื่อง ไปประยุกต์ใช้กับงานการรู้จำรูปแบบ รวมถึง วิธีการเรียนรู้ของเครื่องแบบต่าง ๆ ที่หลากหลาย. หัวข้อ 4.1 นำเสนอตัวอย่างระบบวิเคราะห์พฤติกรรมลูกค้า ที่ใช้แบบจำลองการเรียนรู้ของเครื่องประกอบอยู่ในระบบ ทั้งในส่วนการตรวจสอบจับตำแหน่งลูกค้า และการแสดงผล. หัวข้อ 4.1 อภิราย เทคนิคหลาย ๆ อย่างที่ใช้ประกอบกับวิธีการเรียนรู้ของเครื่อง เพื่อสามารถใช้กับงานการรู้จำรูปแบบได้อย่างมีประสิทธิผล. หัวข้อ 4.2 อภิราย ซัพพอร์ตเวก เตอร์แมชชีน ซึ่งเป็นหนึ่งในวิธีการเรียนรู้ของเครื่องอีกชนิดที่ได้รับความนิยมอย่างมาก. ซัพพอร์ตเวกเตอร์แมชชีน ออกแบบและพัฒนาโดยอาศัยศาสตร์การหาค่าดีที่สุด มีทฤษฎีรองรับมั่นคง และมีเบื้องหลังการออกแบบที่ сложสลวย.

4.1 การวิเคราะห์พฤติกรรมลูกค้า

เนื้อหาในหัวข้อนี้ ได้รับอิทธิพลหลัก ๆ จากโครงการวิจัยการวิเคราะห์พฤติกรรมลูกค้าผ่านข้อมูลวิดีโอ จากกล้องวงจรปิด[106].

การวิเคราะห์พฤติกรรมลูกค้าจากภาพวิดีโอ

ข้อมูลวิดีโอด้วยกล้องวงจรปิดในร้านค้าปลีก นอกจากใช้เพื่อเหตุผลด้านความปลอดภัยแล้ว ข้อมูลวิดีโอยังสามารถนำมาใช้ เพื่อการวิเคราะห์พฤติกรรมการจับจ่ายของลูกค้าที่เข้ามาภายในร้านได้. การเข้า



รูปที่ 4.1 แผนภาพความร้อน (รูปจาก งานศึกษาพฤติกรรมลูกค้าจากข้อมูลวิดีโอ[106]). ตัวอย่างภาพสรุป ซึ่งเป็นภาพช้อนของภาพถ่ายแสดงผังบริเวณร้านจากกล้องวงจรปิด และช้อนทับด้วยแผนภาพความร้อนที่มีลักษณะกึ่งโปร่งใส. แผนภาพความร้อนแสดงบริเวณที่พบลูกค้าบ่อยที่สุดด้วยสีแดง และสีโภนที่เย็นลงสื่อถึงความถี่ที่พบลูกค้าลดลง และความถี่ต่ำที่สุดแทนด้วยสีที่เย็นที่สุดคือสีน้ำเงิน (ภาพนี้เป็นภาพสี และในภาพได้ทำการปิดบังตราสัญลักษณ์ของร้านค้าไว้)

ให้พฤติกรรมของลูกค้าสามารถนำมาช่วยเพิ่มโอกาสทางธุรกิจได้ เช่น ความเข้าใจบริเวณและเวลาที่ลูกค้าใช้ขณะเข้ามาภายในร้าน สามารถใช้ประกอบการตัดสินใจ สำหรับการจัดวางสินค้า หรือการจัดกิจกรรมส่งเสริมการตลาดได้.

ระบบวิเคราะห์พฤติกรรมอัตโนมัติ อาศัยข้อมูลวิดีโอที่ถ่ายจากกล้องวงจรปิด เพื่อตรวจหาตำแหน่งของลูกค้า เมื่อลูกค้าเข้ามาใช้บริการภายในร้านค้า. ตำแหน่งที่ได้ สามารถนำไปทำแผนภาพสรุปที่อาจช่วยให้เข้าใจรูปแบบการเข้าใช้พื้นที่ในบริเวณร้านได้ดีขึ้น และช่วยประกอบการตัดสินใจด้านการตลาดได้.

ข้อมูลจากการกล้องวงจรปิดภายในร้านค้าปลีก ถูกนำมาใช้เพื่อประมวลผล และตรวจหาตำแหน่งของลูกค้า แล้วจัดทำแผนภาพสรุป. รูปที่ 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*). เช่นเดียวกับงานของเบนเนนลันและคณะ[11] ที่ศึกษาการตรวจหาและระบุตำแหน่งคนเดินเท้า ซึ่งเป็นงานในลักษณะใกล้เคียงกัน งานการตรวจหาและระบุตำแหน่งลูกค้า้นี้ ดำเนินการโดย การแปลงจากวิดีโอมาเป็นชุดลำดับของภาพ และวิจัยใช้วิธีการประมวลผลภาพ เพื่อตรวจหาและระบุตำแหน่งของลูกค้า.

การระบุตำแหน่งลูกค้าอาจทำได้หลายวิธี เช่น (ก) แนวทางวิธีลบจากพื้นหลัง และ (ข) แนวทางวิธีตรวจหาภาพวัตถุ.

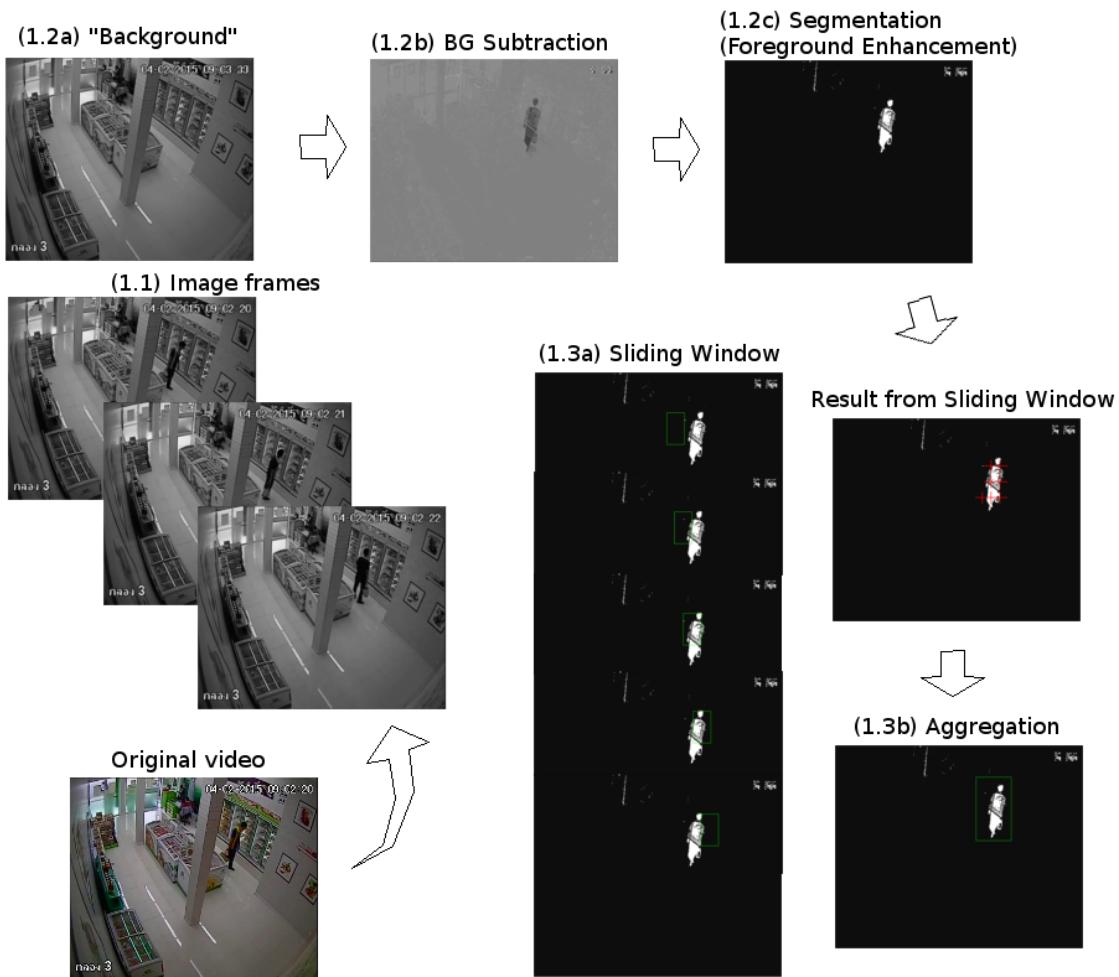
แนวทางวิธีลบจากพื้นหลัง

แนวทางวิธีลบจากพื้นหลัง ไม่ได้ใช้เทคนิคใดจากศาสตร์การเรียนรู้ของเครื่อง แต่ใช้อาศัยเทคนิคการประมวลผลภาพ และอาศัยลักษณะเฉพาะของข้อมูล. จากการสังเกตลักษณะเฉพาะของข้อมูล พื้นหลังของภาพวิดีโอก่อค่อนข้างคงที่ ส่วนใหญ่ สิ่งที่เคลื่อนที่ในภาพมักจะเป็นคน ที่รวมทั้งลูกค้าและพนักงาน. แนวทางวิธีลบจากพื้นหลัง อาศัยลักษณะเฉพาะของข้อมูลนี้เข้ามาช่วย.

แนวทางวิธีลบจากพื้นหลัง (*Background Subtraction*) แสดงดังรูปที่ 4.2 ซึ่งเป็นแผนภูมิลำดับกระบวนการ. ข้อมูลวิดีโอ (ภาพช้ายล่าง) จะถูกแปลงเป็นเฟรมภาพ (ขั้นตอน 1.1). หลังจากนั้น นำแต่ละภาพที่ได้ไปลบออกจากภาพพื้นหลัง (*background* หรือภาพในบริเวณร้านที่ไม่มีลูกค้าอยู่) รวมถึงการปรับปรุงเพื่อยายความต่างให้ชัดเจนขึ้น (ขั้นตอนที่ 1.2a – 1.2c). สุดท้าย ทำการค้นหาและระบุตำแหน่งลูกค้าในภาพ (ขั้นตอนที่ 1.3a – 1.3b).

ขั้นตอนที่ 1.1. การแปลงจากวิดีโอไฟล์เป็นภาพ มีกระบวนการทางเทคนิคที่มีรายละเอียดมาก รวมไปถึงขั้นตอนต่าง ๆ ตามมาตรฐานของชนิดข้อมูลวิดีโอ แต่ในทางปฏิบัติมีเครื่องมือที่ช่วยทำงานเหล่านี้ได้มากmany หนึ่งในเครื่องมือที่นิยมใช้ ก็ เช่น โอเพนซีวี (*OpenCV*¹) ซึ่งเป็นไลบรารีรหัสเปิด (*open-source library*) สำหรับงานด้านทัศนคณภาพ (*computer vision*). ขั้นตอนที่ 1.2 การปรับปรุงภาพ เพื่อให้ตำแหน่งของลูกค้าเด่นชัดขึ้น เพื่อช่วยให้การตรวจหาตำแหน่งของลูกค้าในขั้นตอนต่อไปทำได้ง่าย. ขั้นตอนย่อยในการปรับปรุงภาพนี้ ได้แก่ เทคนิควิธีลบพื้นหลัง ที่นำภาพพื้นหลังไปลบออกจากภาพที่ต้องการตรวจหาตำแหน่งลูกค้า. สิ่งที่ต่างจากพื้นหลังก็จะปรากฏเด่นขึ้น (1.2a และ 1.2b) และ เทคนิควิธีแยกส่วนโดยการกำหนดระดับค่าชีดแบ่ง (*segmentation by thresholding*) ก็สามารถนำมาใช้ เพื่อแยกจากหน้า (ที่อาจเป็นลูกค้า) ออกจากหลัง. เทคนิควิธีแยกส่วนโดยการกำหนดระดับค่าชีดแบ่ง คือการตั้งค่าระดับชีดแบ่งขึ้นมาหนึ่งค่า โดย หากค่าความเข้มของพิกเซลมากกว่าระดับนั้น ก็จะปรับเพิ่มค่าความเข้มของพิกเซลนั้นไปจนมีค่ามากที่สุด (สว่างเต็มที่) ไม่เช่นนั้น ก็ปรับลดค่าลงเป็นศูนย์ (มืดเต็มที่). ดังนั้น ภาพจะถูกแยกเป็นส่วนสว่างและมืดอย่างชัดเจน. ขั้น

¹<http://opencv.org/>



รูปที่ 4.2 วิธีการลบฉากหลัง (รูปจาก งานศึกษาพัฒนาระบบลูกค้าจากข้อมูลวิดีโอ[228]). แผนภาพแสดงขั้นตอนต่าง ๆ ใน การตรวจหาตำแหน่งลูกค้าจากข้อมูลวิดีโอด้วยแนววิธีการลบฉากหลัง

ตอน 1.3 เป็นการนำภาพที่ปรับปรุงความต่างระหว่างฉากหน้าและฉากหลัง ไปตรวจหาตำแหน่งของ ลูกค้า ซึ่งใช้เทคนิคการตรวจหาแบบหน้าต่างเลื่อน (*sliding window detection*). เทคนิคการตรวจหาแบบหน้าต่างเลื่อน เป็นหนึ่งในวิธีที่นิยมใช้สำหรับงานการตรวจหาภาพวัตถุ. เทคนิคการตรวจหาแบบหน้าต่างเลื่อน เป็นเทคนิคการเลือกส่วนภาพ โดยส่วนภาพ(ในขนาดที่พอเหมาะสมแก่การตรวจสอบ ว่าเป็นวัตถุที่ต้องการหรือไม่) จะถูกเลือกขึ้นมาจากการที่ส่องสัย. การเลือกจะเริ่มที่มุมด้านหนึ่งของภาพ และขยายไปเรื่อยๆ จนครอบคลุมทั้งภาพ. ส่วนภาพขนาดเล็กที่ถูกเลือกออกมานั้นจะถูกตรวจสอบว่ามีวัตถุที่ค้นหาอยู่ในส่วนภาพนั้นหรือไม่ (ขั้นตอน 1.3a). เมื่อได้ผลการตรวจสอบส่วนภาพขนาดเล็กจากตำแหน่งต่างๆ แล้ว ส่วนภาพต่างๆ ที่ได้ผลการตรวจเป็นบางที่มีตำแหน่งใกล้ๆ กัน จะถูกรวมกันตามเกณฑ์การรวมที่กำหนด และตำแหน่งของการรวมนั้นจะถูกบันทึกเป็นตำแหน่งของลูกค้าทั้งตัว (ขั้นตอน 1.3b). เมื่อถึงขั้นตอนนี้ ระบบก็สามารถระบุได้แล้วว่า ลูกค้าอยู่ที่ตำแหน่งใดในภาพ.

แนวทางของวิธีตรวจหาวัตถุ

กลไกจริง ๆ ของวิธีการลับจากหลัง คือ การตรวจหาตำแหน่งของวัตถุที่ต่างจากหลัง ซึ่งสมมติฐานก็คือน่าจะเป็นลูกค้า. วิธีการลับจากหลังมีข้อดีคือ สามารถทำได้ง่ายและรวดเร็ว (ในกรณีที่มีภาพจากหลังแล้ว). แต่ข้อเสีย คือความสามารถดังกล่าวไม่สามารถขยายไปใช้ในการแยกวัตถุอื่นได้ เช่น ยกที่จะแยกคนกับรถออกจากกันได้ และไม่สามารถนำไปใช้ในกรณีที่จากหลังมีการเปลี่ยนแปลงสูงได้ เช่น ไม่สามารถนำไปใช้ตรวจจับภาพคนเดินถนน ที่จากหลังเป็นสภาพการจราจรได้ และก็ไม่สามารถใช้ในกรณีคนมีการขยับน้อย เช่น ในสวนที่มีคนนั่งอ่านหนังสือ ฝึกสมาธิ หรืออนหลับอยู่.

อีกแนวทางหนึ่ง คือ แนวทางวิธีตรวจหาวัตถุ (**object detection**) เป็นภารกิจการหาตำแหน่งของวัตถุในภาพ หากในภาพมีวัตถุปรากฏอยู่. แนวทางของวิธีตรวจหาวัตถุมีพื้นฐานมาจากสมมติฐานทางสถิติ เช่น รูปทรงของคนแม้จะมีความหลากหลาย แต่ก็มีรูปแบบ และลักษณะร่วมกันอยู่มาก และวิธีการคือ การหาลักษณะร่วมเหล่านั้นของภาพ และใช้มันช่วยในการจำแนกระหว่างคนกับสิ่งอื่น ๆ ที่ไม่ใช่คน. รูปที่ 4.3 แสดงขั้นตอนอย่าง ๆ ในการตรวจหาตำแหน่งของลูกค้า ด้วยวิธีการตรวจหาวัตถุ โดยเริ่มตั้งแต่

1. การแปลงข้อมูลวิดีโอออกมาเป็นข้อมูลภาพราย ๆ ภาพ (frame capture) เพียงเท่าขั้นตอน 1.1 ของรูปที่ 4.2 ของแนวทางการลับจากหลัง. นั่นคือ การปรับปัญหาจากการทำงานกับข้อมูลวิดีโอ เป็นปัญหาเสมือนที่ทำงานกับภาพแทน.
2. การย่อและขยายภาพให้อยู่ในหลาย ๆ ขนาด (scaling) เพื่อให้วัตถุที่อยู่ห่างกล้องในระยะต่าง ๆ มีโอกาสที่จะถูกตรวจพบใกล้เคียงกัน.
3. การเลือกส่วนภาพ (window sampling). แนวทางนี้ ก็ยังคงเลือกใช้เทคนิคน้ำต่างเลื่อน. นั่นคือ ตีกรอบปัญหาการตรวจจับภาพวัตถุ เป็นปัญหาการจำแนกค่าทวิภาค โดยใช้การเลือกส่วนภาพออกมานี้เพื่อเป็นอินพุตของแบบจำลองจำแนก (ในขั้นตอนต่อไป).
4. การจำแนกส่วนภาพว่ามีภาพคนอยู่หรือไม่ (window classification). ภาพขนาดเท่า ๆ กัน (ได้จากการสูมภาพด้วยวิธีหน้าต่างเลื่อน) จะถูกผ่านเข้าแบบจำลองจำแนกข้อมูล โดยแบบจำลองจะทำหน้าที่คำนวณว่าภาพอินพุตนั้นมีภาพของวัตถุที่ต้องการ (ในที่นี้คือ ภาพคน) หรือไม่.
5. การลดผลการตรวจหาที่ซ้ำซ้อน (redundancy removal). การสูมด้วยวิธีหน้าต่างเลื่อน มักทำในลักษณะที่มีการซ้อนทับกัน เพื่อป้องกันการตัดกรอบที่กลางวัตถุ แต่การสูมในลักษณะนี้ก็อาจทำให้ การสูมในตำแหน่งใกล้เคียงกัน ตรวจพบวัตถุเดียวกันได้. ดังนั้น หลังการตรวจพบวัตถุแล้ว จึงต้องมีการทำการลดการซ้ำซ้อนลง. หลังจากขั้นตอนการลดการซ้ำซ้อน เราจะสามารถระบุตำแหน่งของวัตถุที่ตรวจพบได้.

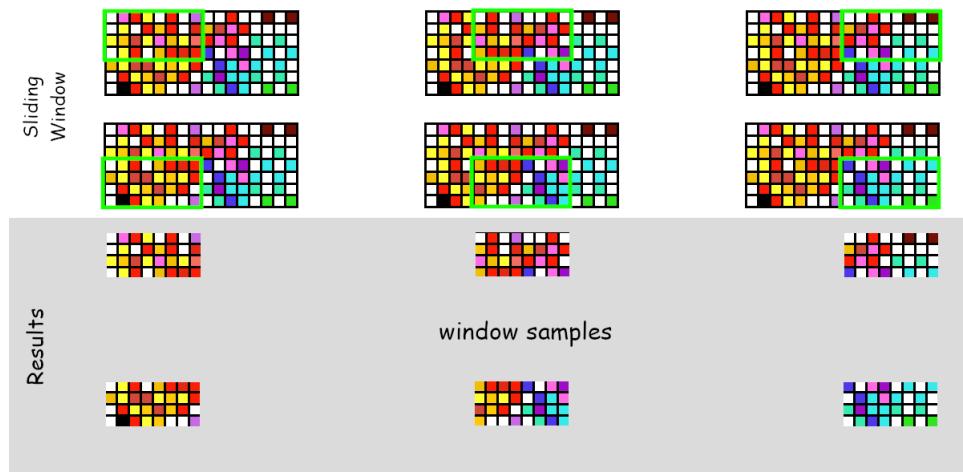


รูปที่ 4.3 เทคนิคหน้าต่างเลื่อน (รูปจาก งานศึกษาพฤติกรรมลูกค้าจากข้อมูลวิดีโอ[106]). แผนภาพแสดงขั้นตอนต่าง ๆ ใน การตรวจหาตำแหน่งลูกค้าจากข้อมูลวิดีโอด้วยแนววิธีการตรวจจับภาพวัตถุ

จากนั้น ผลการตรวจหาที่ได้จะนำไปสรุป และแสดงผลด้วยวิธีแพนที่ความร้อน (hot zone analysis ซึ่งเป็นขั้นตอนที่ 6 ในรูปที่ 4.3). ขั้นตอนที่ 2 ถึง 5 คือขั้นตอนที่ต่างจากแนวทางวิธีลับจากหลัง (เปรียบเทียบกับ 1.2a ถึง 1.3b ในรูปที่ 4.2) ถึงแม้วิธีทางจะใช้กลไกของการเลือกส่วนภาพด้วยวิธีหน้าต่างเลื่อนก็ตาม.

การทำเทคนิคหน้าต่างเลื่อน. วิธีหน้าต่างเลื่อน[206] เป็นวิธีการเลือกส่วนภาพขนาดที่กำหนดจาก ข้อมูลภาพใหญ่ โดย การเลือกส่วนภาพจะเลือกทั้งหมดจากทุกบริเวณในภาพใหญ่ โดยอาจเริ่มจากมุม ซ้ายบนของภาพใหญ่ เลือกส่วนภาพอ กมา แล้วย้ายไปทางขวา และทำเช่นนี้ไปจนสุดปลายด้านขวา แล้วจึงย้ายลงล่างและไปเริ่มจากซ้ายสุด และทำลักษณะเช่นนี้อีก จนครอบคลุมบริเวณทั้งภาพใหญ่. ลำดับของภาพที่เลือกอ กมา จะดูคล้ายกับลำดับของภาพที่มีองค์ประกอบหน้าต่างที่เลื่อนไปตามตำแหน่งต่าง ๆ ของภาพใหญ่ ดังนั้นเทคนิคนี้จึงเรียกว่า เทคนิคหน้าต่างเลื่อน. ขนาดของหน้าต่าง (window size) ซึ่ง คือขนาดของส่วนภาพที่เลือก และขนาดของการย้ายหน้าต่าง ที่มักเรียกว่า **ขนาดย้ายเลื่อน (stride)** ซึ่งเป็นจำนวนพิกเซลของการย้ายการเลือกส่วนภาพแต่ละครั้ง เป็นอภิธานพารามิเตอร์ของวิธีหน้าต่าง เลื่อน.

รูปที่ 4.4 แสดงตัวอย่างการทำงานของวิธีหน้าต่างเลื่อน. ในตัวอย่าง ภาพใหญ่ขนาด 16 พิกเซล ครอบหน้าต่างเลือกกว้าง 8 พิกเซล และขนาดย้ายเลื่อนแนวนอนเป็น 4 พิกเซล ดังนั้นจึงสามารถ ย้ายได้ 3 ตำแหน่งในแนวนอน (ได้แก่ เริ่มต้นที่พิกเซล 0, ย้ายไปพิกเซล 4, และย้ายไปพิกเซล 8). จำนวนตำแหน่งของหน้าต่างในแนวนอนและแนวตั้งสามารถเขียนเป็นการคำนวณทั่วไปได้ดังนี้ กำหนดให้เมทริกซ์ $\mathbf{F} = [f_{m,n}]$, $m = 0, \dots, C-1$ และ $n = 0, \dots, R-1$ แทนภาพใหญ่ขนาด $C \times R$ และ $f_{m,n} \in \mathbb{I}$ เป็นค่าความเข้มของพิกเซลที่ตำแหน่ง (m, n) และให้ $\mathbf{W}_{ij} \in \mathbb{I}^{A \times B}$ แทนส่วนภาพ ขนาด $A \times B$ ของด้านหน้าต่าง (i, j) . หากขนาดย้ายเลื่อนตามแนวนอนและตั้งเป็น (a, b) และ



รูปที่ 4.4 ตัวอย่างการเลือกส่วนภาพด้วยวิธีหน้าต่างเลื่อน. ภาพใหญ่และกรอบหน้าต่างเลือก แสดงในส่วนบน (พื้นหลังสีขาว). กรอบหน้าต่างเลือก แสดงด้วยกรอบสีเขียว. ส่วนภาพที่เลือกมาจากการขยับหน้าต่างแต่ละครั้ง แสดงในส่วนล่าง (พื้นหลังสีเข้ม). ในตัวอย่าง ภาพใหญ่ขนาด 16×7 (กว้าง คูณ สูง). ขนาดกรอบหน้าต่างเลือกเป็น 8×4 และขนาดขยับเลื่อน เป็น 4×3 .

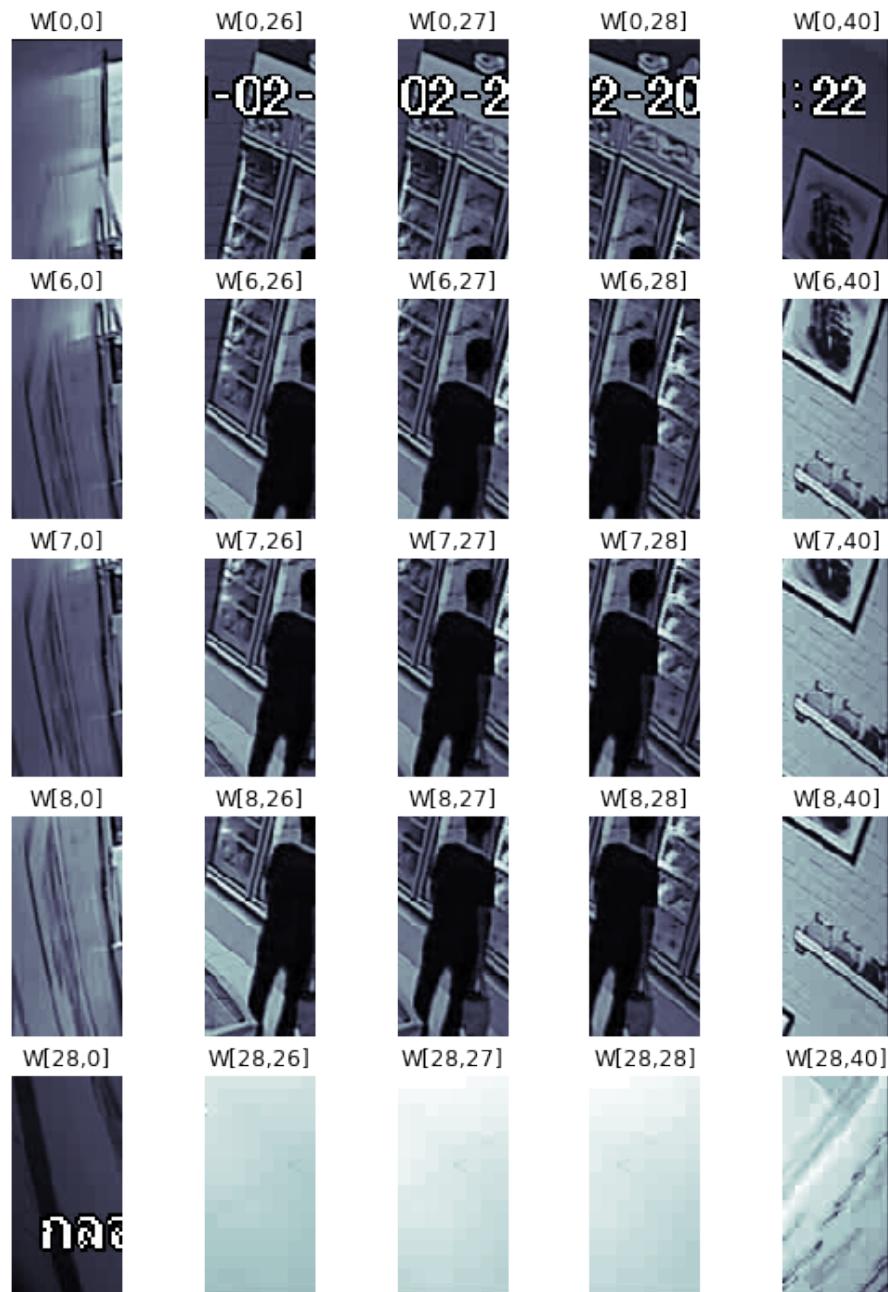
วิธีหน้าต่างเลื่อน เป็นสมือนฟังก์ชันแปลง $S : \mathbf{F} \mapsto \{\mathbf{W}_{ij}\}$ สำหรับดัชนี $i = 0, \dots, \lfloor \frac{C-A}{a} \rfloor$ และ ดัชนี $j = 0, \dots, \lfloor \frac{R-B}{b} \rfloor$. แต่ละส่วนภาพ $\mathbf{W}_{ij} = [w_{p,q}(i,j)]$ เมื่อ $p = 0, \dots, A-1$ และ $q = 0, \dots, B-1$ เป็นเมทริกซ์โดย $w_{p,q}(i,j) = f_{a \cdot i + p, b \cdot j + q}$.

รูปที่ 4.5 แสดงตัวอย่างการใช้วิธีหน้าต่างเลื่อนกับงานการตรวจจับภาพเป้าหมาย. ผลลัพธ์ที่ได้คือ ส่วนภาพต่าง ๆ. ส่วนภาพต่าง ๆ จะถูกนำไปตรวจสอบ ด้วยของส่วนภาพที่พบว่ามีเป้าหมายอยู่ จะถูกนำไปใช้ระบุตำแหน่งของเป้าหมายในภาพใหญ่ (หากพบเป้าหมายในภาพ). จากอภินานพารามิเตอร์ ที่ใช้ (หน้าต่างขนาด 128×64 ใช้ขนาดขยับเลื่อนเป็น 16) และขนาดของภาพใหญ่ (576×704) 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} แบบจำลองจำแนกค่า ทวิภาคสามารถใช้เพื่อทำนายว่าในส่วนภาพมีวัตถุเป้าหมายอยู่หรือไม่. นั่นคือ แบบจำลองจำแนกค่า ทวิภาค เป็นสมือนฟังก์ชันแปลง $f : \mathbf{W}_{ij} \mapsto y_{ij}$ เมื่อ y_{ij} คือค่าทวิภาคที่ทำนาย ซึ่งในกรณีตัวอย่างนี้ นิยามเป็น $+1$ (มีเป้าหมายที่ค้นหาอยู่) หรือ -1 (ไม่มีเป้าหมายที่ค้นหาอยู่). หาก $y_{ij} = 1$ นั่นหมาย ถึง ส่วนภาพ \mathbf{W}_{ij} มีเป้าหมายอยู่ และตำแหน่งของเป้าหมาย ก็คือตำแหน่งต่าง ๆ ที่ \mathbf{W}_{ij} ครอบคลุม.



รูปที่ 4.5 วิธีหน้าต่างเลื่อนกับงานการตรวจจับภาพเป้าหมาย (รูปจาก งานศึกษาพัฒนาระบบลูกค้าจากข้อมูลวิดีโอ[228]). ภาพช้ายแอบบันสุด แสดงส่วนภาพแรก ($W[0,0]$) ที่วิธีหน้าต่างเลื่อนเริ่มต้น. และภาพขาวແවล่างสุด แสดงส่วนภาพสุดท้าย ($W[28,40]$) ที่วิธีหน้าต่างเลื่อนเลือกออกมา เมื่อภาพใหญ่มีขนาด 576×704 และหน้าต่างขนาด 128×64 ใช้ขนาดขยายบล็อกเลื่อนเป็น 16.

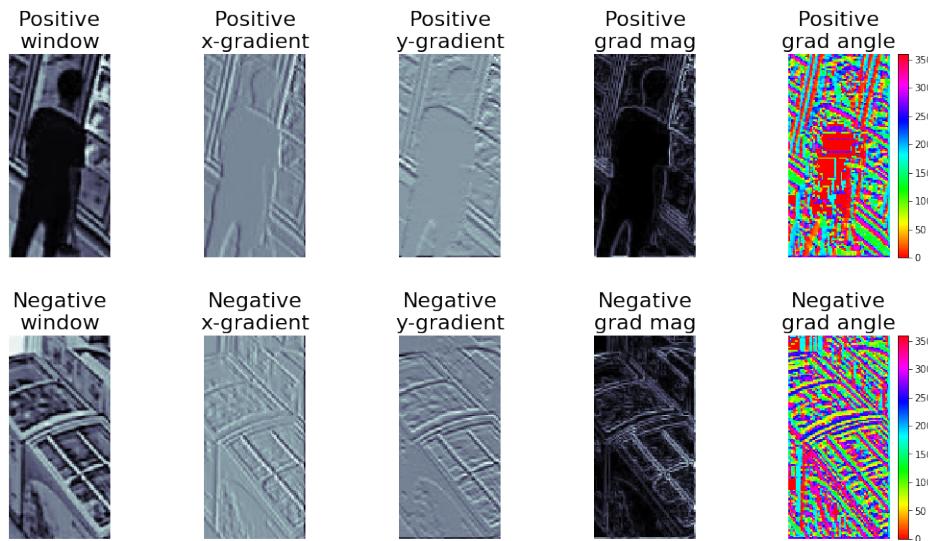
การแปลงจากส่วนภาพเป็นค่าทวิภาค ในตัวอย่างนี้ จะดำเนินเป็นสองขั้นตอน ได้แก่ (1) การแปลงส่วนภาพ ที่มักมีจำนวนมิติสูงมาก เป็นลักษณะสำคัญ ที่มีจำนวนมิติน้อยลงและเกี่ยวข้องการจำแนกเป้าหมายจากสิ่งอื่น ๆ และ (2) การแปลงลักษณะสำคัญที่ได้จากขั้นตอนแรกไปเป็นค่าทวิภาคที่ต้องการทำนาย. ลักษณะสำคัญ เป็นตัวแทนของอินพุตตันฉบับในแบบที่ช่วยให้ภาระกิจเป้าหมายดำเนินการได้ง่ายขึ้น. การเทคนิคที่สำคัญต่าง ๆ ในงานการตรวจจับภาพวัตถุ ล้วนเกี่ยวข้องโดยตรงกับการทำลักษณะสำคัญแทนของอินพุตตันฉบับ ไม่ว่าจะเป็น ลักษณะhaar (Haar features[206]) หรือ แผนภูมิแห่งของทิศทางเกรเดียนต์ (Histogram of Oriented Gradient) หรือ ถุงของทศนะถ้อยคำ (Bag of Visual Words[67]) เป็นต้น. ลักษณะสำคัญที่พัฒนาขึ้นมาในยุคหลัง ๆ อาจสร้างขึ้นจากลักษณะสำคัญที่พัฒนามาก่อนแล้ว เช่น แบบจำลองแปลงรูป (deformable model[68]) ที่ใช้แผนภูมิแห่งของทิศทางเกรเดียนต์ เป็นพื้นฐาน. แม้แต่แนวทางการเรียนรู้เชิงลึก (บท 5) ที่โดยทั่วไปแล้ว ไม่จำเป็นต้องดำเนินการแปลงเป็นสองขั้นตอน นั่นคือ ไม่ต้องเตรียมลักษณะสำคัญให้แบบจำลอง และสามารถรับอินพุตเป็นภาพตันฉบับโดยตรง ก็มีการใช้ลักษณะสำคัญ เพียงแต่แบบจำลองทำการสร้างลักษณะสำคัญขึ้นได้เองจากข้อมูลจำนวนมากที่ใช้ฝึก.

ตัวอย่างนี้ดำเนินการจำแนกค่าทวิภาคของส่วนภาพโดย (1) ส่วนภาพ \mathbf{W}_{ij} จะถูกแปลงเป็นเวกเตอร์ลักษณะสำคัญ \mathbf{X}_{ij} และจากนั้น (2) ลักษณะสำคัญ \mathbf{X}_{ij} จะถูกแปลงเป็นฉลากทำนาย $y_{ij} \in \{-1, +1\}$. ตัวอย่างนี้แสดงการใช้แผนภูมิแห่งของทิศทางเกรเดียนต์[49] เป็นลักษณะสำคัญ และใช้ชัฟฟอร์ดเวกเตอร์แมชชีน[43] เป็นแบบจำลองทำนาย.

หมายเหตุ แนวทางนี้ เพียงตรวจจับตำแหน่งของบุคคลในภาพ และไม่ได้มีการจำแนกแยกแยะมโนคติระดับสูง ว่าบุคคลในภาพเป็นลูกค้าจริง ๆ หรือเป็นพนักงานร้าน. เพื่อจะวิเคราะห์ในมโนคติระดับสูงตั้งกล่าว รูปแบบของเครื่องแบบ และเส้นทางการเคลื่อนที่ อาจนำมาใช้ประกอบได้.

การสกัดลักษณะสำคัญ. หนึ่งในลักษณะสำคัญที่นิยมใช้ในงานคอมพิวเตอร์วิทัศน์ คือ แผนภูมิแห่งของทิศทางเกรเดียนต์ (Histogram of Oriented Gradients[49]) ที่มักย่อเป็น เอชโอจี (HOG). เอชโอจี เป็นฟังก์ชันแปลง $H : \mathbf{W} \mapsto \mathbf{x}$ เมื่อ $\mathbf{W} \in \mathbb{I}^{A \times B}$ เป็นเมทริกซ์ของค่าความเข้มพิกเซล และ $\mathbf{x} \in \mathbb{R}^D$ เป็นเวกเตอร์ค่าลักษณะสำคัญเอชโอจี. โดยทั่วไปแล้ว ขนาดของเวกเตอร์ \mathbf{x} จะเล็กกว่าขนาดของ \mathbf{W} มา (นั่นคือ $D << A \times B$). สมมติฐานของเอชโอจี คือการแจกแจงพิกเซลเกรเดียนต์ของภาพสามารถเป็นนัยที่บ่งชี้รูปร่างและสามารถใช้ระบุเป้าหมายได้.

เอชโอจีเริ่มด้วยการคำนวณพิกเซลเกรเดียนต์ \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 $f_{rc} \equiv 0$ เมื่อ r หรือ c เกินขอบเขตภาพ ($r > R$ หรือ $c > C$). พิกเซลเกรเดียนต์ สามารถเขียนในรูปขนาดและมุมได้ ด้วยสัญกรณ์ $\mathbf{g}_{rc} = m_{rc} \angle \theta_{rc}$ โดยขนาด $m_{rc} = \sqrt{g_x^2(r, c) + g_y^2(r, c)}$



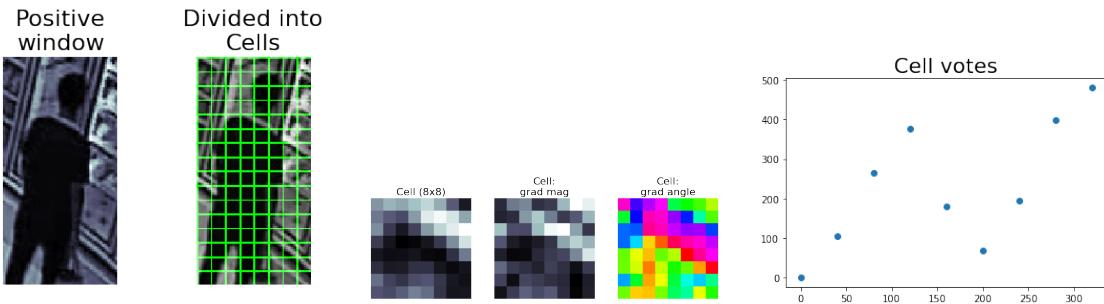
รูปที่ 4.6 ตัวอย่างพิกเซลเกรเดียนต์ของภาพ (รูปจาก งานศึกษาพัฒนกรรมลูกค้าจากข้อมูลวิดีโอ[228]). แคว้น แสดงตัวอย่างเมื่อส่วนภาพมีเป้าหมายอยู่. แคว้น แสดงตัวอย่างเมื่อส่วนภาพไม่มีเป้าหมายอยู่. ภาพชัยสุดแสดงส่วนภาพที่ในสเกลเท่า. ภาพที่สองและสามจากชัย แสดงพิกเซลเกรเดียนต์ในแนวอนและแนวตั้งตามลำดับ. ถัดอีกสองภาพ แสดงขนาด และมุม ของพิกเซลเกรเดียนต์ ตามลำดับ. สำหรับภาพแสดงขนาดเกรเดียนต์ สีขาวแทนขนาดที่มีค่ามาก และสีดำ แทนขนาดที่มีค่าน้อย. สำหรับภาพแสดงมุม สีแทนองศาของมุม ตามที่ระบุด้วยแบบสีด้านข้าง. สีที่ใช้สำหรับแสดงมุม ใช้ระบบสีลักษณะวุ้นจักร เนื่องจาก 360 องศา เป็นทิศทางเดียวกับ 0 องศา.

และมุม² $\theta_{rc} = \arctan \frac{g_y(r,c)}{g_x(r,c)}$. รูปที่ 4.6 แสดงค่าพิกเซลเกรเดียนต์ของส่วนภาพตัวอย่างที่มีเป้าหมาย และที่ไม่มีเป้าหมายอยู่.

จากนั้น เอชโอลจีดำเนินการโดยแบ่งส่วนภาพ \mathbf{W} เป็นส่วนย่อย ๆ และเรียกแต่ละส่วนย่อยว่า เชลล์ (cell). ในแต่ละเชลล์ เอชโอลจีจัดทำข้อมูล โดยจินตนาการเป็นเหมือนการทำแผนภูมิแท่ง โดยแต่ละแท่ง แทนค่าขนาดของเกรเดียนต์ในแต่ละทิศทาง (ทิศทางที่ใกล้เคียงกันจะถูกรวบอยู่ในแท่งเดียวกัน) และความสูงของแต่ละแท่ง เรียกว่า โหวต (vote) คำนวนจากผลรวมขนาดของเกรเดียนต์ ในทิศทางของแท่งนั้น ๆ. ขึ้นตอนสุดท้าย เชลล์ต่าง ๆ ที่จะถูกรวบกันเป็นบล็อก (block) ในลักษณะซ้อนทับกัน และค่าโหวตจากเชลล์ในบล็อกจะถูกน้อมร์โนร์มาลайซ์ภายในบล็อก. ค่าลักษณะสำคัญของเอชโอลจี คือค่าโหวตที่ถูกน้อมร์โนร์มาลайซ์แล้วจากบล็อกต่าง ๆ. นั่นคือ ขึ้นตอนแรก คำนวนโหวตของเชลล์จากส่วนภาพ \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} เป็นเวกเตอร์ของเชลล์ โหวต ที่เชลล์มีขนาด $h_v \times w_v$.

หากเลือกจำนวนทิศทางของแผนภูมิแท่งเป็น K ทิศทาง เชลล์ $\mathbf{v}_{ij} \in \mathbb{R}^K$ จะมีส่วนประกอบที่ k^{th} ของเชลล์ ที่เขียนเป็นสัญกรณ์ $v_k(i, j)$ และสามารถคำนวณค่าได้จากผลรวมของขนาดของเกรเดียนต์

² 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 θ_{rc} จะไม่มีความหมาย ถ้า $g_x(r,c) = 0$ และ $g_y(r,c) = 0$.

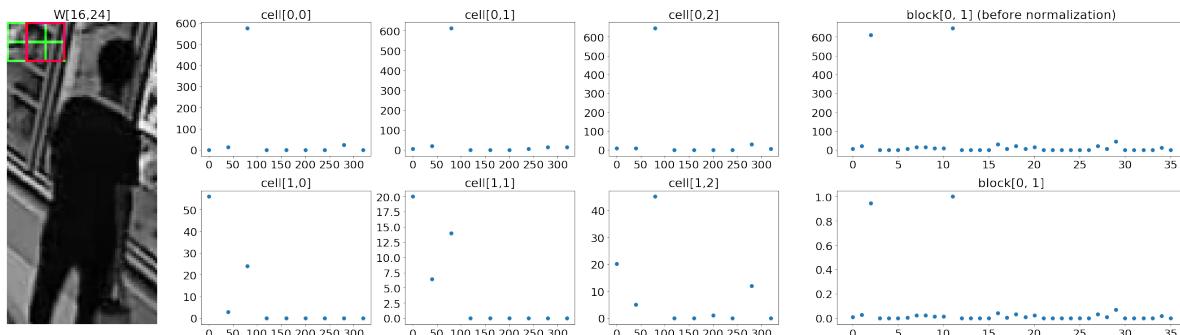


รูปที่ 4.7 ตัวอย่างแสดงการทำเอชโอลีด. ภาพซ้ายสุดแสดงตัวอย่างส่วนภาพ. ภาพที่สองจากซ้ายแสดงตัวอย่างส่วนภาพ พื้นที่ของเขตของแต่ละเซลล์ ซึ่งแสดงด้วยเส้นสีเขียว. เส้นสีเขียวในภาพทำเพื่อการแสดงผลให้เห็นพื้นที่ของเขตของแต่ละเซลล์เท่านั้น. เส้นสีเขียวไม่ได้เกี่ยวข้องกับการทำลักษณะสำคัญเอชโอลี. ส่วนภาพขนาด 128×64 ถูกแบ่งเป็นเซลล์ต่าง ๆ ที่แต่ละเซลล์ขนาด 8×8 . ภาพที่สามแสดงเซลล์ $v_{0,0}$ ซึ่งเป็นเซลล์แรกอยู่มุมซ้ายบนของส่วนภาพ. ภาพที่สี่และห้าแสดงขนาดและมุมของพิกเซลเกรเดียนต์ของเซลล์ $v_{0,0}$. ภาพสุดท้าย (ขวาสุด) แสดงค่าเซลล์ให้เห็น เมื่อเลือกจำนวนทิศทาง $K = 9$.

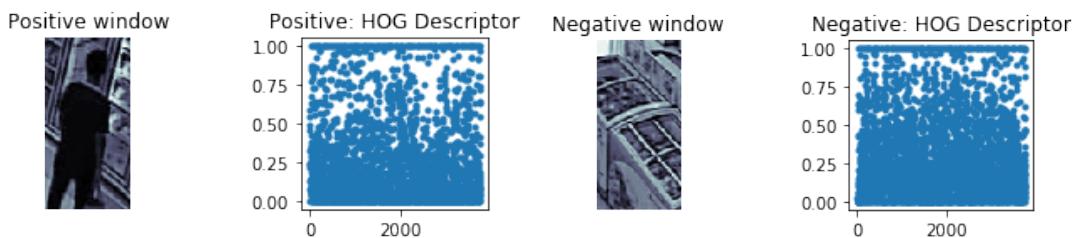
ในทิศทางที่ k^{th} รวมถึงทิศทางใกล้เคียง ของพิกเซลที่อยู่ในเขตของเซลล์. นั่นคือ หากส่วนภาพ $\mathbf{W} = [w_{r,c}]$ โดย $r = 0, \dots, A - 1$ และ $c = 0, \dots, B - 1$ มีขนาดพิกเซลเกรเดียนต์ m_{rc} และ มุมพิกเซลเกรเดียนต์ θ_{rc} แล้วเซลล์ให้ $v_k(i, j) = \sum_{r,c \in \Omega_{\text{cell}}} m_{rc}$ สำหรับ $k = 0, \dots, K - 1$ เมื่อเขต Ω_{cell} แทนเงื่อนไขพิกเซลในเขตของเซลล์ 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 $\frac{k \cdot 360}{K} \leq \theta_{rc} < \frac{(k + 1) \cdot 360}{K}$.

รูปที่ 4.7 แสดงตัวอย่างการแบ่งส่วนภาพออกเป็นเซลล์. จากอภิธานพารามิเตอร์ของตัวอย่าง ส่วนภาพถูกแบ่งออกเป็นเซลล์ $v_{0,0}, \dots, v_{15,7}$ รวมทั้งหมด 128 เซลล์. ค่าขนาดของพิกเซลเกรเดียนต์ในเซลล์จะถูกนำมารวมกันตามทิศทาง. ในภาพแสดงตัวอย่างเมื่อเลือกทำ 9 ทิศทาง ดังนั้นผลลัพธ์คือหนึ่งเซลล์จะมีส่วนประกอบ 9 ตัวสำหรับทิศทาง 0, 40, 80, 120, 160, 200, 240, 280, 320 องศา. แต่ละทิศทางครอบคลุมทิศทางใกล้เคียง เช่น 0 องศา ครอบคลุม $0 \leq \theta_{rc} < 40$. หมายเหตุภาพในรูปที่ 4.7 มีการใช้ค่าชดเชย (offset) เพื่อให้ทิศทางตัวแทนอยู่ตรงกลาง. นั่นคือใช้เงื่อนไขทิศทาง $\frac{k \cdot 360}{K} + \delta \leq \theta_{rc} < \frac{(k + 1) \cdot 360}{K} + \delta$ และใช้ค่าชดเชย $\delta = -\frac{360}{2K}$ ซึ่งในกรณีนี้คือ -20 . นั่นทำให้ 0 องศา ครอบคลุม $-20 \leq \theta_{rc} < 20$.

จากเซลล์ให้ได้ เพื่อลดผลกระทบของแสงและเงาในบริเวณต่าง ๆ เซลล์จะถูกรวมเป็นบล็อก. นั่นคือ หากบล็อกมีขนาด $n_y \times n_x$ เซลล์ และมีขนาดขับเลื่อนบล็อกเป็น $m_y \times m_x$ 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} จะมีส่วนประกอบเป็นค่าเซลล์ให้ของเซลล์ในเขตของบล็อก. นั่นคือ $\mathbf{b}_{pq} = \{\hat{v}_{ij}\}_{i,j \in \Omega_{\text{block}}}$ เมื่อเขต Ω_{block} แทนเงื่อนไข $p \cdot m_y \leq i < p \cdot m_y + n_y$ และ $q \cdot m_x \leq j < q \cdot m_x + n_x$. เวกเตอร์ \hat{v}_{ij} คือค่าเซลล์ให้หลังการทำอร์มอร์เลอร์. 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 $v_k(i, j)$ คือเซลล์ให้ที่ k^{th} ของเซลล์ (i, j) และ $v_{\max}(p, q)$ กับ $v_{\min}(p, q)$ คือค่าเซลล์ให้ที่



รูปที่ 4.8 ขั้นตอนการทำเอชโอเจ็บล็อก. บล็อกขนาด 2×2 เซลล์ และใช้ขนาดขัยบล็อก 1×1 เซลล์. แต่ละเซลล์ทำให้หาต 9 ทิศทาง. ภาพซ้าย แสดงส่วนภาพ โดยเส้นสีเขียวแสดงขอบเขตแบ่งเซลล์ $v_{0,0}$ ถึง $v_{1,2}$ และเส้นสีแดงแสดงขอบเขตของบล็อก $b_{0,1}$. ภาพถัดมา (ซึ่งภาพ $\text{Cell}[0,0]$ ถึง $\text{Cell}[1,2]$) แสดงค่าเซลล์ให้เห็นของเซลล์ $v_{0,0}$ ถึง $v_{1,2}$. ภาพขวามัน แสดงเซลล์ให้เห็นภายในบล็อกก่อนที่ทำการอัปเดต. ภาพขวาล่าง ค่าของบล็อก.



รูปที่ 4.9 ตัวอย่างลักษณะสำคัญเอชโอจี. สองภาพทางซ้าย แสดงตัวอย่างสำหรับส่วนภาพที่มีเป้าหมายอยู่. สองภาพทางขวา แสดงตัวอย่างสำหรับส่วนภาพที่ไม่มีเป้าหมายอยู่. ภาพแรกและสามจากซ้าย แสดงส่วนภาพ W ขนาด 128×64 (เท่ากับ 8192 มิติ). ภาพสองและสี่จากซ้าย แสดงลักษณะสำคัญเอชโอจี x ขนาด 3780. มิติของลักษณะสำคัญเอชโอจีน้อยกว่ามิติของส่วนภาพมาก.

มากที่สุดกับน้อยที่สุดในบล็อก ตามลำดับ.

รูปที่ 4.8 แสดงตัวอย่างขั้นตอนการทำบล็อก. ในตัวอย่าง บล็อกขนาด 2×2 เซลล์ รวมผลให้เหตุของ 4 เซลล์ หรือเท่ากับ 36 ค่าให้เหตุ. หากส่วนภาพมีจำนวนเซลล์เป็น 16×8 เซลล์ จะมีจำนวนบล็อกทั้งหมดเป็น $(\lfloor \frac{16-2}{1} \rfloor + 1) \times (\lfloor \frac{8-2}{1} \rfloor + 1) = 105$ บล็อก. ดังนั้น สำหรับส่วนภาพขนาด 128×64 ใช้เซลล์ขนาด 8×8 ทำให้เหตุ 9 ทิศทาง ใช้บล็อกขนาด 2×2 และขนาดขัยบล็อก 1×1 ลักษณะสำคัญของเอชโอจี จะมี $105 \times 2 \times 2 \times 9 = 3780$ ค่า.

รูปที่ 4.9 แสดงตัวอย่างของลักษณะสำคัญของเอชโอจี สำหรับส่วนภาพที่มีเป้าหมาย และส่วนภาพที่ไม่มีเป้าหมาย. ลักษณะสำคัญที่เปล่งมาอาจจะดูยากด้วยตาเปล่า. การวัดผลที่เหมาะสมจะมีความสำคัญมาก.

ข้อสังเกต การตรวจจับภาพวัตถุ ที่ใช้วิธีการหน้าต่างเลื่อนกับลักษณะสำคัญเอชโอจี มีการทำงานในลักษณะพื้นที่ย่อย. นั่นคือ หน้าต่างและขนาดขัยบล็อกในวิธีหน้าต่างเลื่อน แบ่งจากภาพใหญ่เป็นส่วนภาพ. เซลล์ในเอชโอจี แบ่งจากส่วนภาพเป็นเซลล์ แล้วใช้บล็อกกับขนาดขัยบล็อก แบ่งจากส่วนภาพเป็นบล็อก โดยอาศัยค่าที่ได้จากการคำนวณของเซลล์. ทั้งสามารถดูมีการทำงานในลักษณะคล้าย ๆ กัน ขั้น

ตอนหนึ่งต่อจากอีกขั้นตอนหนึ่ง. บทที่ 5 อภิรายณ์คิดของการเรียนรู้เชิงลึก และโครงสร้างคอนโวลูชัน ที่ทำแนวคิดในลักษณะนี้ แต่ทำในลักษณะที่ทั่วไปและยืดหยุ่นขึ้น. ปัจจุบัน การเรียนรู้เชิงลึกและโครงสร้างคอนโวลูชัน เป็นศาสตร์และศิลป์ของการตรวจจับภาพวัตถุ และสามารถให้ผลการทำงานที่แม่นยำมาก.

การจำแนกค่าทวิภาค. จากภาพ \mathbf{F} เลือกส่วนภาพต่าง ๆ \mathbf{W}_{ij} ออกมาด้วยวิธีหน้าต่างเลื่อน. แต่ละส่วนภาพ \mathbf{W}_{ij} จะถูกสักด้เป็นลักษณะสำคัญ \mathbf{x}_{ij} . ตอนนี้จากลักษณะสำคัญ \mathbf{x}_{ij} แบบจำลองจำแนกค่าทวิภาคสามารถนำมาใช้เพื่อทำนายผลว่าที่ตำแหน่ง (i, j) มีเป้าหมายอยู่หรือไม่. แบบจำลองจำแนกค่าทวิภาค มีหลายชนิด. บทที่ 3 อภิรายณ์โครงข่ายประสาทเทียม. โครงข่ายประสาทเทียม ก็สามารถนำมาใช้ได้ แต่ตัวอย่างนี้ นำเสนอแบบจำลองจำแนกค่าทวิภาค อีกชนิดที่ได้รับความนิยมมาก คือ ซัพพอร์ต เวกเตอร์เมชชิน.

แบบจำลองจำแนกค่า (ทั้งการจำแนกค่าทวิภาค และการจำแนกกลุ่ม) มีหลายชนิด และอาจแบ่งเป็นแนวทางใหญ่ ๆ ได้สามแนวทาง. แนวทางแรก แนวทางแบบจำลองแบ่งแยก (discriminative model) เริ่มจากการสร้างแบบจำลองแบ่งแยก ที่ทำนายความน่าจะเป็นแบบมีเงื่อนไข ที่เอาร์พุตจะเป็นหนึ่ง สำหรับอินพุตที่ถูก นั่นคือ $\Pr(y = 1|\mathbf{x})$ หรือสำหรับการจำแนกกลุ่ม ความน่าจะเป็นแบบมีเงื่อนไขของเอาร์พุตกลุ่มที่ k^{th} สำหรับอินพุตที่ถูก นั่นคือ $\Pr(y = k|\mathbf{x})$. หลังจากนั้น ใช้ทฤษฎีการตัดสินใจ เช่น วิธีระดับค่าขีดแบ่ง เพื่อเลือกค่าทวิภาค หรือฉลากของกลุ่ม สำหรับกรณีการจำแนกกลุ่ม. โครงข่ายประสาทเทียม (บท 3) สำหรับการจำแนกค่าทวิภาค หรือสำหรับการจำแนกกลุ่ม ก็จัดเป็นแบบจำลองแบ่งแยก. อาย่างไรก็ตาม การตีความเอาร์พุตของโครงข่ายประสาทเทียมในเชิงความน่าจะเป็น โดยเฉพาะกรณีจำแนกกลุ่มว่า $\hat{y}_k \approx \Pr(y = k|\mathbf{x})$ มีข้อสังเกต ข้อสงสัย และประเด็นที่กำลังสำรวจและศึกษาวิจัยอยู่[136].

แนวทางที่สอง แนวทางแบบจำลองสร้างกำเนิด (generative model) อาศัยความน่าจะเป็นแบบมีเงื่อนไขของอินพุต สำหรับเอาร์พุตแต่ละแบบ นั่นคือ $\Pr(\mathbf{x}|y)$ และความน่าจะเป็นก่อนของเอาร์พุตแต่ละแบบ นั่นคือ $\Pr(y)$ เพื่ออนุมานความน่าจะเป็นภายหลัง จากกฎของเบล์. 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 แม้ออาศัยการอนุมานการแยกแยะของอินพุต แต่อาจไม่ได้ประมาณ $\Pr(\mathbf{x}|y)$ ออกมาโดยตรง เช่น โครงข่ายปรับกษ์เชิงสร้างกำเนิด (Generative Adversarial Network[77], หัวข้อ 7.2) หรือ ตัวเข้าอัตราหัส (Autoencoder[111]).

แนวทางที่สาม แนวทางพังก์ชันแบ่งแยก (discriminant function) เป็นการสร้างพังก์ชันที่แปลงอินพุตไปเป็นเอาร์พุตโดยตรง นั่นคือ $f : \mathbf{x} \mapsto y$ โดยไม่ได้ออาศัยความน่าจะเป็น หรือไม่สามารถตีใน

เชิงความเป็นความน่าจะเป็น. ซัพพอร์ตเวกเตอร์แมชีน ก็จัดอยู่ในแนวทางนี้. หัวข้อ 4.2 อภิปราย ซัพพอร์ตเวกเตอร์แมชีน และทฤษฎีเบื้องหลัง.

การกำจัดการระบุซ้ำซ้อน

หลังจากขั้นตอนการจำแนกส่วนภาพที่มีเป้าหมายแล้ว ตำแหน่งของส่วนภาพที่ถูกจำแนกกว่ามีเป้าหมาย จะถูกบันทึกเป็นค่าตำแหน่งที่ตราชพบ. ตำแหน่ง อาจบันทึกเป็นพิกัดของกล่องขอบเขต (bounding box) เช่น พิกัด (x, y) มุมขวาบนของกล่องขอบเขต กับพิกัดมุมล่างซ้าย หรืออาจจะเป็น พิกัดมุมขวาบนกับความกว้างและความสูง. กล่องขอบเขต สามารถนิยามเป็นขอบเขตของหน้าต่างที่เลือกส่วนภาพ ออกมาก. ส่วนภาพในบริเวณใกล้เคียงกัน อาจถูกระบุว่ามีเป้าหมาย โดยที่เป้าหมายที่ส่วนภาพเหล่านั้น มี เป็นเป้าหมายเดียวกัน ซึ่งเป็นการระบุซ้ำซ้อน. รูปที่ 4.5 แสดงตัวอย่างส่วนภาพ ที่ได้จากการวิธีหน้าต่าง เลื่อน. สังเกตว่า มีหลายส่วนภาพที่ครอบคลุมเป้าหมายเดียวกัน และอาจมีการระบุซ้ำซ้อนเกิดขึ้น.

จากตำแหน่งของส่วนภาพต่าง ๆ ที่ซ้ำซ้อน จะมีแค่ตำแหน่งเดียวที่จะเป็นตัวแทนของตำแหน่งของ เป้าหมาย และที่เหลือจะถูกลบทิ้งไป. ขั้นตอนการกำจัดการระบุซ้ำซ้อนนี้ จะเรียกว่า การกำจัดความซ้ำซ้อน (redundancy removal). การกำจัดความซ้ำซ้อน ดำเนินการตั้งแต่ตรวจสอบความซ้ำซ้อน และกำจัดความซ้ำซ้อนที่พบ. ผลลัพธ์คือ ตำแหน่งต่าง ๆ ของการตรวจจับ ที่ไม่ซ้ำซ้อนกัน.

เพื่อตรวจสอบความซ้ำซ้อน แนวทางที่นิยม คือ กำหนดค่าระดับขีด贲ง τ และกล่องขอบเขตสอง กล่อง จะถือว่าซ้ำซ้อนกัน เมื่อ บริเวณซ้อนทับกันมีค่าการซ้อนทับมากกว่า ค่าระดับขีด贲ง τ . ค่าการซ้อนทับ มักถูกวัดด้วย ไอโอ尤 (IoU ซึ่งย่อมาจาก Intersect over Union) ซึ่งเป็นสัดส่วนพื้นที่ซ้อนทับกันต่อพื้นที่รวม. 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 ที่ซ้ำซ้อนกัน การกำจัดความซ้ำซ้อน อาจทำโดยสุ่มเลือกกล่องขอบเขตตัวใดกล่องหนึ่ง และตัดกล่องที่เหลือทิ้งไว้ แต่อาจทำให้คุณภาพโดยรวมของการตรวจจับด้อยลง. วิธีการระงับค่าไม่มากสุดท้องถิ่น (non-local-maximum suppression[126]) จะระงับหรือตัด ทึ้งกล่องขอบเขต ที่มีค่าความเหมือนไม่มากที่สุด เมื่อเทียบกับกล่องขอบเขตอื่น ๆ ที่อยู่รอบ ๆ กล่องนั้น. ค่าความเหมือนของกล่องขอบเขต อาจได้มาจากการแบบจำลองจำแนก เช่น กรณีโครงข่ายประสาทเทียม ค่าเอาร์พุตของโครงข่าย (ก่อนผ่านการตัดสินใจด้วยวิธีระดับค่าขีด贲ง) ถูกตีความ เป็นค่าความน่าจะเป็น และสามารถนำมาใช้เป็นค่าความเหมือนได้. สำหรับกรณีซัพพอร์ตเวกเตอร์ แมชีน ค่าคะแนนตัดสินใจ (decision score สมการ 4.19) สามารถนำมาใช้ได้. アナเซตรอบกล่องที่พิจารณา โดยทั่วไป มักหมายถึงกล่องที่อยู่ติดกัน แต่อย่างไรก็ตาม ความกว้างของアナเซตันี้ สามารถกำหนดเป็นอภิมานพารามิเตอร์ได้.

การนำเสนอผลด้วยแผนที่ความร้อน

แผนที่ความร้อน เป็นแผนภาพสี ที่ให้ข้อมูลความถี่เชิงพื้นที่. ความถี่ของตำแหน่งที่พบลูกค้าบ่อย ๆ อนุมานมาจากพิกัดตำแหน่งต่าง ๆ ที่ตรวจพบลูกค้า จากชุดลำดับภาพของวิดีโอ. เวลาที่ลูกค้าใช้ในแต่ละตำแหน่ง จะสะท้อนอุปมาในความถี่ที่แสดงนี้.

จากพิกัดตำแหน่งที่ตรวจพบลูกค้า ซึ่งอาจเป็นพิกัดของกล่องขอบเขต จะถูกแปลงเป็นพิกัดตัวแทน ซึ่งอาจใช้จุดศูนย์กลางของกล่องขอบเขต. จากนั้น พิกัดตำแหน่งที่ตรวจพบลูกค้าในแต่ละภาพจะถูกนำมารวมกัน ซึ่งเขียนเป็น $\mathbf{D} = \{\mathbf{d}_i\}$ สำหรับ $i = 1, \dots, N$ เมื่อ $\mathbf{d}_i = [x_i, y_i]^T$ เป็นพิกัดตำแหน่งที่พบลูกค้า ในแนวนอนและแนวตั้งของภาพ ตามลำดับ. ดังนี้ i เป็นดัชนีของพิกัด และ N คือจำนวนพิกัดทั้งหมดที่ต้องการนำมาสรุปเป็นความถี่เชิงพื้นที่. หมายเหตุ ในทางปฏิบัติ การเลือกพิกัดตรวจพบมากขึ้น อาจเลือกตามระยะเวลา เช่นภายในหนึ่งเดือนที่ผ่านมา หรือ อาจเลือกตามช่วงเวลาที่สนใจได้ เช่นแยกสรุประหว่างวันธรรมดា และวันเสาร์อาทิตย์. แต่ ณ ที่นี่ แสดงดัชนี i เป็นดัชนีสำหรับพิกัดที่คัดเลือกมาแล้ว และ N เป็นจำนวนทั้งหมดที่ต้องการนำมาสรุปรวมกัน.

การสร้างแผนที่ความร้อน ก็คือ การแปลงข้อมูล $\{\mathbf{d}_i\}$ ไปเป็นภาพสีขนาด $H \times W$. วิธีหนึ่งที่นิยมคือ วิธีการประมาณความหนาแน่นแก่น (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

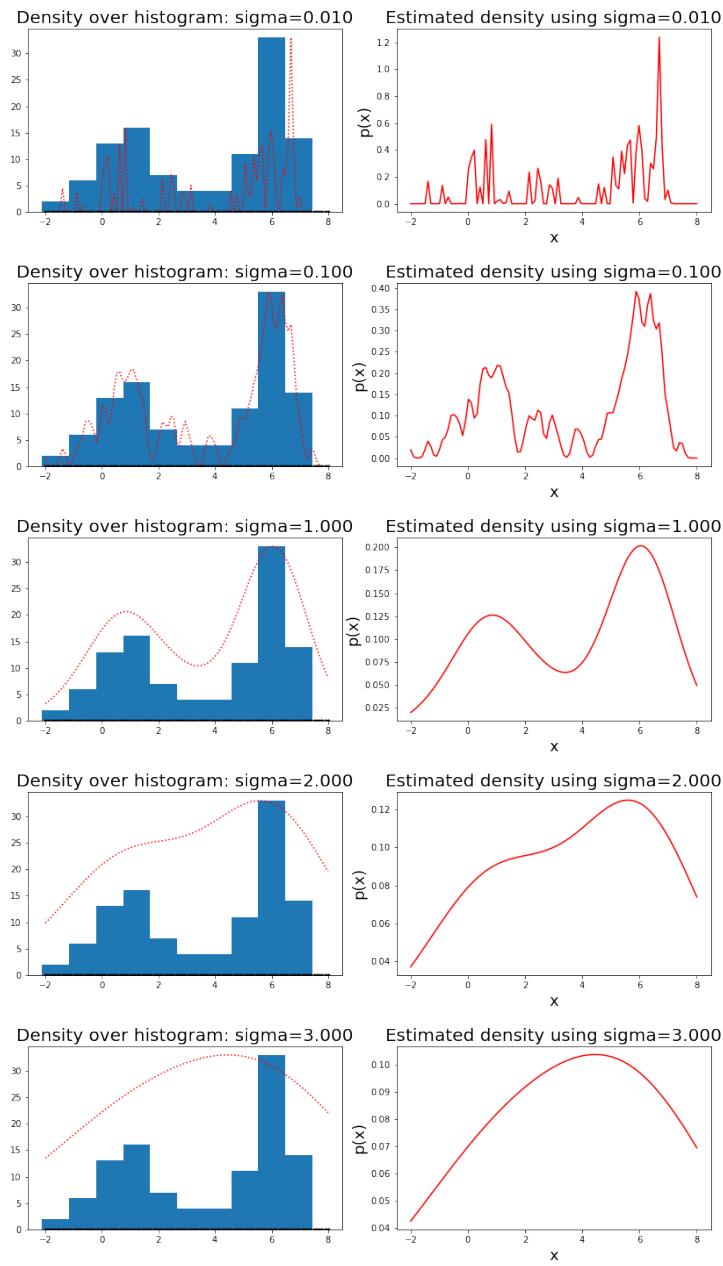
$$\hat{p}(\mathbf{v}) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot z(\mathbf{v}) \quad (4.3)$$

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 σ เป็นอภิมานพารามิเตอร์ ซึ่งควบคุมความระบริءนความต่อเนื่องของผลลัพธ์.

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} จะถูกนำมาเป็นภาพสี โดยการแปลงค่า z ที่แต่ละพิกเซลเป็นสีตามแต่ระบบสีที่จะเลือกใช้.



รูปที่ 4.10 ผลการประมาณความหนาแน่นความน่าจะเป็น ด้วยวิธีการประมาณความหนาแน่นแก่น เมื่อใช้ค่า σ ต่าง ๆ ได้แก่ 0.01, 0.1, 1, 2, และ 3. ค่า σ ระบุไว้เหนือรูป. ภาพชี้รายแสดงความหนาแน่นที่ประมาณ ข้อนอยู่บนฮิสโตแกรม ของจุดข้อมูล โดยความหนาแน่นที่ประมาณถูกปรับขนาด เพื่อเปรียบเทียบกับฮิสโตแกรมได้ชัดเจน. ภาพชี้รายแสดงความหนาแน่นที่ประมาณ โดยแกนนอนแสดงค่าข้อมูล x และแกนตั้งแสดงค่าความหนาแน่นความน่าจะเป็น $p(x)$ ที่ได้จากการประมาณ.

รูปที่ 4.10 แสดงผลการประมาณความหนาแน่นความน่าจะเป็น ด้วยวิธีการประมาณความหนาแน่นแก่น เมื่อใช้ค่า σ ต่าง ๆ ค่า σ อาจมองเห็นเป็นการปรับความเรียบของความหนาแน่นที่จะประมาณ หรืออาจเปรียบเสมือนสมมติฐานเบื้องต้น เกี่ยวกับการความหนาแน่นของข้อมูล ว่าข้อมูลมีความหนาแน่นที่มีลักษณะเป็นการแจกแจงฐานนิยมเดียว (*unimodal distribution*) หรือแบบพหุฐาน (*multimodal*) และการแจกแจกมีความหลากหลาย มีความซับซ้อนมากขนาดไหน. ภาพในรูป แสดงค่า σ ขนาดเล็กให้ผลการประมาณความหนาแน่นที่มีความซับซ้อนมาก มีจำนวนฐานมาก ฐานแคบ. ค่า σ ขนาดใหญ่ให้ผลการประมาณความหนาแน่นที่ซับซ้อนน้อย มีจำนวนฐานน้อยลง และความหนาแน่นมีการกระจายตัวออกไปมากขึ้น ฐานกว้าง.

การประเมินผลการตรวจจับ

การประเมินผลเป็นหัวใจของการเรียนรู้ของงานการเรียนรู้ของเครื่อง เป็นหัวใจของงานวิศวกรรมและวิทยาศาสตร์ และเป็นหัวใจของ น่าจะเรียกได้ว่า ทุกภาระกิจ. การประเมินผลการตรวจจับวัตถุ อาจทำได้หลายวิธี. หนึ่งในวิธีที่นิยม คือ การประเมินด้วยค่าเฉลี่ยค่าประมาณความเที่ยงตรง.

ค่าเฉลี่ยค่าประมาณความเที่ยงตรง (mean Average Precision คำย่อ mAP) เป็นตัวชี้วัดที่นิยมใช้ประเมินคุณภาพระบบตรวจจับภาพวัตถุ. ค่าเฉลี่ยค่าประมาณความเที่ยงตรงสามารถใช้ประเมินความแม่นยำในการตรวจจับตำแหน่ง พร้อมกับประเมินความแม่นยำในการทายชนิดของวัตถุ โดยเป็นการประเมินความสามารถของระบบโดยรวม ไม่เฉพาะเจาะจงกับการเลือกรัดบ์ค่าชีดแบ่ง. (ดูแบบฝึกหัด 3.17 สำหรับตัวอย่างผลจากการเลือกรัดบ์ค่าชีดแบ่งที่ต่างกัน.) ค่าเฉลี่ยค่าประมาณความเที่ยงตรง สามารถคำนวณได้จาก

$$\text{mAP} = \frac{1}{K} \sum_{k \in \text{Classes}} \text{AP}_k \quad (4.5)$$

เมื่อ **Classes** คือเซตของลักษณะต่าง ๆ ที่เกี่ยวข้อง. ค่า K คือจำนวนของชนิดลักษณะที่เกี่ยวข้อง (ขนาดของเซต **Classes**). ค่า AP_k คือค่าประมาณความเที่ยงตรงของสำหรับการตรวจจับภาพของวัตถุชนิด k .

ค่าประมาณความเที่ยงตรงของวัตถุแต่ละชนิด 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 .

รูปที่ 4.11 แสดงเฉลย และผลการตรวจจับวัตถุสองชนิด จากภาพสองภาพ. แต่ละการตรวจจับจะมีค่าการตรวจจับระบุอยู่ (ν ในภาพ). มีหลายวิธีในการคำนวณแม่นยำในการตรวจจับตำแหน่ง เช่น

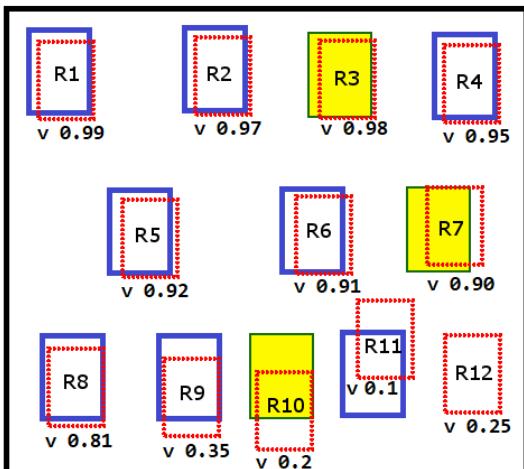


Image 1

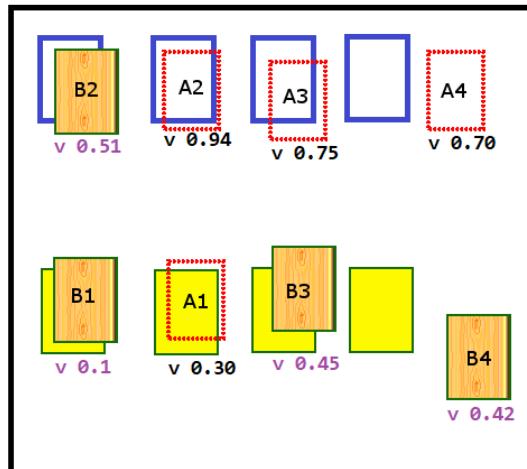


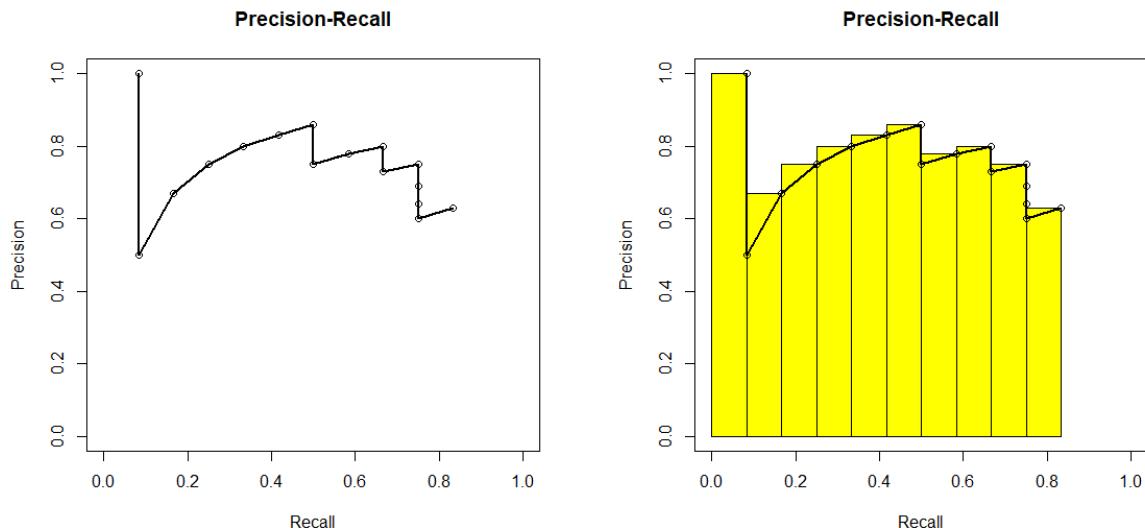
Image 2



รูปที่ 4.11 ตัวอย่างแสดงผลลัพธ์การตรวจจับและผลเฉลยของวัตถุชนิด A และชนิด B สำหรับภาพ 2 ภาพ. ทุกกล่องขอบเขตของการตรวจจับ จะมีค่าการตรวจจับ v และตัวเลขกำกับ. ค่าการตรวจจับ v จะนำไปใช้จัดลำดับการตรวจจับได้. ภาพ 1 (ทางซ้าย) มีวัตถุชนิด A (กรอบเส้นทึบสีน้ำเงิน) อยู่ 8 วัตถุ. วัตถุชนิด B (กรอบเส้นทึบสีเขียวภายในสีเหลือง) อยู่ 3 วัตถุ การตรวจจับให้ผลลัพธ์ออกมาเป็น กล่องขอบเขต 12 ตำแหน่งสำหรับชนิด A (กรอบเส้นประสีแดง) และตรวจไม่พบ วัตถุชนิด B เลย (ไม่มีกล่องขอบเขตของ B ที่แทนด้วยกรอบลายไม้). ภาพ 2 (ทางขวา) มีวัตถุชนิด A อยู่ 4 วัตถุ. วัตถุชนิด B มีอยู่ 4 วัตถุ. การตรวจจับให้ผลลัพธ์ออกมาเป็น กล่องขอบเขต 4 ตำแหน่งสำหรับชนิด A และ 4 ตำแหน่งสำหรับชนิด B. ผลลัพธ์จากการตรวจจับมีทั้ง (1) กรณีที่ตรวจจับได้ถูกต้องทั้งตำแหน่งและชนิด ได้แก่ ชนิด A คือ R1, R2, R4, R5, R6, R8, R9, R11, A2, A3 และชนิด B คือ B1, B3. (2) กรณีตรวจจับตำแหน่งได้ถูกต้องแต่ผิดชนิด ได้แก่ หาย A ให้ B คือ R3, R7, R10, A1 และหาย B ให้ A คือ B2. (3) กรณีผิดทั้งตำแหน่งและชนิด ได้แก่ R12, A4, B4. และ (4) กรณีตรวจไม่พบวัตถุ ทั้ง ๆ ที่มีวัตถุอยู่ ได้แก่ ภาพขวา (Image 2) วัตถุ A แบบบนตำแหน่งที่สองจากขวา และวัตถุ B แบบล่างตำแหน่งสองจากขวา. นอกจากนั้น ตัวอย่างนี้ยังแสดงความแม่นยำในการระบุตำแหน่งที่แตกต่างกันอีกด้วย.

มาร่วมด้วย. วิธีง่าย ๆ อาจใช้วิธีระดับค่าขีดเบ่ง กับการวัดไอโอyu เพื่อตัดการตรวจจับที่ตำแหน่งคลาดเคลื่อนมากทึ้ง. ตัวอย่างเช่น หากกล่องขอบเขตของการตรวจจับ มีค่าไอโอyu กับกล่องขอบเขตเฉลย ต่ำกว่า 0.5 ถือว่าผิด นั่นคือเท่ากับ ตรวจจับเกินหนึ่ง สำหรับการตรวจจับโดย แลตรวจสอบไม่พบหนึ่ง สำหรับเฉลยที่ไม่มีกล่องขอบเขตตรวจพบ. หรืออาจใช้ค่าไอโอyu เข้าไปคำนวนประกอบกับค่าความมั่นใจอื่น เพื่อสรุปอุปกรณามาเป็นค่าการตรวจจับก็ได้ ซึ่งจะทำให้กล่องขอบเขตของการตรวจจับ ที่มีตำแหน่งคลาดเคลื่อนไปมาก จะได้คะแนนค่าการตรวจจับน้อย.

ตัวอย่าง การประเมินค่าเฉลี่ยค่าประมาณความเที่ยงตรง จากรูปที่ 4.11 ค่าเฉลี่ยค่าประมาณความเที่ยงตรงของการตรวจจับ สามารถหาได้จาก (1) เรียงลำดับการตรวจจับตามชนิด (2) ตรวจสอบผลลัพธ์จากการตรวจจับ เปรียบเทียบกับผลเฉลย (3) คำนวนหาค่าความเที่ยงตรง p_{kj} และค่าการเรียกกลับ r_{kj} (4) คำนวนพื้นที่ใต้กราฟความเที่ยงตรงและการเรียกกลับ (Area under P-R curve) และ (5) คำนวนพื้นที่ใต้กราฟเฉลี่ยของทุก ๆ ชนิด ซึ่งคือค่าเฉลี่ยค่าประมาณความเที่ยงตรง mAP. ตาราง



รูปที่ 4.12 ภาพซ้าย กราฟความเที่ยงตรงและการเรียกกลับ ของการตรวจจับวัตถุชนิด A (ดูตารางที่ 4.1 ประกอบ). ภาพขวา ค่าประมาณพื้นที่ได้กราฟความเที่ยงตรงและการเรียกกลับ ของการตรวจจับวัตถุชนิด A

ที่ 4.1 แสดงตัวอย่างการคำนวณค่าเฉลี่ยค่าประมาณความเที่ยงตรง.

ผลลัพธ์การตรวจหาวัตถุกวนนำมาจัดลำดับตามค่าการตรวจจับ. ค่าการตรวจจับนี้อาจเป็นค่าความน่าจะเป็นที่ระบบตรวจจับวัตถุให้ออกมา หรืออาจจะเป็นค่าอื่นในลักษณะคล้ายกัน เช่น ค่าความมั่นใจ[160] หรือค่าไอโอดิย (สมการ 4.2) หรือค่าความน่าจะเป็นคุณกับค่าไอโอดิย[161] ก็ได้. ในตัวอย่างนี้ ค่าการตรวจจับที่สูง หมายถึงการตรวจจับได้รับลำดับความสำคัญเป็นลำดับต้น ๆ. สังเกตว่า การจัดลำดับ ตามนิदิจวัตถุที่ทำนาย เช่น ทุกการตรวจจับวัตถุที่ถูกทำนายเป็นชนิด A จะถูกนำมาจัดลำดับด้วยกัน ไม่ว่าจะเป็นการทำนายที่ภาพใด (หรือผลการทำนายถูกหรือไม่ หรือว่าเฉลยจริงเป็นชนิดใด).

แต่ละผลลัพธ์จากการตรวจจับ จะถูกเปรียบเทียบกับผลเฉลย. ในตารางที่ 4.1 แสดงค่าความถูกต้อง จะระบุเป็น 1 หากมีผลเฉลยชนิดนั้นในตำแหน่งบริเวณนั้น และค่าความถูกต้อง จะระบุเป็น 0 หากไม่ใช่. ตัวอย่างเช่น ในรูปที่ 4.11 ภาพซ้าย กล่องขอบเขต R3, R7, และ R10 ที่ทำนายตำแหน่งของวัตถุชนิด A แต่บริเวณนั้นไม่มีวัตถุชนิด A อยู่ (มีแต่วัตถุชนิด B) หรือ กล่องขอบเขต R12 ที่ทำนายตำแหน่งของวัตถุชนิด A แต่บริเวณนั้นไม่มีวัตถุใดอยู่เลย ค่าความถูกต้องของกล่องขอบเขตเหล่านี้ จะเป็นศูนย์.

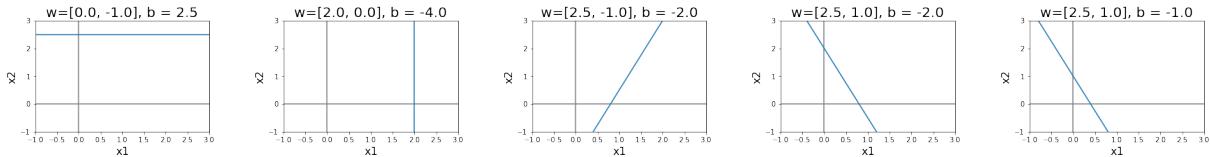
หากผลลัพธ์จากการตรวจจับถูกต้อง ค่าบวกจริง TP จะเพิ่มขึ้นหนึ่ง (เริ่มจากลำดับบนสุด) แต่หากผลลัพธ์จากการตรวจจับไม่ถูกต้อง ค่าบวกเท็จ FP จะเพิ่มขึ้นหนึ่ง (เริ่มจากลำดับบนสุด). ค่าความเที่ยงตรง p_{kj} ซึ่งเป็นอัตราส่วนการทำนายถูกต่อการทำนายทั้งหมด จะสามารถคำนวณได้จาก $p_{kj} = TP / (TP + FP)$ เมื่อ TP และ FP เป็นค่าบวกจริง และค่าบวกเท็จ สำหรับชนิดวัตถุ k ที่ลำดับ j เช่น สำหรับการทำนายวัตถุชนิด A ที่ลำดับแรกสุด นั่นคือ การทำนายกล่องขอบเขต R1 ทำให้ได้ $TP = 1, FP = 0$ และ $p_{A1} = 1$. แต่ที่ลำดับที่สอง (เปรียบเสมือนการตั้งค่าขีดแบ่งอ่อนลง) นั่นคือ การทำนายกล่องขอบเขต R1 กับกล่องขอบเขต R3 ทำให้ได้ $TP = 1, FP = 1$ และ $p_{A2} = 0.5$.

ตารางที่ 4.1 ตัวอย่างการคำนวณค่าเฉลี่ยค่าประมาณความเที่ยงตรง

ผลลัพธ์	ค่าการตรวจจับ	ความถูกต้อง	ชนิด	<i>TP</i>	<i>FP</i>	p_{kj}	r_{kj}	Δr_{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
$AP_A =$									0.65
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
$AP_B =$									0.14
$mAP = \text{mean}_k AP_k = \frac{AP_A + AP_B}{2} \approx$									0.40

ค่าการเรียกกลับ r_{kj} เป็นอัตราส่วนการทายถูกต่อผลเฉลยทั้งหมด สำหรับชนิดวัตถุ k ที่ลำดับ j ซึ่งคำนวณได้จาก $r_{kj} = TP/N_k$ เมื่อ N_k คือจำนวนผลเฉลยทั้งหมดของวัตถุชนิด k . ในที่นี้ (ดูรูปที่ 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) ค่าการเรียกกลับ $r_{A1} = 1/12 \approx 0.08$. แต่ที่ลำดับที่สิบหก (ลำดับสุดท้าย เทียบเท่า การทายผลลัพธ์ทั้ง 16 อันออกไป) ซึ่งทายถูก 10 อัน ($TP = 10$) ทำให้ได้ค่าการเรียกกลับ $r_{A16} = 10/12 \approx 0.83$.

ค่า p_{kj} และ r_{kj} ที่ได้สามารถนำมาราฟความเที่ยงตรงและการเรียกกลับ (Area under P-R curve) ได้. รูปที่ 4.12 (ภาพข่าย) แสดงกราฟความเที่ยงตรงและการเรียกกลับของการตรวจ



รูปที่ 4.13 อภิรานบในสองมิติ เทียบเท่าเส้นตรง. แต่ละภาพแสดงอภิรานบในสองมิติ เมื่อใช้ค่า w และ b ต่าง ๆ ซึ่งค่าระบบอยู่ด้านบนของภาพ. อภิรานบแสดงด้วยเส้นทึบสีฟ้า.

จับวัตถุนิด A. ค่าประมาณความเที่ยงตรงเป็นการประมาณพื้นที่ให้กราฟความเที่ยงตรงและการเรียงกลับ. หากพื้นที่ให้กราฟมีขนาดใหญ่ หมายถึงคุณภาพที่ดีของตรวจสอบจับภาพวัดถูก. การประมาณพื้นที่ให้กราฟความเที่ยงตรงและการเรียงกลับของการตรวจจับวัตถุนิด A แสดงในรูปที่ 4.12 (ภาพขวา).

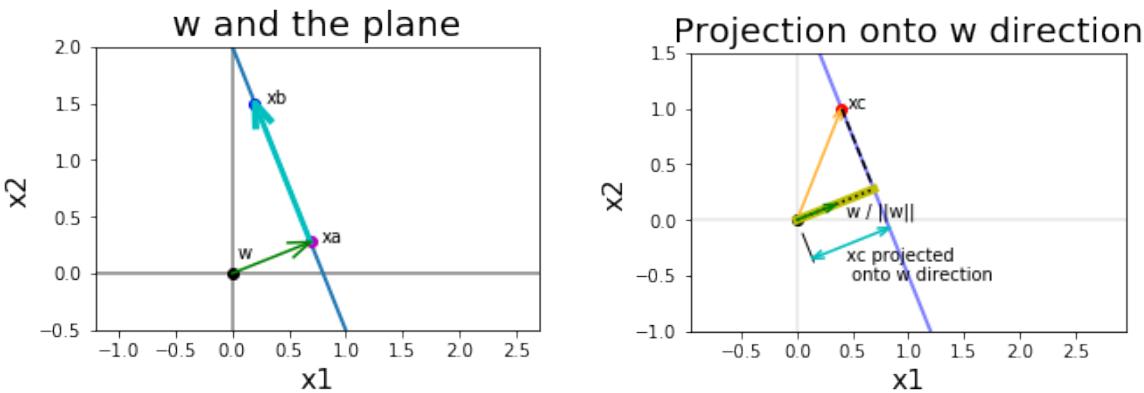
การประมาณพื้นที่ให้กราฟความเที่ยงตรงและการเรียงกลับ อาจใช้วิธีการคำนวนสี่เหลี่ยมคงหู ที่ช่วยให้ได้พื้นที่ที่แม่นยำกว่าได้ แต่โดยทั่วไปแล้ว การประมาณคร่าว ๆ ด้วยสี่เหลี่ยมก็เพียงพอ. พื้นที่ให้กราฟ หรือค่าประมาณความเที่ยงตรงของวัตถุแต่ละชนิด จะถูกนำมาเฉลี่ยกัน เพื่อคำนวนเป็นค่าเฉลี่ยค่าประมาณความเที่ยงตรง เช่นในตัวอย่างนี้ $mAP = 0.40$.

4.2 ชั้พพอร์ตเวกเตอร์แมชชีน

ชั้พพอร์ตเวกเตอร์แมชชีน (Support Vector Machine[43] คำย่อ SVM) เป็นแบบจำลองจำแนกค่าทวิภาค ในแนวทางพัฟ์กชันแบ่งแยก ซึ่งแปลงค่าอินพุต $\mathbf{x} \in \mathbb{R}^D$ ไปเป็นเอาต์พุต $\hat{y} \in \{-1, +1\}$ โดยตรง และไม่มีความเชื่อมโยงกับค่าความน่าจะเป็น. กลไกการทำงานของชั้พพอร์ตเวกเตอร์แมชชีน อาศัยการแปลงข้อมูลจากปริภูมิของข้อมูลต้นฉบับไปสู่ปริภูมิใหม่ ซึ่งอาจเรียกว่า **ปริภูมิลักษณะสำคัญ (feature space)** โดยปริภูมิลักษณะสำคัญนี้จะช่วยให้การแบ่งแยกข้อมูลออกเป็นกลุ่มได้ง่ายขึ้น และอาศัยอภิรานบในปริภูมิลักษณะสำคัญ เพื่อตัดแบ่งแยกข้อมูลออกเป็นสองกลุ่ม.

อภิรานบ (hyperplane) หมายถึง ระนาบในปริภูมิหลายมิติ. ในปริภูมิสองมิติ อภิรานบ จะหมายถึง เส้นตรง. นั่นคือ อภิรานบ 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. ในปริภูมิสาม มิติ อภิรานบ จะหมายถึง ระนาบ (แผ่นตรงเรียบ ในสามมิติ). นั่นคือ อภิรานบ จะสามารถระบุได้ ด้วยสมการ $w_1x_1 + w_2x_2 + w_3x_3 + b = 0$ เมื่อ $\mathbf{x} = [x_1, x_2, x_3]^T$ เป็นตัวแปรในปริภูมิ. ใน ปริภูมิ D มิติ อภิรานบ อาจจะยากที่จะจินตนาการ แต่ อภิรานบ ก็จะสามารถระบุได้ด้วยสมการ $\mathbf{w}^T \mathbf{x} + b = 0$ เมื่อ $\mathbf{w}, \mathbf{x} \in \mathbb{R}^D$.

รูปที่ 4.13 แสดงอภิรานบในสองมิติ. สังเกตความสัมพันธ์ระหว่างทิศทางและตำแหน่งของอภิรานบ กับค่าของ \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 $\mathbf{w}^T \mathbf{x}_b + b = 0$ เวกเตอร์จาก \mathbf{x}_a ไป \mathbf{x}_b ซึ่งคือ $\mathbf{x}_b - \mathbf{x}_a$ เป็นเวกเตอร์ในแนวของ



รูปที่ 4.14 ความสัมพันธ์ระหว่างพารามิเตอร์ w และ b กับคุณลักษณะของภาระนาบ. ภาพซ้าย x_a และ x_b เป็นจุดใด ๆ บนภาระนาบ. ระยะนับแสดงด้วยเส้นสีฟ้า. เวกเตอร์จาก x_a ไป x_b แสดงด้วยลูกศรสีฟ้าเขียว. เวกเตอร์จากจุดกำเนิดไป w แสดงด้วยลูกศรสีเขียว. ภาพขวา x_c เป็นจุดใด ๆ บนภาระนาบ ซึ่งระยะนับแสดงด้วยเส้นสีฟ้า. เวกเตอร์จากจุดกำเนิดไปหัวจุด x_c แสดงด้วยลูกศรสีส้ม. เวกเตอร์หนึ่งหน่วยในทิศทาง w ซึ่งคือ $w/\|w\|$ แสดงด้วยลูกศรสีเขียว. ขนาดของเวกเตอร์ x_c ที่ฉายลงบน $w/\|w\|$ แสดงด้วยเส้นสีเหลือง. แต่บริเวณนั้นมีการซ้อนทับกันมาก อาจมองไม่ชัด เส้นสีฟ้าเขียวที่มีลูกศรส่องทาง ขยับออกมาระยะห่างของการฉายให้ชัดเจนขึ้น.

ระยะนับ. ดูรูปที่ 4.14 ภาพซ้ายประกอบ. พิจารณาผลคูณเวกเตอร์ระหว่าง w กับเวกเตอร์ในแนวระยะนับ และจากคุณสมบัติของจุดบนระยะนับ ทำให้พบว่า $w^T \cdot (x_b - x_a) = w^T x_b - w^T x_a = w^T x_b + b - w^T x_a - b = 0$. ผลคูณของ w กับเวกเตอร์ในแนวระยะนับ จะเป็นศูนย์เสมอ. นั่นแปลว่า เวกเตอร์ w ตั้งฉากกับแนวระยะนับ. ดังนั้น ทิศทางของระยะนับกำหนดด้วยค่าของเวกเตอร์ w .

แต่ระยะจะห่างจากจุดกำเนิดเท่าไร พิจารณารูปที่ 4.14 ภาพขวา. จุดกำเนิด (origin) คือจุด $[0, 0]^T$ ในปริภูมิสองมิติ แสดงด้วยจุดกลมสีดำในภาพ (ภาพขวา จุดนี้อาจลูกบังจากเวกเตอร์ต่าง ๆ). ระยะห่างระหว่างจุดกำเนิดกับระยะนับ คือระยะจากจุดกำเนิดไประยะนับในทิศทางที่ตั้งฉากกับระยะนับ. นั่นคือ หากกำหนดให้ x_c เป็นจุดใด ๆ ในระยะนับ ขนาดของภาพฉายของ x_c ลงบนเวกเตอร์หนึ่งหน่วยในแนว w จะเป็นระยะทางจากจุดกำเนิดไปถึงระยะนับ. ดังนั้น ระยะทางจากจุดกำเนิดไประยะนับสามารถคำนวณได้จาก $d = \frac{w^T}{\|w\|} \cdot x_c = \frac{w^T x_c}{\|w\|}$ เมื่อ d คือระยะทางจากจุดกำเนิดไประยะนับ. จากคุณสมบัติของระยะนับ $w^T x_c + b = 0$ ทำให้ พบร่วม $d = \frac{-b}{\|w\|}$. หมายเหตุ ขนาดของการฉายภาพเป็นลบ หมายถึงทิศทางของภาพพิจารณา จะกลับทิศกับเวกเตอร์หนึ่งหน่วยที่เป็นเสมือนจาก. ถ้า $b < 0$ ทำให้ ระยะ $d > 0$ ระยะจะห่างจุดกำเนิดออกไปขนาด $|d|$ ทางทิศ w ถ้า $b > 0$ ทำให้ ระยะ $d < 0$ ระยะจะห่างจุดกำเนิดออกไปขนาด $|d|$ ทางทิศตรงข้ามกับ w และถ้า $b = 0$ หมายถึงระยะจะผ่านจุดกำเนิด.

การวางแผนปัญหาของชั้พพอร์ตเวกเตอร์แมชชีน. ชั้พพอร์ตเวกเตอร์แมชชีน ในปัจจุบันถูกประยุกต์ในงานหลากหลายทั้งงานการหาค่าลดถอย และการจำแนกกลุ่ม แต่ตั้งเดิมเริ่มต้น ชั้พพอร์ตเวกเตอร์แมชชีน ถูกออกแบบสำหรับการจำแนกค่าทิวภาค. ข้อมูลจะถูกแบ่งออกเป็น กลุ่มบาง และกลุ่ม

ลบ. แนวคิดของชัพพอร์ตเวกเตอร์แมชชีน คือ การใช้อภิรานาเป็นเส้นแบ่งการตัดสินใจ และข้อมูลฝึกจะถูกนำมาใช้ เพื่อเลือกอภิรานาในปริภูมิลักษณะสำคัญ ที่ทำให้ช่องว่างที่แบ่งระหว่างจุดข้อมูลกลุ่มบวกกับกลุ่มลบห่างกันมากที่สุด.

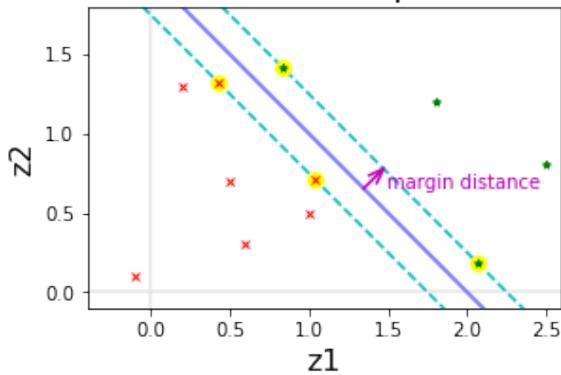
ชัพพอร์ตเวกเตอร์แมชชีน อาศัยกลไกที่สำคัญสองอย่าง. กลไกสำคัญแรก คือ การแปลงจุดข้อมูลไปสู่ปริภูมิลักษณะสำคัญ. เนื่องจากอภิรานาเป็นพังก์ชันเชิงเส้น บทบาทของการแปลงข้อมูล จะช่วยในการแปลงข้อมูลไปสู่ปริภูมิที่ข้อมูลจะสามารถถูกแบ่งได้ด้วยอภิรานา. กำหนดให้ลักษณะสำคัญ $z = \phi(x)$ เมื่อ x เป็นจุดข้อมูลในปริภูมิข้อมูลดั้งเดิม และ $\phi : \mathbb{R}^D \mapsto \mathbb{R}^M$ เป็นพังก์ชันที่ใช้แปลงข้อมูลไปสู่ปริภูมิลักษณะสำคัญ โดย D และ M คือจำนวนมิติของปริภูมิข้อมูลดั้งเดิมและของปริภูมิลักษณะสำคัญตามลำดับ. ดังนั้น จุดข้อมูล x (ในปริภูมิข้อมูลดั้งเดิม) จะถูกแทนด้วย z ในปริภูมิลักษณะสำคัญ. การใช้งานชัพพอร์ตเวกเตอร์แมชชีนให้มีประสิทธิผล เกี่ยวพันโดยตรงกับการเลือกพังก์ชันลักษณะสำคัญให้เหมาะสม ซึ่งจะได้อภิรายละเอียดในหัวข้อ 4.2. หมายเหตุ การเลือกที่จะทำการแบ่งข้อมูลในปริภูมิดั้งเดิม สามารถทำได้ และในหลาย ๆ กรณี ก็เป็นทางเลือกที่เหมาะสม. การเลือกที่จะทำการแบ่งข้อมูลในปริภูมิดั้งเดิม ก็เปรียบเสมือนการเลือกให้พังก์ชันเอกลักษณ์เป็นพังก์ชันลักษณะสำคัญ นั่นคือ $\phi(x) = x$ และ $z = x$.

กลไกสำคัญที่สอง คือ การหาอภิรานาที่ทำให้ขอบเขตของการแบ่งกว้างที่สุด. รูปที่ 4.15 แสดงตัวอย่างความสัมพันธ์ระหว่าง จุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสำคัญ อภิรานาตัดสินใจ และขอบเขตของการแบ่ง. ออกแบบเพื่อการแบ่งกลุ่มสองกลุ่มโดยเฉพาะ จุดประสงค์ของชัพพอร์ตเวกเตอร์แมชชีน ‘ไม่ใช่แค่หาอภิรานาได้ก็ได้ที่แบ่งข้อมูลได้ แต่ต้องการหาอภิรานาที่แบ่งข้อมูลได้ และแบ่งได้โดยมีขอบเขตของการแบ่ง (margin of separation) ที่กว้างที่สุดด้วย. สังเกตว่า ในรูปที่ 4.15 หากเอียงอภิรานาตัดสินใจเพิ่มขึ้นหรือลดลงเล็กน้อย ผลที่ได้ก็จะยังคงสามารถแบ่งข้อมูลได้สมบูรณ์ แต่ขอบเขตของการแบ่งจะแคบลง.

จุดข้อมูลที่อยู่บนแนวขอบเขตของการแบ่ง ซึ่งเป็นจุดข้อมูลที่อยู่ใกล้กับจุดข้อมูลจากต่างกลุ่มมากที่สุด เป็นจุดที่แบ่งยากที่สุด และอภิรานาจะถูกกำหนดด้วยจุดข้อมูลเหล่านี้. จุดข้อมูลเหล่านี้จะเรียกว่า ชัพพอร์ตเวกเตอร์ (support vectors) ซึ่งเป็นที่มาของชื่อ ชัพพอร์ตเวกเตอร์แมชชีน. จุดข้อมูลอื่น ๆ ที่อยู่ลึกลงไปในเขตของกลุ่ม เป็นจุดข้อมูลที่แบ่งแยกได้ยากกว่า จะไม่มีบทบาทในการกำหนดอภิรานา.

ปัญหาการจำแนกค่าทวิภาคในทางปฏิบัติ อาจจะมีขอบเขตของการแบ่ง ที่ซับซ้อนกว่าสถานการณ์ในรูปที่ 4.15 ซึ่งสามารถแบ่งกลุ่มได้อย่างสมบูรณ์ด้วยอภิรานา. ในที่นี้ พิจารณาการพัฒนาชัพพอร์ตเวกเตอร์แมชชีน สำหรับกรณีที่ข้อมูลสามารถแบ่งแยกได้สมบูรณ์ก่อน และหัวข้อ 4.2 อภิรายการพัฒนาขยายความสามารถสำหรับกรณีทั่วไป (ซึ่งรวมถึงสถานการณ์ที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์).

Datapoints and hyperplane in feature space



รูปที่ 4.15 ช่องว่างของการแบ่งกลุ่ม. จุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสำคัญ (จุดกากราฟสีแดงกลุ่มลบ และจุดดาวสีเขียวกลุ่มบวก) และอภิรະนาบทั้งสอง (เส้นทึบสีน้ำเงิน). เส้นประสีฟ้าเขียวแสดงขอบเขตของการแบ่ง. ระยะจากอภิรະนาบทั้งสองของการแบ่งแสดงในรูปด้วยลูกศรสีม่วง. จุดข้อมูลที่อยู่บนแนวขอบเขตของการแบ่ง เน้นด้วยสีเหลืองรอบ ๆ จุด.

การหาอภิรະนาบ. สำหรับอภิรະนาบ $\mathbf{w}^T \mathbf{z} + b = 0$ และข้อมูลฝึก $\{\mathbf{x}_i, y_i\}_{i=1,\dots,N}$ ซึ่งเทียบเท่า $\{\mathbf{z}_i, y_i\}$ โดย $\mathbf{z}_i = \phi(\mathbf{x}_i)$ แล้ว อภิรະนาบที่แบ่งแยกข้อมูลได้อย่างสมบูรณ์ คือ อภิรະนาบที่มีค่าพารามิเตอร์ \mathbf{w} กับ b ที่ทำให้

$$\begin{aligned}\mathbf{w}^T \mathbf{z}_i + b &> 0 & \text{เมื่อ } y_i = 1 \\ \mathbf{w}^T \mathbf{z}_i + b &< 0 & \text{เมื่อ } y_i = -1\end{aligned}\tag{4.7}$$

สำหรับ $i = 1, \dots, N$ โดย N เป็นจำนวนข้อมูลฝึก.

เพื่อความสะดวก กำหนดฟังก์ชันแบ่งแยก f เป็น

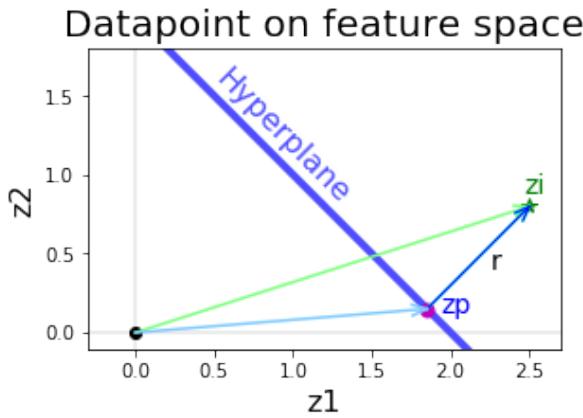
$$f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} + b.\tag{4.8}$$

ผลทายกลุ่ม หรือผลตัดสินใจ \hat{y} สามารถคำนวณได้จาก

$$\hat{y} = \begin{cases} 1 & \text{เมื่อ } f(\mathbf{z}) > 0, \\ -1 & \text{เมื่อ } f(\mathbf{z}) < 0. \end{cases}\tag{4.9}$$

นอกจากแบ่งแยกข้อมูลได้อย่างสมบูรณ์แล้ว เรายังต้องการให้ขอบเขตของการแบ่งกว้างที่สุดด้วย. พิจารณาจะต้องหาน้ำหนักที่ดีที่สุด ดูรูปที่ 4.16 ประกอบ. เมื่อแทนจุดใด ๆ ในปริภูมิด้วยเวกเตอร์จากจุดกำเนิดไปจุดนั้น เวกเตอร์ \mathbf{z}_i สามารถเขียนในรูปส่วนประกอบได้ว่า

$$\mathbf{z}_i = \mathbf{z}_p + r\vec{u}$$



รูปที่ 4.16 ระยะไปสู่จุดข้อมูลใด ๆ จากอภิรະนาบแทนด้วย r ซึ่งคือขนาดของเวกเตอร์จากจุด z_p ไป z_i . จุดข้อมูลใด ๆ z_i แสดงด้วยดาวสีเขียวเข้ม (บางส่วนถูกบัง). อภิรະนาบ แสดงด้วยเส้นหนาสีน้ำเงิน. จุด z_p (จุดกลมสีม่วง บางส่วนถูกบัง) คือจุดที่ถูกฉายจาก z_i ลงบนอภิรະนาบ. เวกเตอร์จากจุดกำหนดไป z_i (เวกเตอร์สีเขียว) เท่ากับเวกเตอร์จากจุดกำหนดไป z_p (เวกเตอร์สีฟ้าอ่อน) บวกกับเวกเตอร์จาก z_p ไป z_i (เวกเตอร์สีฟ้าเข้ม).

เมื่อ z_p คือจุดภาพฉายเชิงตั้งจากจุด z_i ลงบนอภิรະนาบ และ $\vec{u} = \mathbf{w}/\|\mathbf{w}\|$ คือเวกเตอร์หนึ่งหน่วยในทิศทางตั้งจากกับอภิรະนาบ และ r คือระยะห่างระหว่างจุด z_i กับอภิรະนาบ. ดังนั้น จุดใด ๆ $z_i = z_p + r\mathbf{w}/\|\mathbf{w}\|$ และค่าฟังก์ชันแบ่งแยก

$$\begin{aligned} f(z_i) &= f(z_p + r\mathbf{w}/\|\mathbf{w}\|) = \mathbf{w}^T \cdot (z_p + r\mathbf{w}/\|\mathbf{w}\|) + b \\ &= \mathbf{w}^T z_p + b + r\|\mathbf{w}\|^2/\|\mathbf{w}\| = r\|\mathbf{w}\| \\ r &= \frac{f(z_i)}{\|\mathbf{w}\|}. \end{aligned} \quad (4.10)$$

นั่นคือ ระยะห่างระหว่างจุดใด ๆ z_i กับอภิรະนาบจะเท่ากับค่าฟังก์ชันแบ่งแยกหารด้วยขนาดของ \mathbf{w} . ถ้า $f(z_i) = 0$ ก็คือระยะห่าง $r = 0$ จุดอยู่บนอภิรະนาบ. ถ้า $f(z_i) > 0$ และจุดนั้นอยู่ห่างจากอภิรະนาบตามคำนวณ ไปทางฝั่งกลุ่มบวก. ถ้า $f(z_i) < 0$ และจุดนั้นอยู่ห่างจากอภิรະนาบไปทางฝั่งกลุ่มลบ. หมายเหตุ จุดกำหนด $\mathbf{0}$ จะอยู่ห่างจากอภิรະนาบ $r = f(\mathbf{0})/\|\mathbf{w}\| = b/\|\mathbf{w}\|$. สังเกตว่า ระยะ r คือระยะจากอภิรະนาบไปจุดใด ๆ ซึ่งเมื่อพิจารณาที่จุดกำหนด ระยะ r กับระยะจากจุดกำหนดไปนานา d (ที่อภิปรายตอนต้นหัวข้อ) จะกลับทิศทางกัน.

พารามิเตอร์ของอภิรະนาบที่ทำให้ขอบเขตของการแบ่งกวนหัวใจสุด อาจเป็นเงื่อนไขที่สำคัญมาก

$$\begin{aligned} \mathbf{w}^T z_i + b &\geq +1 \quad \text{สำหรับ } y_i = +1 \\ \mathbf{w}^T z_i + b &\leq -1 \quad \text{สำหรับ } y_i = -1 \end{aligned} \quad (4.11)$$

ในสถานการณ์ที่ข้อมูลสามารถแบ่งแยกได้โดยสมบูรณ์ ค่าพารามิเตอร์ \mathbf{w} และ b ที่ได้ สามารถนำไปปรับขนาดโดยคูณค่าคงที่เข้าไป เพื่อให้เงื่อนไขในสมการ 4.11 เป็นจริงได้. จุดข้อมูล i^{th} ที่ทำให้

เงื่อนไขในสมการ 4.11 ทำงาน³ จะอยู่บนแนวขอบเขตของการแบ่ง และจุดเหล่านี้จะเรียกว่า ชัพพอร์ตเวกเตอร์. (ดูรูปที่ 4.15 ประกอบ). ค่าฟังก์ชันแบ่งแยกของชัพพอร์ตเวกเตอร์ 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 คือค่าลักษณะสำคัญและเฉลยของชัพพอร์ตเวกเตอร์ตัวที่ i^{th} .

ระยะจากอภิริยานไปชัพพอร์ตเวกเตอร์ \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

$$y_i \cdot (\mathbf{w}^T \mathbf{z}_i + b) \geq 1. \quad (4.12)$$

นั่นคือ กรอบปัญหาการฝึกชัพพอร์ตเวกเตอร์แมชชีน สามารถเขียนได้เป็น

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq 1 \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.13)$$

ฟังก์ชันจุดประสงค์ของชัพพอร์ตเวกเตอร์แมชชีน เป็นคอนเวกซ์ (convex function) และข้อจำกัดเป็นฟังก์ชันเชิงเส้น ซึ่งเหล่านี้ล้วนเป็นคุณสมบัติที่ดี จากมุมมองของการแก้ปัญหาค่าตัดที่สุด (หัวข้อ 2.3) เพราะว่า ลักษณะเหล่านี้ ทำให้ เมื่อแก้ปัญหาและพบค่าทำให้น้อยที่สุดท้องถิ่นแล้ว ค่าทำให้น้อยที่สุดท้องถิ่นจะเป็น ค่าทำให้น้อยที่สุดทั่วหมดด้วย.

การใช้งานชัพพอร์ตเวกเตอร์แมชชีนในทางปฏิบัติ จะไม่แก้ปัญหานี้โดยตรง แต่จะใช้คุณสมบัติของภาวะคู่กัน (ดูแบบฝึกหัด 2.28 เพิ่มเติม) เพื่อแปลงปัญหาในนิพจน์ 4.13 ซึ่งเป็นปัญหาบัญญาต ไปอยู่ในรูปปัญหาคู่⁴ ซึ่งจะสามารถใช้งานได้มีประสิทธิภาพกว่า.

ปัญหาคู่. จากวิธีการกรานจ์[39] จุดประสงค์และข้อจำกัดที่ระบุด้วยนิพจน์ 4.13 จะสามารถแทนได้ด้วย ลagrajan's ฟังก์ชัน

$$J(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i \cdot (y_i \cdot (\mathbf{w}^T \mathbf{z}_i + b) - 1) \quad (4.14)$$

³ ในทฤษฎีการหาค่าตัดที่สุด เงื่อนไขอสมการหรือข้อจำกัดอสมการ (inequality constraint) เมื่อนำมาเขียนในรูป $g(x) \geq 0$ จะเรียกว่า ทำงาน (active) ที่ค่า x_0 ต้า $g(x_0) = 0$. 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 $\mathbf{w}^T \mathbf{z}_i + b - 1 = 0$.

เมื่อ $\alpha = [\alpha_1, \dots, \alpha_N]^T$ เป็นลักษณะพารามิเตอร์ โดย $\alpha_i \geq 0$ สำหรับ $i = 1, \dots, N$.

จากทฤษฎีบทคุณทักษะเกอร์ (ดูแบบฝึกหัด 2.26) ที่กล่าวว่า หากกำหนดให้ w_o และ b_o แทน ชุดค่าพารามิเตอร์ที่ดีที่สุด (ค่าทำให้น้อยที่สุด) และ ณ จุดที่ดีที่สุด เงื่อนไขต่อไปนี้จะต้องเป็นจริง. เงื่อนไขที่หนึ่ง คือ $\alpha_i \geq 0$ สำหรับทุก ๆ ค่าของ i . เงื่อนไขที่สอง คือ

$$\nabla_{\mathbf{w}} J(\mathbf{w}_o, b_o, \alpha) = 0$$

$$\nabla_{b} J(\mathbf{w}_o, b_o, \alpha) = 0$$

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 เรา รู้แล้วว่า $y_i (\mathbf{w}_o^T \mathbf{z}_i + b_o) - 1 = 0$. นั่นคือข้อจำกัดทำงาน ซึ่งหมายถึง จุดข้อมูลที่ i^{th} เป็น ชั้พพอร์ตเวกเตอร์.

กำหนดให้ J' แทนลักษณะฟังก์ชัน เมื่อใช้ชุดค่าพารามิเตอร์ที่ดีที่สุด (ใช้ w_o และ b_o) พร้อม แทนค่าจากสมการ 4.15 และ 4.16 จะได้

$$J'(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{z}_i^T \mathbf{z}_j$$

เมื่อ $\alpha \geq 0$ และ $\sum_{i=1}^N \alpha_i y_i = 0$. กำหนดให้ ฟังก์ชันเครอร์เนล $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x})_i^T \phi(\mathbf{x})_j = \mathbf{z}_i^T \mathbf{z}_j$.

ดังนั้นปัญหาคือ สามารถระบุได้เป็น

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \\ & \alpha \geq 0 \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.17)$$

สังเกตว่า (1) ปัญหาปัญมเป็นปัญหาค่าน้อยที่สุด แต่ปัญหาคุ้เป็นปัญหาค่ามากที่สุด⁴. (2) ปัญหาคุ้อยู่ในรูปของตัวแปร α เท่านั้น ไม่มี w ไม่มี b .

หาก α^* เป็นผลการคำนวณตามเงื่อนไขที่ได้ที่สุดที่หาได้มา แล้วการคำนวณกลุ่มของจุดข้อมูล x สามารถคำนวณได้โดยค่าฟังก์ชันแบ่งแยก $f(\phi(x)) = w_o^T \phi(x) + b_o$. เพื่อความสะดวก นิยาม $g(x) = w_o^T \phi(x) + b_o$. เมื่อแทนค่าสมการ 4.15 เข้าไปแล้วจะได้

$$\begin{aligned} g(x) &= \sum_{i=1}^N \alpha_i^* y_i \phi(x_i)^T \phi(x) + b_o \\ &= \sum_{i=1}^N \alpha_i^* y_i k(x_i, x) + b_o \end{aligned} \quad (4.18)$$

เมื่อ x_i, y_i คือจุดข้อมูลฝึก.

เนื่องจากตัวแปร α_i^* ทำหน้าที่ของผลการคำนวณตามเงื่อนไขที่ได้ที่สุดที่ไม่ได้ทำงาน ซึ่งสัมพันธ์กับจุดข้อมูลที่อยู่ลึกลงไปในกลุ่ม ไม่ได้อยู่บริเวณขอบเขตของการแบ่ง ไม่ใช่ชัพพอร์ตเวกเตอร์ค่า α_i^* ของจุดข้อมูลเหล่านั้นจะเป็นศูนย์ (เงื่อนไขที่สามและที่หนึ่งของคารูชคุณทักษะ). สำหรับ $\alpha_i^* = 0$ ไม่ได้ส่งผลต่อการคำนวณสมการ 4.18 เลย. ดังนั้น การใช้ชัพพอร์ตเวกเตอร์แมชีนอัตโนมานกลุ่ม จึงไม่จำเป็นต้องใช้ข้อมูลทุกตัว ใช้เฉพาะชัพพอร์ตเวกเตอร์ก็พอ. นั่นคือ ค่าการอนุมานกลุ่ม คำนวณจาก

$$g(x) = \sum_{i \in S} \alpha_i^* y_i k(x_i, x) + b_o \quad (4.19)$$

เมื่อ S คือ เซตของตัวชี้ของชัพพอร์ตเวกเตอร์ นั่นคือ $S = \{i : \alpha_i^* > 0\}$.

สำหรับค่าของพารามิเตอร์ b_o พิจารณาจากชัพพอร์ตเวกเตอร์ ($i \in S$) ที่มี $\alpha_i^* > 0$ และจากทฤษฎีบทของคารูชคุณทักษะทำให้รู้ว่า ถ้า $\alpha_i^* > 0$ หมายถึง เงื่อนไข $y_i(w_o^T z_i + b_o) \geq 0$ ทำงาน. นั่นคือ สำหรับ $i \in S$ แล้ว

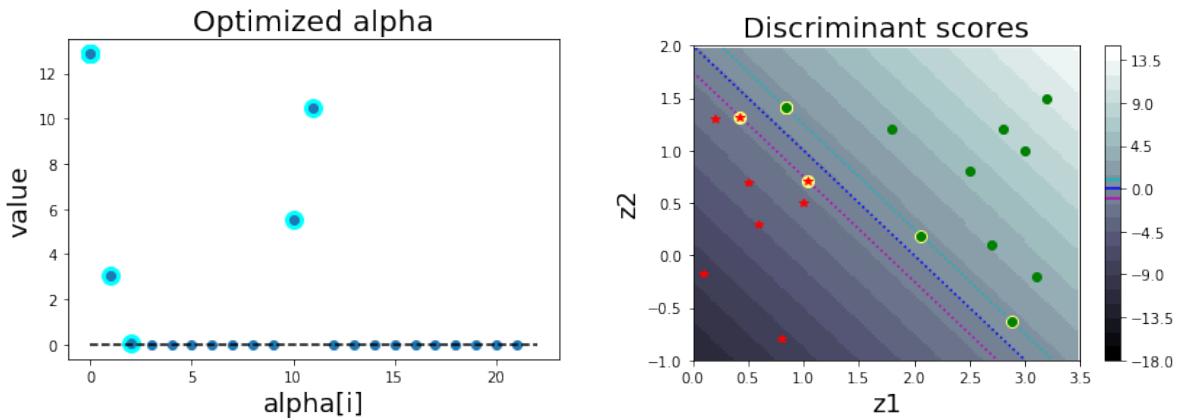
$$y_i \cdot g(x_i) = 1 \quad (4.20)$$

$$y_i \cdot \left(\sum_{j \in S} \alpha_j^* y_j k(x_j, x_i) + b_o \right) = 1. \quad (4.21)$$

เมื่อคุณ y_i เข้าไปทั้งสองข้าง (ซึ่ง $y_i^2 = 1$) และจัดรูปใหม่จะได้

$$b_o = y_i - \sum_{j \in S} \alpha_j^* y_j k(x_j, x_i)$$

⁴พิจารณาฟังก์ชันจุดประสงค์ของปัญหาคุ้ จะเห็นว่าฟังก์ชันจุดประสงค์ของปัญหาคุ้ ได้รับอิทธิพลส่วนหนึ่งมาจากข้อจำกัดในปัญหาปัญม. ปัญหาปัญม ต้องการหาค่าที่ทำน้อยที่สุด ภายใต้ข้อจำกัด. ปัญหาคุ้ ต้องการหาค่ามากที่สุด เพื่อจะรักษาข้อจำกัดปัญมไว้ ได้โดยไม่ทำร้ายจุดประสงค์ปัญม.



รูปที่ 4.17 ผลลัพธ์ของชัพพอร์ตเวกเตอร์เมชชีน. ภาพซ้าย แสดงค่า α ที่ได้จากการฝึก. $\alpha_i > 0$ เน้นด้วยสีฟ้าเขียวรอบ ๆ. เส้นประสีดำ แสดงแนวของค่าศูนย์. ภาพขวา แสดงค่าฟังก์ชันแบ่งแยกของชัพพอร์ตเวกเตอร์เมชชีนในปริภูมิลักษณะสำคัญ. ค่าฟังก์ชันแบ่งแยกเป็นค่าต่อเนื่อง แต่ในภาพค่าฟังก์ชันแบ่งแยกแสดงด้วยระดับสีเทา 18 ระดับ ซึ่งค่าระบุด้วยແບสีด้านข้าง. เส้นประสีม่วง สีน้ำเงิน และสีฟ้าเขียว แสดงแนวที่ค่าฟังก์ชันแบ่งแยกเป็น $-1, 0$, และ 1 ตามลำดับ. จุดข้อมูลฝึก แสดงด้วยวงกลมสีเขียว (กลุ่มขวา) และดาวสีแดง (กลุ่ม左). จุดข้อมูลฝึกที่ถูกเลือกเป็นชัพพอร์ตเวกเตอร์ เน้นด้วยสีเหลืองรอบ ๆ.

การคำนวณ อาจสุ่มเลือกดัชนี 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.

ตัวอย่างผลลัพธ์จากการฝึกชัพพอร์ตเวกเตอร์เมชชีน แสดงในรูปที่ 4.17. ภาพซ้ายเป็นค่า α ที่ได้จากการฝึก. ภาพขวาแสดงค่าฟังก์ชันแบ่งแยก ที่คำนวณจากสมการ 4.19. ชัพพอร์ตเวกเตอร์ คือจุดข้อมูลที่เน้น ดังแสดงในภาพขวา ซึ่งระบุได้จากค่า α_i ที่สัมพันธ์กับมันมีค่ามากกว่าศูนย์.

สถานการณ์ที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์

ในทางปฏิบัติ การแยกของกลุ่มข้อมูลอาจทำให้บริเวณของข้อมูลมีการซ้อนทับกันได้. สมมติฐาน การแบ่งแยกได้อย่างสมบูรณ์ จะจะทำให้ได้แบบจำลองที่การโอเวอร์ฟิต ขาดคุณสมบัติความทั่วไป. ดังนั้น เพื่อผ่อนสมมติฐานการแบ่งแยกได้อย่างสมบูรณ์ ควรจะยอมให้มีบางจุดข้อมูลที่อาจล้ำเข้าไป ในขอบเขตของการแบ่งบ้าง หรือแม้แต่ยอมให้มีบางจุดข้อมูลที่ถูกจำแนกกลุ่มผิดบ้าง.

ชัพพอร์ตเวกเตอร์เมชชีนผ่อนปรนสมมติฐานการแบ่งแยกสมบูรณ์ลง ด้วยการผ่อนปรนข้อจำกัดของขอบเขตของการแบ่ง (อสมการ 4.11) ผ่านกลไกของตัวแปรช่วย ξ_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.

ดังนั้น ถ้า $\xi_i = 0$ หมายถึง จุดข้อมูลที่ i^{th} จะอยู่ห่างจากเส้นแบ่งอยู่ต่ำกว่าเส้นแบ่งจริง แต่ไม่ห่างจากเส้นแบ่งมากเกินไป. รูปที่ 4.18 แสดงตัวอย่างต่าง ๆ เมื่อผ่อนปรนเงื่อนไขแบ่งแยกสมบูรณ์ลง. โดยในรูป จุด b, c และ d จะมี $\xi_b, \xi_c, \xi_d = 0$. ถ้า $\xi_i > 0$ หมายถึง จุดข้อมูลที่ i^{th} อยู่ล้ำแนวของขอบเขตของการแบ่งออกไว. ในรูป จุด a, e จะมี $\xi_a, \xi_e > 0$.

พิจารณากรณีที่ $0 < \xi_i < 1$ นั่นคือ จุดข้อมูลอยู่ล้ำแนวของขอบเขตของการแบ่งออกไว แต่ยังไม่ถูกกำหนดเป็นจุดเดียว เช่น จุด e ในรูป จะมี $0 < \xi_e < 1$. และเนื่องจากจุดข้อมูลยังอยู่ผิดของกลุ่มอยู่ จุดข้อมูลที่มี $0 < \xi_i < 1$ จะยังถูกจำแนกได้ถูกต้อง. กรณีที่ $\xi_i = 1$ นั่นคือ จุดข้อมูลอยู่ล้ำแนวของขอบเขตของการแบ่งออกไว และไปอยู่บนเส้นแบ่งจริง พอดี เช่น ในรูป $\xi_d = 1$. กรณีที่ $\xi_i > 1$ นั่นคือ จุดข้อมูลอยู่ล้ำแนวของขอบเขตของการแบ่งออกไวมาก มากจนเลยแนวของเส้นแบ่ง ข้ามไปอยู่อีกฝั่งของการจำแนก ดังนั้น จุดข้อมูลจะถูกจำแนกผิด เช่น ในรูป $\xi_a > 1$.

เงื่อนไขในสมการ 4.23 เขียนให้กระชับขึ้นเป็น

$$y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq 1 - \xi_i \quad \text{สำหรับ } i = 1, \dots, N \quad (4.24)$$

เมื่อ $\xi_i \geq 0$. การปรับเงื่อนไขนี้ เปรียบเสมือนการปรับจากข้อจำกัดที่เข้มงวด ผ่อนปรนลงมาเป็นข้อจำกัดที่อ่อนลง. ครอบปัญหา จึงถูกกว้างใหม่เป็น

$$\begin{aligned} \underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \text{for } i = 1, \dots, N \end{aligned} \quad (4.25)$$

เมื่อ $C > 0$. อกีมานพารามิเตอร์ C เป็นเหมือนค่าที่ใช้ควบคุมความเข้มงวดของข้อจำกัด. ค่า C ที่เล็กจะยอมให้มีการจำแนกผิดได้มากขึ้น ในขณะที่ค่า C ใหญ่จะบังคับให้แบบจำลองจำแนกผิดให้น้อยลง.

ปัญหาครุ. ในทำนองเดียวกัน ลากกราฟพังก์ชันของปัญหาบูรณา (นิพจน์ 4.25) คือ

$$J(\mathbf{w}, b, \xi, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \cdot (y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i$$

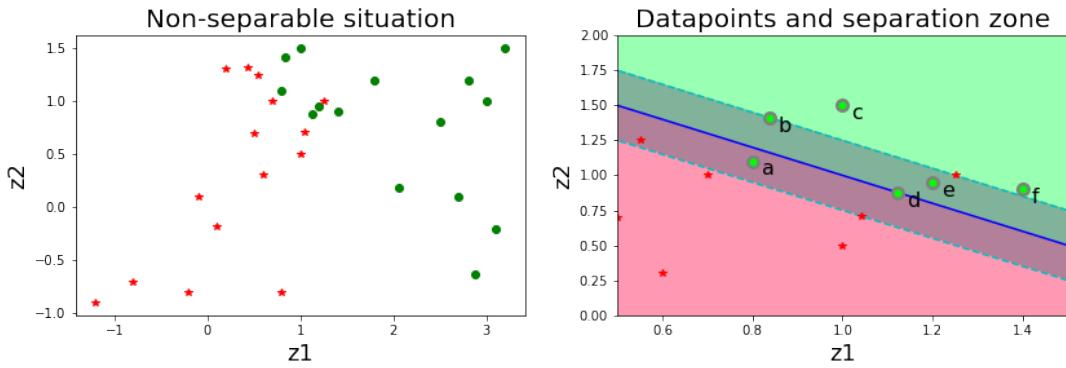
โดย $\alpha_i, \beta_i \geq 0$. ทั้ง α_i และ β_i เป็นลากกราฟพารามิเตอร์.

จากทฤษฎีบทค่ารูซคุณทักษะ ณ จุดที่ดีที่สุด ค่าพารามิเตอร์ที่ดีที่สุด \mathbf{w}_o, b, ξ_o จะทำให้เงื่อนไขดังนี้เป็นจริง. เงื่อนไขที่หนึ่ง $\alpha_i \geq 0$ และ $\beta_i \geq 0$. เงื่อนไขที่สอง

$$\nabla_{\mathbf{w}} J(\mathbf{w}_o, b, \xi_o) = 0$$

$$\nabla_b J(\mathbf{w}_o, b, \xi_o) = 0$$

$$\nabla_{\xi} J(\mathbf{w}_o, b, \xi_o) = 0.$$



รูปที่ 4.18 ข้อมูลที่ไม่สามารถแบ่งแยกกลุ่มได้สมบูรณ์. ภาพซ้าย แสดงจุดข้อมูลต่าง ๆ ในปริภูมิลักษณะสำคัญ. จุดข้อมูลไม่สามารถถูกแบ่งแยกกลุ่มได้อย่างสมบูรณ์ด้วยอวิรานาบ. วงกลมสีเขียว แทนจุดข้อมูลของกลุ่มบวก. ดาวสีแดง แทนจุดข้อมูลของกลุ่มลบ. ภาพขวา แสดงจุดข้อมูลกับบริเวณของการแบ่งต่าง ๆ. เส้นทึบสีน้ำเงิน แทนอวิรานาบ. เส้นประสีฟ้าเขียว แทนแนวขอของขอบเขตของการแบ่ง. บริเวณพื้นหลังสีเขียวอ่อน แทนบริเวณของกลุ่มบวก. บริเวณพื้นหลังสีเขียวมะกอก แทนบริเวณที่อยู่ภายใต้ขอบเขตของการแบ่งฝั่งกลุ่มลบ. บริเวณพื้นหลังสีชมพูอ่อน แทนบริเวณของกลุ่มลบ. บริเวณพื้นหลังสีม่วง แทนบริเวณที่อยู่ภายใต้ขอบเขตของการแบ่งฝั่งกลุ่มลบ. จุดข้อมูล a เป็นจุดข้อมูลกลุ่มบวก ที่ตำแหน่งไปอยู่ในฝั่งของกลุ่มลบ จุดนี้จะถูกจำแนกผิดเป็นกลุ่มลบ. จุดข้อมูล b อยู่พอดีบนแนวขอของขอบเขตของการแบ่งฝั่งกลุ่มบวก. จุดข้อมูล c และ f อยู่ลึกลงไปในบริเวณของกลุ่มบวก. จุดข้อมูล d อยู่พอดีบนอวิรานาบแบ่ง. จุดข้อมูล e อยู่ล้ำแนวขอมาจนเข้าไปอยู่ในขอบเขตของการแบ่ง แต่ยังอยู่ในฝั่งของกลุ่มบวก.

หลังจากหาอนุพันธ์และแก้สมการแล้ว สรุปได้ว่า

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i) \quad (4.26)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (4.27)$$

$$C - \alpha_i - \beta_i = 0 \text{ for } i = 1, \dots, N. \quad (4.28)$$

สมการ 4.28 คือ $\beta_i = C - \alpha_i$. แทนค่าเหล่านี้ เข้าไปในลากรานจ์ฟังก์ชันแล้ว ลากรานจ์ฟังก์ชัน จะที่ดีที่สุด J' สามารถเขียนได้ว่า

$$J'(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (4.29)$$

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 $\alpha_i \leq C$. ดังนั้นปัญหาคุ้มครองจะได้เป็น

$$\begin{aligned} & \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{s.t.} \\ & \sum_{i=1}^N \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.30)$$

สังเกตว่า ปัญหาคู่สำหรับกรณีทั่วไป แทนจะเหมือนกับปัญหาคู่กรณีข้อมูลแบ่งแยกได้โดยสมบูรณ์เลย ต่างกันเพียงแต่เงื่อนไขของค่า α_i ที่เปลี่ยนมาเป็น $0 \leq \alpha_i \leq C$.

การอนุมานค่าฟังก์ชันแบ่งแยกก็ทำได้โดยการคำนวณสมการ 4.18 เช่นเดิม. และค่าพารามิเตอร์ b_o ในกรณีที่ข้อมูลไม่สามารถแบ่งแยกได้สมบูรณ์ สามารถพิจารณาจาก เงื่อนไข $y_i (\mathbf{w}^T \mathbf{z}_i + b) \geq 1 - \xi_i$ (สมการ 4.24) ที่เมื่อ $\alpha_i > 0$ แล้ว เงื่อนไขจะทำงาน. นั่นคือ $y_i (\mathbf{w}^T \mathbf{z}_i + b) = 1 - \xi_i$. แต่เรายังไม่สามารถแก้สมการนี้ได้ เพราะเรายังไม่รู้ค่า ξ_i . อย่างไรก็ตาม จากการที่ $\xi_i \geq 0$ เป็นเงื่อนไข ที่ควบคุมด้วยลักษณะพารามิเตอร์ β_i . นั่นคือ เรารู้ว่า เมื่อ $\beta_i > 0$ (เทียบเท่า $\alpha_i < C$) แล้ว เงื่อนไข จะทำงาน ซึ่งคือ $\xi_i = 0$. ดังนั้น จุดข้อมูลที่ $0 < \alpha_i < C$ จะบอกได้ว่า $y_i (\mathbf{w}^T \mathbf{z}_i + b) = 1$ ซึ่งเรา สามารถใช้จุดข้อมูลเหล่านี้ แก้สมการหาค่า b_o ได้. นั่นคือ

$$y_i \cdot g(\mathbf{x}_i) = 1 \text{ เมื่อ } i \in \{j : 0 < \alpha_j < C\} \quad (4.31)$$

$$\begin{aligned} \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) + b_o &= y_i \\ b_o &= y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i). \end{aligned} \quad (4.32)$$

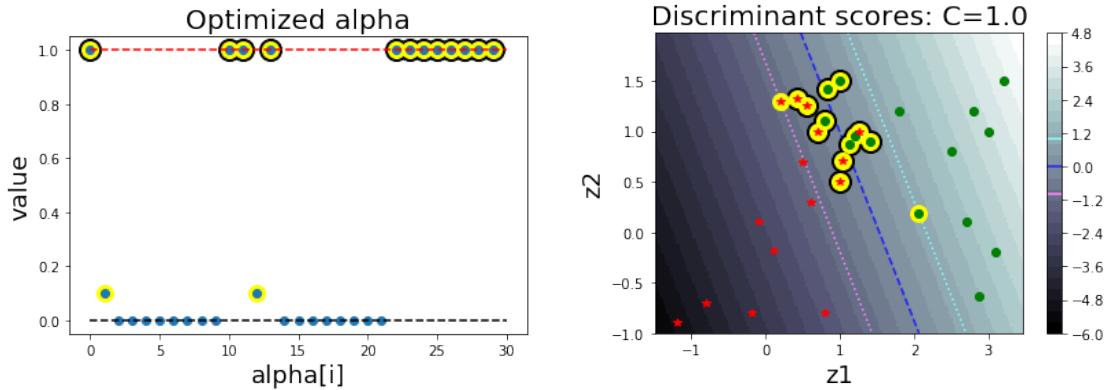
เช่นเดียวกับ i อาจเลือกจากตัวนี้หนึ่ง ซึ่งทำให้ $0 < \alpha_i < C$ หรือ อาจใช้ค่าเฉลี่ย ซึ่งคือ

$$b_o = \frac{1}{|S'|} \sum_{i \in S'} \left(y_i - \sum_{j \in S} \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) \right) \quad (4.33)$$

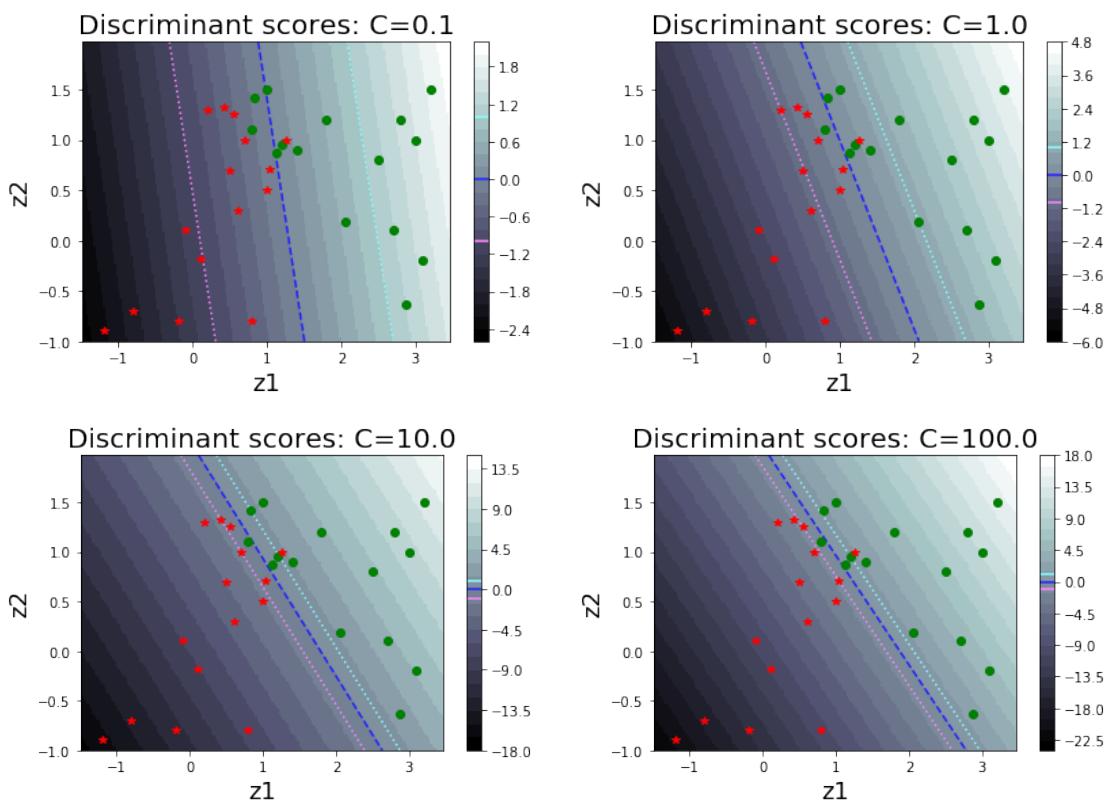
เมื่อ $S = \{i : \alpha_i > 0\}$ และ $S' = \{i : 0 < \alpha_i < C\}$ เป็นเซตตัวนี้ของชัพพอร์ตเวกเตอร์ และชัพพอร์ตเวกเตอร์ที่แนวขอบเขตของการแบ่ง ตามลำดับ. หมายเหตุ S มาจากสมการ 4.26 ที่คำนวณทุกตัว แต่ $\alpha_i = 0$ ไม่มีผล ดังนั้นจึงเลือกเฉพาะที่ $\alpha_i > 0$ มาคำนวณ เพื่อลดการคำนวณที่ไม่จำเป็นและ ลดข้อมูล (\mathbf{x}_i, y_i) ที่ต้องเก็บรักษาไว้. ส่วน S' มาจากทฤษฎีบทครูชุนท์ก์เกอร์ที่ทำให้อสมการ 4.24 เปลี่ยนมาอยู่ในรูปสมการ 4.31 เพื่อทำให้สามารถคำนวณค่า b_o ได้.

รูปที่ 4.19 แสดงค่า α_i ต่าง ๆ ที่ฝึกเสริจ (ภาพซ้าย) และค่าฟังก์ชันแบ่งแยก (ภาพขวา). รูปที่ 4.20 แสดงตัวอย่างของพฤติกรรมการจำแนกของแบบจำลอง เมื่อเลือกค่า C ต่าง ๆ. สังเกตว่า ที่ค่า C ขนาดเล็ก จะเห็นขอบเขตของการแบ่งกว้าง และมีจุดข้อมูลล้าแนวขอบเขตของการแบ่งจำนวนมาก. ที่ค่า C ขนาดใหญ่ ฟังก์ชันแบ่งแยกจะปรับการคำนวณ เพื่อให้จุดข้อมูลล้าแนวขอบเขตของการแบ่งออกไปน้อยลง แต่ก็ทำให้ขอบเขตของการแบ่งแคบลง.

นอกจาก ชัพพอร์ตเวกเตอร์แมชีนในรูปแบบดังเดิม ที่ระบุด้วยนิพจน์ 4.30 ยังมีรูปแบบที่พัฒนา



รูปที่ 4.19 ผลการฝึกชั้พพอร์ตเวกเตอร์เมชชีนกรณีทั่วไป. ภาพซ้าย แสดงค่า α_i ที่ดัชนี i ต่าง ๆ. เส้นประสีดា และเส้นประสีแดง แสดงแนวศูนย์ และแนวค่า C ซึ่งเป็นขอบของช่วงค่าที่อนุญาตสำหรับ α_i . ค่า $\alpha_i > 0$ เน้นด้วยสีเหลือง และค่า $\alpha_i = C$ เน้นด้วยขอบสีดำอีกที. ภาพขวา แสดงจุดข้อมูลและค่าฟังก์ชันแบ่งเขตในปริภูมิลักษณะสำคัญ. ชัพพอร์ตเวกเตอร์ ($\alpha_i > 0$) เน้นด้วยสีเหลือง และชัพพอร์ตเวกเตอร์ที่มี $\alpha_i = C$ เน้นด้วยขอบสีดำอีกที.



รูปที่ 4.20 พฤติกรรมของชัพพอร์ตเวกเตอร์เมชชีนที่ค่า C ต่าง ๆ. แต่ละภาพ แสดง จุดข้อมูล (วงกลมสีเขียว แทนจุดข้อมูลกลุ่มบวก. ดาวสีแดง แทนจุดข้อมูลกลุ่มลบ) ค่าฟังก์ชันแบ่งเขต (เส้นที่แสดงในระบบระดับสีเทา โดยค่าของสีแสดงด้วยแบบสีด้านข้าง) อภิรະนา (เส้นประสีน้ำเงิน) และแนวขอบเขตของการแบ่ง (เส้นประสีม่วงอ่อน สำหรับแนวผั่งลบ และเส้นประสีฟ้าเขียว สำหรับแนวผั่งบวก). ค่าพารามิเตอร์ C แสดงอยู่เหนือภาพ.

ขึ้นมาใหม่อีก เช่น นูชพอร์ตเวกเตอร์แมชชีน (V -SVM[181]) ที่ทางกรอบปัญหาเป็น

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \\ & \sum_{i=1}^N \alpha_i \geq v, \\ & 0 \leq \alpha_i \leq 1/N \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (4.34)$$

โดย v เป็นอภิมานพารามิเตอร์ แทน C ในนิพจน์ 4.30.

ฟังก์ชันเครอร์เนล

ซัพพอร์ตเวกเตอร์แมชชีน จัดการการคำนวณอย่าง сложสลายที่ในการฝึก (นิพจน์ 4.30 และสมการ 4.33) และการอนุमาน (สมการ 4.19) สามารถทำงานโดยตรงกับฟังก์ชันเครอร์เนล (kernel function) $k(\mathbf{x}, \mathbf{x}')$ โดยไม่จำเป็นต้องอาศัยฟังก์ชันลักษณะสำคัญ $\phi(\mathbf{x})$. โอกาสเช่นนี้ ทำให้การใช้งานซัพพอร์ตเวกเตอร์แมชชีน สามารถกำหนดฟังก์ชันเครอร์เนล ที่เทียบเท่าการทำงานในปริภูมิลักษณะสำคัญที่มีจำนวนมิติมาก ๆ ได้ โดยไม่จำเป็นต้องเข้าไปทำงานในปริภูมิที่มีมิติสูงนั้นโดยตรง. การใช้ประโยชน์แห่งมุมนี้ 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 แต่ต้องตรวจสอบว่า ฟังก์ชันที่กำหนดนั้น มีคุณสมบัติเป็นฟังก์ชันเครอร์เนลได้.

การตรวจสอบนั้น อาจจะทำโดยใช้ทฤษฎีบทของเมอร์เซอร์ (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 $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. ฟังก์ชันจะมีคุณสมบัติเป็นฟังก์ชันเครอร์เนลได้ ก็ต่อเมื่อแกรมเมทริกซ์เป็นเมทริกซ์บวกแน่นอน⁵ (positive definite matrix) สำหรับทุก ๆ ค่าที่เป็นไปได้ของ \mathbf{x} และ \mathbf{x}' .

⁵ 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.

แต่ (3) วิธีที่สอดคลายกว่าในการสร้างฟังก์ชันเครื่องเรียน คือสร้างฟังก์ชันเครื่องเรียนใหม่ จากฟังก์ชันเครื่องเรียนเดิมที่ถูกตรวจสอบมาแล้ว ด้วยคุณสมบัติดังนี้ (จาก [15]). หาก $k_1(\mathbf{x}, \mathbf{x}')$ และ $k_2(\mathbf{x}, \mathbf{x}')$ มีคุณสมบัติเป็นฟังก์ชันเครื่องเรียนได้แล้ว ฟังก์ชันต่อไปนี้ก็จะมีคุณสมบัติเป็นเครื่องเรียนได้เช่นกัน

$$k(\mathbf{x}, \mathbf{x}') = ck_1(\mathbf{x}, \mathbf{x}') \quad (4.36)$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}') \quad (4.37)$$

$$k(\mathbf{x}, \mathbf{x}') = \text{polynomial}^+(k_1(\mathbf{x}, \mathbf{x}')) \quad (4.38)$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}')) \quad (4.39)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \quad (4.40)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \quad (4.41)$$

$$k(\mathbf{x}, \mathbf{x}') = k_3(g(\mathbf{x}), g(\mathbf{x}')) \quad (4.42)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}' \quad (4.43)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (4.44)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) \cdot k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (4.45)$$

เมื่อ $c > 0$ เป็นค่าคงที่. ฟังก์ชัน $f : \mathbb{R}^D \mapsto \mathbb{R}$ เป็นฟังก์ชันใด ๆ. ฟังก์ชัน polynomial^+ เป็นฟังก์ชันพหุนามที่สัมประสิทธิ์ไม่มีค่าลบ. ฟังก์ชัน $g : \mathbb{R}^D \mapsto \mathbb{R}^M$ และ k_3 มีคุณสมบัติเป็นฟังก์ชันเครื่องเรียนสำหรับ \mathbb{R}^M . เมทริกซ์ \mathbf{A} สมมาตร และเป็นบวกกึ่งແน่นอน⁶ 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 เป็นคุณสมบัติเป็นฟังก์ชันเครื่องเรียนสำหรับปริภูมิย่ออย่างกล่าว.

ฟังก์ชันเครื่องเรียนที่นิยม และแนะนำสำหรับการเริ่มต้นใช้งานชี้พพอร์ตเวกเตอร์เมชีน[32] ได้แก่ ฟังก์ชันเครื่องเรียนเชิงเส้น และฟังก์ชันเครื่องเรียนเกาส์เชียน.

ฟังก์ชันเครื่องเรียนเชิงเส้น (linear kernel) ที่นิยามเป็น

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' \quad (4.46)$$

ซึ่งคือ ฟังก์ชันลักษณะสำคัญเป็นฟังก์ชันเอกลักษณ์ $\phi(\mathbf{x}) = \mathbf{x}$. ฟังก์ชันเครื่องเรียนเชิงเส้น สร้างจากนิยามของเครื่องเรียนในสมการ 4.35.

ฟังก์ชันเครื่องเรียนเกาส์เชียน (Gaussian kernel) นิยามเป็น

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (4.47)$$

ฟังก์ชันเครื่องเรียนเกาส์เชียน อาจจะมองว่าสร้างมาจากคุณสมบัติของเครื่องเรียน. พิจารณา

$$\|\mathbf{x} - \mathbf{x}'\|^2 = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{x}' + (\mathbf{x}')^T \mathbf{x}'$$

⁶ คุณสมบัติบวกกึ่งແน่นอน (positive semidefinite).

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 ใช้ฟังก์ชันเครื่องเนลเชิงเด็น $\mathbf{x}^T \mathbf{x}'$ เป็นพื้นฐาน และใช้คุณสมบัติในสมการ 4.37, 4.39, และ 4.36 ประกอบ.

รูปที่ 4.21 แสดงค่าฟังก์ชันแบ่งแยก เมื่อใช้ฟังก์ชันเครื่องเนลเก้าส์เซียน ที่ค่า σ ต่าง ๆ. สังเกตว่า ในภาพสีน้ำค่าฟังก์ชันแบ่งแยกเป็นศูนย์ (ซึ่งสะท้อนถึงอภิรานนาบในปริภูมิลักษณะสำคัญ) สามารถโค้งเลี้ยวไปตามข้อมูลได้ในปริภูมิข้อมูล. ภาพต่างทางขวา แสดงขอบเขตตัดลินใจ (decision boundary) ซึ่งเป็นส่วนในปริภูมิข้อมูล ที่ข้อมูลที่อยู่ภายใต้บริเวณจะถูกตัดสินตามชนิดของขอบเขตตัดลินใจ.

รูปที่ 4.22 แสดงการใช้ฟังก์ชันเครื่องเนลเชิงเด็น เพื่อเปรียบเทียบ.

4.3 อภิธานศัพท์

การตรวจหาวัตถุ (object detection): การกิจการหาตำแหน่งของวัตถุในภาพ หากภาพมีวัตถุอยู่.

วิธีหน้าต่างเลื่อน (sliding window): วิธีการเลือกส่วนภาพขนาดที่กำหนดจากข้อมูลภาพใหญ่ โดย

การเลือกส่วนภาพ จะเลือกทั่วถึงจากทุกบริเวณในภาพใหญ่ ซึ่งอาจเริ่มจากมุมซ้ายบนของภาพใหญ่ เลือกส่วนภาพอ กมา แล้วขยับไปทางขวา และทำเช่นนี้ไปจนสุดปลายด้านขวา แล้วจึง ขยับลงล่างและไปเริ่มจากซ้ายสุด และทำลักษณะเช่นนี้อีก จนครอบคลุมบริเวณทั้งภาพใหญ่.

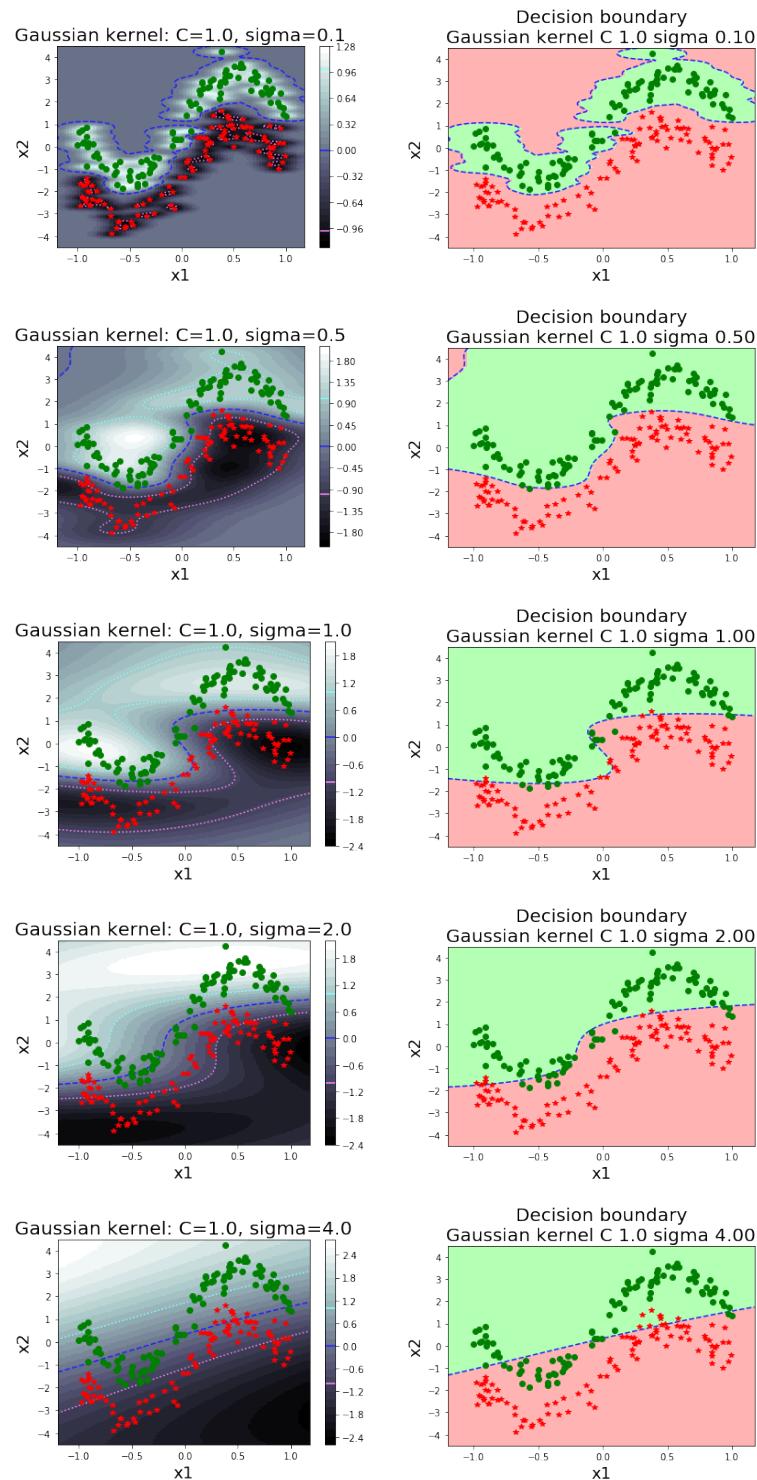
ขนาดขยับเลื่อน (stride): ขนาดของการเลื่อนหน้าต่างแต่ละครั้ง.

การสกัดลักษณะสำคัญ (feature extraction): การแปลงอินพุตตั้งเดิม ให้อยู่ในรูปแบบใหม่ โดยที่ รูปแบบใหม่นี้จะช่วยให้ภาระกิจที่ต้องการดำเนินการได้สะดวกขึ้น.

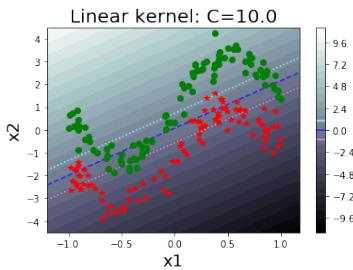
แบบจำลองแบ่งแยก (discriminative model): แบบจำลองการจำแนกกลุ่ม ที่อาศัยหรือตีความได้ว่าใช้ความน่าจะเป็นภายหลัง $\Pr(y|x)$ เมื่อ y เป็นค่ากลุ่มที่ต้องการทำนาย และ x เป็นอินพุต หรือตัวแปรต้น. ตัวอย่างเช่น โครงข่ายประสาทเทียม.

แบบจำลองสร้างกำเนิด (generative model): แบบจำลองการจำแนกกลุ่ม ที่อาศัยความน่าจะเป็น $\Pr(x|y)$ ทางตรงหรือทางอ้อม เมื่อ x เป็นอินพุตหรือตัวแปรต้น และ y เป็นค่ากลุ่ม. โดยทั่วไป แล้ว 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 ทำการคำนวณค่าเพื่อจำแนกกลุ่มโดยตรง ไม่อาศัยและไม่สามารถตีความในเชิงความน่าจะเป็น. ตัวอย่างเช่น ซัพพอร์ต เวกเตอร์แมชีน.



รูปที่ 4.21 การทำงานของชั้นพอร์ตเกรตเตอร์เมชชีน ด้วยเก้าส์เชียนเครื่องเรียนที่ค่า σ ต่าง ๆ. ค่า C และ σ จะระบุไว้เหนือภาพ. ภาพซ้ายแสดงค่าฟังก์ชันแบบแยกของชั้นพอร์ตเกรตเตอร์เมชชีนในปริภูมิข้อมูล. ค่าฟังก์ชันแบบแยกแสดงด้วยระดับสีเทา ซึ่งค่าจะบุนเดินตามข้าง. เส้นประสม่วง สีน้ำเงิน และสีฟ้าเขียว แสดงแนวที่ค่าฟังก์ชันแบบแยกเป็น $-1, 0$, และ 1 ตามลำดับ. จุดข้อมูลฝึก แสดงด้วยวงกลมสีเขียว (กลุ่มบวก) และดาวสีแดง (กลุ่มลบ). ภาพขวา แสดงจุดข้อมูลฝึก กับขอบเขตตัดสินใจสำหรับกลุ่มบวก แสดงด้วยสีเขียวอ่อน. ขอบเขตตัดสินใจสำหรับกลุ่มลบ แสดงด้วยสีชมพู.



รูปที่ 4.22 การทำงานของชั้พพอร์ตเวกเตอร์แมชชีน ด้วยเกอร์เนลเชิงเส้น เพื่อเปรียบเทียบกับเก้าส์เซียนในรูปที่ 4.21.

กล่องขอบเขต (bounding box): บริเวณสี่เหลี่ยมที่เป็นขอบเขตภายในภาพ ซึ่งแต่ละกล่องขอบเขต สามารถระบุได้ด้วยสี่ค่า เช่น พิกัด (x, y) มุมซ้ายบน และขนาดความกว้างกับความสูงของ กล่อง (w, h) .

การกำจัดการระบุซ้ำซ้อน (redundancy removal): กลไกที่สำคัญสำหรับการตรวจจับภาพวัตถุ เพื่อ กำจัดการระบุการตรวจพบตั้งแต่สองอันขึ้นไป ที่จริง ๆ แล้วระบุถึงวัตถุเดียวกัน.

วิธีระงับค่าไม่มากสุดท้องถิ่น (non-local-maximum suppression): วิธีหนึ่งในการกำจัดการระบุ ซ้ำซ้อน ที่ดำเนินการด้วยการตัดทิ้งกล่องขอบเขตที่มีค่าความเมะમะลงไม่นามากที่สุด เมื่อเปรียบ เทียบกับกล่องขอบเขตอื่น ๆ ที่อยู่รอบ ๆ กล่องนั้น โดย ค่าความเมะมะลง คือค่าที่ใช้วัดความ มั่นใจว่ากล่องขอบเขตนั้นมีวัตถุที่ค้นหาอยู่.

ไอโอ尤ย (IoU หรือ intersection of union): ปริมาณวัด ที่วัดจากสัดส่วนพื้นที่ซ้อนทับกันของกล่อง ขอบเขตสองกล่อง ต่อพื้นที่รวม เพื่อบอกความใกล้เคียงของตำแหน่งการตรวจจับ อาจใช้ใน กลไกของการตรวจจับ หรือใช้ในกระบวนการประเมินผล.

วิธีการประมาณความหนาแน่นแก่น (kernel density estimation): วิธีหนึ่งในการประมาณความ หนาแน่นความน่าจะเป็นของข้อมูล.

ค่าเฉลี่ยค่าประมาณความเที่ยงตรง (mean Average Precision หรือ mAP): วิธีการวัดความสา- มารถของระบบตรวจจับภาพวัตถุ ที่คำนวนจากค่าประมาณพื้นที่ได้กราฟของค่าความเที่ยง ตรงและค่าระลึกกลับของวัตถุชนิดต่าง ๆ และคำนวณมาเฉลี่ยกัน.

ปริภูมิลักษณะสำคัญ (feature space): ปริภูมิหรือเซตของค่าต่าง ๆ ที่เป็นไปได้ทั้งหมดของข้อมูล ในรูปแบบที่น่าจะช่วยให้ภาระกิจที่ต้องการทำได้ง่ายขึ้น.

ชัพพอร์ตเวกเตอร์แมชชีน (support vector machine): แบบจำลองจำแนกค่าทวิภาค ที่จำแนก ข้อมูล ด้วยการใช้อภิรະนาบในปริภูมิลักษณะสำคัญ. อภิรະนาบที่ใช้ ถูกเลือกมาจากอภิรະนาบ ที่สามารถแบ่งข้อมูลตัวอย่างได้ขอบเขตของการแบ่งกวนที่สุด.

อภิรະนาบ (hyperplane): ธนาบในปริภูมิหลายมิติ. สำหรับปริภูมิสองมิติ อภิรະนาบคือเส้นตรง. สำหรับปริภูมิสามมิติ อภิรະนาบคือแผ่นตรงเรียบ. ปริภูมิหลายมิติได ๆ (กี่มิติก็ตาม) อภิรະนาบ สามารถบรรยายได้ด้วยสมการ $\mathbf{w}^T \mathbf{x} + b = 0$ เมื่อ \mathbf{x} เป็นจุดใด ๆ ในปริภูมิ และ \mathbf{w} กับ b

ເປັນພາຣາມີເຕອີ່ຣ໌ຂອງອກົງຮານາບ.

ขอบเขตຂອງການແບ່ງ (margin of separation): ຄວາມໜ່າງທີ່ແບ່ງກລຸ່ມຂໍ້ອມູລສອງກລຸ່ມອອກຈາກກັນ (ຈຶ່ງຈາຈະແບ່ງໄດ້ສົມບູຮົນ ອີເວີໂມ່ສົມບູຮົນກີ່ຕາມ).

ขอบเขตຕັດສິນໃຈ (decision boundary): ບຣິເວັນໃນປຣິກູມີຂໍ້ອມູລ ທີ່ຈຸດຂໍ້ອມູລຕ່າງໆ ແກ້ວຍໆຢ່າງໃນ ບຣິເວັນຈະຖູກຈຳແນກຈົດຕາມໝົດຂອງຂອບເຂດຕັດສິນໃຈ.

ຫັ້ພພອ້ຣ໌ເວກເຕອີ່ຣ໌ (support vectors): ຈຸດຂໍ້ອມູລທີ່ສຳຄັຟຸ່ງຕ່າງໆກ່າວກົດວິທີການ.

ສູກເລ່ນເຄອົງເນລ (kernel tricks): ກາຣອາຕີຢູ່ປະກາດກົດວິທີການຂອງຫັ້ພພອ້ຣ໌ເວກເຕອີ່ຣ໌ແມ່ນຈືນ ທີ່ສາມາດກົດກຳນົດພັກໜັກໄດ້ໂດຍຕຽນ ແລະ ໄມ່ຕ້ອງກົດກຳນົດພັກໜັກລັກໝະນະສຳຄັຟຸ່ງ.

ພັກໜັກເຄອົງເນລ (kernel function): ພັກໜັກກົດວິທີການຄ່າສເກລ່າຮ້າ ທີ່ປະກາດກົດວິທີການສັນພັນຮ່າງຈຸດຂໍ້ອມູລສອງຈຸດ. ໃນປຣິກູມີລັກໝະນະສຳຄັຟຸ່ງ.

ພັກໜັກເຄອົງເນລເຊີ່ງເສັ້ນ (linear kernel): ພັກໜັກເຄອົງເນລ $k(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T \mathbf{x}'$.

ພັກໜັກເຄອົງເນລເກາສ໌ເຊີ່ຍນ (gaussian kernel): ພັກໜັກເຄອົງເນລ $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$.

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 σ หลาย ๆ ค่า. สังเกตผล อวิปρาย 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 จากการแจกแจงแบบเกาส์เชิง จำนวน 100 จุด ข้อมูล โดยมีค่าเฉลี่ยเป็น 3 และค่าเบี่ยงเบนมาตรฐานเป็น 2. บรรทัดที่สอง สร้างค่า x ที่ต้องการตาม และบรรทัดที่สาม คือการเรียกโปรแกรม `kde` เพื่อประมาณความหนาแน่นความน่าจะเป็นของข้อมูล `datax`. บรรทัดสุดท้ายเป็นการวาดกราฟแสดงความหนาแน่นที่ค่าอินพุตต่าง ๆ. ดูตัวอย่างจากรูปที่ 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

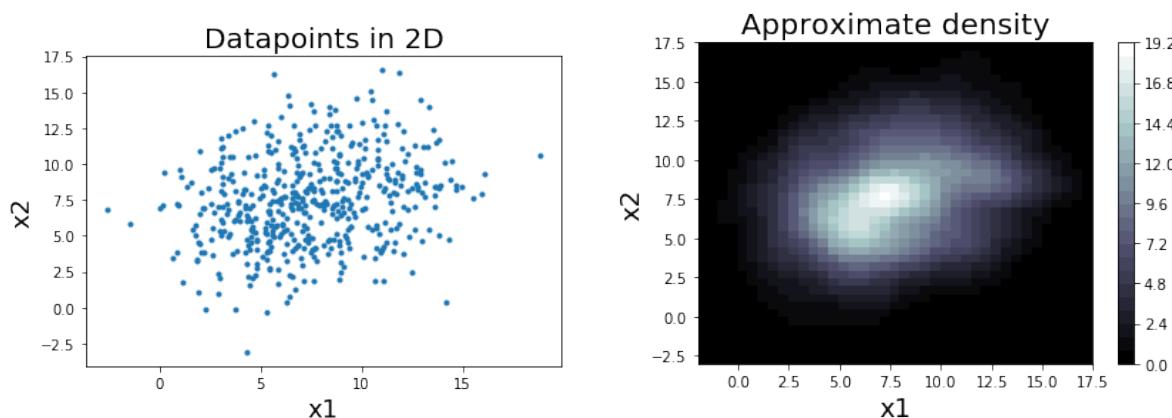
แบบฝึกหัด 4.2

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 และสร้างเป็นข้อมูลสองมิติจำนวน 500 จุดข้อมูล. และใช้วิธีการประมาณความหนาแน่นแก่น (สมการ 4.3 และ 4.4) เพื่อประมาณความหนาแน่นความน่าจะเป็น จากข้อมูลนั้น โดยทดลองค่า σ หลาย ๆ ค่า. สังเกตผลอภิปราย และสรุป.

รูปที่ 4.23 แสดงตัวอย่างข้อมูลสองมิติที่สร้างขึ้น และค่าความหนาแน่นที่ประมาณอອกมา.



รูปที่ 4.23 ตัวอย่างการประมาณค่าความหนาแน่นความน่าจะเป็นสำหรับข้อมูลสองมิติ. ภาพซ้าย แสดงจุดข้อมูลตัวอย่าง. ภาพขวา แสดงค่าประมาณความหนาแน่นความน่าจะเป็นของข้อมูล. ระหว่างที่แทนค่าความหนาแน่นความน่าจะเป็น ซึ่งค่าแสดงด้วยແບສີຕ້ານຂ້າງ.

แบบฝึกหัด 4.3

แบบฝึกหัดนี้ เราจะศึกษารูปปั๊มน์ของเซ็พพอร์ตเวกเตอร์แมชชีน. จงสร้างข้อมูลขึ้นมาจำนวน 200 จุดข้อมูล โดยเป็นกลุ่มบวกและกลุ่มลบอย่างละครึ่ง จุดข้อมูลอยู่ในปริภูมิสองมิติ และข้อมูลสามารถแบ่งแยกได้อย่างสมบูรณ์ซึ่งเส้น เช่น ข้อมูลที่แสดงในรูปที่ 4.24. และแก้ปัญหาในรูปปั๊มน์ของเซ็พพอร์ตเวกเตอร์แมชชีน เพื่อหาอภิรະนาบแบ่งแยก นั่นคือ หากค่าพารามิเตอร์ w และ b ดังแสดงในรูปที่ 4.25.

ทบทวนปัญหาปัญหานี้ สำหรับข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น คือ

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{s.t. } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, \dots, N. \end{aligned}$$

หมายเหตุ ปัญหานี้ เราจะใช้ฟังก์ชันเอกลักษณ์เป็นลักษณะสำคัญ นั่นคือ $\mathbf{z} = \phi(\mathbf{x}) = \mathbf{x}$ (ซึ่งเทียบเท่าการใช้เครื่องเรนาลเชิงเส้น $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$).

เราอาจสามารถแก้ปัญหาได้ด้วยวิธีการลงโทษ เช่นอาจกำหนดฟังก์ชันลงโทษ เป็น

$$P(\mathbf{w}, b) = \sum_i \text{relu}(1 - y_i \cdot (\mathbf{w}^T \mathbf{x}_i + b))$$

เมื่อ

$$\text{relu}(a) = \begin{cases} a & \text{เมื่อ } a \geq 0, \\ 0 & \text{หากเป็นกรณีอื่น.} \end{cases}$$

จากปัญหาปัญหานี้ ของชั้พพอร์ตเวกเตอร์แมชชีน 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)
```

ใช้คันหาอภิรานา (ระบุด้วย \mathbf{w} และ b ซึ่งคือ $wbo[:2]$ และ $wbo[2]$ ตามลำดับ). โปรแกรม **loss** และ **dloss** รวมถึงโปรแกรมอื่นที่เกี่ยวข้องกำหนดดังแสดงในรายการ 4.2. โปรแกรม **gd** (แสดงในรายการ 4.3) คำนวณวิธีลิงเกรเดียนต์.

รายการ 4.2: ตัวอย่างโปรแกรมการค้นหาอภิรานา จากปัญหาปัญหานี้ ของชั้พพอร์ตเวกเตอร์แมชชีน ด้วยวิธีลิงเกรเดียนต์.
ตัวแปร **datax** และ **datay** แทนข้อมูลอินพุตและฉลากกลุ่ม ตามลำดับ โดยทั้งคู่เป็น **np.array** สัดส่วน **(2,N)** และ **(1,N)** เมื่อ **N** เป็นจำนวนจุดข้อมูล. หมายเหตุ ตัวแปร **datax** และ **datay** เป็นตัวแปรล่วงกาล (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

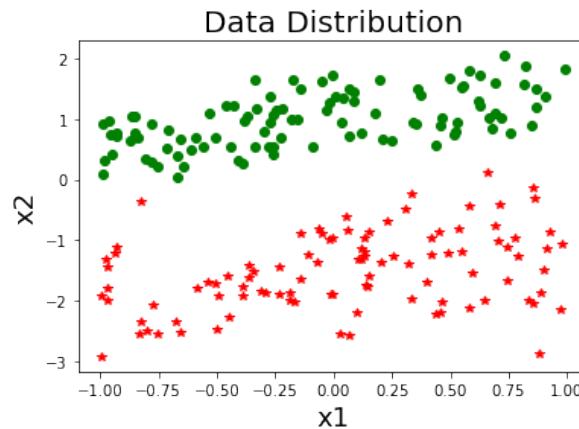
```

รายการ 4.3: โปรแกรมวิธีลิงเกรเดียนต์

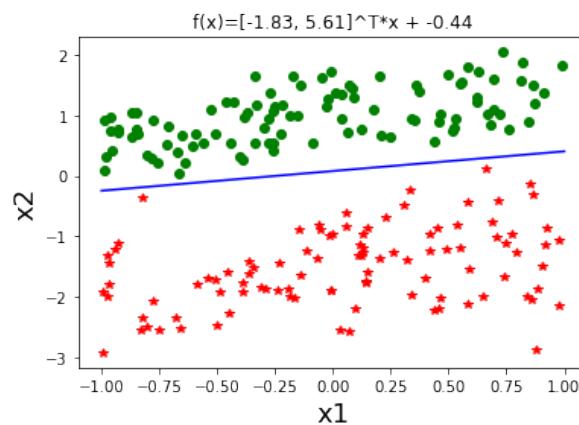
```

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 ตัวอย่างอภิรະนาบที่หาได้จากรูปปัจมุณ สำหรับข้อมูลที่สามารถแบ่งแยกได้โดยสมบูรณ์เชิงเส้น. เส้นทึบสีน้ำเงินแสดงอภิรະนาบที่หาได้.

มอดูลไซไฟ. หัวข้อนี้ แนะนำมอดูลไซไฟ (*Scipy*) ซึ่งมีเครื่องมือหลายอย่างสำหรับงานคำนวณทางวิทยาศาสตร์ รวมถึงมอดูล **optimize**. มอดูล **optimize** มีเครื่องมือการหาค่าตี่สุดอยู่หลายวิธี. แม้วิธีลงเกรเดียนต์ (หัวข้อ 2.3) เป็นวิธีการที่ใช้งานได้ แต่ในทางปฏิบัติ วิธีที่มีประสิทธิภาพมากกว่าวิธีลงเกรเดียนต์อยู่มากมาย และด้วยแบบจำลอง **optimize** เราสามารถไปใช้วิธีเหล่านั้นได้ 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 แล้วแก้ปัญหาในรูปปัจจุบันของชั้พพอร์ตเวกเตอร์แมชชีน เพื่อหาอภิรະนาบที่ดีที่สุด

ศึกษาการเขียนโปรแกรมด้วยการใช้มือดูล **optimize** (ดังแสดงในตัวอย่างของแบบฝึกหัดนี้) เปรียบเทียบกับด้วยวิธีลงเกรเดียนต์ (แบบฝึกหัด 4.3) ทดลองใช้งาน สังเกตผล อภิปราย และสรุป.

ตัวอย่างคำสั่งข้างล่าง ฝึกแบบจำลอง และรายงานผลอภิรະนาบที่พบ

```
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 เป็นการสร้างตัวแปรตัตุ **svm** จากคลาส **primal_SVM** ซึ่งแสดงในรายการ 4.4. บรรทัดที่สอง กำหนดฟังก์ชันลักษณะสำคัญเป็นฟังก์ชันเอกลักษณ์. บรรทัดที่สาม เป็นการฝึกชั้พพอร์ตเวกเตอร์แมชชีน ด้วยข้อมูล **datax** และ **datay**. ผลลัพธ์ที่ได้จากการฝึกจะสรุปออกมาเป็นค่าของพารามิเตอร์ **svm.wopt** และ **svm.bopt** ที่ใช้บรรยายอภิรະนาบ. สังเกตว่า เมื่อตัตุ **train** มีการเรียกใช้

```
res = optimize.minimize(minf, wb0, method='SLSQP', jac=gradf,
                        constraints=ineq_cons, options=options)
```

ซึ่งเป็นเครื่องมือการหาค่าตี่ที่สุด โดยระบุวิธีเป็น '**SLSQP**' (ศึกษารายละเอียดเพิ่มเติมจากเวปไซต์ของไซไฟ หากสนใจ) โดยภายใน เมื่อตัตุ **train** ได้กำหนดค่า **minf** และ **gradf** ซึ่งคือ ฟังก์ชันจุดประสงค์ และฟังก์ชันเกรเดียนต์ ตามลำดับ พร้อมทั้งข้อจำกัดแบบอสมการ **ineq_cons**.

ในการใช้งานจริง สิ่งที่ต้องการ คือการทำนายกลุ่มของข้อมูล. เมื่อตัตุ **decision_score** ของ **primal_SVM** ใช้คำนวนค่าฟังก์ชันแบ่งแยก ซึ่งจะนำไปใช้ตัดสินกลุ่ม. ส่วนค่าของ **w** และ **b** นั้น เป็นรายละเอียดภายใน ไม่จำเป็นต้องรายงาน และการใช้งานจริงของชั้พพอร์ตเวกเตอร์แมชชีน ก็จะไม่มีการคำนวนค่า **w** ออกมาก เพราะว่า รูปแบบการคำนวนถูกแปลงไป เพื่อใช้ประโยชน์จากฟังก์ชันโคอร์เนล. ดูแบบฝึกหัด 4.5 สำหรับโปรแกรมชั้พพอร์ตเวกเตอร์แมชชีนที่ใช้งานจริง.

รายการ 4.4: โปรแกรมชั้พพอร์ตเวกเตอร์แมชชีน จากปัญหาปัจจุบัน สำหรับกรณีแบ่งแยกโดยสมบูรณ์

```
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.

$$\begin{aligned} \sum_{i=1}^N \alpha_i y_i &= 0, \\ 0 \leq \alpha_i &\leq C \quad \text{สำหรับ } i = 1, \dots, N. \end{aligned}$$

เนื่องจาก ปัญหาคู่' สำหรับกรณีแบ่งแยกได้โดยสมบูรณ์ กับกรณีที่ว่าไปต่างกัน เพียงข้อกำหนด $\alpha_i \leq C$ ซึ่งหากเลือกค่า C ขนาดใหญ่ก็จะให้ผลแบบเดียวกับกรณีแบ่งแยกได้โดยสมบูรณ์ ตัวอย่างข้างล่างนี้ จึงใช้รูปแบบที่มีอภิมานพารามิเตอร์ C .

โปรแกรมในรายการ 4.5 แสดงตัวอย่างโปรแกรมชั้พพอร์ตเวกเตอร์แมชชีน. สังเกต วิธีการเขียนโปรแกรมจะใช้การทำเวคตอร์ เช่น หาการคำนวณค่าพารามิเตอร์ 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 ซึ่งจะซับซ้อนกว่าตัวอย่างการทำเวคตอร์ เช่นที่อภิปรายข้างต้นเล็กน้อย.

คล้ายกับตัวอย่างในแบบฝึกหัด 4.4 คำสั่งข้างล่าง แสดงตัวอย่างการฝึก และการอนุมานด้วยชั้พพอร์ตเวกเตอร์แมชชีน

```
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 และจำนวนจุดข้อมูล ตามลำดับ. คำสั่งสุดท้าย ใช้คำนวณค่าฟังก์ชันแบ่งแยกสำหรับ ข้อมูลทดสอบ **testx**.

งศึกษาวิธีการเขียนโปรแกรม ทดลองใช้งาน สังเกตผล อภิปราย และสรุป. ทดลองค่า C ต่าง ๆ และรายงานตัวอย่างในรูปที่ 4.20.

รายการ 4.5: ชั้พพอร์ตเวกเตอร์แมชชีน

```
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:  $\theta \leq \alpha_i \leq C$  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(
57         alpha > self.epsilon,
58         alpha < C - self.epsilon) )[0]
59     Ns = len(sv_ids)
60     Np = len(svp_ids)
61
62     if Np == 0: # No support vector on the edge
63         print('# No support vector on the edge.')
64         svp_ids = sv_ids
65         Np = Ns
66
67     # support vectors
68     self.sv = x[:, sv_ids] # D x Ns
69     self.sy = y[0, sv_ids].reshape((1,-1)) # 1 x Ns
70     spy = y[0, svp_ids].reshape((1,-1)) # 1 x Np
71
72     # support alphas
73     self.salpha = alpha[sv_ids]
74
75     # optimal b
76     sK = K[np.repeat(sv_ids, Np),
77             np.tile(svp_ids, Ns)].reshape((Ns, Np))
78     self.bopt = np.mean(spy - \
79                         np.dot(self.salpha.T * self.sy, sK))
80     return res
81
82     def decision_score(self, x):
83         assert x.shape[0] == self.sv.shape[0] # x in D x N
84         # Compute support kernels
85         sK = self.kernel(x, self.sv) # N x Ns
86         assert (sK.shape[0] == x.shape[1])
87                     and (sK.shape[1] == self.sv.shape[1])
88
89         yhat = np.dot(sK,
90                     self.salpha * self.sy.reshape((-1,1))) + self.bopt
91         return yhat.T      # 1 x N

```

แบบฝึกหัด 4.6

จงสร้างข้อมูลที่ไม่สามารถแบ่งแยกสมบูรณ์ได้เชิงเส้น โดยจะสร้างให้มีลักษณะเดียวกัน แต่ควรทำให้

สามารถตรวจได้สะดวก เช่น อินพุตควรจะเป็นสองมิติ. ตัวอย่างอาจจะเห็นที่แสดงในรูปที่ 4.26. ทดลองใช้ชั้พพอร์ตเวกเตอร์แมชชีน กับเครื่องเนลเชิงเส้น และเครื่องเนลเก้าส์เชียนที่ค่า σ ต่าง ๆ. ดูรูปที่ 4.22 กับ 4.21 สำหรับตัวอย่าง. ออกแบบการทดลอง เพื่อศึกษาผลของ C และ σ . ทดลอง สังเกตผล ภัยประย และสรุป.

ตัวอย่าง คำสั่งข้างล่างแสดงการกำหนดค่าเครื่องเนลของตัวแปรตั้ง (จาก SVM) ให้เป็นเครื่องเนลเก้าส์เชียน (ใช้ $\sigma = 2$)

```
svm.kernel = lambda xq, xp: gaussian(xq, xp, sigma=2)
```

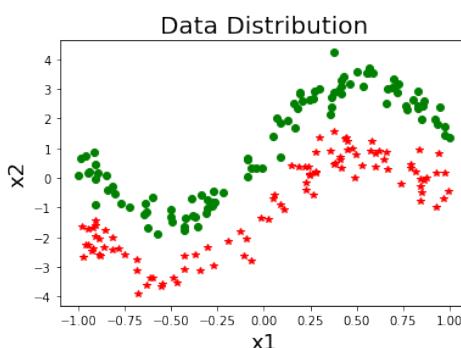
โดยโปรแกรม gaussian แสดงในรายการ 4.6.

รายการ 4.6: โปรแกรมเครื่องเนลเก้าส์เชียน 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

“สิ่งที่คุณได้จากการบรรลุจุดมุ่งหมายไม่สำคัญเท่ากับ สิ่งที่คุณได้เป็นจากการบรรลุจุดหมาย.”

—โยหันน์ โวล์ฟกัง วอน เกอเอ

โครงข่ายประสาทเทียมความลึกสองชั้นนั้น แม้จะสามารถทำงานหลาย ๆ อย่างได้ดี ดังที่แสดงในตัวอย่างบท 3 และในทางทฤษฎีนั้น โครงข่ายประสาทเทียมความลึกสองชั้นนั้น จะสามารถฝึกให้ประมาณฟังก์ชันอะโรเก็ตได้^[47, 91] แต่ในทางปฏิบัติแล้ว สำหรับงานที่ซับซ้อนมาก ๆ ทั้งจำนวนหน่วยซ่อนที่ต้องเพิ่มจำนวนมหาศาล และการฝึกที่ใช้ทรัพยากรการคำนวณมาก รวมถึงข้อมูลพร้อมฉลากที่ต้องมีจำนวนมากพอ ทำให้ การประยุกต์กับงานการรู้จำรูปแบบของโครงข่ายประสาทเทียมสองชั้น จำกัดอยู่มาก โดยเฉพาะกับงานที่ซับซ้อนมาก ๆ เช่น งานรู้จำภาพ เสียงพูด หรือภาษาธรรมชาติ.

จนกระทั่ง ความก้าวหน้าล่าสุด ก็คือ แนวทางของการเรียนรู้เชิงลึก (deep learning) ที่ได้ขยายความสามารถของการประยุกต์ใช้โครงข่ายประสาทเทียมไปแบบกว้างกระโดด การเรียนรู้เชิงลึก แม้จะมีโครงสร้างพื้นฐานเป็นโครงข่ายประสาทเทียม แต่มิกกลไกสำคัญหลายอย่างที่ช่วยขยายความสามารถ ซึ่งหนึ่งในนั้นคือ การใช้โครงสร้างเชิงลึก หรือการใช้โครงข่ายประสาทเทียมที่มีจำนวนชั้นคำนวณมาก¹.

หากจะเริ่มต้นกล่าวถึงการเรียนรู้แบบลึก แม้แนวคิดจะมีมานานมากพอ ๆ กับจุดกำเนิดของโครงข่ายประสาทเทียมเอง และวิธีการแพร่กระจายย้อนกลับ (หัวข้อ 3.3) ก็มีความทั่วไปมากพอ ที่จะใช้ใน

¹ การนับจำนวนชั้นของโครงข่าย 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} คือเอาต์พุตที่ทำนาย และ \mathbf{x} คืออินพุตที่ถูก. โครงข่ายลักษณะนี้ อาจนับเป็นสองชั้น ตามจำนวนชุดของค่าน้ำหนัก $\mathbf{W}^{(2)}$ และ $\mathbf{W}^{(1)}$ ซึ่ง ณ ที่นี่ ใช้แบบแผนนี้ในการนับ. แต่ผู้อ่านอาจพบ บางแห่งนับเป็นสามชั้น โดยนับชุดค่าของหน่วยย่อย ได้แก่ \mathbf{y} , $\mathbf{z}^{(1)}$, และ \mathbf{x} โดยมองเสื่อม眼光ว่า อินพุตที่เป็นชุดค่าของหน่วยย่อย. บางแห่ง อาจเลือกอ้างถึงโครงข่ายนี้ ว่า เป็นโครงข่ายจำนวนหนึ่งชั้น ได้แก่ ชั้นช่อง $\mathbf{z}^{(1)}$ เนื่องจากหากมองอินพุตกับเอาต์พุตเป็นชั้นของโครงข่าย อินพุตกับเอาต์พุต ก็จะเป็นชั้นที่มีกันทุก ๆ โครงข่าย ฉะนั้น รายงานเฉพาะส่วนที่ต่าง ซึ่งก็คือจำนวนชั้นช่องก็พอ. วิธีการนับจำนวนชั้นของโครงข่าย จะมีความสำคัญน้อยลง เมื่อพิจารณาโครงข่ายที่มีความซับซ้อนมากขึ้น เช่น โครงข่ายอเล็กซ์เน็ต (หัวข้อ 6.5) ที่มีการแยกเส้นทางการคำนวณ และนำผลการคำนวณจากแต่ละเส้นทางกลับมาร่วมกันภายหลัง.

กระบวนการฝึกของโครงข่ายประสาทเทียมก็ชั้นกีดี. แต่การประยุกต์ใช้โครงข่ายประสาทเทียมแบบลึกในช่วงก่อนศัตรูที่ยังสืบอี้ดันน้ำจำกัดอยู่มาก. อุปสรรคที่ที่สำคัญสำหรับการประยุกต์ใช้โครงข่ายประสาทเทียมแบบลึก ก็คือ ปัญหาการเลือนหายของเกรเดียนต์.

จนกระทั่งงานศึกษาที่สำคัญของฮินตันและชาลาคุทธินอฟ[87] ที่พัฒนาระบบฝึกโครงข่ายประสาทเทียมแบบลึกได้อย่างมีประสิทธิภาพ. หลังจากนั้น ก็มีการศึกษาการเรียนรู้เชิงลึกอย่างกว้างขวาง และการเรียนรู้เชิงลึกกีกลายเป็นศาสตร์และศิลป์ที่สำคัญสำหรับศาสตร์หลาย ๆ แขนง [114, 97, 178, 115, 48, 52, 64, 26, 157, 35, 123, 220] เช่น คอมพิวเตอร์วิทัค์ (computer vision), การรู้จำคำพูด (speech recognition), การประมวลผลภาษาธรรมชาติ (natural language processing), การค้นหายา (drug discovery), และ จีโนมิกส์ (genomics). ความสนใจในการเรียนรู้เชิงลึกและโครงข่ายประสาทเทียมมีสูงมาก จนทำให้เกิดการศึกษาและพัฒนาอุปกรณ์คำนวณ สำหรับโครงข่ายประสาทเทียมโดยเฉพาะ ได้แก่ หน่วยประมวลผลเชิงประสาทแปลง (Neuromorphic Processing Unit[127] คำย่อ NPU).

ปัจจัยของความสำเร็จของการเรียนรู้เชิงลึก หรือโครงข่ายประสาทเทียมแบบลึก มีอยู่หลายประการ ตั้งแต่ ฮาร์ดแวร์ที่เร็วขึ้น, ข้อมูลที่มากและหลากหลายขึ้น, วิธีการเตรียมข้อมูลที่ดีขึ้น รวมถึง การทำการฝึกก่อน[65], ขั้นตอนวิธีการหาค่าดีที่สุดที่มีประสิทธิภาพมากขึ้น[93, 110], การฝึกที่ละเอียด[13], การใช้แบบจำลองที่จับลักษณะสำคัญของข้อมูลได้ดีขึ้น เช่น โครงข่ายคอนโวโลชัน[114] และโครงข่ายประสาทเวียนกลับ[88, 89, 180], การใช้กลไกการตกอອก[189], การใช้กลไกความลaise[54, 203], การใช้กลไกโครงข่ายปรับษ์เชิงสร้าง[77, 145] ไปจนถึงการเปลี่ยนฟังก์ชันกระตุ้นจากซิกมอยด์ไป เป็นฟังก์ชันกระตุ้นที่ลดช่วงค่าอิมตัว เช่น เรลู[87].

บทที่ 5 นี้ อภิปรายการแก้ปัญหาการเลือนหายของเกรเดียนต์ ด้วยฟังก์ชันกระตุ้นเรลู (หัวข้อ 5.1), เทคนิคการจัดการฝึกกับข้อมูลขนาดใหญ่ด้วยการฝึกที่ละเอียด[13] (หัวข้อ 5.2), เทคนิคการตกอອก (หัวข้อ 5.3), วิธีการกำหนดค่าน้ำหนักเริ่มต้น (หัวข้อ 5.4), และขั้นตอนวิธีการฝึกที่มีประสิทธิภาพมากขึ้น (หัวข้อ 5.5). นอกจากนั้น ปัจจัยหนึ่งที่มีส่วนอย่างมาก ในพัฒนาการ และความสนใจ ไปจนถึงการประยุกต์ใช้ที่กว้างขวาง ก็คือเครื่องมือที่ช่วยให้การใช้งานเทคนิคต่าง ๆ เหล่านี้ทำได้สะดวกมากขึ้น หัวข้อ 5.7 อภิปรายตัวอย่างเครื่องมือที่ได้รับความนิยมอย่างสูง สำหรับการประยุกต์ใช้การเรียนรู้เชิงลึก.

การใช้โครงข่ายคอนโวโลชัน ที่หมายกับข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง เช่น ข้อมูลภาพ รวมไปจนถึงเทคนิคการฝึกก่อน และตัวอย่างโครงสร้างของโครงข่ายคอนโวโลชันที่รู้จักกันอย่างกว้างขวาง อภิปรายในบทที่ 6. การนำโครงข่ายคอนโวโลชัน ไปประยุกต์ใช้กับงานการรู้จำทัศนรูปแบบ เป็นเนื้อหาหลักที่อภิปรายในบทที่ 7.

บทที่ 8 อภิปรายโครงข่ายประสาทเวียนกลับ ที่หมายกับข้อมูลที่มีลักษณะเชิงลำดับ. ตัวอย่าง การประยุกต์ใช้โครงข่ายประสาทเวียนกลับ กับงานการรู้จำรูปแบบเชิงลำดับ เช่น การประมวลผลภาษาธรรมชาติ รวมไปถึงกลไกความลaise เป็นเนื้อหาหลักที่อภิปรายในบทที่ 9.

5.1 ปัญหาการเลือนหมายของเกรเดียนต์

ปัญหาสำคัญที่ทำให้โครงข่ายลึกไม่ได้รับความนิยมในยุคต้นของพัฒนาการ คือ การฝึกโครงข่ายประสาทเทียมที่ลึกมากนั้นทำได้ยากมาก.

รูปที่ 5.1 แสดงความก้าวหน้าของการฝึก เมื่อใช้ความลึกต่าง ๆ. แนวโน้มรวม ก็คือ ยิ่งความลึกมาก ดูเหมือนจะต้องการจำนวนสมัยฝึกที่มากขึ้น ค่าฟังก์ชันสูญเสียต่อสมัยของความลึกที่มากขึ้น ลุล่องที่จำนวนสมัยมากขึ้น. . สังเกต ความก้าวหน้าของการฝึกเมื่อใช้ความลึกสิบชั้น (สันทิบหนาสีดำ) ซึ่งมีลักษณะลุล่องราบรื่นเรียบร้อย แสดงถึงการฝึกที่สมบูรณ์ แต่ผลการฝึกได้คุณภาพการทำนายแย่มาก. นั่นคือ การฝึกเสร็จสิ้น แต่ฝึกไม่สำเร็จ ซึ่งยืนยันอย่างชัดเจนจากรูปที่ 5.2 ที่สรุปค่าฟังก์ชันสูญเสียของการฝึก เมื่อใช้ความลึกต่าง ๆ.

ปัญหาของการฝึกโครงข่ายลึกแบบนี้ ภายหลังพบว่า สาเหตุคือ ขนาดเกรเดียนต์ในชั้นต้น ๆ ของโครงข่ายที่เล็กมากเกินกว่าจะที่ปรับค่าน้ำหนักและไบอัสได้อย่างมีประสิทธิภาพ. รูปที่ 5.3 แสดงตัวอย่างขนาดเฉลี่ยของเกรเดียนต์ที่ชั้นต่าง ๆ ของโครงข่ายประสาทเทียมสิบชั้น. รูปที่ 5.4 สรุปค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ในชั้นต่าง ๆ. สังเกตว่า ขนาดเฉลี่ยเกรเดียนต์แต่ละชั้นแตกต่างกันอย่างมาก. (แบบฝึกหัด 5.1).

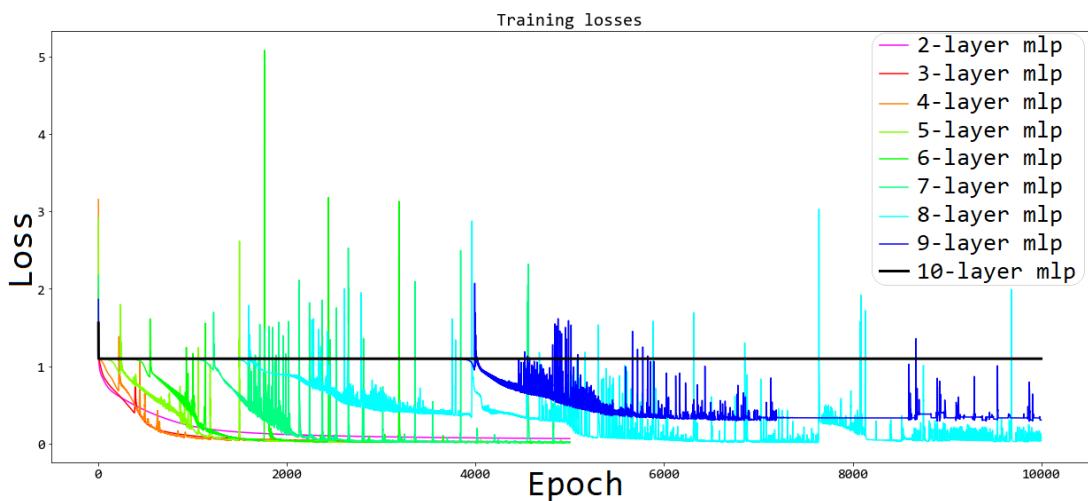
การฝึกโครงข่ายประสาทเทียม ก็คือการหาค่าน้ำหนักที่เหมาะสม ซึ่งแนวทาง คือใช้เกรเดียนต์ของฟังก์ชันจุดประสงค์ต่อค่าน้ำหนัก. แต่หากเกรเดียนต์มีขนาดเล็กมาก การหาค่าน้ำหนักที่เหมาะสมก็ทำได้ยาก และในหลายสถานการณ์ก็คือความล้มเหลวของการฝึกโครงข่าย. ปัญหาการฝึกโครงข่ายลึก ดังที่อภิปรายนี้ รู้จักกันในชื่อปัญหาการเลือนหมายของเกรเดียนต์ (*vanishing gradient problem*). นั่นคือ ค่าเกรเดียนต์ที่คำนวณจากวิธีแพร่กระจายอยู่กลับสำหรับค่าน้ำหนักในชั้นต้น ๆ (ใกล้อินพุต) จะมีขนาดเล็กลงมาก จนแทบไม่สามารถปรับค่าน้ำหนักได้ ซึ่งส่งผลให้ การฝึกโครงข่ายลึกล้มเหลว.

การเลือนหมายของเกรเดียนต์เอง ก็พบว่าสาเหตุหลักมาจากการใช้ฟังก์ชันกระตุนซิกมอยด์ ที่มีช่วงพลวัตรแคบ. เพื่อแก้ปัญหาช่วงพลวัตรของฟังก์ชันกระตุนซิกมอยด์ ฟังก์ชันกระตุนเรคติไฟร์ลีเนียร์ (*rectified linear* ซึ่งมักย่อว่า เรลู *relu* สำหรับ *rectified linear unit*[87]) ถูกเสนอขึ้นมา. การเปลี่ยนไปใช้ฟังก์ชันกระตุน ที่เป็นฟังก์ชันเรคติไฟร์ลีเนียร์ หรือเรลู เป็นปัจจัยที่สำคัญ ที่ช่วยให้การฝึกโครงข่ายประสาทเทียมเชิงลึกทำได้ง่ายขึ้น และช่วยลดปัญหาการเลือนหมายของเกรเดียนต์. สมการ 5.1 แสดงการคำนวณของฟังก์ชันกระตุนเรลู

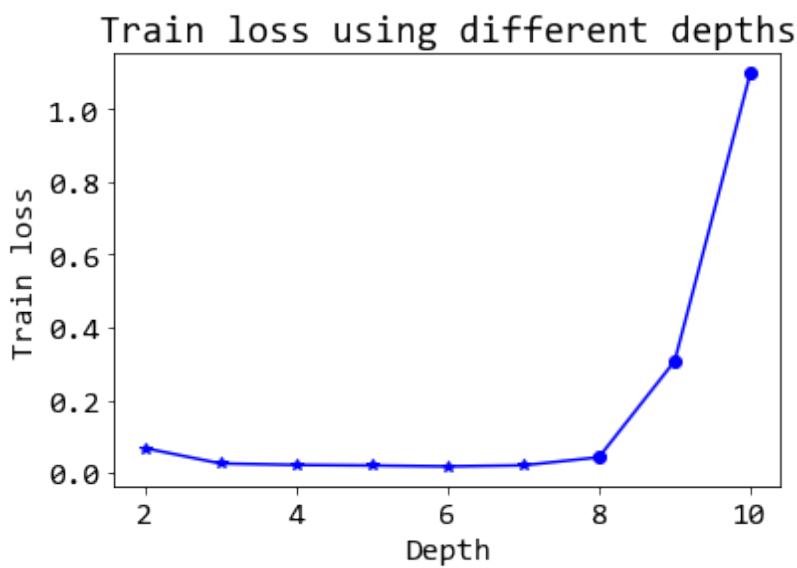
$$\text{relu}(a) = \begin{cases} a & \text{เมื่อ } a \geq 0, \\ 0 & \text{เมื่อ } a < 0. \end{cases} \quad (5.1)$$

ซึ่งอนุพันธ์สามารถคำนวณได้จาก

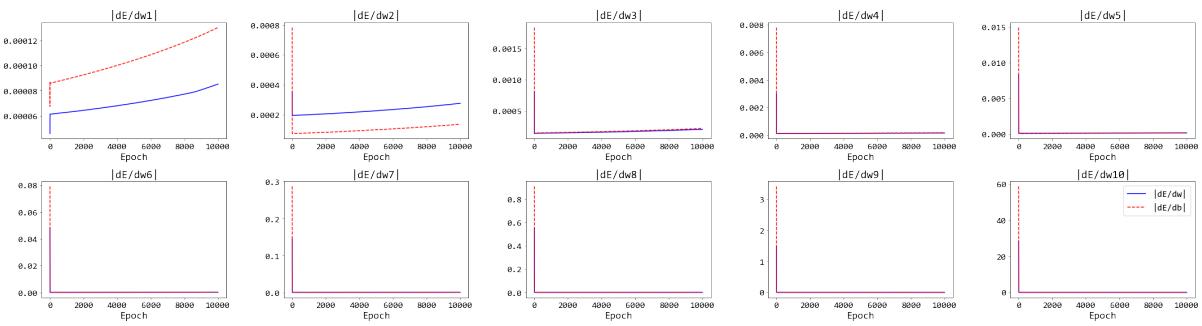
$$\frac{d\text{relu}}{da} = \begin{cases} 1 & \text{เมื่อ } a \geq 0, \\ 0 & \text{เมื่อ } a < 0. \end{cases} \quad (5.2)$$



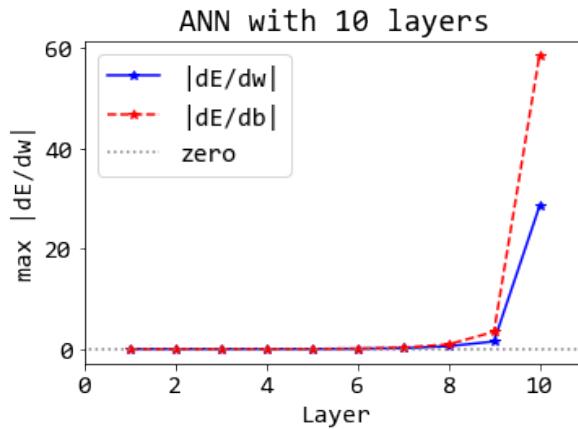
รูปที่ 5.1 ค่าฟังก์ชันสูญเสียต่อสมัยฝึก เมื่อใช้โครงข่ายประสาทเทียมที่ความลึกต่าง ๆ โดย ความลึกสองขั้นถึงเจ็ดขั้น โครงข่ายสามารถถูกฝึกได้ภายใน 5000 สมัย แต่ความลึกแปดขั้นและเก้าขั้น ต้องทำการฝึกถึง 10000 สมัย และที่ความลึกสิบขั้น (เส้นทึบหนาสีดำ) แม้ทำการฝึกไป 10000 สมัยแล้ว ซึ่งความก้าวหน้าของการฝึกถูกคลายการฝึกสมมูลรุนแรง แต่ได้ผลการฝึกที่แย่มาก.



รูปที่ 5.2 การฝึกโครงข่ายลึกล้มเหลว. ภาพแสดงการสรุปค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึกที่ความลึกต่าง ๆ. สรุปค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึก หลังจากฝึกเสร็จ (5000 สมัยสำหรับโครงข่าย 2 ถึง 7 ขั้น และ 10000 สมัยสำหรับโครงข่าย 8 ถึง 10 ขั้น). ในขณะที่โครงข่ายที่ดีกว่าสามารถฝึกได้ แต่การฝึกโครงข่ายลึกกลับล้มเหลว.



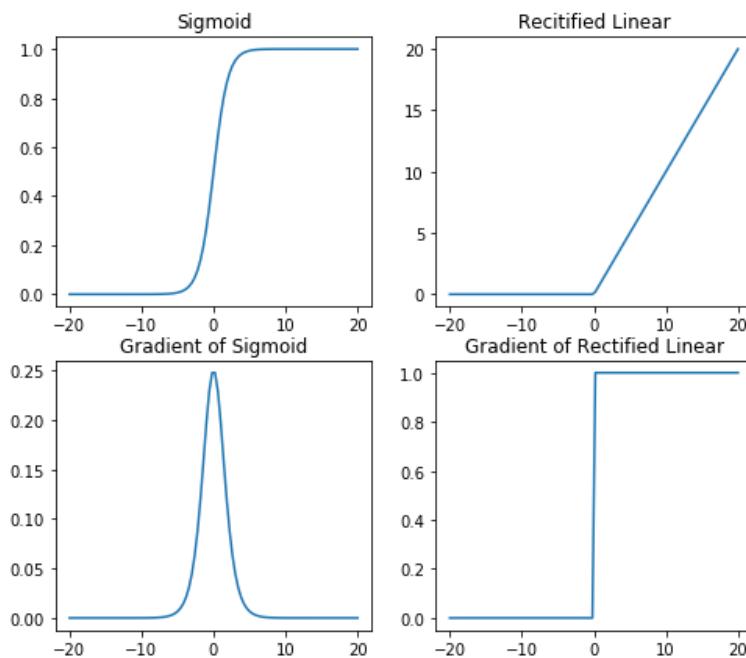
รูปที่ 5.3 ปัญหาการเลือนหมายของเกรเดียนต์. ตัวอย่างความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบชั้น. แต่ละภาพแสดงค่าเฉลี่ยเกรเดียนต์ของค่าน้ำหนัก (เส้นทึบสีฟ้า) และไปอัลส์ (เส้นประสีแดง) ของแต่ละชั้นคำนวณ (ระบุเหนือภาพ) โดยแกนนอนเป็นสมัยฝึก.



รูปที่ 5.4 การเลือนหมายของเกรเดียนต์ในโครงข่ายสิบชั้น. ภาพแสดงขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้นเปรียบเทียบกัน โดยแกนตั้ง แสดงค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้น และแกนนอนระบุชั้นคำนวณ. เกรเดียนต์ของค่าน้ำหนัก แสดงด้วยเส้นทึบสีฟ้า. เกรเดียนต์ของค่าไปอัลส์ แสดงด้วยเส้นประสีแดง. เส้นเขียวลาสีเทา แสดงแนวค่าศูนย์. ขนาดเกรเดียนต์ตั้งแต่ชั้นที่หกย้อนไปถึงชั้นที่หนึ่งมีค่าน้อยมาก ๆ (ใกล้ศูนย์) ซึ่งเป็นสาเหตุทำให้การฝึกล้มเหลว.

รูปที่ 5.5 ภาพบนซ้ายแสดงการกระจายของพังก์ชันซิกมอยด์ เมื่อเปรียบเทียบกับการกระจายต้นของพังก์ชันเรลู (ภาพบนขวา) และค่าอนุพันธ์ของพังก์ชันซิกมอยด์ (ภาพล่างซ้าย) และค่าอนุพันธ์ของพังก์ชันเรลู (ภาพล่างขวา). สังเกตว่า ค่าอนุพันธ์ของพังก์ชันซิกมอยด์ จะมีช่วงพลวัตรอยู่ในบริเวณแคบ ๆ ใกล้ ๆ ศูนย์ ในขณะที่ ค่าอนุพันธ์ของพังก์ชันเรลู จะมีช่วงพลวัตรครอบคลุมบริเวณที่มีค่าเป็นบวกทั้งหมด. การที่ค่าอนุพันธ์ของพังก์ชันซิกมอยด์มีช่วงพลวัตรแคบ ทำให้การฝึกโครงข่ายประสาทเทียมที่ใช้พังก์ชันรายต้นเป็นซิกมอยด์ทำได้ยาก. ในที่นี้ ช่วงพลวัตร หมายถึง ช่วงบริเวณของอินพุตที่ค่าอนุพันธ์มีขนาดใหญ่. จากรูปที่ 5.5 เราจะเห็นว่าค่าอนุพันธ์ของพังก์ชันเรลู มีขนาดเป็นหนึ่ง ตลอดช่วงอินพุตที่มีค่าเป็นบวก เปรียบเทียบกับค่าอนุพันธ์ของพังก์ชันซิกมอยด์ ที่มีค่ามากกว่าศูนย์อย่างชัดเจน อยู่แค่บริเวณที่อินพุตมีค่าใกล้ ๆ ศูนย์เท่านั้น.

รูปที่ 5.6 และ 5.7 แสดงให้เห็นว่า การเปลี่ยนพังก์ชันรายต้นจากซิกมอยด์มาเป็นเรลู ช่วยแก้



รูปที่ 5.5 พังค์ชันกระตุนต่าง ๆ ได้แก่ พังค์ชันกระตุนซิกมอยด์ (ภาพบนขวา) พังค์ชันกระตุนrelu (ภาพบนซ้าย) เกรเดียนต์ของพังค์ชันซิกมอยด์ (ภาพล่างขวา) เกรเดียนต์ของพังค์ชันrelu (ภาพล่างซ้าย).

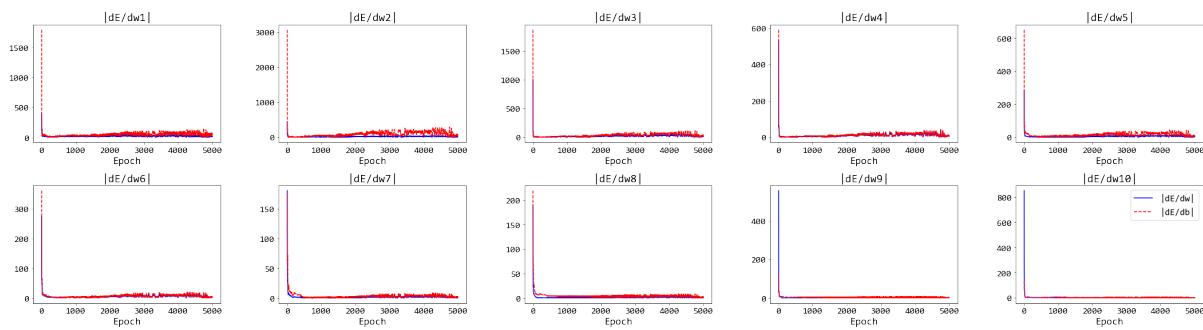
ปัญหาการเลือนหายของเกรเดียนต์ได้อย่างชัดเจน. เปรียบเทียบรูปที่ 5.4 (ใช้พังค์ชันกระตุนซิกมอยด์) กับรูปที่ 5.7 (ใช้พังค์ชันกระตุนrelu) ซึ่งทั้งคู่ เป็นโครงข่ายประสาทเทียมสิบชั้นเหมือนกัน เพียงแต่ใช้พังค์ชันกระตุนต่างกัน. จะเห็นว่า เมื่อใช้พังค์ชันกระตุนซิกมอยด์ ขนาดของเกรเดียนต์จะลดลงเรื่อย ๆ จากชั้นสุดท้ายไปสู่ชั้นต้น. นอกจากขนาดเกรเดียนต์ลดลงเรื่อย ๆ แล้ว ขนาดเกรเดียนต์ยังลดลงไปอยู่ในระดับใกล้ศูนย์ด้วย (เส้นประสีเทา แสดงแนวค่าศูนย์). ขณะที่ เมื่อใช้พังค์ชันกระตุนrelu (รูปที่ 5.7) นอกจาก จะไม่เห็นแนวโน้มการลดลงของเกรเดียนต์จากชั้นสุดท้ายไปชั้นต้นแล้ว (1) ขนาดเกรเดียนต์ มีค่าใหญ่ขึ้นมาก และ (2) ขนาดเกรเดียนต์มีค่าอยู่ในระดับมากกว่าศูนย์อย่างเห็นได้ชัด (อยู่เหนือเส้นประสีเทา).

รูปที่ 5.8 แสดงให้เห็นว่า เมื่อแก้ปัญหาการเลือนหายของเกรเดียนต์ได้ การฝึกโครงข่ายลึกก็สามารถทำได้ดีขึ้นมาก.

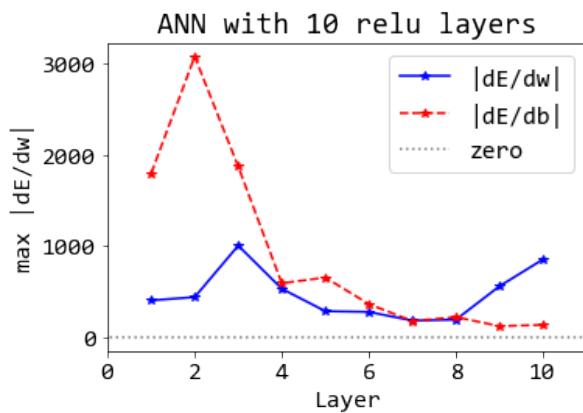
5.2 การฝึกทีละหมู่เล็ก

การมีข้อมูลจำนวนมาก แม้เป็นโอกาสที่ดี แต่ก็ต้องการกลไกที่ช่วยในการจัดการข้อมูลมหาศาล เพื่อการใช้งานหน่วยประมวลผลและหน่วยความจำได้อย่างมีประสิทธิภาพ. หนึ่งในกลไกที่สำคัญนั้น คือ การฝึกทีละหมู่เล็ก (*minibatch training*).

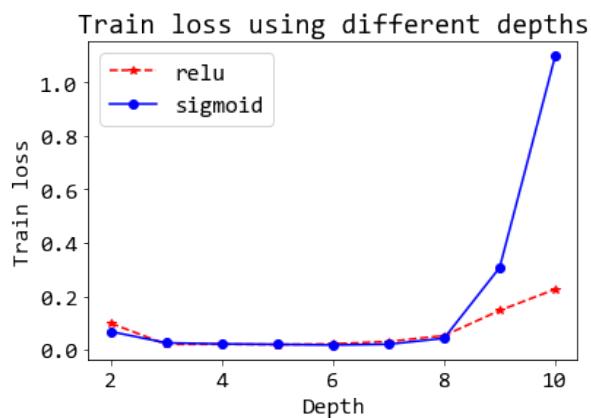
บท 3 ได้อธิบายถึง ทางเลือกในการฝึก โดย การฝึกที่ใช้ข้อมูลทั้งหมดในการปรับปรุงค่าน้ำหนัก พร้อม ๆ กันที่เดียว ซึ่งเรียกว่า การฝึกแบบอฟไลน์ (*offline training*) หรือ การฝึกแบบหมู่ และ



รูปที่ 5.6 ตัวอย่างความก้าวหน้าของการฝึกโครงข่ายประสาทเทียมสิบชั้น เมื่อใช้ฟังก์ชันกราฟตุ้นrelu. แต่ละภาพแสดงค่าเฉลี่ยเกรเดียนต์ของค่าน้ำหนัก (เส้นทึบสีฟ้า) และไบอัส (เส้นประสีแดง) ของแต่ละชั้นคำนวณ (ระบุเหนือภาพ) โดยแกนนอนเป็นสมัยฝึก.



รูปที่ 5.7 การเลือนหมายของเกรเดียนต์บรรเทาลงด้วยการเปลี่ยนฟังก์ชันกราฟตุ้น. ขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้นเบรียบเทียบกัน โดย แกนตั้ง แสดงค่าใหญ่ที่สุดของขนาดเฉลี่ยเกรเดียนต์ของแต่ละชั้น และแกนนอนระบุชั้นคำนวณ. เกรเดียนต์ของค่าน้ำหนัก แสดงด้วยเส้นทึบสีฟ้า. เกรเดียนต์ของค่าไบอัส แสดงด้วยเส้นประสีแดง. การใช้ฟังก์ชันกราฟตุ้นrelu ช่วยให้ค่าเกรเดียนต์ของทั้งสิบชั้นมีขนาดค่าใหญ่เพื่อสำหรับการปรับค่าน้ำหนักและไบอัสอย่างมีประสิทธิภาพ.



รูปที่ 5.8 ค่าฟังก์ชันสูญเสียสำหรับข้อมูลฝึกที่ความลึกต่าง ๆ เปรียบเทียบเมื่อใช้ฟังก์ชันกราฟตุ้นซิกมอยด์กับเมื่อใช้ฟังก์ชันกราฟตุ้นrelu. ฟังก์ชันกราฟตุ้นreluช่วยให้การฝึกโครงข่ายลึกทำได้ดีขึ้นมาก.

การฝึกที่ใช้ข้อมูลที่ลงทะเบียนข้อมูลและปรับปรุงค่าน้ำหนักที่ลงทะเบียนสำหรับแต่ละจุดข้อมูล ซึ่งเรียกว่า การฝึกแบบออนไลน์ (*online training*) หรือ การฝึกแบบส่วนเพิ่ม (*incremental mode*). นอกจากนั้น ยังได้อภิปรายถึงข้อดีและข้อเสียต่าง ๆ ของทางเลือกทั้งสองนี้ นั่นคือ การฝึกแบบอффไลน์ สามารถทำการคำนวณได้อย่างรวดเร็ว และยังสามารถใช้การประมวลผลแบบขนาน (*parallel processing*) มาช่วยทำให้การคำนวณมีประสิทธิภาพมากขึ้นได้ แต่ข้อเสียคือ ต้องการหน่วยความจำมาก. ในขณะที่ ข้อดีของการฝึกแบบออนไลน์ นอกจากต้องการใช้หน่วยความจำประมาณน้อยกว่า คือ เมื่อทำร่วมกับการสลับลำดับของข้อมูลแต่ละสมัยฝึก จะช่วยลดความเสี่ยงในการเข้าไปติดอยู่ในค่าตัวทำต่ำสุดท้องถิ่นได้ หรือ กล่าวอีกอย่างอาจช่วยปรับปรุงคุณภาพของการฝึกได้. ประเด็นเรื่องการฝึกแบบออนไลน์ ช่วยคุณภาพของการฝึกนี้ ถูกเพโลและคณะ[76] เสริมว่า การฝึกโดยใช้ข้อมูลที่ลงทะเบียนน้อย ๆ อาจจะช่วยให้ผลคล้ายการทำเรกูลารายเซอร์[219] ซึ่งอาจจะ เพราะสัญญาณรบกวนที่ปนเข้ามาในกระบวนการฝึก.

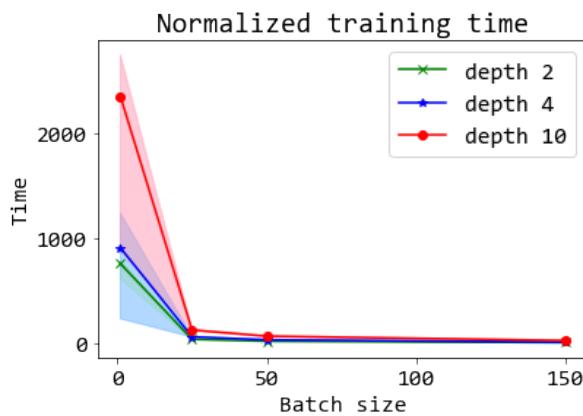
การฝึกที่ลงทะเบียนเล็ก (*minibatch*) เป็นเทคนิคที่ประณีประนอม ระหว่างแนวทางการฝึกแบบหมู่ และการฝึกแบบออนไลน์ เพื่อใช้เวลาในการฝึกไม่นานเกินไป ใช้หน่วยความจำไม่มากเกินไป และได้คุณภาพการฝึกที่ดี. นั่นคือ การฝึกที่ลงทะเบียนเล็ก จะแบ่งข้อมูลออกเป็นกลุ่มเล็ก ๆ โดยที่ แต่ละกลุ่มนี้มีจำนวนข้อมูลมากกว่าหนึ่งจุดข้อมูล แต่น้อยกว่าจำนวนข้อมูลฝึกทั้งหมด. การฝึกแต่ละสมัย จะปรับค่าน้ำหนักและใบอัลฟาร์บสำหรับการคำนวณกับแต่ละหมู่เล็กนี้ จนครบทุกหมู่. สมการ 5.3 แสดงการคำนวณของค่าน้ำหนักสำหรับแต่ละหมู่เล็ก (การคำนวณใบอัลฟาร์บได้ในลักษณะเดียวกัน)

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \alpha \frac{1}{|B_m|} \sum_{n \in B_m} \frac{\partial E_n}{\partial w_{ji}^{(l)}}, \quad (5.3)$$

สำหรับ $m = 1, 2, \dots, \lceil \frac{N}{|B|} \rceil$ เมื่อ $w_{ji}^{(l)}$ เป็นค่าน้ำหนักระหว่างหน่วย j ของชั้น l และหน่วย i ของชั้นก่อนหน้า. 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} และขนาดของหมู่เล็กส่วนใหญ่ ตามลำดับ. ขนาดของหมู่เล็ก (*batch size*) คือ จำนวนจุดข้อมูลในหมู่เล็ก. เชต B_m เป็นเซตของดัชนีของจุดข้อมูลที่ถูกจัดอยู่ในหมู่เล็กที่ m^{th} โดยดัชนีของจุดข้อมูล จะถูกสุ่มจัดเข้าหมู่เล็ก และหากจำนวนข้อมูลทั้งหมดไม่อ่าจแบ่งได้เท่า ๆ กันทุกหมู่เล็ก จะมีหมู่เล็กหนึ่งหมู่ที่มีจำนวนต่างจากหมู่อื่น ๆ (หรือหมู่เศษนี้อาจถูกตัดทิ้ง เพื่อประสิทธิภาพของการคำนวณ).

การทำลักษณะเช่นนี้ จะคล้ายกับการฝึกแบบหมู่ ในเมื่อว่า แต่ละการคำนวณกับหมู่เล็กจะเป็นการคำนวณกับจุดข้อมูลหลาย ๆ จุดพร้อม ๆ กัน และก็จะคล้ายกับการฝึกแบบออนไลน์ ในเมื่อว่า ค่าน้ำหนักระดับหลาย ๆ ครั้งในหนึ่งสมัย (แต่ละครั้ง ปรับสำหรับแต่ละหมู่เล็ก).

ถูกเพโลและคณะ[76] อภิปรายว่า ขนาดของหมู่เล็กที่เหมาะสม ขึ้นกับทั้งฮาร์ดแวร์และขั้นตอนวิธีการหาค่าดีที่สุดที่เลือกใช้ เช่น หากคำนวณด้วยหน่วยประมวลผลกราฟิกส์ (*Graphics Processing*



รูปที่ 5.9 ตัวอย่างเวลาการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ โครงการข่ายขนาดสองชั้น สี่ชั้น และสิบชั้น แสดงด้วยสัญลักษณ์ ตั้งระบุในภาพ. พื้นที่สีเขียวอ่อน ฟ้าอ่อน และชมพู แสดงช่วงระหว่างค่ามากที่สุดและน้อยที่สุดของเวลาฝึกโครงข่ายสองชั้น สี่ชั้น และสิบชั้น ตามลำดับ. หมายเหตุ พื้นที่สีเขียวอ่อน อาจสังเกตได้ยากในภาพ เนื่องจากเวลาที่ทดสอบพบว่ามีความผันผวนต่ำ.

Unit หรือ *GPU*) และเลือกขนาดหมู่เล็กเป็นจำนวนของสองยกกำลัง เช่น 16, 32, 64, 128, 256 อาจช่วยให้ประสิทธิผลได้เร็วขึ้น. ขั้นตอนวิธีการหาค่าดีที่สุดที่ใช้แค่ค่ากรเดียนต์ เช่น วิธีลงกรเดียนต์ โดยทั่วไปแล้ว มักจะทนทานและสามารถได้งานได้ดีกับขนาดต่าง ๆ ของหมู่เล็กได้.

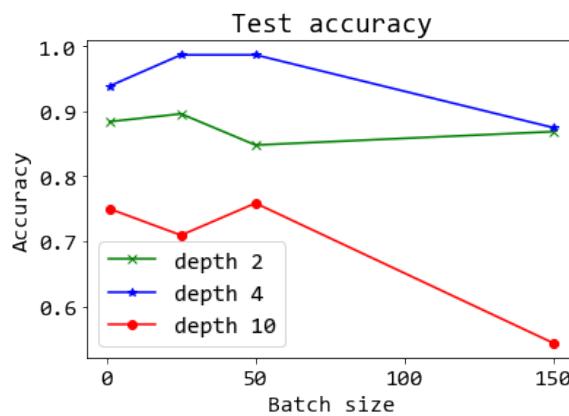
หมายเหตุ แต่ขั้นตอนวิธีการหาค่าดีที่สุดที่ใช้หั้งค่ากรเดียนต์และเซเชี่ยน² ต้องการขนาดหมู่เล็กที่ใหญ่พอที่จะสามารถประมาณค่าเซเชี่ยนได้อย่างมีประสิทธิภาพ เช่น[76] 10000.

รูปที่ 5.9 แสดงเวลาที่ใช้ในการฝึก เมื่อใช้ขนาดหมู่เล็กต่าง ๆ. หมายเหตุ การใช้ขนาดหมู่เล็กเป็นหนึ่ง เทียบเท่าการฝึกแบบออนไลน์ และการใช้ขนาดหมู่เล็กเท่ากับหรือมากกว่าจำนวนข้อมูล (ซึ่งในตัวอย่างแสดงในรูป คือ 150) เทียบเท่าการฝึกแบบหมู่. สังเกตว่า ขนาดหมู่ที่เล็กลง จะใช้เวลาในการฝึกนานขึ้น. แม้ว่า การใช้ขนาดหมู่ที่เล็กลง จะทำให้เวลาในการฝึกนานขึ้น แต่ขนาดหมู่ที่เล็กลง ทำให้ในการคำนวณทำงานกับเมตริกซ์ขนาดเล็กลงด้วย. การเลือกขนาดหมู่ เป็นสเมื่อนการหาสมดุลระหว่างเวลาฝึกและขนาดหน่วยความจำ.

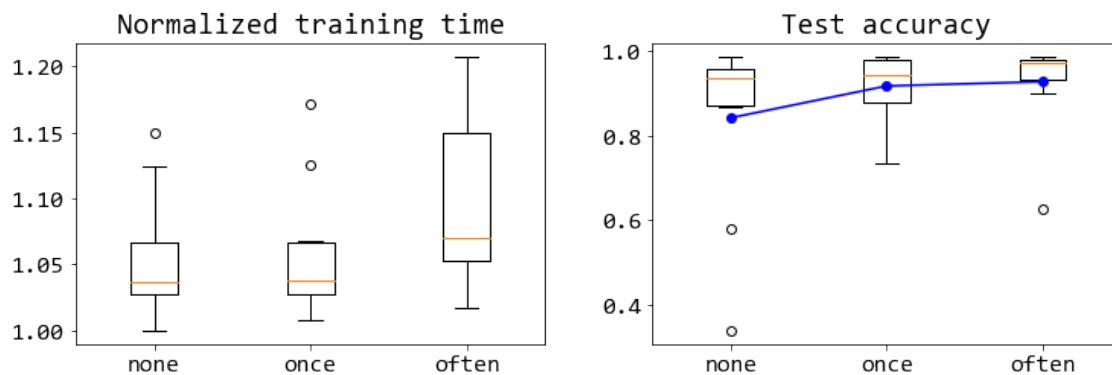
นอกจากนี้ ผลทางอ้อมของการฝึกหมู่เล็ก ยังช่วยให้คุณภาพการฝึกดีขึ้นได้ด้วย (ถ้าทำอย่างเหมาะสม) ดังแสดงในรูปที่ 5.10. กูเพโลและคณะ[76] อภิปรายว่า ขนาดหมู่เล็ก อาจจะช่วยให้ผลในเชิงการเร่งเวลาได้ หรือคุณภาพซับซ้อนของแบบจำลอง ช่วยลดโอกาสการโวเออร์ฟิตข้อมูลลง ซึ่งอาจจะเป็น เพราะผลจากสัญญาณรบกวนจากการฝึกหมู่เล็ก ในกระบวนการหาค่าดีที่สุด. ในทางปฏิบัติ การใช้ขนาดหมู่ที่เล็กลง มักจะทำให้ต้องการจำนวนสมัยฝึกน้อยลง (ถึงแม้แต่สมัย อาจจะใช้เวลาฝึกนานขึ้น).

กลไกที่สำคัญในการฝึกหมู่เล็ก คือการสุมลำดับของข้อมูล. การสุมนี้อาจจะสุมครั้งเดียว และใช้

²เซเชี่ยน (*Hessian*) คือ เมทริกซ์ของอนุพันธ์อันดับที่สอง. ขั้นตอนวิธีการหาค่าดีที่สุดหลายอย่าง อาศัยเซเชี่ยน เพื่อเพิ่มประสิทธิภาพในการทำงาน. ตัวอย่างเช่น วิธีนิวตัน (Newton method) และ วิธีบีอฟจีเอส (BFGS method) ที่สามารถทำงานได้อย่างรวดเร็ว แต่ต้องการเซเชี่ยน. ดู ของและเชค[39] เพิ่มเติมสำหรับรายละเอียดของขั้นตอนวิธีการหาค่าดีที่สุดที่อาศัยเซเชี่ยน.



รูปที่ 5.10 ตัวอย่างคุณภาพการทำนายเมื่อใช้ขนาดหมู่เล็กต่าง ๆ ในด้านคุณภาพการทำนาย ผลอาจจะไม่ได้ชัดเจนในแนวทางโดยอย่างชัดเจนมาก นอกจากการใช้ขนาดหมู่ที่เล็กลง โดยรวมแล้ว ช่วยเพิ่มคุณภาพการทำนายจากการฝึกแบบหมู่ (Batch size 150 ในภาพ). โครงข่ายขนาดสองชั้น สี่ชั้น และสิบชั้น แสดงด้วยสัญลักษณ์ดังระบุในภาพ.



รูปที่ 5.11 ตัวอย่างผลจากการใช้วิธีจัดหมู่เล็กแบบต่าง ๆ. ภาพซ้าย แผนภูมิกล่องแสดงเวลาที่ใช้. ภาพขวา แผนภูมิกล่องแสดงผลการทดสอบของแบบจำลองที่ฝึกแบบหมู่เล็ก โดยจัดหมู่เล็ก (1) ตามลำดับข้อมูล (ไม่มีการเปลี่ยนลำดับ **none**) (2) ตามการสุ่มลำดับโดยสุ่มครั้งเดียวและใช้ลำดับที่สุ่มน้ำ้าตลอดทุกสมัย (**once**) และ (3) ตามการสุ่มลำดับโดยสุ่มใหม่สำหรับแต่ละสมัย (**often**). จุดสีน้ำ้าเงิน แสดงค่าเฉลี่ยความแม่นยำ.

ลำดับนั้นตลอด หรือจะสุ่มทุกสมัยฝึกก็ได้. ในทางปฏิบัติ กุดเพโลและคณะ[76] อภิปรายว่า ผลจากการสุ่มแค่ครั้งเดียว กับการสุ่มทุกสมัยฝึก ไม่ได้ต่างกันมาก แต่สำคัญมาก ๆ ที่ต้องทำการสุ่ม. รูปที่ 5.11 ภาพซ้าย แสดงเวลาในการฝึก เมื่อฝึกหมู่เล็กด้วยลำดับข้อมูลแบบต่าง ๆ ได้แก่ ไม่มีการสุ่มลำดับ, สุ่มลำดับครั้งเดียว, และสุ่มลำดับทุกสมัยฝึก. ภาพขวา แสดงผลความแม่นยำของการทำนาย. จากภาพเวลาในการฝึกเมื่อทำการสุ่มครั้งเดียว ไม่ได้ต่างจากการไม่สุ่มมาก แต่การสุ่มทุกรอบฝึกมีผลในการเพิ่มเวลาฝึกอย่างชัดเจน ในขณะที่ คุณภาพของการฝึก (ความแม่นยำในการทำนาย) การสุ่มครั้งเดียว และการสุ่มทุกสมัย ไม่ได้ต่างกันมาก แต่มีผลต่อกว่าการไม่สุ่มอย่างชัดเจน.

5.3 เทคนิคการตกลอก

เทคนิคการตกลอก (drop out[189]) เป็นกลไกสำหรับการทำเรกุลารีซ์สำหรับโครงข่ายประสาทเทียม โดยได้รับแรงบันดาลใจ จากการทำงานของโครงข่ายประสาททางชีววิทยา ที่โดยรวม ผลการทำงานเชื่อถือได้สูง ในขณะที่ โดยส่วนย่อย เชลล์ประสาทต่าง ๆ ที่เป็นส่วนประกอบของโครงข่ายแต่ละเชลล์มีการทำงานที่เชื่อถือไม่ค่อยได้. นั่นคือ ในขณะที่ แต่ละเชลล์ บางครั้งอาจทำงาน บางครั้งอาจจะไม่ทำงาน แต่ด้วยการที่โครงข่ายมีเชลล์จำนวนมาก และการเข้มต่อได้เตรียมสำหรับความไม่แน่นอนนี้ไว้ ทำให้ผลโดยรวม ยังคงรักษาการทำงานที่เชื่อถือได้สูง.

สำหรับโครงข่ายประสาทเทียม การตกลอก สามารถทำได้โดยการสุ่มเลือกหน่วยคำนวน ที่จะปิดการทำงาน ซึ่ง ทางการคำนวน สามารถทำได้ง่ายๆ โดยคุณด้วยค่าศูนย์. ดังนั้น อาจมองได้ว่า การตกลอก เป็นเสมือน การใช้หน้ากาก หรือค่าสัมประสิทธิ์ของการตกลอก m ไปคูณกับค่าหน่วยคำนวน z โดย ค่าของ m สุ่มมาจากค่าศูนย์หรือหนึ่ง.

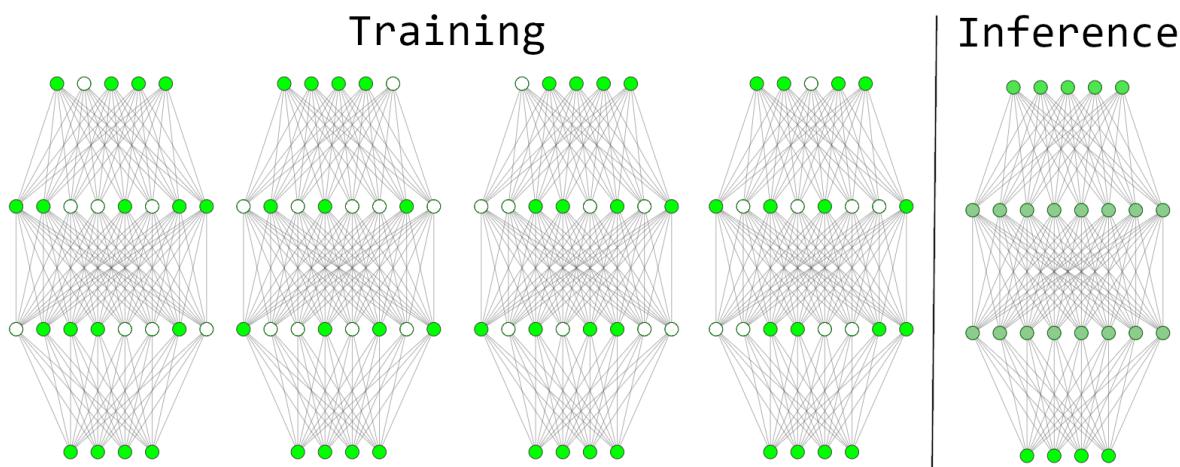
นั่นคือ ค่าหน่วยคำนวนหลังทำการตกลอก \tilde{z} คำนวนได้จาก $\tilde{z} = m \cdot z$ เมื่อ $m \sim \text{Bernoulli}(p)$ โดย $\text{Bernoulli}(p)$ หมายถึง การแจกแจงแบบเบรนูลลี (*Bernoulli distribution*) ที่โอกาสที่ค่า $m = 1$ คือ p นอกนั้น (โอกาส $1 - p$) $m = 0$. การคำนวนการตกลอก เขียนในรูปเวกเตอร์ได้เป็น

$$\tilde{\mathbf{z}} = \mathbf{m} \odot \mathbf{z} \quad (5.4)$$

เมื่อ หน้ากาก \mathbf{m} มีส่วนประกอบแต่ละตัวเป็นค่าที่สุ่มมาจากหนึ่งหรือศูนย์ (การแจกแจงแบบเบรนูลลี).

การตกลอก จะให้ผลในลักษณะคล้ายกับการทำเรกุลารีซ์ นั่นคือ ช่วยคุณสมบัติความทั่วไปของแบบจำลอง. นอกจานั้น ยังเชื่อว่า การตกลอก ยังช่วยเพิ่มความยืดหยุ่น ความทนทานในการเข้มต่อ ในลักษณะที่ช่วยลดการพิงพาคุณลักษณะที่สำคัญไม่เกี่ยวย่างลง และเพิ่มโอกาสที่ทำให้แบบจำลองได้เรียนรู้คุณลักษณะที่สำคัญต่าง ๆ ของรูปแบบได้ครบถ้วนมากขึ้น. การสุ่มปิดการทำงานของหน่วยคำนวน เชื่อว่า น่าจะช่วยหยุดการปรับตัวร่วมกัน (*break co-adaptation*) และน่าจะส่งผลให้เกิดความหลากหลายในการเข้มต่อมากขึ้น. และเพิ่มความหลากหลายในการเข้มต่อ สัมพันธ์โดยตรงกับความยืดหยุ่นกับความทนทานของระบบโดยรวม จึงเชื่อว่า การตกลอก จะช่วยให้แบบจำลองมีความยืดหยุ่น และทนทาน(ต่ออินพุตที่หลากหลาย)ได้ดีขึ้น.

การใช้งานการตกลอก. การสุ่มปิดการทำงาน มักจะใช้เฉพาะตอนฝึกเท่านั้น. ตอนใช้งานอนุมาน จะเปิดการทำงานของทุกส่วน. เนื่องจาก หากฝึกได้ดีพอ ส่วนย่อยต่าง ๆ ในโครงข่ายจะทำงานได้ดีพอ สมควร ดังนั้น หากเปิดทุกส่วนหมดพร้อมกัน จะต้องทำการซัดเซย เพื่อไม่ให้อาตตพุตที่ได้มีค่ามากเกินไป. การซัดเซยนี้ มักถูกเรียกว่า การปรับส่วนค่าน้ำหนัก (*weight scaling*). นั่นคือ หากสุ่มด้วยความน่าจะเป็น 0.5 หมายถึง โดยประมาณ หน่วยคำนวนต่าง ๆ ในโครงข่ายจะทำงานแค่ครึ่งเดียว. ถ้าหากเปิดทุกหน่วยคำนวนพร้อม ๆ กัน จะต้องซัดเซยด้วยการลดความแรงของค่าหน่วยคำนวนลงครึ่งหนึ่ง.



รูปที่ 5.12 ภาพประกอบแสดงแนวคิดของการตอกออก. ภาพแสดงโครงข่ายประสาทเทียมสามชั้น อินพุต (ห้ามติ) อยู่ด้านบน และเอาต์พุต (สีมิติ) อยู่ด้านล่าง ชั้นซ่อนทั้งสองชั้น แต่ละชั้นมีจำนวนหน่วยซ่อนแปดหน่วย. การฝึกโครงข่าย ใช้การตอกออกโดย ความน่าจะเป็นของการคงอยู่เป็น 0.8 สำหรับชั้นอินพุต และ 0.5 สำหรับชั้นซ่อนทั้งสอง. สีโครงข่ายทางชัย แสดงตัวอย่างของการทำงานของโครงข่าย ขณะฝึก ซึ่งแต่ละโครงข่าย สำหรับการคำนวณเกรเดียนต์แต่ละครั้ง. แต่ละครั้ง การตอกออกจะสุ่ม และให้ผลเป็นหน่วยคำนวณที่คงอยู่ต่าง ๆ กันไป. วงกลมสีเขียว แสดงหน่วยที่ทำงาน และวงกลมสีขาว แสดงหน่วยที่ถูกปิด. โครงข่ายทางชยวัด แสดงการทำงานของโครงข่าย ขณะใช้งานอนุมาน. เมื่อใช้งาน ทุกหน่วยคำนวณ จะทำงานพร้อม ๆ กันทั้งหมด แต่ค่าของหน่วยคำนวณจะถูกลดความแรง (ทำการบวกส่วนค่าน้ำหนัก) เพื่อไม่ให้อาตพุ มีค่ามากเกินไป. สีของวงกลมที่ต่างไป สะท้อนการปรับค่าลงของหน่วยคำนวณ โดยชั้นอินพุตปรับลงเป็น 0.8 เท่า และชั้นซ่อนปรับลงเป็น 0.5 เท่า.

และในกรณีที่ π เป็นสำหรับความน่าจะเป็นของการคงอยู่ p และ ค่าหน่วยคำนวณ

$$\mathbf{z}' = p \cdot \mathbf{z} \quad (5.5)$$

เมื่อ \mathbf{z}' คือค่าหน่วยคำนวณหลังการปรับส่วนค่าน้ำหนัก และ p เป็นค่าความน่าจะเป็นของการคงอยู่ และ \mathbf{z} คือค่าหน่วยคำนวณ (ก่อนการปรับส่วนค่าน้ำหนัก).

รูปที่ 5.12 แสดงภาพประกอบแนวคิดของการตอกออก. ขณะฝึก จะมีหน่วยคำนวณบางส่วนถูกปิดไป และส่งผลเสมือนว่า กำลังใช้งานโครงข่ายอยอยอย โดยที่โครงข่ายย่อยจะเปลี่ยนไปแบบสุ่มในการคำนวณเกรเดียนต์แต่ละครั้ง. ขณะใช้งาน ทุกหน่วยคำนวณจะทำงานพร้อมกัน ดังนั้น เพื่อไม่ให้อาตพุมีค่ามากเกินไป ค่าหน่วยคำนวณจะถูกปรับขนาดลงอย่างเหมาะสม. สังเกตว่า การตอกออก จะไม่ทำกับชั้นเอตพุ.

ในทางปฏิบัติ การคำนวณค่าอนุมาน โดยใช้การตอกออกที่ทำในขณะฝึก ค่อนข้างเหละ และทำให้การใช้งานแบบจำลองต้องมีความต้องการมากในการคำน้ำหนักที่ฝึกแล้วมาใช้. นั่นคือ หากระหว่างการฝึก ไม่ได้ใช้การตอกออก ก็ต้องไม่ทำการปรับส่วนค่าน้ำหนัก แต่หากระหว่างการฝึก ทำการตอกออก ก็ต้องทำการปรับส่วนค่าน้ำหนักและปรับส่วนค่าน้ำหนักด้วยค่า p ที่ใช้ (และแต่ละชั้นคำนวณสามารถทำการตอกออก ด้วยค่า p ที่ต่างกันได้). ดังนั้น การนำค่าน้ำหนักที่ฝึกจากหลาย ๆ วิธีมาใช้ จะค่อนข้าง

ยุ่งยากและมีความเสี่ยงมาก รวมถึงยังจำกัดการฝึกด้วยว่า หากฝึกด้วยการตกลอก แล้วต้องทำตลอดทุกสมัยฝึก และใช้ค่า p เท่าเดิมตลอด.

เพื่อลดปัญหาดังกล่าว การทำการตกลอก 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 ยึดหยุ่น สะดวก และลดความเสี่ยงของการซัด夷ผิดลง. การใช้ $\frac{1}{p}$ (ซึ่งมากกว่าหรือเท่ากับหนึ่ง) เทียบเท่า การขยายขนาดของหน่วยคำนวณ ขณะทำการตกลอก ซึ่งจะบังคับให้แบบจำลองเรียนรู้ค่าน้ำหนัก ที่จะไม่ทำให้เอาต์พุตมีค่ามากเกินไป ในอัตราส่วนที่สัมพันธ์กับโอกาสการคงอยู่. ดังนั้น ค่าน้ำหนักที่ได้จึงสามารถนำไปใช้งานได้เลย โดยไม่ต้องทำการปรับส่วนค่าน้ำหนักอีก.

ประโยชน์ของการตกลอก ยังถูกมองว่า เป็นเพิ่มความน่าเชื่อถือได้ของการอนุมาน ในลักษณะคล้ายแนวทางการจัดถุง (bagging). แนวทางการจัดถุง เป็นหนึ่งในแนวทางหลัก ของการประسانการเรียนรู้ (ensemble learning). การประسانการเรียนรู้ เป็นเทคนิคของการเรียนรู้ของเครื่องเพื่อปรับปรุงคุณภาพการอนุมาน โดยใช้ค่าทำนายจากหลาย ๆ แบบจำลอง และนำค่าทำนายต่าง ๆ เหล่านั้นมาสรุปรวมเป็น ค่าทำนายของการประسانการเรียนรู้. วิธีการสรุปอาจทำได้หลายแบบ ขึ้นกับภารกิจการทำนาย เช่น หากเป็นการทำนายค่าตัดตอน (เอาต์พุต $y \in \mathbb{R}$) อาจใช้ค่าเฉลี่ยจากค่าทำนายของแบบจำลองต่าง ๆ. แต่หากเป็นการทำนายแบบกลุ่ม (เอาต์พุต $y \in \{1, \dots, K\}$ เมื่อ K เป็นจำนวนกลุ่ม) อาจสรุปโดยการลงคะแนนเสียง (vote) นั่นคือ การใช้ค่าฐานนิยม หรือสรุปเป็นกลุ่มที่ถูกจำแนกมากที่สุด (ซึ่งอาจต้องการกลยุทธ์ในการจัดการกับกรณีเสมอ กัน).

ในขณะที่ การประسانการเรียนรู้ เป็นเทคนิคแนวทางกว้างๆ ที่เน้นการนำผลทำนายจากหลาย ๆ แบบจำลอง มาสรุปร่วมกัน. แนวทางการจัดถุง เป็นแนวทางการเตรียมแบบจำลองต่างๆ สำหรับใช้ในการประسانการเรียนรู้.

แบบจำลองต่างๆ ที่กล่าวถึงในการประسانการเรียนรู้ หมายถึง พังก์ชันการทำนายใดๆ ที่สร้างมาต่างกัน อาจจะโดยมีโครงสร้างทางคณิตศาสตร์ที่ต่างกัน (เช่น โครงข่ายประสาทเทียมสามชั้น กับโครงข่ายประสาทเทียมห้าชั้น หรือโครงข่ายประสาทเทียม กับเซ็ตเวกเตอร์แมชชีน) หรืออาจจะโดยมีโครงสร้างทางคณิตศาสตร์ที่เหมือนกัน แต่ผ่านกระบวนการฝึกที่ต่างกัน เช่น ใช้ข้อมูลในการฝึกที่ต่างกัน หรือต่างกันทั้งโครงสร้างทางคณิตศาสตร์และกระบวนการฝึก.

แนวทางการจัดถุง เน้นการเตรียมแบบจำลองที่ต่างกัน ด้วยการใช้ข้อมูลฝึกที่ต่างกัน นั่นคือ หากต้องการเตรียม M แบบจำลองสำหรับใช้ในการประسانการเรียนรู้ จากชุดข้อมูลฝึกที่มีจำนวนจุดข้อมูล เป็น N แนวทางการจัดถุง จะสร้างข้อมูลสำหรับฝึกขึ้นมา M ชุด โดยแต่ละชุด จะสุมจุดข้อมูลจากชุดฝึกมาแบบหยิบคืน (sample with replacement) โดยจำนวนข้อมูลในแต่ละชุด N' เป็นอภิมานพารามิเตอร์ของการจัดถุง. จากนั้น แบบจำลองแต่ละตัว จะถูกฝึกกับข้อมูลที่สร้างขึ้นแต่ละชุด และแบบจำลองทั้งหมดที่ฝึกเสร็จ ก็จะสามารถนำไปใช้ในการประسانการเรียนรู้ได้.

การตกลอก ถูกมองว่า เป็นกลไกในลักษณะคล้ายแนวทางการจัดถุ่ง จากการที่ ขณะฝึก การปรับค่าน้ำหนักแต่ละครั้ง จะมีเฉพาะบางส่วนของโครงข่ายเท่านั้นที่จะถูกปรับค่า ส่วนที่ถูกปิดการทำงานจะไม่ได้ถูกปรับค่าน้ำหนัก ดังนั้น เมื่อใช้งาน และเปิดการทำงานของทุกส่วน จึงคล้ายการประสานการเรียนรู้ของส่วนย่อยต่าง ๆ ภายในโครงข่าย แต่การตกลอก ก็ไม่ได้ทำให้การใช้งานโครงข่ายประสาน เที่ยมเหมือนการประสานการเรียนรู้แบบดั้งเดิม เพราะว่า โดยทั่วไปแล้ว ส่วนต่าง ๆ ของโครงข่ายที่ เปิดและปิดขณะฝึกจากกลไกของการตกลอก จะมีการซ้อนทับกันอยู่มาก ซึ่งต่างจาก การประสานการเรียนรู้แบบดั้งเดิม ที่แต่ละแบบจำลองมีความเป็นอิสระต่อกันสูงกว่ามาก อย่างไรก็ตาม ด้วยเหตุผลดัง นี้ ในการตกลอก อาจทำให้การฝึกทำได้ช้าลง ศรีวิสาทฯและคณะ[189] รายงานว่า การตกลอก ให้ผลช่วยการฝึกได้ดีกว่าการทำเรกูลาริซซ์ราย ๆ วิธี รวมถึง วิธีค่าน้ำหนักเลื่อน ข้อเสียของการใช้ การตกลอกที่สำคัญ ก็เช่น[76] อาจทำให้การฝึกทำได้ช้าลง และอาจทำให้ต้องการแบบจำลองที่ใหญ่ ขึ้น ลักษณะเดียวกับการทำเรกูลาริซซ์ การตกลอก อาจไม่ได้ช่วยมาก หากข้อมูลที่ฝึกมีปริมาณมาก ซึ่งประโยชน์ที่ได้จากการทำการตกลอก อาจจะน้อยกว่าข้อเสียที่จะทำให้ต้องการแบบจำลองใหญ่ขึ้น และทำให้การฝึกช้าลง.

นอกจากเทคนิคการตกลอกแล้ว เทคนิคการตกลอก ยังเป็นแรงบันดาลใจให้เกิดการพัฒนาเทคนิค อื่น ๆ ที่คล้าย ๆ กันจำนวนมาก เช่น การตกลอกเร็ว (fast drop out[209]), การส่งเสริมการตกลอก (dropout boosting[212]), การเชื่อมตกลอก (DropConnect[207]) อย่างไรก็ตาม ด้วยผลลัพธ์การทำงาน ประสิทธิภาพ และความสะดวกของการใช้งาน การตกลอก เป็นแนวทางที่ได้รับความนิยมสูง กว่าวิธีที่พัฒนาต่อ ๆ ขึ้นมาเหล่านี้ ถูกเพโลและคณะ[76] อภิปรายสรุปว่า กลไกสำคัญที่เทคนิคการตกลอกเป็นตัวแทน คือ การใส่ความไม่แน่นอนเข้าไปในการฝึกโครงข่าย และทำการอนุมานโดยสรุป จากผลต่าง ๆ ที่ผ่านความไม่แน่นอน ซึ่งในผลในลักษณะการการจัดถุ่ง โดยมีการใช้พารามิเตอร์ร่วม กัน การใส่ความไม่แน่นอน เข้าไปจะช่วยให้แบบจำลองเรียนรู้ที่จะยึดหยุ่นขึ้น และครอบคลุมขึ้น ซึ่ง อาจจะคล้ายกับคน ที่เรียนรู้ที่จะยึดหยุ่นขึ้นและรอบคอบขึ้น เมื่อคำนึงความไม่แน่นอนที่อาจเกิดขึ้น คณะของศรีวิสาทฯ[189] ได้ทดลองใช้หน้ากากค่าจริง $\mathbf{m} \sim \mathcal{N}(\mathbf{1}, \mathbf{I})$ (การแจกแจงปกติ ที่มีค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานเป็นหนึ่ง ไม่มีสหสัมพันธ์ระหว่างตัวแปร) แทนการแจกแจงเบรนูลลี่ (สมการ 5.4) และพบว่า ได้ผลการทำงานที่ดีเช่นกัน นอกจากนั้น หน้ากากค่าจริงนี้ มีค่าคาดหมาย $E[\mathbf{m}] = \mathbf{1}$ จึงไม่ต้องทำการปรับส่วนค่าน้ำหนัก.

5.4 การกำหนดค่า \mathbf{n} ให้กับเริ่มต้น

การฝึกโครงข่ายประสาทเทียมแบบลีก ใช้ขั้นตอนวิธีการหาค่าดีที่สุดที่อาศัยเกรเดียนต์. ค่าเริ่มต้นของตัวแปร มีผลอย่างมากต่อการทำงานการหาค่าดีที่สุด เช่นการเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์พอดี จะช่วยทำให้การฝึกโครงข่ายทำได้ง่ายและเร็วขึ้น.

ในขณะที่การเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์เปลี่ยนแปลงอย่างรุนแรง อาจนำไปสู่ปัญหาเสถียรภาพของการฝึก หรือหากเริ่มต้นในตำแหน่งที่ค่าเกรเดียนต์มีค่าน้อยมาก (มักอ้างถึงด้วยคำว่า “ทีราบ” หรือ plateau) อาจทำให้การฝึกไม่ก้าวหน้า หรือหยุดชะงักได้.

เนื่องจากความเข้าใจในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมยังไม่กระจ่างชัดสมบูรณ์ ปัจจัยต่าง ๆ ของการฝึก รวมถึงการกำหนดค่า \mathbf{n} ให้กับเริ่มต้น จึงยังไม่มีข้อสรุปที่แน่นชัด. นอกจาก สิ่งหนึ่งที่จำเป็น คือ ค่า \mathbf{n} ให้กับเริ่มต้น ต้องช่วยลดการปรับตัวไปเมื่อ่อน ๆ กัน (มักอ้างถึงเป็นการซ่วย break symmetry).

การปรับตัวไปเมื่อ่อน ๆ กัน มาจากการที่โครงข่ายประสาทเทียมมีวิธีการคำนวณแต่ละหน่วยคำนวณเหมือน ๆ กัน. ดังนั้นการที่แต่ละหน่วยคำนวณเริ่มต้นด้วยค่าเดียวกัน จะทำให้มันมีค่าเกรเดียนต์เท่ากัน และถูกปรับค่าไปเท่า ๆ กัน จนสุดท้าย แต่ละหน่วยคำนวณจะทำงานเหมือนกัน ตอบสนองกับรูปแบบย่อยเดียวกัน ไม่ได้แยกกันรับผิดชอบแต่ละรูปแบบย่อย ๆ ทำให้ความสามารถโดยรวมของโครงข่าย ที่แม้จะมีจำนวนหน่วยคำนวณมาก แต่ให้ประสิทธิผลการทำงานเมื่อ่อนโครงข่ายที่มีหน่วยคำนวณน้อย (หรือ ในกรณีสุดโต่ง อาจทำงานเมื่อันมีหน่วยคำนวณเดียว).

นอกจากลดการปรับตัวไปเมื่อ่อนกัน อีกปัจจัยหนึ่งที่สำคัญในการกำหนดค่าเริ่มต้น คือ ขนาดของค่า \mathbf{n} ให้กับเริ่มต้น คือ การใช้การสุ่มค่า เพื่อกำหนดค่า \mathbf{n} ให้กับเริ่มต้น. การกำหนดค่า \mathbf{n} ให้กับเริ่มต้นด้วยการสุ่มจากการแจกแจงที่มีอนโตรปีสูง (เช่น การแจกแจงเอกรูป) สามารถทำได้ง่ายๆ และนำมายोกานสนอยมาก ที่ค่า \mathbf{n} จะไปเริ่มต้นที่เดียวกัน แม้ว่าจะมีพารามิเตอร์ค่า \mathbf{n} จำนวนมาก.

กฎเพโลและคณ[76] อภิปรายว่า แทนที่การสุ่ม เราอาจจะคำนวณหาชุดค่า \mathbf{n} ที่แต่ละชุดแตกต่างกันมาก ๆ ได้ เช่น ในกรณีที่เหมาะสม อาจใช้ขั้นตอนวิธีแกรมชmidt³ แต่แนวทางนี้ มักจะเพิ่มภาระการคำนวณก่อนการฝึกขึ้นมาก และภาระการคำนวณก่อนการฝึกที่เพิ่มขึ้นมากนี้ อาจไม่คุ้มกับผลประโยชน์ที่ช่วยลดการภาระการคำนวณระหว่างการฝึกลง เมื่อเปรียบเทียบกับการใช้แนวทางการสุ่ม.

ค่าพารามิเตอร์ \mathbf{n} นิยมกำหนดค่าเริ่มต้นด้วยการสุ่ม ส่วนค่าพารามิเตอร์ \mathbf{b} อาจกำหนดค่าเริ่มต้นเป็นค่าคงที่ หรืออาจจะสุ่มค่า เช่นเดียวกันก็ได้[76]. การสุ่มค่า \mathbf{n} มักนิยมสุ่มจากการแจกแจงเอกรูป หรือการแจกแจงปกติ. ค่าการแจกแจงที่นิยม[74] คือ กำหนดค่าเริ่มต้น \mathbf{n}

³ ขั้นตอนวิธีแกรมชmidt (Gram-Schmidt algorithm) เป็นวิธีคำนวณหาเวกเตอร์ที่ตั้งฉากกับเวกเตอร์ที่กำหนด. ดู [39] เพิ่มเติมสำหรับรายละเอียด.

จากการแจกแจงเอกรูป $\mathcal{U}\left(-\frac{1}{\sqrt{m_i}}, \frac{1}{\sqrt{m_i}}\right)$ เมื่อ m_i เป็นจำนวนอินพุตของชั้นคำนวณ (อาจอ้างถึงว่าเป็น “จำนวนแผ่นเข้า” หรือ a number of fan-in units). ค่า $\frac{1}{\sqrt{m_i}}$ เพื่อป้องกันไม่ให้ผลคำนวณมีค่าใหญ่เกินไป จนผลเสียต่อสเกลรูปภาพของการฝึก สำหรับโครงข่ายขนาดใหญ่.

ตัวอย่างเช่น หากชั้นคำนวณ ทำ $\mathbf{a} = \mathbf{w}^T \mathbf{x} + \mathbf{b}$ กับ $\mathbf{z} = h(\mathbf{a})$ เมื่อ h เป็นฟังก์ชันกระตุ้น และ อินพุตของชั้น $\mathbf{x} \in \mathbb{R}^{m_i}$. 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 โกลโลต์ และโยชัว เบนจิโอ[74] ศึกษาความยากของการฝึกโครงข่ายประสาทเทียม ตีความผลที่ได้ และเมื่อประกอบกับผลงานศึกษาของเบรดลีย์[22] ที่พบว่าความแปรปรวน (variance) ของค่าเกรเดียนต์ที่แพร่กระจายย้อนกลับ ลดลงเรื่อยๆ ตามชั้นที่ย้อนกลับ ทั้งคุ้สันนิษฐานว่า หากความแปรปรวน ของผลการกระตุ้น \mathbf{z} และความแปรปรวนค่าเกรเดียนต์ ของแต่ละชั้นคำนวณ มีค่าพอๆ กัน จะช่วยให้สารสนเทศไหลผ่านได้ดีขึ้น และจะช่วยให้การฝึกโครงข่ายทำได้สะดวกขึ้น. จากข้อสันนิษฐานดังกล่าว ทั้งคุวิเคราะห์ความแปรปรวนของของแต่ละชั้นคำนวณโดยประมาณ (อาศัยสมมติฐานหลายอย่าง รวมถึงสมมติฐานเชิงเส้น) และเสนอว่า ควรกำหนดค่าเริ่มต้นสำหรับค่าน้ำหนักโดยให้

$$\forall l, \text{var}[\mathbf{w}^{(l)}] = \frac{2}{m_i^{(l)} + m_o^{(l)}} \quad (5.8)$$

เมื่อ $\text{var}[\mathbf{w}^{(l)}]$ คือความแปรปรวนของค่าน้ำหนักชั้นคำนวณที่ l^{th} และ $m_i^{(l)}$ คือจำนวนแผ่นเข้า และ $m_o^{(l)}$ คือจำนวนหน่วยคำนวณในชั้น (หรือจำนวนเอ้าต์พุตของชั้นคำนวณ ที่อาจอ้างถึงเป็น “จำนวนออก” หรือ a number of fan-out units). เมื่อนำมาใช้กับการแจกแจงเอกรูป 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 การกำหนดค่าน้ำหนักด้วย **วิธีเชเวียร์** (Xavier weight initialization).

การวิเคราะห์ของโกลโลต์และเบนจิโอ[74] คิดจากฟังก์ชันกระตุ้นไฮเปอร์บอลิกแทนเจนต์ (\tanh) และฟังก์ชันกระตุ้นเครื่องหมายอ่อน (softsign , $h(a) = \frac{a}{1+|a|}$). ทั้งคู่เป็นฟังก์ชันที่สมมาตรที่ศูนย์⁴ คอมิ่ง เห้อ และคณะ[84] พบว่า เงื่อนไขที่โกลโลต์และเบนจิโอวิเคราะห์ อาจจะไม่เหมาะสม เมื่อพิจารณา ฟังก์ชันกระตุ้นที่ไม่สมมาตรที่ศูนย์ เช่น ฟังก์ชันReLU ที่นิยมใช้กับโครงข่ายลึก. ตามแนวทางของโกลโลต์

⁴ ในเมื่อว่า ค่าห่างจากศูนย์ไปทางซ้ายและขวาเท่าๆ กัน จะหักล้างกันได้.

และเป็นจิโอ คณะของไอคอมิค เห้อ[84] ทำการวิเคราะห์เงื่อนไขของค่า a หนัก โดยพิจารณาฟังก์ชันกระตุนเรลู และฟังก์ชันอื่นในลักษณะคล้ายกัน. ฟังซั่งตรรกะลู 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 α คือ พารามิเตอร์ของฟังก์ชัน. หาก $\alpha = 0$ จะทำให้ $h(a)$ เป็นฟังก์ชันเรลู. หาก $\alpha > 0$ เป็นค่าคงที่ โดยเป็นอภิมานพารามิเตอร์ที่กำหนดโดยผู้ใช้ จะทำให้ $h(a)$ เป็นฟังก์ชันเรลูร์ว (leaky relu). นอกจากนั้น คณะของเห้อ ได้เสนอฟังก์ชันกระตุนที่สามารถปรับตัวได้ โดยให้ $\alpha > 0$ เป็นค่าพารามิเตอร์ที่ถูกฝึกไปพร้อม ๆ กับค่า a หนักและไบอส และคณะของเห้อ เรียกฟังก์ชันกระตุนนี้ว่า ฟังก์ชันพีเรลู (PReLU).

เงื่อนไขที่คณะของเห้อเสนอ แสดงในสมการ 5.11.

$$\text{var}[w^{(l)}] = \frac{2}{(1 + \alpha^2) \cdot m_i^{(l)}} \quad (5.11)$$

เมื่อ α เป็นพารามิเตอร์ของฟังก์ชันตรรกะลู.

เมื่อนำเงื่อนไขในสมการ 5.11 ไปใช้กับการแจกแจงปกติ (ที่มักนิยมใช้กับเงื่อนไขของคณะของเห้อ) จะได้ว่า

$$w_{kj} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{(1 + \alpha^2) \cdot m_i}}\right). \quad (5.12)$$

การกำหนดค่า a หนักเริ่มต้น ด้วยนิพจน์ 5.12 รู้จักกันทั่วไปในชื่อ การกำหนดค่า a หนักด้วยวิธีไอคอมิค (Kaiming weight initialization). สังเกต หากพิจารณากรณีฟังก์ชันกระตุนเรลู ($\alpha = 0$) ความแปรปรวนของค่า a หนัก จากเงื่อนไขไอคอมิค คือ $\frac{2}{m_i}$. ในขณะที่นิพจน์ 5.7 ส่งผลให้ ความแปรปรวนของค่า a หนัก คือ $\frac{1}{3 \cdot m_i}$ (โดยนิพจน์ 5.7 ไม่คำนึงถึงฟังก์ชันกระตุน).

นอกจาก การกำหนดค่า a หนักเริ่มต้น ด้วยการสุ่มดังอภิรายนี้แล้ว การทำการฝึกก่อน (หัวข้อ 5.5) เพื่อได้ค่า a หนักที่ดี ก่อนที่จะทำการฝึกแบบจำลองสำหรับภารกิจที่ต้องการจริงๆ เป็นแนวทางหนึ่งที่ให้ผลดีมากในทางปฏิบัติ.

5.5 กลไกช่วยการฝึก

ขณะที่วิธีลงเกรเดียนต์ หรือมักนิยมเรียก วิธีลงเกรเดียนต์สโตกแคลสติก (stochastic gradient descent) ที่เน้นถึงการสุ่มลำดับของการฝึกทีละหมู่เล็ก เป็นวิธีที่นิยมใช้ในการฝึกโครงข่ายประสาทเทียม.

แต่ปัจจุบันที่อาจพบว่า วิธีลงเกรเดียนต์สโตร์กัสติก ทำให้การฝึกทำได้ช้า. หลาย ๆ เทคนิคจากศาสตร์การหาค่าดีที่สุด ได้ถูกนำมาใช้ เพื่อปรับปรุงประสิทธิภาพการฝึก. นอกจากนี้ ยังมีเทคนิคจำนวนมากที่พัฒนาขึ้นมาโดยเฉพาะสำหรับการเรียนรู้ของเครื่อง โดยเฉพาะการเรียนรู้เชิงลึก. หัวนี้ ยกไปรายเทคนิคต่าง ๆ บางส่วน โดยเฉพาะ เทคนิคเด่น ๆ ที่มีการใช้อย่างกว้างขวางกับการเรียนรู้เชิงลึก.

กลไกโมเมนตัม

ในสถานะการณ์ที่ค่าฟังก์ชันจุดประสังค์ต่อค่าตัวแปรตัดสินใจต่าง ๆ (ซึ่งหมายถึง ฟังก์ชันสูญเสียและค่านำหนักและใบอัสหั้งห้าย ในกรณีโครงข่ายประสาทเทียม) มีลักษณะความสัมพันธ์ที่มีการเปลี่ยนแปลงเร็ว หรืออาจจะมีสัญญาณรบกวนมาก กลไกของโมเมนตัม (momentum[153]) นิยมถูกนำมาใช้เพื่อช่วยเพิ่มประสิทธิภาพการทำงานของขั้นตอนวิธีการหาค่าดีที่สุด.

รูปที่ 5.13 แสดงพฤติกรรมการทำงานของวิธีลงเกรเดียนต์ เมื่อ ความสัมพันธ์ของฟังก์ชันสูญเสีย กับค่าน้ำหนัก ที่มีลักษณะโค้ง แต่ความโค้งแตกต่างกันมากระหว่างน้ำหนักแต่ละตัว. พฤติกรรมการทำงานปรับค่าน้ำหนัก ของวิธีลงเกรเดียนต์ จะแสดงออกในลักษณะส่ายเข้าหากำตอบ.

แทนที่จะใช้ค่าเกรเดียนต์เพียงอย่างเดียว กลไกของโมเมนตัม เสนอที่จะใช้ ทิศทางเดิม ประกอบกับทิศทางใหม่ เพื่อลดการส่ายเข้าหากำตอบ เพื่อปรับค่าตัวแปร. นั่นคือ

$$\boldsymbol{v}^{(i+1)} = \beta \boldsymbol{v}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)}) \quad (5.13)$$

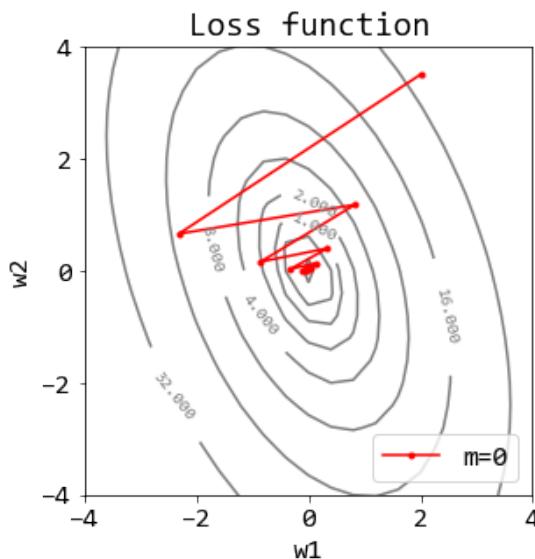
$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \boldsymbol{v}^{(i+1)} \quad (5.14)$$

เมื่อ \boldsymbol{v} เป็นเวกเตอร์สำหรับปรับค่าตัวแปร และ β เป็นค่าโมเมนตัม. ส่วน $\nabla L(\boldsymbol{\theta}^{(i)})$ คือ เกรเดียนต์ ต่อตัวแปรตัดสินใจ และ α คืออัตราการเรียนรู้. และ $\boldsymbol{\theta}$ เป็นตัวแปรตัดสินใจ เช่น ค่าน้ำหนัก และใบอัส. ตัวยก ระบุสมัยฝึก. ค่าเริ่มต้นของทิศทางเดิม \boldsymbol{v} อาจกำหนดเป็น $\mathbf{0}$.

เปรียบเทียบกับ $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)})$ ซึ่งเป็นวิธีลงเกรเดียนต์ที่ไม่มีกลไกโมเมนตัม จะเห็นว่า หากให้ $\beta = 0$ นั่นเท่ากับปิดกลไกโมเมนตัม และการทำงานของสมการ 5.13 และ 5.14 จะลดรูปมาเป็นวิธีลงเกรเดียนต์ดั้งเดิม.

รูปที่ 5.14 แสดงตัวอย่างที่กลไกโมเมนตัม ช่วยปรับปรุงประสิทธิภาพการฝึก โดยลดการส่ายเข้าหากำตอบระหว่างการฝึก. นอกจาก กลไกของโมเมนตัม ซึ่งเป็นเทคนิคที่รู้จักกันดีในวงการการหาค่าดีที่สุดอยู่แล้ว อีเลีย ซุตส์เกเวอร์ (Ilya Sutskever) นักวิจัยชั้นนำด้านการเรียนรู้ของเครื่อง ได้เสนอ เนลเตอรอฟโมเมนตัม (Nesterov momentum[192]) ซึ่งคำนวณสมการ 5.15 แทนสมการ 5.13

$$\boldsymbol{v}^{(i+1)} = \beta \boldsymbol{v}^{(i)} - \alpha \nabla L(\boldsymbol{\theta}^{(i)} + \beta \boldsymbol{v}^{(i)}). \quad (5.15)$$



รูปที่ 5.13 พฤติกรรมซิกแซก. ภาพคอนทัวร์ของฟังก์ชันสูญเสียต่อตัวแปรค่าน้ำหนักสองตัว w_1 และ w_2 พร้อมเส้นทางการปรับค่าน้ำหนัก. เส้นสีเทา แสดงระดับค่าของฟังก์ชันสูญเสีย. เส้นทึบสีแดง แสดงเส้นทางการปรับค่าน้ำหนัก ด้วยวิธีลิงเกรเดียนต์. ในภาพ ค่าเริ่มต้นจาก $(w_1, w_2) = (2, 3.5)$ (บริเวณด้านบนทางขวา). จุดที่ค่าฟังก์ชันสูญเสียต่ำสุดอยู่ที่ $(0, 0)$ (กลางภาพ). สังเกตเส้นทางการปรับค่าตัวแปร และเป็นลักษณะซิกแซก.

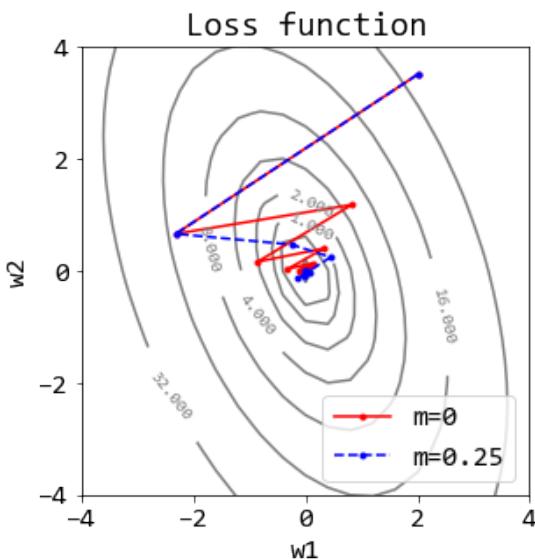
เปรียบเทียบกับสมการ 5.13 เนสเตอร์อพโนเมนตัม ใช้ค่าเกรเดียนต์ ที่คำนวณ ณ ตำแหน่งค่าตัวแปรที่ขยับต่อออกมาตามโนเมนตัม แทนตำแหน่งค่าตัวแปรปัจจุบัน. ดูแบบฝึกหัด 5.20 เพิ่มเติมสำหรับการใช้งานกลไกโนเมนตัม.

ขั้นตอนวิธีที่ปรับค่าอัตราเรียนรู้

อดาแกรต. อดาแกรต (AdaGrad) ปรับอัตราเรียนรู้สำหรับพารามิเตอร์แต่ละตัว โดยลดขนาดอัตราเรียนรู้ลง ตามขนาด rak ที่สองของผลรวมกำลังสองของเกรเดียนต์ที่ผ่านมา. นั่นคือ พารามิเตอร์ θ 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 α คืออัตราเรียนรู้(ฐาน) ที่ผู้ใช้กำหนด และ ϵ คือค่าคงที่ขนาดเล็ก เช่น 10^{-7} สำหรับเสถียรภาพการคำนวณ. ค่า $r^{(0)}$ อาจกำหนดเป็น $\mathbf{0}$. สังเกตว่า อดาแกรต ปรับอัตราการเรียนรู้แยกกันสำหรับพารามิเตอร์แต่ละตัว. ในทางปฏิบัติพบว่า อดาแกรต ใช้งานได้ดีบางครั้ง และอาจลดอัตราเรียนรู้มากเกินไปในบางครั้ง[76].



รูปที่ 5.14 ภาพแสดงการทำงานของกลไกโมเมนตัม (สเน็ปปรัชโน้ต้าเงิน) เปรียบเทียบกับ การไม่ใช้มोเมนตัม (สเน็ทบีสีแดง). พื้นหลังแสดงคอนัวร์ของฟังก์ชันสูญเสียต่อค่าตัวแปร. ในภาพ ทั้งสองวิธีเริ่มต้นจาก $(w_1, w_2) = (2, 3.5)$ (บริเวณด้านบนทางขวา). จุดที่ค่าฟังก์ชันสูญเสียต่ำสุดอยู่ที่ $(0, 0)$ (กลางภาพ). สังเกต โมเมนตัมช่วยลดการส่ายของสเน็ททางการปรับค่าตัวแปรลง.

อาร์เอมเอสพรอป. อาร์เอมเอสพรอป (*RMSProp*) ปรับปรุงอัตราการตัวอย่างการใช้ค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลัง (*exponentially weighted moving average*). นั่นคือ พารามิเตอร์ θ จะถูกปรับค่าโดย

$$\theta^{(i+1)} = \theta^{(i)} - \frac{\alpha}{\sqrt{r^{(i+1)} + \epsilon}} \odot g \quad (5.17)$$

เมื่อค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลัง $r^{(i+1)} = \rho \cdot r^{(i)} + (1 - \rho) \cdot g \odot g$ โดยอัตราการเสื่อมน้ำหนัก ρ เป็นอภิมานพารามิเตอร์ที่เพิ่มขึ้นมา และค่าคงที่ ϵ มักถูกเลือกเป็น 10^{-6} .

แม้ว่า การเสนอาร์เอมเอสพรอปครั้งแรกไม่ได้ถูกเผยแพร่ด้วยช่องทางปกติสำหรับงานวิชาการ (การตีพิมพ์ในวารสารหรือการประชุมวิชาการ) แต่เป็นส่วนหนึ่งของการบรรยายในการสอนออนไลน์ [86] ในทางปฏิบัติ อาร์เอมเอสพรอปเป็นหนึ่งในขั้นตอนวิธีที่ใช้งานได้ดี และถูกใช้อย่างแพร่หลาย สำหรับการฝึกแบบจำลองเชิงลึก [76].

อดัม. อดัม (Adam [110] ย่อจาก adaptive moments) รวมอาร์เอมเอสพรอป เข้ากับโมเมนตัมโดยเพิ่มกลไกค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบชี้กำลังกับการคำนวณโมเมนตัม และการปรับแก้ขนาดตามสมัยผีก. นั่นคือ สำหรับสมัยผีก i และเกรเดียนต์ g การปรับค่าพารามิเตอร์สามารถทำได้ดัง

สมการ 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 α คืออัตราเรียนรู้(ฐาน), ϵ คือค่าคงที่ขนาดเล็ก (ซึ่งอาจใช้ 10^{-8}), ค่า $\rho_1 \in [0, 1]$ และ $\rho_2 \in [0, 1]$ โดย ค่าที่แนะนำคือ $\rho_1 = 0.9$ และ $\rho_2 = 0.999$.

อดัม เป็นอีกขั้นตอนวิธีที่นิยมใช้กับแบบจำลองเชิงลึก และพบว่าค่อนข้างทนทานต่อค่าอภิมานพารามิเตอร์ที่เลือก แต่อาจจะต้องปรับค่าอัตราเรียนรู้ α บ้างเท่านั้น.

ปัจจุบันยังไม่มีข้อสรุปถึงขั้นตอนวิธีที่ดีที่สุดโดยทั่วไป แต่ขั้นตอนวิธีที่นิยมใช้คือ[76] วิธีลงเกรเดียนต์, วิธีลงเกรเดียนต์กับโมเมนตัม, อาร์เอมแอกซ์พรอป, อาร์เอมแอกซ์พรอปกับโมเมนตัม, และอดัม.

แบบนอร์มอย่างเช่น

แบบนอร์มอย่างเช่น (batch normalization) หรือเรียกสั้นว่า แบบนอร์ม (batch norm) จะริง ๆ แล้ว ไม่ใช้ขั้นตอนวิธีการหาค่าดีที่สุด แต่เป็นกลไกเพื่อช่วยให้การฝึกทำได้ง่ายขึ้น.

ไอโอฟิกบเซเจดี[93] ตั้งข้อสังเกตว่า ความยากของการฝึกโครงข่ายเชิงลึก ส่วนหนึ่งมาจากการเปลี่ยนแปลงอยู่ตลอดของการแยกแจงของอินพุตสำหรับแต่ชั้นคำนวณ ซึ่งการเปลี่ยนแปลงนี้ เกิดจาก การเปลี่ยนแปลงของค่าพารามิเตอร์ในชั้นคำนวณก่อนหน้า. ดังนั้น การฝึกจึงทำได้ช้า เพราะไม่สามารถเลือกค่าอัตราเรียนรู้ที่สูงได้ และยังต้องระวังอย่างมากในการกำหนดค่าพารามิเตอร์เริ่มต้น และยังสร้างปัญหาอย่างมากกับการใช้ฟังก์ชันกระตุ้นที่มีช่วงอิมตัว (เช่น ซิกมอยด์). ไอโอฟิกบเซเจดี เรียก การเปลี่ยนแปลงของการแยกแจงของอินพุตสำหรับแต่ชั้นคำนวณ จากการเปลี่ยนแปลงของค่าพารามิเตอร์ ในชั้นคำนวณก่อนหน้า ว่า การเลื่อนของความแปรปรวนร่วมเกี่ยวภายใน (internal covariance shift) และเสนอกลไก แบบนอร์ม เพื่อลดการเลื่อนของความแปรปรวนร่วมเกี่ยวภายใน.

หากกำหนดให้ $\mathbf{X} = [x_{dn}]$ เป็นอินพุตของชั้นคำนวณ โดย $d = 1, \dots, D; n \in B$ และ D เป็นจำนวนมิติ และ B เป็นเซตของดัชนีจุดข้อมูลในหมู่เล็ก แล้ว แบบนอร์ม เสนอแปลงอินพุตของชั้นคำนวณนี้ ดังสมการ 5.23 และ 5.24. ค่าคงที่ขนาดเล็ก ϵ มีเพื่อรักษาเสถียรภาพของการคำนวณ (อาจกำหนดให้ $\epsilon = 10^{-8}$).

$$x'_{dn} = \frac{x_{dn} - \mu_d}{\sqrt{\sigma_d^2 + \epsilon}}, \quad (5.23)$$

$$\hat{x}_{dn} = \gamma_d \cdot x'_{dn} + \beta_d. \quad (5.24)$$

ค่า γ_d กับ β_d เป็นพารามิเตอร์ของแบบนอร์ม ที่เรียนรู้ระหว่างการฝึก. ส่วน μ_d และ σ_d^2 คือค่าเฉลี่ยและความแปรปรวน นั้นคือ ในระหว่างการฝึก $\mu_d = \frac{1}{|B|} \sum_{n \in B} x_{dn}$ กับ $\sigma_d^2 = \frac{1}{|B|} \sum_{n \in B} (x_{dn} - \mu_d)^2$.

สำหรับการใช้งานอนุมานหลังฝึกเสร็จ ค่า μ_d และ σ_d^2 สามารถใช้ค่าประมาณ ที่เตรียมไว้ระหว่างการฝึกได้. ค่าประมาณสำหรับใช้งานอนุมานหลังฝึกเสร็จ นิยมประมาณด้วยค่าเฉลี่ยเคลื่อนที่ ถ่วงน้ำหนักแบบซึ่งกำลัง เช่น $\hat{\mu}_d^{(new)} = \rho \cdot \mu_d + (1 - \rho) \cdot \hat{\mu}_d^{(old)}$ และ $\hat{\sigma}_d^{(new)} = \rho \cdot \sigma_d^2 + (1 - \rho) \cdot \hat{\sigma}_d^{(old)}$ เมื่อ ค่าเสื่อมน้ำหนัก ρ เป็นอภิมานพารามิเตอร์ และ μ_d กับ σ_d^2 เป็นค่าที่คำนวณจากหมู่เล็กที่กำลังฝึก. ตัวแปร $\hat{\mu}_d$ และ $\hat{\sigma}_d^2$ เป็นค่าประมาณของ μ_d และ σ_d^2 ตามลำดับ ซึ่งค่าของ $\hat{\mu}_d$ และ $\hat{\sigma}_d^2$ จะถูกเก็บไว้ใช้ในการอนุมานภายหลัง.

การใช้งานแบบนอร์ม. ขั้นตอน โดยทั่วไป ทำการคำนวณ $h(\mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)})$ เมื่อ h เป็นฟังก์ชันกระตุ้น และ $\mathbf{W}^{(q)}$ กับ $\mathbf{b}^{(q)}$ คือพารามิเตอร์ของชั้น. การทำแบบนอร์ม อาจทำกับค่า $\mathbf{Z}^{(q-1)}$ โดยตรง หรืออาจทำกับ $\mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)}$ ก็ได้. ไอโอพีและเซเจดี[93] แนะนำให้ทำกับตัวกระตุ้น $\mathbf{A}^{(q)} = \mathbf{W}^{(q)} \cdot \mathbf{Z}^{(q-1)} + \mathbf{b}^{(q)}$. การทำแบบนอร์มกับตัวกระตุ้นของชั้น ช่วยปรับค่าตัวกระตุ้นให้อยู่ในย่านที่ฟังก์ชันกระตุ้นทำงานง่ายขึ้นด้วย. นอกจากนั้น เมื่อร่วมผลลัพธ์จาก การคำนวณผลคูณค่าน้ำหนัก กับสมการ 5.23 และ 5.24 แล้วจะเห็นว่า พารามิเตอร์ γ_i และ β_i ช่วยให้อิสระและความหลากหลายในการเลือกปรับค่าความแปรปรวนและค่าเฉลี่ยได้. อีกเรื่องที่ควรกล่าวถึงคือ เมื่อร่วมการคำนวณแบบนอร์มเข้าไปด้วยแล้ว จะเห็นว่า ใบอัส $\mathbf{b}^{(q)}$ ชำรุด เกินความจำเป็น และสามารถตัดออกได้.

กลไกของแบบนอร์ม ช่วยให้การฝึกของโครงข่ายประสาทเทียมทำได้ง่ายขึ้น ไอโอพีและเซเจดี[93] พบว่า แบบนอร์ม อาจช่วยให้การฝึกทำได้เร็วขึ้นถึงสิบสี่เท่า และช่วยให้การกำหนดค่าเริ่มต้นและการเลือกค่าอัตราเรียนรู้ทำได้ง่ายขึ้น (สามารถเลือกค่าได้ช่วงกว้างขึ้น โดยที่ผลลัพธ์ไม่แย่ลงมาก เมื่อเปรียบเทียบกับการเลือกค่าที่ดี).

หมายเหตุ การทำแบบนอร์ม ควรดำเนินการอย่างระมัดระวัง เพื่อไม่ใช้สูญเสียสารสนเทศที่สำคัญไป (ดูแบบฝึกหัด 5.21 ประกอบ). ดังเช่นที่ไอโอพีและเซเจดี[93] ได้แนะนำการประยุกต์ใช้กลไกของแบบนอร์ม กับโครงข่ายโครงข่ายคอนโวลูชันไว้เฉพาะ. โครงข่ายคอนโวลูชัน (บทที่ 6) นิยมใช้กับงานคอมพิวเตอร์วิทัศน์ ซึ่งข้อมูลมีลักษณะเชิงโครงสร้างของพิกเซล. นั่นคือ แต่ละจุดข้อมูลประกอบด้วยค่าพิกเซลหลาย ๆ ค่าที่จัดเรียงกันในโครงสร้าง โดยความสัมพันธ์ของค่าพิกเซลกับตำแหน่งในโครงสร้าง

มีสารสนเทศที่สำคัญอยู่ หากกำหนดให้ อินพุต (ค่าการกระตัน) $\mathbf{X} = [x_{ijc}(n)] \in \mathbb{R}^{H \times W \times C \times N}$ 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 ซึ่งคือภาพถ่ายของจากเหตุการณ์เดียวกัน แต่ใช้อุปกรณ์รับสัญญาณหลายตัว และแต่ละตัวทำงานกับช่วงความถี่สัญญาณคลื่นแม่เหล็กไฟฟ้าต่าง ๆ กัน $C > 1$. ในกรณีที่ไปจำนวนช่องสี อาจมองเป็นจำนวนลักษณะสำคัญ) และการทำแบบนอร์มอาจทำได้ดังสมการ 5.25 และ 5.26 สำหรับ $c = 1, \dots, C$.

$$x'_{ijc}(n) = \frac{x_{ijc}(n) - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \quad (5.25)$$

$$\hat{x}_{ijc}(n) = \gamma_c \cdot x'_{ijc}(n) + \beta_c \quad (5.26)$$

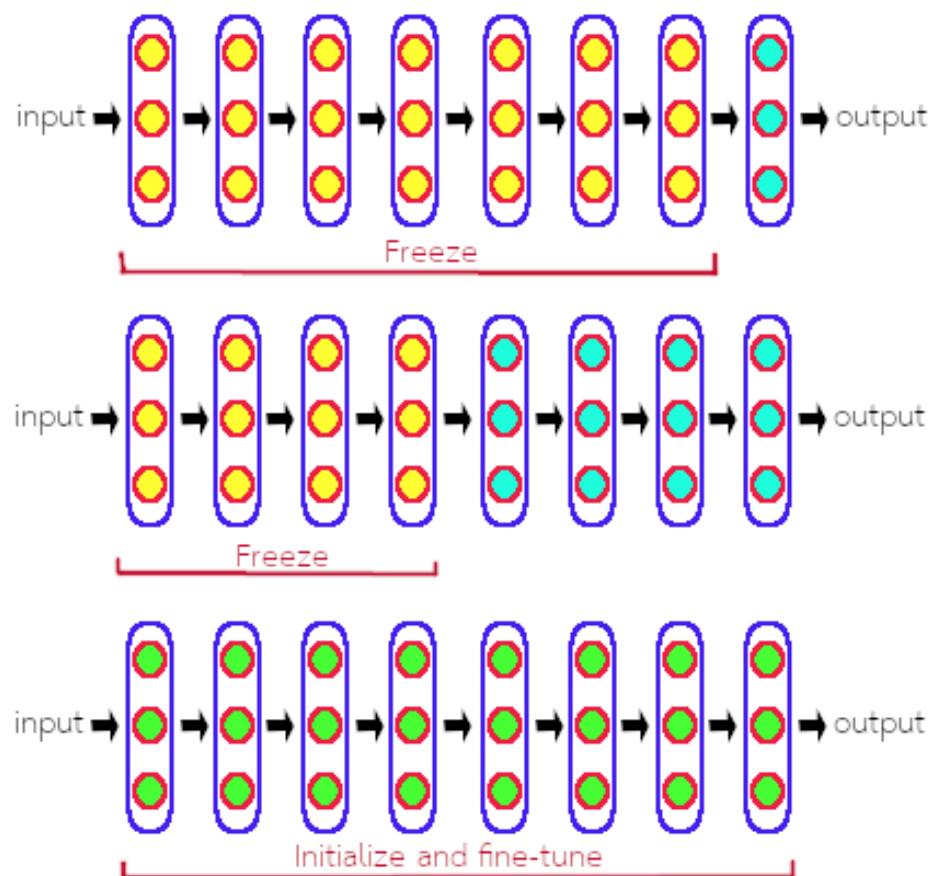
เมื่อ $\mu_c = \frac{1}{|B| \cdot H \cdot W} \sum_{n \in B} \sum_i \sum_j x_{ijc}(n)$ และ $\sigma_c^2 = \frac{1}{|B| \cdot H \cdot W} \sum_{n \in B} \sum_i \sum_j (x_{ijc}(n) - \mu_c)^2$. สังเกต แบบนอร์มสำหรับชั้นคำนวนคอนโวจูชัน จะใช้พารามิเตอร์ γ_c และ β_c หนึ่งคู่ต่อหนึ่งช่องสี.

กลไกช่วยการฝึกอื่น ๆ

กลไกการฝึกก่อน (pre-training) ที่ใช้การฝึกแบบจำลองกับปัญหาที่ง่ายขึ้น หรือปัญหาที่ใกล้เคียงก่อนจะนำค่าน้ำหนักที่ได้จากการฝึก(เบื้องต้น) มาฝึกต่อ (หรือบางครั้งนิยมอ้างถึงว่าเป็น การปรับละเอียด fine tuning) กับปัญหาที่ต้องการจริง ๆ ที่มักเรียกว่า ปัญหาเป้าหมาย.

การฝึกก่อน อาจทำโดยใช้แบบจำลองแบบเดียวกับแบบจำลองสุดท้ายที่ต้องการ แต่ฝึกกับข้อมูลอีกชุด หรือเป้าหมายอีกแบบ หรือ อาจจะฝึกแบบจำลองที่เล็กกว่า และค่อยเพิ่มขยายเป็นแบบจำลองที่ต้องการ เมื่อนำมาใช้กับการกิจเป้าหมาย เช่น ในงานคอมพิวเตอร์วิศวกรรม การทำแบบจำลองอาจจะเลือกแบบจำลองที่นิยมอยู่แล้ว เช่น อเล็กซ์เน็ต (หัวข้อ 6.5) พร้อมการเริ่มต้น ด้วยค่าน้ำหนักของอเล็กซ์เน็ตที่ผ่านการฝึกมาแล้ว แทนที่จะเริ่มจากค่าน้ำหนักสุ่ม. และจึงค่อยดำเนินการฝึกต่อ กับข้อมูลและฟังก์ชันสัญเสียงของภารกิจเป้าหมาย. บางครั้ง เพื่อให้แบบจำลองที่นำมา เหมาะสมกับภารกิจเป้าหมาย อาจมีการปรับแต่งแบบจำลองบาง ได้แก่ เปลี่ยนชั้นคำนวนท้าย ๆ เช่น เปลี่ยนชั้นสุดท้ายให้มีจำนวนเอตพุตสุดท้ายตามที่ต้องการ.

การนำค่าน้ำหนักที่ฝึกแล้วมาใช้ในการฝึกต่อ หากมีข้อมูลของภารกิจเป้าหมายมีปริมาณไม่มาก การดำเนินการ นิยมตรึงค่าน้ำหนักที่ฝึกมาก่อนไว้ (ไม่มีการปรับค่าน้ำหนักเหล่านี้) แต่ปรับค่าน้ำหนักเฉพาะกับชั้นคำนวนหลัง ๆ ซึ่งเชื่อว่าเกี่ยวข้องกับภารกิจเป้าหมายมากกว่า. แต่หากมีข้อมูลของภารกิจเป้าหมายมีปริมาณมาก อาจลดจำนวนชั้นคำนวนต้น ๆ ที่ตรึงค่าน้ำหนักที่ฝึกก่อนลง และเพิ่มจำนวนชั้นคำนวนหลัง ๆ ที่ปรับค่าน้ำหนักให้มากขึ้น หรือ อาจจะเพียงใช้ค่าน้ำหนักที่ฝึกก่อนมาแทนค่าน้ำ



รูปที่ 5.15 แนวทางที่นิยมดำเนินการกับค่าคำน้าหนักจากการฝึกก่อน. ภาพบนสุด แสดงกรณีที่ข้อมูลเป้าหมายค่อนข้างน้อย หรือที่นิยม คือ กำหนดค่าคำน้าหนักของชั้นคำนวนต้น ๆ ด้วยค่าคำน้าหนักจากการฝึกก่อน และปรุงค่าเหล่านี้ไว้ และดำเนินการฝึกโครงข่ายด้วยการปรับค่าคำน้าหนักเฉพาะชั้นคำนวนหลัง ๆ. ภาพกลาง กรณีที่มีข้อมูลเป้าหมายมีมากพอสมควร อาจลดจำนวนชั้นที่ต้องคำน้าหนักลง และฝึกจำนวนชั้นคำนวนมากขึ้น. ภาพล่างสุด แสดงกรณีที่ใช้คำน้าหนักจากการฝึกก่อน เป็นเพียงค่าคำน้าหนักเริ่มต้น และทำการฝึกทั้งโครงข่ายใหม่. การเริ่มต้นฝึกจากค่าคำน้าหนักที่ได้จากการฝึกก่อน จะช่วยให้การฝึกต่อทำได้ง่ายและเร็วขึ้น.

หนักเริ่มต้น แล้วฝึกค่าคำน้าหนักทั้งหมดในโครงข่ายเลย. รูปที่ 5.15 แสดงแนวทางที่นิยมดำเนินการกับค่าคำน้าหนักจากการฝึกก่อน.

ค่าคำน้าหนักของการฝึกก่อน อาจได้มาโดยการเรียนรู้แบบมีผู้สอน ดังที่ได้อธิบายไป หรืออาจได้มาโดยการเรียนรู้แบบไม่มีผู้สอน (*unsupervised pre-training* ดูการศึกษาของเออรานและคณะ[64] เพิ่มเติม). การฝึกก่อน จะช่วยทั้งในแง่ของลดเวลาในการฝึก เพิ่มคุณภาพ รวมถึงคุณสมบัติความทั่วไป และยังมองได้ว่า เป็นความสามารถในการถ่ายโอนการเรียนรู้ (*transfer learning*[227, 120, 146]) อีกด้วย. การถ่ายโอนการเรียนรู้ อ้างถึง สถานการณ์ที่เราสามารถใช้ประโยชน์จากการเรียนรู้ในภารกิจหนึ่ง เพื่อช่วยการเรียนรู้ในอีกภารกิจได้ โดย การเรียนรู้ในภารกิจใหม่ ที่ได้รับการถ่ายโอนการเรียนรู้มา จะทำได้ดีหรือเร็วกว่า การเรียนรู้ในภารกิจใหม่ ที่ไม่มีการถ่ายโอนการเรียนรู้ และต้องเริ่มเรียนทุกอย่างจากศูนย์. ความสามารถในการถ่ายโอนการเรียนรู้ ในโครงข่ายลึก เป็นอีกปัจจัยที่ช่วยให้การประยุกต์

ใช้การเรียนรู้เชิงลึกทำได้ง่ายขึ้น กว้างขวางขึ้น และมีส่วนอย่างมากที่ช่วยเร่งการพัฒนาของศาสตร์อย่างมาก.

แนวทางหรือกลไก ที่อาจมองว่าคล้ายการฝึกก่อน เช่น พิตเน็ต (*FitNets*[168]) ที่ใช้แบบจำลองครู (teacher model) กับแบบจำลองนักเรียน (student model) โดยแบบจำลองครูเป็นแบบจำลองที่ตื้นแต่กว้าง (นั่นคือ มีจำนวนชั้นคำนวนน้อย แต่ว่าแต่ละชั้นมีจำนวนหน่วยคำนวนมาก) ซึ่งฝึกได้ง่ายกว่า. ส่วนแบบจำลองนักเรียนจะลึกแต่แคบ ทำให้มีประสิทธิภาพในการคำนวนมากกว่า แต่ฝึกยากกว่า. กลไกการฝึกแบบครูนักเรียนนี้ คือ ในการฝึกแบบจำลองนักเรียน นอกจากจะฝึกแบบจำลองนักเรียนสำหรับจุดประสงค์หลัก แล้วยังฝึกให้แบบจำลองนักเรียน โดยเฉพาะในชั้นคำนวนต้น ๆ ทำนายค่าผลการกระตุ้นของชั้นคำนวนซ่อนในแบบจำลองครูด้วย. การทำดังนี้ เป็นเสมือน การใช้ค่าผลการกระตุ้นในแบบจำลองครู เป็นตัวช่วยนำทาง สำหรับการฝึกชั้นคำนวนซ่อนของแบบจำลองนักเรียน.

การเรียนหลักสูตร (*curriculum learning*[12]) เป็นอีกแนวทางหนึ่งของกลไกฝึกระดับสูง. กล่าวโดยทั่วไป การเรียนหลักสูตร จะจัดการฝึกเป็นหลาย ๆ ยก โดยเริ่มจากยกแรก ๆ ที่ทำการฝึกที่ง่ายแล้วเพิ่มความยากในการฝึกขึ้นในแต่ละยก จนสุดท้าย คือการฝึกกับปัญหาที่ต้องการ. บนจิโอล([12]) ทดลองการเรียนหลักสูตร ซึ่งทำการฝึกข้อมูลที่ง่ายก่อน แล้วจึงค่อยฝึกข้อมูลที่ยาก เปรียบเทียบกับการฝึกปกติ ที่ใช้ข้อมูลที่ยากเป็นเป้าหมายตั้งแต่แรก. ผลที่ได้พบว่า แบบจำลองที่เรียนตามหลักสูตรสามารถเรียนรู้ได้ดีกว่าอย่างชัดเจน.

นอกจากขั้นตอนวิธีและกลไกต่าง ๆ ที่ช่วยการฝึกแล้ว แนวทางที่ประสบความสำเร็จอย่างมากเลยคือ การออกแบบโครงสร้างแบบจำลอง เพื่อช่วยให้การฝึกทำได้ง่ายขึ้น. จริง ๆ แล้ว การเปลี่ยนฟังก์ชันกระตุ้น ก็เป็นการเปลี่ยนโครงสร้างของแบบจำลอง เพื่อช่วยให้การฝึกทำได้ง่ายขึ้น. กฎเพโลและคณะ[76] ตั้งข้อสังเกตว่า โครงสร้างที่มีลักษณะเชิงเส้นมากขึ้น จะช่วยให้การฝึกทำได้ง่ายขึ้น. กลไกของแบบชนอร์ม ก็มีลักษณะเป็นการเปลี่ยนโครงสร้างของแบบจำลอง. แบบจำลองหลายชนิด ถูกออกแบบมาให้มีเส้นทางการเชื่อมต่อระหว่างชั้นคำนวน โดยอาจมีการเชื่อมต่อข้ามชั้นคำนวนได้ เพื่อช่วยในการฝึก เช่น เรสเน็ต (*ResNet*[85]) ที่มีการเชื่อมต่อข้ามชั้นคำนวน เพื่อช่วยให้การแพร่กระจายของเกรเดียนต์กลับไปหาชั้นคำนวนต้น ๆ ทำได้มีประสิทธิภาพขึ้น. โครงข่ายคอนโวลูชัน (บทที่ 6) ก็เป็นลักษณะของการเปลี่ยนโครงสร้าง ซึ่งโครงสร้างของโครงข่ายคอนโวลูชัน ช่วยลดจำนวนพารามิเตอร์ที่ต้องการลง โดยอาศัยคุณสมบัติที่เหมาะสมกับข้อมูลที่มีลักษณะเชิงท้องถิ่น เช่น ภาพ. การลดจำนวนพารามิเตอร์ที่ไม่จำเป็นลง ช่วยโดยตรงต่อกระบวนการฝึก. แบบจำลองความจำรยะสั้นที่ยาว (บทที่ 8) ก็ถูกออกแบบมา สำหรับข้อมูลเชิงลำดับ เพื่อช่วยให้การฝึก เรียนรู้ ความสัมพันธ์เชิงลำดับระยะยาวทำได้มีประสิทธิภาพมากขึ้น.

5.6 อภิธานศัพท์

การเรียนรู้เชิงลึก (deep learning): การเรียนรู้ของเครื่องที่ใช้โครงข่ายประสาทเทียมจำนวนชั้นคำนวนมาก รวมไปจนถึงเทคนิคและกลไกอื่นๆ ที่เกี่ยวข้อง

ปัญหาการเลื่อนหายของเกรเดียนต์ (vanishing gradient problem): ปัญหา หรือปราชญากรณ์ ที่ขนาดเฉลี่ยของเกรเดียนต์ลดลงอย่างมากที่ชั้นคำนวนต้น ๆ เมื่อเปรียบเทียบกับค่าเฉลี่ยชั้นคำนวนปลาย ๆ

ฟังก์ชัน rectified linear function (relu): หรือ เรลู (relu): ฟังก์ชัน $\text{relu}(a) = \max(a, 0)$

หมู่เล็ก (minibatch): ส่วนของข้อมูลที่ถูกแบ่งเป็นกลุ่มเล็ก ๆ สำหรับการฝึก โดยในหนึ่งสมัยฝึก จะต้องทำการปรับค่าน้ำหนักหลายครั้ง แต่ละครั้งสำหรับแต่ละหมู่เล็ก และการปรับแต่ละครั้ง คำนวนจากจุดข้อมูลต่างๆ ในหมู่เล็ก และจะใช้ค่าเฉลี่ยของเกรเดียนต์ภายในหมู่ในการปรับค่าน้ำหนัก เปรียบเทียบกับการฝึกแบบออนไลน์ ที่การปรับค่าน้ำหนักแต่ละครั้ง คำนวนจากหนึ่งจุดข้อมูล และเปรียบเทียบกับการฝึกแบบอффไลน์ ที่การปรับค่าน้ำหนัก คำนวนจากข้อมูลทั้งหมดที่เดียว และปรับค่าแค่ครั้งเดียว ต่อสมัยฝึก

การตกออก (drop out): กลไกการทำเรกูลารีซ์สำหรับโครงข่ายประสาทเทียม โดยการสุ่มปิดผลการกระตุ้น ของหน่วยคำนวนย่อยต่าง ๆ

แบตช์โนร์ม (batch norm) หรือ แบตช์โนร์มอยไลเซชัน (batch normalization): กลไกช่วยฝึกโครงข่ายประสาทเทียม โดยปรับค่าเฉลี่ยและความแปรปรวนของตัวกระตุ้นในชั้นคำนวน

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.”

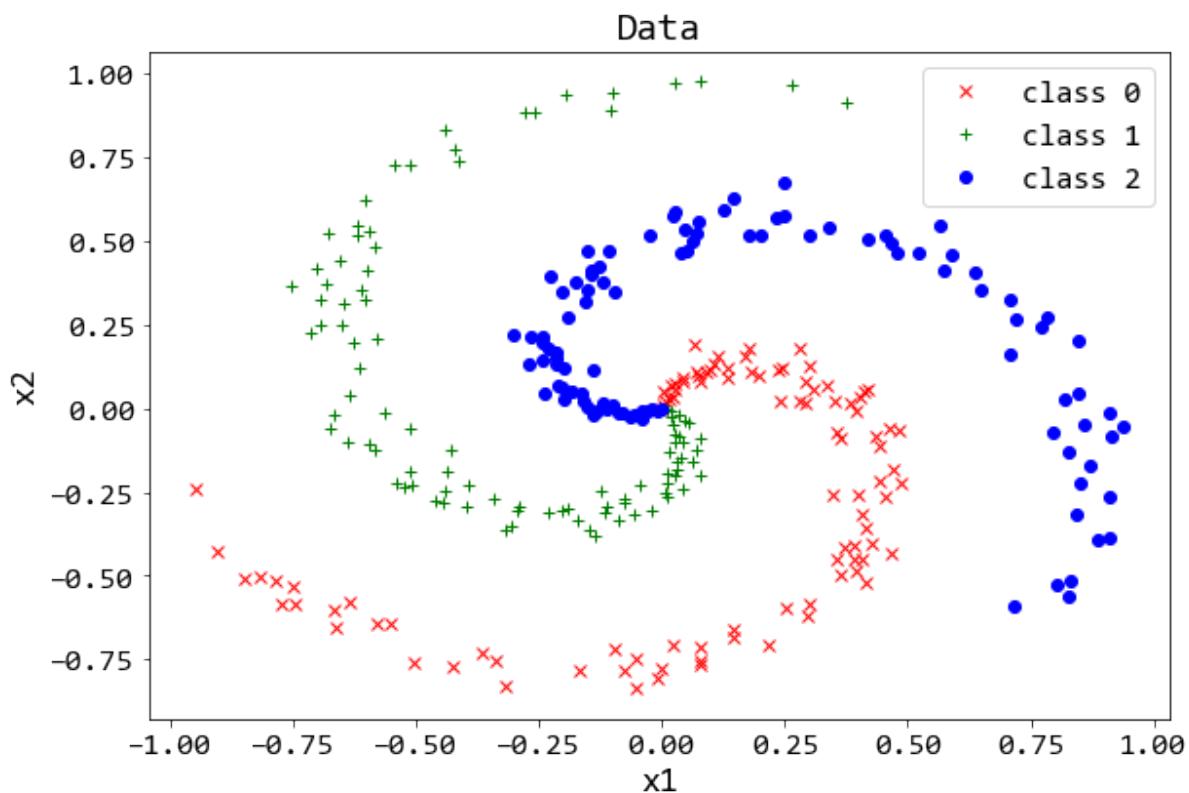
---Marcus Du Sautoy

“สำหรับนักวิทยาศาสตร์ ความท้าทายจริง ๆ ไม่ใช่การพักอยู่ภายในสวนที่ปลอดภัยของสิ่งที่รู้ แต่เป็นการท่องออกไปในป่าของความไม่รู้.”

—มาร์คัส ดู โซวทอย

แบบฝึกหัด 5.1

จงศึกษาตัวอย่างและแสดงปัญหาการเลือนหายของเกรเดียนต์ พร้อมเปรียบเทียบผลลัพธ์จากการบรรเทา โดยเปลี่ยนมาใช้ฟังก์ชันกระตุ้นเรลู.



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 เป็นตัวเลขของกลุ่ม 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. รูปที่ 5.16 แสดงตัวอย่างข้อมูล⁵ ที่เป็นปัญหาการจำแนกกลุ่ม โดยอินพุตมี 2 มิติ และเอาต์พุตเป็นชนิดมี 3 ชนิด ซึ่งสร้างจากตัวอย่างคำสั่งข้างล่าง

```
N = 100
X = np.zeros((2, N*3)) # Initialize dummy input
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 โดยเพิ่มความลึกขึ้นเรื่อย ๆ และสังเกตความยากของการฝึก. ดูหัวข้อ 5.1 ประกอบ. (ตัวอย่างโปรแกรม ศึกษาได้จากหัวข้อ 3.7.)

สุดท้าย (3) ทดลองเปลี่ยนฟังก์ชันกราฟต้นเป็นReLU (ตัวอย่างโปรแกรมการคำนวนReLU แสดงในรายการ 4.2.) สังเกตผล เปรียบเทียบ และอภิปราย.

แบบฝึกหัด 5.2

จากแบบฝึกหัด 5.1 ตั้งสมมติฐานถึงสาเหตุของปัญหาการฝึกโครงข่ายประสาทเทียมลึก ออกแบบการทดลอง เพื่อพิสูจน์และศึกษาสมมติฐานนั้น ดำเนินการทดลอง สังเกตผล วิเคราะห์ สรุป วิจารณ์ และอภิปราย. ศึกษาและทดลองทั้งฟังก์ชันกราฟต้นซิกมอยด์ และReLU พร้อมสังเกตขนาดของกราฟเดียนต์ที่ซั่นต่าง ๆ ขณะฝึก. อภิปรายถึงสาเหตุอื่นที่อาจเป็นไปได้ นอกจากขนาดของกราฟเดียนต์. ดูรูปที่ 5.17 และผลในหัวข้อ 5.1 ประกอบ.

รายการ 5.1 แสดงโปรแกรมโครงข่ายประสาทเทียมที่ปรับปรุงใหม่ โดยเขียนอยู่ในรูปแบบโปรแกรมเชิงวัตถุ และที่เมท็อด `train` มีอาร์กิวเมนต์ `track_grad` ที่สามารถสั่งให้เก็บขนาดของกราฟเดียนต์ไว้เพื่อตรวจสอบภายหลังได้.

ตัวอย่างคำสั่งข้างล่าง ฝึกและทดสอบโครงข่ายสามชั้น (จำนวนหน่วยซ่อนเป็น 4 และ 8 ชั้น ตามลำดับ) สำหรับข้อมูล `datax` และ `y_onehot` ที่อินพุตมีขนาดสองมิติและเอาต์พุตอยู่ในรูปแบบรหัสหนึ่งร้อย สำหรับงานจำแนกกลุ่มที่มีสามกลุ่ม โดยมีจำนวนข้อมูลฝึกเป็น 300 จุดข้อมูล

⁵ตัดแปลง จาก https://cs224d.stanford.edu/notebooks/vanishing_grad_example.html (ข้อมูลเมื่อ 24 พ.ค. 2560).

```

net = w_initn([2, 4, 8, 3])
net['act1'] = sigmoid
net['act2'] = sigmoid
net['act3'] = softmax
ann = ANN(net, NB=300, shuffle='once')
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 คำนวณผลรวมของค่าฟังก์ชันสูญเสียต่อจุดข้อมูลของค่า \hat{y}_i ของ y_i . การกำหนดค่าอัตราการเรียนรู้ $lr=0.3/300$ ให้ผลในการฝึก เช่นเดียวกับค่า α/N ที่ได้แสดงในรายการ 3.18. นั่นคือ $w - (\alpha/N) \cdot \sum_n \nabla E \equiv w - \alpha \cdot \frac{1}{N} \sum_n \nabla E$. แม้ว่าผลจริงไม่ได้แตกต่างกัน แต่การใช้ค่าเฉลี่ย (ในวิธีที่แสดงนี้) ช่วยให้การเลือกอัตราเรียนรู้ทำได้สะดวกขึ้น. ค่าอัตราเรียนรู้สามารถเลือกได้โดยไม่ต้องคำนึงถึงจำนวนจุดข้อมูลฝึก.

หมายเหตุ นอกจากการเขียนในรูปโปรแกรมเชิงวัตถุ และเพิ่ม **track_grad** แล้ว อีกส่วนที่สำคัญ โปรแกรมในรายการ 5.1 ได้เตรียมความสามารถในการฝึกหมู่เล็ก (หัวข้อ 5.2). การฝึกหมู่เล็กไม่ใช่จุดประสงค์ของแบบฝึกหัดนี้ และดังเช่นที่แสดงในตัวอย่างคำสั่งข้างต้น สามารถกำหนดให้ทำการฝึกแบบหมู่ได้โดยการกำหนดจำนวนหมู่เล็ก ให้เท่ากับ(หรือมากกว่า) จำนวนของจุดข้อมูลฝึก ดังเช่น

```
ann = ANN(net, NB=300, shuffle='once')
```

เมื่อ 300 คือจำนวนจุดข้อมูลฝึก.

รายการ 5.1: คลาส สำหรับคำนวณการฝึกและการนำทางของโครงข่ายประสาทเทียม

```

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,
32               epochs=1000, track_grad=False, term=1e-8,
33               term_count_max=5):
34         num_layers = self.net_params['layers']
35         last_layer = num_layers-1
36
37         out_act = 'act%d'%last_layer
38         _, N = trainX.shape
39         A = {}
40         Z = {}
41         delta = {}
42         dEw = {}
43         dEb = {}
44         train_losses = []
45         term_count = 0
46
47         # Minibatch
48         self.prepare_minibatches(N)

```

```

49
50     step_size = lr
51     if track_grad:
52         magGrad = {}
53         for i in range(1, num_layers):
54             magGrad['dEw%d'%i] = []
55             magGrad['dEb%d'%i] = []
56
57     for nt in range(epochs):
58         for ib in range(self.NMB):
59             Z[0],batchY = self.getbatch(ib,trainX,trainY)
60             # (1) Forward pass
61             for i in range(1, num_layers):
62                 b = self.net_params['bias%d'%i]
63                 w = self.net_params['weight%d'%i]
64                 act_f = self.net_params['act%d'%i]
65                 A[i] = np.dot(w, Z[i-1]) + b # A: M x N
66                 Z[i] = act_f(A[i])           # Z: M x N
67             # end forward pass
68             Yp = Z[i]
69
70             # (2) Calculate output dE/da
71             delta[last_layer] = Yp - batchY # M x N
72
73             # (3) Backpropagate: do dE/da for Layer i-1
74             for i in range(last_layer, 1, -1):
75                 b = self.net_params['bias%d'%i]
76                 w = self.net_params['weight%d'%i]
77                 act_f = self.net_params['act%d'%(i-1)]
78
79                 sumdw = np.dot(w.transpose(), delta[i])
80                 if act_f == sigmoid:
81                     delta[i-1] = dsigmoid(Z[i-1])*sumdw
82                 elif act_f == relu:
83                     delta[i-1] = drelu(A[i-1])*sumdw
84                 else:
85                     assert act_f==sigmoid or act_f==relu
86
87             # (4) Calculate gradient dE/dw and dE/db
88             dEw[i]=delta[i].dot(Z[i-1].T)

```

```

89         dEb[i]=delta[i].dot(np.ones((self.NB,1)))
90         if track_grad:
91             magE = np.mean(np.abs(dEw[i]))
92             magB = np.mean(np.abs(dEb[i]))
93             magGrad['dEw%d'%i].append(magE)
94             magGrad['dEb%d'%i].append(magB)
95         # end backpropagate
96
97         # Calculate gradient dE/dw and dE/db
98         dEw[1] = delta[1].dot(Z[0].T)
99         dEb[1] = delta[1].dot(np.ones((self.NB,1)))
100
101        if track_grad:
102            magE = np.mean(np.abs(dEw[1]))
103            magB = np.mean(np.abs(dEb[1]))
104            magGrad['dEw1'].append(magE)
105            magGrad['dEb1'].append(magB)
106
107        # Update parameters w/ Gradient Descent
108        gnorm = 0
109        for i in range(1, num_layers):
110            b = self.net_params['bias%d'%i]
111            w = self.net_params['weight%d'%i]
112            b -= step_size * dEb[i]
113            w -= step_size * dEw[i]
114
115            gnorm += np.linalg.norm(dEb[i])
116            gnorm += np.linalg.norm(dEw[i])
117        # end update parameters
118
119        # Calculate loss at each batch
120        lossn = np.sum(loss(Yp, batchY), axis=0)
121        train_losses.append(np.mean(lossn))
122
123        # Check termination condition
124        if gnorm < term:
125            term_count += 1
126
127            if term_count > term_count_max:
128                print('Term. at %d(%d)'%(nt,ib))

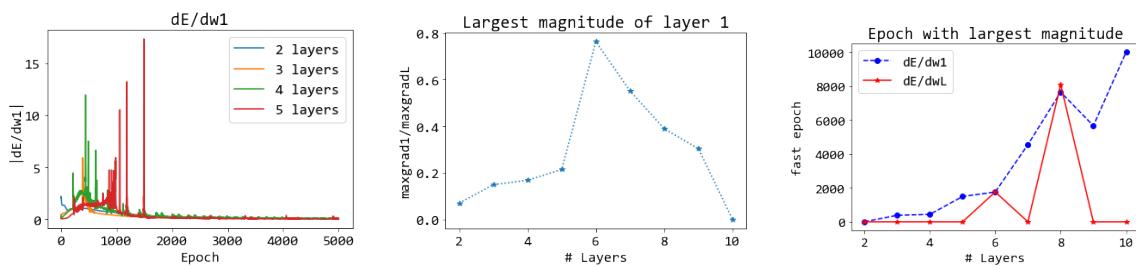
```

```

129         if track_grad:
130             return train_losses, magGrad
131         return train_losses
132     else: # reset term_count
133         term_count = 0
134     # end if term_count
135     # end ib
136 # end epoch nt
137
138     if track_grad:
139         return train_losses, magGrad
140     return train_losses
141
142 def predict(self, X):
143     num_layers = self.net_params['layers']
144     Z = X
145     for i in range(1, num_layers):
146         b = self.net_params['bias%d'%i]
147         w = self.net_params['weight%d'%i]
148         act_f = self.net_params['act%d'%i]
149         A = np.dot(w, Z) + b      # A: M x N
150         Z = act_f(A)            # Z: M x N
151
152     return Z # M x N

```

รูปที่ 5.17 แสดงตัวอย่างการนำเสนอผล. จากผล อาจอภิปราย ได้ดังนี้ (1) ภาพกลางแสดงในเห็นชัดเจนว่า ส่วนใหญ่ขนาดของเกรเดียนต์ในชั้นแรก น้อยกว่าชั้นสุดท้าย และน้อยกว่ามาก ๆ โดยส่วนใหญ่. แต่แนวโน้ม ไม่ได้เป็นไปในทางเดียว นั่นคือ พบรขนาดเกรเดียนต์ในชั้นแรกที่ใหญ่ที่สุด เมื่อใช้ความลึก 6 ชั้น (ซึ่งมีขนาดถึงเกือบ 0.8 หรือเกือบ 80% ของขนาดเกรเดียนต์ชั้นสุดท้าย) และผลลัพธ์แสดงการลดลงในทั้งสองทิศทาง โดยที่ความลึกสิบชั้น ขนาดเกรเดียนต์ในชั้นแรกมีค่าต่ำมาก เมื่อเทียบกับชั้นสุดท้าย. (2) ภาพซ้าย และภาพขวา แสดงสาเหตุในเห็นอีกมุมหนึ่ง คือไม่ใช่แค่ขนาดที่น้อยอย่างเดียว แต่เป็น เมื่อไรที่ชั้นคำนวนตัน ๆ จะได้เกรเดียนต์ขนาดใหญ่. ภาพซ้าย แสดงให้เห็นว่า เกรเดียนต์ชั้นแรกที่มีขนาดใหญ่จะมาซ้ำ ในโครงข่ายที่ลึกขึ้น. ภาพขวา ยืนยันเรื่องที่เกรเดียนต์ขนาดใหญ่มาซ้ำ ในโครงข่ายลึก. สังเกตว่า ชั้นสุดท้าย (เส้น涵สีแดง) จะเห็นเกรเดียนต์ขนาดใหญ่ที่สุด ในสมัยฝึกต้นๆ (เห็นเร็ว) แทนจะทุกระดับความลึก (ยกเว้นความลึก 8). แต่ชั้นแรก (เส้นประสีน้ำเงิน) จะเห็นเกรเดียนต์ขนาดใหญ่ที่สุด ซ้ำแล้วซ้ำอีก ๆ (สมัยฝึกสูง) เมื่อความลึกเพิ่มขึ้นเรื่อยๆ โดยแนวโน้มแบบจะเป็นลำดับทางเดียว (monotonic). การที่เห็นเกรเดียนต์ขนาดใหญ่ซ้ำ อาจหมายถึง การปรับค่าน้ำหนักของชั้นทำได้ซ้ำด้วย ซึ่งตีความได้ว่า การใช้โครงข่ายที่ลึกนั้น ต้องการการฝึกที่ยาวนานขึ้น และการฝึกที่ยาวนานขึ้น โดยทั่วไปแล้ว หมายถึง การฝึกที่ยาก.



รูปที่ 5.17 ปัญหาการเลือนหายของเกรเดียนต์. ภาพซ้าย แสดงขนาดเฉลี่ยของเกรเดียนต์ชั้นที่หนึ่ง ต่อสมัยฝึก ของโครงสร้างประสาทเพิ่มความลึกต่าง ๆ. ภาพกลาง แสดงอัตราส่วนระหว่างขนาดที่ใหญ่ที่สุดจากเกรเดียนต์ชั้นที่หนึ่ง กับขนาดที่ใหญ่ที่สุดจากเกรเดียนต์ชั้นสุดท้าย เมื่อใช้ความลึกต่าง ๆ. ภาพขวา แสดงสมัยฝึกที่เกรเดียนต์มีขนาดใหญ่ที่สุด ของชั้นแรก และชั้นสุดท้าย เมื่อใช้ความลึกต่าง ๆ.

แบบฝึกหัด 5.3

จากแบบฝึกหัด 5.1 และ 5.2 ออกแบบการทดลอง เพื่อวัดผลการแก้ปัญหาการฝึกโครงข่ายลึก และผลการบรรเทาปัญหาการเลือนหายของเกรเดียนต์ เมื่อใช้ฟังก์ชันกระตุ้นเร็ว เปรียบเทียบกับชิกมอยด์ ดำเนินการทดลอง สังเกต วัดผล สรุปและนำเสนอผลให้ชัดเจน ทั้งประเด็นใหญ่ (การฝึกโครงข่ายลึก) และประเด็นย่อย (การเลือนหายของเกรเดียนต์).

แบบฝึกหัด 5.4

จากหัวข้อ 5.2 ออกแบบการทดลอง เพื่อศึกษาผลของขนาดหมู่ลึก ต่อเวลาในการฝึก ความยากง่ายในการฝึก และคุณภาพการฝึก โดยมีปัจจัยประกอบคือ (1) ความลึกของโครงข่ายประสาทเพิ่ม และ (2) จำนวนข้อมูลฝึก. เลือก (หรือสร้าง) ข้อมูลขึ้นมา ดำเนินการทดลอง สังเกตและบันทึกผล สรุป และอภิปราย.

ด้วยข้อมูลที่มีเพิ่มมากขึ้น ชุดข้อมูลที่มีขนาดใหญ่มาก ๆ อาจพบการฝึกแบบหมู่ลึกที่ทำเพียงสมัยเดียว หรือแม้แต่บางครั้งอาจจะไม่สามารถฝึกได้ครบทุกหมู่ลึก (ไม่ครบสมัย และไม่ได้เห็นข้อมูลครบทั้งหมด). สำหรับชุดข้อมูลที่มีขนาดใหญ่มาก ๆ อาจพบปัญหาประสิทธิภาพของการคำนวณ และหากเลือกใช้ข้อมูลเพียงบางส่วน อาจเกิดปัญหาการอันเดอร์พิทได้. อภิปราย ประเด็นการทำงานกับข้อมูลขนาดใหญ่มาก และศึกษาเพิ่มเติมจากบทความวิจัยต่าง ๆ.

ไฟฟอร์ช. โปรแกรมการเรียนรู้เชิงลึก สามารถเขียนด้วยมือໄพได้ แต่เนื่องจากการประยุกต์ใช้ที่เด่น ๆ ของเกี่ยวกับข้อมูลที่มีมิติและจำนวนมหาศาล การคำนวณด้วยจีพียู จะช่วยการทำงานกับข้อมูลเหล่านั้นให้เสร็จได้เร็วขึ้นมาก. หัวข้อนี้ แนะนำมอdule **ไฟฟอร์ช** (PyTorch) ซึ่งเป็นหนึ่งในเครื่องมือที่นิยมใช้กับการเรียนรู้เชิงลึก. มอdule ไฟฟอร์ช ช่วยอำนวยความสะดวก ตั้งแต่การย้ายการคำนวณไปทำที่จีพียู การหาค่าเกรเดียนต์อัตโนมัติ ไปจนถึงโปรแกรมสำเร็จรูปสำหรับกลไกการเรียนรู้เชิงลึกเด่น ๆ ซึ่งจะช่วยให้การใช้งาน และการเรียนรู้การเรียนรู้เชิงลึกทำได้สะดวกมากยิ่งขึ้น.

อย่างไรก็ตาม ถึงแม้ PyTorch จะได้เตรียมโปรแกรมสำหรับต่าง ๆ ไว้ให้ แต่การได้เขียนโปรแกรมจากปฏิบัติการพื้นฐานขึ้นเอง ก็ยังเป็นกระบวนการเรียนรู้ที่สำคัญ ที่ช่วยให้เข้าใจอย่างแท้จริง. ดังนั้น การดำเนินเนื้อหาจะเป็นลักษณะเช่นเดิม นั่นคือ เริ่มจากการเขียนโปรแกรมกลไกต่าง ๆ ขึ้นเอง จากปฏิบัติการพื้นฐาน แล้วค่อยทดลองใช้เครื่องมือสำเร็จที่มี ในลักษณะค่อย ๆ ขยับทีละขั้น เพื่อสร้างทั้งความเข้าใจ ความคุ้นเคย และสำคัญไม่แพ้กันคือ ความมั่นใจ.

การติดตั้ง PyTorch แนะนำให้ศึกษาจากเวป <https://pytorch.org/> โดยหากระบบมีจีพีyu และยังไม่ได้เตรียมการใช้งาน แนะนำให้ติดตั้งและเตรียมการใช้งานจีพีyu ก่อนติดตั้ง PyTorch. หลังติดตั้งเรียบร้อย เช่นเดียวกับการใช้งานโมดูลเพิ่มเติมอื่น ๆ เราต้องนำเข้า 모ดูล PyTorch ก่อน ด้วยคำสั่งเช่น `import torch` เมื่อนำเข้าสมบูรณ์ สามารถทดสอบง่าย ๆ ได้โดยการตรวจสอบเวอร์ชันของ PyTorch เช่น

```
>>> print(torch.__version__)
1.0.0
```

ชิ้น 1.0.0 คือเวอร์ชันที่ใช้⁶ หาก PyTorch ที่ติดตั้งเป็นเวอร์ชันอื่นก็จะได้ค่าอื่นก็ตาม.

รายการ 5.2 แสดงโปรแกรมฟังก์ชันกระตุนเร็ว ซอฟต์แวร์ และครอสแอนโตรปี พร้อมฟังก์ชันกำหนดค่าเริ่มต้น ซึ่งทั้งหมดเปลี่ยนเครื่องมือจากนั้นมาเป็น PyTorch. หมายเหตุ ฟังก์ชันครอสแอนโตรปี ใช้ `eps` เป็นกลไกในการลดปัญหาการคำนวณเชิงเลข. นั่นคือ กรณีที่ค่าที่ทายเป็นศูนย์ สำหรับเฉลยเป็นหนึ่ง (ทายผิดมาก ๆ อาจเกิดตอนเริ่มต้น) จะทำให้เกิด $-\log(0) \rightarrow \infty$. กรณีเช่นนี้ จะทำให้การคำนวณพัง และไม่สามารถคำนวณต่อไปได้. กลไกในการแก้คือใช้ค่าเล็ก ๆ เติมเข้าไป $-\log(0 + \epsilon) \rightarrow v_{\max}$ ซึ่ง v_{\max} คือค่ามากที่สุด (≈ 103) เท่าที่ `-torch.log` จะสามารถคำนวณได้ก่อนจะให้ค่าอภิมาเป็น `inf`. ค่า `1e-45` ที่เลือกใช้ มาจากค่าบวกที่เล็กที่สุด ที่เลขทศนิยมขนาดสามสิบสองบิตจะแทนได้ ซึ่งตัวเลขนี้จะต่างจาก `1e-323` ในรายการ 3.18 ที่สำหรับเลขทศนิยมขนาดหกสิบสี่บิต ซึ่งเป็นข้อมูลดีฟอลต์ของนัมปี. รายการ 5.4 แสดงโปรแกรมคำนวณโครงข่ายประสาทเทียม ด้วย PyTorch. สังเกตว่า การสร้างเทนเซอร์ใหม่ จะมีการกำหนด `device` ด้วย ซึ่ง การกำหนดนี้จะช่วยให้เราสามารถเปลี่ยนการคำนวณระหว่าง จีพีyu และจีพีyuได้สะดวกขึ้น. ดูแบบฝึกหัด 5.6 สำหรับการคำนวณด้วยจีพีyu.

รายการ 5.2: ฟังก์ชันกระตุน เปลี่ยนด้วย PyTorch

```
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
```

⁶ตัวอย่างคำสั่งและโปรแกรมต่าง ๆ ที่จะแสดงนี้ ทดสอบกับ PyTorch เวอร์ชัน 1.0.0.

```

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 ที่เขียนด้วยไฟฟอร์ช. เมื่อเปรียบเทียบโปรแกรมในรายการ 5.4 กับโปรแกรมในรายการ 5.1 จะพบว่า (1) คลาส **tANN1** รับมรดก⁷ มาจากคลาส **ANN** (รายการ 5.1) เพื่อลดความซ้ำซ้อน และ (2) เมท็อด **train** และ **predict** เพียงเปลี่ยนมาใช้คำสั่งของไฟฟอร์ชเท่านั้น⁸. 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 เมื่อ

⁷ การรับมรดก (inheritance) เป็นกลไกการเขียนโปรแกรมเชิงวัตถุ (object-oriented programming) ที่สำคัญ ช่วยให้เราสามารถใช้โปรแกรมเดิมซ้ำได้ โดยเปลี่ยนเฉพาะส่วนที่จำเป็น.

⁸ เพื่อลดความซับซ้อนของโปรแกรม สำหรับซัพชั่นของ คลาส **tANN1** รับฟังก์ชันกระตุ้น **relu** ได้เท่านั้น.

ใช้ไฟฟอร์ช ข้อมูลแทนเซอร์ที่ประมวลผลทุกตัว ต้องอยู่ในรูปแบบเทนเซอร์ของไฟฟอร์ช.

ตัวอย่างคำสั่งต่อไปนี้ ฝึก 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` เป็นเอ้าต์พุตของข้อมูลทดสอบในรูปแบบนัมมไฟ.

ข้อมูลสามารถแปลงไปมาระหว่างรูปแบบของนัมมไฟและไฟฟอร์ช ได้ เช่น คำสั่ง

`ypn = yp.to(torch.device('cpu')).data.numpy()`

แปลง `yp` จากรูปแบบไฟฟอร์ช ออกมานัมมไฟอาร์เรย์. การแปลงจากข้อมูลนัมมไฟอาร์เรย์ ก็สามารถแปลงเป็นไฟฟอร์ช ได้ เช่น

`x = torch.from_numpy(trainx).float().to(dev)`

เป็นการแปลงข้อมูลนัมมไฟอาร์เรย์ `trainx` มาเป็นรูปแบบไฟฟอร์ช.

รายการ 5.4: คลาส สำหรับคำนวณการฝึกและการทำนายของโครงข่ายประสาทเทียม ด้วยไฟฟอร์ช

```

1 class tANN1(ANN):
2     def train(self, trainX, trainY, loss, lr=0.1,
3               epochs=1000, 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     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
32             # (3) Backpropagate. do dE/da for Layer i-1
33             for i in range(last_layer, 1, -1):
34                 b = self.net_params['bias%d'%i]
35                 w = self.net_params['weight%d'%i]
36                 act_f = self.net_params['act%d'%(i-1)]
37                 sumdw = w.transpose(0, 1).mm(delta[i])
38                 if act_f == trelu:
39                     delta[i-1] = tdrelu(A[i-1])*sumdw
40                 else:
41                     assert act_f == trelu
42
43             # (4) Calculate gradient dE/dw and dE/db
44             dEw[i] = delta[i].mm(
45                 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('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. จงออกแบบการทดลองเพื่อทดสอบเปรียบเทียบโปรแกรมทั้งสองแบบ ทั้งในเชิงเวลาในการฝึก เวลาในการอนุमาน คุณภาพการฝึก โดยคำนึงถึงปัจจัยประกอบคือ ความลึกและความซับซ้อนของโครงข่ายประสาทเทียมที่เลือกใช้ และจำนวนจุดข้อมูลกับจำนวนมิติของอินพุต. ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

การคำนวณด้วยจีพีью. จุดประสงค์หลักของการใช้ไฟฟอร์ช คือ การที่ไฟฟอร์ชสามารถส่งการคำนวณไปทำในจีพีьюได้ โดยไม่ต้องยุ่งเกี่ยวกับรายละเอียดปลีกย่อยระดับล่างของการเขียนโปรแกรมขนาดและการเขียนโปรแกรมจีพีyu.

คำสั่ง `torch.cuda.device_count()` ตรวจสอบจำนวนจีพีਯที่สามารถใช้งานได้. คำสั่ง `dev = torch.device('cuda:0')` เตรียมตัวแปรวัตถุ `dev` สำหรับการอ้างถึงอุปกรณ์จีพีyu และเพื่อจะคำนวณด้วยจีพีyu ตัวแปรเห็นเซอร์ทุกตัว จะต้องกำหนดอุปกรณ์เป็นจีพีyu เช่น `x = torch.randn(D,N, device=dev, dtype=torch.float)` เมื่อ D และ N เป็นจำนวนส่วนประกอบในลำดับมิติที่หนึ่งและสองตามลำดับ. หรือแม้แต่การแปลงตัวแปรจากนัมไฟوار์เรย์ เช่น `torchx= torch.from_numpy(datax).float().to(dev)` เมื่อ `datax` เป็นข้อมูลในรูปแบบนัมไฟوار์เรย์ที่ต้องการ. สังเกตว่า นอกจากการกำหนดอุปกรณ์คำนวณแล้ว ชนิดของข้อมูลก็ถูกกำหนดเป็นเลขทศนิยมขนาดสามสิบสองบิต (32-bit floating point number).

แบบฝึกหัด 5.6

คล้ายกับแบบฝึกหัด 5.5 จงออกแบบการทดลองเพื่อทดสอบเปรียบเทียบโปรแกรมในรายการ 5.4 เมื่อทำการคำนวณด้วยจีพีyu เปรียบเทียบกับ เมื่อทำการคำนวณด้วยจีพีyu ทั้งในเชิงเวลาในการฝึก เวลาในการอนุมาน คุณภาพการฝึก โดยคำนึงถึงปัจจัยประกอบคือ ความลึกและความซับซ้อนของโครงข่ายประสาทเทียมที่เลือกใช้ และจำนวนจุดข้อมูลกับจำนวนมิติของอินพุต. ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

หมายเหตุ ดังที่ได้อภิปราย การเปลี่ยนอุปกรณ์คำนวณ สามารถทำได้โดยการระบุอุปกรณ์ที่เห็นเซอร์ทุกตัว 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)
```

เพื่อระบุอุปกรณ์ให้กับเทนเซอร์ของข้อมูลที่จะนำไปคำนวณ.

การหาเกรเดียนต์อัตโนมัติ. นอกจากความสามารถในการเปลี่ยนอุปกรณ์การคำนวณเป็นจีพียูแล้ว ความสามารถที่สุดวากมากอย่างหนึ่งของไฟฟอร์ช คือ การหาค่าเกรเดียนต์ได้โดยอัตโนมัติ (ผ่านกลไกของมอดูลอย่าง `torch.autograd`). นั่นหมายถึง เราไม่จำเป็นต้องคำนวณและเตรียมโปรแกรม เพื่อคำนวณเกรเดียนต์เอง ดังเช่น โปรแกรมที่เขียนสำหรับเมท็อด `train` ในรายการ 5.4.

การหาเกรเดียนต์อัตโนมัติด้วยไฟฟอร์ช (1) จะต้องระบุในตัวแปรที่ต้องการคำนวณเกรเดียนต์ โดยกำหนด `requires_grad` ของเทนเซอร์ให้ค่าเป็น `True` ตัวอย่างเช่น หากต้องการคำนวณ เกรเดียนต์ $\nabla_w E$ ซึ่งเป็นเกรเดียนต์ของค่า E ต่อตัวแปร w อาจจะระบุที่ตัวแปร w โดยตรงด้วย `w.requires_grad = True` หรืออาจจะระบุไปพร้อมกับการทำหนดค่าเริ่มต้น ด้วย

```
w = torch.randn(M, D, requires_grad=True)
```

ก็ได้. การกำหนด `requires_grad` เป็น `True` จะบอกให้ไฟฟอร์ชติดตามการคำนวณที่เกี่ยว- ข้องกับตัวแปร เพื่อนำมาคำนวณหากค่าเกรเดียนต์ได้ถูกต้อง.

จากนั้นหลังการคำนวณค่าเป้าหมาย E เสร็จสิ้น (2) ต้องระบุให้ไฟฟอร์ชคำนวณเกรเดียนต์ ด้วย คำสั่ง เช่น `E.backward()` เมื่อ E เป็นตัวแปรเทนเซอร์แทนค่าเป้าหมาย E . ค่าเกรเดียนต์ $\nabla_w E$ ที่คำนวณได้ จะเก็บไว้ที่ลักษณะประจำ (attribute) `grad` ของตัวแปร เช่น ในตัวอย่างนี้ คือ `w.grad`. แต่การปรับค่าพารามิเตอร์ ต้องทำการลบออกจากผลของการคำนวณเกรเดียนต์อัตโนมัติ และหลังการปรับค่า ต้องล้างค่าเกรเดียนต์ออกสำหรับการคำนวณครั้งต่อไป. ตัวอย่างเช่น เมื่อต้องการ ปรับค่าพารามิเตอร์ อาจทำโดย

```
with torch.no_grad():
    w -= learning_rate * w.grad
    w.grad.zero_()
```

เมื่อ `learning_rate` เป็นค่าอัตราการเรียนรู้.

แบบฝึกหัด 5.7 แสดงตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่เขียนโดยใช้การหาเกรเดียนต์ อัตโนมัติ และการเรียกใช้.

แบบฝึกหัด 5.7

รายการ 5.5 แสดงโปรแกรมโครงข่ายประสาทเทียมที่เขียนด้วยไฟฟอร์ชและใช้การหาเกรเดียนต์ อัตโนมัติ โดยเพื่อลดความซ้ำซ้อน คลาส `tANN2` รับมรดก จากคลาส `tANN1` (รายการ 5.4).

นอกจาก คลาส **tANN2** สังเกตว่า ฟังก์ชันต่าง ๆ ในสิ่นทางของการแพร่กระจายอยู่กลับ ต้องถูก เขียนใหม่ และการเขียนเมื่อตัด **backward** ต้องเขียนการคำนวณอนุพันธ์ย้อน เช่น $\frac{\partial E}{\partial a}$ ซึ่ง $\frac{\partial E}{\partial a} = \frac{\partial z}{\partial a} \cdot \frac{\partial E}{\partial z} = h'(a) \cdot \frac{\partial E}{\partial z}$. กลไกของการหากรเดียนต์อัตโนมัติ จะคำนวณส่วน $\frac{\partial E}{\partial z}$ มาให้. (เปรียบเทียบกับ **drelu** ในรายการ 5.2 ซึ่งคำนวณ $h'(a)$. ดูสมการ 3.31 ประกอบ.) ตัวอย่างนี้ แสดง ฟังก์ชันrelu และฟังก์ชันเอกลักษณ์ ฟังก์ชันอื่น ๆ ก็สามารถทำได้ในลักษณะเดียวกัน.

ดังที่ได้อธิบาย ตัวแปรที่ต้องการคำนวณกรเดียนต์ต้องถูกระบุอย่างชัดเจน ซึ่งดำเนินการใน โปรแกรม **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** เป็นอินพุตและเอาต์พุตของข้อมูลทดสอบในรูปแบบนัมป์ไฟ.

รายการ 5.5: คลาสโครงข่ายประสาทเทียม สำหรับการหาค่าลดถอย ที่เขียนด้วยไฟฟอร์ชและการหากรเดียนต์อัตโนมัติ.

```
1 class tANN2(tANN1):
2     def train(self, trainX, trainY, lossf, lr=0.1,
3               epochs=1000, 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]
41                 w = self.net_params['weight%d'%i]
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('Term. at %d(%d)'%(nt,ib))
59             return train_losses
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
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 `lse`. ดังนั้นงานการหาค่าลดลง สามารถทำได้ทันที แต่งานอื่น ๆ เช่น การจำแนกค่าทวิภาค (ต้องการฟังก์ชันซิกมอยด์และครอสเอนโตรปีสำหรับสองค่า) หรือการจำแนกกลุ่ม (ต้องการฟังก์ชันซอฟต์แมกซ์และครอสเอนโตรปี) ซึ่งสามารถทำได้เช่นเดียวกัน แต่ต้องเตรียมฟังก์ชันที่เกี่ยวข้องให้พร้อมก่อน.

มอดูลย่อย nn. การหาเกรเดียนต์อัตโนมัติ ช่วยลดภาระทั้งการวิเคราะห์เกรเดียนต์ และการเขียนโปรแกรมลงไปมาก. แม้จะลดภาระลงไปมาก แต่การโปรแกรมโครงข่ายประสาทเทียม จากปฏิบัติ การพื้นฐาน (ดังเช่นที่ทำตัวอย่างในรายการ 5.5) ถือเป็นการเขียนโปรแกรมในระดับล่าง ซึ่งเป็นภาระ เชิงปัญญา (*cognitive burden*). เพื่อช่วยลดภาระนี้ รวมถึงช่วยในแง่ของลำดับชั้นของความคิด⁹ (*hierarchy of abstraction*) การประยุกต์ใช้งานโครงข่ายประสาทเทียมเชิงลึก จะทำได้เมื่อประสิทธิภาพ กว่า เมื่อใช้มอดูลสำเร็จ เช่น มอดูล `nn`. มอดูล `nn` มีโครงสร้างและฟังก์ชันสำเร็จต่าง ๆ สำหรับกลไก ที่มีการใช้อย่างแพร่หลาย. ตัวอย่างคำสั่ง กำหนดโครงข่ายด้วยไฟฟอร์ช `nn` แสดงในรายการ 5.6 โดย ตัวอย่างคำสั่ง สำหรับการฝึกและทดสอบ แสดงในรายการ 5.7. หมายเหตุ ในรายการ 5.7 โปรแกรม ทำ `model.zero_grad()` ในช่วงปลายสมัย (หลังจากทำอย่างอื่นเสร็จ) เพื่อให้เปรียบเทียบได้

⁹ ลำดับชั้นของความคิด เป็นแนวคิดทางวิศวกรรมคอมพิวเตอร์ และวิศวกรรมทั่วไป (และจริง ๆ แล้วก็รวมถึงกิจกรรมต่าง ๆ ไปจนถึงการยกระดมของมนุษยชาติ) ที่จะมองหรือแก้ปัญหาในหลาย ๆ ระดับของรายละเอียด โดยการคิดในระดับหนึ่ง จะลดการพิจารณา รายละเอียดของระดับล่างกว่า ที่เกินความจำเป็นลง. การละรายละเอียดระดับล่างออก ช่วยลดภาระเชิงปัญญา ทำให้สามารถมอง ปัญหาไปได้ไกลขึ้น กว้างขึ้น และเป็นองค์รวมมากขึ้น.

ตรงมาตรงไปกับการฝึกที่แสดงในรายการ 5.5. อย่างไรก็ตาม ความนิยม คือทำการล้างค่าเกรเดียนต์ ช่วงต้นสมัยฝึก (หาก่อนที่จะทำอย่างอื่น).

รายการ 5.6: ตัวอย่างการใช้เพทอร์ช `nn` เพื่อกำหนดโครงข่ายประสาทเทียมสำหรับการจำแนกกลุ่ม โดยอินพุตมีสองมิติ และเอาต์พุตมีสามมิติ ชั้นช่อนมี 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 โดยใช้ฟังก์ชันสูญเสียเป็นครอส เอนโนทรีป. ข้อมูลฝึก อินพุต `tdatax` เป็นไฟฟอร์ชแทนเชอร์สัดส่วน $D \times N$ เมื่อ D เป็นจำนวนมิติ และ N เป็นจำนวนข้อมูล. ฉลากเฉลยของเอาต์พุต `tdatay` เป็นไฟฟอร์ชแทนเชอร์สัดส่วน $1 \times N$. เนื่องจาก โมดูล `nn` รับค่าอินพุตในสัดส่วน $N \times D$ และ `torch.nn.NLLLoss` รับฉลากเฉลยของเอาต์พุต ในรูปแบบจำนวนเต็ม ในสัดส่วน N ดังนั้น คำสั่งตัวอย่างจึงใช้ `tdatax.transpose(0,1)` และ `tdatay[0].long()` สำหรับการป้อนข้อมูลฝึก. ข้อมูลทดสอบ อินพุต `ttestx` เป็นไฟฟอร์ชแทนเชอร์สัดส่วน $D \times N'$ เมื่อ N' คือจำนวนข้อมูลทดสอบ. ส่วนฉลากเฉลยของข้อมูลทดสอบ `ttesty` เป็นไฟฟอร์ชแทนเชอร์สัดส่วน $1 \times N'$. 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())
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 คำนวนซอฟต์แมกซ์ด้วย `torch.nn.Softmax(dim = 1)` และคำนวนครอสเอนไทร์ด้วย `loss = loss_fn(torch.log(yhat), tdatay[0].long())` โดย `loss_fn = torch.nn.NLLLoss()`. การจัดการคำนวนเช่นนี้ เพื่อให้โปรแกรมในรายการ 5.6 สามารถเปรียบเทียบกับโปรแกรมที่เขียนจากปฏิบัติการพื้นฐานได้สะดวกขึ้น. แต่ในทางปฏิบัติ การคำนวนจะมีประสิทธิภาพมากกว่า โดยการใช้ `nn.LogSoftmax` คู่กับ `nn.NLLLoss` หรือสะดวกที่สุด โดยใช้ `nn.CrossEntropyLoss` ซึ่งคำนวนทั้งซอฟต์แมกซ์และครอสเอนไทร์รวมกันเลย.

หมายเหตุ โดยเดิมอยู่ `nn.NLLLoss` และ `nn.CrossEntropyLoss` คำนวนค่าสูญเสียเฉลี่ยต่อของหมู่เล็กๆ ของค่า (เดิมอยู่ `reduction='mean'`. ดูรายละเอียดการทำงานของแต่ละฟังก์ชันได้จาก <https://pytorch.org/docs/stable/nn.html>.)

ในเชิงตรรกะการทำงานแล้ว การใช้ผลรวมหรือค่าเฉลี่ย ต่างกันเพียงค่าคงที่ที่นำไปหารค่าฟังก์ชันสูญเสียเท่านั้น. แต่ในทางปฏิบัติ ความต่างนี้มีผลโดยตรง คือ (1) หากเขียนโปรแกรมเอง ผลรวม อาจทำได้อย่างมีประสิทธิภาพมาก ผ่านการจัดการคุณเมทริกซ์ แต่ค่าเฉลี่ยต้องเพิ่มการหารเข้ามา ซึ่งการหารนี้ อาจทำได้อย่างมีประสิทธิภาพมาก โดยทำที่อัตราการเรียนรู้. (2) ไม่ว่าจะเขียนโปรแกรมเอง หรือใช้โปรแกรมสำเร็จ การใช้ค่าเฉลี่ย จะให้ผลคำนวนที่ค่อนข้างคงที่ เมื่อเทียบกับจำนวนข้อมูล. นั่นคือ หากใช้ผลรวม เมื่อจำนวนข้อมูลมาก ค่าสูญเสียที่เห็น (ซึ่งคือผลรวมค่าสูญเสีย) จะมีตัวเลขใหญ่. ดังนั้น เมื่อเพิ่มจำนวนข้อมูลผีกเข้าไป ค่าสูญเสียจะเพิ่มขึ้น จนมีค่ามากขึ้น เมื่อเปรียบเทียบกับการผีกด้วยข้อมูลน้อย ๆ (ซึ่งไม่ได้แปลว่า การผีกແย่อง). แต่หากใช้ค่าเฉลี่ย ค่าสูญเสียที่เห็น (ซึ่งคือค่าสูญเสียเฉลี่ย) จะมีตัวเลขที่อยู่ในระดับเดียวกันได้ ไม่ว่าจะใช้จำนวนจุดข้อมูลผีกเท่าไร. (3) การเลือกค่าอัตราเรียนรู้ จะทำได้สะดวกกว่าในกรณีค่าเฉลี่ย. นั่นคือ หากพบค่าอัตราเรียนรู้ที่ใช้ได้กับชุดข้อมูล เมื่อมีจำนวนข้อมูลน้อย ๆ แล้วถ้ามีจำนวนข้อมูลเพิ่มขึ้นมาก การใช้ค่าอัตราเรียนรู้เดิม โดยทั่วไป ก็จะสามารถใช้ได้ดี. แต่หากใช้ผลรวม เมื่อจำนวนข้อมูลเพิ่มขึ้น จะทำให้ผลรวมค่าสูญเสียและผลรวมเกรเดียนต์มากขึ้นโดยธรรมชาติ เพราะมีพจน์ที่จะรวมมากขึ้น. ดังนั้น ค่าอัตราเรียนรู้เดิม อาจจะใช้ได้ไม่ดี และอาจจะต้องปรับลดลงเป็นอัตราส่วนตามจำนวนข้อมูลที่เพิ่มขึ้น. การรู้ระลึกถึงประเด็นนี้ ผลกระทบหรือค่าเฉลี่ยนี้ จะช่วยให้การเลือกค่าอัตราเรียนรู้ และการอ่านผลความก้าวหน้าการผีก ทำได้ดียิ่งขึ้น.

การใช้ `nn.Sequential` แม้จะสะดวก แต่หากต้องการกำหนดทอโพโลยี (*topology* การเชื่อมต่อ) ที่อิสระ ยืดหยุ่น และหลากหลายมากขึ้น การใช้ `nn.Module` (ดังแสดงในรายการ 5.8) อาจจะเหมาะสมกว่า. ตัวอย่างทอโพโลยีที่เกินกว่า `nn.Sequential` จะสามารถบรรยายได้ มีมากมาย รวมถึง อเล็กซ์เน็ต[113] (หัวข้อ 6.5).

การบันทึกแบบจำลองที่ผีกแล้วก็สามารถทำได้ เช่น

```
torch.save(net.state_dict(), './sav/nn1.pth')
```

เมื่อ `net` เป็นแบบจำลองที่ต้องการบันทึกค่าเก็บไว้ และ '`.sav/nn1.pth`' เป็นสิ่งทางและชื่อไฟล์ที่บันทึก. การเรียกใช้แบบจำลองที่บันทึกไว้สามารถทำได้ เช่น

```
net = Net().to(device)
net.load_state_dict(torch.load('.sav/nn1.pth'))
```

เมื่อ `Net()` เป็นโครงสร้างของแบบจำลอง. สังเกตว่า การบักทึกค่า จะบันทึกเฉพาะค่าของพารามิเตอร์ ดังนั้น การเรียกใช้แบบจำลองจึงประกอบด้วยการสร้างตัวแปรวัตถุของแบบจำลองขึ้นมาใหม่ และกำหนดค่าของพารามิเตอร์ตามค่าที่บันทึกไว้.

รายการ 5.8: ตัวอย่าง แสดงโครงข่ายสามชั้น แบบเดียวกับโปรแกรมในรายการ 5.6 แต่ใช้การกำหนดท่อพอโลยีอง (เมท็อด `forward`) โดยคลาสรับมรดกจาก `nn.Module` ซึ่งยืดหยุ่นกว่าท่อพอโลยีของ `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))
12        return z3
```

แบบฝึกหัด 5.8

จงทดสอบโปรแกรมโครงข่ายประสาทเทียม ที่เขียนโดยใช้มอดูล `nn` เปรียบเทียบกับโปรแกรมที่เขียนการคำนวนเกรเดียนต์เอง (เช่น โปรแกรมในรายการ 5.4) โดย ออกรูปแบบการทดลอง เลือกหรือสร้างข้อมูล ดำเนินการทดลอง สังเกต บันทึกผล สรุปและอภิปราย.

มอดูลช่วยจัดหมุ่ยอย `utils.data.DataLoader`. การทำการฝึกหุ่นเล็ก (ดังเช่น โปรแกรมในรายการ 5.1) ก็สามารถดำเนินการได้ด้วยมอดูลย่อย `utils.data.DataLoader`. การใช้งาน จะต้องสร้างตัวแปรวัตถุของ `DataLoader` โดยการสร้างตัวแปรนี้ ต้องกำหนดข้อมูลที่ต้องการเข้าไป และข้อมูลนี้ต้องอยู่ในรูปแบบของ `utils.data.Dataset` ที่ตัวอย่าง คำสั่งข้างล่างใช้คลาส `MyDataset` (รายการ 5.9) เข้ามาช่วย.

```
mydat = MyDataset()
mydat.assign_data(DX, DY)
```

```
datloader = torch.utils.data.DataLoader(mydat, batch_size=50,
                                         shuffle=True, num_workers=0)
```

เมื่อกำหนดขนาดหมู่เล็กเป็น 50.

ตัวแปร \mathbf{DX} และ \mathbf{DY} เป็นข้อมูลอินพุตและเอาต์พุต ชนิดไฟทอร์ชเทนเซอร์ สัดส่วน $N \times D_x$ และ $N \times D_y$ ตามลำดับ โดย N เป็นจำนวนจุดข้อมูล และ D_x กับ D_y เป็นมิติของอินพุตและเอาต์พุต.

การเรียกใช้ **trainloader** ก็สามารถทำได้ เช่นเดียวกับตัวแปรวนช้ำ¹⁰ อื่น ๆ ของไฟรอน เช่น

```
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()
```

เมื่อ **num_epochs** และ **learn_rate** เป็นจำนวนสมัยฝึกและอัตราการเรียนรู้ ตามลำดับ. ในตัวอย่างคำสั่งนี้ โปรแกรม **loss_fn** รับเฉลยในรูปแบบไฟทอร์ชเทนเซอร์ หนึ่งลำดับชั้น¹¹ ดังนั้น คำสั่ง **loss = loss_fn(yhat, labels[:,0])** จะต้องจัดผลลัภเฉลยให้อยู่ในรูปแบบดังกล่าว.

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
```

¹⁰ ตัวแปรวนช้ำ (iterable variable) หมายถึง ตัวแปรวัตถุ ที่สามารถให้ค่าสมาชิกของมันออกมากได้ทีละตัว ซึ่งสามารถใช้งานได้สะดวกกับการวนช้ำด้วยคำสั่ง **for**. ไฟรอน มีข้อมูลหลายชนิดที่เป็นตัวแปรวนช้ำ เช่น ลิสต์ ทูเพิล และดิกชันนารี.

¹¹ นั่นคือ เมื่อตรวจสอบสัดส่วน เช่น รันคำสั่ง **labels[:,0].shape** และจะเห็นสัดส่วนเป็น **torch.Size([50])** ไม่ใช่ **torch.Size([50, 1])** ที่หมายถึง ไฟทอร์ชเทนเซอร์ สองลำดับชั้น.

```

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 จงเขียนโปรแกรมโดยใช้มอดูล `nn` และการหากรเดินต์อัตโนมัติ พร้อมด้วยจัดการข้อมูลฝึกด้วย `utils.data.DataLoader` และเปรียบเทียบผล กับผลลัพธ์จากแบบฝึกหัด 3.16 สรุป และอภิปราย.

แบบฝึกหัด 5.10

จากแบบฝึกหัด 5.9 ที่เราดาวน์โหลดข้อมูลเอง เตรียมข้อมูลเอง จัดรูปแบบต่าง ๆ จนข้อมูลสามารถนำเข้าไปใช้กับ ตัวแปรวัตถุของ `DataLoader` ได้. อาย่า่รีเก็ตตาม พัฒนาการของการเรียนรู้ของเครื่องและการรู้จำรูปแบบ ก้าวหน้าไปมาก และมีชุดข้อมูลที่มีการศึกษาอย่างกว้างขวาง และนิยมใช้เพื่อเรียนรู้ หรือเพื่อการทดสอบกลไกใหม่ ๆ. สำหรับชุดข้อมูลที่นิยมหลาย ๆ ชุด ในทอร์ชมีกลไกช่วยเหลือ ด้วยมอดูล `torchvision` เพื่อลดภาระในการเตรียมข้อมูลเหล่านี้ลง. ตัวอย่างคำสั่งข้างล่าง เตรียมชุดข้อมูลเอมนิสต์ตั้งแต่ดาวน์โหลด (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 โดยใช้ชุดข้อมูลโหลดสำเร็จ. สังเกตผล สรุป และอภิปราย. หมายเหตุ การวัดค่าความแม่นยำของข้อมูลทดสอบที่แบ่งเป็นหมู่เล็ก จะช่วยลดภาระการใช้หน่วยความจำที่เดียวมาก ๆ ได้.

มอดูลย่อย optim. ดังที่อภิปรายในหัวข้อ 5.5 มีขั้นตอนวิธีมากมาย ที่สามารถนำมาฝึกแบบจำลองได้. มอดูล **optim** จัดเตรียมขั้นตอนวิธีที่นิยมต่าง ๆ ไว้ให้. โดยตัวอย่างคำสั่งต่อไปนี้ แสดงการใช้งาน วิธีลงเกรเดียนต์¹² ที่ใช้อัตราเรียนรู้เป็น 0.001 และโมเม้นตัมเป็น 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)
```

และการฝึกสามารถทำได้ดังแสดงในรายการ 5.10. สังเกต โปรแกรมในรายการ 5.10 ล้างค่าเกรเดียนต์ **net.zero_grad()** ตั้งแต่ต้นของลูป.

รายการ 5.10: ตัวอย่างการฝึกแบบจำลอง ด้วยมอดูล **optim** โดย **nepochs** เป็นจำนวนสมัยฝึก. อินพุต **inputs** และเอาต์พุต **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

¹²โปรแกรม **optim.SGD** หมายถึง วิธีลงเกรเดียนต์แบบสโตกัสติก (*stochastic gradient descent method*) ซึ่งมีกลไกหลัก คือ วิธีลงเกรเดียนต์ และเน้นการสุ่มลำดับข้อมูล (ดูหัวข้อ 5.2) นอกจากนั้น **optim.SGD** ยังมีกลไกโมเม้นตัมด้วย (หัวข้อ 5.5).

จะเลือกหรือสร้างข้อมูล เลือกแบบจำลอง ฝึกโดยใช้การหาค่าดีที่สุดจากมอดูล **optim** ทดสอบ สรุปและอภิราย.

แบบฝึกหัด 5.12

จากหัวข้อ 5.3 จงศึกษาและเขียนโปรแกรมสำหรับกลไกการตกลอก จงเลือกหรือสร้างข้อมูล ทำแบบจำลอง โดยใช้เทคนิคการตกลอกที่เขียนขึ้น ทดสอบ สังเกตผล สรุป และอภิราย. หมายเหตุ การใช้การตกลอก อาจทำให้ต้องการแบบจำลองที่ใหญ่ขึ้น(ชับซ้อนขึ้น) จากการทำแบบจำลองที่ไม่ใช้การตกลอก รวมถึงอาจทำให้ต้องการจำนวนสมัยฝึกที่มากขึ้น. คำใบ้ การฝึกอาจทำได้ช้าลง แต่ให้สังเกต การลดลงของค่าฟังก์ชันสูญเสีย และในการทดสอบ อย่าลืมเชค เผื่องการตกลอก. แบบฝึกหัดนี้ ต้องการให้ได้ทดลองฝึกเขียนโปรแกรมด้วยตนเอง 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 ถึงแม้ชื่อคือ การตกลอก แต่ค่าความน่าจะเป็น (ตัวแปร **oneprob** ในรายการ 5.11 ซึ่งจะรับค่า 0.8 และ 0.5 ในรายการ 5.12) จะถูกความน่าจะเป็นของการคงอยู่. ข้อควรระวัง การใช้งานมอดูลสำเร็จ ควรศึกษาตรวจสอบพฤติกรรมการทำงานให้ชัดเจนก่อน.

รายการ 5.11: ตัวอย่างโปรแกรมการตกลอก. 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 ค่าหน่วยย่อหยssp ทำการตกลอก และก่อนทำการตกลอก ตามลำดับ. ค่า $\frac{\partial z}{\partial z'} = m$ เมื่อ m คือ หน้ากาก หรือค่าสัมประสิทธิ์ของการตกลอก ($m \in \{0, 1\}$ และ m มีการแจกแจงแบบเบอร์นูลลี่ และความน่าจะเป็นของค่าหนึ่ง แทนด้วย อาร์กิวเม้นต์ **onprob**).

```

1 class mdropout(torch.autograd.Function):
2     @staticmethod
3     def forward(ctx, z, onprob=0.5):
4         d = torch.distributions.Bernoulli(
5             torch.tensor([onprob]))
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
```

รายการ 5.12: ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้การตัดออกที่เขียนขึ้นเอง. สังเกต (1) การตัดออก ทำเฉพาะ ตอนฝึก (2) การอนุมาน ต้องซัดเฉยการตัดออก. โปรแกรม ใช้กลไกของ `nn.Module` ที่มีสถานะ `self.training` ใน การตรวจสอบว่า กำลังฝึก หรือกำลังใช้งานอนุมาน.

```
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 = 0.8 * x
18            z1 = torch.relu(self.fc1(xm))
19            z1m = 0.5 * z1
20            z2 = self.fc2(z1m)
21
22        return z2
```

รายการ 5.13: ตัวอย่างการฝึกและทดสอบโครงข่ายประสาทเทียมที่ใช้การตัดออกที่เขียนขึ้นเอง. สังเกต เพื่อรับว่า การคำนวณเป็นการฝึก 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 การตอกออก อาจดำเนินการชดเชย โดยใช้การหารค่าความน่าจะเป็น ออกจากค่าหน่วยอย่าง -tonfik แทนการคูณเข้า ขณะทำการอนุมาน. ตัวอย่างโปรแกรมในรายการ 5.14 แสดงโปรแกรมการตอกออก ที่เขียนโดยใช้การหาร ตอนฟิก เพื่อชดเชยค่าที่ตอกออกไป. งบเปรียบเทียบ ความต่างกับโปรแกรมในรายการ 5.11 ทั้งวิธีการเขียน และพฤติกรรมการทำงาน.

รายการ 5.14: ตัวอย่างโปรแกรมการตอกออก โดยการหารค่าความน่าจะเป็น ขณะฟิก เพื่อชดเชยการตอกออก. หมายเหตุ เพื่อให้แนวคิดตระการการทำงานชัดเจน โปรแกรมในตัวอย่างใช้การหาร. ในทางปฏิบัติ การคูณด้วย 1.25 และ 2 จะให้ ประสิทธิภาพและเสถียรภาพดีกว่า 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 = 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) / 0.8
13          z1 = torch.relu(self.fc1(xm))
14          z1m = self.do1(z1, 0.5, 1) / 0.5
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

```

การตัดออก ด้วย **`nn.Dropout`**. โมดูล `nn` มีมอดูลอย่างสำหรับทำการตัดออก คือ `nn.Dropout`. ตัวอย่างคำสั่งข้างล่าง แสดงการใช้ `nn.Dropout` เพื่อใช้งานกลไกการตัดออก ในลักษณะเดียวกับ โปรแกรมในรายการ 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 (เช่นที่ ต้องทำในรายการ 5.12) เพราะว่า `nn.Dropout` มีกลไกภายในที่จัดการเรื่องนี้ให้.

นอกจากนั้น `nn.Dropout` รับความน่าจะเป็นที่จะตัดออก (เปรียบเทียบกับ โปรแกรมใน รายการ 5.11 ที่เป็นความน่าจะเป็นของกรองอยู่) 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` เปรียบเทียบโปรแกรมการทำงาน และผลการทำงาน สรุปผล และอภิราย.

แบบฝึกหัด 5.16

จงศึกษาการทำงานและผลของการใช้การตัดออก โดยเปรียบเทียบกับ (1) การไม่ใช้เทคนิคการตัดออก และ (2) การทำค่าหน้าหนักเลื่อน. ทดสอบกับข้อมูลที่มีความยากต่าง ๆ กัน มีปริมาณข้อมูลต่าง ๆ กัน และเมื่อแบบจำลองมีความซับซ้อนต่าง ๆ กัน. สังเกตผล สรุปและอภิราย.

แบบฝึกหัด 5.17

จงเขียนโปรแกรมโครงข่ายประสาทเทียม ที่มีชั้นสัญญาณรบกวน ที่รับค่าหน่วยย่ออยู่ z เป็นอินพุต และให้ค่า z' เป็นเอาต์พุต โดย $z' = m \odot z$ เมื่อ m เป็นเมตริกซ์ขนาดเดียวกับ z และแต่ละส่วน ประกอบ $m \sim \mathcal{N}(1, \sigma)$ และ σ เป็นอภิมานพารามิเตอร์ กำหนดจากผู้ใช้.

ออกแบบการทดลอง เพื่อทดสอบประสิทธิภาพการใช้ชั้นสัญญาณรบกวน เปรียบเทียบกับการตัดออก ทั้งเรื่องการฝึก และผลของแบบจำลองที่ฝึกได้. ดำเนินการทดลอง สังเกตผล สรุปและอภิราย. ศึกษางานวิจัยของศรีวิวัฒนาและคณะ[189] อภิรายผลที่ได้ เปรียบเทียบกับผลจากศรีวิวัฒนาและคณะ.

รายการ 5.15 แสดงตัวอย่างโปรแกรมชั้นสัญญาณรบกวน.

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(
5             torch.tensor([1.0]), 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

การทํานายการแยกแยะ. เม็งการหาค่าถดถอย การจำแนกค่าทวิภาค และการจำแนกกลุ่ม เป็นกลุ่มภาระกิจที่มีการใช้งานมากที่สุด แต่การใช้งานโครงข่ายประสาทเทียม ไม่ได้จำกัดอยู่แต่เฉพาะกลุ่มภาระกิจที่นิยมเหล่านี้. โครงข่ายประสาทเทียม สามารถประยุกต์ใช้งานได้กว้างขวาง¹³. หลักการของวิธีค่าพังก์ชันควรจะเป็นสูงสุด (*maximum likelihood*) เป็นแนวทางหนึ่งที่ทั่วไปมากพอ ที่สามารถใช้ออกแบบพังก์ชันจุดประสงค์สำหรับภาระกิจต่าง ๆ ที่ต้องการได้.

หลักการของวิธีค่าพังก์ชันควรจะเป็นสูงสุด 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).

¹³เนื้อหา ในแบบฝึกหัดนี้ได้รับอิทธิพลหลัก ๆ จากกฎเพโลและคณ [76, §6.2.2.4].

แบบฝึกหัดนี้ เราจะศึกษาการทำโครงข่ายประสาทเทียมสำหรับทำนายการแยกของข้อมูล. นั่นคือ จากที่เคยใช้โครงข่ายประสาทเทียม f ทำนายค่าเออต์พุต y จากอินพุต x แบบฝึกหัดนี้จะใช้โครงข่ายประสาทเทียมทำนายการแยกของเออต์พุต y จากอินพุต x . แนวทางคือ แทนที่จะใช้โครงข่ายประสาทเทียมทำนายค่าความน่าจะเป็น¹⁴ $p(y|x)$ โดยตรง เราจะใช้โครงข่ายประสาทเทียมทำนาย $\Theta(x)$ ซึ่งนำไปใช้คำนวณค่าประมาณความน่าจะเป็น $p(y; \Theta(x)) \approx p(y|x)$ อีกต่อหนึ่ง.

แบบฝึกหัดนี้ การประมาณความน่าจะเป็น $p(y; \Theta(x))$ จะคำนวณด้วย แบบจำลองความหนาแน่นผสม (*mixture density model*). แบบจำลองความหนาแน่นผสม เป็นแบบจำลองทั่วไปในการประมาณค่าเออต์พุต จากอินพุต โดยรวมค่าประมาณจากส่วนผสมต่าง ๆ เข้าด้วยกัน อาจมองว่า แบบจำลองความหนาแน่นผสม มีพื้นฐานจากกฎผลบวก และกฎผลคูณของทฤษฎีเบลส์ได้ อันคือ $p(y|x) = \sum_{i=1}^M p(y|c=i)p(c=i|x)$ เมื่อ $c = i$ แทนส่วนผสม i และ M คือจำนวนส่วนผสมทั้งหมด. ส่วนผสม $c = i$ อาจมองเสมือนว่าเป็นสถานะภายในของความสัมพันธ์ระหว่าง x กับ y ได้. ค่า $p(y|c=i)$ ถูกประมาณด้วยความหนาแน่นของการแยกเงาส์เชียน. ดังนั้น สรุปคือ แบบจำลองความหนาแน่นผสม คำนวณ

$$p(y; \Theta(x)) = \sum_{i=1}^M p(c=i|x) \cdot \mathcal{N}(y; \mu_i(x), \sigma_i(x)) \quad (5.28)$$

เมื่อ $p(c=i|x)$ แทนความน่าจะเป็นของส่วนผสม i และ $\mathcal{N}(y; \mu_i(x), \sigma_i(x))$ เป็นค่าความหนาแน่นความน่าจะเป็นของการแยกเงาส์เชียน ที่มีค่าเฉลี่ย $\mu_i(x)$ กับค่าเบี่ยงเบนมาตรฐาน $\sigma_i(x)$. แบบจำลองเกาส์เชียนผสม สามารถใช้ประมาณค่าความน่าจะเป็นจากการแยกเงาได้ ได้ หากมีจำนวนส่วนผสมเพียงพอ. จำนวนส่วนผสม M เป็นอภิมานพารามิเตอร์ของแบบจำลอง.

สังเกต รูปแบบสมการ 5.28 เขียนสำหรับกรณีเออต์พุตมิติเดียว ($y \in \mathbb{R}$). กรณีที่ว่าไป ก็สามารถคำนวณได้ในลักษณะเดียวกัน นั่นคือ $p(\mathbf{y}; \Theta(\mathbf{x})) = \sum_{i=1}^M p(c=i|\mathbf{x}) \cdot \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_i(\mathbf{x}), \boldsymbol{\Sigma}_i(\mathbf{x}))$.

สำหรับจุดข้อมูลที่ n^{th} ค่าความน่าจะเป็น $p(y_n|x_n) \approx p(y_n; \Theta(x_n))$ และด้วยสมมติฐานไ.ไ.ดี. (i.i.d. ย่อจาก independent and identically distributed random variables ซึ่ง ณ ที่นี่ หมายถึง สมมติฐานว่าจุดข้อมูลแต่ละจุดเป็นอิสระต่อกัน และมีการแยกเงาเหมือนกัน) จะได้ว่า

$$p([y_1, y_2, \dots, y_N] | [x_1, x_2, \dots, x_N]) = \prod_{n=1}^N p(y_n|x_n)$$

เพื่อความสะดวกในการคำนวณ ค่าลอกการวิธีของฟังก์ชันควรจะเป็น (*log likelihood*) จะถูกนิยมมากกว่า. นอกจากนั้น สำหรับการฝึกแบบจำลอง นิยมวงกรอบเป็นปัญหาค่าน้อยที่สุด ค่าฟังก์ชันสูญเสีย สามารถกำหนดเป็น ค่าลบลอกการวิธีของฟังก์ชันควรจะเป็น (*negative log likelihood*). ดังนั้น

¹⁴ เพื่อให้เนื้อหาไม่ซับซ้อน ให้เขียนว่า “ความน่าจะเป็น” แทน “ค่าความน่าจะเป็น” แต่ในกรณีที่ตัวแปรที่เกี่ยวข้องเป็นตัวแปรสุ่มต่อเนื่อง จะหมายถึง ความหนาแน่นความน่าจะเป็น. คุณอาจจะรู้ว่า ความน่าจะเป็น คือความหนาแน่นความน่าจะเป็น.

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 ได้จากการใช้แบบจำลองความหนาแน่นผล. โครงข่ายประสาทเทียม สามารถใช้เพื่อประมาณ พารามิเตอร์ของแบบจำลองความหนาแน่นผล $\boldsymbol{\Theta} = [p(c=i|x), \mu_i(x), \sigma_i(x)]^T$ สำหรับ $i = 1, \dots, M$.

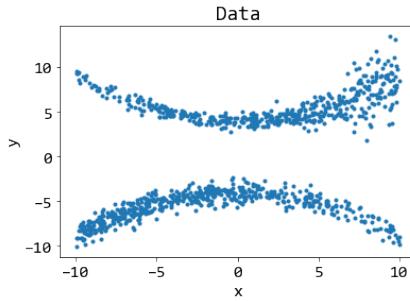
จงศึกษาการทำโครงข่ายประสาทเทียม สำหรับประมาณการแจกแจง ซึ่งมีรายละเอียด คือ (1) จงสร้างข้อมูล $\{x_n, y_n\}$ สำหรับ $n = 1, \dots, N$ โดยกำหนดความสัมพันธ์ระหว่างตัวแปรต้น x_n และตัวแปรตาม y_n ดังนี้

- (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 $0.25 + \frac{0.5}{1+\exp(-0.4x)} \times 100\%$. นอกนั้น y จะแสดงออกในลักษณะที่สอง.
- (2) จงกำหนดโครงข่ายประสาทเทียม ฝึก ทดสอบ สังเกตผล สรุป และอภิปราย.

จากข้อกำหนดของข้อมูล สังเกตว่า (ก) ลักษณะข้อมูลเป็นไปตามแบบจำลองความหนาแน่นผล และจำนวนลักษณะแสดงออก 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 เป็นฟังก์ชันของ x . (ค) ลักษณะแรก ทั้งค่าเฉลี่ย μ_0 และค่าเบี่ยงเบนมาตรฐาน σ_0 เป็นฟังก์ชันของ x . ส่วนลักษณะที่สอง ค่าเฉลี่ย μ_1 เป็นฟังก์ชันของ x แต่ค่าเบี่ยงเบนมาตรฐาน σ_1 เป็นค่าคงที่. (ง) ตัวแปรต้น $x \in \mathbb{R}$ ดังนั้น โครงข่ายประสาทเทียมรับอินพุตหนึ่งมิติ. (จ) พารามิเตอร์ของแบบจำลองความหนาแน่นผล จะมีทั้งหมด 6 ตัว ได้แก่ $\boldsymbol{\Theta} = [p_0, p_1, \mu_0, \mu_1, \sigma_0, \sigma_1]^T$ ซึ่งทั้ง 6 ค่านี้ จะคำนวณมาจากโครงข่ายประสาทเทียม. ดังนั้น โครงข่ายประสาทเทียมให้อาต์พุทกมิติ. สรุปคือ โครงข่ายประสาทเทียม $f : \mathbb{R} \mapsto \mathbb{R}^6$.

รูปที่ 5.18 แสดงตัวอย่างจุดข้อมูลที่สร้างตามข้อกำหนด. สังเกตว่า ที่ตัวแปรต้น x แต่ละค่า ตัวแปรตาม จะแสดงออกเป็นสองสูตร ซึ่งทั้งสองลักษณะมีการแจกแจงแบบสุ่ม โดยลักษณะแรก มีค่ามากกว่าลักษณะที่สอง และแนวโน้มข้อมูลจะโค้งขึ้น ในขณะที่ลักษณะที่สอง แนวโน้มข้อมูลจะโค้งลง. (เกี่ยวข้องกับ μ_0 และ μ_1 .) จุดข้อมูลลักษณะแรก จะเบาบาง (มีสัดส่วนจำนวนจุดน้อยกว่า) จุด

ข้อมูลลักษณะที่สอง ในช่วงค่า $x < 0$. จุดข้อมูลลักษณะที่สอง ดูเบาบางลง เมื่อ $x > 0$. (เกี่ยวข้องกับ p_0 และ p_1 .) การแจกแจงของจุดข้อมูลลักษณะที่สอง ดูคงที่ตลอดช่วงค่าของ x แต่จุดข้อมูลลักษณะแรก ดูเหมือนมีการแจกแจงเพิ่มขึ้นอย่างเห็นได้ชัดในช่วง x มีค่ามาก ๆ. (เกี่ยวข้องกับ σ_0 และ σ_1 .)



รูปที่ 5.18 ตัวอย่างจุดข้อมูล สำหรับโครงข่ายประสาทเทียม เพื่อทำนายการแจกแจง. แกนนอน แสดงค่าตัวแปรต้น x และ แกนตั้ง แสดงค่าตัวแปรตาม y .

ตัวอย่างโปรแกรม สำหรับสร้างข้อมูล แสดงในรายการ 5.16. โปรแกรมเขียนเป็นคลาส และเมท็อดที่ใช้สร้างข้อมูล คือ `sim_y` ซึ่งจะสร้างข้อมูลตัวแปรตาม ขึ้นมาจากการให้ข้อมูลตัวแปรต้นที่รับเข้าไป. โปรแกรม สามารถทดสอบได้ง่าย ๆ ด้วยคำสั่ง

```
r = relation()
xs = np.linspace(-10, 10, 1000)
ys = r.sim_y(xs)
```

ซึ่งค่า `xs` และ `ys` สามารถนำไปวาดกราฟ เพื่อศึกษาความสัมพันธ์ได้.

รายการ 5.16: โปรแกรม `relation` เพื่อสร้างข้อมูลสำหรับการทำนายการแจกแจง

```
1 class relation:
2     def __init__(self):
3         self.num_modes = 2
4         self.mode_chances = [lambda x: 0.25+0.5/
5                               (1+np.exp(-0.4*x)),
6                               lambda x: 1-(0.25+0.5/(1 + np.exp(-0.4*x)))]
7         self.fmu_y = [lambda x: 0.05*x**2 + 4,
8                       lambda x: -0.05*x**2 - 4]
9         self.fsigma_y = [lambda x: 0.2+np.log(1+np.exp(x-5)),
10                         lambda x: 0.5*np.ones(x.shape)]
11
12     def sim_y(self, xs):
13         N = len(xs)
14         p = np.random.uniform(0,1,N)
```

```

15     ys = np.array([])
16     for n in range(N):
17         xn = xs[n]
18         mode = self.num_modes-1
19         for i in range(self.num_modes-1):
20             pi = self.mode_chances[i](xn)
21             p[n] -= pi
22             if p[n] < 0:
23                 mode = i
24                 break
25         mun = self.fmu_y[mode](xn).reshape((-1,))
26         D = mun.shape[0]
27         sigman = self.fsigma_y[mode](xn).reshape((D,D))
28         yn = np.random.multivariate_normal(\n
29                                         mun,sigman,1).item()
30         ys = np.r_[ys, yn]
31     return ys

```

ตัวอย่างโปรแกรมคำนวณฟังก์ชันสัญเสียง (คำนวณสมการ 5.30) แสดงในรายการ 5.17. สังเกต (1) แบบจำลองความหนาแน่นผสม ถูกโปรแกรมเป็นส่วนหนึ่งของฟังก์ชันสัญเสียง. (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: โปรแกรมคำนวณค่าลับของการวิเคราะห์ของฟังก์ชันควรจะเป็น ที่ใช้แบบจำลองความหนาแน่นผสมเป็นพื้นฐาน. โปรแกรมรับค่า $yhat$ ที่อยู่ในรูปไฟรอนทุเพิล ($dmode$, 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 $-\infty$.

```

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

```

ค่าเอาร์พุตของໂຄງข่าย $\hat{y} = [p_0, p_1, \mu_0, \mu_1, \sigma_0, \sigma_1]^T$ 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)

ใช้ฟังก์ชันซอฟต์แมกซ์¹⁵ สำหรับ $[p_0, p_1]^T$ เพื่อคุณเงื่อนไขนี้. ค่า μ_0 และ μ_1 ไม่ใช้อัจฉริยะ. ค่า σ_0 และ σ_1 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 ใช้จำนวนหน่วยชั้นในเป็น 8 ใช้relu เป็นฟังก์ชันกระตุ้นชั้นชั้นใน และให้อาตต์พุตหากมิติ โดยอาตต์พุตแยกออกเป็นสามชุด 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) การฝึกสามารถทำได้ในลักษณะเดียวกับภาระกิจอื่น ๆ. อย่างไรก็ตาม รายการ 5.19 แสดงโปรแกรม **train** สำหรับตัวอย่างการฝึกโครงข่ายเพื่อทำนายการแยกแยะ. การฝึกสามารถทำได้ เช่นตัวอย่าง คำสั่ง

```

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 ไม่จำเป็นต้องใช้โครงสร้างแบบจำลองตามตัวอย่าง หรือไม่จำเป็นต้องฝึกดังโปรแกรม **train** ไม่จำเป็นต้องใช้ขั้นตอนวิธีดั้ม หรือไม่จำเป็นต้องสร้างข้อมูลใหม่ทุกสมัย ฝึกสามารถเลือกวิธีทำ และดำเนินการได้อย่างอิสระ.

รายการ 5.19: โปรแกรม **train** ฝึกโครงข่ายประสาทเทียม เพื่อทำนายการแยกแยะ. โดย **train** รับ **net**, **device**, **nepochs**, และ **lr** สำหรับโครงสร้างของแบบจำลอง, อุปกรณ์ที่ใช้รัน, จำนวนสมัยฝึก, และอัตราการเรียนรู้ ตามลำดับ. โปรแกรม **train** ใช้ขั้นตอนวิธีดั้มในการปรับค่าน้ำหนัก. และสร้างข้อมูลใหม่ทุก ๆ สมัยฝึก ด้วย **getdata**.

¹⁵เนื่องจาก p_0 และ p_1 ไม่ได้เบี่ยงเทียบกับหัวหนี้ร้อน การใช้ซอฟต์แมกซ์ อาจเพิ่มการคำนวณโดยไม่จำเป็น และอาจส่งผลเสียต่อเสียงรบกวนและคุณภาพของฝึกอีกด้วย. กรณีนี้ การทำอัตโนมัติให้เหมาะสม อาจจะเป็นทางเลือกที่ดีกว่า.

```

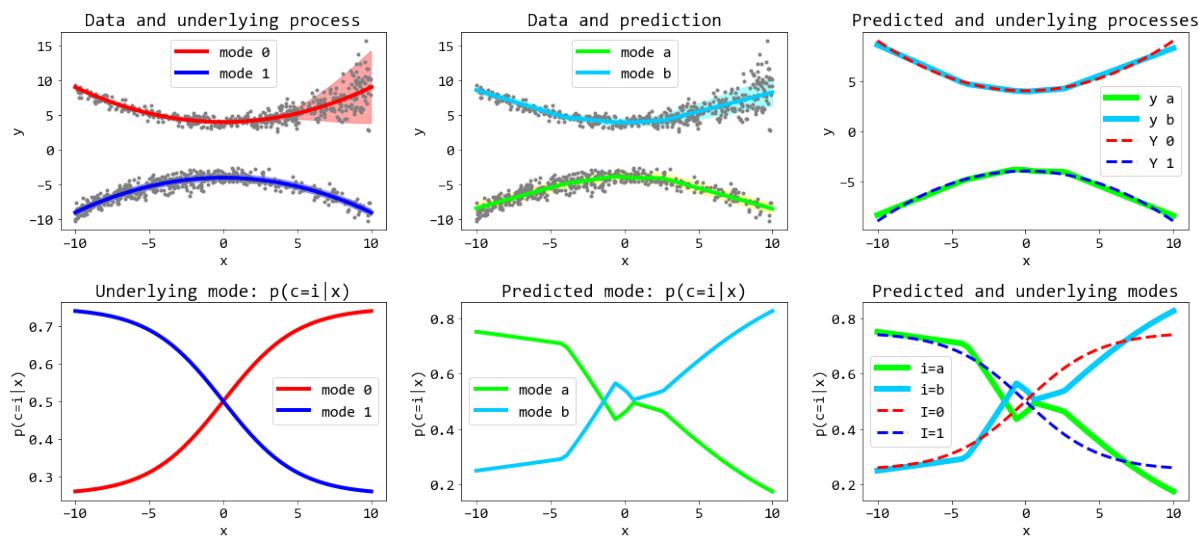
1 def train(net, device, nepochs, lr):
2     r = relation()
3     optimizer = torch.optim.Adam(net.parameters(), lr=lr)
4     net.train()
5     train_losses = []
6     for t in range(nepochs):
7         optimizer.zero_grad()
8         x, y = getdata(device, r)
9         yhat = net(x)
10        loss = 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

```

รูปที่ 5.19 แสดงตัวอย่างผลลัพธ์¹⁶. เนื่องจาก ลำดับของลักษณะ ไม่ได้มีความสำคัญ ดังนั้นเพื่อลดความสับสนจากลำดับ ค่าเฉลี่ยที่ใช้สร้างข้อมูลจะติดฉลากเป็น mode 0 และ mode 1 ในขณะที่ ผลทำนายจากแบบจำลอง จะติดฉลากเป็น mode a และ mode b.

จากการเปรียบเทียบ จะเห็นว่า แบบจำลองทำนายค่าเฉลี่ย μ_0 และ μ_1 ได้ดีมาก เปรียบเทียบภาพบนซ้ายกับภาพบนกลาง จะเห็นแนวเส้นคล้ายกันมาก (เส้นทึบฟ้า mode b คล้ายเส้นทึบแดง mode 0 และ เส้นทึบเขียว mode a คล้ายเส้นทึบน้ำเงิน mode 1) ความน่าจะเป็นของส่วนผสมแบบจำลองก็ทำนายได้ดีพอสมควร เปรียบเทียบภาพล่างซ้ายและภาพล่างกลาง.

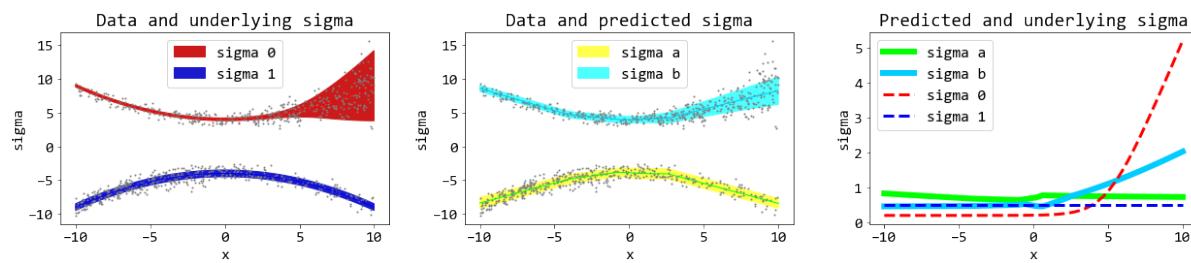
¹⁶หลังจากฝึกไป 4000 สมัยฝึก โดยเริ่มต้นค่าน้ำหนัก ด้วยวิธีกำหนดค่าหน้าที่นักเชิงเวียร์ ด้วยอัตรา $\sqrt{2}$ โดยการฝึก 2500 สมัยแรก ฝึกกับข้อมูลที่ $\sigma_0 = \sigma_1 = 0.5$ และ 1500 สมัยต่อมา จึงฝึกกับข้อมูลที่มีค่า σ_0 และ σ_1 ตั้งกำหนด. การฝึกกับข้อมูลดังกล่าว ตั้งแต่แรก พบร่ว่าให้ผลไม่ต่างกันอย่างมีนัยสำคัญ.



รูปที่ 5.19 ผลลัพธ์การเรียนการแยกจำแนก. ภาพบนช้ายและกลาง แสดงจุดข้อมูล (จุดสีเทา) พร้อมค่าเฉลี่ย (μ_0 เส้นสีแดง กับ μ_1 สีน้ำเงิน) ในภาพช้าย และเส้นฟ้ากับสีเขียวในภาพกลาง) และค่าเบี่ยงเบนมาตรฐาน (σ_0 และ σ_1) ซึ่งแสดงด้วยความกว้างของพื้นที่แรงงาน. ภาพบนขวา แสดงค่าเฉลี่ยของหัวค่าที่ทำนาย (ใช้สัญลักษณ์ y_a และ y_b สำหรับ mode a และ mode b ตามลำดับ) และค่าเฉลี่ย (ใช้สัญลักษณ์ Y_0 และ Y_1 สำหรับ mode 0 และ mode 1 ตามลำดับ). ภาพล่างช้ายและกลาง แสดงค่าความน่าจะเป็นของส่วนผสม (p_0 เส้นสีแดง กับ p_1 เส้นสีเขียว ในภาพช้าย และเส้นสีฟ้ากับเส้นสีเขียวในภาพกลาง). ภาพล่างขวา แสดงค่าความน่าจะเป็นของส่วนผสมหัวค่าที่ทำนาย (ใช้สัญลักษณ์ $i=a$ และ $i=b$ สำหรับ mode a และ mode b ตามลำดับ) และค่าเฉลี่ย (ใช้สัญลักษณ์ $I=0$ และ $I=1$ สำหรับ mode 0 และ mode 1 ตามลำดับ).

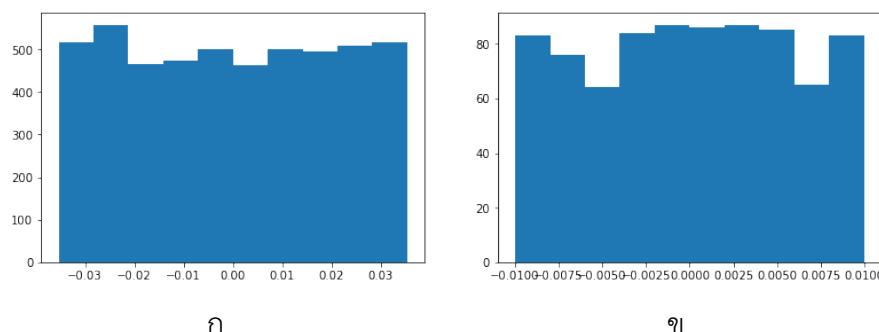
ภาพขวบันและล่าง แสดงค่าเฉลี่ย (ภาพบน) และความน่าจะเป็นของส่วนผสม (ภาพล่าง) หัวของเฉลี่ยและหัวที่ทำนายในภาพเดียวกัน. ค่าเบี่ยงเบนมาตรฐาน แสดงด้วยความหนาของพื้นที่แรงงาน ในภาพบนช้าย (เฉลี่ย) และภาพบนกลาง (ค่าทำนาย) ซึ่ง ลักษณะที่สอง (mode 1 ภาพช้าย และ mode a ภาพกลาง) อาจจะมองเห็นความหนาได้ยาก แต่ลักษณะแรก โดยเฉพาะช่วงปลาย เห็นชัดเจนว่า เฉลี่ย มีค่าเบี่ยงเบนมาตรฐานที่หนามาก แต่ค่าที่ทำนาย แม้จะดูหนาขึ้นในช่วงปลาย แต่ก็ดูแคบกว่าเฉลี่ยมาก.

รูปที่ 5.20 เน้นแสดงผลจากค่าเบี่ยงเบนมาตรฐาน (จุดข้อมูลแสดงด้วยขนาดที่เล็กลง และสีพื้นที่แรงงาน เลือกให้เข้มขึ้น ในภาพช้ายและภาพกลาง). ภาพขวา แสดงค่าเบี่ยงเบนมาตรฐาน ของหัวเฉลี่ยและหัวที่ทำนายในภาพเดียวกัน. ถึงแม้ค่าที่ทำนายอาจจะยังดูห่างจากเฉลี่ยมาก แต่เห็นได้ชัดว่าแบบจำลองสามารถจับแนวโน้มของ σ_0 (sigma b) ที่เพิ่มในช่วงปลาย และ σ_1 (sigma a) ที่คงที่ตลอดช่วงได้.



รูปที่ 5.20 ผลลัพธ์การเรียนค่าเบี่ยงเบนมาตรฐาน. ภาพซ้ายและกลาง แสดง จุดข้อมูล (จุดสีเทา) และค่าเบี่ยงเบนมาตรฐาน ที่แทนด้วยความหนาของพื้นที่เรขา. 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 $\text{fc1} = \text{torch.nn.Linear}(800, 5000)$ จะสร้างพารามิเตอร์ของชั้นคำนวณได้แก่ fc1.weight และ fc1.bias ซึ่งเป็นเทนเซอร์ สัดส่วน $(800, 5000)$ และ (5000) ตามลำดับ พร้อมค่าเริ่มต้น. โดยตัวฟอลต์ของไฟฟอร์ช ค่าเริ่มต้นทั้งของค่าน้ำหนักและไบอัส จะถูกกำหนดดังเช่นสมการ 5.7 นั่นคือ $\theta \sim \mathcal{U}\left(-\frac{1}{\sqrt{m_i}}, -\frac{1}{\sqrt{m_i}}\right)$ เมื่อ θ คือค่าน้ำหนักหรือไบอัสแต่ละค่า และ m_i คือจำนวนແຜ່ເຂົາຂອງชั้นคำนวณ. ตัวอย่างนี้ $m_i = 800$ และหากตรวจสอบการกระจายของค่าเริ่มต้นที่สร้างขึ้น ด้วยคำสั่ง เช่น $\text{plt.hist}(\text{fc1.bias.detach}())$ จะเห็นແນວภูมิແທ່ງຄລ້າຍຕ້ອງຢ່າງໃນรูปที่ 5.21 (ภาพ ก). สังเกต ค่าຕໍ່ສຸດສູງສຸດປະມານ -0.035 และ 0.035 ($\frac{1}{\sqrt{800}} \approx 0.035$).



รูปที่ 5.21 ตัวอย่างการแจกแจงค่าเริ่มต้นของไบอัส. ภาพ ก. $b \sim \mathcal{U}(-0.035, 0.035)$ และภาพ ข. $b \sim \mathcal{U}(-0.01, 0.01)$

หากต้องการกำหนดค่าเริ่มต้นนี้เป็นอื่นก็สามารถทำได้ ดังต่อไปนี้

```
with torch.no_grad():
    fc1.bias.data = 2*0.01*torch.rand(800) - 0.01
```

เปลี่ยนค่าไบอสเป็น $b \sim \mathcal{U}(-0.01, 0.01)$ ซึ่งเมื่อตรวจสอบ จะเห็นภาพคล้ายตัวอย่างในรูปที่ 5.21 (ภาพ ข). หมายเหตุ จุดสำคัญอยู่ที่ค่าสูงสุดต่ำสุด ไม่ใช่ความสูงต่ำของแผนภูมิแท้จริงแต่ละแท่ง (ที่โดยรวมแสดงการแจกแจงเอกรูป 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` เป็นตัวแปรแทนโครงข่ายประสาทเทียม ที่มีชั้นคำนวณ `fc1` และ `fc2` และต้องการกำหนดค่าเริ่มต้นของไบอสแต่ละค่า ให้เป็นค่าสุ่มจากการแจกแจงเอกรูป $\mathcal{U}(0, 1)$. การกำหนดค่าน้ำหนักก็สามารถทำได้ในลักษณะเดียวกัน.

อย่างไรก็ตาม เพื่อความสะดวก สำหรับการกำหนดค่าเริ่มต้นชั้นคำนวณต่าง ๆ ด้วยวิธีเดียวกัน เมื่อ `apply` ของ `nn.Module`¹⁷ สามารถช่วยลดภาระ การโปรแกรมช้าลงได้ ดังคำสั่ง

```
with torch.no_grad():
    net.apply(initx)
```

เมื่อ `net` คือตัวแปรโครงข่ายประสาทเทียมที่ต้องการกำหนดค่าน้ำหนักเริ่มต้น และ `initx` คือฟังก์ชันกำหนดค่าเริ่มต้นที่ต้องการใช้กับค่าน้ำหนัก (และไบอส) ทุกชั้นคำนวณ. รายการ 5.21 แสดงตัวอย่างโปรแกรมของฟังก์ชันที่ใช้กำหนดค่าน้ำหนักและไบอส (วิธีเซเวียร์ ดูสมการ 5.9 ประกอบ). เนื่องจาก เมื่อ `apply` จะรันฟังก์ชันกับทุก ๆ มอดูลอยู่ของ `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())
```

¹⁷`nn.Module` จะเป็นคลาสแม่ของคลาสโครงข่ายประสาทเทียมที่เราสร้างขึ้น ดังนั้น ตัวแปรวัตถุของคลาสโครงข่ายประสาทเทียมที่เราสร้างขึ้น จึงสามารถใช้เมื่อต้องต่อต่อกัน ของคลาส `nn.Module` ได้.

แบบฝึกหัด 5.19

จากตัวอย่างวิธีกำหนดค่าเริ่มต้น จงเขียนโปรแกรมกำหนดค่าเริ่มต้นแบบโคมิ้ง แล้วศึกษางานของโกลโลร์ตและเบนจิโอ[74] จากนั้น เลือกชุดข้อมูล ออกแบบการทดลอง เพื่อศึกษาผลกระทบจากฟังก์ชันกราฟตุ้น และวิธีการกำหนดค่าเริ่มต้น แล้วนำเสนอดังที่ได้เรียนรู้ อภิราย และสรุปผล.

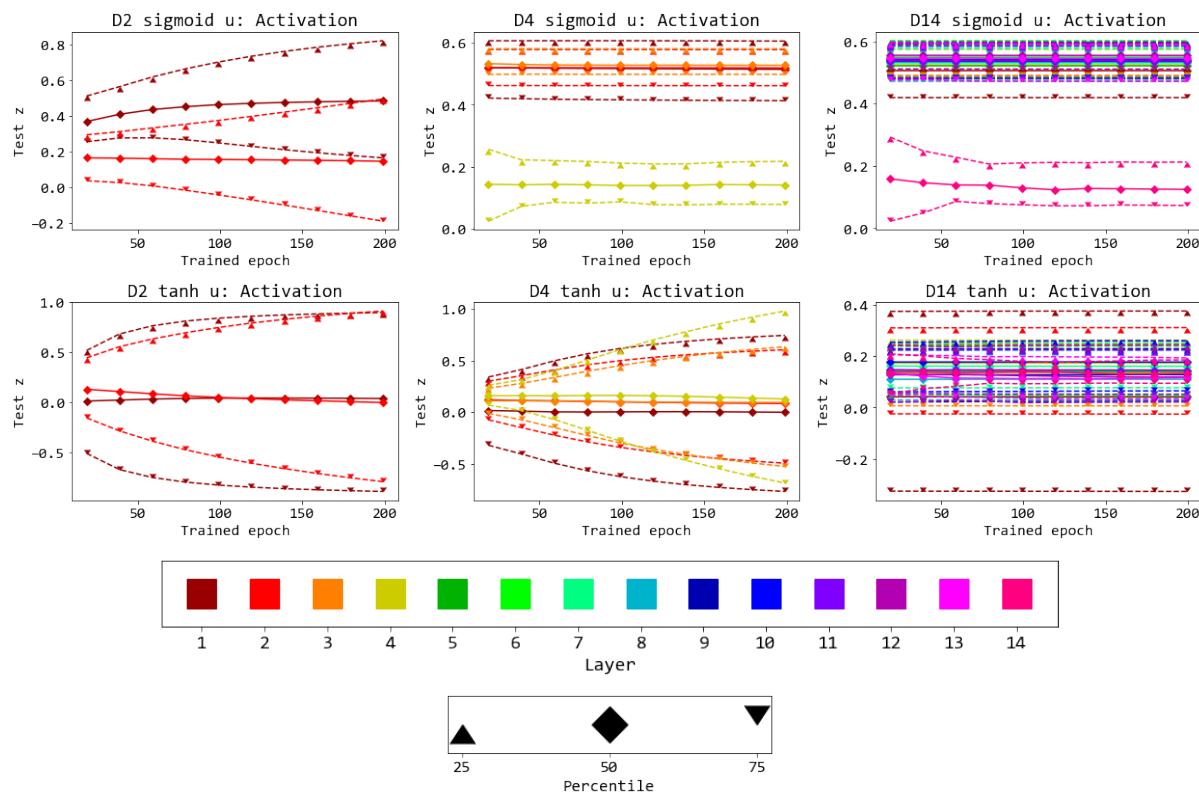
ตัวอย่างนำเสนอผลและอภิราย. รูปที่ 5.22 แสดงตัวอย่างผลที่คาดว่า เป็นส่วนหนึ่งที่ทำให้โกลโลร์ตและเบนจิโอ[74] ตั้งสมมติฐานว่า หากความแปรปรวนของค่าน้ำหนัก ที่ซั้นต่าง ๆ มีค่าใกล้เคียงกัน จะช่วยให้การฝึกทำได้ดียิ่ง. สังเกตว่า ในกรณีที่การฝึกทำได้ดี ความห่างระหว่างเปอร์เซ็นไทล์ที่ 25 และ 75 (สื่อถึงความแปรปรวน) ของชั้นคำนวนต่าง ๆ จะมีความห่างใกล้เคียงกัน แต่อาจจะมีการขยายไปเป็นซั้น ๆ จากชั้นตัน ๆ ที่จะขยายก่อนและໄไปเป็นหลัง ๆ ซึ่งต่างจากผลที่เห็น ในกรณีการฝึกล้มเหลว ที่ความแปรปรวนระหว่างชั้นคำนวนต่างกันอย่างชัดเจน นอกจานนี้ ก็ยังไม่เห็นการขยายของความแปรปรวน.

รูปที่ 5.23 แสดงตัวอย่างผลสรุปที่สำคัญ ได้แก่ (1) ฟังก์ชันกราฟตุ้น **tanh** และ **relu** ทำงานดีกว่า **sigmoid** ไม่ว่าจะใช้โครงข่ายที่มีความลึกเท่าใด. (2) ผลจากการกำหนดค่าเริ่มต้นด้วยวิธีเชเวียร์และโคมิ้ง (สัญกรณ์ σ และ k ในภาพ) จะเห็นชัดเจนขึ้น เมื่อใช้งานกับโครงข่ายที่ลึกขึ้น. (3) ทั้งวิธีเชเวียร์ และวิธีโคมิ้ง ให้ผล ตั้งที่คาดหมาย นั่นคือ วิธีเชเวียร์ ช่วยในกรณี **tanh** เมื่อเปรียบเทียบกับวิธีพื้นฐาน (สมการ 5.7 สัญกรณ์ σ ในภาพ). และวิธีโคมิ้ง ช่วยในกรณี **relu** เมื่อเปรียบเทียบกับทั้งวิธีเชเวียร์และวิธีพื้นฐาน. สังเกตว่า ทั้งวิธีเชเวียร์และวิธีโคมิ้งพัฒนาโดยอาศัยสมมติฐานเชิงเส้นที่แม่นจะไม่ตรงกับสถานการณ์จริง แต่ในทางปฏิบัติ กลับพบว่า ทั้งสองวิธีทำงานได้ดีอย่างชัดเจน.

นอกจาก ผลสรุปเรื่องวิธีการกำหนดค่าเริ่มต้น อีกประเด็นหนึ่งที่โกลโลร์ตและเบนจิโอ[74]ได้ศึกษา ก็คือ การตรวจดูค่าโดยรวม ของผลการกราฟตุ้นและค่าเกรเดียนต์ระหว่างฝึก (ดังเช่นที่แสดงในรูปที่ 5.22) และตรวจสอบการแยกแยะของผลการกราฟตุ้นภายหลังการฝึก (ดังเช่นรูปที่ 5.24) ที่ทั้งคู่คิดว่า น่าจะช่วยให้สามารถทำความเข้าใจพฤติกรรมการเปลี่ยนแปลงของโครงข่ายจากการฝึกได้ดีขึ้น.

รูปที่ 5.24 แสดงให้เห็นว่า กรณีที่การฝึกทำได้ดี (โครงข่ายสองชั้นทุกแบบ, โครงข่ายสี่ชั้น เมื่อใช้ **tanh** หรือ **relu**, โครงข่ายสิบสี่ชั้น เมื่อใช้ **tanh** และวิธีเชเวียร์ หรือเมื่อใช้ **relu** และวิธีโคมิ้ง. ดูรูปที่ 5.23 ประกอบ) หน่วยคำนวนส่วนใหญ่ (ในเกือบทุกกรณี ยกเว้น **tanh** และวิธีเชเวียร์) อยู่ในระดับอิมิตัว (*saturation*) นั่นคือ ค่าผลการกราฟตุ้น $z = 0$ หรือ $z = 1$ สำหรับฟังก์ชันซิกมอยด์, ค่าการกราฟตุ้น $z = -1$ หรือ $z = 1$ สำหรับฟังก์ชันไอกลีฟแทนเจนต์, และค่าการกราฟตุ้น $z = 0$ สำหรับเรลู.

หมายเหตุ การฝึกเขียนโปรแกรมเอง (เช่น รายการ 5.21) จะช่วยให้เข้าใจกลไกภายในได้ดี แต่การใช้งานในทางปฏิบัติ การใช้ฟังก์ชันสำเร็จรูป จะช่วยเพิ่มความสะดวกในการทำงานและการสื่อสารได้ดีขึ้น (โดยเฉพาะในกรณีทำงานด้วยกันหลายคน). ไฟฟอร์ชัมฟังก์ชันสำเร็จรูป สำหรับการกำหนดค่าเริ่มต้นด้วยวิธีที่รู้จักดีต่าง ๆ รวมถึงวิธีเชเวียร์และโคมิ้ง เช่น `nn.init.xavier_uniform_(w)` สำหรับการกำหนดค่าน้ำหนักเริ่มต้นให้กับพารามิเตอร์ W ด้วยวิธีเชเวียร์.



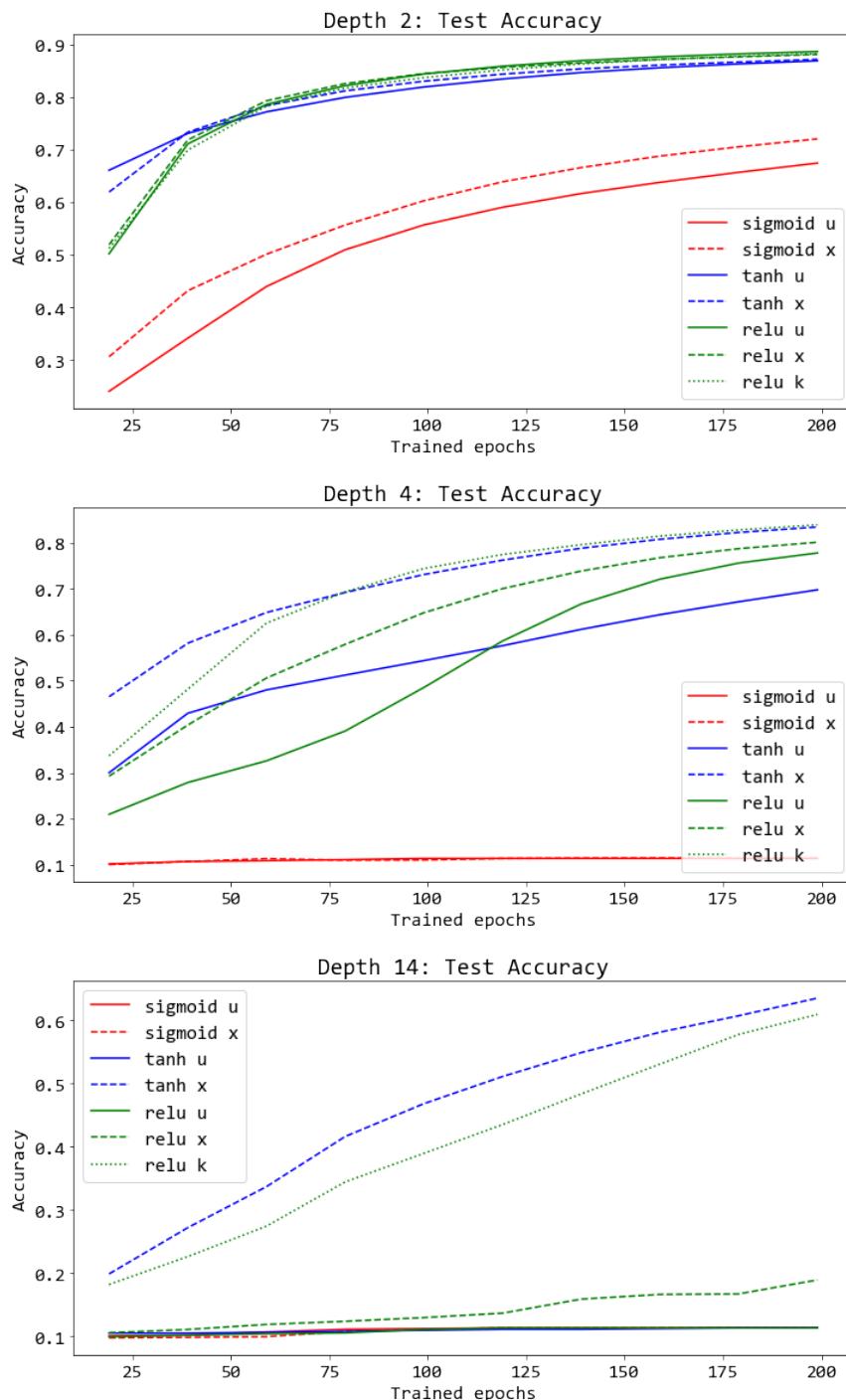
รูปที่ 5.22 ค่าผลการกระตุนระหว่างการฝึก (แสดงด้วยค่าเปอร์เซ็นไทล์ที่ 25, 50, และ 75) เมื่อใช้ฟังก์ชันซิกมอยด์ (ภาพในแนวนอน) และไฮเปอร์บolic tangent (ภาพในแนวตั้ง) โดยมีโครงข่ายมีความลึก 2 ชั้น (ภาพซ้าย ระบุเหนือภาพด้วย D2), 4 ชั้น (ภาพกลาง ระบุด้วย D4), และ 14 ชั้น (ภาพขวา ระบุด้วย D14). ภาพในแนวนอน แสดงสี สำหรับค่าผลการกระตุนที่ชั้นคำนวณต่างๆ และสัญลักษณ์ที่ระบุค่าเปอร์เซ็นไทล์. ในหกแบบจำลองนี้ มีสามแบบจำลองที่การฝึกทำได้สำเร็จได้แก่ แบบจำลองสองชั้นที่ใช้ซิกมอยด์ (ภาพซ้ายบน) แบบจำลองสองและสี่ชั้นที่ใช้ไฮเปอร์บolic tangent (ภาพซ้ายและกลาง 例外ที่สอง).

แบบฝึกหัด 5.20

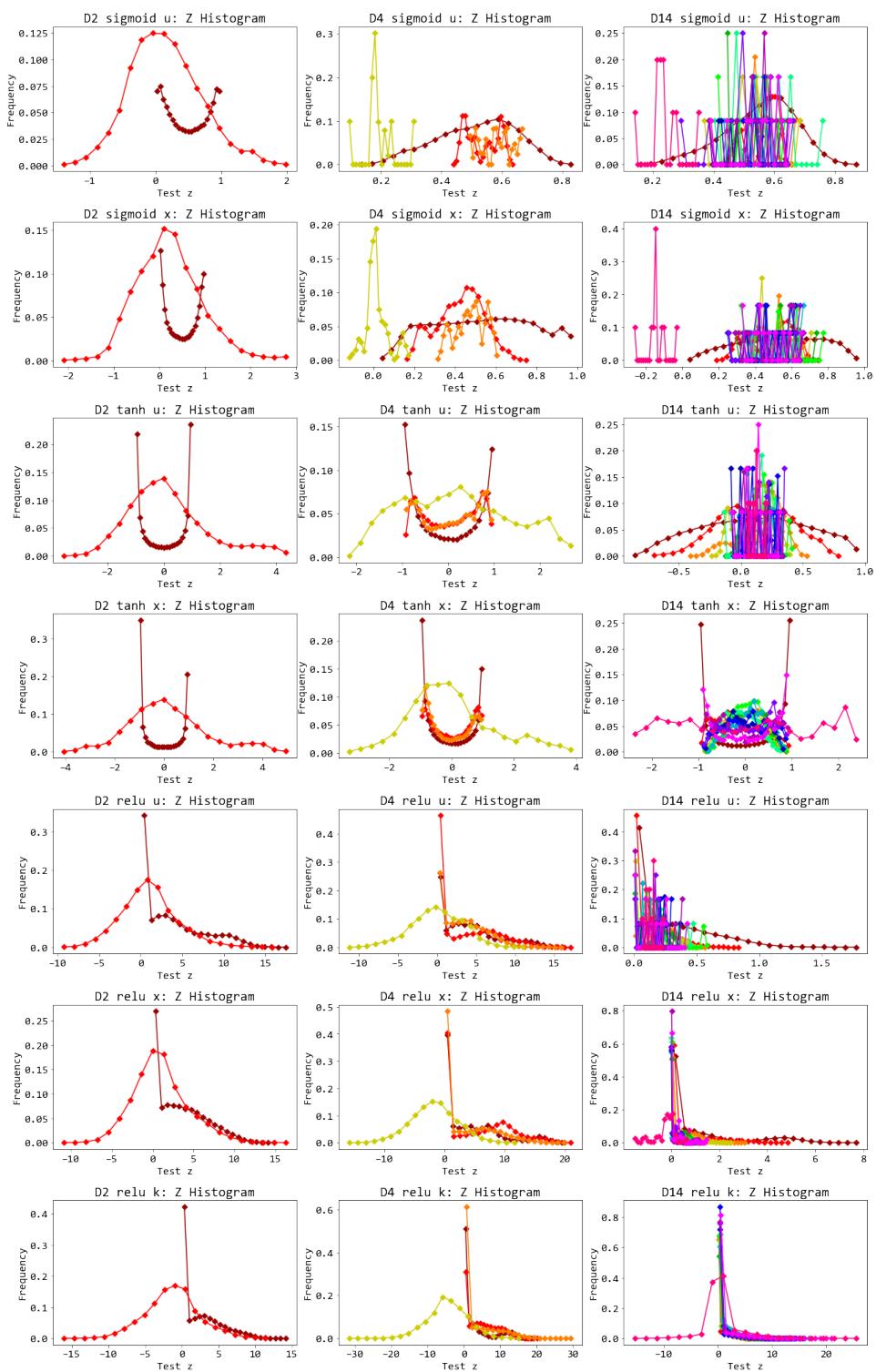
จะเลือกขั้นตอนวิธีการฝึก จากวิธีลงเกรเดียนต์กับกลไกโมเมนตัม, วิธีอาร์เอมเอสพรอป, วิธีอาร์เอมเอสพรอป กับกลไกโมเมนตัม, วิธีอัตโนมัติ หรือวิธีอื่น ๆ ที่สนใจ และเขียนโปรแกรมวิธีดังกล่าว เปรียบเทียบผลการทำงานกับโปรแกรมสำเร็จรูปของวิธีนั้น (ได้แก่ **optim.SGD**, **optim.RMSprop**, และ **optim.Adam**) และเปรียบเทียบกับวิธีลงเกรเดียนต์ เลือกชุดข้อมูลขึ้นมาเพื่อทดสอบ ออกแบบการทดลอง เพื่อวัดผลทั้งในเชิงความเร็วและคุณภาพในการเรียนรู้ รวมถึงความทนทานต่อค่าอภิมานพารามิเตอร์ต่าง ๆ ที่เลือกใช้ และความทนทานกับการกำหนดค่าเริ่มต้นแบบต่าง ๆ ปกปาราย และสรุป.

แบบฝึกหัด 5.21

จะออกแบบการทดลอง เพื่อทดสอบการทำงานของแบบนอร์ม วัดผลทั้งในความเร็วในการฝึก คุณภาพการฝึก ความทนทานต่อค่าอัตราการเรียนรู้ ผลกระทบขนาดของหมู่เล็ก รวมถึงตำแหน่งที่ทำแบบนอร์ม (ทำที่ตัวกระตุน นั่นคือ ก่อนฟังก์ชันกระตุน เปรียบเทียบกับทำที่ผลการกระตุน นั่นคือ หลังฟังก์ชัน



รูปที่ 5.23 ผลเปรียบเทียบฟังก์ชันกระตุ้นและวิธีกำหนดค่าเริ่มต้น. ค่าความแม่นยำกับข้อมูลทดสอบ เมื่อใช้โครงข่ายความลึก 2 ชั้น (ภาพบนสุด) 4 ชั้น (ภาพกลาง) และ 14 ชั้น (ภาพล่าง) ประกอบกับฟังก์ชันกระตุ้นในชั้นซ่อนแบบต่างๆ (**sigmoid** ซิกมอยด์, **tanh** ไฮเปอร์บolic tangent เจนต์, และ **relu** เรลู) และวิธีกำหนดค่าเริ่มต้นแบบต่างๆ (**u** วิธีพื้นฐาน สมการ 5.7, **X** วิธีเซเวียร์ สมการ 5.9, และ **K** วิธีเคลิง สมการ 5.12). ค่าความแม่นยำที่แสดง เป็นค่าเฉลี่ยจากการทำซ้ำสิบครั้ง.



รูปที่ 5.24 การแจกแจงของค่าการกระตุน. แต่ละภาพแสดงการแจกแจงของค่าการกระตุน สำหรับแต่ละกรณี (ความลึก พิมพ์ชั้นกระตุนที่ใช้ และวิธีกำหนดค่าเริ่มต้น ระบุหนึ่งตัวอย่างภาพ ด้วยรหัสเข่นเดียวกับรูปที่ 5.22 และ 5.23). แกนนอนแสดงค่าผลการกระตุน และแกนตั้ง แสดงค่าความถี่หารด้วยจำนวนทั้งหมด.

กระตุ้น) 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.

รายการ 5.22: ตัวอย่างโปรแกรมโครงข่ายประสาทเทียมที่ใช้แบบนอร์ม. แบบนอร์มเหมือนชั้นคำนวนที่เพิ่มขึ้น. คลาส MyBN เป็นชั้นคำนวนแบบนอร์ม กำหนดดังแสดงในรายการ 5.23. หมายเหตุ การใช้แบบนอร์ม ทำให้ใบอัสเกินความจำเป็นและซ้ำซ้อน และสามารถตัดออกได้. แต่ในตัวอย่างนี้ไม่ได้ตัดค่าใบอัสออก. หากต้องการตัดใบอัสออก สามารถทำได้โดยคำสั่ง เช่น `self.fc1 = 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
19         return self.z3

```

รายการ 5.23: โปรแกรมแบบนอร์ม. การใช้ `nn.Parameter` ช่วยให้มีเมธอด `parameters()` ของแบบจำลอง รู้จักพารามิเตอร์ที่ทำหนดชื่น (และการปรับค่าพารามิเตอร์ตามเกรเดียนต์ที่จะถูกทำโดยอัตโนมัติ เช่นเดียวกับค่าน้ำหนักและใบอัส). การใช้ `register_buffer` ลงทะเบียนตัวแปร เพื่อให้ตัวแปรที่ลงทะเบียน ถูกรวบเข้าไป เมื่อทำการกำหนดอุปกรณ์ (เช่น `net.to(device)`) หรือการบันทึกแบบจำลอง (เช่น `torch.save(net.state_dict(), 'savnet.pth')`) แต่ไม่ถูกรวบเข้าไปในกลุ่มพารามิเตอร์ของแบบจำลอง (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))

```

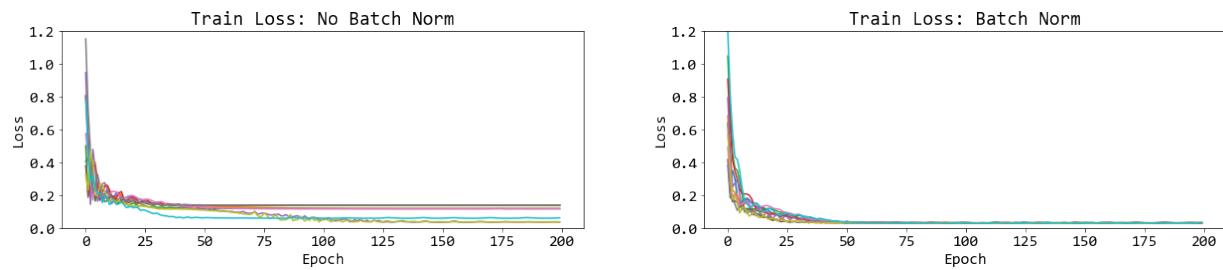
```

8     self.bias = nn.Parameter(torch.zeros(num_features))
9     self.register_buffer('running_mean',
10                         torch.zeros(num_features))
11    self.register_buffer('running_var',
12                         torch.ones(num_features))
13
14    def forward(self, z):
15        mu = self.running_mean
16        svar = self.running_var
17        if self.training:
18            with torch.no_grad():
19                mu = torch.mean(z, dim=0)
20                svar = torch.var(z, dim=0)
21                # Tracing running_mean and running_var
22                p = self.momentum
23                q = 1 - p
24                self.running_mean = p*mu + q*self.running_mean
25                self.running_var = p*svar + q*self.running_var
26            # end self.training
27        zn = (z - mu)/torch.sqrt(svar + self.eps)
28        zns = zn * self.weight + self.bias
29        return zns

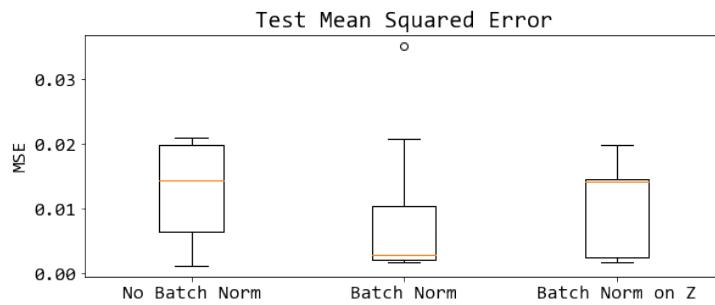
```

รูปที่ 5.25, 5.26, 5.27, และ 5.28 แสดงตัวอย่างการนำเสนอผล. รูปที่ 5.25 แสดงค่าสูญเสียระหว่างการฝึก จากการทดสอบสิบช้า ภาพชาย เมื่อไม่ได้ใช้แบนดอร์ม และภาพขาว เมื่อใช้แบนดอร์ม. เห็นได้ชัดเจนว่า แบนดอร์มช่วยให้การฝึกทำได้เร็วขึ้นและแน่นอนขึ้น.

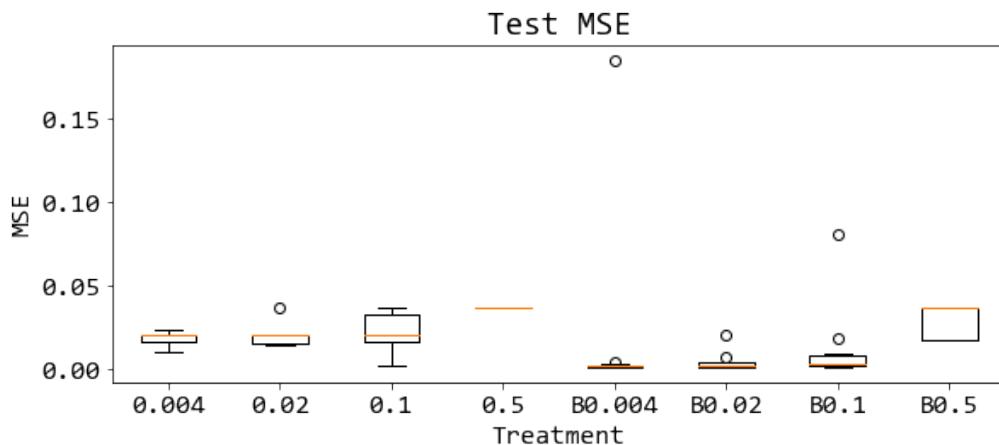
รูปที่ 5.26 แสดงค่าทดสอบ ซึ่งในที่นี้ใช้ค่าเฉลี่ยกำลังสองน้อยที่สุด. ภาพแสดงด้วยแผนภูมิกล่องกล่องช้ายสุด แสดงค่าผิดพลาด เมื่อไม่ใช้แบนดอร์ม. กล่องกลาง เมื่อใช้แบนดอร์ม (ทำที่ตัวกระตุน นั่นคือ ใช้ ทำที่ \mathbf{A} เมื่อ ขั้นคำนวณ ทำ $h(\mathbf{A})$ โดย $\mathbf{A} = \mathbf{W} \cdot \mathbf{Z} + \mathbf{b}$ หรือ ทำก่อนเข้าฟังก์ชันกระตุน). กล่องขวา เมื่อใช้แบนดอร์ม แต่ทำแบนดอร์มที่ผลการกระตุน แทนที่จะทำที่ตัวกระตุน (นั่นคือ ทำที่ \mathbf{Z} หรือทำหลังฟังก์ชันกระตุน). รูปที่ 5.26 แสดงในเห็นว่า ไม่เพียงแต่ แบนดอร์มช่วยให้การฝึกดำเนินการได้เร็วขึ้น แบนดอร์มยังช่วยคุณภาพการฝึกด้วย และเพื่อให้ได้ประสิทธิภาพที่ดี การทำแบนดอร์มควรทำที่ค่าตัวกระตุน (ค่าก่อนเข้าฟังก์ชันกระตุน).



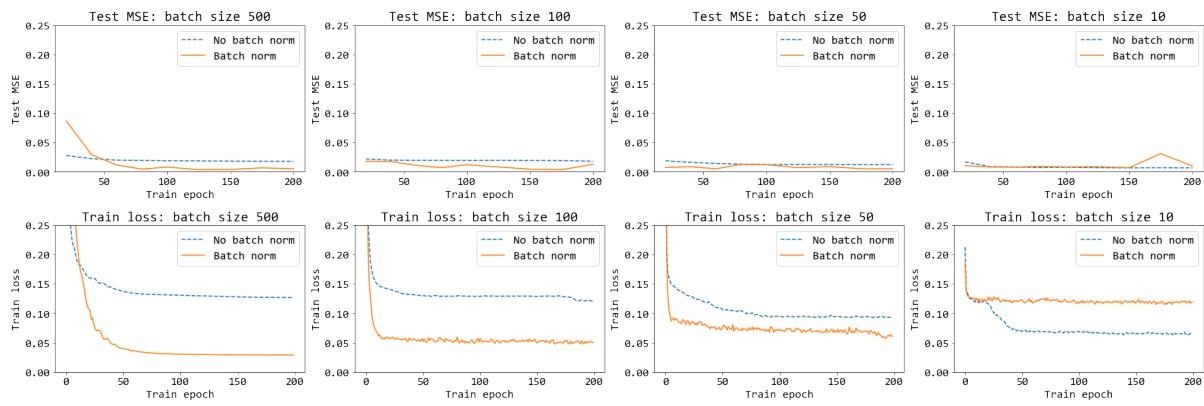
รูปที่ 5.25 ค่าฟังก์ชันสูญเสียต่อสมัยฝึก จากการทดสอบซ้ำ 10 ครั้ง เมื่อไม่ใช้แบนด์อร์ม (ภาพซ้าย) และใช้แบนด์อร์ม (ภาพขวา).



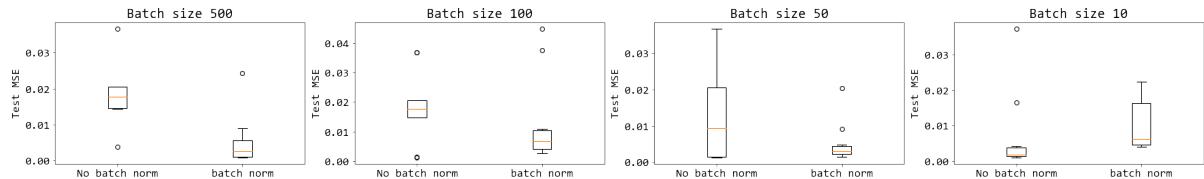
รูปที่ 5.26 ค่าผิดพลาดเมื่อไม่ใช้และใช้แบนด์อร์ม. แผนภูมิกล่อง แสดงค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากการทดสอบ 10 ครั้ง เมื่อไม่ใช้แบนด์อร์ม (กล่องซ้าย), เมื่อใช้แบนด์อร์ม (กล่องกลาง) และเมื่อใช้แบนด์อร์ม แต่ทำแบนด์อร์มที่ผลการกระตุ้น (กล่องขวา).



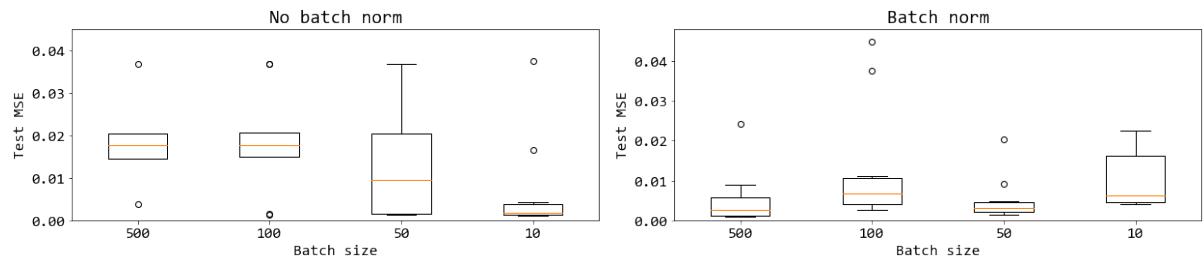
รูปที่ 5.27 ค่าผิดพลาดเมื่อไม่ใช้และใช้แบนด์อร์ม ที่อัตราเรียนรู้ต่าง ๆ. แผนภูมิกล่องแสดงค่าเฉลี่ยค่าผิดพลาดกำลังสอง จากการทดสอบ 10 ครั้ง เมื่อไม่ใช้แบนด์อร์ม และเมื่อใช้แบนด์อร์ม ที่อัตราเรียนรู้ต่าง ๆ. ตัวเลขที่แสดง หมายถึง ค่าอัตราเรียนรู้ที่ใช้ และอักษร B ที่กำกับหมายถึง มีการใช้แบนด์อร์ม.



รูปที่ 5.28 ค่าเฉลี่ยค่าผิดพลาด ระหว่างการฝึก เมื่อใช้และไม่ใช้แบบนอร์ม ที่ขนาดหมู่เล็กต่าง ๆ (ภาพต่าง ๆ ในแถวบน). ค่าเฉลี่ยค่าฟังก์ชันสูญเสีย ระหว่างการฝึก เมื่อใช้และไม่ใช้แบบนอร์ม ที่ขนาดหมู่เล็กต่าง ๆ (ภาพต่าง ๆ ในแถวล่าง)..



รูปที่ 5.29 แบบนอร์มและขนาดของหมู่เล็ก. แผนภูมิกล่อง แสดงค่าผิดพลาด จากการทดสอบ 10 ครั้ง เมื่อไม่ใช้และใช้แบบนอร์ม กับขนาดหมู่เล็กต่าง ๆ (จากข้ามสุด ไปข้ามสุด ขนาดหมู่เล็ก 500, 100, 50, และ 10 ตามลำดับ).



รูปที่ 5.30 แบบนอร์มเปลี่ยนความล้มเหลวของการฝึกและขนาดหมู่เล็ก. แผนภูมิกล่อง แสดงค่าผิดพลาด จากการทดสอบ 10 ครั้ง เมื่อใช้ขนาดหมู่เล็กต่าง ๆ ในกรณีที่ไม่ใช้และใช้แบบนอร์ม. เปรียบเทียบกับรูปที่ 5.29 รูปนี้นำเสนอด้วยกันมุ่งมอง หนึ่ง นั่นคือ การเลือกไม่ใช้หรือใช้แบบนอร์ม ควรพิจารณาประกอบกับขนาดของหมู่เล็กที่จะเลือกใช้ด้วย.



รูปที่ 5.31 คุณภาพของแบบ_norm กับการเลื่อนของความแปรปรวนร่วมเกี่ยวกับขนาดบัช. ภาพซ้าย แสดงการทำงานของแบบ_norm กับการเลื่อนของความแปรปรวนร่วมเกี่ยวกับขนาดบัช. อัตราการปรับปรุงคุณภาพการทำงาน เมื่อใช้แบบ_norm เทียบกับไม่ใช้ แสดงด้วยเส้นประสีแดง โดย ค่าเป็นบทบาทหมายถึงคุณภาพดีขึ้น เมื่อใช้แบบ_norm ศูนย์หมายถึงคุณภาพเท่ากัน และค่าเป็นลบหมายถึงคุณภาพแย่ลง. เส้นทึบสีน้ำเงิน แสดงค่ามากที่สุดของความต่างระหว่างค่าเฉลี่ยของหมู่กับค่าเฉลี่ยของข้อมูลทั้งหมด (ความต่างคิดเป็นค่าสัมบูรณ์). ภาพขวา แสดงค่ามากที่สุดของความต่าง (แคนนอน) กับเบอร์เช็นต์ การปรับปรุงคุณภาพเมื่อใช้แบบ_norm (แกนตั้ง).

รูปที่ 5.27 แสดงให้เห็นว่า แบบ_norm ทำงานได้ดีที่ค่าอัตราเรียนรู้ต่าง ๆ. รูปที่ 5.28 แสดง การทำงานของแบบ_norm ในสถานการณ์ของขนาดหมู่เล็กต่าง ๆ ในช่วงสมัยฝึกต่าง ๆ. สังเกตว่า เมื่อใช้ หมู่เล็กขนาดเล็กเกินไป (ภาพขวาสุด บนและล่าง) แบบ_norm ทำงานได้ไม่ดี และนำไปสู่การฝึกที่แยกกว่า การฝึกที่ไม่ใช้แบบ_norm. รูปที่ 5.29 แสดงค่าความผิดพลาดเมื่อนำแบบจำลองที่ฝึกไปทดสอบ. รูปที่ 5.29 ยืนยันว่า หากใช้แบบ_norm และเลือกขนาดหมู่เล็กที่เล็กเกินไป จะทำให้ผลการฝึกแย่ลงได้.

แบบ_norm ออกแบบมาเพื่อแก้ไขการเลื่อนของความแปรปรวนร่วมเกี่ยวกับใน ที่เกิดจากการปรับค่าพารามิเตอร์ระหว่างการฝึก แต่การทำแบบ_norm ที่ปรับค่าเฉลี่ยและความแปรปรวนของหมู่เล็ก ถ้าเสียงที่จะทำสารสนเทศจากข้อมูลเสียหาย. หากความต่างของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากตัวข้อมูลเอง ไม่ใช่มาจากการเปลี่ยนแปลงของค่าพารามิเตอร์ในชั้นคำนวนก่อนหน้า.

หากการแจกแจงสารสนเทศของข้อมูลในหมู่เล็กแต่ละหมู่ ค่อนข้างคงเส้นคงวา และสามารถแทน การแจกแจงสารสนเทศของข้อมูลโดยรวมได้ การเลื่อนของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากการเปลี่ยนแปลงของค่าพารามิเตอร์ของชั้นคำนวนก่อนหน้า การทำแบบ_norm จะมีประสิทธิผล ตามที่ออกแบบไว้.

แต่หากการเลื่อนของค่าเฉลี่ยและความแปรปรวนระหว่างหมู่ มาจากสารสนเทศที่ต่างกันของข้อมูลระหว่างหมู่เอง ความเสียงของการทำแบบ_norm จะสูงขึ้นมาก และเมื่อประกอบกับแนวทางปฏิบัติ ของการสุมหมู่เล็ก ที่มักทำการสุมแค่ครั้งแรก และใช้ลำดับและการจัดกลุ่มนั้นตลอด ยิ่งจะซ้ำเติมความเสียงนี้เข้าไปอีก.

ตัวอย่างของกรณีที่ใช้ขนาดหมู่เล็กเป็นสิบ ภาพขวาสุดที่แสดงในรูปที่ 5.29 แสดงให้เห็นว่า เมื่อ ขนาดหมู่เล็กลง โอกาสที่การแจกแจงสารสนเทศของข้อมูลระหว่างหมู่จะสม่ำเสมอหรือจะเป็นตัวแทน

ของข้อมูลทั้งหมดได้ จะน้อยลง และเมื่อการแจกแจงสารสนเทศของข้อมูลระหว่างหมู่ไม่สม่ำเสมอ การทำแบบชันอิร์มจึงทำการสนับสนุนอย่างเสียหายไป และส่งผลให้การฝึกทำได้แย่. รูปที่ 5.30 เปรียบเทียบให้เห็นว่า ความสัมพันธ์ระหว่างคุณภาพการฝึกกับขนาดของหมู่เล็ก เปลี่ยนไป เมื่อใช้แบบชันอิร์ม. ในทางปฏิบัติ หลายครั้ง ขนาดของหมู่เล็ก อาจถูกจำกัดจากหน่วยความจำ และการเลือกใช้หรือไม่ใช้แบบชันอิร์ม ควรคำนึงถึงความเสี่ยง จากประเด็นความสม่ำเสมอระหว่างหมู่เล็ก ของการแจกแจงสารสนเทศจากตัวข้อมูลเองประกอบ.

รูปที่ 5.31 แสดงประเด็นความสัมพันธ์ระหว่างประสิทธิผลการทำงานของแบบชันอิร์มกับการแจกแจงข้อมูลระหว่างหมู่เล็ก. จากรูป เมื่อ การแจกแจงข้อมูลระหว่างหมู่เล็ก ทำให้เกิดความต่างระหว่างหมู่เล็กมาก ประสิทธิผลการทำงานของแบบชันอิร์มจะต่ำลง และอาจต้องการใช้แบบชันอิร์มจะเป็นผลเสียมากกว่าผลดี ดังแสดงออกมาเป็นเปอร์เซ็นต์ปรับปรุงที่ติดลบ.

แล้วกรณีที่ข้อมูลมีรูปแบบแบลก ๆ ที่พับได้ยาก หรือกรณีประดิ่นเรื่องวิธีประมาณค่า μ_i และ σ_i^2 ที่ใช้แบบชันอิร์มภายหลังการฝึก อาจก่อให้เกิดความเสี่ยงอย่างไรบ้าง จะระดมความคิด ภูมิปัญญา และสรุป.

แบบฝึกหัด 5.22

การศึกษาหัวข้อที่สนใจ.¹⁸ บางครั้งในบางจังหวะเวลา ศาสตร์ที่เรารู้ ไม่สามารถเปลี่ยนแปลงพัฒนาที่รวดเร็วมาก และอาจจำเป็นต้องศึกษาความก้าวหน้าและพัฒนาการล่าสุดจากแหล่งอื่น ๆ เพิ่มเติม.

จะเลือกหัวข้อเรื่องที่สนใจ (เช่น การบรรยายภาพอัตโนมัติ, การแต่งเพลงอัตโนมัติ, การทำนายพฤษศาสตร์แบบโปรตีน) และศึกษา ทำความเข้าใจในหัวเรื่องดังกล่าว.

การศึกษา ทำความเข้าใจ อาจทำโดย (1) ค้นหาและรวบรวมแหล่งข้อมูล เกี่ยวกับหัวข้อที่สนใจ ซึ่งแหล่งข้อมูลอาจรวมถึง หนังสือ บทความวิชาการ บทความทั่วไป วีดีโอ เว็บเพจ เป็นต้น.

(2) ศึกษาแต่ละแหล่งข้อมูลที่ได้มาคร่าว ๆ (ถ้าเป็นบทความวิชาการ รวมถึงบทความวิจัย อย่างน้อย อ่านบทคัดย่อและบทหน้า) และอาจจะทำบันทึกย่อ ว่า แต่ละแหล่งข้อมูลเกี่ยวข้องกับหัวข้อที่สนใจมากน้อยขนาดไหน และเราเข้าใจเนื้อหาเข้าใจมากน้อยขนาดไหน รวมถึงอาจจัดลำดับความสำคัญ และหมายเหตุถึงสิ่งที่คิดว่าจะดำเนินการต่อ เช่น ไม่ค่อยเกี่ยวข้อง ตัดทิ้งไปก่อน หรือ เกี่ยวข้องมาก ศึกษาให้เข้าใจ หรือ เกี่ยวข้องประมาณ 50% พอเข้าใจแล้ว เก็บไว้เป็นตัวอย่าง หรือ ไม่แน่ใจว่าเกี่ยวข้องเข้าใจดี ขอบเทคนิคที่เข้าใช้ อาจใช้เป็นประโยชน์กับงานของเราได้ เก็บไว้อ้างอิงทีหลัง. หรือ ไม่แน่ใจว่าเกี่ยวข้อง ไม่ค่อยเข้าใจ เก็บไว้ดูอีกทีหลังจากเข้าใจหัวข้อนี้ดีขึ้น.

โดยทั่วไป ถ้าเป็นบทความวิชาการหรือบทความวิจัย ที่ไม่ใช่บทความทบทวน หรือไม่ใช่บทความสำรวจ เราอาจจะต้องอ่านตั้งแต่สามจนถึงสิบบทความ จึงอาจจะพอเข้าใจหัวข้อนั้นในระดับเบื้องต้นได้. นักศึกษาปริญญาเอก ซึ่งถูกคาดหวังว่าจะเข้าใจในหัวข้อเป็นอย่างดี อาจต้องอ่านบทความ ไม่น้อยกว่า

¹⁸ ตัดแปลงจาก [4].

100 บทความ. และก็ไม่เปลี่ยนที่จะเห็น บทความทบทวน หรือบทความสำรวจอหัวข้อใด ๆ ที่คณผู้เขียนอาจต้องอ่านบทความต่าง ๆ ที่เกี่ยวข้องกับหัวข้อนั้น ๆ มากกว่า 300 บทความ.

(3) หากไม่สามารถเข้าใจบทความวิชาการส่วนใหญ่ได้ ให้ระบุศาสตร์พื้นฐานที่อาจจะขาดไป เช่น บทความที่ทึ่ง เกี่ยวข้องมาก แต่ไม่เข้าใจเลย ดูเหมือนจะใช้ทฤษฎีความน่าจะเป็นและแคลคูลัสของ การแปรผัน (*calculus of variations*). บทความที่สอง เกี่ยวข้อง แต่อ่านไม่เข้าใจ ใช้พีชคณิตเชิงเส้น, ความน่าจะเป็น และทฤษฎีสารสนเทศ. บทความที่ห้า เกี่ยวข้องมาก แต่ยังอ่านไม่เข้าใจ ใช้ทฤษฎีความน่าจะเป็น, กระบวนการสโทแคสติก และการหาค่าดีที่สุด. บทความที่แปด เกี่ยวข้องบ้าง แต่อ่านไม่เข้าใจ เพราะประยุกต์ใช้กับงานชีวการแพทย์ ใช้การหาค่าดีที่สุด และชีวเคมี. เช่นนี้ ก็จะช่วยให้เราพอเห็นว่า เราขาดพื้นฐานอะไรไปบ้าง และเราสามารถจัดลำดับความสำคัญ และเลือกที่จะศึกษาพื้นฐานเหล่านี้ก่อน. หมายเหตุ เราไม่จำเป็นต้องสร้างพื้นฐานทุก ๆ อย่าง เช่น เราอาจเลือกว่า สิ่งที่เราสนใจไม่ได้เกี่ยวข้องกับงานชีวการแพทย์ เราอาจจะไม่เลือกเรียนรู้พื้นฐานด้านชีวเคมี ก็ได้ หรือ เราคิดว่า ทฤษฎีกระบวนการสโทแคสติก มีใช้บ้างกับงานบางประเภท บางแนวทาง ซึ่งเราอาจจะยังไม่สนใจ ก็สามารถทำได้. แต่ระวังว่า หากอ่านบทความวิชาการไม่รู้เรื่อง และพบว่าพื้นฐานอะไรบ้างที่เราต้องการ แต่เราไม่อยากเรียนรู้พื้นฐานเหล่านั้นเลย (โดยเฉพาะพื้นฐานที่จำเป็น) อาจเป็นสัญญาณเตือนว่า จริง ๆ แล้ว ใจเราอาจจะไม่อยากศึกษาหัวข้อที่เลือกจริง ๆ.

(4) จากแหล่งข้อมูลที่ได้ศึกษาเบื้องต้น เลือกแหล่งที่เกี่ยวข้องมาก ๆ ออกแบบศึกษาต่อ ให้ละเอียดขึ้น. สำหรับความวิจัย ให้ลองสรุปบทความ โดยระบุการค้นพบที่สำคัญ, วิธีที่ใช้, และคุณค่าเมื่อมองจากภาพรวมใหญ่ของหัวข้อ รวมถึงความสัมพันธ์กับงานอื่น ๆ. อาจอภิปรายประเด็นเพิ่มเติมด้วย เช่น ข้อจำกัด หรือศักยภาพ หรือการตีความจากมุมมองอื่น หรืออาจจะเป็นความเห็นส่วนตัว หรือสิ่งที่ชอบประเด็นที่ไม่ชอบ หากมี.

(5) ถ้าสามารถทำได้ หากลุ่มคนที่สนใจเรื่องเดียวกัน และอภิปรายเรื่องที่เรียนรู้ต่าง ๆ ด้วยกัน (อาจเป็นกลุ่มที่สามารถพบปะตัวต่อตัว หรือกลุ่มแบบออนไลน์ก็ได้). เรื่องที่จะอภิปรายอาจจะเลือกอย่างอิสระตามความสนใจของกลุ่ม หรือหากไม่รู้จะเริ่มจากเรื่องใด อาจลองพิจารณาจากคำถามเหล่านี้ เป้าหมายที่สำคัญของหัวข้อนี้คืออะไร? หัวข้อนี้มีศักยภาพและประโยชน์ต่อสังคมในวงกว้างอย่างไร? ความท้าทายที่สำคัญของหัวข้อนี้มีอะไรบ้าง? แนวทางและวิธีการต่าง ๆ ที่ใช้เพื่อจัดการกับความท้าทาย มีอะไรบ้าง? และแต่ละแนวทางมีข้อดีข้อเสียอะไร? งานเด่น ๆ ในหัวข้อดีมีอะไรบ้าง ทำไมมันถึงเด่นกว่างานอื่น ๆ? อะไรคือสิ่งที่คนในวงการนี้สนใจและอยากได้มากที่สุด? ทำไมถึงอย่างได้? ในความเห็นส่วนตัวแล้ว คิดว่า นอกจากแนวทางหลัก ๆ แล้ว มีแนวทางอื่นอีกไหม? แนวทางไหนบ้างที่น่าสนใจและทำไม? เป็นต้น

คำแนะนำในการอ่านบทความวิจัย. ก่อนอ่าน อาจจะถามตัวเองว่า ต้องการอะไรบ้าง จากบทความที่กำลังจะอ่าน. หากมีสิ่งที่ต้องการรู้เฉพาะจากบทความ เช่น อยากรู้วิธีประเมินผล เมื่ออ่านก็ให้ความสำคัญเป็นพิเศษกับวิธีประเมินผล และหลังอ่านเสร็จให้กลับมาตอบตัวเอง ว่าได้รู้สิ่งที่ค้นหาไว้อย่างไร.

แต่หากเป็นการอ่านเพื่อความเข้าใจภาพโดยทั่วไป ไม่ได้มีประเด็นที่จะเป็นพิเศษ อาจลองวิธี

ดังนี้ (1) อ่านผ่าน ๆ รอบแรก โดยอ่านชื่อเรื่อง บทคัดย่อ และรูปภาพ. (2) อ่านบทนำ และบทสรุป แล้วตีความและเนื้อหาส่วนอื่น ๆ คร่าว ๆ. (3) อ่านเนื้อหาต่าง ๆ ในบทความ โดยอาจจะยังไม่ต้องสนใจรายละเอียด โดยเฉพาะนิพจน์หรือสมการคณิตศาสตร์มากนัก. (4) ทำความเข้าใจส่วนต่าง ๆ รวมถึงพจน์ นิพจน์ และสมการคณิตศาสตร์ต่าง ๆ. (5) ตั้งคำถามกับตัวเอง เช่น บทความนี้พยายามศึกษาหรือแก้ปัญหาอะไรอยู่? แนวทางหรือวิธีที่ใช้ มันมีอะไรเป็นปัจจัยสำคัญ? เนื้อหาที่อ่านมีประโยชน์อะไรบ้างกับเรา? มีอ้างอิงรายการไหนบ้างที่เรารอယากจะตามศึกษาต่อ? (6) หากสนใจบทความ อาจจะลองอภิปรายเพิ่มเติม เช่น ผลการศึกษาอาจมีข้อจำกัดอะไรบ้าง หรืออาจแสดงถึงศักยภาพอะไรบ้าง? จุดน่าสนใจ ความคิดสร้างสรรค์ของงานนี้ มีที่ใดบ้าง?

หลังอ่านจบแล้ว อาจจะงบทวนดูว่า มันช่วยตอบคำถามอะไรบ้างในภาพรวม ยังมีอะไรบ้างที่เป็นสิ่งที่เราสงสัยอยู่? สำหรับนักศึกษาปริญญาเอก หากสิ่งที่เราสงสัย เป็นสิ่งที่ในวงการก็ยังไม่รู้ (ศึกษาแหล่งข้อมูลให้มากพอ เพื่อแนใจว่า ในวงการยังไม่รู้) และเป็นสิ่งที่หากรู้แล้วจะมีประโยชน์ สิ่งนั้นอาจเป็นตัวเลือกที่น่าสนใจสำหรับหัวข้อวิจัยได้ ถ้าเราพอที่จะช่วยคลายความสงสัยนั้นลงได้บ้าง (การเลือกหัวข้อวิจัยมีความเสี่ยงสูงมาก ควรปรึกษาอาจารย์ที่ปรึกษาก่อนตัดสินใจ).

บทที่ 6

โครงข่ายคอนโวลูชัน

``Adapt or perish, now as ever, is nature's inexorable imperative." ที่ไม่อาจหลีกเลี่ยงได้ ทั้งตอนนี้และforevermore."

---H. G. Wells

—เอช จี เวลส์

ดังที่กล่าวในบทที่ 5 โครงข่ายคอนโวลูชัน เป็นศาสตร์และศิลป์ที่สำคัญของการเรียนรู้ของเครื่อง. โครงข่ายคอนโวลูชัน อาศัยโครงสร้างที่เหมาะสมกับข้อมูลที่มีลักษณะเชิงท้องถิ่น (*high locality*) และมีลักษณะไม่แปรเปลี่ยนตามการเลื่อนขาน (*highly translation invariant*). ข้อมูลภาพในงานคอมพิวเตอร์วิทัค์ ที่มีลักษณะเช่นนี้. งานคอมพิวเตอร์วิทัค์ เป็นงานแรก ๆ ที่ โครงข่ายคอนโวลูชัน ได้พิสูจน์ให้เห็นศักยภาพของการเรียนรู้เชิงลึก และปัจจุบัน โครงข่ายคอนโวลูชัน ได้กลายเป็นศาสตร์และศิลป์ของคอมพิวเตอร์วิทัค์ในหลาย ๆ ด้าน. ยาน เลอคุน (Yann LeCun) ผู้เชี่ยวชาญการเรียนรู้เชิงลึก และผู้บุกเบิกการใช้งานโครงข่ายคอนโวลูชัน ยกย่อง[114] โครงข่ายคอนโวลูชัน ว่าเป็นส่วนที่นำพาการที่สำคัญมาให้กับการประมวลผลภาพ วิดีโอ คำพูด และเสียง.

โครงข่ายคอนโวลูชัน (*Convolutional Neural Network* 俗稱 CNN) เป็นโครงข่ายประสาทเทียม ที่จำกัดการเชื่อมต่อของหน่วยอยู่ และมีการใช้ค่าน้ำหนักร่วมกัน. โครงข่ายคอนโวลูชัน ถูกออกแบบมาสำหรับข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง และรูปแบบเชิงท้องถิ่นมีลักษณะร่วมกัน ไม่ซึ้งกับตำแหน่งที่ปรากฏ. ดังนั้นการจำกัดการเชื่อมต่อและการใช้ค่าน้ำหนักร่วมกัน จึงช่วยเพิ่มประสิทธิภาพการคำนวณอนุมาณและการฝึกโครงข่ายขึ้นอย่างมาก เพราะ ได้ลดภาระการคำนวณของการเชื่อมต่อที่มีผลกระทบน้อยออกไป และสามารถใช้สารสนเทศในแต่ละจุดข้อมูลได้คุ้มค่ามากขึ้น.

ข้อมูลหลาย ๆ ชนิด มีลักษณะเชิงท้องถิ่นสูง และเป็นลักษณะเชิงท้องถิ่นภายในตัวโครงสร้างของลำดับมิติ เช่น ภาพสเกลเทา ถูกแทนด้วยข้อมูลสองลำดับชั้นของค่าความเข้มพิกเซล (ลำดับค่าพิกเซลในแกนนอน และลำดับค่าพิกเซลในแกนตั้ง), ภาพสี ถูกแทนด้วยข้อมูลสามลำดับชั้น (ลำดับค่าพิกเซลในแกนนอน, ลำดับค่าพิกเซลในแกนตั้ง, และชุดของช่องสีแดงเขียวเข้ม), ภาพมัลติสเปกตรัม (multispectral image) ถูกแทนด้วยข้อมูลหลายลำดับชั้น (ลำดับค่าพิกเซลในแกนนอน, ลำดับค่าพิก-

เซลในแกนตั้ง, และชุดต่าง ๆ แต่ละชุดแทนค่าของแต่ละสเปกตรัม) วิดีโอุกแทนด้วยลำดับของภาพ, เสียงอุกแทนด้วยสเปกโตแกรมเสียง (audio spectrogram), ข้อความภาษาเขียนอุกแทนด้วยลำดับของอักษร เป็นต้น.

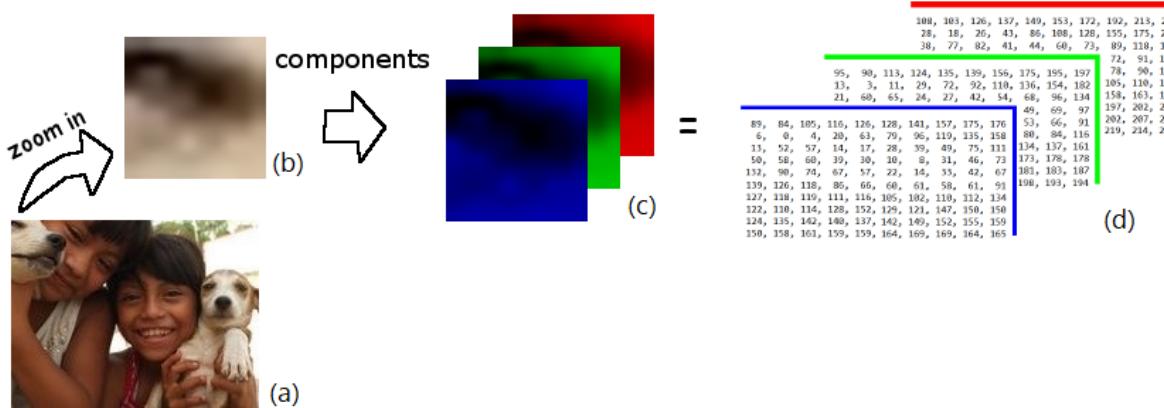
การที่กล่าวว่า ข้อมูลมีลักษณะเชิงท้องถิ่น คือตัวอย่าง เช่น ภาพถ่ายมุมสูงของหมู่บ้าน ภายในภาพจะเห็นบริเวณและตัวบ้านหลาย ๆ หลัง. รูปแบบของบ้านแต่ละหลัง จะสังเกตได้ทันทีจากตัวบ้านและบริเวณใกล้เคียง โดยไม่ต้องอาศัยส่วนอื่นของภาพที่อยู่ห่างออกไปประกอบ. นอกจากนั้น ภาพของหมู่บ้านภาพเดียว อาจแสดงรูปแบบของบ้านหลาย ๆ หลัง และแม้ว่าบ้านแต่หลังอาจแตกต่างกัน แต่มักจะมีลักษณะบางอย่างร่วมกันอยู่ เช่น อาจจะเป็น ขนาด รูปทรง หรือวัสดุที่ใช้ทำหลังคा. นั่นคือ นอกจากรูปแบบของบ้านจะมีลักษณะเชิงท้องถิ่นแล้ว รูปแบบเชิงท้องถิ่นยังมีลักษณะร่วมกันอีกด้วย ซึ่งลักษณะร่วม เช่นนี้ ที่ทำให้วิธีการใช้ค่าน้ำหนักร่วมของโครงข่ายคอนโวลูชัน สามารถใช้ประโยชน์จากแต่ละจุดข้อมูลได้คุ้มค่า เช่น การเรียนรู้รูปแบบของบ้าน จากส่วนภาพของบ้านหลาย ๆ หลัง ที่ปรากฏในภาพเดียวกัน.

โครงข่ายคอนโวลูชัน กับมิติและลำดับชั้นของข้อมูล. บทที่ 2 ได้อธิบาย ความหมายของมิติจากมุมมองต่าง ๆ และแนะนำความหมายของลำดับชั้น. ลำดับชั้น (rank) คือโครงสร้างลำดับมิติ หรือชุดมิติ.

โครงสร้างของโครงข่ายประสาทเทียมดังเดิม มองข้อมูลในรูปมิติที่ไม่มีโครงสร้างระหว่างมิติ ตัวอย่าง เช่น ข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ (แบบฝึกหัด 3.15) แต่ละจุดข้อมูลจะมี 6 เขตข้อมูล ซึ่ง เขตข้อมูลความร้ายแรงเป็นเอตพุต 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 ข้อมูลภาพสีในรูปที่ 6.1. ข้อมูลภาพมีโครงสร้างมิติที่ชัดเจน นั่นคือมี (1) ชุดมิติของสี ที่มีสำหรับสีน้ำเงิน เขียว และแดง (2) ชุดลำดับมิติของพิกเซลตามแนวตั้ง และ (3) ชุดลำดับมิติของพิกเซลตามแนวอน. หากแต่ละค่าความเข้มพิกเซลมีค่าอยู่ระหว่าง 0 ถึง 255 สามารถใช้ เทคนิคลำดับชั้นสาม $\mathbf{X} \in \{0, \dots, 255\}^{3 \times 133 \times 175}$ สำหรับแทนภาพสีหนึ่งภาพ ขนาดสูง 133 พิกเซล และกว้าง 175 พิกเซลได้. ข้อมูลลักษณะนี้ แม้ว่าแต่ละจุดข้อมูล (แต่ละภาพ) จะมี 69825 ค่า (ภาพหนึ่งภาพดังตัวอย่าง ต้องการหน่วยความจำ 69825 ไบต์) แต่ค่าเหล่านี้ มีความสัมพันธ์กันเชิงลำดับด้วย เช่น ถ้าค่าพิกเซลที่อยู่ติด ๆ กันมีค่าใกล้เคียงกัน นั่นอาจหมายถึงลายเส้นที่สื่อความหมาย และแม้จะมีบางค่าพิกเซลขาดหรือเกินไปบ้าง ความหมายก็ไม่ได้เปลี่ยนแปลง. ความหมายได้จากผลเชิงรวมของค่าพิกเซลบริเวณใกล้เคียงกัน. การที่ค่าต่าง ๆ บริเวณใกล้เคียงมีความสัมพันธ์ต่อกันในเชิงความหมายรวม จะเรียกว่า ลักษณะเชิงท้องถิ่น.

โครงข่ายประสาทเทียมดังเดิม ไม่มีกลไกที่รองรับลำดับมิติ แต่ก็สามารถนำข้อมูลที่มีลำดับมิติ



รูปที่ 6.1 โครงสร้างของมิติ. ตัวอย่างข้อมูลที่เป็นภาพซึ่งมีโครงสร้างของมิติ (dimensional structure). ภาพล่างช้าย (a) เป็นภาพสีขนาดสูง 133 พิกเซล กว้าง 175 พิกเซล. เมื่อลองขยายมุมบนซ้ายของภาพ บริเวณขนาด 10×10 พิกเซล (ขยาย 10 เท่า) จะได้ภาพ ดังแสดงในภาพ (b). ภาพ (b) ที่เป็นภาพสี นั้นมีส่วนประกอบของสามสี ดังแสดงในภาพ (c). และค่าความเข้มของพิกเซลที่ดำเนินการห่างต่าง ๆ ของแต่ละช่องสี แสดงตั้งตัวอย่างในภาพ (d). ภาพสีตั้งกล่าว (ภาพ a) จะมีโครงสร้างเป็นสามชุดมิติ หรือแทนด้วย เทนเซอร์ลำดับชั้นสามเช่น $\mathbf{X} \in \{0, \dots, 255\}^{3 \times 133 \times 175}$ สำหรับภาพสี (3 ช่องสี) ขนาดสูง 133 พิกเซล กว้าง 175 พิกเซล แต่ละค่าแทนความเข้มของสีระหว่าง 0 ถึง 255.

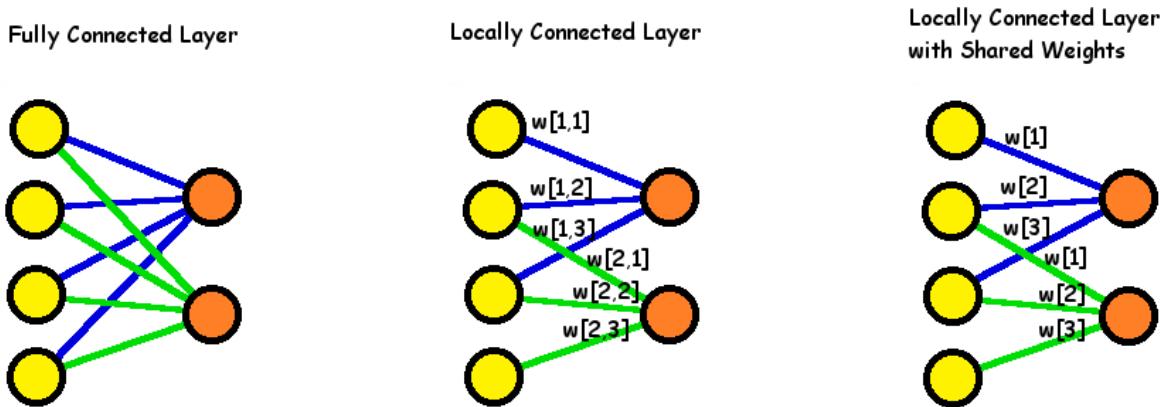
เข้าไปประมวลผลได้ โดยการบุบชุดลำดับมิติรวมกัน เช่น จุดข้อมูล $\mathbf{X} \in \mathbb{R}^{3 \times 133 \times 175}$ บุบเป็นจุดข้อมูล $\mathbf{x} \in \mathbb{R}^{69825}$. แม้ว่า ข้อมูลที่มีลำดับมิติจะสามารถถูกบุบได้ แต่การทำเช่นนี้จะทำให้โครงสร้างธรรมชาติของข้อมูลสูญหายไป (หรือไม่ชัดเจน) และมีผลทำให้การฝึกโครงข่ายประสาทเทียม เพื่อทำงานกับข้อมูลลักษณะนี้ทำได้ยากขึ้น. โครงข่ายคอนโวลูชัน ออกแบบมาโดยเฉพาะ เพื่อรับข้อมูลที่มีโครงสร้างลำดับมิติ.

เพื่อลดการสับสนกับคำว่ามิติในความหมายเดิม คำว่า ลำดับมิติ หรือ ชุดมิติ หรือ ชุดลำดับมิติ หรือ ลำดับชั้น จะถูกใช้สำหรับเน้นถึงโครงสร้างที่มีลักษณะลำดับมิติ.

กลไกที่สำคัญของโครงข่ายคอนโวลูชันที่ใช้ประโยชน์จากลักษณะโครงสร้างมิติของข้อมูล ประกอบด้วย (1) การเชื่อมต่อท้องถิ่น (local connections), (2) การใช้ค่าน้ำหนักร่วม (shared weights), (3) การดึงรวม (pooling), (4) การใช้โครงสร้างต่อ กันหลายชั้น (multiple layers). กลไกเหล่านี้ ดำเนินการผ่านชั้นคำนวนสองแบบ คือ ชั้นคอนโวลูชัน (convolution layer) และ ชั้นดึงรวม (pooling layer). ชั้นคอนโวลูชัน ทำกลไกของการเชื่อมต่อท้องถิ่น และการใช้ค่าน้ำหนักร่วม. ชั้นดึงรวม ทำกลไกการเชื่อมต่อท้องถิ่น และการดึงรวม. โครงข่ายคอนโวลูชัน จะใช้ทั้งชั้นคอนโวลูชันและชั้นดึงรวม หลาย ๆ ชั้นต่อ กัน และอาจใช้ชั้นคำนวนแบบดึงเดิม ที่มักเรียกว่า ชั้นเชื่อมต่อเต็มที่ (fully connected layer)¹ สำหรับคำนวนเอาต์พุตในรูปแบบที่ต้องการ.

รูปที่ 6.2 แสดงแผนภาพเบรียบเทียบ ชั้นเชื่อมต่อเต็มที่, ชั้นเชื่อมต่อท้องถิ่น, และ ชั้นเชื่อมต่อท้องถิ่นที่ใช้ค่าน้ำหนักร่วม. รูปที่ 6.2 แสดงกรณีข้อมูลที่มีโครงสร้างหนึ่งลำดับชั้น. ชั้นคอนโวลูชัน ก็คือชั้นเชื่อมต่อท้องถิ่นที่ใช้ค่าน้ำหนักร่วม. รูปที่ 6.2 เป็นแผนภาพอย่างง่าย ที่แสดงข้อมูลหนึ่งลำดับชั้น. การ

¹ คำว่า ชั้นเชื่อมต่อเต็มที่ มักใช้เพื่อจำแนกชั้นที่มีการเชื่อมต่อแบบดึงเดิม ออกจาก ชั้นคอนโวลูชัน

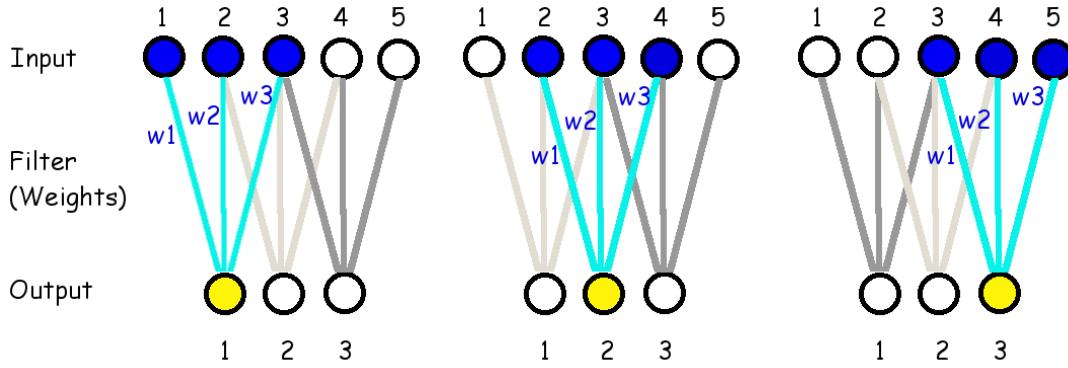


รูปที่ 6.2 ขั้นการเชื่อมต่อแบบต่าง ๆ รูปเปรียบเทียบการเชื่อมต่อเต็มที่ (ภาพซ้าย) การเชื่อมต่อห้องถิน (ภาพกลาง) และการเชื่อมต่อห้องถินที่มีการใช้ค่าน้ำหนักร่วม (ภาพขวา). อินพุต แสดงด้วย วงกลมทางซ้ายของแต่ละภาพ (อินพุตมี 4 หน่วยในแต่ละภาพ). เอาต์พุต แสดงด้วย วงกลมทางขวาของแต่ละภาพ (เอาต์พุตมี 2 หน่วยในแต่ละภาพ). เส้นตรงที่ เชื่อมระหว่างอินพุตและเอาต์พุต แทนการเชื่อมต่อ หรือค่าน้ำหนักของเอาต์พุตสำหรับอินพุตต่างๆ. ภาพซ้ายแสดงชั้นเชื่อม ต่อเต็มที่ เอาต์พุตแต่ละตัวมีการเชื่อมต่อกับอินพุตทุกๆตัว จำนวนค่าน้ำหนักที่ต้องการ เท่ากับ จำนวนเอาต์พุตคูณจำนวน อินพุต (8 ค่าในภาพตัวอย่าง). ภาพกลางแสดงชั้นเชื่อมต่อห้องถิน เอาต์พุตแต่ละตัวมีการเชื่อมต่อกับอินพุตแค่บางตัว เท่านั้น จำนวนค่าน้ำหนักที่ต้องการ เท่ากับ จำนวนเอาต์พุตคูณจำนวนอินพุตที่อยู่ในห้องถิน (6 ค่าในภาพตัวอย่าง). ภาพ ขวาแสดงชั้นเชื่อมต่อห้องถินที่มีการใช้ค่าน้ำหนักร่วม เอาต์พุตแต่ละตัวมีการเชื่อมต่อกับอินพุตแค่บางตัวเท่านั้น แต่การ เชื่อมต่อของเอาต์พุตแต่ละตัว จะใช้ค่าน้ำหนักชุดเดียวกัน. จำนวนค่าน้ำหนักที่ต้องการ เท่ากับ จำนวนอินพุตที่อยู่ในห้อง ถิน (3 ค่าในภาพตัวอย่าง).

แสดงภาพของข้อมูลที่มีโครงสร้างมิติที่ซับซ้อนทำได้ยาก. อย่างไรก็ตาม รูปที่ 6.7 แสดงตัวอย่างสำหรับ ทั้งข้อมูลหนึ่งลำดับชั้น และข้อมูลสองลำดับชั้น.

หมายเหตุ แนวคิดของชั้นคอนโวลูชันนั้นมีความทั่วไปมากพอ ที่จะนำไปปรับใช้กับลักษณะข้อมูลที่ อาจมีโครงสร้างมิติแบบต่าง ๆ ได้. แต่เพื่อความกระชับ หัวข้อนี้อภิปรายโครงข่ายคอนโวลูชัน สำหรับ ข้อมูลภาพ ซึ่งมักมีโครงสร้างมิติสำหรับหนึ่งภาพคือ $C \times H \times W$ เมื่อ C แทนจำนวนลักษณะอิสระ เช่น จำนวนช่องสี และ H กับ W แทนขนาดของชุดลำดับมิติ (ขนาดของภาพ ได้แก่ ความสูงกับความ กว้าง). กรณีเช่นนี้ มักถูกเรียกว่า คอนโวลูชันสองมิติ (*two-dimension convolution*) สำหรับสองชุด มิติลำดับสัมพันธ์ ซึ่งหมายกับข้อมูล เช่น ภาพ. สังเกตว่า แต่ภาพ อาจแทนด้วย เทคนิคสามลำดับ ชั้น $\mathbf{X} \in \mathbb{B}^{C \times H \times W}$ แต่ความสัมพันธ์เชิงลำดับมีเฉพาะชุดมิติที่สองและสาม. ชุดมิติแรก (ชุดมิติของ สี) ไม่มีความสัมพันธ์เชิงลำดับ.

คอนโวลูชันหนึ่งมิติ (*one-dimension convolution*) สำหรับหนึ่งชุดมิติลำดับ จะหมายกับข้อมูล เช่น เสียง โดย ชุดมิติที่มีลำดับ คือ เวลา. คอนโวลูชันสามมิติ (*three-dimension convolution*) จะ หมายกับข้อมูล เช่น วิดีโอ โดย ชุดมิติที่มีลำดับ คือ ชุดมิติพิกเซลแนวตั้ง, ชุดมิติพิกเซลแนวอน, และ ชุดมิติเวลา.



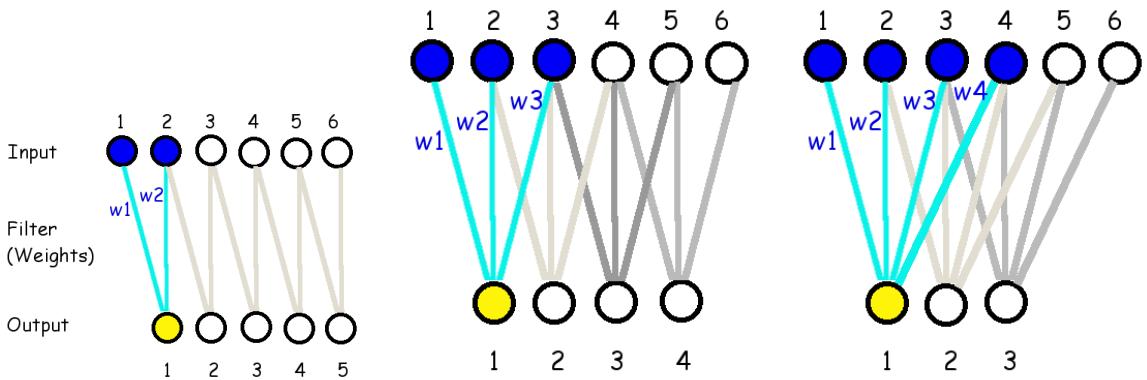
รูปที่ 6.3 แผนภาพการเชื่อมต่อของชั้นconvโวโลชั่น (convolution layer) เมื่ออินพุตมีโครงสร้างชุดมิติเดียวกับ $\mathbf{x} \in \mathbb{R}^H$ เมื่อ H เป็นขนาดชุดลำดับมิติ (ในรูป $H = 5$). เอาร์พุตแต่ละตัวเชื่อมต่อกับอินพุตจำนวนจำกัด (ในรูป ขนาดจำกัดที่ 3 ตัว) และใช้ค่าน้ำหนักร่วม (w_1, w_2, w_3 ในรูปเขียน w_1 , w_2 , และ w_3 ตามลำดับ) นั่นคือ convโวโลชั่นเอาร์พุต $a_i = w_1 \cdot x_i + w_2 \cdot x_{i+1} + w_3 \cdot x_{i+2} + b$ เมื่อ $i = 1, 2, 3$ และ b แทนค่าไบอัส (ไม่ได้แสดงในภาพ). ภาพซ้ายเน้นการเชื่อมต่อของ a_1 ภาพกลาง a_2 และภาพขวา a_3 .

6.1 ชั้นconvโวโลชั่น

ชั้นconvโวโลชั่น (convolution layer) เป็นชั้นคำนวณ ที่มีกลไกการเชื่อมต่อท้องถิ่น, การใช้ค่าน้ำหนักร่วม และการรักษาโครงสร้างมิติของเอาร์พุต. รูปที่ 6.3 แสดงกลไกของการเชื่อมต่อท้องถิ่น และการใช้ค่าน้ำหนักร่วม. ค่าเอาร์พุตของชั้น $z_i = h(a_i)$ โดย $h(\cdot)$ แทนฟังก์ชันกระตุ้น และ convโวโลชั่นเอาร์พุต $a_i = w_1 \cdot x_i + w_2 \cdot x_{i+1} + w_3 \cdot x_{i+2} + b$ 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 เน้นให้เห็นการคำนวณสำหรับ 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).

สังเกตว่า (1) ค่าน้ำหนักร w_1, w_2 , และ w_3 ถูกใช้ร่วมกัน (การใช้ค่าน้ำหนักร่วม) และจำนวนค่าน้ำหนักร คือ 3. (2) แม้ว่าไบอัสไม่ได้แสดงในภาพ แต่ควรบันทึกไว้ว่า ค่าไบอัสก็ใช้ร่วมกัน นั่นคือ ทั้ง a_1, a_2, a_3 ถ้าใช้ไบอัส b ค่าเดียวกัน. (3) เอาร์พุตแต่ละตัว เชื่อมต่อกับอินพุตจำนวนจำกัด ซึ่งจำนวนจะเท่ากับจำนวนค่าน้ำหนักร (การเชื่อมต่อท้องถิ่น). (4) อินพุตขนาดเป็น 5 แต่เอาร์พุตมีขนาดเป็น 3. ถ้าใช้เอาร์พุตขนาดน้อยกว่า 3 จะไม่สามารถครอบคลุมอินพุตได้ครบถ้วน. ถ้าใช้เอาร์พุตขนาดมากกว่า 3 เอาร์พุตตัวที่สี่ ตัวที่ห้า และตัวที่หก จะมีอินพุตไม่ครบ.

เนื่องจากประวัติการพัฒนาชั้นconvโวโลชั่น มาจากกลุ่มงานทางด้านการประมวลผลภาพ ค่าน้ำหนักร่วมเหล่านี้ มักถูกเรียกว่า พิลเตอร์ (filter) หรือ เครอร์เบล (kernel). จำนวนค่าน้ำหนักร ซึ่งกำหนดขนาดของอินพุตที่เชื่อมต่อกับเอาร์พุตแต่ละตัว จะถูกเรียกเป็น ขนาดของพิลเตอร์ (filter size). ขนาดของพิลเตอร์เป็นพารามิเตอร์ของแบบจำลองที่ผู้ใช้เลือก. รูปที่ 6.4 แสดงให้เห็นการเชื่อมต่อเมื่อ



รูปที่ 6.4 ขั้นตอนโวลูชันที่ใช้ฟิลเตอร์ขนาดต่าง ๆ (filter sizes). ภาพซ้าย ฟิลเตอร์ขนาด 2. ภาพกลาง ฟิลเตอร์ขนาด 3. ภาพขวา ฟิลเตอร์ขนาด 4.

ใช้ฟิลเตอร์ขนาดต่าง ๆ การคำนวณค่าคอนโวลูชันเอาร์พุตทำโดย

$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot x_{k+j-1} \quad (6.1)$$

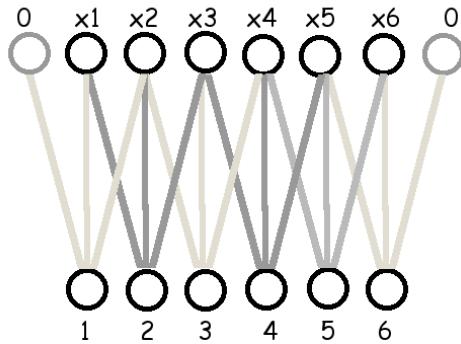
สำหรับ $k = 1, \dots, H - H_F + 1$ เมื่อ b คือค่าไบอัส. ค่าคงที่ H_F คือขนาดของฟิลเตอร์. ตัวแปร w_j คือค่าน้ำหนัก. ตัวแปร x_i คืออินพุต สำหรับ $i = 1, \dots, H$ โดย H คือขนาดของอินพุต.

สังเกต (1) ยิ่งขนาดของฟิลเตอร์ใหญ่ การเชื่อมต่อยิ่งครอบคลุมอินพุตจำนวนมากขึ้น (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 วงการการเรียนรู้ของเครื่อง จะเรียกกระบวนการดังสมการ 6.1 ว่า คอนโวลูชัน. ในขณะที่คณิตศาสตร์โดยทั่วไป และศาสตร์การประมวลผลสัญญาณ (signal processing) มักเรียกกระบวนการ เช่นนี้ ว่า สหลักษันพันธ์ข้าม (cross-correlation) และใช้คำว่า คอนโวลูชัน กับปฏิบัติการ เช่น $a_k = \sum_j w_j \cdot x_{k-j-1}$ ซึ่งมีจุดต่างสำคัญอยู่ที่การกลับลำดับของตัวถูกดำเนินการตัวหนึ่ง. เนื่องจากค่าน้ำหนักฟิลเตอร์ที่ใช้ของศาสตร์การเรียนรู้ของเครื่อง มักได้จากการวนเรียนรู้ การทำหรือไม่ทำขั้นตอนการกลับลำดับ ไม่มีผลกับผลลัพธ์สุดท้าย. ดังนั้นการตัดขั้นตอนกลับลำดับออก ช่วยให้โปรแกรมซับซ้อนน้อยลง และทำงานได้เร็วขึ้น.

การเติมเต็มด้วยศูนย์. ในทางปฏิบัติ เทคนิคการเติมเต็ม (padding) หรือเรียกว่า การเติมเต็มด้วยศูนย์ (zero-padding) มักถูกนำมาใช้ 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 $H_F = 3, \mathbf{x} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ (ขนาด $H = 6$)



รูปที่ 6.5 แผนภาพแสดงการเติมเต็มด้วยศูนย์ (zero padding). อินพุตเดิม x_1, \dots, x_6 ขนาด 6 ถูกเติมขนาดให้เป็น 8 ด้วยค่าศูนย์ทั้งสองปลาย. เอ้าต์พุตจะมีขนาด 6 (เท่ากับขนาดอินพุตด้ังเดิม)

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 ทำให้การเติมเต็มด้วยศูนย์ สามารถเติมได้อย่างสมดุลย์ทั้งสองปลาย². ขนาดของฟิลเตอร์เป็นเลขคู่ก็สามารถทำได้ เพียงแต่จะมีปลายด้านหนึ่งที่จะถูกเติมมากกว่าอีกด้านเท่านั้น. รูปที่ 6.5 แสดงแผนภาพการเชื่อมต่อ เมื่อทำการเติมเต็มด้วยศูนย์. ในภาพ ขนาดของฟิลเตอร์เป็น 3 ดังนั้นต้องเติมศูนย์ $H_F - 1 = 2$ ตำแหน่ง โดยกระจายการเติมไปทั้งสองปลาย. เมื่อมีการเติมเต็ม การคำนวณค่าคอนโวโลชันเอ้าต์พุตทำโดย

$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot \hat{x}_{k+j-1} \quad (6.2)$$

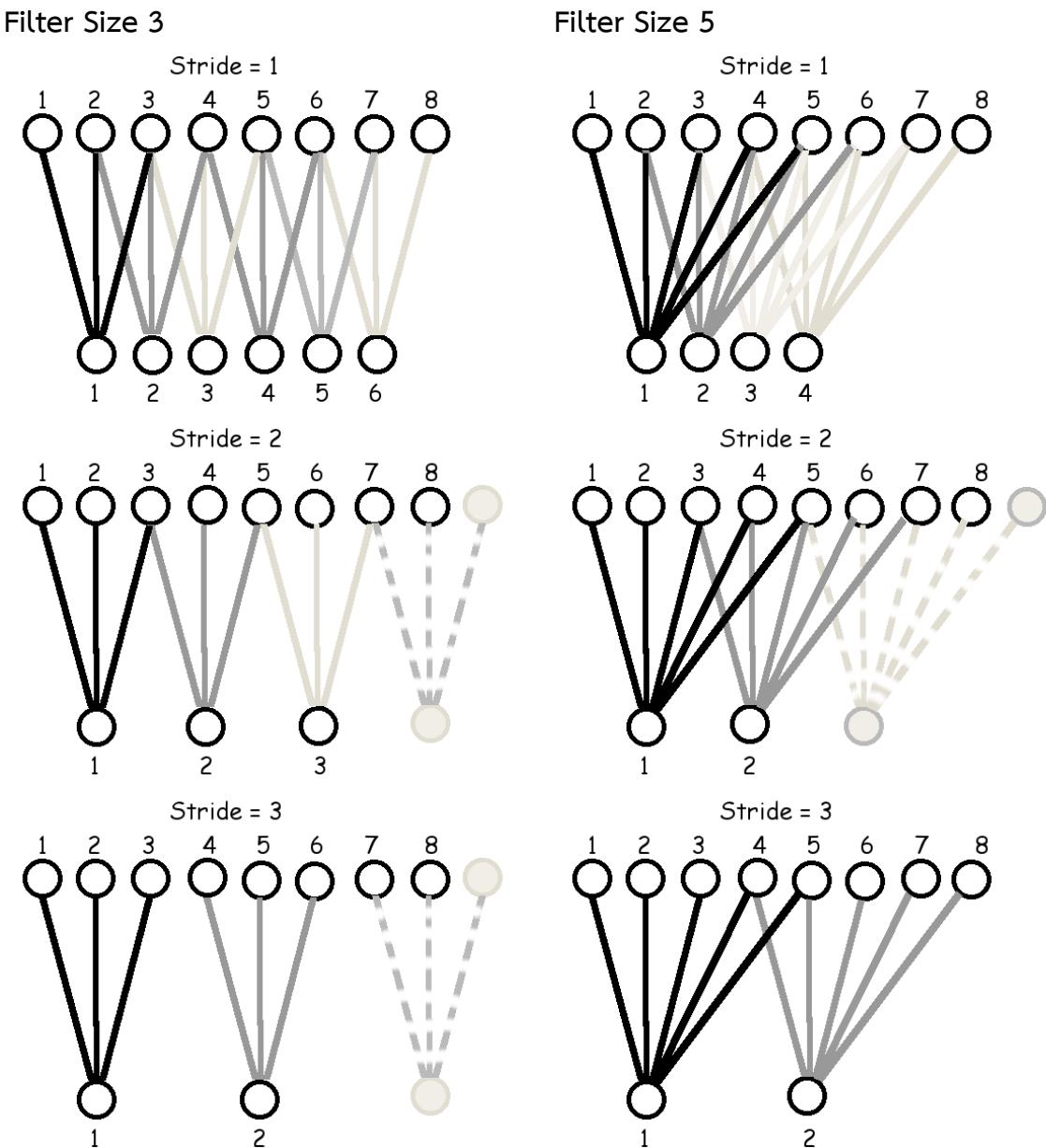
สำหรับ $k = 1, \dots, H$ และ \hat{x}_i คืออินพุตที่ถูกเติมเต็มด้วยศูนย์ สำหรับ $i = 1, \dots, H + H_F - 1$ โดย H คือขนาดของอินพุตเดิม. สำหรับกรณีขนาดฟิลเตอร์เป็นเลขคี่ ($H_F \bmod 2 = 1$)

$$\hat{x}_i = \begin{cases} 0, & \text{สำหรับ } i \leq \frac{H_F-1}{2} \text{ หรือ } i > \frac{H_F-1}{2} + H, \\ x_{i-(H_F-1)/2}, & \text{นอกเหนือจากข้างต้น.} \end{cases} \quad (6.3)$$

โดย $i = 1, \dots, H + H_F - 1$.

ขนาดก้าวย่าง. ตัวอย่างข้างต้น เอ้าต์พุตแต่ละหน่วยจะรับอินพุตต่างจากเอ้าต์พุตหน่วยข้าง ๆ โดยขยับลำดับอินพุตไปแค่หนึ่งตำแหน่ง. การขยับตำแหน่งของอินพุตสำหรับเอ้าต์พุตหน่วยข้าง ๆ กันนี้ ไม่

²หากให้ฟิลเตอร์ขนาดเป็นเลขคี่ และก้าวย่างเป็นหนึ่ง จะทำให้จำนวนศูนย์ที่ต้องเติมเป็นเลขคู่. ดูสมการ 6.5 เพิ่มเติม.



รูปที่ 6.6 ขนาดก้าวย่าง. ชั้นคอนโวลูชันที่ใช้ฟิลเตอร์ขนาด 3 (ภาพด้านซ้าย) และ ฟิลเตอร์ขนาด 5 (ภาพด้านขวา) โดยแต่ละฟิลเตอร์ใช้กับ กับก้าวย่าง (strides) ขนาด 1, 2, และ 3 จากบนลงล่าง. วงกลมที่อยู่ด้านบนของภาพ แทน หน่วยอินพุต. วงกลมที่อยู่ด้านล่างของภาพ แทน หน่วยเอาต์พุต. เส้นตรงทึบ (เอดสี ทำเพื่อให้มองเห็นได้ชัดเจนเท่านั้น) แสดงการเชื่อมต่อ. วงกลมสีเทาและเส้นประ เน้นความสัมพันธ์ของการเติมเต็ม ขนาดฟิลเตอร์ ขนาดย่างก้าว และความครอบคลุมอินพุต.

จำเป็นต้องจำกัดเพียง 1 ตำแหน่งเท่านั้น. การขยับตำแหน่งนี้ สามารถทำทีละหลาຍ ๆ ตำแหน่ง และ การขยับตำแหน่งของเอาต์พุตหน่วยข้าง ๆ กันนี้ จะเรียกว่า ขนาดก้าวย่าง (stride). รูปที่ 6.6 แสดง การเชื่อมต่อ เมื่อใช้ฟิลเตอร์ขนาด 3 และ 5 กับขนาดก้าวย่างต่าง ๆ.

หากอินพุตที่เติมมีขนาด \hat{H} และฟิลเตอร์มีขนาด H_F และก้าวย่างมีขนาด S แล้วขนาดของเอาต์-

พุต H' คำนวณได้จาก

$$H' = \left\lfloor \frac{\hat{H} - H_F}{S} \right\rfloor + 1 \quad (6.4)$$

ดังนั้น หากต้องการทำการเติมเต็ม เพื่อให้ได้ขนาดของເອົາຕົ່ມທີ່ຕ້ອງການ \hat{H}' ເຮັດວຽກຄວາມ
ຂາດຂອງອິນພຸຕ໌ລັ້ງເຕີມເຕີມ \hat{H} ໄດ້ຈາກ

$$\hat{H} = S \cdot (\hat{H}' - 1) + H_F \quad (6.5)$$

ແລະ ຈຳນວນຂອງຄູນຍື່ນທີ່ຕ້ອງເຕີມໃສ່ອິນພຸຕ໌ຈະເຫັກກັບ $\hat{H} - H$.

ນັ້ນຄື່ອງ ພາຍໃນການເຕີມເຕີມ ເພື່ອໃຫ້ໄດ້ຂາດເອົາຕົ່ມທີ່ຕ້ອງກັບຂາດອິນພຸຕ໌ເຕີມ $\hat{H}' = H$ ຈະໄດ້ວ່າ
ຈຳນວນຂອງຄູນຍື່ນທີ່ຕ້ອງເຕີມໃສ່ອິນພຸຕ໌ຈະເຫັກກັບ $S \cdot (H - 1) + H_F - H$. ຜົນຍື່ອ ພາຍໃນການເຕີມເຕີມ ເພື່ອ¹
ໃຫ້ໄດ້ຂາດເອົາຕົ່ມ $\hat{H}' = \lceil \frac{H}{S} \rceil$ ຈະໄດ້ວ່າ ຈຳນວນຂອງຄູນຍື່ນທີ່ຕ້ອງເຕີມໃສ່ອິນພຸຕ໌ຈະເຫັກກັບ $S \cdot (\lceil \frac{H}{S} \rceil - 1) + H_F - H$.

ຕ້ວຍຢ່າງເຊັ່ນ ອິນພຸຕ໌ຂາດ $H = 6$ ໃຊ້ຝຶລເຕອຣ໌ຂາດ $H_F = 3$ ໃຊ້ຂາດຢ່າງກໍ່າວ $S = 1$ ແລະ
ຕ້ອງການໃຫ້ຂາດເອົາຕົ່ມທີ່ຕ້ອງກັບຂາດອິນພຸຕ໌ເຕີມ ຈະໄດ້ $\hat{H} = 1 \cdot (6 - 1) + 3 = 8$ ແລະ ຕ້ອງເຕີມຄູນຍື່ນ
2 ຕັ້ງ. ອິນພຸຕ໌ຂາດ $H = 6$ ໃຊ້ຝຶລເຕອຣ໌ຂາດ $H_F = 3$ ໃຊ້ຂາດຢ່າງກໍ່າວ $S = 2$ ແລະ ຕ້ອງການໃຫ້ຂາດ
ເອົາຕົ່ມທີ່ຕ້ອງກັບຂາດອິນພຸຕ໌ເຕີມ ຈະໄດ້ $\hat{H} = 2 \cdot (6 - 1) + 3 = 13$ ແລະ ຕ້ອງເຕີມຄູນຍື່ນ 7 ຕັ້ງ. ອິນພຸຕ໌
ຂາດ $H = 8$ ໃຊ້ຝຶລເຕອຣ໌ຂາດ $H_F = 5$ ໃຊ້ຂາດຢ່າງກໍ່າວ $S = 3$ ແລະ ຕ້ອງການໃຫ້ຂາດເອົາຕົ່ມ
ທີ່ຕ້ອງກັບຂາດອິນພຸຕ໌ເຕີມ ຈະໄດ້ $\hat{H} = 3 \cdot (8 - 1) + 5 = 26$ ແລະ ຕ້ອງເຕີມຄູນຍື່ນ 18 ຕັ້ງ.

ໜັ້ນຄອນໂວລູ້ຂັ້ນ ທີ່ມີອິນພຸຕ໌ $\mathbf{x} \in \mathbb{R}^H$ ໃຊ້ຝຶລເຕອຣ໌ $\mathbf{w} \in \mathbb{R}^{H_F}$ ມີຂາດກໍ່າວຢ່າງເປັນ S ແລະ ທຳການເຕີມ
ເຕີມດ້ວຍຄູນຍື່ນເພື່ອໃຫ້ເອົາຕົ່ມ \mathbf{a} ມີຂາດ H ສາມາຄືກຳນວນຄ່າຄອນໂວລູ້ຂັ້ນເອົາຕົ່ມແຕ່ລະຄ່າ ໄດ້ຈາກ

$$a_k = b + \sum_{j=1}^{H_F} w_j \cdot \hat{x}_{S \cdot (k-1)+j} \quad (6.6)$$

ເມື່ອ $k = 1, \dots, H$ ແລະ b ແທນຄ່າໄປອ້າສ. ສັງເກດວ່າ \mathbf{a} ຮັກຫາໂຄຮງສ້າງມິຕິຂອງ \mathbf{x} ເວັບ. ອິນພຸຕ໌ \mathbf{x} ມີລຳດັບ
ມິຕິເດືອຍ່າ ຄອນໂວລູ້ຂັ້ນເອົາຕົ່ມ \mathbf{a} ກົມື້ມີລຳດັບມິຕິເດືອຍ່າ.

ກຣນີທີ່ອົກປ່ຽນມາຂ້າງຕັ້ນ ໂຄຮງສ້າງມິຕິຂອງອິນພຸຕ໌ໄນ້ໄດ້ສັບສົນ ຄືອມີເພີ່ມລຳດັບມິຕິເດືອຍ່າ. ປັຈຈີຍ
ທີ່ສຳຄັງຕ່ອມາຂອງໜັ້ນຄອນໂວລູ້ຂັ້ນ ຄື່ອງ ກລັກໄກທີ່ສັດເຈນໃນການຈັດການກັບອິນພຸຕ໌ທີ່ມີໂຄຮງສ້າງມິຕິທີ່ສັບສົນ
ແລະ ກາຍວົມໃຫ້ເອົາຕົ່ມຮັກຫາໂຄຮງສ້າງເຊີງມິຕິບາງສ່ວນຂອງອິນພຸຕ໌ໄວ້ໄດ້. ຮູບທີ່ 6.7 ເປີຍບ່ອນໂຄຮງ-
ສ້າງມິຕິເຊີງເດືອຍ່າ (ກາພ້າຍ) ກັບໂຄຮງສ້າງມິຕິທີ່ມີ 2 ຊຸດລຳດັບ (ກາພກລາງ) ແລະ ຕ້ວອຍ່າງເຊີງຕົວເລີບ (ກາພ
ຂວາ). ຈາກຕ້ວອຍ່າງເຊີງເລີບໃນຮູບຈະເຫັນວ່າ

$$\begin{aligned} \text{เอาต์พุต } a_{1,1} &= 2.7 + (1) \cdot 89 + (2) \cdot 84 + (1.5) \cdot 105 \\ &\quad + (2.5) \cdot 6 + (8) \cdot 0 + (2.3) \cdot 4 \\ &\quad + (1.6) \cdot 13 + (1.8) \cdot 52 + (1.4) \cdot 57 \\ &= 635.6 \end{aligned}$$

ซึ่งเอาต์พุตคำแห่ง $(1, 1)$ ถูกเน้นในภาพขาว.

$$\begin{aligned} \text{เอาต์พุต } a_{1,2} &= 2.7 + (1) \cdot 84 + (2) \cdot 105 + (1.5) \cdot 116 \\ &\quad + (2.5) \cdot 0 + (8) \cdot 4 + (2.3) \cdot 20 \\ &\quad + (1.6) \cdot 52 + (1.8) \cdot 57 + (1.4) \cdot 14 \\ &= 754.1 \end{aligned}$$

...

$$\begin{aligned} \text{เอาต์พุต } a_{2,1} &= 2.7 + (1) \cdot 6 + (2) \cdot 0 + (1.5) \cdot 4 \\ &\quad + (2.5) \cdot 13 + (8) \cdot 52 + (2.3) \cdot 57 \\ &\quad + (1.6) \cdot 50 + (1.8) \cdot 58 + (1.4) \cdot 60 \\ &= 862.7 \end{aligned}$$

เป็นต้น.

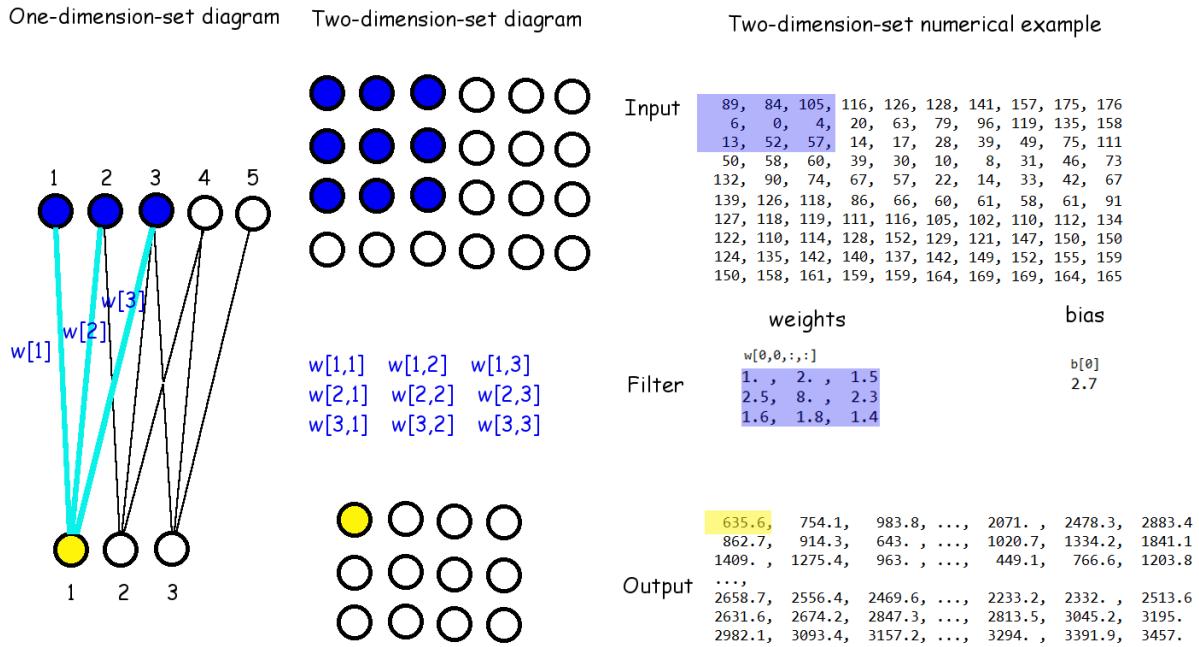
สังเกตว่า (1) รูปที่ 6.7 ไม่มีการทำการเติมเต็มด้วยศูนย์. (2) ขนาดก้าวย่างเป็น 1×1 . นั่นคือ ขับตามแนวตั้งที่ละหนึ่งหน่วยอินพุต (พิกเซล) และขับตามแนวอนก์ที่ละหนึ่งหน่วยอินพุต. (3) คอนโวลูชันเอาต์พุตมี 2 ลำดับมิติ เช่นเดียวกับอินพุต แม้จะขนาดเล็กกว่า เพราะไม่ได้ทำการเติมเต็ม. และ เพื่อเน้นถึงโครงสร้างมิติที่รักษาไว้ นี้ คอนโวลูชันเอาต์พุต อาจจะถูกเรียกว่า แผนที่คอนโวลูชัน (*convolution map*) เพื่อกันการสับสนกับเอาต์พุตสุดท้ายของโครงข่ายของโครงข่ายทั้งหมด ซึ่งเอาต์พุตสุดท้าย คือ เอาต์พุตของชั้นสุดท้ายของโครงข่าย.

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) คอนโวลูชันเอาต์พุต $\mathbf{A} \in \mathbb{R}^{H \times W}$ สามารถคำนวณได้จาก

$$a_{k,l} = b + \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{ij} \cdot \hat{x}_{S_H \cdot (k-1) + i, S_W \cdot (l-1) + j} \quad (6.7)$$

$k = 1, \dots, H, l = 1, \dots, W$ โดย \hat{x} คืออินพุตที่ผ่านการเติมเต็ม และ b คือไบอัส.

อย่างไรก็ตาม เมื่อพิจารณารูปที่ 6.1 จะเห็นว่า ข้อมูลภาพสีหนึ่งภาพมีโครงสร้างมิติเป็น 3 ชุด มิติ ได้แก่ ชุดสำหรับซองสีขนาด 3, ชุดลำดับแนวตั้งขนาด 133 และชุดลำดับสำหรับแนวอนขนาด 175. นั่นคือ มีอินพุต $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$. โดยทั่วไป ชั้นคอนโวลูชัน จะเลือกจัดการกับชุดมิติซองสีเป็น เสมือนชุดมิติที่ไม่มีลำดับ แต่ชุดลำดับแนวตั้งและแนวอนแบบชุดมิติมีลำดับ. การคำนวณคอนโวลูชัน เอาต์พุต $\mathbf{A} \in \mathbb{R}^{H \times W}$ สำหรับกรณีนี้ (โครงสร้างมีชุดมิติที่มีลำดับ และไม่มีลำดับ) จะทำดังสมการ 6.8

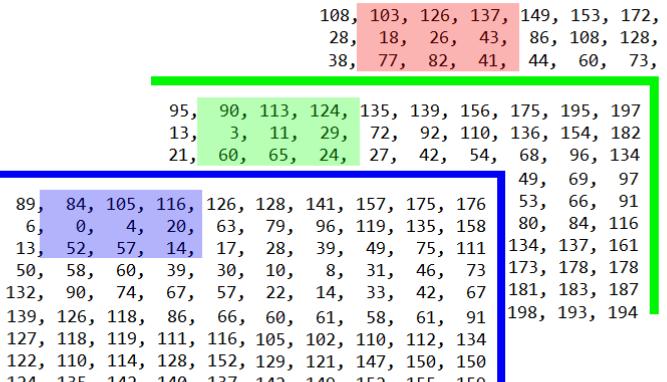
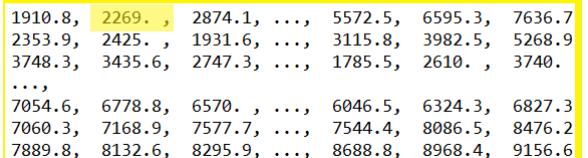


รูปที่ 6.7 แผนภาพแสดงการคำนวณชั้นคอนโวโลชัน. อินพุตแสดงอยู่ด้านบน. การเข้ามต่อ (ค่าน้ำหนัก) แสดงตรงกลาง. และคอนโวโลชันเอาร์พุตแสดงอยู่ด้านล่าง. ภาพซ้ายสุด แสดงแผนภาพการเข้ามต่อของชั้นคอนโวโลชัน เมื่ออินพุตมีชุดลำดับ มิติเดียว $\mathbf{x} \in \mathbb{R}^H$ เมื่อ H เป็นขนาดชุดลำดับมิติ (ในรูป $H = 5$). ภาพกลาง แสดงแผนภาพของชั้นคอนโวโลชัน เมื่ออินพุต มีโครงสร้างชุดลำดับมิติสองชุด $\mathbf{x} \in \mathbb{R}^{H \times W}$ เมื่อ H และ W เป็นขนาดชุดลำดับมิติ (ในรูป สูง $H = 4$ กว้าง $W = 6$). ภาพกลางนี้จะแสดงการคำนวณเมื่อต้องคำนึงถึงความไม่ยุ่งเหงาย. เอาร์พุตแต่ละตัวเข้ามต่อ กับ อินพุตจำนวนจำกัด (ในรูป ขนาดจำกัดที่ 3×3 ตัว) และใช้ค่าน้ำหนักร่วม 9 ตัวดังแสดง. ภาพขวา แสดงตัวอย่างเชิงตัวเลข เมื่ออินพุต (ด้านบน) มีโครงสร้างชุดมิติ สองชุด ขนาด 10×10 . ตระกูลภาพ แสดงค่าน้ำหนัก หรือฟิลเตอร์ ขนาด 3×3 และค่าไบอส. คอนโวโลชันเอาร์พุต แสดงด้านล่าง. ไม่มีการเติมเต็มด้วยศูนย์ และใช้ค่าก้าวย่างเป็น 1×1 .

$$a_{k,l} = b + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{cij} \cdot \hat{x}_{c,S_H \cdot (k-1)+i, S_W \cdot (l-1)+j} \quad (6.8)$$

$k = 1, \dots, H$ 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.

รูปที่ 6.8 แสดงการคำนวณคอนโวโลชัน เมื่ออินพุตมีโครงสร้าง 3 ชุดมิติ โดยชุดมิติแรกไม่มีลำดับ. รูปเน้นให้เห็นความสัมพันธ์ระหว่างเอาร์พุต $a_{1,2}$ ที่เข้ามต่อ กับ อินพุต $x_{1,1,2}, \dots, x_{1,3,4}$ (ซองสีน้ำเงิน), $x_{2,1,2}, \dots, x_{2,3,4}$ (ซองสีเขียว), $x_{3,1,2}, \dots, x_{3,3,4}$ (ซองสีแดง) ดังที่เน้นคำในภาพ. จากตัวอย่าง เชิงเลขในรูปจะเห็นว่า

Input 	weights $w[0,2,:,:]$ 0.8, 1., 0.6 $w[0,1,:,:]$ 0.9, 2.3, 1.3 $w[0,0,:,:]$ 1.3, 2.1, 1.7 2.2, 6.4, 3.2 1.5, 2.1, 1.1 $w[0,0,:,:]$ 1., 2., 1.5 2.5, 8., 2.3 1.6, 1.8, 1.4 bias 2.7	Convolution Result of Filter 0 

รูปที่ 6.8 ตัวอย่างการคำนวณค่าคอนโวลูชันในข้อมูลที่มีโครงสร้างมิติ ณ ตำแหน่ง $(1, 2)$. อินพุตมีโครงสร้างมิติที่ซับซ้อน นั่นคือ มี 3 ช่องสี และแต่ละช่องสีมีความเข้มพิกเซลตามลำดับแนวตั้งและแนวนอน. พิลเตอร์ขนาด 3×3 ที่ใช้ จะต้อง มีชุดมิติแรกขนาดเป็น 3 เช่นกัน (รับกับ 3 ช่องสีของอินพุต). เอ้าต์พุตจะรักษาโครงสร้างมิติเดิมที่มีขนาด $C \times H \times W$ จะแปลงมาเป็น เอ้าต์พุตขนาด $(H - 2) \times (W - 2)$ เนื่องจากขนาดพิลเตอร์เป็น 3×3 ขนาดก้าวย่าง เป็น 1×1 และไม่มีการเติมเต็ม.

$$\begin{aligned}
 \text{เอ้าต์พุต } a_{1,2} &= 2.7 && (\text{เบื้องต้น}) \\
 &+ (1) \cdot 84 + (2) \cdot 105 + (1.5) \cdot 116 \\
 &+ (2.5) \cdot 0 + (8) \cdot 4 + (2.3) \cdot 20 \\
 &+ (1.6) \cdot 52 + (1.8) \cdot 57 + (1.4) \cdot 14 && (\text{ช่องสีน้ำเงิน รวม } 751.4) \\
 &+ (1.3) \cdot 90 + (2.1) \cdot 113 + (1.7) \cdot 124 \\
 &+ (2.2) \cdot 3 + (6.4) \cdot 11 + (3.2) \cdot 29 \\
 &+ (1.5) \cdot 60 + (2.1) \cdot 65 + (1.1) \cdot 24 && (\text{ช่องสีเขียว รวม } 987.8) \\
 &+ (0.8) \cdot 103 + (1) \cdot 126 + (0.6) \cdot 137 \\
 &+ (0.9) \cdot 18 + (2.3) \cdot 26 + (1.3) \cdot 43 \\
 &+ (0.4) \cdot 77 + (0.8) \cdot 82 + (0.2) \cdot 41 && (\text{ช่องสีแดง รวม } 527.1) \\
 &= 2269.0
 \end{aligned}$$

เป็นต้น.

กลไกของคอนโวลูชันที่กล่าวมา เป็นเพียงการคำนวณค่าของคอนโวลูชันเอ้าต์พุต การใช้งานซึ่ง

คอนโวโลชันในโครงข่ายแบบลีกันน์ หลังจากทำการคำนวนคอนโวโลชัน (ผลกระทบเชิงเส้นระหว่างอินพุต กับค่าหน้าหนัก) และ ค่าที่ได้จะนำไปผ่านฟังก์ชันกราฟตุ้น เพื่อให้ได้ผลลัพธ์ของชั้นคอนโวโลชัน ดังแสดงในสมการ 6.9,

$$z_{k,l} = h(a_{k,l}) \quad (6.9)$$

เมื่อ $h(\cdot)$ คือฟังก์ชันการกราฟตุ้น (เช่น เรคติไฟเดลีเนียร์, สมการ 5.1) และ $a_{k,l}$ คือคอนโวโลชันเอาร์พุต (สมการ 6.8). เอาร์พุต $\mathbf{Z} \in \mathbb{R}^{H \times W}$ นี้ อาจถูกเรียกว่า แผนที่ลักษณะสำคัญ (feature map).

การใช้กลไกของการเชื่อมต่อห้องถิน ร่วมกับการใช้ค่าหน้าหนักร่วม จะเปรียบเสมือนการทำเทคนิคหน้าต่างเลื่อน (หัวข้อ 4.1) เพื่อค้นหารูปแบบ ที่อาจปรากฏอยู่ตำแหน่งต่าง ๆ ในอินพุตได้. นั่นคือ เมื่อมองกับการเลื่อนฟิลเตอร์ (เทียบเท่ากับหน้าต่าง) ไปตำแหน่งต่าง ๆ ของอินพุต เพื่อค้นหารูปแบบของลักษณะที่สำคัญ. การใช้ฟิลเตอร์หนึ่งตัว ก็เปรียบเสมือน รูปแบบของลักษณะสำคัญหนึ่งรูปแบบ. การใช้งานโครงข่ายคอนโวโลชัน ในทางปฏิบัติ จะใช้ฟิลเตอร์หลาย ๆ ตัว ซึ่งเปรียบเสมือน การค้นหารูปแบบของลักษณะสำคัญหลาย ๆ รูปแบบ และ ผลลัพธ์ ก็จะได้แผนที่ลักษณะสำคัญหลาย ๆ แผนที่.

ทบทวนการคำนวนชั้นคอนโวโลชัน สำหรับอินพุต $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ และฟิลเตอร์ขนาด $H_F \times W_F$ ทั้งหมด F ตัว นั่นคือ $\mathbf{W} \in \mathbb{R}^{F \times C \times H_F \times W_F}$ และเอาร์พุต $\mathbf{Z} \in \mathbb{R}^{F \times H \times W}$ หาได้จาก

$$a_{f,k,l} = b_f + \sum_{c=1}^C \sum_{i=1}^{H_F} \sum_{j=1}^{W_F} w_{fcij} \cdot \hat{x}_{c,S_H \cdot (k-1)+i, S_W \cdot (l-1)+j} \quad (6.10)$$

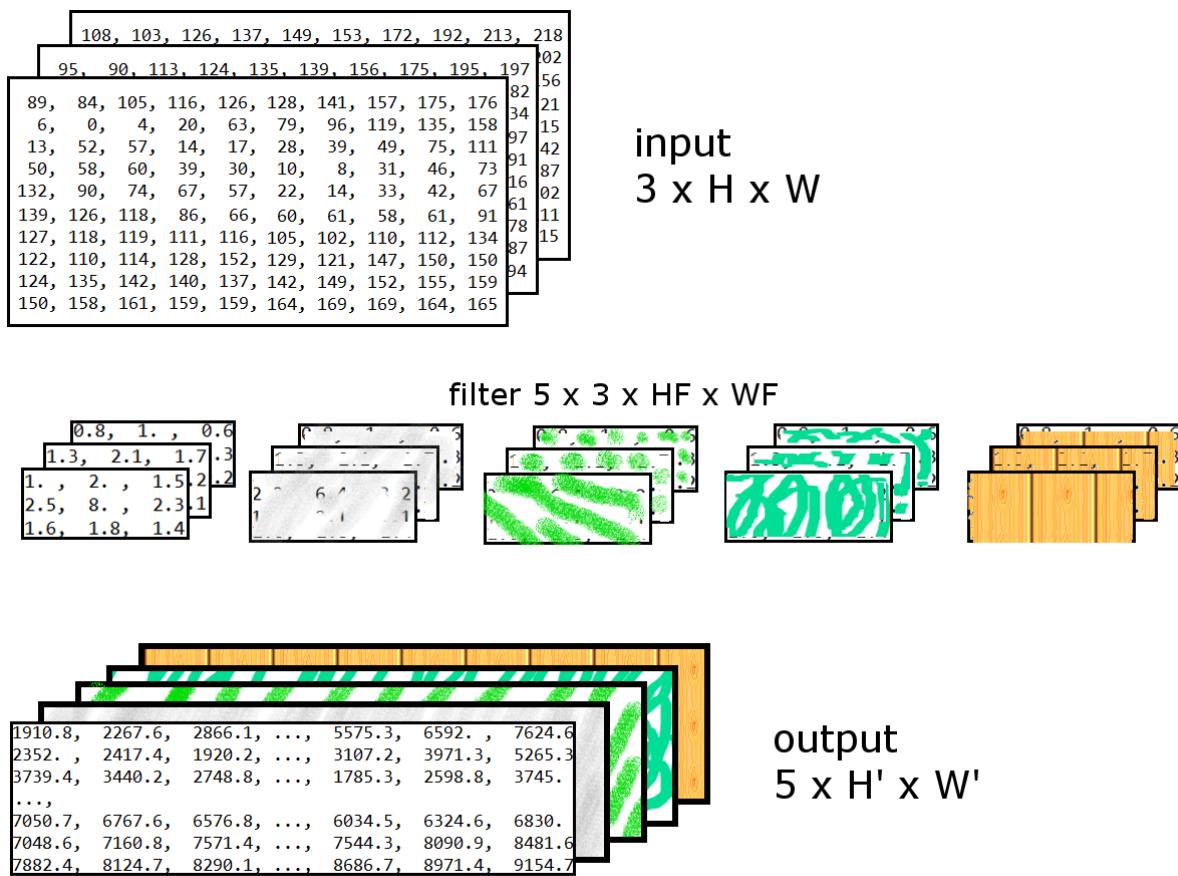
$$z_{f,k,l} = h(a_{f,k,l}) \quad (6.11)$$

$f = 1, \dots, F$, $k = 1, \dots, H$ และ $l = 1, \dots, W$ เมื่อ S_H , S_W คือขนาดก้าวย่าง ตามแนวตั้ง และแนวนอน ตามลำดับ. ฟังก์ชัน $h(\cdot)$ คือ ฟังก์ชันกราฟตุ้น, ตัวแปร b_f คือใบอัสร่วมของฟิลเตอร์ f ส่วนตัวแปร \hat{x} คืออินพุตที่ผ่านการเติมเต็ม.

รูปที่ 6.9 แสดงแผนภาพโครงสร้างมิติของอินพุต ฟิลเตอร์ และเอาร์พุต ของชั้นคอนโวโลชัน. ในรูป ใช้ 5 ฟิลเตอร์ และผลลัพธ์ก็จะได้แผนที่ลักษณะ 5 แผนที่. แต่ละแผนที่ระบุการตอบสนองต่อรูปแบบสำคัญที่ตำแหน่งต่าง ๆ สำหรับแต่ละรูปแบบสำคัญ. สังเกตว่าเอาร์พุตของชั้นคอนโวโลชัน $\mathbf{Z} \in \mathbb{R}^{F \times H' \times W'}$ ซึ่งมีโครงสร้างมิติเป็น 3 ชุด โดยชุดแรกไม่มีลำดับ สองชุดหลังเป็นชุดลำดับที่สัมพันธ์กัน เช่นเดียวกับอินพุต ดังนั้น เอาร์พุตจากชั้นคอนโวโลชันนั้น สามารถส่งต่อไปเป็นอีกชั้นหนึ่งได้โดยไม่ต้องมีการดัดแปลงใด ๆ.

สนามรับรู้. สนามรับรู้ (receptive field) สำหรับโครงข่ายคอนโวโลชัน คือบริเวณพื้นที่ห้องถินของอินพุต ที่หน่วยอยู่ที่สนใจครอบคลุมถึง. ขนาดของสนามรับรู้ สามารถคำนวณได้จาก

$$R_k = 1 + \sum_{j=1}^k (F_j - 1) \prod_{i=0}^{j-1} S_i \quad (6.12)$$



รูปที่ 6.9 โครงสร้างมิติของอินพุต พิลเตอร์ และเอาต์พุตของชั้นคอนโวลูชัน. มีพิลเตอร์ขนาด $HF \times WF$ อยู่ 5 ตัว ดังนั้น เอาต์พุตซึ่งเป็นแผนที่ลักษณะ ก็จะมีอยู่ 5 แผนที่ แต่ละแผนที่ลักษณะคำนวนจากพิลเตอร์แต่ละตัว.

เมื่อ R_k เป็นขนาดของสนามรับรู้ แล้ว F_j เป็นขนาดพิลเตอร์ของชั้นที่ j และ S_i เป็นขนาดก้าวย่าง ของชั้นที่ i และกำหนดให้ $S_0 = 1$.

ตัวอย่างเช่น โครงข่ายคอนโวลูชันสองชั้น ที่ชั้นแรกใช้พิลเตอร์ขนาด 3×3 ก้าวย่างเป็น 1 และ ชั้นที่สองก็ใช้พิลเตอร์ 3×3 และก้าวย่างเป็น 1 แล้ว แต่ละหน่วยอยู่ในชั้นที่สอง จะครอบคลุมพื้นที่ ขนาด 3×3 ในชั้นที่หนึ่ง และจะครอบคลุมพื้นที่ขนาด 5×5 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) เพื่อลดจำนวน ข้อมูลลง และ (2) การสรุปลักษณะสำคัญในรูปแบบไกล์เดียง. กลไกการทำงานของชั้นดึงรวมจะคล้าย กับชั้นคอนโวลูชันที่จะขยับไปทีละก้าวย่าง เพื่อประมวลผลอินพุตในบริเวณจำกัด (การเข้ามต่อท้องถิ่น) แต่การประมวลของชั้นดึงรวมจะแยกมิติที่ไม่มีลำดับออกจากกัน ไม่นำมาประมวลรวมกัน. นั่นคือ

อินพุต $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ โดย C คือชุดมิติที่ไม่มีลำดับ และ H และ W คือชุดมิติที่มีลำดับสัมพันธ์. เอ้าต์พุตของชั้นดึงรวม $\mathbf{Z} \in \mathbb{R}^{C \times H' \times W'}$ 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 พังก์ชันดึงรวม, และ \hat{x} คืออินพุตที่ผ่านการเติมเต็ม เพื่อให้เอ้าต์พุตครอบคลุมอินพุตทุกตัว. สังเกตว่า การดึงรวมจะไม่ยุบชุดมิติอิสระ C . เอ้าต์พุตที่ได้ยังคงมีขนาด C สำหรับชุดมิติแรกเช่นเดิม.

การทำการเติมเต็มสำหรับชั้นดึงรวม จะมีจุดประสงค์ต่างจากสำหรับชั้นคอนโวลูชัน ตรงที่ ชั้นดึงรวมไม่ต้องการรักษาขนาดชุดมิติของอินพุตไว้ ชั้นดึงรวมต้องการลดขนาดชุดมิติลำดับนี้ลง ($H' < H$ และ $W' < W$) แต่การเติมเต็มสำหรับชั้นดึงรวม ทำเพื่อให้เอ้าต์พุตสามารถครอบคลุมอินพุตได้ทุกตัว. ตัวอย่างเช่น ชั้นดึงรวมแบบมากที่สุด (*max-pooling Layer*) ขนาด 5×5 ใช้ก้าวย่างขนาด 2×2 หากอินพุต \mathbf{X} มีขนาด $3 \times 60 \times 80$ จะพบว่า หากไม่ทำการเติมเต็ม อินพุตตัวสุดท้ายของแต่ละชุดมิติ อิสระ $\{x_{1,60,80}, x_{2,60,80}, x_{3,60,80}\}$ จะถูกรับผิดชอบโดยเอ้าต์พุต ที่ตำแหน่ง $(29, 39)$ ซึ่ง (จากสมการ 6.13),

$$\begin{aligned} z_{c,29,39} &= \max\{\hat{x}_{c,56+i,76+j}\}_{i=1, \dots, 5, j=1, \dots, 5} \\ &= \max \left\{ \begin{array}{lllll} \hat{x}_{c,57,77}, & \hat{x}_{c,57,78}, & \hat{x}_{c,57,79}, & \hat{x}_{c,57,80}, & \hat{x}_{c,57,81}, \\ \hat{x}_{c,58,77}, & \hat{x}_{c,58,78}, & \hat{x}_{c,58,79}, & \hat{x}_{c,58,80}, & \hat{x}_{c,58,81}, \\ \hat{x}_{c,59,77}, & \hat{x}_{c,59,78}, & \hat{x}_{c,59,79}, & \hat{x}_{c,59,80}, & \hat{x}_{c,59,81}, \\ \hat{x}_{c,60,77}, & \hat{x}_{c,60,78}, & \hat{x}_{c,60,79}, & \hat{x}_{c,60,80}, & \hat{x}_{c,60,81}, \\ \hat{x}_{c,61,77}, & \hat{x}_{c,61,78}, & \hat{x}_{c,61,79}, & \hat{x}_{c,61,80}, & \hat{x}_{c,61,81} \end{array} \right\}, \end{aligned}$$

$c = 1, 2, 3$ ต้องการค่าอินพุตที่ตำแหน่งแนวนอนที่ 61 และตำแหน่งแนวนอนที่ 81 ซึ่งอินพุตตั้งเดิม นั้นไม่มี. ดังนั้น \hat{x} จึงต้องมีการเติมเต็มในตำแหน่งตั้งกล่าว เพื่อให้การดึงรวมสามารถครอบคลุมอินพุต ตั้งเดิมได้ทั้งหมด.

แต่หากว่าการดึงรวมสามารถครอบคลุมอินพุตตั้งเดิมได้ทั้งหมดแล้วก็ไม่จำเป็นต้องมีการเติมเต็ม. ตัวอย่างเช่น ชั้นดึงรวมแบบมากที่สุด ขนาด 3×3 ใช้ก้าวย่างขนาด 3×3 (ไม่มีการซ้อนทับ) หาก อินพุต \mathbf{X} มีขนาด $3 \times 90 \times 120$ จะพบว่า แม้ไม่ทำการเติมเต็ม อินพุตตัวสุดท้ายของแต่ละชุดมิติ อิสระ $\{x_{1,90,120}, x_{2,90,120}, x_{3,90,120}\}$ จะถูกรับผิดชอบโดยเอ้าต์พุต ที่ตำแหน่ง $(30, 40)$ พอดี โดยไม่ต้องการการเติมเต็มเพิ่ม,

$$\begin{aligned} z_{c,30,40} &= \max\{\hat{x}_{c,87+i,3 \cdot 117+j}\}_{i=1, \dots, 3, j=1, \dots, 3} \\ &= \max \left\{ \begin{array}{lll} \hat{x}_{c,88,118}, & \hat{x}_{c,88,119}, & \hat{x}_{c,88,120}, \\ \hat{x}_{c,89,118}, & \hat{x}_{c,89,119}, & \hat{x}_{c,89,120}, \\ \hat{x}_{c,90,118}, & \hat{x}_{c,90,119}, & \hat{x}_{c,90,120} \end{array} \right\}. \end{aligned}$$

นั่นคือ หาก $(H - H_F) \bmod S_H = 0$ และ $(W - W_F) \bmod S_W = 0$ ก็ไม่จำเป็นต้องทำการเติมเต็ม.

เมื่อทำการเติมเต็มตามความจำเป็น 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 2×2 ก้าวย่างขนาด 2×2 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 ขนาดแผนที่เอาร์พุตจะลดลงเหลือครึ่งหนึ่งของขนาดแผนที่อินพุต

ฟังก์ชันดึงรวม นิยมใช้ ฟังก์ชันทางสถิติ อาทิ ฟังก์ชันดึงรวมแบบมากที่สุด และ ฟังก์ชันดึงรวมแบบเฉลี่ย (average pooling) เป็นต้น. การทำงานของฟังก์ชันเหล่านี้ก็ตรงไปตรงมา คือ ฟังก์ชันดึงรวมแบบมากที่สุด จะให้ค่าที่มากที่สุดในเซตของค่า ฟังก์ชันดึงรวมแบบเฉลี่ย จะให้ค่าเฉลี่ยของค่าต่าง ๆ ในเซตของค่า.

6.3 เกรเดียนต์ของโครงข่ายคอนโวโลชัน

ในการฝึกโครงข่าย ขั้นตอนวิธีที่ใช้ อาศัยค่าเกรเดียนต์ (ดูหัวข้อ 3.3). การใช้งานชั้นคอนโวโลชัน และชั้นดึงรวม ก็มีผลต่อเกรเดียนต์. นอกจากนั้น ชั้นคอนโวโลชันเองก็มีค่าน้ำหนักที่ต้องการการฝึกด้วย.

เกรเดียนต์ของชั้นคอนโวโลชัน

เปรียบเทียบกับหัวข้อ 3.3 พิจารณาความสัมพันธ์ ระหว่างฟังก์ชันเป้าหมายและชั้นคอนโวโลชัน ฟังก์ชันเป้าหมาย E_n จะขึ้นกับค่าน้ำหนัก w_{fcij} (สมการ 6.10 และ 6.11) ผ่านคอนโวโลชันเอาร์พุต a_{fkl} สำหรับทุก ๆ k และ l ที่ใช้ w_{fcij} ร่วม. ดังนั้นจากกฎของ

$$\frac{\partial E_n}{\partial w_{fcij}} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \frac{\partial E_n}{\partial a_{fkl}} \frac{\partial a_{fkl}}{\partial w_{fcij}} \quad (6.16)$$

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 อินพุตของชั้นคอนโวโลชัน ที่ผ่านการเติมเต็ม, S_H และ S_W เป็น
ค่าก้าวบ่างตามแนวตั้งและนอน.

เมื่อเปรียบเทียบกับสมการ 3.28 จะเห็นว่าคล้ายกันมาก ต่างกันเพียง (1) สมการ 6.19 มีการบวก
พจน์ต่าง ๆ ที่ใช้ค่าน้ำหนักร่วมกัน (2) ค่าน้ำหนักไม่ได้เฉพาะเจาะจงกับเอาร์พุต (เพราะใช้ค่าน้ำหนัก
ร่วมกัน), (3) จำนวนค่าน้ำหนักมีน้อยเมื่อเทียบกับจำนวนอินพุตและเอาร์พุต (ปกติ $H_F \ll H$ และ
 $W_F \ll W$).

ในทำนองเดียวกัน เกรเดียนต์ของใบอัส สามารถหาได้จาก

$$\frac{\partial E_n}{\partial b_f} = \sum_{k=1}^{H'} \sum_{l=1}^{W'} \delta_{fkl}. \quad (6.20)$$

พิจารณา δ_{fkl} ของชั้นที่ m^{th} ว่า $a_{fkl}^{(m)}$ เชื่อมไปสู่เอาร์พุตสุดท้ายและฟังก์ชันจุดประสงค์ E_n ผ่าน
เอาร์พุตของชั้น $z_{fkl}^{(m)}$ 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 $.(m)$ เม้นระบุชั้นคำนวณ.

จากสมการ 6.11 จะได้

$$\delta_{fkl}^{(m)} = \frac{\partial E_n}{\partial z_{fkl}^{(m)}} \cdot h' \left(a_{fkl}^{(m)} \right) \quad (6.21)$$

โครงข่ายคอนโวโลชันจะมีชั้นคำนวน 3 ชั้น ได้แก่ ชั้นคอนโวโลชัน, ชั้นดึงรวม, และชั้นเชื่อมต่อเติม ที่ โดย การใช้งาน บางครั้งอาจมีหรือไม่มีชั้นดึงรวมหรือชั้นเชื่อมต่อเติมที่ก็ได้ นอกจากนั้น จำนวนชั้น ทั้งหมดและการเรียงลำดับของชั้นชนิดต่าง ๆ ก็อาจแตกต่างกันไป. การคำนวนค่า $\frac{\partial E_n}{\partial z_{fkl}^{(m)}}$ ของชั้น m^{th} จะขึ้นกับชั้นคำนวนถัดไป (หรือฟังก์ชันจุดประสงค์ หากชั้น m^{th} เป็นชั้นสุดท้าย). เพื่อความสะดวก กำหนด

$$\hat{\delta}_{fkl}^{(m)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(m)}}. \quad (6.22)$$

ดังนั้น เช่นเดียวกับโครงข่ายประสาทเทียมแบบเชื่อมต่อเติมที่ การคำนวนการแพร่กระจายย้อนกลับของโครงข่ายคอนโวโลชัน แต่ละชั้นจะคำนวนค่า $\hat{\delta}$ ออกมาด้วย เพียงแต่ ค่า $\hat{\delta}_{fkl}^{(m)}$ จะคำนวนมา จากชั้นที่ $(m+1)^{st}$ และชั้นที่ m^{th} ก็จะต้องคำนวนค่า $\hat{\delta}_{fkl}^{(m-1)}$ เพื่อให้ชั้น $(m-1)^{st}$ สามารถนำไปคำนวนหาค่าเกรเดียนต์ของค่าน้ำหนักได้.

เนื่องจาก $\hat{\delta}_{fkl}^{(m)}$ จะได้จากการคำนวนจากชั้นที่ $(m+1)^{st}$ จึงจะตรวจสอบว่าที่ เมื่อพิจารณาชั้นคำนวน v^{th} เราจะอภิปรายถึงการคำนวนหาค่า $\hat{\delta}^{(v-1)}$ ซึ่งเน้นว่า ค่า $\hat{\delta}^{(v-1)}$ นี้คำนวนจากชั้น v^{th} แต่นำไปใช้ในการหาค่าอนุพันธ์สำหรับชั้น $(v-1)^{st}$. รูปที่ 6.10 แสดงภาพการผ่านค่า $\hat{\delta}$ ที่คำนวนจากชั้น v^{th} เพื่อนำไปใช้คำนวนค่าเกรเดียนต์ของชั้น $(v-1)^{st}$.

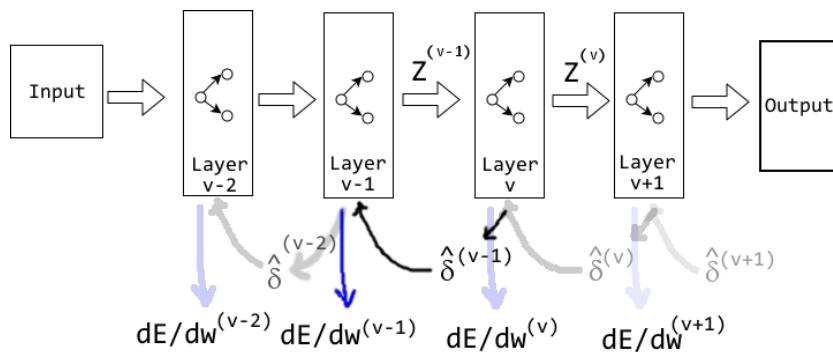
เมื่อพิจารณาโครงข่ายคอนโวโลชัน ชนิดของชั้นคำนวนก่อนหน้าของชั้นคอนโวโลชันจะเป็น ชั้นคำนวนคอนโวโลชัน ชั้นดึงรวม หรืออินพุตได้เท่านั้น. โครงข่ายคอนโวโลชันไม่มีกรณีที่ชั้นเชื่อมต่อเติมที่อยู่ก่อนหน้าชั้นคอนโวโลชัน เพราะชั้นเชื่อมต่อเติมที่ไม่มีโครงสร้างเชิงมิติของข้อมูล ดังนั้นหากจัดเรียงชั้นเชื่อมต่อเติมก่อนชั้นคอนโวโลชัน จึงไม่สามารถใช้ประโยชน์จากการคอนโวโลชันกับโครงสร้างเชิงมิติของข้อมูลได้.

ชนิดของชั้นคำนวนหลังจากชั้นคอนโวโลชัน อาจเป็น ชั้นคอนโวโลชัน ชั้นดึงรวม ชั้นเชื่อมต่อเติมที่ หรือเอาร์พุตก็ได้.

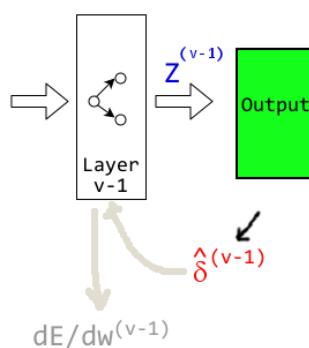
กรณีชั้นเอาร์พุต สำหรับกรณีชั้นคำนวน $(v-1)^{st}$ เป็นชั้นคำนวนสุดท้าย และชั้นคำนวน $(v-1)^{st}$ ต้องการค่า $\hat{\delta}^{(v-1)} = \frac{\partial E_n}{\partial z^{(v-1)}}$ จากชั้นเอาร์พุต. รูปที่ 6.11 แสดงการผ่านค่า $\hat{\delta}$ จากชั้นเอาร์พุตกลับไปให้ชั้นคำนวนสุดท้าย.

ที่ชั้นเอาร์พุต³ ค่า $\hat{\delta}^{(v-1)}$ ก็สามารถหาเกรเดียนต์ได้โดยตรง จากการแทนค่าฟังก์ชันจุดประสงค์ ในพจน์ของตัวแปร $z^{(v-1)}$ และการหาค่าอนุพันธ์.

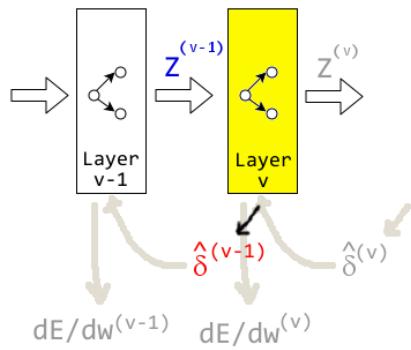
³ กรรมของชั้นเอาร์พุตเป็นชั้นเชิงตรรกะหลังจากชั้นคำนวนสุดท้าย ก็เพื่อให้ง่ายต่อการเรียบเรียงเนื้อหา. นั่นคือ ทุกรณี จะเรียกเป็นชั้น v^{th} และจะแสดงวิธีการหา $\hat{\delta}^{(v-1)}$ สำหรับชั้นก่อนหน้า.



รูปที่ 6.10 แผนภาพการแพร่กระจายย้อนกลับของค่า $\hat{\delta}$ ในโครงข่ายประสาทเทียม 4 ชั้น. จุดประสงค์หลักของการแพร่กระจายย้อนกลับ คือการคำนวณค่าเกรเดียนต์หรืออนุพันธ์ของฟังก์ชันจุดประสีค์เทียบกับค่าน้ำหนัก. 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 $\hat{\delta}$ คืออนุพันธ์ของฟังก์ชันจุดประสีค์เทียบกับเอ้าต์พุตของชั้น และเอ้าต์พุตของชั้นสุดท้ายเชื่อมโยงกับฟังก์ชันจุดประสีค์โดยตรง จึงทำให้ค่า $\hat{\delta}$ สามารถคำนวณได้โดยตรง. นอกจากนั้น สังเกตว่าชั้นคำนวณแรกสุด (ชั้น $v - 2$) ต้องการค่า $\hat{\delta}^{(v-2)}$ จากชั้น $v - 1$. แต่ชั้นคำนวณแรกสุด ไม่จำเป็นต้องผ่านค่า $\hat{\delta}$ ออกมานะ. ชั้นคำนวณ $v - 2$ สามารถคำนวณค่า $\hat{\delta}^{(v-3)}$ ออกมайдี แต่เนื่องจากไม่มีชั้นคำนวณ $v - 3$ ที่ต้องค่า $\hat{\delta}^{(v-3)}$ นี้ จึงทำให้ชั้น $v - 2$ ซึ่งเป็นชั้นคำนวณแรกสุด ไม่จำเป็นต้องผ่านค่า $\hat{\delta}$ ออกมานะ.



รูปที่ 6.11 การแพร่กระจายย้อนกลับจากชั้นเอ้าต์พุต. แผนภาพแสดงการแพร่กระจายย้อนกลับของค่า $\hat{\delta}$ โดยชั้นเอ้าต์พุตจะผ่านค่า $\hat{\delta}$ จากชั้นเอ้าต์พุตกลับไปให้ชั้น $v - 1$ เพื่อให้ชั้น $v - 1$ สามารถใช้คำนวณเกรเดียนต์ต่อน้ำหนักของชั้นได้. ค่า $\hat{\delta}^{(v-1)} = \frac{\partial E}{\partial z^{(v-1)}}$ ซึ่งที่ชั้นเอ้าต์พุต ค่านี้สามารถคำนวณได้ตรงไปตรงมา โดยการแทนค่าฟังก์ชันจุดประสีค์ E ที่กำหนดในพจน์ของ $z^{(v-1)}$ และหาอนุพันธ์.



รูปที่ 6.12 การแพร่กระจายย้อนกลับจากชั้นชั้นก่อน. แผนภาพแสดงการแพร่กระจายย้อนกลับของค่า $\hat{\delta}$ โดยชั้นคำนวณที่ v ผ่านค่า $\hat{\delta}^{(v-1)}$ กลับไปให้ชั้น $v-1$ เพื่อให้ชั้น $v-1$ สามารถใช้คำนวณเกรเดียนต์ต่อน้ำหนักของชั้นได้. ในขณะที่ชั้น v เองก็รับ $\hat{\delta}^{(v)}$ มาเพื่อคำนวณเกรเดียนต์ต่อน้ำหนักของชั้น.

ตัวอย่างเช่น พังก์ชันจุดประสงค์ $E_n = \lambda \sum_{q=1}^F \sum_{r=1}^{H'} \sum_{s=1}^{W'} \left(y_{qrs} - z_{qrs}^{(v-1)} \right)^2$ เมื่อ λ เป็นค่าคงที่, F เป็นจำนวนชุดของเอาร์พุต, H' และ W' เป็นขนาดของเอาร์พุต, y_{qrs} เป็นแทรด (ground-truth), และ $z_{qrs}^{(v-1)}$ คือเอาร์พุตจากชั้น $(v-1)^{st}$.

ดังนั้น จากการแทนค่าพังก์ชันจุดประสงค์ในพจน์ของเอาร์พุต

$$\begin{aligned}\hat{\delta}_{fkl}^{(v-1)} &= \frac{\partial \lambda \sum_{q=1}^F \sum_{r=1}^{H'} \sum_{s=1}^{W'} \left(y_{qrs} - z_{qrs}^{(v-1)} \right)^2}{\partial z_{fkl}^{(v-1)}} \\ &= -2\lambda (y_{fkl} - z_{fkl}^{(v-1)}).\end{aligned}$$

ดูเนื้อหาโยโล่ ในหัวข้อ 7.1 เพิ่มเติมสำหรับตัวอย่างการใช้โครงข่ายคอนโวลูชัน โดยใช้ชั้นคอนโวลูชันเป็นชั้นคำนวณสุดท้าย.

แต่หากชั้นคอนโวลูชันไม่ได้เป็นชั้นคำนวณสุดท้าย ค่า $\hat{\delta}^{(v-1)}$ จะได้จากการคำนวณจากชั้นที่ v^{th} ตามชนิดของชั้น v^{th} . รูปที่ 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 $(v-1)^{st}$ เชื่อมต่อไปถึงพังก์ชันจุดประสงค์ E_n ผ่านชั้น v^{th} โดย เอาร์พุตของชั้น $(v-1)^{st}$ จะกลายเป็นอินพุตของชั้น v^{th} .

เอาร์พุต $z_{fkl}^{(v-1)}$ จากชั้นคอนโวลูชัน เมื่อเข้าเป็นอินพุตของชั้นเชื่อมต่อเต็มที่ จะถูกสลายโครงสร้างลงเป็น $z_q^{(v-1)}$ โดย $z_{fkl} = z_q$ เมื่อ $q = l + W' \cdot (k-1) + H'W' \cdot (f-1)$ สำหรับ $f = 1, \dots, F; k = 1, \dots, H'; l = 1, \dots, W'$.

ดังนั้น

$$\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \frac{\partial E_n}{\partial z_q^{(v-1)}}. \quad (6.23)$$

ค่า $\frac{\partial E_n}{\partial z_q^{(v-1)}}$ ก็สามารถหาได้ เช่นเดียวกับชั้นเชื่อมต่อเติมที่ ซึ่งได้อธิบายในหัวข้อ 3.3.

ทบทวนเรื่องเกรเดียนต์ของชั้นเชื่อมต่อเติมที่ (จากหัวข้อ 3.3) พิจารณาที่ชั้น v^{th} ซึ่งเป็นชั้นเชื่อมต่อเติมที่. เอาร์พุตของชั้น $(v-1)^{st}$ 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 $a_r^{(v)}$ คือค่าการกระตุ้นของชั้นเชื่อมต่อเติมที่ 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

$$\frac{\partial E_n}{\partial z_q^{(v-1)}} = \sum_{r=1}^R \delta_r^{(v)} \cdot w_{rq}^{(v)}$$

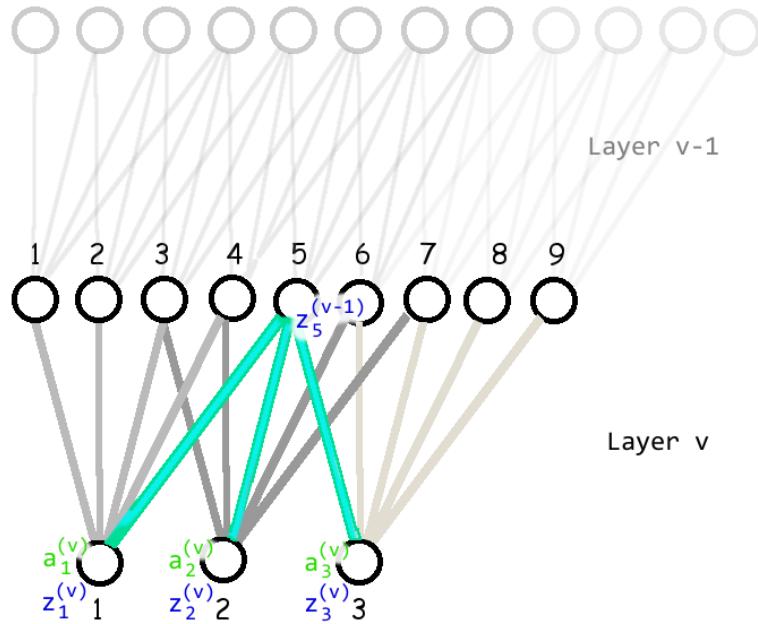
และค่า $\delta_r^{(v)}$ ก็สามารถหาได้จากการแพร่กระจายอย่างลับ ดังอธิบายในหัวข้อ 3.3 (สมการ 3.29 สำหรับชั้นคำนวณสุดท้าย หรือสมการ 3.31 สำหรับชั้นเชื่อมต่อเติมที่ที่ไม่ใช่ชั้นคำนวณสุดท้าย)

กรณีชั้นค่อนโวโลชั่น ที่ชั้นค่อนโวโลชั่น v^{th} ค่า $\hat{\delta}_{fkl}^{(v-1)}$ ก็สามารถหาได้ในลักษณะเดียวกับชั้นเชื่อมต่อเติมที่ เพียงแต่หน่วยเอาร์พุตของชั้นค่อนโวโลชั่นที่ $(v-1)^{st}$ เชื่อมต่อกับหน่วยในชั้น v^{th} เนพาะหน่วยที่สัมพันธ์กัน และค่าน้ำหนักมีการใช้ร่วมกัน.

หน่วย $z_{fkl}^{(v-1)}$ ส่งอิทธิพลไปถึงฟังก์ชันจุดประสงค์ E_n ผ่านหน่วย $a_{qrs}^{(v)}$ และหน่วย $a_{qrs}^{(v)}$ เชื่อมต่อกับหน่วย $z_{fkl}^{(v-1)}$ เนพาะหน่วยที่สัมพันธ์กัน. รูปที่ 6.13 แสดงการเชื่อมต่อของเอาร์พุตของชั้น $(v-1)^{st}$ กับเอาร์พุตของชั้น v^{th} . ในรูปแสดงแผนภาพของชั้นค่อนโวโลชั่นหนึ่งมิติ แต่การเชื่อมต่อชั้นค่อนโวโลชั่นสองมิติ ที่ทำในลักษณะเดียวกัน.

เมื่อพิจารณาการเชื่อมต่อ จะพบว่า ตำแหน่งเอาร์พุตของชั้น v^{th} สัมพันธ์กับตำแหน่งเอาร์พุตของชั้น $(v-1)^{st}$ โดย สำหรับแต่ละลักษณะสำคัญ q , หน่วย $a_{qrs}^{(v)}$ เชื่อมต่อ $z_{fkl}^{(v-1)}$, $f = 1, \dots, F$ เมื่อ F คือจำนวนลักษณะสำคัญของชั้น $(v-1)^{st}$, $r \in \Omega(k, S_H, H_F)$, $s \in \Omega(l, S_W, W_F)$, ขนาด กว้างย่างกับขนาดฟิลเตอร์ของชั้น v^{th} คือ $S_H \times S_W$ กับ $H_F \times W_F$ ตามลำดับ, และ ฟังก์ชันเซต

$$\Omega(k, S, H_F) = \left\{ \frac{k-i}{S} + 1 : (k-i \geq 0) \text{ and } ((k-i) \bmod S = 0) \right\}_{i=1, \dots, H_F} \quad (6.24)$$



รูปที่ 6.13 การแพร่กระจายบัญญาติผ่านโครงสร้างการต่อเชื่อม. แผนภาพการเชื่อมต่อแสดงหน่วยชั้น $v - 1$ มีอิฐพลกับสามหน่วยชั้น v โดยชั้น v ใช้ก้าวย่างขนาด 2 และพิลเตอร์ขนาด 5 และมีเพียงพิลเตอร์เดียว. ตัวอย่างเน้นแสดงเอาต์พุตหน่วยที่ 5 (ชั้น $v - 1$) เชื่อมต่อกับคอนโวลูชันเอาต์พุตหน่วยที่ 1, 2, 3 (ชั้น v). นั่นคือ $z_5^{(v-1)}$ เชื่อมต่อกับ $a_1^{(v)}, a_2^{(v)}, a_3^{(v)}$.

ตัวอย่างเช่น หน่วยเอาต์พุต $z_{fkl}^{(v-1)}$ สำหรับ $f = 1, \dots, 4; k = 1, \dots, 5; l = 1, \dots, 5$. นั่นคือ ชั้น $(v - 1)^{st}$ มีสีลักษณะสำคัญ ที่เกิดจากการใช้พิลเตอร์สี่ตัว และได้ແນที่ลักษณะสำคัญแต่ละอันเป็นขนาด 5×5 . เมื่อ หน่วยเอาต์พุตในชั้น $(v - 1)^{st}$ เชื่อมต่อกับชั้น v^{th} ที่เป็นชั้นคอนโวลูชันโดยชั้น v^{th} มีแปดลักษณะสำคัญ ($q = 1, \dots, 8$) ใช้พิลเตอร์ขนาด 3×3 สำหรับแต่ละลักษณะสำคัญ และใช้ขนาดก้าวย่างเป็น 1×1 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 $r \in \{\frac{1-i}{1} + 1 : (1 - i \geq 0) \text{ and } ((1 - i) \bmod 1 = 0)\}_{i=1,\dots,3}$ และเมื่อพิจารณาสมาชิกกับเงื่อนไข. 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 $i = 1$,

สมมติก $\frac{2-1}{1} + 1 = 2$. ที่ $i = 2$, สมมติก $\frac{2-2}{1} + 1 = 1$. 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) ตัวนี้เชิงลำดับ k และ l ของชั้น $(v-1)^{st}$ จะบวกตัวนี้เชิงลำดับ r และ s สำหรับทุก ๆ ตัวนิอิสระ q ของหน่วยในชั้น v^{th} . ตารางที่ 6.1 แสดงความสัมพันธ์ตำแหน่งหน่วยที่ k ในชั้น $(v-1)^{st}$ กับตำแหน่งหน่วยที่ r ในชั้น v^{th} . ตัวนี้ l ก็เป็นในลักษณะเดียวกัน. (2) หน่วยที่มีตำแหน่งเชิงลำดับเดียวกัน แต่ต่างตำแหน่งอิสระ เช่น $z_{1,5,5}$ กับ $z_{4,5,5}$ เชื่อมต่อกับหน่วยเดียวกัน.

จากการเชื่อมต่อข้างต้น เมื่อพิจารณาชั้นตอนโวลุชั่นที่ v^{th} , หน่วย $z_{fkl}^{(v-1)}$ เชื่อมต่อไปสู่ເອົາຕຸພູດສຸດທ້າຍและค่าຝັກໜຸດປະສົງ ผ่าน $a_{qrs}^{(v)}$ และจากກວລູກໂຈ່ ຈະໄດ້

$$\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} = \sum_{q=1}^Q \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \frac{\partial E_n}{\partial a_{qrs}^{(v)}} \frac{\partial a_{qrs}^{(v)}}{\partial z_{fkl}^{(v-1)}} \quad (6.25)$$

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 $S_H \times S_W$ เป็นขนาดฟิลเตอร์และขนาดก้าวย่างในชั้น v^{th} ตามลำดับ.

จากนิยาม 6.18 และการคำนวณตอนโวลุชั่นสมการ 6.10 (ອິນພູດ $\hat{\mathbf{X}}$ ในสมการ 6.10 คือ อິນພູດของชั้น v^{th} ຊຶ່ງກີ່ມີເອົາຕຸພູດຂອງชັກກົນໜ້າ, ນັ້ນຄື່ອງ $\hat{\mathbf{X}} \equiv \mathbf{Z}^{(v-1)}$ ແລະ ເພື່ອໃຫ້ວແປຣດັບນີ້ຟິລເຕອຮ

ตารางที่ 6.1 ตัวอย่างแสดงความสัมพันธ์ต้นของหน่วย($v - 1$) กับดัชนีของหน่วย(v) ที่เข้มต่อกัน. ตำแหน่งหน่วยที่ k ($v - 1$) เชื่อมต่อกับตำแหน่งหน่วยที่ r (v) เมื่อ $v \equiv k$ ใช้ก้าวย่าง และพิลเตอร์ขนาดต่าง ๆ. วงเล็บได้เขตค่าของ r สรุปย่อ เหตุผลของค่าสมาชิกที่ได้และไม่ได้ เช่น $(-, 1, x)$ ที่ $k = 2, S_H = 2, H_F = 3$ หมายถึง $i = 1$ ไม่ผ่านเงื่อนไขที่สอง $((2 - 1) \bmod 2 = 1)$, $i = 2$ ได้ค่าสมาชิกเป็น 1, และ $i = 3$ ไม่ผ่านเงื่อนไขที่หนึ่ง $(2 - 3 < 0)$ เป็นต้น

ก้าวย่าง	$S_H = 1$		$S_H = 2$	
พิลเตอร์	$H_F = 3$	$H_F = 5$	$H_F = 3$	$H_F = 5$
$k = 1$	$r \in \{1\}$ $(1, x, x)$	$r \in \{1\}$ $(1, x, x, x, x)$	$r \in \{1\}$ $(1, x, x)$	$r \in \{1\}$ $(1, x, x, x, x)$
$k = 2$	$r \in \{1, 2\}$ $(2, 1, x)$	$r \in \{1, 2\}$ $(2, 1, x, x, x)$	$r \in \{1\}$ $(-, 1, x)$	$r \in \{1\}$ $(-, 1, x, x, x)$
$k = 3$	$r \in \{1, 2, 3\}$ $(3, 2, 1)$	$r \in \{1, 2, 3\}$ $(3, 2, 1, x, x)$	$r \in \{1, 2\}$ $(2, -, 1)$	$r \in \{1, 2\}$ $(2, -, 1, x, x)$
$k = 4$	$r \in \{2, 3, 4\}$ $(4, 3, 2)$	$r \in \{1, 2, 3, 4\}$ $(4, 3, 2, 1, x)$	$r \in \{2\}$ $(-, 2, -)$	$r \in \{1, 2\}$ $(-, 2, -, 1, x)$
$k = 5$	$r \in \{3, 4, 5\}$ $(5, 4, 3)$	$r \in \{1, 2, 3, 4, 5\}$ $(5, 4, 3, 2, 1)$	$r \in \{2, 3\}$ $(3, -, 2)$	$r \in \{1, 2, 3\}$ $(3, -, 2, -, 1)$

จำแนกได้ง่าย ที่นี้ใช้ดัชนี \hat{f} แทนดัชนีพิลเตอร์ สำหรับการคำนวณคอนโวโลชัน, เมื่อแทนค่าลงไปในสมการ 6.25 จะได้ว่า

$$\begin{aligned} \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}} &= \sum_q \sum_r \sum_s \delta_{qrs}^{(v)} \frac{\partial \left(b_{q,r,s}^{(v)} + \sum_{\hat{f}} \sum_i \sum_j w_{q\hat{f}ij}^{(v)} z_{\hat{f},S_H \cdot (r-1)+i, S_W \cdot (s-1)+j}^{(v-1)} \right)}{\partial z_{fkl}^{(v-1)}} \\ &= \sum_q \sum_r \sum_s \delta_{qrs}^{(v)} \cdot w_{q,f,k-S_H \cdot (r-1), l-S_W \cdot (s-1)}^{(v)}. \end{aligned} \quad (6.26)$$

สังเกตว่า เพื่อคำนวณหา $\hat{\delta}_{fkl}^{(v-1)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$ สำหรับส่งไปให้ชั้น $(v-1)^{st}$, ชั้น v^{th} ต้องคำนวณ $\delta_{qrs}^{(v)}$ และ จากสมการ 6.21 ค่า $\delta_{qrs}^{(v)} = \frac{\partial E_n}{\partial z_{qrs}^{(v)}} \cdot h' \left(a_{qrs}^{(v)} \right)$ เมื่อ $h'(\cdot)$ คืออนุพันธ์ของฟังก์ชันกระตุนชั้น v^{th} . ส่วนค่า $\frac{\partial E_n}{\partial z_{qrs}^{(v)}} \equiv \hat{\delta}_{qrs}^{(v)}$ ก็ได้มาจากการชั้น $(v+1)^{st}$ อิกทอดหนึ่ง. หัวข้อ 6.4 สรุปย่อชั้นตอนการคำนวณเกรเดียนต์สำหรับโครงข่ายคอนโวโลชัน และรายการ 6.2 แสดงตัวอย่างโปรแกรมโครงข่ายคอนโวโลชัน.

ขนาดก้าวย่างที่นิยม พิจารณา $\hat{\delta}_{fkl}^{(v-1)}$ สำหรับกรณีชั้น v^h ใช้ก้าวย่างขนาด 1×1 ซึ่งเป็นขนาดที่มักถูกใช้กับชั้นคอนโวลูชัน. กรณีนี้จะทำให้

$$\Omega_r = \{k - i + 1 : k - i \geq 0\}_{i=1,\dots,H_F}, \quad (6.27)$$

$$\Omega_s = \{l - i + 1 : l - i \geq 0\}_{i=1,\dots,W_F}. \quad (6.28)$$

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^h เป็นชั้นดึงรวม หน่วย $z_{fkl}^{(v-1)}$ เชื่อมต่อไปสู่เอาร์พุตสุดท้ายและค่าพงก์ชั้นจุดประสงค์ ผ่านหน่วย $a_{frs}^{(v)}$. สังเกตว่า เนื่องจากชั้นดึงรวมไม่ได้ประมวลผลในชุดมิติของมิติอิสระ ดังนั้น หน่วยของชั้นดึงรวมในลักษณะสำคัญ f เกี่ยวข้องกับหน่วยก่อนหน้าในลักษณะสำคัญ f เช่นกันเท่านั้น. ชั้นดึงรวมรักษาชุดมิติอิสระไว้ (ขนาดชุดมิติอิสระของอินพุตและเอาร์พุตเท่ากัน).

พิจารณาการคำนวณเอาร์พุตของชั้นดึงรวม (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 ที่ถูกดึงมารวมกัน, $g(\cdot)$ เป็นฟังก์ชันดึงรวม, และ $z_{frs}^{(v)}$ คือเอาร์พุตของชั้น v^h ซึ่งเป็นชั้นดึงรวม โดย $S_H \times S_W$ กับ $H_F \times W_F$ คือขนาดก้าวย่างกับขนาดฟิลเตอร์ ของชั้น v^h ตามลำดับ.

สังเกตความสัมพันธ์ระหว่าง $z_{fkl}^{(v-1)}$ กับ $z_{frs}^{(v)}$ ในชั้นดึงรวม จะคล้ายกับ ในชั้นคอนโวลูชัน โดยต่างกันที่ (1) ชั้นดึงรวมไม่มีการคำนวณค่ากระตู้น a และ (2) ชั้นดึงรวมไม่ได้ประมวลผลในชุดมิติของมิติอิสระ. แต่หากมองเฉพาะความสัมพันธ์ในชุดมิติเชิงลำดับ จะพบว่าหน่วยในชั้นดึงรวมมีความสัมพันธ์กับหน่วยในชั้นติดกัน ในลักษณะเดียวกับชั้นคอนโวลูชัน (เปรียบเทียบตัวนี้ชุดมิติเชิงลำดับ

สมการ 6.13 กับ สมการ 6.10) ดังนั้น เมื่อโยงความสัมพันธ์ย้อนกลับ 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 $\frac{\partial E_n}{\partial z_{frs}^{(v)}} = \hat{\delta}_{frs}^{(v)}$ 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 โดย

การหาอนุพันธ์ 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*), และแบบอาร์เร็มแอล (root-mean-squared pooling หรือ *rms pooling*).

เมื่อใช้การดึงรวมแบบมากที่สุด นั่นคือ ฟังก์ชันดึงรวม $g(\{z_1, \dots, z_n\}) = \max\{z_1, \dots, z_n\}$ 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 ฟังก์ชันดึงรวม $g(\{z_1, \dots, z_n\}) = \frac{1}{n} \sum_{i=1}^n z_i$ และจะได้ว่า

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{1}{H_F W_F}. \quad (6.36)$$

เมื่อใช้การดึงรวมแบบอาร์เรียมแอล นั่นคือ พังก์ชันดึงรวม $g(\{z_1, \dots, z_n\}) = \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}$ และ จะได้ว่า

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{z_{fkl}^{(v-1)}}{H_F W_F \cdot z_{frs}^{(v)}} \quad (6.37)$$

ชั้นดึงรวมเองไม่มีค่าน้ำหนักที่ต้องปรับ จึงไม่ต้องคำนวณเกรเดียนต์เทียบนำหนักของชั้น แต่ชั้นดึงรวมต้องผ่านค่า $\hat{\delta}_{fkl}^{(v-1)} = \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$ ไปให้ชั้น $(v-1)^{st}$.

6.4 สรุปการคำนวณของโครงข่ายคอนโวลูชันสองมิติ

กำหนดให้อินพุตที่ผ่านการเติมเต็ม $Z^{(0)} \in \mathbb{R}^{C \times H' \times W'}$. คำนวณการแพร์กกระจายไปข้างหน้า ตาม ชนิดของชั้นคำนวณ สำหรับชั้น $m = 1, \dots, M$ เมื่อ M คือจำนวนชั้นคำนวณทั้งหมด ดังนั้นเอาต์พุต ของโครงข่าย คือเอาต์พุตของชั้นสุดท้าย. นั่นคือ $Z^{(M)}$ เป็นเอาต์พุตสุดท้าย.

กรณีชั้นคอนโวลูชัน คำนวณสมการ 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 และ ค่าน้ำหนักของชั้น m^{th} , $h^{(m)}$ คือพังก์ชันกระตุ้นของชั้น m^{th} , โดย F, H, W คือจำนวนลักษณะสำคัญ และขนาดของแผนที่เอาต์พุตของชั้น m^{th} , H_F, W_F คือขนาดพิลเตอร์ของชั้น m^{th} ตามแนวตั้งและ นอนตามลำดับ และ S_H, S_W คือขนาดก้าวย่างของชั้น m^{th} ตามแนวตั้งและนอนตามลำดับ.

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 คือจำนวนลักษณะสำคัญและขนาดของแผนที่เอาต์พุตของชั้น m^{th} , H_F, W_F คือ ขนาดพิลเตอร์ของชั้น m^{th} ตามแนวตั้งและนอนตามลำดับ และ S_H, S_W คือขนาดก้าวย่างของชั้น m^{th} ตามแนวตั้งและนอนตามลำดับ.

กรณีชั้นเชื่อมต่อเต็มที่ สลายโครงสร้างอินพุต (ถ้าจำเป็น). นั่นคือคำนวน

$$z_q^{(m-1)} = z_{fkl}^{(m-1)}$$

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 ขนาดเอาร์พุตแนวอนของชั้น $(m-1)^{st}$ ตามลำดับ.

คำนวนชั้นเชื่อมต่อเต็มที่ 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 $b_f^{(m)}, w_{fq}^{(m)}, h^{(m)}$ คือค่าใบอัส ค่าน้ำหนัก และฟังก์ชันกระตุ้นของชั้น m^{th} โดย F เป็นจำนวนหน่วยเอาร์พุตในชั้น m^{th} .

เกรเดียนต์ของโครงข่ายคอนโวโลชัน

เกรเดียนต์ของโครงข่ายคอนโวโลชันสามารถหาได้ โดยใช้การแพร่กระจายย้อนกลับ โดยคำนวนทีลีชั้น คำนวน เริ่มจากเอาร์พุต แล้วไปชั้นสุดท้าย แล้วไล่ย้อนกลับไปทีลีชั้นจนครบทุกชั้น. นั่นคือ หากชั้น คำนวนมีจำนวน M ชั้น วิธีแพร่กระจายย้อนกลับจะเริ่มที่เอาร์พุต (นับเป็น ชั้น $(M+1)^{st}$) แล้วไล่ย้อนกลับไปจนถึงชั้นแรก (ชั้น 1^{st}).

ค่าที่แต่ละชั้น v^{th} ต้องส่งย้อนกลับไปให้ชั้นก่อนหน้า คือ ค่า $\hat{\delta}_{fkl}^{(v-1)} \equiv \frac{\partial E_n}{\partial z_{fkl}^{(v-1)}}$ สำหรับ $f = 1, \dots, F'; k = 1, \dots, H'; l = 1, \dots, W'$ เมื่อ F', H', W' คือ จำนวนลักษณะสำคัญ ขนาดเอาร์พุตในแนวตั้ง ขนาดเอาร์พุตในแนวอน ของชั้น $(v-1)^{st}$ ตามลำดับ.

กรณีเอาร์พุต ในที่นี้ หมายถึง การคำนวนแรกสุด (ก่อนการคำนวนชั้นสุดท้าย). ที่กรณีชั้น v^{th} เป็นเอาร์พุต ($v = M+1$) เอาร์พุตไม่มีค่าน้ำหนักที่ต้องปรับ แต่ต้องการคำนวน $\hat{\delta}^{(v-1)} = \hat{\delta}^{(M)}$ เพื่อส่งกลับให้ชั้นคำนวนสุดท้าย.

ค่า $\hat{\delta}_{fkl}^{(M)}$ สามารถหาได้จาก

$$\hat{\delta}_{fkl}^{(M)} = \frac{\partial E_n}{\partial z_{fkl}^{(M)}}$$

ซึ่งมักหาได้ไม่ยาก เนื่องจากฟังก์ชันจุดประสงค์ E_n มักถูกนิยามในพจน์ของ $z_{fkl}^{(M)}$.

กรณีขั้นเชื่อมต่อเต็มที่ ที่กรณีขั้น v^{th} เป็นขั้นเชื่อมต่อเต็มที่ ขั้น v^{th} รับ $\hat{\delta}_j^{(v)}$ มาจากขั้น $(v+1)^{st}$ และขั้น v^{th} คำนวณเกรเดียนต์เทียบกับน้ำหนักของขั้นจากสมการ 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 $h'(\cdot)$ เป็นอนุพันธ์ของฟังก์ชันกราฟตุ้นในขั้น v^{th} . ขั้น v^{th} คำนวณ $\hat{\delta}_q^{(v-1)}$ เพื่อส่งย้อนไปให้ขั้น $(v-1)^{st}$ จากสมการ 3.31. นั่นคือ

$$\hat{\delta}_q^{(v-1)} = \sum_j w_{jq}^{(v)} \delta_j^{(v)}$$

สำหรับ $q = 1, \dots, Q$ เมื่อ Q เป็นจำนวนເອົ້າພຸດທະນາຂອງขั้น $(v-1)^{st}$.

หากขั้น $(v-1)^{st}$ เป็นขั้นคอนโวลูชันหรือขั้นดึงรวม 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' คือจำนวนລັກຂະນະສຳຄັນ, ขนาดເອົ້າພຸດແນວຕັ້ງ, ขนาดເອົ້າພຸດແນວອນຂອງขั้น $(v-1)^{st}$ ตามลำดับ.

กรณีขั้นดึงรวม ที่กรณีขั้น v^{th} เป็นขั้นดึงรวม ขั้น v^{th} รับ $\hat{\delta}_{frs}^{(v)}$ มาจากขั้น $(v+1)^{st}$. ขั้นดึงรวม ไม่มีค่าນ้ำหนัก ไม่จำเป็นต้องคำนวณเกรเดียนต์ แต่ขั้นดึงรวมต้องส่ง $\hat{\delta}_{fkl}^{(v-1)}$ ไปให้ขั้น $(v-1)^{st}$. ค่า $\hat{\delta}_{fkl}^{(v-1)}$ คำนวณได้จากสมการ 6.33. นั่นคือ

$$\hat{\delta}_{fkl}^{(v-1)} = \sum_{r \in \Omega_r} \sum_{s \in \Omega_s} \hat{\delta}_{frs}^{(v)} \frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$$

สำหรับ $f = 1, \dots, F'; k = 1, \dots, H'$ และ $l = 1, \dots, W'$ เมื่อ $\Omega_r = \Omega(k, S_H, H_F); \Omega_s = \Omega(l, S_W, W_F)$ และ F', H', W' คือจำนวนລັກຂະນະສຳຄັນ, ขนาดເອົ້າພຸດແນວຕັ້ງ, ขนาดເອົ້າພຸດແນວອນຂອງขั้น $(v-1)^{st}$ ตามลำดับ.

ฟังก์ชันเซตคำนวณได้จากสมการ 6.24,

$$\Omega(k, S, H) = \left\{ \frac{k-i}{S} + 1 : (k-i \geq 0) \text{ and } ((k-i) \bmod S = 0) \right\}_{i=1, \dots, H}.$$

ค่า $\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}}$ ขึ้นกับชนิดการดึงรวม. การดึงรวมแบบมากที่สุด ใช้สมการ 6.35,

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \begin{cases} 1 & \text{เมื่อ } z_{fkl}^{(v-1)} = \max(\{z_{f,S_H \cdot (r-1)+i, S_W \cdot (s-1)+j}^{(v-1)}\}_{i=1, \dots, H_F, j=1, \dots, W_F}), \\ 0 & \text{อื่น ๆ.} \end{cases}$$

การดึงรวมแบบเฉลี่ย ใช้สมการ 6.36,

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{1}{H_F W_F}$$

เมื่อ $H_F \times W_F$ เป็นขนาดของฟิลเตอร์ชั้นที่ v^{th} .

การดึงรวมแบบบาร์เรียมเอส ใช้สมการ 6.37

$$\frac{\partial z_{frs}^{(v)}}{\partial z_{fkl}^{(v-1)}} = \frac{z_{fkl}^{(v-1)}}{H_F W_F \cdot z_{frs}^{(v)}}$$

เมื่อ $H_F \times W_F$ เป็นขนาดของฟิลเตอร์ชั้นที่ v^{th} .

กรณีชั้นคอนโวโลชัน ที่กรณฑ์ชั้น v^{th} เป็นชั้นคอนโวโลชัน ชั้น v^{th} รับ $\hat{\delta}_{qrs}^{(v)}$ มาจากชั้น $(v+1)^{st}$ และชั้น v^{th} คำนวณเกรเดียนต์เทียบกับน้ำหนักของชั้นจากสมการ 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 ขนาดฟิลเตอร์แนวตั้ง ขนาดฟิลเตอร์แนวนอนของชั้น 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 เป็นค่าก้าวย่างตามแนวตั้งและนอนของชั้น $(v-1)^{st}$ และ $\delta_{qrs}^{(v)} = \hat{\delta}_{qrs}^{(v)} \cdot h'(a_{qrs}^{(v)})$ โดย $h'(a_{qrs}^{(v)})$ เป็นอนุพันธ์การกระทุนของชั้น v^{th} .

ชั้น v^{th} คำนวณ $\hat{\delta}_{fkl}^{(v-1)}$ 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 ขนาดฟิลเตอร์ตามแนวตั้ง ขนาดฟิลเตอร์ตามแนวนอน ขนาดก้าวย่างตามแนวตั้ง ขนาดก้าวย่างตามแนวนอนของชั้น v^{th} ตามลำดับ, F', H', W' คือจำนวนลักษณะสำคัญ, ขนาดเออร์พุตแนวตั้ง, ขนาดเออร์พุตแนวนอนของชั้น $(v - 1)^{st}$ 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 แต่ค่าพารามิเตอร์ที่นิยมใช้สำหรับโครงข่ายคอนโวลูชัน ได้แก่ สำหรับชั้นคอนโวลูชัน นิยมใช้กับฟิลเตอร์ขนาดเป็น 3×3 หรือ 5×5 และขนาดก้าวย่างเป็น 1×1 . สำหรับชั้นดึงรวม นิยมใช้กับฟิลเตอร์ขนาดเป็น 2×2 หรือ 3×3 และขนาดก้าวย่างเป็น 2×2 .

6.5 โครงข่ายคอนโวลูชันที่สำคัญ

การออกแบบโครงสร้างของโครงข่ายคอนโวลูชันในปัจจุบัน นิยมทำด้วยมุมมอง การกำหนดสาระสำคัญ ในระดับสูง (*high level of abstraction*). นั่นคือ คล้ายกับการออกแบบแบบวางจริงอิเล็กทรอนิกส์ ที่หากไม่ได้ต้องการคุณสมบัติอะไรมากนัก แทนที่จะออกแบบแบบจริงโดยเลือกอุปกรณ์พื้นฐาน เช่น ทรานซิสเตอร์ ตัวต้านทาน ตัวเก็บประจุ ไดโอด รีเลย์ และการเชื่อมต่อระหว่างอุปกรณ์พื้นฐานเหล่านี้ ปัจจุบัน การออกแบบแบบจริง นิยมเริ่มจากการเลือกวิธีรวมก่อน แล้วค่อยเสริม ประกอบ หรือดัดแปลง ให้เข้ากับความต้องการ.

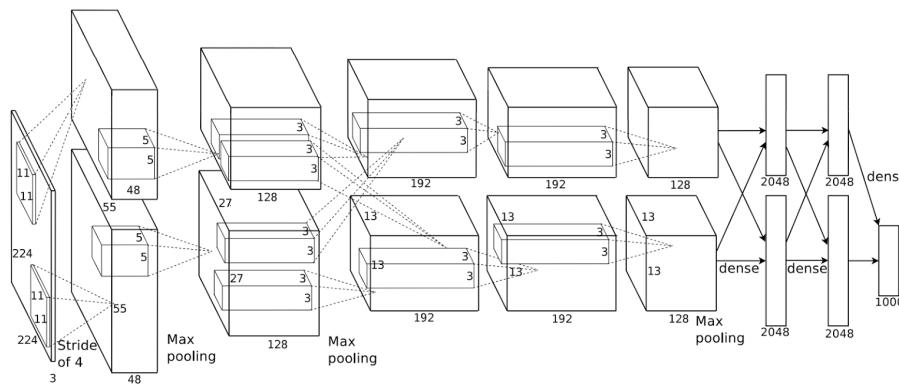
เช่นเดียวกัน การออกแบบโครงสร้างของโครงข่ายคอนโวลูชันในปัจจุบัน ก็นิยมเริ่มจากโครงข่ายคอนโวลูชันที่รู้จักกันดีแล้ว และดัดแปลงตามความเหมาะสม. หัวข้อต่อไปนี้ อภิปรายตัวอย่างของโครงข่ายคอนโวลูชันเด่น ๆ ที่รู้จักกันดี และนิยมถูกเลือกมาเป็นจุดเริ่มต้นของโครงสร้าง เช่น อเล็กซ์เน็ต[113], วีจีจีเน็ต[186], อินเซปชัน[195], เรสนีต[85], และเดนซ์เน็ต[92].

อเล็กซ์เน็ต

อเล็กซ์เน็ต (AlexNet[113]) เป็นโครงข่ายคอนโวลูชัน ที่ได้รับความสนใจอย่างมาก หลังจากชนะการแข่งขันจำแนกชนิดวัตถุในภาพถ่าย อิมเมจเนต (*ImageNet*) ในปี 2012 (ชุดข้อมูลมักถูกอ้างอิงว่า ImageNet LSVRC-2012).

อเล็กซ์เน็ตเป็นงานแรก ๆ ที่แสดงความสามารถในการทำงานจากเครื่อง ที่ใกล้เคียงกับระดับของมนุษย์ได้. การแข่งขัน⁴ ทดสอบผลด้วย ภาพถ่าย 100,000 ภาพ ที่แต่ละภาพมีฉลากเฉลยของ

⁴ <http://www.image-net.org/challenges/LSVRC/2012/> (ข้อมูลปรับปรุง 9 ก.พ. 2014. สืบค้น 25 ก.ค. 2020)



รูปที่ 6.14 โครงสร้างของอเล็กซ์เน็ต (รูปปรับปรุง เพิ่มความสมบูรณ์จากต้นฉบับ[113]). แผนที่อินพุต (ภาพสีที่กำหนดขนาดตามตัว) สัดส่วน $224 \times 224 \times 3$ แสดงด้วยภาพกล่องสี่เหลี่ยมทั้งมีสุด พร้อมขนาดในแนวนอนและแนวตั้ง และจำนวนช่องสี. กล่องสี่เหลี่ยมลึกภายในแผนที่อินพุต แสดงฟิลเตอร์ของชั้นคอนโวโลชัน (ขนาด 11×11) เส้นประจักษ์ฟิลเตอร์ ระบุทิศทางของผลลัพธ์การคำนวณ. กล่องสี่เหลี่ยมถัดมา แสดงแผนที่ลักษณะสำคัญที่ได้จากชั้นคอนโวโลชันแรก. เพื่อแก้ปัญหานาดความจำที่จำกัด อเล็กซ์เน็ตแยกการคำนวณออกเป็นสองเส้นทาง (และใช้การลดจีพียูสองการ์ด แต่ละการ์ดคำนวณแต่ละเส้นทาง). ดังนั้น แผนที่ลักษณะสำคัญขั้นแรก จึงแยกเป็นสองแผนที่ (เส้นทางบน และเส้นทางล่าง) จำนวนช่องในแผนที่ชั้นแรก (ซึ่งเท่ากับจำนวนฟิลเตอร์) คือ 48 (ในแต่ละเส้นทาง หรือรวม 96 ช่องในชั้นแรกทั้งสองเส้นทาง). ชั้นคอนโวโลชันที่สอง ใช้ฟิลเตอร์ขนาด 5×5 จำนวน 128 ในแต่ละเส้นทาง. ชั้นคอนโวโลชันที่สาม ใช้ฟิลเตอร์ขนาด 3×3 โดยมีการนำผลลัพธ์จากแต่ละเส้นทางมาประกอบกันเป็นแผนที่ลักษณะสำคัญขนาด 192 ในแต่ละเส้นทาง. ชั้นคอนโวโลชันที่สี่ ใช้ฟิลเตอร์ขนาด 3×3 จำนวน 192 ชุด ในแต่ละเส้นทาง โดยไม่มีการส่งผลลัพธ์ข้ามเส้นทาง. ชั้นคอนโวโลชันที่ห้า ใช้ฟิลเตอร์ขนาด 3×3 จำนวน 128 ชุด ในแต่ละเส้นทาง. สังเกตว่า หลังชั้นคอนโวโลชันที่หนึ่งที่สอง และที่ห้า มีชั้นดึงรวมแบบมากที่สุด ทำให้ขนาดของแผนที่ลักษณะสำคัญในชั้นถัดไปลดลงร้าว ๆ เท่าตัว (55×55 ลดลงเป็น 27×27 , 27×27 ลดลงเป็น 13×13). ส่วนแผนที่ลักษณะสำคัญของชั้นคอนโวโลชันที่ห้า จะถูกลดขนาดลง ก่อนถูกสลายโครงสร้าง เพื่อนำไปคำนวณกับชั้นเชื่อมต่อเติมที่ต่อไป. หมายเหตุ ขนาดของแผนที่ลักษณะสำคัญในชั้นที่หนึ่ง 55×55 ที่ลดลงจาก 224×224 เกิดจากการใช้ขนาดย่างก้าว 4.). ชั้นเชื่อมต่อเติมที่ชั้นแรก มีจำนวนหน่วยช่อง 2048 ต่อเส้นทาง ซึ่งมีการส่งผลลัพธ์ข้ามเส้นทาง และทำให้โครงสร้างสองเส้นทาง ทำงานเสมือนหนึ่งชั้นเชื่อมต่อเติมที่ขนาด 4096. ชั้นเชื่อมต่อเติมที่ชั้นสอง ก็มีจำนวนหน่วยช่อง 2048 ต่อเส้นทางเช่นกัน. ชั้นเชื่อมต่อเติมที่ชั้นที่สาม ซึ่งเป็นชั้นสุดท้าย และเป็นชั้นเอาต์พุตของอเล็กซ์เน็ต มีจำนวนหน่วยช่อง 1000 หน่วย (เท่ากับจำนวนชนิดคลาสที่ต้องการทำนาย) และใช้ฟังก์ชันกรอบต้นเป็นซอฟต์แมกซ์.

ชนิดวัตถุในภาพ. ชุดข้อมูลครอบคลุมถึง 1000 ชนิดวัตถุ. ผลตัดสินวัดจากค่าผิดพลาดของห้าชนิดอันดับสูงสุด (top-5 error rate) ซึ่งอเล็กซ์เน็ตทำได้ถึง 15.3%. หมายเหตุ ค่าผิดพลาดของห้าชนิดอันดับสูงสุด หมายถึง อัตราการหายผิด ซึ่งคือ อัตราส่วน จำนวนตัวอย่างที่ถูกคาดเดยว่าไม่อยู่ในห้าชนิดอันดับแรกสุดที่ทาย ต่อจำนวนตัวอย่างทั้งหมด.

อเล็กซ์เน็ตใช้ชั้นคอนโวโลชัน 5 ชั้น แล้วตามด้วยชั้นเชื่อมต่อเติมที่ 3 ชั้น รวมแล้วใช้ พารามิเตอร์ 60 ล้านตัว. อเล็กซ์เน็ต ใช้เรลูเป็นฟังก์ชันกระตุ้น เพื่อช่วยให้การเรียนรู้ทำได้ง่ายขึ้น และใช้กลไกตอกอก เพื่อลดปัญหาโวเวอร์ฟิตติ้ง. ที่สำคัญคือ อเล็กซ์เน็ต ใช้การประมวลผลจีพียูอย่างมีประสิทธิภาพ. อเล็กซ์เน็ตถูกฝึกกับตัวอย่างภาพรวม 1.2 ล้านภาพ (จากชุดข้อมูลอิมเมจเนต ของปี 2010 หรือ LSVRC-2010).

โครงสร้างของอเล็กซ์เน็ต แสดงดังรูปที่ 6.14. โครงสร้างของอเล็กซ์เน็ต แยกการคำนวณออกเป็น

สองเส้นทาง เพื่อแก้ปัญหาขนาดความจำ โดยใช้การ์ดประมวลผลจีพียูสองการ์ดร่วมกัน. การคำนวนของชั้นเชื่อมต่อเต็มที่ แม้จะแบ่งส่วนคำนวน (กระจายภาระทางฮาร์ดแวร์) แต่การนำผลลัพธ์มารวมกันทำให้ผลลัพธ์ที่ได้เหมือนกับว่าไม่มีการแบ่งส่วน.

6.6 อภิธานศัพท์

โครงข่าย convolutional neural network: โครงข่ายประสาทเทียมที่มีชั้นconvolutional layer.

ชั้น convoluional layer: ชั้นคำนวนที่อาศัยการเชื่อมต่อท้องถิ่นและการใช้ค่าน้ำหนักร่วม.

ฟิลเตอร์ (filter): ชุดค่าน้ำหนักของชั้น convoluional layer อาจเรียกว่า เคอร์เนล.

เคอร์เนล (kernel): ชุดค่าน้ำหนักของชั้น convoluional layer อาจเรียกว่า ฟิลเตอร์.

การเติมเต็มด้วยศูนย์ (zero-padding): การขยายมิติของอินพุตของชั้น convoluional layer ด้วยการเพิ่มมิติที่มีค่าเป็นศูนย์เข้าไป โดยมักมีจุดประสงค์ เพื่อควบคุมขนาดของเอาร์พุตของชั้น convoluional layer

ขนาดก้าวย่าง (stride): ขนาดการยับตัวແเน່ງของอินพุต เพื่อนำมาคำนวนเอาร์พุตหน่วยถัดไป.

แผนที่ลักษณะสำคัญ (feature map): เอาร์พุตจากชั้น convoluional layer

ชั้นดึงรวม (pooling layer): ชั้นคำนวน ที่สรุปสถิติของบริเวณท้องถิ่นต่าง ๆ ของอินพุตออกมาก.

ชั้นเชื่อมต่อเต็มที่ (fully connected layer): ชั้นคำนวนโครงข่ายประสาทเทียมแบบดั้งเดิม (นั่นคือ ไม่คำนึงถึงโครงสร้างมิติของอินพุต และไม่มีโครงสร้างมิติของเอาร์พุต).

6.7 แบบฝึกหัด

``Success is not final,
failure is not fatal,
it is the courage to continue that counts."

---Winston Churchill

“ความสำเร็จไม่ใช่สิ้นสุด
ความล้มเหลวไม่ใช่จุดจบ
ความกล้าหาญที่จะไปต่อเท่านั้นที่สำคัญ.”

—วินสตัน เชอร์ชิล

แบบฝึกหัด 6.1

จงตอบคำถามต่อไปนี้ เกี่ยวกับขั้นตอนโวโลชัน ลำดับขั้น และชุดมิติต่าง ๆ

- (ก) อินพุตเป็นเกékเตอร์ นั่นคือ $\mathbf{x} \in \mathbb{R}^{10}$ และขั้นตอนโวโลชันใช้ฟิลเตอร์ \mathbf{W} มีขนาด 3×15 ตัว โดยไม่มีการเติมเต็ม ขนาดย่างก้าวเป็น 1 และผลลัพธ์จากขั้นตอนโวโลชัน จะเป็นเทนเซอร์ ขนาดเท่าใด? คำให้ คูณการ 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 ชุดมิติแรก เป็นจำนวนลักษณะสำคัญ และตามด้วยชุดมิติอื่น ๆ (เช่น ชุดมิติลำดับ).
- (ข) อินพุตเป็นเทนเซอร์สองลำดับขั้น คือ $\mathbf{X} \in \mathbb{R}^{8 \times 10}$. ขั้นตอนโวโลชันใช้ฟิลเตอร์ \mathbf{W} ขนาด $8 \times 3 \times 15$ ตัว โดยทำคอนโวโลชัน (การเข้มต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะ กับชุดมิติที่สอง (ชุดมิติแรกเป็นเหมือนช่องลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มี การเติมเต็มอินพุต และใช้ขนาดย่างก้าวเป็น 1. ผลลัพธ์จากขั้นตอนโวโลชัน จะเป็นเทนเซอร์ ขนาดเท่าใด? คำให้ สำหรับฟิลเตอร์แต่ละตัว เอาร์พุต $a_k = b + \sum_c \sum_j w_{c,j} \cdot x_{c,k+j-1}$ โดย c แทนดัชนีของช่องลักษณะสำคัญ (ไม่มีความสัมพันธ์ในเชิงลำดับ) และ $k = 1, \dots, H - H_F + 1$.
- (ค) อินพุตเป็นเทนเซอร์สามลำดับขั้น นั่นคือ $\mathbf{X} \in \mathbb{R}^{3 \times 100 \times 200}$. ขั้นตอนโวโลชันใช้ฟิลเตอร์ \mathbf{W} ขนาด $3 \times 5 \times 5 \times 5$ จำนวน 24 ตัว โดยทำคอนโวโลชัน (การเข้มต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะ กับชุดมิติที่สองและสาม (ชุดมิติแรก เป็นเหมือนช่องลักษณะสำคัญ ที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มีการเติมเต็มอินพุต ใช้ขนาดย่างก้าว 1. ผลลัพธ์จากขั้นตอนโวโลชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำให้ คูณการ 6.8 (สำหรับฟิลเตอร์แต่ละตัว เมื่อขนาดก้าวย่าง เป็นหนึ่ง 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 $\mathbf{X} \in \mathbb{R}^{4 \times 300 \times 400 \times 50}$. ขั้นตอนโวโลชันใช้ฟิลเตอร์ \mathbf{W} ขนาด $4 \times 11 \times 11 \times 7$ จำนวน 64 ตัว โดยทำคอนโวโลชัน (การเข้มต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะ กับชุดมิติที่สอง ที่สาม และที่สี่ (ชุดมิติแรกเป็นเหมือนช่องลักษณะสำคัญ ที่ไม่มีความสัมพันธ์ในเชิงลำดับ). ไม่มีการเติมเต็มอินพุต และใช้ขนาดย่างก้าวเป็น 1. ผลลัพธ์

จากชั้นคอนโวลูชัน จะเป็นเทนเซอร์ขนาดเท่าใด? คำใบ้ สำหรับฟิลเตอร์แต่ละตัว เอาร์พุต
 $a_{k,l,m} = b + \sum_c \sum_i \sum_j \sum_q w_{c,i,j,q} \cdot x_{c,k+i-1,l+j-1,m+q-1}$.

แบบฝึกหัด 6.2

จากแบบฝึกหัด 6.1 จงประมาณขนาดเทนเซอร์ของเอาร์พุต ในกรณีต่าง ๆ เมื่อใช้ขนาดกว้างกว้าง เป็น 2, เป็น 3 และเป็น 4. คำใบ้ ดูสมการ 6.4 ($H' = \left\lfloor \frac{H-H_F}{S} \right\rfloor + 1$).

แบบฝึกหัด 6.3

จากแบบฝึกหัด 6.1 จงประมาณการเติมเต็มด้วยค่าศูนย์ (จำนวนค่าศูนย์ที่ต้องเติม) ทั้ง 4 กรณี โดย

- (แบบที่ 1) เติมให้เอาร์พุตมีขนาดเท่ากับอินพุต เมื่อใช้ขนาดกว้างกว้างเป็น 1 (พิจารณาเฉพาะในชุดมิติที่มีความสัมพันธ์เชิงลำดับ เช่น กรณี ข อินพุต $\mathbf{X} \in \mathbb{R}^{8 \times 10}$ แต่ชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ. ดังนั้น สัดส่วนของเอาร์พุตที่ต้องการคือ 15×10 หรือเอาร์พุต $\mathbf{A} \in \mathbb{R}^{15 \times 10}$. ขนาด 15 มาจากจำนวนฟิลเตอร์ที่ใช้ ไม่เกี่ยวกับการเติมเต็มด้วยค่าศูนย์).
- (แบบที่ 2) เติมให้เอาร์พุตมีขนาดเท่ากับ $\lceil \frac{H}{S} \rceil$ โดย H คือขนาดอินพุต และ S คือขนาดกว้างกว้าง เมื่อใช้ขนาดกว้างกว้างเป็น 2, เป็น 3, และเป็น 4 ตามลำดับ. (พิจารณาเฉพาะในชุดมิติที่มีความสัมพันธ์เชิงลำดับ เช่น กรณี ค อินพุต $\mathbf{X} \in \mathbb{R}^{3 \times 100 \times 200}$ แต่ชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ. ดังนั้น สัดส่วนของเอาร์พุตที่ต้องการคือ $24 \times 50 \times 100$ เมื่อใช้ขนาดกว้างกว้าง 2 และคือ $24 \times 34 \times 67$ เมื่อใช้ขนาดกว้างกว้าง 3 เป็นต้น).

คำใบ้ ดูสมการ 6.5 ซึ่งคือ $\hat{H} = S \cdot (\hat{H}' - 1) + H_F$ และ $\hat{H} = H$.

แบบฝึกหัด 6.4

จงคำนวณขนาดของเอาร์พุตจากชั้นคอนโวลูชัน สำหรับกรณีต่าง ๆ ดังนี้

- (ก) อินพุตเป็นเวกเตอร์ นั่นคือ $\mathbf{x} \in \mathbb{R}^{10}$ และชั้นคอนโวลูชันใช้ฟิลเตอร์ \mathbf{w} ขนาด 3 จำนวน 15 ตัว โดยเติมเต็มด้วยค่าศูนย์จำนวนรวม 2 ตัว ขนาดกว้างกว้างเป็น 1 แล้วผลลัพธ์จากชั้นคอนโวลูชันจะเป็นเทนเซอร์ขนาดเท่าใด?
- (ข) อินพุตเป็นเทนเซอร์สองลำดับชั้น คือ $\mathbf{X} \in \mathbb{R}^{8 \times 10}$. ชั้นคอนโวลูชันใช้ฟิลเตอร์ \mathbf{W} ขนาด 8×3 จำนวน 15 ตัว โดยทำคอนโวลูชัน (การเข้ามต่อท้องถิ่นและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สอง (ชุดมิติแรกเป็นเสมีอนซ่องลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). มีการเติมเต็มอินพุตด้วยค่าศูนย์จำนวน 7 ตัว และใช้ขนาดกว้างกว้างเป็น 2. ผลลัพธ์จากชั้นคอนโวลูชันจะเป็นเทนเซอร์ขนาดเท่าใด?

- (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 (การเข้ามต่อห้องถินและใช้ค่าน้ำหนักร่วม) เฉพาะกับชุดมิติที่สองและสาม (ชุดมิติแรก เป็นช่องลักษณะสำคัญที่ไม่มีความสัมพันธ์ในเชิงลำดับ). 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

จงคำนวณขนาดของสนามรับรู้ของหน่วยย่อยในชั้นสุดท้ายของกรณีต่อไปนี้

- (ก) โครงข่ายคอนโวโลชันหนึ่งชั้น ที่ใช้ฟิลเตอร์ขนาด 5×5 ขนาดก้าวย่างเป็น 1×1 , เป็น 2×2 และเป็น 3×3 ตามลำดับ.
- (ข) โครงข่ายคอนโวโลชันสองชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด 5×5 ขนาดก้าวย่างเป็น 1×1 .
- (ค) โครงข่ายคอนโวโลชันสองชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด 11×11 ขนาดก้าวย่างเป็น 1×1 .
- (ง) โครงข่ายคอนโวโลชันสามชั้น ทั้งสองชั้นใช้ฟิลเตอร์ขนาด 5×5 ขนาดก้าวย่างเป็น 1×1 .
- (จ) โครงข่ายคอนโวโลชันห้าชั้น โดยฟิลเตอร์ชั้นแรก 11×11 ก้าวย่าง 1×1 , ฟิลเตอร์ชั้นสอง 5×5 ก้าวย่าง 1×1 , ฟิลเตอร์ชั้นสามถึงห้าใช้ฟิลเตอร์แบบเดียวกัน คือ 3×3 ก้าวย่าง 1×1 .
- (ฉ) โครงข่ายคอนโวโลชันสิบชั้น โดยทุกชั้นใช้ฟิลเตอร์แบบเดียวกัน คือ 3×3 ก้าวย่าง 1×1 .
- (ช) โครงข่ายคอนโวโลชันสามชั้น ชั้นที่หนึ่งและสามใช้ฟิลเตอร์ขนาด 3×3 ขนาดก้าวย่างเป็น 1×1 แต่ชั้นที่สองใช้ฟิลเตอร์ขนาด 2×2 ก้าวย่าง 2×2 .

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 สำหรับคอนโวโลชันสองมิติ จงเขียนสมการคำนวณแผนที่ลักษณะสำคัญ (เอาร์พุต) ของชั้นคอนโวโลชัน สำหรับ

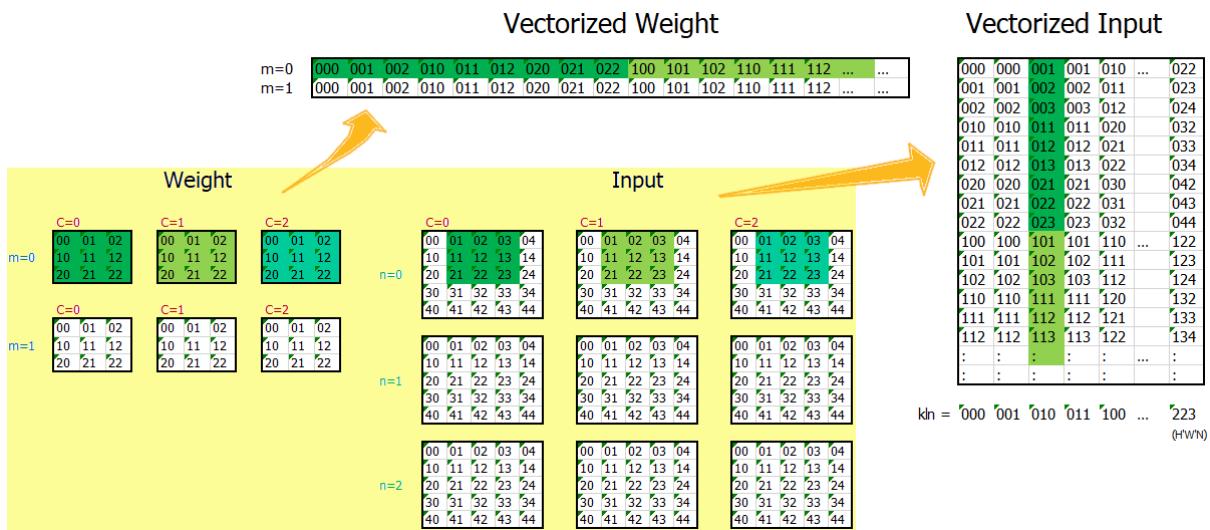
- (ก) คอนโวโลชันหนึ่งมิติ (มีชุดลำดับมิติชุดเดียว อินพุต $\mathbf{X} \in \mathbb{R}^{C \times H}$ โดยชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ).

- (ช) ค่อนโวลูชันสามมิติ (มีชุดลำดับมิติสัมพันธ์สามชุด อินพุต $\mathbf{X} \in \mathbb{R}^{C \times H \times W \times D}$ โดยชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ).
 - (ค) ค่อนโวลูชันลีบมิติ (มีชุดลำดับมิติสัมพันธ์สี่ชุด อินพุต $\mathbf{X} \in \mathbb{R}^{C \times H \times W \times D \times E}$ โดยชุดมิติแรกไม่มีความสัมพันธ์เชิงลำดับ).

การโปรแกรมตระกูลของโครงข่ายคอนโวโลจุชั่น. การคำนวณของโครงข่ายคอนโวโลจุชั่น ประกอบด้วย การคำนวณของชั้นคำนวณสามชนิดหลัก ๆ ได้แก่ ชั้นคำนวณคอนโวโลจุชั่น ชั้นดึงรวม และชั้นเชื่อมต่อ เต็มที่. รายการ 6.1 แสดงตัวอย่างโปรแกรมของชั้นคำนวณคอนโวโลจุชั่น. โปรแกรมในรายการ 6.1 อาศัย การจัดเรียงเทนเซอร์ใหม่ และใช้ประโยชน์จากการคูณเมทริกซ์. รูปที่ 6.15 แสดงแนวคิด การจัดเรียง เทนเซอร์ใหม่ เพื่อที่การคูณเมทริกซ์จะให้ผลลัพธ์เสมือนการคำนวณคอนโวโลจุชั่น. สังเกต สมการ 6.10 เอาร์พุต a เป็นเทนเซอร์สัดส่วน $M \times H' \times W'$ (เมื่อ M เป็นจำนวนลักษณะสำคัญ และ H' กับ W' เป็นขนาดความสูงและกว้างของแพนที่เอาร์พุต) สำหรับจุดข้อมูลแต่ละจุด. ดังนั้น สำหรับจุดข้อมูล หมู่ขนาด N ผลลัพธ์จะเป็นเทนเซอร์สัดส่วน $N \times M \times H' \times W'$. เอาร์พุต จากการคูณเมทริกซ์ $\mathbf{W}_{M \times C \cdot H_f \cdot W_f} \cdot \mathbf{X}_{C \cdot H_f \cdot W_f \times H' \cdot W' \cdot N}$ จะเป็นเมทริกซ์ขนาด $M \times H' \cdot W' \cdot N$ ซึ่งสามารถนำมารับ เรียงเป็นเทนเซอร์สัดส่วน $N \times M \times H' \times W'$ ได้.

หมายเหตุ การเขียนโปรแกรมคำนวณค่อนโวลุชั่น เช่น สมการ 6.10 ด้วยการวนลูป ก็สามารถทำได้ แต่การทำงานอาจทำได้ช้ามาก. ผู้อ่านสามารถทดลองวิธีการเขียนโปรแกรมหลาย ๆ แนวทางและเปรียบเทียบข้อดีข้อเสีย ในแต่ต่าง ๆ เช่น ประสิทธิภาพการทำงาน ความยากง่ายในการแก้ไขและปรับปรุง.

รายการ 6.1: ตัวอย่างโปรแกรมชั้นคำนวณคอนโทรลชั้น



รูปที่ 6.15 ตัวอย่างการจัดเทนเซอร์ของค่าน้ำหนักและอินพุตให้อยู่ในรูปที่การคูณเมทริกซ์สมீองการคำนวนเทนเซอร์. 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 3×3) แสดงด้วยกรอบหกกรอบจัดเรียงเป็นสองแถว (ตามจำนวนฟิลเตอร์) และสามสดมภ์ (ตามจำนวนช่อง). ตัวเลขภายในกรอบแสดงค่าดัชนีเขียวพื้นที่ในแนวตั้งและแนวนอน (อาจเรียกเป็น (i, j) แต่ไม่ได้ระบุในภาพ). อินพุตซึ่งเป็นเทนเซอร์สีลำดับชั้น $x \in \mathbb{R}^{3 \times 3 \times 5 \times 5}$ (สามจุดข้อมูล แต่ละจุดข้อมูลมีสามช่อง แต่ละช่องขนาด 5×5) ใช้ตัว n แสดงดัชนีของจุดข้อมูล. พื้นที่แรงงานสีเขียว (เขียวแก่ เขียวอ่อน และเขียวไข่กา) แสดงความสัมพันธ์ของการคำนวนในก้าวย่างที่สองตามแนวนอน (อาจเรียกเป็น $(k, l) = (0, 1)$ เมื่อ ขนาดก้าวย่าง $S = 1$). ลูกศร เขี้ยวเมเทนเซอร์สีลำดับชั้นในรูปเดิม กับรูปแบบใหม่ที่สีดวงต่อการคำนวน. แต่ละแ夸ของเมทริกซ์ของค่าน้ำหนัก แทนฟิลเตอร์แต่ละตัว (ดัชนี m ช่วยระบุ). แต่ละค่าภายในแล้ว แสดงด้วยดัชนีสามตัว (c, i, j) สำหรับช่อง ตำแหน่งแนวตั้ง และตำแหน่งแนวนอน. สำหรับเมทริกซ์ของอินพุตที่จัดใหม่ แต่ละสดมภ์ แทนตำแหน่งของเอต์พุตและจุดข้อมูล (ดัชนี (k, l, n) สำหรับตำแหน่งเอต์พุตแนวตั้ง แนวนอน และจุดข้อมูลที่ n^h). ภายในสดมภ์ แสดงดัชนีของช่อง ตำแหน่งแนวตั้ง และตำแหน่งแนวนอนของอินพุตเดิม (c, i, j) . สังเกต การจัดเรียงดัชนีในเมทริกซ์อินพุต ที่ทำให้การคูณเมทริกซ์เป็นสมீองการคำนวนคอนเวอลูชัน.

```

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 + \
45                         simplified_w.mm(simplified_z)
46
47     # Restructure convoluted output back
48     conv_out = simplified_out.view(M, Ho, Wo, N)
49     a = conv_out.permute(3, 0, 1, 2) # output (N,M,H',W')
50
51     return a
52
53
54     @staticmethod
55     def _simplify_struct(z, Hf, Wf, S, P):
56         ...
57
58         Collapse z structure such that convolution can be
59         efficiently computed as matrix multiplication.
60         ...
61
62         # Zero-pad the input (on last 2 dimensions)
63         zhat = torch.nn.functional.pad(z, (P,P,P,P,0,0,0,0),
64                                       'constant', 0)
65
66         # Get vectorized indices
67         c, rx, cx = MyConv2D._get_simplified_indices(z.shape,
68                                                       Hf, Wf, S, P)
69         # c.shape = (C Hf Wf, 1)
70         # rx.shape = (C Hf Wf, Ho Wo)
```

```

70      # cx.shape = (C Hf Wf, Ho Wo)
71
72      # Re-arrange input into a simplified structure
73      simz = zhat[:, c, rx, cx] # shape (N, C Hf Wf, Ho Wo)
74
75      num_channels = z.shape[1]
76
77      sim_z = simz.permute(1, 2, 0).contiguous().view(\n
78          num_channels * Hf * Wf, -1)
79
80      return sim_z # shape = (C Hf Wf, Ho Wo N)
81
82      @staticmethod
83      def _get_simplified_indices(input_shape, Hf, Wf, S, P):
84          ...
85          return indices of re-arranged vector ready for
86          dot operation (in lieu of convolution).
87          ...
88
89      N, C, H, W = input_shape
90
91      # Determine output size
92      Ho = int( (H + 2*P - Hf)/S ) + 1
93      Wo = int( (W + 2*P - Wf)/S ) + 1
94
95      # To match the re-arranged weight,
96      # input must be re-arranged accordingly.
97      # weight row: f = 0, 1, ..., (M-1)
98      # weight column: cij = 000, 001, 002, 010, 011, ...
99      # Thus, input row: cij
100     # input column: k, l = 00, 01, 02, ..., (Ho-1)(Wo-1)
101
102     # Work out indices of filter nodes
103     # j, i, c from innermost to outermost along row
104     j = np.tile(np.arange(Wf), Hf * C).reshape(-1, 1)
105     # e.g., j = [0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, ...]
106     i = np.tile(np.repeat(
107                 np.arange(Hf), Wf), C).reshape(-1, 1)
108     # e.g., i = [0, 0, 0, 1, 1, 1, 2, 2, 2, 0, 0, 0, ...]
109     c = np.repeat(np.arange(C), Hf*Wf).reshape(-1, 1)

```

```

110      # e.g., c = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, ...]
111
112      # Work out indices of output nodes
113      # L, k from innermost to outermost, along column
114      l = np.tile(np.arange(Wo), Ho).reshape(1, -1)
115      # e.g., L.T = [0,1,2,3,0,1,2,3,0,1,2,3, ...]
116      k = np.repeat(np.arange(Ho), Wo).reshape(1, -1)
117      # e.g., k.T = [0,0,0,0,1,1,1,1,2,2,2,2, ...]
118
119      # Indices of input nodes
120      rx = S * k + i # shape = (C Hf Wf, Ho Wo)
121      cx = S * l + j # shape = (C Hf Wf, Ho Wo)
122
123      return c.astype(int), rx.astype(int), cx.astype(int)

```

โปรแกรมในรายการ 6.2 แสดงตัวอย่างการเรียกใช้ MyConv2D. โปรแกรม MyConv2D เขียนขึ้นตามรูปแบบของไฟล์orch `nn.Conv2d` ดังนั้น การใช้งานก็ทำในลักษณะเดียวกันได้. โปรแกรมในรายการ 6.3 และ 6.4 แสดงตัวอย่างฝึกและทดสอบโครงข่าย (ค่าอภิมานพารามิเตอร์ต่าง ๆ ใช้ได้ดีกับชุดข้อมูลเอมนิสต์. ดูแบบฝึกหัด 5.10 สำหรับตัวอย่างการนำเข้าชุดข้อมูลเอมนิสต์).

รายการ 6.2: ตัวอย่างการเรียกใช้ชั้นคำนวนคอนโกลุชั้น 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
19         return a6

```

```

20 net = NetConv1().to(device)
21 loss_fn = torch.nn.CrossEntropyLoss()
22 optimizer = optim.Adam(net.parameters(), lr=0.001)

```

รายการ 6.3: การฝึกโครงข่ายที่ใช้ชั้นคำนวนคอนโวลูชัน 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: การทดสอบโครงข่ายที่ใช้ชั้นคำนวนคอนโวลูชัน 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 แล้วทดสอบการทำงานเปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.Conv2d` รวมถึงทดสอบโครงสร้างแบบอื่น ๆ (เปลี่ยนค่าอภิมานพารามิเตอร์ เช่น ขนาดฟิลเตอร์ จำนวนฟิลเตอร์ ขนาดก้าวบ่าง จำนวนการเติมเต็มด้วยศูนย์) อย่างรายละเอียด ในการใช้โปรแกรมสำเร็จรูปจะสะดวกกว่า การอ้างอิงก็ทำได้ง่ายกว่า ถูกยอมรับดีกว่า (โปรแกรมมาตรฐาน เชื่อว่าได้รับการตรวจสอบมาดีกว่า) และดังเช่นที่จะได้เห็นจากการทดลอง ในกรณีนี้ โปรแกรมสำเร็จรูป `nn.Conv2d` ทำงานมีประสิทธิภาพมากกว่าอย่างเห็นได้ชัด (การเขียนโปรแกรมประสิทธิภาพสูง อาจต้องอาศัยการโปรแกรมระดับล่าง ซึ่งอยู่นอกเหนือขอบเขตของหนังสือเล่มนี้). แต่การศึกษาโปรแกรมในรายการ 6.1 ทำเพื่อให้เข้าใจกลไกการทำงานของชั้นคำนวณคอนโวลูชันอย่างกระจุ่งแจ้ง.

แบบฝึกหัด 6.8

การเขียนโปรแกรมชั้นเชื่อมต่อเติมที่กีสามารถทำได้ในลักษณะเดียวกัน. รายการ 6.6 แสดงตัวอย่างโปรแกรมเชื่อมต่อเติมที่ที่เขียนการแพร่กระจายย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `fcf` ที่เขียนดังในรายการ 6.5. การใช้งานชั้นเชื่อมต่อเติมที่ `MyFCBack` ก็ทำเช่นเดียวกับการเรียกใช้ชั้นคำนวณ `nn.Linear` เช่น การเปลี่ยนบรรทัด `self.fc1 = nn.Linear(8*7*7, 10)` ในรายการ 6.2 เป็น `self.fc1 = MyFCBack(8*7*7, 10)` เท่านั้นที่เหลือกีสามารถดำเนินงานสร้างโครงสร้างข่าย 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):  $\partial w_{ji}^{(v)}$ 
5             , b (Mo,1):  $\partial b_j^{(v)}$ 
6             # output: a (N,Mo):  $a_j^{(v)}$ 
7             zT = torch.transpose(zp, 0, 1)
8             a = w.mm(zT) + b
9             ctx.save_for_backward(zp, w, b)
10            return torch.transpose(a, 0, 1)
11
12        @staticmethod
13        def backward(ctx, dEa):
14            # 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:
            $\frac{\partial E}{\partial w_{ji}^{(v)}} = \frac{\partial E}{\partial a_j^{(v)}} \cdot \frac{\partial a_j^{(v)}}{\partial w_{ji}^{(v)}}$  , dEb:  $\frac{\partial E}{\partial b_j^{(v)}} = \frac{\partial E}{\partial a_j^{(v)}} \cdot \frac{\partial a_j^{(v)}}{\partial b_j^{(v)}}$ 
15      N, _ = dEa.shape
16      zp, w, b = ctx.saved_tensors
17      dEzp = dEa.mm(w)
18      dEw = torch.transpose(dEa, 0, 1).mm(zp)
19      dEb = torch.transpose(dEa,
20                           0, 1).mm(torch.ones(N,1).to(dEa.device))
21      return dEzp, dEw, dEb

```

รายการ 6.6: ตัวอย่างโปรแกรมชั้นเชื่อมต่อเติมที่ที่เขียนการแพร่กระจายย้อนกลับของ MyFCBack. ตัวอย่างนี้ เรียกใช้ พงก์ชั้น fcf 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(
8             num_features, input_channels)-sqk
9         initb = 2*sqk*torch.rand(num_features,1) - sqk
10        self.weight = nn.Parameter(initw)
11        self.bias = nn.Parameter(initb)
12        self.fcf = fcf.apply
13
14    def forward(self, z):
15        a = self.fcf(z, self.weight, self.bias)
16        return a

```

จงทดสอบการทำงานของชั้นเชื่อมต่อเติมที่ MyFCBack เปรียบเทียบกับโปรแกรมสำเร็จรูป nn.Linear ทั้งในเชิงการทำงาน และเวลาในการทำงาน. รวมถึง จงทดลองแก้การคำนวณในพงก์ชั้น fcf เพื่อตรวจสอบดูว่าการคำนวณการเชื่อมต่อและการคำนวณแพร่กระจายย้อนกลับ ว่า ได้ทำผ่าน fcf.forward และ fcf.backward จริง. ตัวอย่างเช่น ทดลองแก้บรรทัดคำสั่ง return dEzp, dEw, dEb เป็น return 0*dEzp, 0*dEw, 0*dEb และ สังเกตผล. สรุป และอภิปราย.

หมายเหตุ แม้การเขียนโปรแกรมชั้นเชื่อมต่อเติมได้ถูกอภิปรายไปแล้วในหัวข้อ 5.7 การทบทวนอีกครั้งในแบบฝึกหัด เพื่อให้คุณเคยกับรูปแบบการเขียนโปรแกรมชั้นคำนวณเพื่อใช้กับไฟฟอร์ช ที่ระบบการ

คำนวณการแพร่กระจายย้อนกลับด้วย การทบทวนนี้ จะคาดว่าจะช่วยผู้อ่านเข้าใจกลไกของการเขียนโปรแกรมชั้นคุณพร้อมการระบุการแพร่กระจายย้อนกลับของไฟฟอร์ช ก่อนที่จะเขียนโปรแกรมชั้นตอนโวลูชัน ซึ่งจะบัญชีในแบบฝึกหัด 6.9.

แบบฝึกหัด 6.9

คล้ายกับแบบฝึกหัด 6.8 แบบฝึกหัดนี้ศึกษาการเขียนโปรแกรมชั้นตอนโวลูชันทั้งการคำนวณ และ การแพร่กระจายย้อนกลับ. รายการ 6.8 แสดงตัวอย่างโปรแกรมชั้นตอนโวลูชันที่เขียนการแพร่กระจาย ย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `convf` ที่เขียนดังในรายการ 6.7⁵.

โปรแกรมชั้นตอนโวลูชัน `MyConv2DB` รับมารดมมาจาก `MyConv2D` (รายการ 6.1) เพื่อลดความซ้ำซ้อน ที่จะต้องกำหนดค่าเริ่มต้นค่าน้ำหนัก (ภายในเมธอด `__init__`). การใช้งานชั้นตอนโวลูชัน `MyConv2DB` ก็ทำเช่นเดียวกับ `MyConv2D` เช่น การเปลี่ยน `self.conv1 = MyConv2D(1, 16, 5, 1, 2)` และ `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: ตัวอย่างโปรแกรมการคำนวณตอนโวลูชันพร้อมการแพร่กระจายย้อนกลับ `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         ...
8         F, C, Hf, Wf = w.shape
9         N, D, H, W = zp.shape
10        assert C == D, 'Numbers of channels are not matched.'
11
12        # Determine output size
13        Ho = int((H + 2*P - Hf)/S) + 1
14        Wo = int((W + 2*P - Wf)/S) + 1
15
16        # Simplify z structure
17        simplified_z = MyConv2D._simplify_struct(zp,
18                                              Hf, Wf, S, P)

```

⁵ รหัสโปรแกรมนี้ดัดแปลงจากการรหัสโปรแกรมชิปสเตอร์เน็ต (`Hipsternet`), จาก <https://github.com/wiseodd/hipsternet/tree/master/hipsternet>, ปรับปรุงล่าสุด 12 ก.พ. 2017.

```

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     #          dEw (F, C, Hf, Wf)
41     #          dEb (F, 1)
42     N, F, Ho, Wo = dEa.shape
43     zp, w, b, tensorSP, simplified_z = ctx.saved_tensors
44     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)
51
52     # Restructure dEa from (N, F, H', W') to (F, H' W' N)
53     simplified_dEa = dEa.permute(
54         1,2,3,0).contiguous().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      # First, sum over the feature axis
62      simplified_w = w.view(F,-1)
63      assert simplified_w.shape == (F, C * Hf * Wf)
64
65      wdEa_overF = simplified_w.transpose(0,1).mm(
66                      simplified_dEa) #(C Hf Wf, H'W'N)
67
68      # Sum over spatial indices
69      dEzp = convf.sum_omega(wdEa_overF,zp.shape,Hf,Wf,P,S)
70
71      return dEzp, dEw, dEb, None, None
72
73  @staticmethod
74  def sum_omega(prod_overF, zpshape, Hf, Wf, P, S):
75      """
76      Summation over the two omega sets (~over H' and W')
77      input: prod_overF (C Hf Wf, H' W' N)
78      output: dEzp (N, C, H, W)
79      """
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(
87                      2, 0, 1).cpu().numpy()
88
89      # Prepare result structure
90      sum_result = np.zeros((N,C,H_hat,W_hat),
91                           dtype=prod.dtype)
92
93      # Get vectorized indices
94      c, rx, cx = MyConv2D._get_simplified_indices(zpshape,
95                                                     Hf, Wf, S, P)
96      # c.shape = (C Hf Wf, 1)
97      # rx.shape = (C Hf Wf, H' W')
```

```

98     # cx.shape = (C Hf Wf, H' W')
99
100    # Sum over omega using np.add.at mechanism
101    np.add.at(sum_result, (slice(None), c, rx, cx), prod)
102    tsum = torch.tensor(sum_result).to(prod_overF.device)
103
104    if P != 0:
105        # remove side effect from padding
106        return tsum[:, :, P:-P, P:-P]
107
108    return tsum

```

รายการ 6.8: ตัวอย่างโปรแกรมชั้นตอนโวลูชันที่เขียนการแพร์กระกระจายย้อนกลับของ 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
12     return a

```

จงทดสอบชั้นคำนวณ MyConv2DB ทั้งในเชิงผลการทำงาน และประสิทธิภาพการทำงาน (วัดเวลาทำงาน) รวมถึงทดสอบว่า การแพร์กระกระจายย้อนกลับทำผ่าน `convf.backward` จริง (ดูแบบฝึกหัด 6.8 ประกอบ). แล้วเปรียบเทียบกับโปรแกรมสำเร็จรูป `nn.Conv2d`.

หมายเหตุ การเขียนโปรแกรมเองในที่นี้เพื่อความกระจ่างในการทำงาน แต่ในทางปฏิบัติแนะนำให้ใช้โปรแกรมสำเร็จรูป ด้วยเหตุผลด้านความสะดวก ประสิทธิภาพ การทดสอบที่ดีและครอบคลุมกว่า รวมถึงความยอมรับและความไว้วางใจของผู้เกี่ยวข้อง.

แบบฝึกหัด 6.10

แบบฝึกหัดนี้ศึกษาการเขียนโปรแกรมชั้นดึงรวมแบบมากที่สุด ทั้งการคำนวณ และการแพร์กระกระจายย้อนกลับ. รายการ 6.10 แสดงตัวอย่างโปรแกรมชั้นดึงรวมแบบมากที่สุด ที่เขียนการแพร์กระกระจายย้อนกลับเอง โดยการคำนวณจริงทำผ่านการเรียกฟังก์ชัน `maxpoolf` ที่เขียนดังในรายการ 6.9⁶.

⁶ รหัสโปรแกรมนี้ตัดแปลงจากการหัสดีสเปร์แกรมชิปสเตอร์เน็ต (*HipsterNet*), จาก <https://github.com/wiseodd/>

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. ส่วนที่เหลือ ก็สามารถดำเนินงานสร้างโครงข่าย ฝึก และทดสอบได้เช่นเดิม.

รายการ 6.9: ตัวอย่างโปรแกรมการคำนวณชั้นดึงรวมแบบมากที่สุดพร้อมการแพร์กยะจายย้อนกลับ `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'):    $z_{c,k,l}^{(v)}$ 
7          """
8
9      N, C, H, W = zp.shape
10
11     # Determine output size
12     Ho = int((H + 2*P - Hf)/S) + 1
13     Wo = int((W + 2*P - Wf)/S) + 1
14
15     # Restructure zp
16     # Operation effect of pooling is different from conv
17     # such that channel/feature axis is treated
18     # independently (like datapoint axis).
19
20     restrict_z = zp.view(N * C, 1, H, W)
21     sim_z = MyConv2D._simplify_struct(restrict_z,
22                                         Hf, Wf, S, P)
23     assert sim_z.shape == (Hf * Wf, Ho * Wo * N * C)
24
25     # Perform pooling function
26     # poolz, pool_cache = pool_func(sim_z)
27
28     max_idx = torch.argmax(sim_z, dim=0)
29     poolz = sim_z[max_idx, range(max_idx.size()[0])][]
30     pool_cache = max_idx
31

```

```

32     # Restructure pooling output
33     zpool = poolz.view(Ho, Wo, N, C)
34     z = zpool.permute(2, 3, 0, 1).contiguous()
35
36     ctx.save_for_backward(zp, torch.tensor([Hf,Wf,S,P]),
37                           sim_z, pool_cache)
38
39     return z
40
41     @staticmethod
42     def backward(ctx, dEz):
43         # input: dEz (N, F, H', W'):    $\hat{\delta}_{frs}^{(v)}$ 
44         # output: dEzp (N, F, H, W):    $\hat{\delta}_{fkl}^{(v-1)}$ 
45         zp, tensorHfWfSP, sim_z, pool_cache = ctx.saved_tensors
46         Hf = tensorHfWfSP[0].item()
47         Wf = tensorHfWfSP[1].item()
48         S = tensorHfWfSP[2].item()
49         P = tensorHfWfSP[3].item()
50
51         N, F, H, W = zp.shape
52
53         sim_dEa = torch.zeros(sim_z.shape).to(zp.device)
54         sim_dEz = dEz.permute(2,3,0,1).contiguous().view(-1,)
55
56         # Perform dpooling function
57         # sim_dEa = dpool_func(poolz, pool_cache)
58         sim_dEa[pool_cache, range(pool_cache.size()[0])] =
59                         sim_dEz      # (Hf Wf, H' W' N F)
60
61         # Sum over spatial indices
62         dEzp_sim = convf.sum_omega(sim_dEa,
63                                     (N*F, 1, H, W), Hf, Wf, P, S)
64         dEzp = dEzp_sim.view(zp.shape)
65
66     return dEzp, None, None, None, None

```

รายการ 6.10: ตัวอย่างโปรแกรมชั้นชั้นดึงรวมแบบมากที่สุดที่เขียนการแพร่กระจายย้อนกลับของ 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 การแพร่กระจายย้อนกลับทำผ่าน **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 ด้วยการพิจารณา ซึ่งสูงส่งที่สุด.

สอง ด้วยการเลียนแบบ ซึ่งง่ายที่สุด.

สาม ด้วยประสบการณ์ ซึ่งขึ้นที่สุด.”

—ชنجื้อ

โครงข่ายคอนโวลูชัน เหมาะกับข้อมูลที่มีลักษณะเชิงท้องถิ่นสูง. ในทางปฏิบัติ ข้อมูลหลาย ๆ ชนิด มีลักษณะเชิงท้องถิ่นสูง รวมถึงข้อมูลเชิงทัศนะ เช่น ภาพ และวิดีโอ. โครงข่ายคอนโวลูชันได้รับการประยุกต์ใช้อย่างกว้างขวางกับข้อมูลเชิงทัศนะ ไม่ว่าจะเป็น การจำแนกภาพ (image classification[113, 186, 195, 85, 92]), การตรวจจับวัตถุในภาพ (object detection[160, 161, 164]), การแบ่งส่วนภาพตามความหมาย (semantic segmentation[122]), การตรวจจับท่าทาง (pose detection[28]), การรู้จำใบหน้า (face recognition[179]), 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 อาศัยความคิดสร้างสรรค์และความเข้าใจในการภารกิจ ทฤษฎี กลไกการทำงานของการเรียนรู้ของเครื่อง และในหลาย ๆ ครั้ง ได้พัฒนาทฤษฎีเฉพาะขึ้นมา. การประยุกต์ใช้ที่น่าสนใจ มีมากมายและยังมีการพัฒนาอย่างต่อเนื่อง บทนี้ เลือกอภิปรายบางส่วนของการประยุกต์ใช้ที่น่าสนใจ เพื่อให้เห็นตัวอย่างของความคิดสร้างสรรค์ในการประยุกต์ใช้โครงข่ายคอนโวลูชัน.

7.1 การตรวจจับวัตถุในภาพ

การตรวจจับวัตถุในภาพ (object detection) เป็นภารกิจที่รับอินพุตเป็นภาพ และให้อ乌ตพุต เป็นตำแหน่งของวัตถุที่พบในภาพ พร้อมชนิดของวัตถุ. โดยทั่วไป ตำแหน่งของวัตถุ จะระบุด้วยกล่อง

การจำแนกภาพ (image classification)



output: "Baby Kangaroo"

การตรวจจับวัตถุในภาพ (object detection)



output: $(x, y, w, h, \text{"Baby Kangaroo"})$

รูปที่ 7.1 การจำแนกภาพ และการตรวจจับวัตถุในภาพ. ภาพข้างมือแสดงผลลัพธ์จากการจำแนกภาพ ซึ่งจะระบุแค่ชนิดของวัตถุหลักในภาพ. ภาพขวาเมื่อแสดงผลลัพธ์จากการตรวจจับวัตถุในภาพ ซึ่งนอกจากระบุชนิดของวัตถุแล้วยังต้องระบุตำแหน่งด้วย. กรอบสีเขียวในภาพขวา คือ กล่องขอบเขต ซึ่งมักถูกระบุด้วยพิกัด (แนวอนและแนวตั้ง x, y) และขนาด (ความกว้างและความสูง w, h).

ขอบเขต (*bounding box*) ซึ่งอาจอ้างอิงถึงด้วยพิกัดแนวอนและแนวตั้งของจุดศูนย์กลางของกล่องขอบเขต และความกว้างกับความสูงของกล่องขอบเขต.

รูปที่ 7.1 แสดงผลลัพธ์ของการตรวจจับวัตถุในภาพ (ภาพข่าย) เปรียบเทียบกับการจำแนกภาพ (image classification ในภาพขวา). การจำแนกภาพ เป็นภาระกิจที่รับอินพุตเป็นภาพ และให้เอาต์พุตเป็นชนิดของวัตถุหลักในภาพ ส่วนการตรวจจับวัตถุในภาพ จะเพิ่มภาระบุต้ามแน่นของวัตถุออกมามากด้วย.

การจำแนกภาพ ไม่มีภาระบุต้ามแน่นของวัตถุภายในภาพ และมักถูกตีกรอบปัญหาเป็นปัญหาการจำแนกประเภท (multi-class classification). แบบจำลองของการจำแนกภาพ นิยมใช้โครงข่ายคอนโวโลชัน ที่โครงสร้างเอาต์พุตใช้ชั้นเชื่อมต่อเต็มที่ตามด้วยฟังก์ชันซอฟต์แมกซ์ ดังเช่น แบบจำลองอเล็กซ์เน็ต ที่ได้อธิบายในหัวข้อ 6.5.

การตรวจจับวัตถุในภาพ อาจทำได้หลายวิธี. วิธีแบบดั้งเดิม ใช้การจำแนกประเภท ร่วมกับเทคนิคหน้าต่างเลื่อน ดังอธิบายในหัวข้อ 4.1 (หรือเทคนิคอื่นในลักษณะเดียวกัน [164, 69]). หนึ่งในศาสตร์และศิลป์ของการตรวจจับวัตถุในภาพ คือ โยโล (YOLO[160]) ซึ่งเป็นระบบตรวจจับวัตถุในภาพแบบเวลาจริง (real-time object detection) และโยโลยังมีการทำงานภายใน ที่ใช้โครงสร้างกลไกหลักอย่างเดียว ทำให้การปรับแต่งประสิทธิภาพสามารถทำได้ง่าย.

โยโล. โยโล[160, 161]รับอินพุตเป็นภาพสี และให้เอาต์พุตออกมาเป็นกล่องขอบเขตบุต้ามแน่นของวัตถุที่ตรวจจับได้ในภาพ พร้อมชนิดของวัตถุที่พบ. โยโล ใช้โครงข่ายคอนโวโลชันในการแปลงจากอินพุตไปเป็นเอาต์พุต. ต่างจากระบบการตรวจจับวัตถุในภาพแบบดั้งเดิม โยโล ตั้งปัญหาการตรวจจับวัตถุในภาพ เป็นการหาค่าดัชน้อย เพื่อท่านายตำแหน่งของวัตถุ และการจำแนกประเภท เพื่อท่านายชนิดของวัตถุที่พบ.

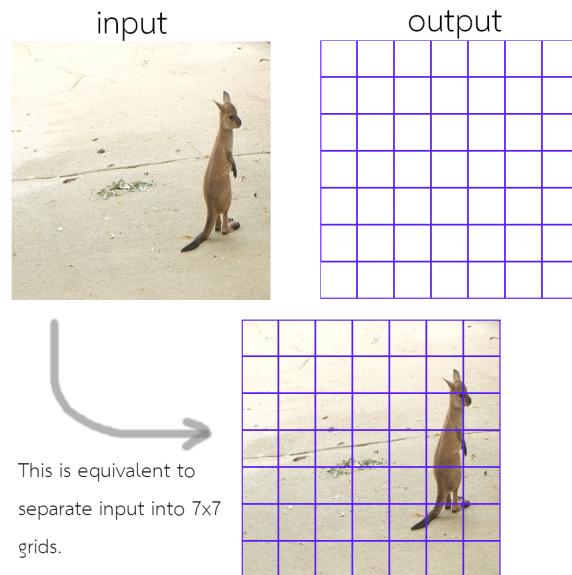
แนวทางคือ เพื่อสามารถตรวจจับตำแหน่งวัตถุได้สูงสุด M วัตถุต่อภาพ เราต้องการเอาต์พุตเป็นเทนเซอร์ขนาดอย่างน้อย $5M$ นั่นคือ สำหรับแต่ละการตรวจจับตำแหน่งวัตถุ โยโล่จะใช้ 5 ค่า เพื่อระบุด้วยพิกัดของจุดศูนย์กลางและขนาดของกล่องขอบเขต (x, y, w, h สำหรับพิกัดแนวอนและแนวตั้ง ความกว้างและความสูง) พร้อมด้วยค่าความมั่นใจ (confidence ใช้สัญกรณ์ C) ว่าภายในกล่องมีวัตถุอยู่. ค่าความมั่นใจนี้ มีเพื่อที่ช่วยให้ผลการทำนายสามารถยืดหยุ่นจำนวนวัตถุในภาพได้ ตั้งแต่ 0 วัตถุ (ทุกตำแหน่งตรวจจับ มีค่าความมั่นใจต่ำมาก) ไปจนถึง M วัตถุ (ทุกตำแหน่งตรวจจับ มีค่าความมั่นใจสูงมาก). อาจมองได้ว่า ค่าความมั่นใจ ทำหน้าที่เป็นเหมือนสวิตซ์ปิดเปิดกล่องขอบเขต ว่าจะเลือกผลลัพธ์กล่องไหนบ้างให้ออกไป.

เพื่อให้การฝึกโครงข่ายทำได้อย่างมีประสิทธิภาพ โยโล่ กำหนดพื้นที่รับผิดชอบของแต่ละตำแหน่งตรวจจับ. การกำหนดพื้นที่รับผิดชอบของตำแหน่งตรวจจับ จำนวน M ตำแหน่งตรวจจับ เทียบเท่ากับการแบ่งพื้นที่รับผิดชอบของภาพอินพุตออกเป็น M ส่วน. โยโล่แบ่งพื้นที่ภาพตามแนวอนและแนวตั้งอย่างละเอียด ๆ กัน และเรียกการแบ่งนี้เป็นเสมือนช่องตาราง หรือ กริด (grid) และเรียกพื้นที่รับผิดชอบแต่ละส่วนว่า กริดเซลล์ (grid cell). รูปที่ 7.2 แสดงแนวคิดนี้ ในรูปแสดงการแบ่งรูปออกเป็น 7×7 ส่วน ($M = 49$). การกำหนดกริดเซลล์ให้รับผิดชอบพื้นที่อินพุตส่วนไหน ช่วยให้การฝึกทำได้มีประสิทธิภาพมากขึ้น โดยลดความสับสนว่า วัตถุควรจะถูกทายด้วยกริดเซลล์ไหน. มันจะช่วยให้ การกำหนดฟังก์ชันสูญเสีย และการทำเฉลย ทำได้ง่ายขึ้น เพราะถ้าไม่กำหนดความรับผิดชอบให้แน่นอน กริดเซลล์ใด ๆ หนึ่งใน M กริดเซลล์ อาจทายวัตถุก็ได้ และการคำนวณค่าฟังก์ชันสูญเสียจะยุ่งยากมาก. (ในระหว่างการฝึก ซึ่งการทำลายอาจจะยังผิดเพี้ยนอยู่มาก มันจะยากที่จะรู้ว่ากริดเซลล์ไหนที่กำลังทายเฉลยวัตถุไหน และยังอาจมีกรณีที่ กริดเซลล์มากกว่าหนึ่งตัว พยายามทายวัตถุเดียวกันอีก.)

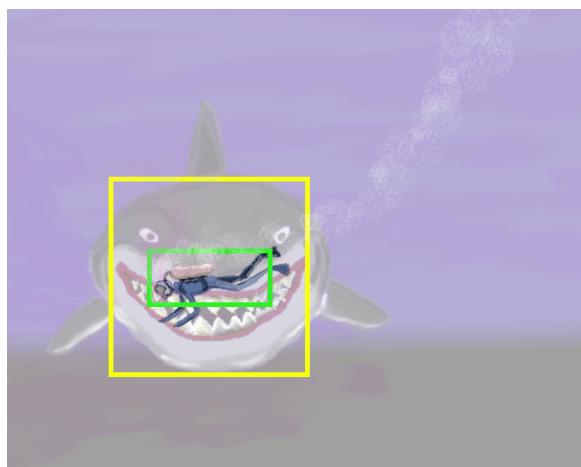
การกำหนดให้แต่ละกริดเซลล์ทายได้เพียงหนึ่งวัตถุ จะจำกัดความสามารถของการตรวจจับภาพวัตถุ โดยเฉพาะกรณีที่วัตถุซ้อนทับกันและมีจุดศูนย์กลางอยู่ใกล้กัน (ทำให้วัตถุกอยู่ในความรับผิดชอบของกริดเซลล์เดียวกัน). หลาย ๆ ครั้ง วัตถุที่ซ้อนทับกันนั้น อาจมีขนาดหรือรูปทรงที่แตกต่างกัน ทำให้ แม้จะทับซ้อนกัน ก็สามารถเห็นวัตถุต่าง ๆ ที่ทับซ้อนกันได้อย่างชัดเจน. รูปที่ 7.3 แสดงตัวอย่างกรณีดังกล่าว.

วิธีแก้ไขเบื้องต้นคือ แทนที่จะให้แต่ละกริดเซลล์ทำนายตำแหน่งวัตถุได้แค่หนึ่งอัน ก็แค่อนุญาตให้หนึ่งกริดเซลล์ทำนายตำแหน่งวัตถุได้หลาย ๆ ตำแหน่ง โดยแต่ละการทำลายก็มีค่าความมั่นใจของตัวเอง. เพียงแต่ ในการฝึก การคำนวณค่าฟังก์ชันสูญเสียจะจัดการให้มีประสิทธิภาพได้อย่างไร. วิธีการที่ออกแบบมาเพื่อบรรเทาประเด็นนี้ คือ เทคนิคกล่องสมอ (anchor box[164]).

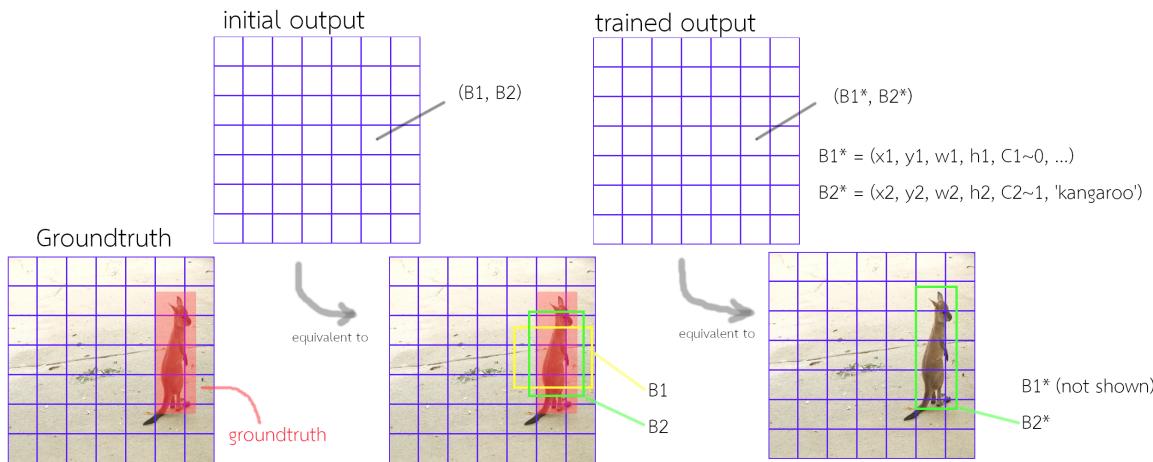
เทคนิคกล่องสมอ. การทายตำแหน่งแต่ละกล่อง ในกริดเซลล์ จะเรียกว่า กล่องสมอ โดยหนึ่งกริดเซลล์ สามารถทำนายกล่องสมอได้ B กล่อง และ กล่องสมอแต่ละกล่อง จะถูกกำหนดค่าเริ่มต้นให้มีขนาดหรือสัดส่วนต่างกัน. การฝึก จะใช้ค่าไอยูรห่วงกล่องสมอทับเฉลย เป็นดัชนีกำหนดว่า กล่องสมอใดจะรับผิดชอบเฉลยวัตถุใด (และอาจมีกฎในการกำหนดความรับผิดชอบในกรณีที่ค่าไอยูรห่วงเอียง



รูปที่ 7.2 ໂປໂລใช้อาต์พุตที่มีโครงสร้างเทียบเท่าการแบ่งส่วนภาพอินพุต. เอาต์พุตมีโครงสร้างเป็น $G \times G$ ส่วน ซึ่งตัวอย่างนี้คือ 7×7 และกำหนดให้แต่ละส่วนแบ่งขอบเขตรับผิดชอบอินพุต ซึ่งเทียบเท่าการแบ่งส่วนภาพอินพุตเป็น 7×7 ส่วน (ໂຢໂລເຮີຍກ ແຕ່ລະສ່ວນວ່າ ກຣິດເຊລ໌). ແຕ່ສ່ວນຄູກຮັບຜິດຂອບດ້ວຍແຕ່ລະກຣິດເຊລ໌ ໂດຍໜຶ່ງກຣິດເຊລ໌ (ໜຶ່ງ “ພິກເຊລ໌” ໃນເອຕົ່ມ) ຈະມີຄ່າຕ່າງໆ ຖໍ່ສໍາຫຼັບຮະບຸຕໍ່ແໜ່ງການຕຽບຈັບແລະໝົດວັດຖຸ.



รูปที่ 7.3 ຕ້າວຍ່າງແສດງກຮົນສໍາຫຼັບເທິກນິກລ່ອງສມອ. ໃນກາພ ນັກດຳນຳທີ່ອູ່ດ້ານໜ້າ ແລະ ຂລາມທີ່ອູ່ດ້ານໜ້າ ມີຈຸດສູນຍົງກລາງຂອງກລ່ອງຂອບເຂດຍູ້ຕໍ່ແໜ່ງເດືອກກັນ. ພາກກຳນົດໃຫ້ການທາຍວັດຖຸສາມາຮັກທໍາໄດ້ເພີ່ງໜຶ່ງວັດຖຸຕ່ອງກຣິດເຊລ໌ ຄວາມສາມາຮັກຂອງຮະບບຈະຄູກຈຳກັດດ້ານໜ້າຢ່າງມາກໃນກຮົນນີ້ເຂັ້ນນີ້. ສັງເກດວ່າ ແນວັດຖຸຈະທັບໜັນກັນແລະມີຈຸດສູນຍົງກລາງເດືອກກັນແຕ່ມີໄດ້ບັງກັນສົນທີ ເນື່ອຈາກວັດຖຸຕໍ່ຕ່າງໆ ທີ່ທັບໜັນກັນ ອາຈນີ້ຂາດຫຼືອຮູປທຽບທີ່ແຕກຕ່າງກັນ. ກລ່ອງຂອບເຂດໃນກາພ ແສດງການທັບໜັນແລະການມີຈຸດສູນຍົງກລາງໃກລ້າເຄີຍກັນ(ອາຈຈະສັງເກດໄດ້ຍາກ) ແຕ່ມີຂາດແລະຮູປທຽບແຕກຕ່າງກັນຂອງວັດຖຸທີ່ທັບໜັນກັນ (ກລ່ອງສື່ເຂົ້າວ ຕຽບຈັບນັກດຳນຳ ສ່ວນກລ່ອງສື່ເລື້ອງຕຽບຈັບຂລາມ).



รูปที่ 7.4 โยโลเลือกกล่องสมอเพื่อรับผิดชอบวัตถุ. ภาพช้ายสุด ถาวรล่าง แสดงภาพอินพุต และส่วนรับผิดชอบต่าง ๆ พร้อมทั้งเฉลย (แสดงด้วยพื้นที่โปร่งใสสีแดง). ในภาพ สังเกต จุดศูนย์กลางของเฉลย จะตกอยู่ภายในกริดเซลล์ที่ห่างจากชัยและสีจากบน และโยโลจะใช้กริดเซลล์นี้รับผิดชอบเฉลย. ภาพชัย ถาวบน แสดงเอาต์พุตขณะก่อนฝึก ที่แต่ละกริดเซลล์จะเป็นเวกเตอร์ทำงานกล่องสมอสองกล่อง (**B1** และ **B2** ในกริดเซลล์ที่รับผิดชอบเฉลย). กล่องสมอแต่ละตัวประกอบด้วยค่าตำแหน่งและขนาดกล่องขอบเขต ค่าความมั่นใจ และชนิดวัตถุ (ชนิดวัตถุ ระบุด้วยค่าความน่าจะเป็นของชนิดต่าง ๆ เตรียมไว้). ภาพล่างกลาง แสดงตัวอย่างของกล่องสมอก่อนเริ่มฝึก เมื่อเทียบกับเฉลย. กล่องสมอ ภายในกริดเซลล์เดียวกัน จะถูกกำหนดค่าเริ่มต้นให้มีขนาดหรือรูปทรงแตกต่างกัน. ขนาดหรือรูปทรงที่แตกต่างกัน ทำให้ค่าไอลอยูระหว่างกล่องสมอต่าง ๆ กับเฉลย ต่างกัน และสามารถใช้เป็นดัชนี เพื่อกำหนดความรับผิดชอบได้. ในภาพ ไอโอยู (ซึ่งคือ สัดส่วนทับซ้อน) ระหว่างกล่องสมอ **B2** กับเฉลย มีค่ามากกว่า ค่าของกล่องสมอ **B1** กับเฉลย. ดังนั้น ในกระบวนการฝึก โยโลจะกำหนดให้ กล่องสมอ **B2** รับผิดชอบเฉลย. ภาพบนขวา แสดงตัวอย่างเอาต์พุตหลังฝึกเสร็จ (กริดเซลล์ที่ดูแลเฉลยเปลี่ยนค่าเป็น **B1*** และ **B2***) โดยเมื่อการฝึกสมูญรุณ ค่าความมั่นใจของกล่องสมอที่ไม่ได้รับผิดชอบเฉลยได้ (**C1**) จะใกล้กับศูนย์ และค่าความมั่นใจของกล่องสมอที่ทำงาน (**C2**) จะใกล้กับหนึ่ง. ภาพล่าง แสดงผลลัพธ์ เมื่อนำไปวัดทับกับอินพุต. สังเกตว่า กล่องสมอที่รับผิดชอบเฉลย จะปรับขนาดและรูปทรงตามเฉลย. ตัวอย่างนี้ กริดเซลล์มีเพียงเฉลยเดียว จึงมีเพียงกล่องสมอเดียวที่ถูกใช้. หากกริดเซลล์รับผิดชอบวัตถุสองวัตถุทับซ้อนกัน กล่องสมอทั้งสองก็จะถูกใช้งาน และเลือกจับคู่โดยอาศัยค่าไอลอยเป็นดัชนี.

เท่ากัน). รูปที่ 7.4 แสดงการใช้งานเทคนิคกล่องสมอในโยโล.

สำหรับการตรวจจับวัตถุในภาพได้ครอบคลุม K ชนิดวัตถุ แต่ละกล่องสมอจะมี $5 + K$ ค่า สำหรับตำแหน่งและขนาดของกล่องขอบเขต (x, y, w, h), ค่าความมั่นใจว่าวัตถุอยู่ภายในกล่อง C , และค่าความน่าจะเป็นของวัตถุแต่ละชนิด $p(1), \dots, p(K)$. ดังนั้น สำหรับ M กริดเซลล์ และ B กล่องสมอต่อกริดเซลล์ แล้ว เอาต์พุตของโยโล $\mathbf{Y} \in \mathbb{R}^{M \cdot B \cdot (5+K)}$ 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 $\tilde{x}_{mb}, \tilde{y}_{mb}, \tilde{w}_{mb}, \tilde{h}_{mb}$ จะถูกคำนวณไปเป็น

ค่าพิกัดและขนาดของกล่องขอบเขต 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} เป็นพิกัดแนวโนนกับแนวตั้งของศูนย์กลางกล่องขอบเขตที่ทำนาย 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{w} มีค่าลบ จะทำให้ \hat{w} แคบกว่าค่าฐาน p_w . ถ้า \tilde{h} มีค่าศูนย์ จะทำให้ \hat{h} กว้างเท่ากับค่าฐาน p_h . ความสัมพันธ์ระหว่างค่า \tilde{h} กับ \hat{h} ก็เป็นในทำนองเดียวกัน.

ค่าฐานของแต่ละกล่องสมอ p_w และ p_h อาจเลือกกำหนดเองตามเห็นว่าเหมาะสม. คณของเรดมอน[161] ใช้การจัดกลุ่มข้อมูลด้วยวิธีเค-มีนล์ (*K-means*) สำรวจกล่องขอบเขตเฉลี่ยของข้อมูลฝึกหัด แล้วใช้ค่าเซนทรอยด์ ต่าง ๆ (*centroids*) ที่ได้มา เป็นค่าฐานของกล่องสมอต่าง ๆ.

โยโล่ นิยาม ค่าความมั่นใจ $\hat{C} = \text{Pr}(\text{Object}) \cdot \text{IOU}$ เมื่อ $\text{Pr}(\text{Object})$ แทนค่าความน่าจะเป็นที่กล่องขอบเขตจะมีวัตถุ และ IOU แทนค่าไอโอyu ระหว่างกล่องขอบเขตที่ทายกับกล่องขอบเขตเฉลี่ย. ดังนั้นค่าความมั่นใจเฉลี่ย $C = 0$ ถ้าไม่มีวัตถุอยู่ภายในกริดเซลล์ และ $C = \text{IOU}$ ถ้ามีวัตถุอยู่ภายในกริดเซลล์.

ในการฝึกการตรวจจับวัตถุในภาพ โยโล่กำหนดฟังก์ชันสูญเสียดังนี้

$$\begin{aligned} \text{loss} &= \lambda_{\text{coord}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \left((\hat{x}_{mb} - x_{mb})^2 + (\hat{y}_{mb} - y_{mb})^2 \right) \\ &\quad + \lambda_{\text{coord}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \left((\sqrt{\hat{w}_{mb}} - \sqrt{w_{mb}})^2 + (\sqrt{\hat{h}_{mb}} - \sqrt{h_{mb}})^2 \right) \\ &\quad + \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot (\hat{C}_{mb} - C_{mb})^2 + \lambda_{\text{noobj}} \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{noobj}} \cdot (\hat{C}_{mb} - C_{mb})^2 \\ &\quad + \sum_{m=1}^M \sum_{b=1}^B \mathbf{1}_{mb}^{\text{obj}} \cdot \sum_{k \in \text{classes}} \cdot (\hat{p}_{mb}(k) - p_{mb}(k))^2 \end{aligned} \quad (7.5)$$

เมื่อ $p_{mb}(k)$ คือเฉลยชนิดวัตถุที่กริดเซลล์ m กล่องสมอ b โดย $p_{mb}(k) = 1$ ถ้าที่กล่องสมอ มีภาพวัตถุนิด k และ $p_{mb}(k) = 0$ ถ้าที่กล่องสมอ มีภาพวัตถุนิดอื่น. สัญกรณ์ $\mathbf{1}_{mb}^{obj}$ ใช้ระบุว่ากล่องสมอ b ในกริดเซลล์ m รับผิดชอบการหาอย่างนี้คือ $\mathbf{1}_{mb}^{obj} = 1$ เมื่อ กล่องสมอ b ในกริดเซลล์ m มีเฉลยที่รับผิดชอบอยู่ ถ้าไม่อย่างนั้นให้ $\mathbf{1}_{mb}^{obj} = 0$. โดยโล่กำหนดการรับผิดชอบของกล่องสมอ โดยให้กล่องสมอที่มีค่าໄอโอยู่ร่วมกับกรอบตัวอย่างเฉลยมากที่สุด ทำหน้าที่รับผิดชอบการหาอย่างนั้น. สัญกรณ์ $\mathbf{1}_{mb}^{noobj}$ ใช้ระบุว่ากล่องสมอ b ในกริดเซลล์ m ไม่มีวัตถุอยู่ ซึ่ง $\mathbf{1}_{mb}^{noobj} = 1 - \mathbf{1}_{mb}^{obj}$.

เนื่องจาก คณะของเรดมอน[160] พบว่า ภาพต่าง ๆ ที่ใช้ฝึก ส่วนใหญ่มีวัตถุอยู่ไม่มาก. กล่องสมอส่วนใหญ่ไม่มีวัตถุ และสัดส่วนการไม่มีวัตถุต่อการมีวัตถุสูงมาก (ข้อมูลไม่สมดุล unbalanced data. แบบฝึกหัด 3.17). ดังนั้น คณะของเรดมอน เลือกใช้แนวทางหนึ่งที่นิยมใช้บรรเทาปัญหาเช่นนี้ คือใช้ค่าน้ำหนักที่ต่างกันเพื่อชดเชย. ค่า λ_{coord} และ λ_{noobj} เป็นเพียงเทคนิคเชิงเลข เพื่อชดเชยความไม่สมดุลของข้อมูล (ซึ่งคณะของเรดมอน เลือกใช้ $\lambda_{coord} = 5$ และ $\lambda_{noobj} = 0.5$).

สังเกตว่า การคำนวณค่าผิดพลาดของความกว้างและความสูง ทำผ่านค่ารากที่สอง. เนื่องจาก ค่าผิดพลาดสัมบูรณ์ ของการทำนายขนาดสำหรับกล่องของเขตขนาดเล็ก แม้ตัวเลขจะเท่ากับค่าผิดพลาดสัมบูรณ์ ของการทำนายขนาดสำหรับกล่องของเขตขนาดใหญ่ แต่ถือเป็นความผิดพลาดที่รุนแรงกว่า. ตัวอย่างเช่น การหาความกว้างผิดไป 10 สำหรับความกว้าง 500 นั้นถือว่าเล็กน้อยมาก จนผู้ใช้อาจไม่ได้เห็นความแตกต่าง แต่ การหาความกว้างผิดไป 10 สำหรับความกว้าง 5 นั้นถือว่าผิดพลาดรุนแรงมาก และผลลัพธ์ที่เห็นได้อย่างชัดเจน. คณะของเรดมอน[160] ใช้เทคนิคเชิงเลข โดยคำนวณความแตกต่างของค่ารากที่สองแทน เพื่อช่วยบรรเทาปัญหานี้.

ในกระบวนการฝึก คณะของเรดมอน[160] ใช้การฝึกก่อน โดยฝึกแบบจำลองกับงานจำแนกชนิดวัตถุหลักในภาพก่อนจนแบบจำลองทำงานได้ดีแล้ว. จากนั้นจึงเพิ่มขั้นคำนวณท้าย ๆ (ด้านเอาร์พุต) เข้าไปแล้วจึงฝึกแบบจำลองสำหรับภารกิจการตรวจจับวัตถุในภาพ.

หลังจากฝึกเสร็จ ในการงานอนุมาน ค่าเอาร์พุตจะถูกนำมาประมวลผล โดยค่าของกล่องสมอที่ $\hat{C} > \tau$ จะถูกนำมาคำนวณตำแหน่งและขนาดของกล่องของเขต (สมการ 7.1, 7.2, 7.3, 7.4) และวัตถุจะถูกอนุมานเป็นชนิด $k^* = \arg \max_k \hat{p}(k)$ เมื่อ τ เป็นระดับค่าขีดแบ่งที่กำหนด.

7.2 การซ่อม เสริม และก่อกำเนิดภาพ

การซ่อมภาพ คือการเติมส่วนของภาพที่ต้องการ (ส่วนของภาพที่เสียหาย) โดยคำนึงถึงบริเวณรอบข้างและลักษณะของภาพโดยรวม. การเสริมภาพ มีความหมายครอบคลุมการเพิ่มความละเอียดให้กับภาพ (หัวข้อ 7.2). การก่อกำเนิดภาพ คือการสร้างภาพขึ้นมาใหม่ทั้งภาพ โดยภาพที่สร้างขึ้นเป็นภาพในลักษณะที่ต้องการ เช่น ดูคล้ายภาพจริง (หัวข้อ 7.2).

การซ่อม เสริม และก่อกำเนิดภาพ เป็นศาสตร์หนึ่งที่ได้รับความสนใจอย่างมาก และมีการพัฒนาอย่างรวดเร็ว มีหลายแนวทาง อาทิ พิกเซลอาร์เอนเน็น (PixelRNN[201]), ตัวเข้ารหัสอัตโนมัติแบบเปลี่ยน-

แปลง (Variational Autoencoder[213]), หรือโครงข่ายปรับเปลี่ยนสร้าง (Generative Adversarial Network[77, 145] หัวข้อ 7.2).

ความท้าทายที่สำคัญสำหรับภาระกิจเช่นนี้ โดยเฉพาะภารกิจกำเนิดภาพ คือ ตัวภาระกิจเป็นเสมือนการเรียนรู้ความน่าจะเป็นของภาพ (ไม่ว่าจะเรียนรู้ชัดแจ้งโดยตรง ซึ่งได้ค่าฟังก์ชันความหนาแน่นความน่าจะเป็นอุปกรณ์ หรือโดยนัย ซึ่งคือทำการภารกิจได้ แต่ไม่ได้ค่าฟังก์ชันความหนาแน่นความน่าจะเป็น). จากมุมของปริภูมิมิติ ภาพเป็นจุดข้อมูลที่อยู่ในปริภูมิหลายมิติ ที่มีจำนวนมิติมหาศาล¹. 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.

โครงข่ายปรับเปลี่ยนสร้าง จัดเป็นศาสตร์และศิลป์ที่สำคัญของการเรียนรู้ของเครื่อง ดังอภิราย เกริ่นในบทที่ 5 และ ได้แสดงให้เห็นว่า โครงข่ายปรับเปลี่ยนสร้างเป็นแนวทางที่ช่วยแก้ปัญหาของภารกิจการก่อกำเนิดภาพได้. หัวข้อ 7.2 อภิรายโครงข่ายปรับเปลี่ยนสร้าง รวมถึงอุปสรรคความท้าทายในการประยุกต์ใช้โครงข่ายปรับเปลี่ยนสร้าง และแนวทางในการบรรเทาอุปสรรค.

การเพิ่มความละเอียดให้กับภาพ

การเพิ่มความละเอียดให้กับภาพ คือ กระบวนการที่รับภาพความละเอียดต่ำ (low-resolution image) และประมาณภาพความละเอียดสูง (high-resolution) อุปกรณ์. การเพิ่มความละเอียดให้กับภาพ อาจทำได้หลายวิธี. ตัวอย่าง เช่น คณะของตง[57] ทำอัพแซมป์ลิ้ง (upsampling) ซึ่งคือการเพิ่มพิกเซลเข้าไปในภาพ โดยค่าพิกเซลที่เพิ่มขึ้นจะได้จากการประมาณค่าในช่วงแบบไบคิวบิก (bicubic interpolation). จากนั้นใช้โครงข่ายคอนโวลูชันในการประมาณภาพความละเอียดสูงอุปกรณ์ (เพื่อปรับปรุงคุณภาพจากการประมาณค่าในช่วงแบบไบคิวบิก).

กล่าวคือ จากภาพความละเอียดต่ำ $\tilde{\mathbf{X}}$ คณะของตงสร้างภาพความละเอียดสูง \mathbf{X}' ขึ้นด้วยวิธีการประมาณค่าในช่วงแบบไบคิวบิก และใช้ \mathbf{X}' เป็นอินพุตของโครงข่ายคอนโวลูชัน เพื่อทำนาย $\hat{\mathbf{X}}$ (ซึ่งแม้จะความละเอียดเท่ากัน แต่ $\hat{\mathbf{X}}$ มีคุณภาพดีกว่า \mathbf{X}'). หาก f คือฟังก์ชันที่แทนการคำนวณของโครงข่าย และ Θ เป็นค่าพารามิเตอร์ต่าง ๆ ของโครงข่าย โครงข่ายคอนโวลูชันถูกฝึกให้ทำนาย $\hat{\mathbf{X}} = f(\mathbf{X}'; \Theta)$ ให้ใกล้เคียงกับเฉลย (ที่เป็นภาพความละเอียดสูง) \mathbf{X} ให้มากที่สุด. นั่นคือ ฟังก์ชันสูญเสีย $\text{loss}(\Theta) = \frac{1}{N} \sum_{n=1}^N \|f(\mathbf{X}'_n; \Theta) - \mathbf{X}_n\|^2$ เมื่อ N คือจำนวนข้อมูลฝึกหัด.

คุณภาพของการเพิ่มความละเอียดให้กับภาพ อาจประเมินจาก ค่าผิดพลาดกำลังสองเฉลี่ย เช่น เดียวกับภารกิจการหาค่าผลโดยทั่วไป เช่น $\text{mse} = \frac{1}{W \cdot H} \sum_i \sum_j (\hat{x}_{i,j} - x_{i,j})^2$ เมื่อ W กับ H เป็นความกว้างกับสูงของภาพ และ $\hat{x}_{i,j}$ เป็นค่าพิกเซลของภาพที่เพิ่มความละเอียดขึ้นจากภาพความละเอียดต่ำ $\tilde{\mathbf{X}}$. โดย ภาพความละเอียดต่ำ $\tilde{\mathbf{X}}$ เป็นภาพที่ถูกลดความละเอียดลงจากภาพความละเอียด

¹ลักษณะภารกิจที่การดำเนินการทำได้ไม่ยาก เมื่อมิติของปริภูมิมีจำนวนน้อย แต่ทำได้ยากมาก หรือไม่อาจทำได้เลยในทางปฏิบัติ หากมิติของปริภูมิมีจำนวนมาก มักถูกอ้างอิงถึงว่าเสมอเป็น คำสาปของมิติ (curses of dimensionality).

สูง \mathbf{X} . ภาพความละเอียดสูง \mathbf{X} มีค่าพิกเซลต่าง ๆ เป็น $x_{i,j}$ และ i กับ j คือ ดัชนีแนวนอนกับแนวตั้งของภาพ. อย่างไรก็ได้ เมื่อค่าพิกเซลกำลังสองเฉลี่ยพอใช้งานได้ แต่นักวิจัยต่างพบว่า ค่าพิกเซลกำลังสองเฉลี่ยไม่สัมพันธ์กับคุณภาพของภาพที่คนรับรู้^[211]. คุณภาพของภาพจึงมักวัดด้วยดัชนีอื่น 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] คณะของชื่อ^[183, 184] ใช้โครงข่ายคอนโวลูชัน เพื่อเพิ่มความละเอียดให้กับภาพ โดยรับอินพุตเป็นภาพความละเอียดต่ำโดยตรง และให้อาต์พุตสุดท้ายออกมาเป็นภาพความละเอียดสูงได้เลย. การใช้โครงข่ายคอนโวลูชันกับภาพความละเอียดต่ำโดยตรง ช่วยลดภาระการคำนวณลงไปได้มาก และคณะของชื่อ ยังแสดงให้เห็นคุณภาพของผลลัพธ์ที่ดีขึ้นด้วย.

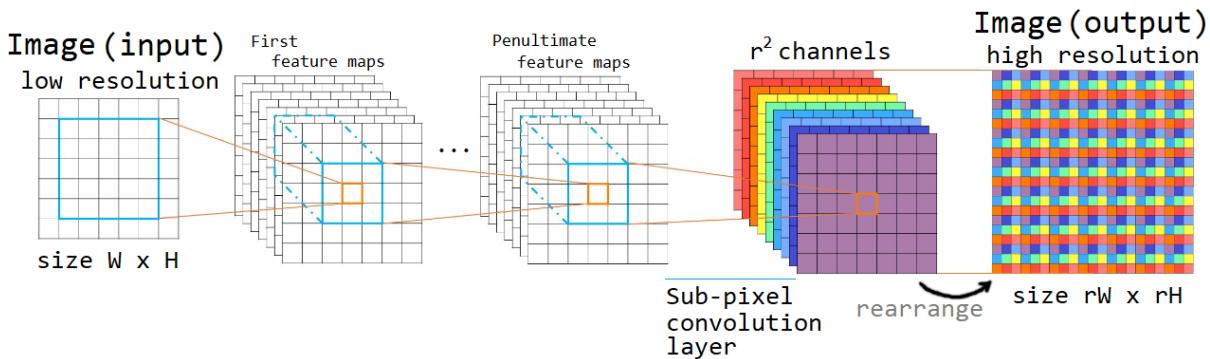
กลไกสำคัญที่ชื่อและคณะใช้ อยู่ที่ชั้นคำนวณท้ายสุด. สำหรับภาพขนาด $W \times H$ (ช่องสีเดียว²) และต้องการขยายความละเอียดขึ้น r เท่า (นั่นคือ ภาพจะถูกขยายเป็น ภาพผลลัพธ์ขนาด $rW \times rH$) คณะของชื่อ ออกแบบโครงข่ายคอนโวลูชันที่ให้อาต์พุตออกมา เป็นแผนที่ลักษณะลำดับข้าดาเท่ากับขนาดภาพอินพุต แต่มีจำนวนแผนที่เท่ากับ r^2 . แล้วภาพผลลัพธ์ จะสร้างขึ้นจากการจัดเรียงแผนที่ลักษณะลำดับข้าดา $W \times H$ จำนวน r^2 แผ่นที่ ให้เป็นแผนที่เดียว ขนาด $rW \times rH$ ซึ่งคือภาพความละเอียดสูงที่ต้องการ. กลไกของการจัดเรียงผลลัพธ์แผนที่สำคัญเช่นนี้ เรียกว่า กลไกพิกเซลย้อยซึ่งเป็นรูปหนึ่งของชั้นดีคอนโวลูชัน (deconvolution layer^[224]). ชั้นดีคอนโวลูชัน อาจทำได้หลายรูปแบบ ชื่อและคณะ^[184] แจกแจงและอภิปรายข้อแตกต่างของชั้นดีคอนโวลูชันแบบต่าง ๆ. รูปที่ 7.5 แสดงจุดเด่น (รับอินพุตเป็นภาพความละเอียดต่ำ และให้อาต์พุตเป็นภาพความละเอียดสูงได้โดยตรง) และกลไกสำคัญ (ชั้นดีคอนโวลูชัน ที่จัดเรียงแผนที่ลักษณะลำดับ r^2 แผ่นที่ เป็นผลลัพธ์ ภาพเดียวที่เป็นมีความละเอียดเพิ่มเป็น r เท่า).

โครงข่ายปรับเปลี่ยนสร้าง

โครงข่ายปรับเปลี่ยนสร้าง³ (Generative Adversarial Networks คำย่อ GANs) หมายถึง โครงข่ายประสาทเทียมที่สามารถเรียนรู้ความน่าจะเป็นของข้อมูลที่มีจำนวนมิติสูง ๆ และมีหลาย ๆ โหมดได้ (high-dimensional and multi-modal distribution) โดยโครงข่ายถูกเตรียมด้วยวิธีการฝึกแบบปรับเปลี่ยน. ด้วยกลไกวิธีการฝึกแบบปรับเปลี่ยน ที่ใช้การเรียนรู้แบบกึ่งมัลติชัลล่อนและการเรียนรู้แบบไม่มี

²ที่นี่ ยกตัวอย่างภาพช่องสีเดียว เพื่อความกระชับของเนื้อหา. เทคนิคที่อธิบายนี้ สามารถประยุกต์ใช้กับภาพหลายช่องสี ได้อย่างต่อไปต่อมา.

³เนื้อหาในหัวข้อนี้ ได้รับอิทธิพลหลัก ๆ จากเครสวอลและคณะ^[45].



รูปที่ 7.5 การขยายความละเอียดภาพด้วยชั้นดีคอนโวโลชัน (ดัดแปลงจากซีอและคณะ[183])

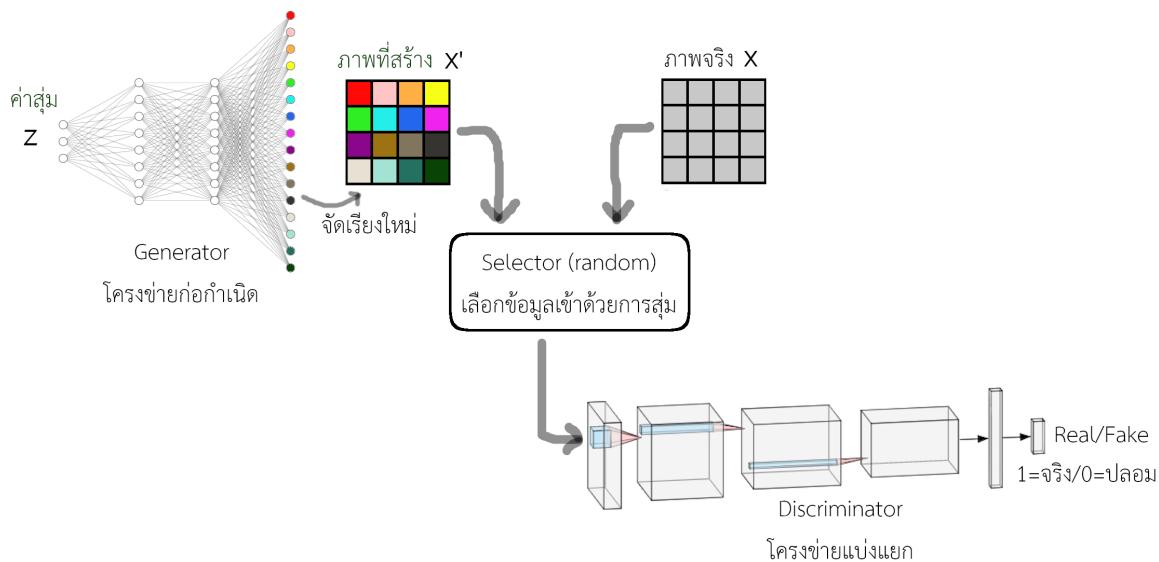
ผู้สอน โครงข่ายปรับก์เชิงสร้างสามารถใช้ประโยชน์จากข้อมูลจำนวนมากที่ไม่มีฉลากเฉลยได้ ซึ่งข้อมูลจำนวนมากนั้น จำเป็นต่อการเรียนรู้ความน่าจะเป็นของข้อมูลที่มีจำนวนมิติสูง ๆ (เช่น ข้อมูลรูปภาพ).

อย่างไรก็ตาม การเรียนรู้ความน่าจะเป็นของข้อมูลด้วยโครงข่ายปรับก์เชิงสร้าง อาจเป็นเพียงการเรียนรู้เชิงนัย นั่นคือ อาจไม่ได้ค่าความน่าจะเป็นหรือไม่ได้ฟังก์ชันความหนาแน่นความน่าจะเป็น. นั่นคือ โครงข่ายอาจสามารถสังเคราะห์ หรือสร้างตัวอย่างข้อมูลขึ้นมาใหม่ได้ โดยตัวอย่างข้อมูลที่สร้างขึ้นมาใหม่ มีลักษณะในแบบเดียวกับตัวอย่างข้อมูลจริง (กล่าวคือ มีความเป็นไปได้สูงว่ามาจากการแจกแจงเดียวกัน). หากข้อมูลที่กล่าวถึง คือภาพ โครงข่ายปรับก์เชิงสร้าง อาจสามารถสร้างตัวอย่างภาพขึ้นมาใหม่ ซึ่งภาพที่สร้างขึ้นนี้อาจดูเหมือนภาพถ่ายจริง ซึ่งเบื้องหลังหมายถึงว่า โครงข่ายได้เรียนรู้การแจกแจงข้อมูลของภาพถ่ายจริง และสามารถสังเคราะห์ตัวอย่างข้อมูลจากการแจกแจงนั้นได้ แต่ฟังก์ชันการแจกแจงนั้น อาจไม่สามารถเข้าถึงได้โดยตรง.

วิธีการฝึกแบบปรับก์ (adversarial training) มีลักษณะเด่น คือ การใช้โครงข่ายสองโครงข่ายในการฝึก และโครงข่ายทั้งสองถูกฝึกโดยมีเป้าหมายที่ขัดแย้งกัน. คณะของกูดเฟโล[77] เสนอโครงข่ายปรับก์เชิงสร้าง เพื่อสร้างตัวอย่างภาพต่าง ๆ ที่เหมือนภาพถ่ายจริงอ่อนโยน. วิธีการฝึกแบบปรับก์ ใช้โครงข่ายสองโครงข่าย หนึ่งเรียกว่า โครงข่ายแบ่งแยก (discriminator ใช้สัญกรณ์ D) และอีกหนึ่งเรียกว่า โครงข่ายก่อกำเนิด (generator ใช้สัญกรณ์ G). โครงข่ายแบ่งแยก D รับอินพุตเป็นภาพ และทำหน้าที่判断ว่า ภาพที่รับเข้ามาเป็นภาพถ่ายจริง หรือว่าเป็นภาพปลอม (ภาพที่สร้างขึ้น). โครงข่ายก่อกำเนิด G รับอินพุตเป็นค่าสุ่ม⁴ และทำหน้าที่สร้างภาพขึ้นมา. ในกระบวนการฝึก โครงข่ายแบ่งแยก D ถูกฝึก โดยมีเป้าหมายคือ การแยกแยะให้ถูกต้องมากที่สุด ในขณะที่โครงข่ายก่อกำเนิด G ถูกฝึก โดยมีเป้าหมายคือ การสร้างภาพปลอมให้เหมือน จนโครงข่ายแบ่งแยก D ทายถูกน้อยที่สุด.

รูปที่ 7.6 แสดงแนวคิดของวิธีการฝึกแบบปรับก์ ซึ่งเป็นกลไกหลักของโครงสร้างปรับก์เชิงสร้าง. 在การฝึก โครงข่ายแบ่งแยก D จะรับอินพุต เป็นภาพ ที่ถูกสุ่มขึ้นมา โดยภาพที่ได้อาจสุ่มจากภาพจริง

⁴ที่ต้องรับอินพุตเป็นค่าสุ่ม เพื่อให้โครงข่ายก่อกำเนิด G เรียนรู้ที่จะสร้างเอกสารพุตที่หลากหลาย. นั่นคือ อินพุตเป็นค่าหนึ่ง โครงข่ายก่อกำเนิด G สร้างภาพหนึ่ง. อินพุตเป็นอีกค่าหนึ่ง โครงข่ายก่อกำเนิด G สร้างภาพอีกภาพหนึ่ง.



รูปที่ 7.6 วิธีการฝึกแบบปรปักษ์. โครงข่ายก่อกำเนิด \mathcal{G} (แสดงด้วยโครงข่ายเชื่อมต่อเต็มที่ มุมบนซ้าย) รับอินพุตเป็นเวกเตอร์ค่าสุ่ม z และให้อาต์พุต \mathbf{X}' ออกมาน (ในภาพ แสดงเอาต์พุตถูกจัดเรียงใหม่เพื่อให้มีโครงสร้างเหมือนภาพจริง \mathbf{X}). ภาพจริง \mathbf{X} ถูกสุ่มอกมาจากชุดข้อมูล. โครงข่ายแบ่งแยก \mathcal{D} (แสดงด้วยโครงข่ายคอนโวลูชัน มุมล่างขวา) รับอินพุตที่เป็นภาพ โดยโครงข่ายแบ่งแยกไม่รู้ว่าภาพอินพุตที่ได้ ถูกเลือกมาจากภาพจริง หรือภาพที่สร้างขึ้น (การเลือกทำด้วยการสุ่ม) และโครงข่ายแบ่งแยกจะต้องทายว่าอินพุตที่เห็น เป็นภาพจริง หรือเป็นภาพที่สร้างขึ้น.

หรืออาจสุ่มมาจากภาพปลอมที่สร้างโดย โครงข่ายก่อกำเนิด \mathcal{G} แล้วให้ โครงข่ายแบ่งแยก \mathcal{D} ทำงาน. ผลของการทำงานผิดหรือถูก จะถูกนำไปปรับค่าน้ำหนักเพื่อให้ โครงข่ายแบ่งแยก \mathcal{D} ทำงานได้ดีขึ้น แบ่งแยกได้ดีขึ้น (ทายถูกมากขึ้น) และก็จะถูกนำไปปรับค่าน้ำหนัก โครงข่ายก่อกำเนิด \mathcal{G} เพื่อให้ \mathcal{G} สร้างภาพได้ดีขึ้น (หลอก \mathcal{D} ได้ดีขึ้น ทำให้ \mathcal{D} ทายถูกน้อยลง). หากโครงข่ายก่อกำเนิด \mathcal{G} สามารถหลอกโครงข่ายแบ่งแยก \mathcal{D} ได้โดยสมบูรณ์แล้ว โครงข่ายแบ่งแยก \mathcal{D} จะทายถูกได้ประมาณครึ่ง ๆ. นั่นคือ ถ้าภาพที่สร้างขึ้นเหมือนภาพจริง โอกาสคือเท่ากับเดาสุ่ม ซึ่งถ้าเดาดี ๆ โอกาสถูกคือแค่ประมาณ 0.5 (หรือ 50%). หมายเหตุ รูปที่ 7.6 อาจแสดงโครงข่ายก่อกำเนิดด้วยโครงข่ายเชื่อมต่อเต็มที่ แต่การเปลี่ยนโครงสร้างไปเป็นโครงข่ายคอนโวลูชันก็สามารถทำได้ (ดู [156, 53] เพิ่มเติม).

การฝึกโครงข่ายปรปักษ์เชิงสร้าง กล่าวโดยเจาะจงแล้วก็คือการแก้ปัญหาค่าตี่ที่สุด ในนิพจน์ 7.6 ได้แก่

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) \quad (7.6)$$

เมื่อ

$$V(\mathcal{G}, \mathcal{D}) = E_{\mathbf{X} \sim p_{data}}[\log \mathcal{D}(\mathbf{X})] + E_{\mathbf{X} \sim p_{\mathcal{G}}}[\log(1 - \mathcal{D}(\mathbf{X}))] \quad (7.7)$$

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 ค่าคาดหมายของผลลัพธ์จากโครงข่ายแบ่งแยก เมื่ออินพุตของโครงข่ายแบ่งแยกมีการแจกแจงตามข้อมูลจริง (ดังระบุด้วยสัญกรณ์ $\mathbf{X} \sim p_{data}$) หรือกล่าวง่าย ๆ คือ เมื่ออินพุตเป็นภาพจริง. หากโครงข่ายแบ่งแยกทำงานถูกต้องโดยสมบูรณ์ แล้ว $\mathcal{D}(\mathbf{X}) \approx 1$ สำหรับทุก ๆ ภาพจริง และ พจน์ $E_{\mathbf{X} \sim p_{data}}[\log \mathcal{D}(\mathbf{X})] \approx 0$.

ส่วนพจน์⁵ $E_{\mathbf{X} \sim p_G}[\log(1 - \mathcal{D}(\mathbf{X}))]$ แสดงค่าคาดหมายของผลการที่มีของ $1 - \mathcal{D}(\mathbf{X})$ เมื่อ อินพุตของมีการแจกแจงตามการแจกแจงจากโครงข่ายก่อกำเนิด (ดังระบุด้วยสัญกรณ์ $\mathbf{X} \sim p_G$) หรือ กล่าวง่าย ๆ คือ เมื่ออินพุตถูกสร้างขึ้นจากโครงข่ายก่อกำเนิด. หากโครงข่ายแบ่งแยกทำงานถูกต้องโดย สมบูรณ์ แล้ว $\mathcal{D}(\mathbf{X}) \approx 0$ สำหรับทุก ๆ ภาพที่สร้างขึ้น และ พจน์ $E_{\mathbf{X} \sim p_G}[\log(1 - \mathcal{D}(\mathbf{X}))] \approx 0$. แต่หากโครงข่ายแบ่งแยก ทำให้ได้ $\log(0) \rightarrow -\infty$ หรือทำให้ได้ค่าที่ต่ำมาก ๆ.

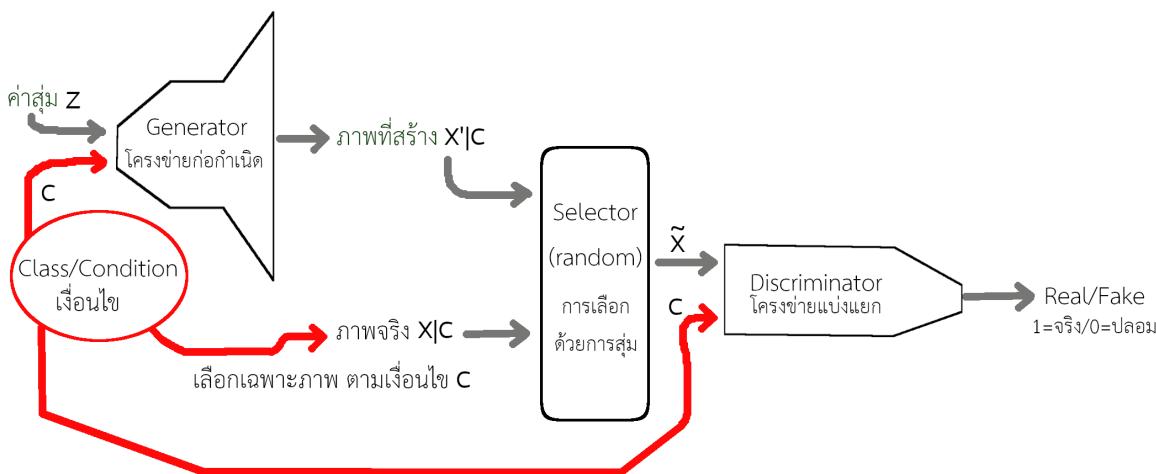
หมายเหตุ สัญกรณ์ที่ในนิพจน์ 7.6 ใช้เพื่อความกระชับ. เช่นเดียวกับการฝึกโครงข่ายประสาท เทียมอื่น ๆ การฝึกโครงข่ายก่อกำเนิดและโครงข่ายแบ่งแยก ก็ดำเนินการผ่านการปรับค่าพารามิเตอร์ ต่าง ๆ ของโครงข่าย. นั่นคือ หากจะเขียนนิพจน์ 7.6 ให้ละเอียดถูกต้องมากยิ่งขึ้น อาจเขียนเป็น $\min_{\boldsymbol{\theta}} \max_{\mathbf{w}} V(\mathcal{G}_{\boldsymbol{\theta}}, \mathcal{D}_{\mathbf{w}})$ เมื่อโครงข่ายก่อกำเนิด $\mathcal{G}_{\boldsymbol{\theta}}$ และโครงข่ายแบ่งแยก $\mathcal{D}_{\mathbf{w}}$ ถูกควบคุมพุทธิ- กรรมด้วยพารามิเตอร์ $\boldsymbol{\theta}$ และ \mathbf{w} ตามลำดับ.

โครงข่ายแบ่งแยก \mathcal{D} จะถูกฝึกเพื่อให้ค่าฟังก์ชันจุดประสงค์นี้สูงที่สุด ผ่านกลไกการทำนาย $\mathcal{D}(\mathbf{X})$. ในขณะที่ โครงข่ายก่อกำเนิด จะพยายามทำให้ค่าฟังก์ชันจุดประสงค์นี้ต่ำที่สุด โดยผ่านกลไก $\mathbf{X} \sim p_G$ ซึ่งคือ การสร้างภาพให้เหมือนภาพจริงที่สุด หรือพยายามเรียนรู้ให้การแจกแจง p_G ใกล้เคียงกับ p_{data} มากที่สุด. โครงข่ายก่อกำเนิดในอุดมคติ จะมี $p_G \approx p_{data}$.

ปัจจัยสำคัญประการหนึ่งคือ โครงข่ายก่อกำเนิด \mathcal{G} ไม่ได้รับข้อมูลเกี่ยวกับภาพจริงโดยตรงเลย โครงข่ายก่อกำเนิด \mathcal{G} ถูกบังคับให้เรียนรู้การแจกแจงของภาพจริงผ่านปฏิสัมพันธ์กับโครงข่ายแบ่งแยก \mathcal{D} . โครงข่ายแบ่งแยก \mathcal{D} เห็นทั้งภาพจริง และภาพที่สร้างขึ้น และได้รับเฉลยผ่านเกรเดียนต์หลังจาก ทายไป. อีกทอดหนึ่ง โครงข่ายก่อกำเนิด \mathcal{G} ก็ได้รับเกรเดียนต์ของมันผ่านเกรเดียนต์ของโครงข่ายแบ่งแยก \mathcal{D} อีกด้วย.

โครงข่ายก่อกำเนิด อาจถูกมองว่าเป็นการเรียนรู้ เพื่อที่จะแปลงข้อมูลจากปริภูมิของตัวแทนสู่ ที่อาจถูกเรียกว่า ปริภูมิตัวแทน (*representation space*) หรือปริภูมิช่องเร้น (*latent space*) ไปสู่ ปริภูมิของข้อมูล. นั่นคือ $\mathcal{G} : \mathbf{z} \rightarrow \mathbf{X}$ เมื่อ \mathbf{z} คือตัวแปรในปริภูมิช่องเร้น และ \mathbf{X} คือตัวแปรใน ปริภูมิข้อมูล. ส่วนโครงข่ายแบ่งแยกก็เป็นเหมือนการแปลงจากข้อมูลไปสู่ค่าระหว่างศูนย์กับหนึ่ง. นั่น คือ $\mathcal{D} : \mathbf{X} \rightarrow (0, 1)$. ภายหลังการฝึกเสร็จสิ้น โครงข่ายก่อกำเนิด \mathcal{G} สามารถนำไปใช้สร้างตัวอย่าง ข้อมูลได้ตามต้องการ.

⁵ คำแนะนำของกูดเพลโล[77] ใช้พจน์ $E_{\mathbf{z} \sim p_z}[\log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z})))]$ เมื่อ $\mathcal{G}(\mathbf{z})$ แทนภาพที่สร้างจากโครงข่ายก่อกำเนิดตามค่าสุ่ม \mathbf{z} และ $E_{\mathbf{z} \sim p_z}$ หมายถึงค่าคาดหมายคำนวนตามการแจกแจงของตัวแปรสุ่ม. แต่ ณ ที่นี่เขียนพจน์นี้ ตามเครสเวลและคณะ[45] เพื่อความ กระชับในการอธิบาย.



รูปที่ 7.7 การฝึกโครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข. คล้ายการฝึกโครงสร้างปรับปั้นเชิงสร้างแบบดั้งเดิม (ไม่มีเงื่อนไข) การฝึกโครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข เพิ่มข้อมูลของเงื่อนไข ให้ข้อมูลเงื่อนไขนี้กับโครงสร้างก่อกำเนิด และโครงสร้างแบ่งแยก รวมถึงข้อมูลที่จริงที่จะเลือกมา ก็ต้องถูกควบคุมให้เป็นข้อมูลที่ตรงกับเงื่อนไขด้วย. การกำหนดเงื่อนไข เช่นนี้ ช่วยให้เราสามารถควบคุมเอาร์พุตของโครงสร้างก่อกำเนิด เพื่อให้สร้างเอาต์พุตตามเงื่อนไขที่เราต้องการได้.

โครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข. เมียร์ชชและคณะ[129] ขยายความสามารถของโครงสร้างปรับปั้นเชิงสร้าง โดยใช้ความน่าจะเป็นแบบมีเงื่อนไข. โครงสร้างปรับปั้นเชิงสร้างที่มีความสามารถที่เพิ่มขึ้นมาเช่นนี้ ถูกเรียกว่า โครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข (Conditional Generative Adversarial Networks). กระบวนการฝึกของโครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข 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 $\mathcal{D}(\mathbf{X}|\mathbf{C})$ แทนผลการทำนายจากโครงสร้างแบ่งแยก ที่รับอินพุตหลักเป็น \mathbf{X} และรับอินพุตรอง เป็น \mathbf{C} ซึ่งใช้ระบุเงื่อนไข. สัญกรณ์ $\mathbf{X} \sim p_{data|\mathbf{C}}$ หมายถึง ตัวแปร \mathbf{X} มีการแจกแจงตามข้อมูลจริง ที่เป็นไปตามเงื่อนไข \mathbf{C} . สัญกรณ์ $\mathbf{X} \sim p_{\mathcal{G}|\mathbf{C}}$ หมายถึง ตัวแปร \mathbf{X} มีการแจกแจงตามการแจกแจง จากโครงสร้างก่อกำเนิด ที่เงื่อนไข \mathbf{C} . รูปที่ 7.7 แสดงกลไกเพิ่มเติม เพื่อเพิ่มคุณสมบัติการใช้เงื่อนไข ให้กับโครงสร้างปรับปั้นเชิงสร้าง. โครงสร้างและรายละเอียดในการทำโครงสร้างปรับปั้นเชิงสร้างแบบมีเงื่อนไข อาจแตกต่างไปได้ เช่น อินโฟแกน (InfoGAN[36]).

การประยุกต์ใช้โครงสร้างปรับปั้นเชิงสร้าง. โครงสร้างปรับปั้นเชิงสร้าง เป็นพัฒนาการที่สำคัญสำหรับ การเรียนรู้เชิงลึก และได้ทำให้เกิดการประยุกต์ใช้อย่างกว้างขวาง ขยายเข้าไปแม้แต่ในวงการศิลปะ. การศึกษาวิจัยและขอบเขตการใช้งานของโครงสร้างปรับปั้นเชิงสร้าง เป็นไปอย่างรวดเร็วและต่อเนื่อง จนโครงสร้างปรับปั้นเชิงสร้างเป็นเสมือนศาสตร์ย่อย ๆ ในตัวเอง. ตัวอย่างการประยุกต์ใช้ทั่ว ๆ ไปส่วน

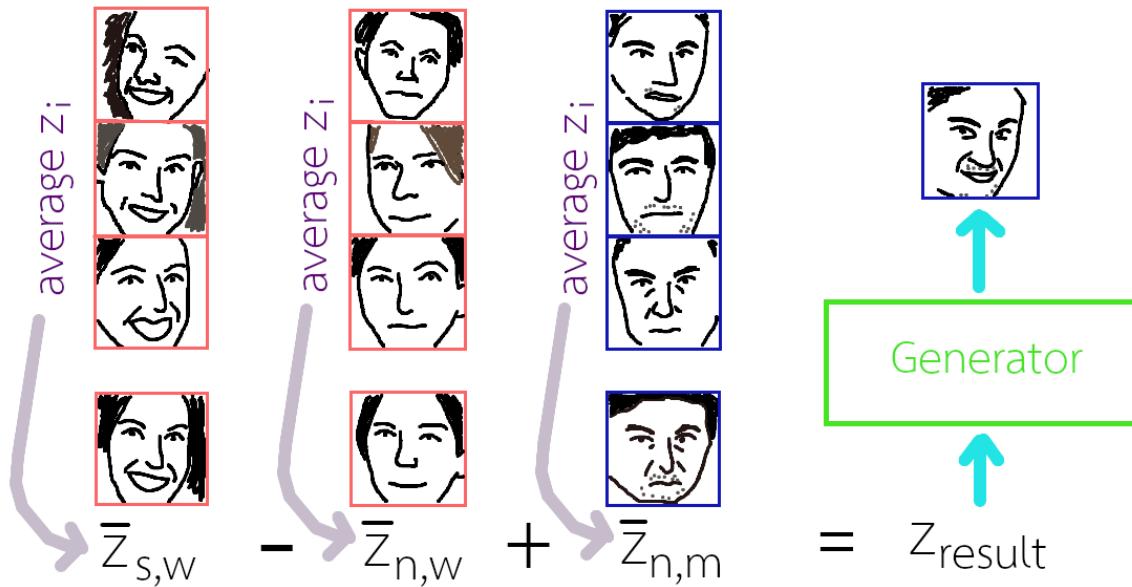
หนึ่งของโครงข่ายปรัปักษ์เชิงสร้าง ได้แก่ การจำแนกกลุ่ม (เช่น การนำโครงข่ายแบ่งแยกไปใช้), การสกัดลักษณะสำคัญ (ซึ่งอาจจะได้จากทั้งค่าເວົ້າພຸດໜັ້ນຂອງພາຍໃນโครงสร้างของโครงข่ายแบ่งแยก หรืออาจจะได้จากการทำพີ່ຄນິຕເວກເຕອຣ໌ ທີ່ຈະອົບຍາເພີ່ມເຕີມຕ່ອງໄປ), การສັງເຄຣະຫົ່ວໝູລ (ซຶ່ງຄື່ອງ การสร้างຂໍ້ມູນ ໂດຍໃຫ້ໂຄຮງຂ່າຍກ່ອກມີເນີດ ແລະນີ້ຄື່ອງ ຈຸດປະສົງຄໍຫລັກຂອງໂຄຮງຂ່າຍปรັບກົງທີ່), การແປລັງຮູບໜັ້ນໄປສູ່ອົກຮູບໜັ້ນ, การເພີ່ມຄວາມລະເອີຍໃຫ້ກັບພາພ ເປັນຕົ້ນ.

ຄະນະຂອງເຮັດ[163] ໃຫ້ໂຄຮງຂ່າຍปรັບກົງທີ່ໃຫ້ໃນການສ້າງພາພຂຶ້ນມາຕາມຄຳບຽຍ. ໂຄຮງຂ່າຍປະກົງທີ່ສ້າງຮູບໜັ້ນຈາກສ່ວນພາພເລີກ ຖ້າ ທີ່ແຕ່ລະສ່ວນພາພສ້າງຂຶ້ນມາຕາມທຳແໜ່ງທີ່ກຳນົດ ແລະຕາມລັກຊະນະພື້ນຜົວທີ່ບຽຍ. ນອກຈາກນີ້ ມີການໃຫ້ໂຄຮງຂ່າຍປະກົງທີ່ສ້າງໄປໃຫ້ໃນກະບວນແກ້ໄຂແລະຕົກແຕ່ງຮູບ[23, 225]. ການແປລັງຮູບໜັ້ນໄປສູ່ອົກຮູບໜັ້ນ[94] ກີ່ມີການປະຢຸກໃຫ້ທີ່ຫລາກຫລາຍ ເຊັ່ນ ການແປລັງກະບວນແບບສິລປະ (artistic style transfer[117]), ການແປລັງກະບວນແບບ (style transfer[101]), ການສ້າງພາພເໝືອນຈິງຕາມສ່ວນພາພ ທີ່ສິລປິນສາມາດວາດພາກຄ່າວ່າແລ້ວໃຫ້ໂຄຮງຂ່າຍປະກົງທີ່ສ້າງໜີ່ຢ່າງເຕີມຮາຍລະເອີຍດ (semantic image synthesis[148, 147]), ການສ້າງພາພລ້ວມເລີຍນຸ່ມຄວາມ (automatic caricature generation[185]), ການເພີ່ມອາຍຸໃຫ້ໜ້າ (age progression[51]).

ພີ່ຄນິຕເວກເຕອຣ໌ (vector arithmetic) ຄື່ອງ ການນຳເວກເຕອຣ໌ຄ່າສຸ່ມ z ທີ່ເປັນອິນພຸດຂອງໂຄຮງຂ່າຍກ່ອກມີເນີດ ສໍາຫັບພາພຕ່າງ ພ້າ ມາທຳບວກຫີ່ອລົບກັນ ແລ້ວນຳເວກເຕອຣ໌ພລັບພົບເຂົ້າໄປເປັນອິນພຸດຂອງໂຄຮງຂ່າຍກ່ອກມີເນີດ ພລັບພົບທີ່ໄດ້ພບວ່າມີລັກຊະນະສຳຄັນຈາກພາພຂອງເວກເຕອຣ໌ທີ່ເປັນຕົວຖຸກທຳເນີນການ ພສມກັນໃນລັກຊະນະເຊີງເສັ້ນ.

ຕ້ວຍຢ່າງເຊັ່ນ ແຮດໂວຣັດແລະຄະນະ[156] ເລືອກພາພຸ້ໜູງຍື້ມສາມພາພອອກມາ ແລ້ວທາເວກເຕອຣ໌ເລີ່ມ $\bar{z}_{\text{smile,woman}}$ ຈາກເວກເຕອຣ໌ຄ່າສຸ່ມຂອງພາພທີ່ສາມ ນຳໄປລົບອອກດ້ວຍເວກເຕອຣ໌ $\bar{z}_{\text{neutral,woman}}$ ທີ່ເລີ່ມຈາກເວກເຕອຣ໌ຄ່າສຸ່ມຂອງພາພຸ້ໜູງໜ້າເຊຍ (ໄມ້ໄດ້ຍື້ມ) ແລ້ວນຳໄປບວກດ້ວຍເວກເຕອຣ໌ $\bar{z}_{\text{neutral,man}}$ ທີ່ເລີ່ມຈາກເວກເຕອຣ໌ຄ່າສຸ່ມຂອງພາພຸ້ໜ້າເຊຍ (ໄມ້ໄດ້ຍື້ມ) ສຸດທ້າຍນຳເວກເຕອຣ໌ພລັບພົບເຂົ້າໂຄຮງຂ່າຍກ່ອກມີເນີດ ແລະຮູບພາພທີ່ສ້າງຂຶ້ນມາ ພບວ່າເປັນພາພຸ້ໜ້າຍື້ມ. ນັ້ນຄື່ອງ ໂຄຮງຂ່າຍກ່ອກມີເນີດ ໄດ້ເຮັນຮູ່ທີ່ຈະເຂົ້າຮ້າສັກຊະນະສຳຄັນຂອງພາພໄວ້ໃນເວກເຕອຣ໌ຄ່າສຸ່ມ z ແລະການເຂົ້າຮ້າສ້າງເປັນໄປໃນລັກຊະນະເຊີງເສັ້ນ (ຈຶ່ງສາມາດລົບແລະບວກ ແລ້ວໄດ້ພລັບພົບໃນລັກຊະນະເຊີງເສັ້ນອອກມາ). ດ້ວຍຄຸນສົມບັດເຊັ່ນນີ້ ອາຈນອງວ່າໂຄຮງຂ່າຍກ່ອກມີເນີດໄດ້ເຮັນຮູ່ທີ່ຈະກຳນົດຄວາມໝາຍຂອງລັກຊະນະສຳຄັນໄວ້ທີ່ຄ່າຂອງເວກເຕອຣ໌ z . ເນື່ອຈາກຄວາມໝາຍຂອງລັກຊະນະສຳຄັນນີ້ ໄນໄດ້ຖຸກກຳນົດອອກມາຍ່າງໜັດເຈນ ຕ້ອງອາສີການສືບການສັງເກເຕີຈະພອເຫັນຄວາມເຂື່ອມໂຍງ ເວກເຕອຣ໌ z ບາງຄັ້ງຈຶ່ງຖຸກເຮັດວຽກເປັນລັກຊະນະໜີ່ອັນເຮັນ (latent representation) ແລະປະປຸງມີຂອງ z ຈຶ່ງມັກຖຸກອ້າງຄົງເປັນປະປຸງມີໜີ່ອັນເຮັນ ອີ່ວິປະປຸງມີຕົວແທນ ຕາມທີ່ອົງປະກາດໄປກ່ອນໜ້າ. ຮູບທີ່ 7.8 ແສດພາພປະກອບທີ່ວາດຂຶ້ນ (ດູພາພຈິງຈາກ [156]).

ແມ່ວ່າໂຄຮງຂ່າຍກ່ອກມີເນີດທີ່ຖຸກເຝຶກມາດີແລ້ວຈະສາມາດແປລັງຄ່າຈາກປະປຸງມີໜີ່ອັນເຮັນ ໄປສູ່ປະປຸງມີໜີ່ອັນເຮັນໄດ້ ແຕ່ເຊັ່ນເດືອກກັບໂຄຮງຂ່າຍປະສາທເຫີມທີ່ໄປ ຄື່ອງ ພາກທີ່ຕ້ອງການຈະຄຳນວນຢັ້ງກັນ ຊຶ່ງຄື່ອງການແປລັງຈາກພາພ X ກັບມາເປັນເວກເຕອຣ໌ໜີ່ອັນເຮັນ z ນັ້ນ ໄນໄມ້ສາມາດກຳນົດໄດ້ໂດຍຕຽນ. ອີ່ຢ່າງໄຮັກຕາມ ມີ

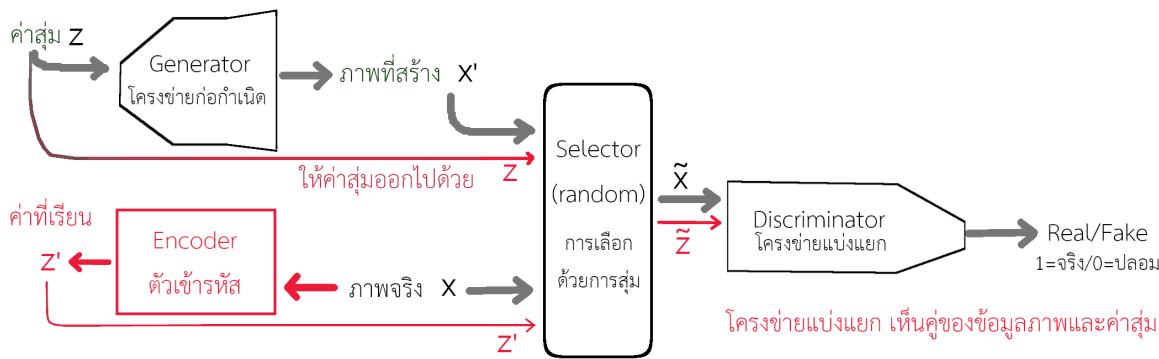


รูปที่ 7.8 การทำพีชคณิต图像เตอร์กับเวกเตอร์ค่าสุ่มของโครงข่ายก่อกำเนิด (ภาพวัดขึ้นโดยอาศัยตัวอย่างของแอดฟอร์ด และคณะ[156]). ภาพผู้หญิงยิ้มถูกคัดเลือกมาสามภาพ โดยเก็บเวกเตอร์ค่าสุ่ม z_i ของแต่ละภาพมาด้วย นำเวกเตอร์ค่าสุ่มมาหาค่าเฉลี่ย $\bar{z}_{s,w}$. ทำแบบเดียวกันกับภาพผู้หญิงหน้าเฉย และภาพผู้ชายหน้าเฉย ได้ค่าเฉลี่ย $\bar{z}_{n,w}$ สำหรับภาพผู้หญิงหน้าเฉย และ $\bar{z}_{n,m}$ สำหรับภาพผู้ชายหน้าเฉย. คำนวณ $z_{result} = \bar{z}_{s,w} - \bar{z}_{n,w} + \bar{z}_{n,m}$ และนำ z_{result} เข้าโครงข่ายก่อกำเนิด ภาพผลลัพธ์ $\mathbf{X}' = G(z_{result})$ ที่ได้พบว่า ภาพ \mathbf{X}' คล้ายภาพของผู้ชายยิ้ม. หมายเหตุ การทำพีชคณิตทำในปริภูมิช่องเร้น (ทำกับเวกเตอร์ค่าสุ่ม) ถึงจะได้ผลลัพธ์คล้ายการเลือกลักษณะสำคัญ. หากทำพีชคณิตในปริภูมิของภาพโดยตรง (เช่น $\mathbf{X}_{result} = \bar{\mathbf{X}}_{s,w} - \bar{\mathbf{X}}_{n,w} + \bar{\mathbf{X}}_{n,m}$) ภาพที่ได้จะเลอเทอะ และยากจะมองออก. ดูตัวอย่างภาพจริงจากแอดฟอร์ด และคณะ[156].

ความพยายามที่จะเพิ่มกลไกภายใน เพื่อให้โครงข่ายปรับปรุงเชิงสร้างสามารถคำนวนกลับไปกลับมาระหว่างปริภูมิช่องเร้น และปริภูมิข้อมูลได้. รูปที่ 7.9 แสดงโครงสร้างของใบแกน (BiGAN[56]). การอนุมานที่เรียนเชิงปรับปรุง (Adversarially Learned Inference คำย่อ ALI[59]) ซึ่งเป็นแนวคิดเดียวกันกับใบแกน ก็ถูกเสนอในช่วงเวลาเดียวกัน. กลไกที่สำคัญ สำหรับทั้งใบแกนและการอนุมานที่เรียนเชิงปรับปรุง คือ การเพิ่มตัวเข้ารหัส (encoder) เพื่อเรียนรู้ความสัมพันธ์ระหว่างปริภูมิช่องเร้นและปริภูมิข้อมูล. หากจำเพาะลงไปก็คือ ตัวเข้ารหัส ทำหน้าที่เรียนรู้การแยกแบบมีเงื่อนไข $p(\mathbf{z}|\mathbf{X})$ เมื่อ \mathbf{X} แทนข้อมูลภาพ และ \mathbf{z} คือ ค่าลักษณะช่องเร้น ที่โครงข่ายก่อกำเนิดใช้. อย่างไรก็ตาม เครื่องเวลและคณะ[45] ให้ความเห็นว่า ภาพที่สร้างจากใบแกนหรือการอนุมานที่เรียนเชิงปรับปรุงยังมีคุณภาพค่อนข้างต่ำ. นั่นอาจหมายถึงว่า การศึกษาวิจัย ถึงกลไกแปลงกลับจากปริภูมิข้อมูลไปสู่ปริภูมิช่องเร้น ในโครงข่ายปรับปรุงเชิงสร้าง ยังอยู่ในขั้นเริ่มต้นเท่านั้น (ณ เวลาที่เขียนต้นฉบับนี้).

ปัญหาในการฝึก. การฝึกโครงข่ายปรับปรุงเชิงสร้าง ถูกรายงาน[156, 45] ว่าทำได้ยาก และมีโอกาสล้มเหลวสูงมาก จากหลาย ๆ สาเหตุ รวมถึง

- การถูเข้ายาก[156] ที่นักวิจัยมักพบว่า มันยากที่ทำให้การฝึกโครงข่ายก่อกำเนิด ถูเข้า. ปัญหา



รูปที่ 7.9 โครงสร้างของใบแกนและการอนุมานที่เรียนเชิงปรับักษ์ เพื่อเรียนรู้ความสัมพันธ์ระหว่างปริภูมิช่องเร้นและปริภูมิข้อมูล. กลไกสำคัญอยู่ที่ตัวเข้ารหัส. โครงข่ายก่อกำเนิด เรียนรู้การแยกแยะแบบมีเงื่อนไข $p(\mathbf{X}|z)$. ส่วนตัวเข้ารหัส ที่เห็นข้อมูลจริง แต่พยายามเรียนรู้การแยกแยะแบบมีเงื่อนไข $p(z|\mathbf{X})$. ทั้งโครงข่ายก่อกำเนิดและตัวเข้ารหัส พยายามเรียนรู้เพื่อจะสร้างคู่ข้อมูล $(\tilde{\mathbf{X}}, \tilde{z})$ ที่โครงข่ายแบ่งแยก จำแนกได้ยากว่าเป็นภาพปลอม (คู่ (\mathbf{X}', z) จากโครงข่ายก่อกำเนิด) หรือเป็นภาพจริง (คู่ (\mathbf{X}, z') จากภาพจริงและตัวเข้ารหัส).

นี้ ส่วนหนึ่งอาจมาจากการรวมชาติของข้อมูลภาพ. ข้อมูลภาพมีปริภูมิที่ขนาดใหญ่มาก ๆ (รูปสีขนาด $W \times H$ พิกเซล เทียบเท่าจุดข้อมูลในปริภูมิขนาด $3 \cdot W \cdot H$ มิติ) แต่ตัวอย่างข้อมูลต่าง ๆ ที่มี (เช่น ภาพจริงต่าง ๆ) เป็นข้อมูลสำหรับการสนับสนุนของฟังก์ชันความหนาแน่นความน่าจะเป็น⁶ ครอบคลุมเพียงบริเวณเล็ก ๆ ในปริภูมิ และเมื่อเทียบกับขนาดของปริภูมิทั้งหมดแล้ว บริเวณที่ครอบคลุมมีขนาดเล็กมาก ๆ. กล่าวคือ เมื่อจะใช้ตัวอย่างข้อมูลจำนวนมากแล้ว แต่จำนวนตัวอย่างที่ใช้ ก็ยังน้อยมากเมื่อเทียบกับขนาดประชากร (โอกาสทั้งหมดที่เป็นไปได้ของข้อมูล) และตัวอย่างข้อมูลเหล่านี้ ก็ยากที่จะเป็นตัวแทนที่พอเพียงได้.

นอกจากนั้น ยังมีการศึกษา[45, 6] ที่พบว่า ก่อนการฝึก การแยกแยะก่อกำเนิด p_G อาจจะไม่มีการซ้อนทับกับการแยกแยะเป้าหมาย p_{data} เลย. หากเป็นเช่นนั้นจริง ผลคือ โครงข่ายแบ่งแยกจะสามารถถูกฝึกได้อย่างง่ายดายและรวดเร็ว เพื่อที่จะสามารถจำแนกตัวอย่างจริง $\mathbf{X} \sim p_{data}$ ออกจากตัวอย่างปลอม $\mathbf{X} \sim p_G$ ได้อย่างแม่นยำสมบูรณ์ (ความแม่นยำ 100%) นั่นคือ การฝึกโครงข่ายแบ่งแยกจะถูกเข้าใจ ฟังก์ชันจุดประสงค์มีค่าเป็นศูนย์⁷ ได้อย่างรวดเร็ว และส่งผลให้เกรดเดียนต์ต่าง ๆ เป็นศูนย์ ซึ่งจะทำให้การฝึกโครงข่ายก่อเนิดไม่สามารถทำต่อไปได้.

อีกประเด็นหนึ่ง เคร斯เวลและคณะ[45] อภิปรายประเด็นจากการศึกษาทฤษฎีโครงข่ายปรับักษ์เชิงสร้าง[77] กับฟังก์ชันจุดประสงค์ที่ใช้ ว่า หากโครงข่ายแบ่งแยกไม่ได้อยู่ในสภาพที่ดีที่สุดแล้ว การฝึกโครงข่ายก่อเนิดก็อาจจะไม่แม่นยำ หรืออาจได้ผลลัพธ์ผิดความหมายได้. นี่อาจหมายถึง ความจำเป็นในการออกแบบฟังก์ชันจุดประสงค์ใหม่ สำหรับโครงข่ายปรับักษ์เชิง

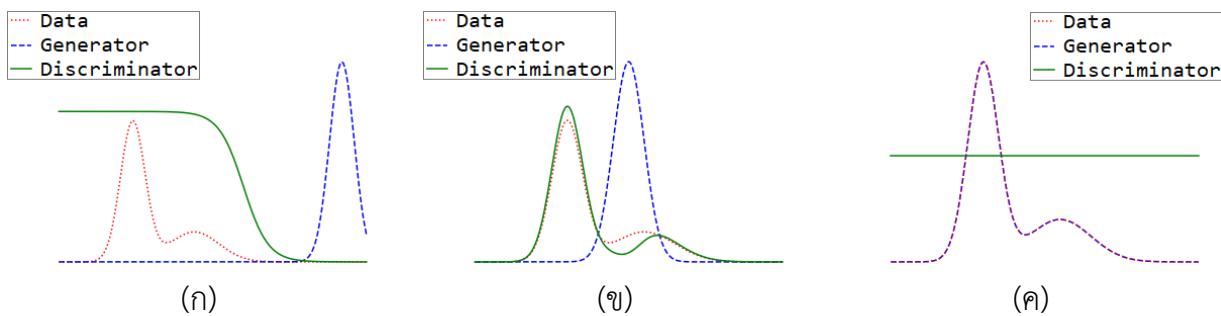
⁶ในทางคณิตศาสตร์และสถิติศาสตร์ การสนับสนุนของฟังก์ชันค่าจริง (*the support of a real-valued function*) หมายถึง เขตย่อของโดเมน ที่เซตย่ออยู่นั้นมีจุดข้อมูลที่ค่าของฟังก์ชันที่สนใจไม่เป็นศูนย์อยู่. ในบริบทของการเรียนรู้การแยกแยะ การสนับสนุนของฟังก์ชันความหนาแน่นความน่าจะเป็น หมายถึง เซตย่อในปริภูมิที่มีจุดข้อมูลจริงอยู่.

สร้าง. อย่างไรก็ตาม ด้วยฟังก์ชันจุดประสังค์ดังเช่นนิพจน์ 7.7 ประเด็นนี้ ที่เมื่อประกอบกับข้อ อภิปรายข้างต้นแล้ว จะช่วยให้เห็นความยากของการฝึกโครงข่ายปรปักษ์เชิงสร้าง ที่หากโครง ข่ายแบ่งแยกทำงานได้ดีเกินไป การฝึกโครงข่ายกำเนิดก็จะทำได้ยาก หรืออาจล้มเหลว และ หากโครงข่ายแบ่งแยกทำงานไม่ดีเลย การฝึกโครงข่ายกำเนิดก็จะไปผิดทาง.

- การพังทลายของภาวะ (mode collapse[173]) ที่หมายถึง โครงข่ายกำเนิดสร้างแต่เอาร์พุต ที่เหมือน ๆ กัน แม้ว่าจะรับอินพุตต่างกัน. จุดประสังค์ คือ ต้องการได้โครงข่ายกำเนิดที่สามารถ สร้างแต่เอาร์พุตออกมานี้ได้ โดยเอาร์พุตที่ได้ มีการแยกแยะคล้ายข้อมูลจริงมากที่สุด. ตัวอย่าง เช่น แทนที่โครงข่ายกำเนิดจะสามารถสร้างภาพเหมือนจริงใหม่ ๆ ออกมานี้ได้เรื่อย ๆ แต่โครง ข่ายกำเนิดกลับสร้างภาพเหมือนจริงภาพเดิม ๆ ออกมานี้ แม้ว่าจะรับอินพุต (ซึ่งคือค่าสุ่ม) ค่า ใหม่แล้วก็ตาม.
 - การฝึกโครงข่ายแบ่งแยกได้เร็วและดีเกินไป[45]. หากโครงข่ายแบ่งแยกทำงานได้ดีมาก ๆ อาจ ทำให้ $V(\mathcal{G}, \mathcal{D}) \approx 0$ ซึ่งมีผลให้เกรเดียนต์มีค่าใกล้ศูนย์ และทำให้การฝึกโครงข่ายก่อกำ- เนิดทำได้ยากมาก หรืออาจล้มเหลวได้. รูปที่ 7.10 แสดงสมมติฐานกลไกการเรียนรู้การแยกแยะ ของโครงข่ายปรปักษ์เชิงสร้าง ในสถานการณ์ต่าง ๆ.
- แม้มองผิวเผินอาจจะดูดี แต่การฝึกโครงข่ายแบ่งแยกให้ได้สมบูรณ์ก่อน (เมื่อเทียบกับความ สามารถของโครงข่ายก่อกำเนิด) แล้วจึงค่อยฝึกโครงข่ายก่อกำเนิด ไม่ใช่แนวทางที่นิยมปฏิ- บัติ ด้วยเหตุผลข้างต้น. แนวทางปฏิบัติที่นิยมคือ การฝึกโครงข่ายแบ่งแยกไปก่อนระยะหนึ่ง แล้วจึงค่อยฝึกโครงข่ายก่อกำเนิดไปพร้อม ๆ กัน. นอกจากนั้น ด้วยเหตุผลด้านเสถียรภาพ เชิงตัวเลข ในการฝึกโครงข่ายก่อกำเนิด มักนิยมใช้จุดประสังค์ $\max_{\mathcal{G}} E_{\mathbf{X} \sim p_{\mathcal{G}}} [\log \mathcal{D}(\mathbf{X})]$ มากว่า $\min_{\mathcal{G}} E_{\mathbf{X} \sim p_{\mathcal{G}}} [\log(1 - \mathcal{D}(\mathbf{X}))]$.

เทคนิคในการฝึกโครงข่ายปรปักษ์เชิงสร้าง. จากความท้าทายในการฝึกโครงข่ายปรปักษ์เชิงสร้าง ที่อภิปรายข้างต้น แรดฟอร์ดและคณะ[156] ได้เสนอดีชีแกน (DCGAN จาก Deep Convolutional Generative Adversarial Networks) เพื่อบรรเทาปัญหา. ดีชีแกน มีปัจจัยที่สำคัญคือ (1) การใช้ค่อน โวลุชั่นก้าวยาว (strided convolution) แทนการใช้ชั้นดึงรวม ในโครงสร้างของโครงข่ายแบ่งแยก \mathcal{D} . ค่อนโวลุชั่นก้าวยาว หมายถึง ชั้นค่อนโวลุชั่นที่ใช้ก้าวย่างขนาดใหญ่กว่าหนึ่ง เช่น การใช้ขนาดก้าวย่าง $S = 2$. ผลลัพธ์ของการใช้ค่อนโวลุชั่นก้าวยาว จะให้ผลเหมือนการลดขนาดแผนที่ลักษณะลำคัญลง (หรือ spatially downsampling). (2) การใช้ชั้นค่อนโวลุชั่น โดยเฉพาะใช้การทำการค่อนโวลุชั่นก้าวเศษ (fractionally-strided convolution หรือ transposed convolution[60]) ในโครงสร้างของโครง ข่ายก่อกำเนิด \mathcal{G} .

หากจะอธิบายโดยง่ายแล้ว ภายในตัวบริบทนี้ ค่อนโวลุชั่นก้าวเศษ ก็เสมือน การขยายขนาดแผนที่ ลักษณะลำคัญที่เป็นอินพุต แล้วจึงทำการคำนวนค่อนโวลุชั่น. การขยายขนาดแผนที่ลักษณะลำคัญ

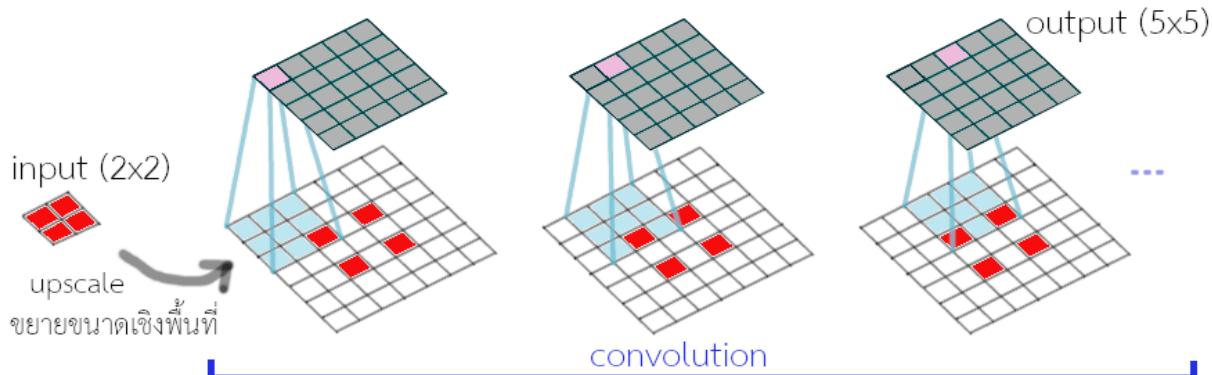


รูปที่ 7.10 ภาพแสดงสมมติฐานการเรียนรู้การแยกแยะข้อมูลของโครงข่ายประปักษ์เชิงสร้าง. ทั้งสามภาพ แสดง ภาพอย่างง่ายของค่าความหนาแน่นความน่าจะเป็น (แกนตั้ง) ต่อค่าข้อมูล (แกนนอน) ของข้อมูลจริง (แสดงด้วยเส้นไข่ปลา) กับ ของที่สร้างขึ้นจากโครงข่ายก่อกำเนิด (แสดงด้วยเส้นประ) พร้อมทั้งแสดงค่าเออร์พุตของโครงข่ายแบ่งแยก (แสดงด้วยเส้นทึบ). ภาพซ้ายสุด (ก) การแยกแยะของข้อมูลจริง ต่างจากการแยกแยะจากโครงข่ายก่อกำเนิดมาก ไม่มีส่วนที่ซ้อนทับกันเลย. โครงข่ายแบ่งแยก สามารถจำแนกข้อมูลจริงออกจากข้อมูลปลอมได้อย่างสมบูรณ์ ด้วยความแม่นยำสูงสุด. กรณีเช่นนี้ จะทำให้การฝึกโครงข่ายก่อกำเนิดไม่สามารถดำเนินการต่อได้. ภาพกลาง (ข) การแยกแยะของข้อมูลจริง ต่างจากการแยกแยะจากโครงข่ายก่อกำเนิด แต่มีส่วนซ้อนทับกันอยู่มาก. โครงข่ายแบ่งแยก ไม่สามารถจำแนกข้อมูลจริงออกจากข้อมูลปลอม ได้อย่างสมบูรณ์. การฝึกโครงข่ายก่อกำเนิด สามารถดำเนินการต่อไปได้. ภาพขวา (ค) โครงข่ายก่อกำเนิดสามารถเรียนรู้ การแยกแยะของข้อมูลจริง และสามารถสร้างข้อมูลจากการแยกแยะที่เหมือนของข้อมูลจริง. โครงข่ายแบ่งแยก ไม่สามารถ จำแนกข้อมูลจริงออกจากข้อมูลปลอมได้เลย.

(ซึ่งขยายเฉพาะในมิติลำดับเชิงพื้นที่) ทำด้วยการเติมค่าศูนย์เข้าไประหว่างค่าพิกเซลเดิม (รวมการเติม เหตุผลด้วยศูนย์ ที่เติมค่าศูนย์ที่บริเวณขอบของแผนที่ด้วย). ผลลัพธ์ของการใช้คอนโวลูชันก้าวเศษ จะให้ผลเหมือนการเพิ่มขนาดแผนที่ลักษณะสำคัญขึ้น (หรือ spatially upsampling). รูปที่ 7.11 แสดง กลไกที่ค่อนโวลูชันก้าวเศษ ช่วยขยายขนาดแผนที่ลักษณะสำคัญขึ้น. หากสังเกตการทำค่อนโวลูชันใน รูป เมื่อมองจากปฏิสัมพันธ์ระหว่างฟิลเตอร์และอินพุต อาจดูเหมือนกับว่าฟิลเตอร์ขยายบ้านพิกเซลข้า ลง คล้ายกับว่า ใช้ขนาดก้าวย่างที่เล็กกว่าหนึ่ง ซึ่งเป็นที่มาของชื่อ ค่อนโวลูชันก้าวเศษ.

หมายเหตุ ค่อนโวลูชันก้าวเศษ บางครั้งเรียก ค่อนโวลูชันลับเปลี่ยน (transposed convolution) ซึ่งมาจากการตีความทางคณิตศาสตร์. นั่นคือ หากดำเนินค่อนโวลูชันด้วยการแปลงอินพุตและค่า- น้ำหนักของฟิลเตอร์เป็นเมทริกซ์ โดยจัดรูปเมทริกซ์ทั้งสองให้ถูกต้อง (มีการใช้ค่าซ้ำและมีการเติม ศูนย์เข้าไป ดูแบบฝึกหัด 6.7 ประกอบ) ซึ่งทำให้ได้เมทริกซ์ของค่าน้ำหนักเป็น เมทริกซ์มากເเหลศูนย์ (sparse matrix) และ การทำค่อนโวลูชัน ก็เหมือนกับการคุณเมทริกซ์อินพุตเข้ากับเมทริกซ์ค่าน้ำหนัก แล้วนำผลลัพธ์ที่ได้ไปจัดรูปให้เข้ากับโครงสร้างที่ถูกต้อง. ในทำนองเดียวกัน ค่อนโวลูชันก้าวเศษ ก็ เสมือนการคุณเมทริกซ์อินพุตเข้ากับการลับเปลี่ยนของเมทริกซ์ค่าน้ำหนัก. ดังนั้น กระบวนการนี้จึง เรียกว่า ค่อนโวลูชันลับเปลี่ยน. (ศึกษาเพิ่มเติมได้จาก ดูมูลินและวิชัน[60])

ค่อนโวลูชันก้าวเศษ บางครั้ง อาจถูกเรียกว่า การถอดค่อนโวลูชัน (deconvolution). อย่างไร ก็ตาม การถอดค่อนโวลูชัน มีความหมายอื่น (ซึ่งเป็นคนละเรื่อง) และถูกยอมรับมากกว่า. ความหมายที่ ถูกยอมรับมากกว่าของ การถอดค่อนโวลูชัน คือ การถอดพารามิเตอร์ของโครงข่ายค่อนโวลูชันย้อนกลับ เพื่อศึกษาภัยการทำงานของโครงข่ายค่อนโวลูชัน ว่า ฟิลเตอร์แต่ละตัวที่ใช้ในโครงข่ายค่อนโวลูชัน



รูปที่ 7.11 คอนโวโลชันก้าวเศษซ่วยขยายขนาดแผนที่ลักษณะสำคัญ (ภาพดัดแปลงจาก ดูมูลินและวิชิน[60, รูป 4.5]). อินพุตขนาด 2×2 พิกเซล ถูกขยายเป็นເອົາຕົ້ມພຸດທະນາດ 5×5 พิกเซล ເນື່ອໃຫ້ພິລເຕອຣ໌ທະນາດ 3×3 ໂດຍການເຕີມສູນຍໍ່ທີ່ນີ້ ຕ້ວຮະຫວ່າງພິກເຊລ ແລະ ທຳການເຕີມເຕີມຕ້ວຍຄູນຍໍ່ດ້ວຍຄູນຍໍ່ສອງຕົວທີ່ຂອບແຕ່ລະຂ້າງ. ພະນັກເອົາຕົ້ມພຸດ $o' = s(i' - 1) + k$ ເມື່ອ i' ຄືອ່ານາດອິນພຸດ ແລະ k ຄືອ່ານາດພິລເຕອຣ໌ ແລະ ເຕີມສູນຍໍ່ຮ່ວ່າງພິກເຊລ $s - 1$ ຕ້າວ. ກຽນນີ້ $o' = 2(2 - 1) + 3 = 5$. (ດູມຸລິນແລະວິຈິນ[60].)

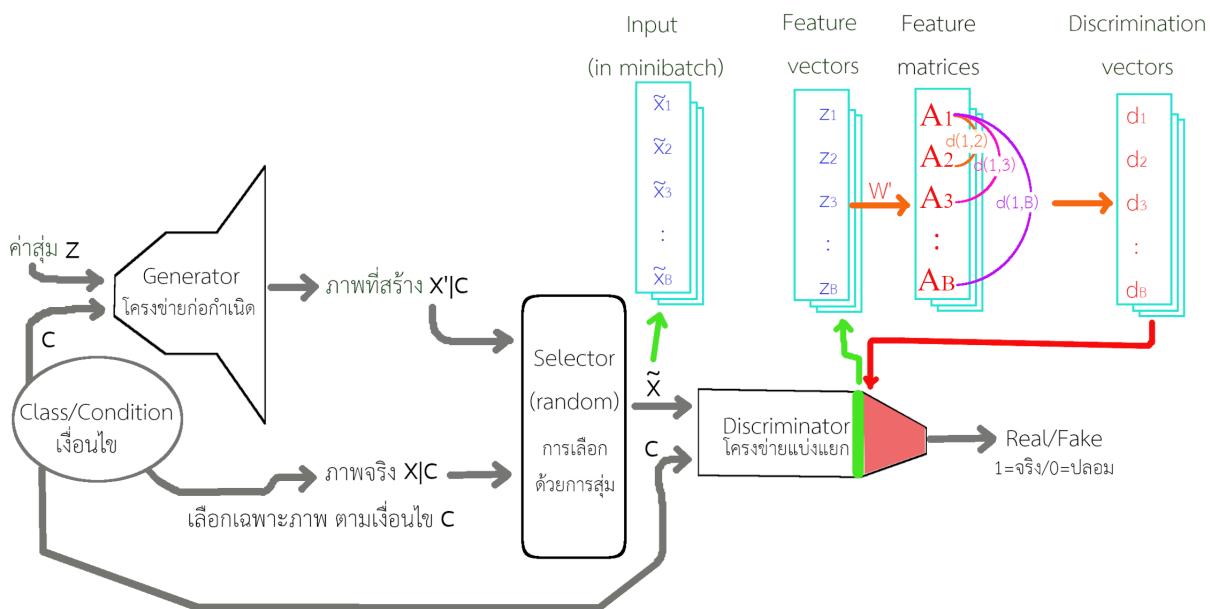
ໄດ້ເຮີນຮູ້ເພື່ອຈະຕຽບຈັບລັກຂະນະຮູປແບບເບິ່ງໄວ. ສໍາຫັບການຄອດຄອນໂວລູ້ໜັ້ນ ໃນຄວາມໝາຍທີ່ນີ້ມີສາມາດສຶກສາເພີ່ມເຕີມໄດ້ຈາກທົກວາມ [223, 224, 187] ເປັນຕົ້ນ.

ນອກຈາກການໃຊ້ຄອນໂວລູ້ໜັ້ນກ້າວຍາວແລະຄອນໂວລູ້ໜັ້ນກ້າວເຄີຍໃນໂຄຮງສ້າງຂອງໂຄຮງຂ່າຍແບ່ງແຍກແລະໂຄຮງຂ່າຍກ່ອກກຳນຳແລ້ວ ແຮດຝອຣົດແລະຄນະ[156] ຍັງແນະນຳການໃຊ້ແບ່ນອົບມ (ຫວັງຂ້ອ 5.5), ແນະນຳໃຊ້ໜັບຄອນໂວລູ້ໜັ້ນລຶກ ຖ (ຫລາຍ ຖ ຊັ້ນ) ແນການໃຊ້ໜັບເຊື່ອມຕ່ອເຕີມທີ່⁷, ແນະນຳການໃຊ້ເຮົງ ສໍາຫັບຝັກໆ-ໜັນກະຮະຕຸນຂອງທຸກ ຖ ຊັ້ນຄໍານວນໃນໂຄຮງຂ່າຍກ່ອກກຳນຳ ຍັກເວັນໜັບເອົາຕົ້ມພຸດທີ່ແນະນຳໃຫ້ໃຊ້ເປົ້ອບອລິກ ແນກເຈນຕົ້ນ, ແນະນຳການໃຊ້ເຮົງຮົວ ສໍາຫັບຝັກໆ-ໜັນກະຮະຕຸນຂອງທຸກ ຖ ຊັ້ນຄໍານວນໃນໂຄຮງຂ່າຍແບ່ງແຍກ.

ເທົ່ານີ້ ຖ ທີ່ມີຮາຍງານວ່າ ເປັນປັຈຢັງສຳຄັນ ປ່ວຍການຝຶກໂຄຮງຂ່າຍປັບປຸງເຊີງສ້າງ ໄດ້ແກ່ ການຈັບຄູ່ລັກຂະນະສຳຄັນ[173], ການແຍກແຍ່ງໜູ່ເລີກ[173], ການແລ້ຍຕາມປະວັດ[173], ການທຳລາກຮາບຮັນທາງເດືອນ[173], ການທຳໜູ່ເລີກເລີນຈິງ[173], ການໃລ່ສັນຍານົບກວນ[197, 6] ໄປຈົນລຶງການເປີ່ຍິນຝັກໆ-ໜັນຈຸດປະສົງ (ດູ [142, 7] ເພີ່ມເຕີມ) ເປັນຕົ້ນ. ເຄຣສເວລແລະຄນະ[45] ໄທ້ຄວາມເຫັນວ່າ ໂຄຮງຂ່າຍປັບປຸງເຊີງສ້າງທີ່ຝຶກໄດ້ຈຳກັດທີ່ສຸດ ນໍາຈະເປັນແບບຈຳລອງທີ່ເສັນໂດຍຄນະຂອງອາຣ໌ໂຈຟສັກ[7] ຢ່ອຍອັນຄນະຂອງມາຄົງໜານ[125].

ການຈັບຄູ່ລັກຂະນະສຳຄັນ (feature matching) ເປີ່ຍິນຝັກໆ-ໜັນຈຸດປະສົງສໍາຫັບໂຄຮງຂ່າຍກ່ອກກຳນຳ ເປັນ $\min_{\mathcal{G}} \|E_{\mathbf{X} \sim p_{data}}[\mathbf{f}(\mathbf{X})] - E_{\mathbf{X} \sim p_{\mathcal{G}}}[\mathbf{f}(\mathbf{X})]\|_2^2$ ເນື່ອ $\mathbf{f}(\mathbf{X})$ ເປັນລັກຂະນະສຳຄັນທີ່ໄດ້ຈາກໂຄຮງຂ່າຍແບ່ງແຍກ (ຄ່າເວັກເຕອຣ໌ຂອງໜັນຄໍານວນໜັ້ນທີ່ອູ່ຢ່າງໃນໂຄຮງສ້າງຂອງໂຄຮງຂ່າຍແບ່ງແຍກ) ໃນຂະໜາດທີ່ຍັງໃຊ້ຝັກໆ-ໜັນຈຸດປະສົງແບບເຕີມສໍາຫັບໂຄຮງຂ່າຍແບ່ງແຍກ (ເຊັ່ນ $\max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) = E_{\mathbf{X} \sim p_{data}}[\log \mathcal{D}(\mathbf{X})] + E_{\mathbf{X} \sim p_{\mathcal{G}}}[\log(1 - \mathcal{D}(\mathbf{X}))]$).

⁷ອ່າຍ່າໃກ້ຕາມ ໃນໂຄຮງຂ່າຍກ່ອກກຳນຳ ການໃຊ້ໜັບເຊື່ອມຕ່ອເຕີມທີ່ເປັນໜັນຄໍານວນແຮກ (ໜັນທີ່ຮັບອິນພຸດເປັນເວັກເຕອຣ໌ຄ່າສຸມ) ອາຈະສະດວກທີ່ຈະໃຊ້ໜັບເຊື່ອມຕ່ອເຕີມທີ່ ມາກກ່າວການໃຊ້ໜັບຄອນໂວລູ້ໜັ້ນ ຍັງແມ່ຈະມີຫລາຍວິທີທີ່ຈະປະຍຸດໃຫ້ໜັບຄອນໂວລູ້ໜັ້ນກັບອິນພຸດເຕອຣ໌ໄດ້ກືດາມ ອາທີ ກາຈດເຮັງໂຄຮງສ້າງຂອງອິນພຸດໃໝ່ ໃຫ້ເໜາະກັບການທຳຄອນໂວລູ້ໜັ້ນ ເປັນຕົ້ນ.



รูปที่ 7.12 การแยกแยะหมู่เล็ก. แต่ละอินพุตในหมู่เล็ก \tilde{x}_n จะถูกแปลงเป็นเวกเตอร์ลักษณะสำคัญ z_n (เลือกมาจากເອົາດີພຸດຂອງຫັ້ນຄຳນວນຫັ້ນທີ່ໃນໂຄຮງຂ່າຍແຍກແຍກ). ການແຍກແຍກหมู่เล็ก ຈະແປລງເວກເຕັກລັກຊະນະສຳຄັຟ z_n ເປັນເມທຣິກ໌ລັກຊະນະສຳຄັຟ \mathbf{A}_n . ແລ້ວຈາກນັ້ນ ວັດຄວາມຕ່າງ ຮະຫວ່າງເມທຣິກ໌ລັກຊະນະສຳຄັຟນັ້ນ ເບຣີຍບເຫັນກັບເມທຣິກ໌ລັກຊະນະສຳຄັຟອື່ນ ຈະໃນหมู่เล็ก ແລ້ວສຽງປອກມາເປັນເວກເຕັກແຕ່ວິຍແຍກແຍກ d_n . ຄ່າຂອງເວກເຕັກແຕ່ວິຍແຍກແຍກຈະຄູກນຳໄປປະກອບເປັນອິນພຸດສະເໝີໃຫ້ກັບຫັ້ນຄຳນວນຕ່ອງໄປໃນໂຄຮງຂ່າຍແຍກແຍກ. ການພ້າທລາຍຂອງກວາວ ທີ່ໂຄຮງຂ່າຍກ່ອນມີກຳນົດມັກສ້າງເອົາດີພຸດ ຄລ້າຍ ຈະກັນອອກມາ ຈະທຳໃຫ້ເວກເຕັກແຕ່ວິຍແຍກແຍກມີຂາດເລື່ອ ເມື່ອເປີຍບເຫັນກັບອິນພຸດຈິງທີ່ມີຄວາມຫລາກຫລາຍ. ດັ່ງນັ້ນໂຄຮງຂ່າຍແຍກແຍກ ຈະສາມາດຈຳແນກອິນພຸດປລອມຈາກການພ້າທລາຍຂອງກວາວໄດ້ຈ່າຍຫຼັ້ນຈາກເວກເຕັກແຕ່ວິຍແຍກແຍກ ແລ້ວເກົ່າກຳກັບກວາວໄດ້ດີ່ເກີ້ນ.

การແຍກແຍກหมู่เล็ก (minibatch discrimination) ເພີ່ມສ້າງມູນສາຮສະເໜີທີ່ໃໝ່ຢູ່ໂຄຮງຂ່າຍແຍກແຍກວ່າອິນພຸດທີ່ໄດ້ ເໜື່ອນຫຼືອແຕກຕ່າງຈາກອິນພຸດອື່ນໃນມູ່ເລື່ອການນ້ອຍຂາດໄໝ່. ດັ່ງນັ້ນ ໂຄຮງຂ່າຍແຍກແຍກ ຈະສາມາດຮະບູອິນພຸດປລອມທີ່ມີປຸງຫາໂຄຮງຂ່າຍແຍກແຍກໄດ້ອ່າຍໆຈ່າຍດ້າຍ.

รูปที่ 7.12 แสดงกลไกของการແຍກແຍກມູ່ເລື່ອການມູ່ເລື່ອການແຍກແຍກມູ່ເລື່ອການ. ຄົນຂອງແຜລລິມັນສ [173] ແປລງເວກເຕັກລັກຊະນະສຳຄັຟ $z_n \in \mathbb{R}^M$ (ເລື່ອກຈາກຫັ້ນຄຳນວນໃນໂຄຮງຂ່າຍແຍກແຍກ ແລ້ວ M ເປັນຈຳນວນຂອງເວກເຕັກ) ໄທເປັນເມທຣິກ໌ລັກຊະນະສຳຄັຟ $\mathbf{A}_n \in \mathbb{R}^{P \times Q}$ (ດ້ວຍການຄຸນກັບເຫັນເຊື່ອຮັບຄ່ານໍາຫັກ \mathbf{W}' ທີ່ປັບຄ່າໄດ້ໃນກະບວນການຝຶກ) ເມື່ອ P ແລ້ວ Q ເປັນຈຳນວນແກ່ລະສົມກົດທີ່ຕ້ອງການ. ຄວາມຕ່າງຮ່າງເມທຣິກ໌ລັກຊະນະສຳຄັຟ ຖືກະບູດ້ວຍເວກເຕັກ $\mathbf{d}(i, j) \in \mathbb{R}^P$ ທີ່ມີສ່ວນປະກອບ $d_r(i, j) = \exp(-\|\text{row}_r(\mathbf{A}_i) - \text{row}_r(\mathbf{A}_j)\|_1)$ ສໍາຮັບ $r = 1, \dots, P$ ເມື່ອ ດັ່ງນີ້ຂອງຕ້ວອຍ່າງໃນມູ່ເລື່ອການ $i, j \in \{1, \dots, B\}$ ແລ້ວ B ອີ່ຈຳນວນຕ້ວອຍ່າງໃນມູ່ເລື່ອການ. ຕ້ວດໍາເນີນການ $\text{row}_r(\mathbf{A})$ ມໍາຍືງ ແລ້ວທີ່ r^{th} ຂອງເມທຣິກ໌ \mathbf{A} ແລ້ວ $\|[v_1, v_2, \dots, v_N]^T\|_1 = \sum_{n=1}^N |v_n|$ ຢີ້ວ L^1 ນອർມ (L1 norm). ຄວາມຕ່າງຮ່າງເມທຣິກ໌ ຖື ສຽງປອກມາເປັນເວກເຕັກແຕ່ວິຍແຍກແຍກ $\mathbf{d}_i = [\sum_{j=1}^B \mathbf{d}_1(i, j), \dots, \sum_{j=1}^B \mathbf{d}_P(i, j)]^T$ ແລ້ວຄ່າເວກເຕັກ \mathbf{d}_i ບືກປັນຮ່ວມກັບເວກເຕັກລັກຊະນະສຳຄັຟ z_i (ສໍາຮັບຕ້ວອຍ່າງທີ່ i^{th} ໃນມູ່ເລື່ອການ) ເຂົ້າໄປສູ່ຫັ້ນຄຳນວນຕ່ອງໄປໃນໂຄຮງຂ່າຍແຍກແຍກ.

การเฉลี่ยตามประวัติ (*historical averaging*) เป็นการเพิ่มพจน์ที่ช่วยลดการเปลี่ยนค่าพารามิเตอร์อย่างรุนแรง ในระหว่างการฝึก เพื่อช่วยให้ระบบเข้าสู่สมดุลย์ได้ง่ายขึ้น. ตัวอย่างเช่น คณะของแซลลีมันส์[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 $\boldsymbol{\theta}^{(i)}$ แทนค่าในสมัยผีกที่ i^{th} ของพารามิเตอร์ต่าง ๆ และ T คือจำนวนสมัยผีกที่ผ่านมา. พจน์ที่คณะของแซลลีมันส์ เพิ่มเข้าไปเป็นระยะทางยุคลีเดียน ระหว่างค่าพารามิเตอร์ปัจจุบัน กับค่าเฉลี่ยที่ผ่านมา. การเพิ่มพจน์นี้เข้าไปในฟังก์ชันสูญเสีย ส่งผลให้กระบวนการฝึกลดการปรับค่าพารามิเตอร์อย่างมากลงได้. อย่างไรก็ตามการใช้การเฉลี่ยตามประวัติ ควรทำอย่างระมัดระวัง และควรเลือกค่า ϵ ให้เหมาะสม ไม่ใช่ค่าฟังก์ชันสูญเสียเดิม กับค่าของพจน์การเฉลี่ยตามประวัตินี้อย่างเหมาะสม.

การทำลาก拉บรืนทางเดียว (*one-sided label smoothing*) คือ การเปลี่ยนค่าเป้าหมายของฉลากเฉลี่ยของโครงข่ายเบ่งแยก จากเฉลี่ยว่าเป็นภาพจริง $y = 1$ ลดลงเป็น $1 - \epsilon$ แต่ค่าเฉลี่ยภาพปลอม $y = 0$ ไว้เหมือนเดิม เช่น เปลี่ยนจากเฉลี่ยภาพจริง จากค่า 1 เป็น 0.9 ($\epsilon = 0.1$). เทคนิคนี้ดัดแปลงจากการทำลาก拉บรืน (*label smoothing*) เพื่อป้องกันไม่ได้โครงข่ายเบ่งแยกมีความมั่นใจมากเกินไป.

การทำลาก拉บรืน[132, 196] เป็นเทคนิคที่เปลี่ยนค่าเป้าหมายของฉลากเฉลี่ย เพื่อป้องกันไม่ให้แบบจำลองมีความมั่นใจมากเกินไป (over-confidence). การทำลาก拉บรืน ตั้งเดิมออกแบบมาสำหรับการจำแนกคุณิต (multi-class classification) โดยการปรับค่าเป้าหมายของฉลากเฉลี่ยสำหรับประเภท k^{th} เป็น $q_k = (1 - \epsilon)y_k + \epsilon p(k)$ เมื่อ y_k คือค่าฉลากเฉลี่ยของประเภท k^{th} (อยู่ในรูปรหัสหนึ่งร้อน นั่นคือ $y_k = 1$ เมื่อเฉลี่ยคือชนิด k^{th} และ $y_k = 0$ เมื่อเฉลี่ยไม่ใช่ชนิด k^{th}) และ $p(k)$ คือการแจกแจงของข้อมูลชนิด k^{th} ส่วน ϵ คือภิมานพารามิเตอร์ที่เลือกได้ตามความเหมาะสม.

เซเจดีและคณะ[196] เลือกประมาณ $p(k)$ ด้วยการแจกแจงเอกรูป นั่นคือ ค่าฉลากเฉลี่ย $q_k = (1 - \epsilon)y_k + \frac{\epsilon}{K}$ เมื่อ K คือจำนวนประเภททั้งหมด. ตัวอย่างเช่น กรณีการจำแนก 5 ประเภท แล้วเลือก $\epsilon = 0.1$ ค่าเป้าหมาย จะถูกปรับเป็น 0.92 สำหรับประเภทที่ถูกต้อง และ 0.02 สำหรับประเภทที่ไม่ถูกต้อง. ดังนั้นแบบจำลองที่ถูกฝึกอย่างดีแล้วจะปรับค่าทำนายเข้ามาที่ 0.92 ซึ่งอาจตีความว่ามั่นใจมาก แต่ไม่ร้อยเปอร์เซ็นต์ มีเพื่อให้ไว้บ้าง ซึ่งหากมองเชิงการคำนวณ การปรับค่าฉลากเฉลี่ยจะช่วยป้องกันไม่ให้แบบจำลองถูกปรับค่าเข้าไปสู่ช่วงอิมตัว (*saturation region*). แบบจำลองที่ถูกปรับค่าเข้าไปสู่ช่วงอิมตัว จะทำให้การฝึกต้องทำได้ยาก และให้ผลลัพธ์การเกิดโควาอร์ฟิต. (ดูแบบฝึกหัด 7.5 เพิ่มเติมสำหรับการทำลาก拉บรืน)

เพื่อลดการขึ้นกับข้อมูลภายในหมู่เล็กมากเกินไป เมื่อใช้แบบอร์ม คณะของแซลลีมันส์[173] ใช้การทำหมู่เล็กเสมือนจริง (*virtual minibatch*) ทำแบบอร์มกับจุดข้อมูล โดยใช้ค่าสถิติที่คำนวณจากจุดข้อมูลนั้น ๆ และหมู่อ้างอิง (*reference batch*) ซึ่งหมู่อ้างอิง ถูกเลือกขึ้นมาก่อนการฝึก และใช้หมูนี้ตลอด (ไม่มีการเปลี่ยนแปลง). เนื่องจากการทำหมู่เล็กเสมือนจริง ทำการคำนวณมากขึ้น เพราะว่า ต้องทำการคำนวณไปข้างหน้า (*forward pass*) สำหรับสองหมู่เล็ก ดังนั้นคณะของแซลลีมันส์ จึง

ใช้การทำหมุนเล็กเลนโน่นจริงเฉพาะกับการฝึกโครงข่ายก่อกำเนิด

การใส่สัญญาณรบกวน (*noise addition*) คือการใส่สัญญาณรบกวน เช่น สัญญาณรบกวนที่มีการแจกแจงแบบเกาส์เซียน เข้าไปในทั้งภาพจริง และภาพที่สร้างจากโครงข่ายก่อกำเนิด. ชอนเดอบายและคณะ[197] อ้างว่าการใส่สัญญาณรบกวน ให้ผลดีกว่าการทำฉลากกราฟเร็นท์ทางเดียว. การใส่สัญญาณรบกวนเข้าไปกับทั้งภาพจริงและภาพปลอม เป็นคล้าย ๆ 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 already, without a shadow of doubt, what is laid before him.”

---Leo Tolstoy

“เรื่องที่ยากที่สุดสามารถอธิบายให้คนที่หัวข้าที่สุดเข้าใจได้ ถ้าเขามิฝังใจคิดไปเอง ก่อนแล้ว. แต่เรื่องที่ง่ายที่สุดไม่อาจจะอธิบายให้คนที่ฉลาดที่สุดเข้าใจได้ ถ้าเขามั่นใจดี ติดสิ่งที่เขาคิด โดยที่ไม่สนใจความจริงที่อยู่ตรงหน้าเลยสักนิด.”

—ลีโอ โตล์สตอย

เกร็ดความรู้ รูปแบบ “ประหลาด” ของจิต แม้ว่า ความเชื่อหลักในวงการแพทย์ เชื่อว่า (1) จิตเกิดจากสมอง และ (2) ชีวิตสิ้นสุดเมื่อคนตาย. แนวคิดนี้ อีดเวิร์ด เคลลี่[108] เรียกว่า กายภาพนิยม (*physicalism*). กายภาพนิยม เป็นกรอบความคิด และมุ่งมองโลกที่มองว่า ทุกอย่างเป็นกายภาพ. นั่นรวมถึง ความคิดที่ว่า จิตเกิดมาจากกิจกรรมของสมอง สติ และความรู้ตัวก็เป็นผลพลอยได้จากการของเซลล์ประสาท และเนื่องจากแนวคิดนี้เชื่อว่า จิตมาจากการของ ดังนั้นมือตัวตาย สมองหยุดทำงาน จิตจะจบหายไป. และถึงแม้ว่าคนส่วนใหญ่ก็เชื่อเช่นนั้น แต่แนวคิดนี้ก็ไม่ได้ถูกพิสูจน์หรือทดสอบ และติพมพ์อย่างเป็นทางการเลย จนกระทั่งงานศึกษาที่สำคัญของพาร์เนียและคณะ[150] กับทีมของแวนโนลมเมล[202].

ขณะ พาร์เนีย เป็นแพทย์โรคหัวใจ ซึ่งเชี่ยวชาญในการกู้ชีพผู้ป่วยหัวใจวาย เนื่องจากต้องการลดความเสี่ยงภาวะเจ้าชายนิทรรขอผู้ป่วย พาร์เนียจึงได้ศึกษาวิจัยเกี่ยวกับสติรู้ตัว (*consciousness*) และรวมไปถึง การศึกษาประสบการณ์เฉียดตาย (*Near Death Experience* คำย่อ NDE) ของผู้ป่วย.

ประสบการณ์เฉียดตาย เป็นประสบการณ์การรับรู้ของผู้ป่วยที่อยู่ในสถานการณ์ที่ใกล้จะตาย หรืออย่างครั้งผู้ป่วยหยุดหายใจ หัวใจหยุดเต้น และไม่มีกิจกรรมทางสมองแล้ว แต่แพทย์ พยาบาล เจ้าหน้าที่ สามารถกู้ชีพกลับมาได้. แม้เชื่อจะบอกว่าเป็น ประสบการณ์เฉียดตาย แต่จริง ๆ แล้ว ส่วนใหญ่ผู้ที่มีประสบการณ์นี้ ก็คือ ผู้ป่วยที่ได้ตายไปแล้วในช่วงเวลาสั้น ๆ แต่ได้รับการกู้ชีพกลับมาสำเร็จ. ถึงแม้ จะมีรายงานประสบการณ์เฉียดตาย จากอาการป่วยหลักหลายประเภท แต่งานวิจัยของพาร์เนียและคณะจะเน้นที่กลุ่มผู้ป่วยภาวะหัวใจวาย.

จากการวิจัย[149] พาร์เนียและคณะพบว่า ในจำนวนผู้ป่วยที่รอดชีวิตและให้สัมภาษณ์ มีราว ๆ 46% ที่มีความทรงจำ โดย 9% จัดเป็นประสบการณ์เฉียดตาย (ตามเงื่อนไขที่กำหนดในงานวิจัย). มี 2% ที่รู้ตัว โดย “เห็น” หรือ “ได้ยิน” เหตุการณ์เกี่ยวกับการกู้ชีพอย่างชัดเจน. มีกรณีหนึ่งที่ยืนยันได้ว่า ช่วงที่ผู้ป่วยรู้ตัวอยู่นั้น ไม่พบกิจกรรมหรือสัญญาณทางสมอง.

การที่มีการรู้ตัวในช่วงที่ไม่พบกิจกรรมทางสมอง อาจบอกได้ว่า (1) จิตไม่ได้ถูกสร้างจากสมอง หรือ (2) วิธีการวัดในปัจจุบันไม่สามารถวัดกิจกรรมที่เกี่ยวข้องนี้ได้ ซึ่งเป็นกันยอย่างชัดเจนว่า เราไม่ได้มีเครื่องมือหรือวิธีวัดที่จะวัดได้ทุกอย่าง โดยเฉพาะเรื่องเกี่ยวกับจิต. การอ้างการรู้ตัวในช่วงระหว่างการกู้ชีพได้ถูกตรวจสอบอย่างละเอียด. หนึ่งในการทดสอบก็คือ การทดสอบประสบการณ์ออกจากร่าง (*out-of-body experience*) ที่มักบรรยายถึง ความรู้สึกอยู่ออกจากร่างกายของ

ตัวเอง และมองเห็นภาพต่าง ๆ จากมุมสูง. คณของพาร์เนียเตรียมการทดสอบ โดยการติดตั้งห้องไว้ในห้องที่มีโอกาสสูงที่จะเกิดเหตุการณ์หัวใจวาย. บนตั้ง จวางรูปเอาไว้ โดยรูปหันหน้าขึ้นเพดาน ซึ่งผู้ที่อยู่ในห้องไม่สามารถที่จะมองเห็นภาพในรูป. ภาพในรูปจะใช้เพื่อพิสูจน์ความถูกต้องของคำบรรยายที่ได้จากประสบการณ์อกร่าง. การทดสอบพบว่า ผู้ที่อ้างประสบการณ์อกร่างสามารถบรรยายภาพได้อย่างถูกต้อง. การบรรยายภาพในรูปได้ถูกต้อง บอกได้ว่า (1) การรับรู้สามารถแยกออกจากร่างกายได้ และ (2) ประสบการณ์ที่บรรยายเป็นประสบการณ์จริง ไม่ใช่ความฝัน จินตนาการ หรือผลของกิจกรรมที่สมองสร้างขึ้นมาเอง.

นอกจาก งานของพาร์เนียและคณะแล้ว ยังมีการศึกษาอีก 1 วิจัย[202, 199] ที่สนับสนุนสมมติฐานว่า (1) จิตไม่ได้เกิดจากสมอง และ (2) การรับรู้ของจิตสามารถแยกออกจากสมองได้. แม้ว่าจะมีหลักฐานสนับสนุนหนักแน่น แต่ว่าการจิตวิทยาและประสาทวิทยาส่วนใหญ่ ก็ยังเชื่อในแนวคิดเดิมอยู่ ส่วนหนึ่งก็ เพราะว่า แม้หลักฐานจะบอกว่า จิตไม่ได้เกิดจากสมอง แต่ธรรมชาติของจิต การแยกออกจากสมอง ความสัมพันธ์กับสมอง ความสัมพันธ์กับชีวิต ชีวิตหลังความตาย กลไกที่อยู่เบื้องหลัง เนื่องไขของประสบการณ์เฉียดตาย เรื่องเหล่านี้ วงการวิทยาศาสตร์ยังไม่รู้อะไรเลย. ปัจจุบันวงการวิชาการรู้เรื่องจิตน้อยมาก และหลาย ๆ อายุที่คิดว่ารู้ ก็อาจจะไม่ถูกต้อง.

“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.”

---Al-Kindi

“เราไม่ควรอายที่จะยอมรับความจริง ไม่ว่าเราจะได้รับมันมาจากไหน ถึงแม้ว่ามันจะมาจากคนรุ่นก่อนหรือมาจากคนต่างชาติ. สำหรับผู้แสวงหาความจริง ไม่มีอะไรมีค่ามากกว่าความจริง.”

—อัลคินดี

การกลับชาติมาเกิด ในขณะที่ เราอาจจะเข้าใจความสัมพันธ์ของจิตและชีวิต สิ่งหนึ่งที่น่าสนใจ และอาจจะช่วยเติมภาพความสัมพันธ์นี้ให้เข้มขึ้น คือ การศึกษาเรื่องการกลับชาติมาเกิด (reincarnation). ภาควิชาการศึกษาการรับรู้ มหาวิทยาลัยเวอร์จิเนีย (Division of Perceptual Studies, University of Virginia) ดำเนินการศึกษาเรื่องการกลับชาติมาเกิดมากกว่า 50 ปี ซึ่งทัคเกอร์และคณะ[199] ได้สรุปสาระสำคัญของผลจากการศึกษาว่า เด็กที่ร่ำลึกชาติได้ มีมากกว่า 2,500 คนทั่วโลก เป็นเด็กอายุน้อยมาก ๆ (ไม่เกิน 6 ขวบ) พุดถึงชาติที่แล้ว ซึ่งเป็นชีวิตของคนธรรมดากล่าว 70% จะพูดถึงชาติที่แล้ว ที่ไม่ได้ตายตามธรรมชาติ เช่น ถูกฆ่าตาย หลาย ๆ คน มีอารมณ์หรือพฤติกรรม ที่สัมพันธ์กับคนในชาติที่แล้วที่อ้างถึง เด็กบางคนมีปaineหรือตำแหน่งตั้งแต่เกิด ที่เข้ากับผลของการศึกษาที่แล้วที่อ้าง เช่น มีเด็กอินเดียคนหนึ่งร่ำลึกได้ว่า ชาติที่แล้ว เขาเกิดอุบัติเหตุ เครื่องจักรตัดนิ่วเมื่อขาเข้าอกไป ตัวเด็กเองเกิดมา โดยไม่มีนิ่วเมื่อขา แต่มีข้อจำกัด

เรื่องการกลับชาติมาเกิดไม่ได้เกี่ยวกับเชื้อชาติ หรือความเชื่อ เช่น กรณีของหนุน้อยเจมส์ ไลนินเกอร์ (James Leininger) ที่เป็นลูกชายของครอบครัวชาวคริสต์ที่หลุยส์เซย์น่า สหรัฐอเมริกา เดิมครอบครัวไม่ได้เชื่อเรื่องการกลับชาติมาเกิดเลย. แต่ช่วงเวลา ๆ เจมส์อายุได้สองขวบ เจมส์ก็เริ่มมีฝันร้ายบ่อย ๆ เจมส์ดีนทุรุทุรายและร้อง “ไฟไหม้เครื่องบิน หนูน้อยออกໄไปเมได้” (“Airplane catches on fire. Little man can't get out.”) เวลากลางวัน เจมส์อาเครื่องบินมาเล่น แล้วก็เล่นทำเครื่องบินตก ทำแบบนั้นซ้ำ ๆ พ่อคุยกับเจมส์ เจมส์เล่าว่า เครื่องเข้าถูกยิงตกโดยพวกรถยนต์ปุ่น เจมส์ว่าเข้าขับเครื่องครัวแชร์ (Corsair). ตอนอายุ 28 เดือน เจมส์บอกว่าเข้าบินออกจากเรือ พ่อฟ้องถึงเรือ เจมส์บอกว่าเรือซื้อน้ำใหญ่. ซึ่งช่วงสงครามโลกครั้งที่สอง ก็มีเรือรบญี่อสโซส นาโนมา เบร์ (USS *Natoma Bay*) ที่ประจำการอยู่ในแปซิฟิก พอดีรูป เจมส์ก็วาดแต่รูปเครื่องบินตก วาดเป็นสิบ ๆ รูป จนพ่อของเจมส์เริ่มคิดว่า หรือว่าเจมส์ร่ำลึกชาติได้จริง ๆ

ตอนเจมส์อายุ 4 ขวบครึ่ง พ่อของเจมส์ไปร่วมงานสังสรรค์ท่าเรือเกี้ยวนของ ญี่อสโซส นาโนมา เบร์ ถึงได้รู้ว่า มีนักบินคนเดียวในปฏิบัติการที่ถูกฆ่าตาย นักบินคนนั้นคือ เจมส์ ฮัสตัน (James Huston). เมื่อคณานักวิจัยเบรียบเทียบ สิ่งที่หนูน้อยเจมส์พูดกับประวัติของญี่สตันก็พบว่า

เจมส์ จูสตัน	เจมส์ โจนเนียร์
<ul style="list-style-type: none"> เป็นนักบินของ ยูเออสแอส นาโนม่า เบอร์ ๔๖๗ เคยบินเครื่องครัวเรซซ์ ถูกยิงตกโดยทหารญี่ปุ่น เป็นนักบินคนเดียวของ ยูเออสแอส นาโนม่า เบอร์ ที่ถูกยิงตกตายในปฏิบัติการอิโรชิม่า พยานที่เห็นเหตุการณ์รายงานว่า “ถูกยิงส่วนหน้าตระกลางเครื่อง” เครื่องตกน้ำ และจมลงอย่างรวดเร็ว แจ็ค ลาร์เซ่น เป็นนักบินเครื่องที่อยู่ใกล้กับเครื่องของ จูสตัน วันที่จูสตันเครื่องตกตาย 	<ul style="list-style-type: none"> เขียนต่อในรูปว่า เจมส์ที่สาม (James 3) บอกว่าบินออกจากนาโนม่า บอกว่าบินเครื่องคอร์แซร์ บอกว่าถูกยิงตกโดยทหารญี่ปุ่น บอกว่าตายที่อิโรชิม่า บอก “เครื่องบินผมถูกยิงที่เครื่อง ตกลงน้ำ นั่นแหละ ที่ผมตาย” ผ่านรายถึงเครื่องบินตกและจมน้ำบ่อย ๆ บอกว่าเพื่อนผม แจ็ค ลาร์เซ่น (Jack Larsen) อยู่ที่นั่นด้วย

ในปี พ.ศ. 2560 หนุ่มนายเจมส์ ไลนินเกอร์ อายุ 18 ปี เรียนจบมัธยมและได้เข้าทำงานกับกองทัพเรือสหรัฐอเมริกา.

“Your assumptions are your windows on the world. Scrub them off every once in a while, or the light won’t come in.”

“พิธีเป็นเหมือนหน้าต่างที่มองโลกของคุณ ขัดมันออกบ้าง ไม่อย่างนั้นแสงมันจะไม่สองฟ้ามา.”

---Isaac Asimov

—ໄວແຈ່ຄ ອາຈີມອົບ

ประสบการณ์เฉียดตาย นอกจากการกลับชาติมาเกิด ทัคเกอร์และคณะ[199] ยังได้สรุปงานศึกษาประสบการณ์เฉียดตาย ที่ดำเนินการร่วม 40 ปี ของภาควิชาการศึกษาการรับรู้ ไว้ว่า ประสบการณ์เฉียดตาย พบร้อยละ 20% ในผู้ป่วยหัวใจวาย บรูซ เกรย์สัน (Bruce Grayson) หนึ่งในคณะได้เสนอแบบจำลอง ที่ใช้วัดความเห็นขั้นของประสบการณ์เฉียดตาย จากสีส่วนประกอบ ได้แก่ (1) การเปลี่ยนกระบวนการความคิด, (2) การเปลี่ยนสถานะของอารมณ์ความรู้สึก, (3) ลักษณะเชิงปฏิหารย์, และ (4) ลักษณะเชิงโลกลอื่น.

การเปลี่ยนกระบวนการความคิด เช่น ความรู้สึกถึงการปราศจากเวลา (sense of timelessness), ความคิดที่รอดเร็ว และชัดเจนกว่าปกติ การทบทวนชีวิต (life review) ที่ผู้ป่วยรายงานว่า เห็นชีวิตที่ผ่านมาทั้งหมดฉายผ่านตา เมื่อนอนกับ เป็นสรุปของชีวิต ความรู้สึกว่าเข้าใจ รู้สึกลึกซึ้ง ฯ ชัดเจนแจ่มแจ้ง การเปลี่ยนแปลงกระบวนการความรู้สึก เช่น ความรู้สึก ถึงความสงบ ความพอใจ ความรู้สึกดี (sense of peace and well-being), รู้สึกมีความสุข (sense of joy), รู้สึกเป็นหนึ่งเดียว (sense of oneness or cosmic unity), รู้สึกถึงความรักและความอบอุ่น ลักษณะของปาฏิหารย์ เช่น การมีชีวิต ชีวाओของสัมผัสด่างๆ ที่ผู้ป่วยรายงานว่า เห็นสีสันต่าง ๆ ที่ไม่เคยเห็นในโลกมาก่อน ได้ยินเสียงที่ไม่เคยได้ยินมาก่อน การรับ รู้ถึงเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ระหว่างที่ผู้ป่วยหัวใจวาย การรู้เห็นถึงอนาคต การรู้สึกว่าได้ออกจากร่าง ลักษณะของการ สัมผัสโลกอื่น เช่น การได้เข้าไปในโลกอื่น การได้พบกับสิ่งมีชีวิตที่ลีกลับ การได้พบกับวิญญาณของคนที่ตายไปแล้ว การได้ พบรากับจิตวิญญาณเชิงศาสนา หรือว่า การได้ไปถึงจุดที่กลับไม่ได้ (a point of no return) ที่หากข้ามไปแล้ว จะกลับไม่ได้.

บราซ เกรสัน บอกว่า ประสบการณ์เฉียดตายส่วนใหญ่จะมีลักษณะดังกล่าวผสม ๆ กัน โดยสัดส่วนแตกต่างกันไปตามแต่ละคน โดยได้ยกตัวอย่างประสบการณ์เฉียดตายของผู้หญิงคนหนึ่ง ที่เล่าไว้

“ในช่วงสงกราม ฉันนอนป่วยอยู่ในโรงพยาบาล. เข้าวันหนึ่ง นางพยาบาลเข้ามา และพบร่างฉันไม่มีสัญญาณชีพใด ๆ เลย. นางพยาบาลตามหอมมา ซึ่งหมอก็พบร่างฉันตายแล้วเช่นกัน และฉันก็ตายอยู่อย่างนั้นราوا ๆ 20 นาที ตามที่หมอบอก ฉันในภายหลัง.

ฉันรับรู้แสงสว่างแพร่พราว ที่ฉันรู้สึกถูกเย้ายวนตามมันไป. ตอนนั้นเหมือนกับว่า เวลามันแตกต่างออกไป เมื่อมันไม่มีเวลาอยู่ที่นั่น ไม่ว่าที่นั่น มันจะคือที่ไหน. แสงนั้นสวยงามมาก และมันก็ให้ความรู้สึกของความรักที่ปราศจากเงื่อนไข (*unconditional love*) และความสงบสุข. เมื่อมองไปรอบๆ ฉันก็พบว่า ฉันอยู่ในที่ที่สวยงาม เธียว เป็นเงินชั้นลง. แล้วฉัน ก็เห็นนายทหารนุ่มนุ่มกับทหารอีกหลายนายเดินเข้ามา. นายทหารนุ่มนุ่มเป็น อัลบิน ญาติคนโปรดของฉัน. ตอนนั้น ฉันไม่รู้ ว่าอัลบินตายแล้ว และฉันก็ไม่เคยเห็นอัลบินในชุดเครื่องแบบมาก่อนด้วย. แต่ร่างที่ฉันเห็น ก็ยืนยันได้จากภาพถ่ายที่ฉัน ได้เห็นหลายปีหลังจากนั้น. ฉันคุยกับอัลบินอย่างมีความสุขอยู่สักพัก แล้วอัลบินกับเพื่อนทหารก็เดินแผลวอกไป. แล้วคน ข้าง ๆ ฉันก็อธิบายว่า ทหารเหล่านี้ได้รับอนุญาตให้ไปทักทายคนอื่น ๆ ที่เพียงตาย และช่วยแนะนำเขากับความตาย. ความ ทรงจำที่มีชีวิตชีวาต่อมา ก็คือการมองจากความสูงประมาณเพดานลงไปที่เตียง บนเตียง มีร่างซูบผอมนอนอยู่. มีหม้อและ พยานาลัยร่อง ๆ เตียง. ฉันตะโกนเรียก แต่ไม่มีใครได้ยินฉัน. ฉันเห็นทุกอย่างอย่างชัดเจน และรู้สึกอบอุ่น ปลอดภัย และ สุขสงบ.

อีดใจต่อมา ฉันมองขึ้นไปเห็นหมอกับพยานาลัยล่ามัน แลกรู้สึกผิดหวังอย่างแรง. ฉันฟังอุกมาจากสิ่งที่น่าเบิกบาน ใจ นำพาใจอย่างที่สุด. สองวันหลังจากนั้น หมอก็หายไป ฉันโชคดีที่ยังไม่ตาย. ฉันตอบหม้อไปว่า ฉันตายไปแล้ว. หมอมองฉันแบบแปลกๆ แล้วก็นัดให้ฉันไปประเมินสภาพจิต. และฉันก็ได้เรียนรู้ว่าจะหุบปากเรื่องนี้ ตั้งแต่นั้นเป็นต้นมา.”

เกรสันอภิปรายว่า อิทธิพลของความเชื่อและวัฒนธรรมไม่ได้มีผลต่อประสบการณ์เฉียดตาย แต่ความเชื่อและวัฒน- ธรรมมีอิทธิพลต่อการตีความของประสบการณ์เฉียดตาย เช่น ผู้ผ่านประสบการณ์เฉียดตายในโลกที่สามจะบรรยายถึง ถ้า หรือบ่อน้ำ แทนอุโมงค์ ที่ผู้ผ่านประสบการณ์เฉียดตายในอเมริการะยะ.

คณะผู้วิจัยได้ทำการศึกษาและยืนยันถึงความน่าเชื่อถือของความทรงจำ ในประสบการณ์เฉียดตายที่คงเส้นคงวา แม้ว่าจะเปรียบเทียบการให้สัมภาษณ์ถึงประสบการณ์เฉียดตาย ที่คณะผู้วิจัยกลับไปสัมภาษณ์หลังการสัมภาษณ์เดิมที่เวลา ต่างกันร่วม 10 ถึง 20 ปี. และ เพื่อตอบคำถามว่าความทรงจำในประสบการณ์เฉียดตาย เป็นความทรงจำของเหตุการณ์ จริงๆ ไม่ใช่แค่ความทรงจำของจินตนาการ หรือจากภาพหลอน คณะผู้วิจัยใช้แบบสอบถามลักษณะพิเศษของความทรงจำ (*memory characteristics questionnaire*[96]) ที่ออกแบบมาเพื่อจำแนกแยก ความทรงจำของเหตุการณ์จริง ออกจาก ความทรงจำของเหตุการณ์ในจินตนาการ.

แบบสอบถามลักษณะพิเศษของความทรงจำ ทดสอบความทรงจำใน 5 แร่มุน ซึ่งสามารถแยกความทรงจำของเหตุ- การณ์จริง ออกจากเหตุการณ์สมมติ หรือเหตุการณ์ในจินตนาการได้อย่างน่าเชื่อถือ. แร่มุนต่างๆ ได้แก่ แร่ความชัดเจนของ ความทรงจำ (clarity of memories) ซึ่งรวมถึง รายละเอียดของสิ่งที่เห็น, แร่การรับสัมผัส (sensory aspects) เช่น เสียง กลิ่น รส, แร่ของบริบท (contextual features) เช่น ความทรงจำเกี่ยวกับตำแหน่ง และ การจัดเรียงเชิงพื้นที่ (spatial arrangements), แร่ความคิดและความรู้สึก (thoughts and feelings) ระหว่างที่ระลึกถึงเหตุการณ์, และแร่ความเข้มข้น ของความรู้สึก (intensity of feeling) ระหว่างเหตุการณ์และขณะรำลึกถึง.

คณะผู้วิจัยประเมินผู้ผ่านประสบการณ์เฉียดตาย สำหรับเหตุการณ์ประสบการณ์เฉียดตาย เหตุการณ์จริงอื่นที่เกิดขึ้น ในชีวิตในเวลาใกล้เคียงกัน และเหตุการณ์ในจินตนาการที่เกิดขึ้นในช่วงเวลาอื่น. การทดสอบพบว่า ผู้ผ่านประสบการณ์ เฉียดตายจำประสบการณ์เฉียดตาย ได้ชัดเจน ละเอียด มีบริบท และด้วยความเข้มข้นของความรู้สึก ที่มากกว่า เหตุการณ์ จริงอื่นที่เกิดในช่วงเวลาใกล้เคียงกัน. ประสบการณ์เฉียดตายถูกระบุถึงว่า จริงกว่าเหตุการณ์จริง ในระดับขั้นเดียวกับ ที่เหตุการณ์จริง จริงกว่าเหตุการณ์ในจินตนาการ. ในขณะที่ผู้ที่ไม่ได้มีประสบการณ์เฉียดตาย แต่ผ่านเหตุการณ์ภัยพิบัติใน ลักษณะคล้ายกัน จะรายงานความทรงจำช่วงเหตุการณ์ชีวิตนั้น ว่าจริงในระดับขั้นเดียวกันกับเหตุการณ์จริงอื่น ๆ เท่านั้น ไม่ได้พบว่าจริงมากกว่า.

เกรสันและคณะยังไม่พบปัจจัยใดที่จะสามารถทำนายถึงผู้ที่จะมีประสบการณ์เฉียดตายได้ ไม่ว่าจะเป็นปัจจัย อายุ เชื้อชาติ เพศ ศาสนา ความเครื่องศาสนา หรืออาการป่วยทางจิต และได้ให้ข้อสังเกตว่า แม้จะมีแนวคิดที่พยายามเชื่อมโยง สภาวะทางสุริวิทยา ทางกายภาพ ทางชีวภาพ เข้ากับประสบการณ์เฉียดตาย แต่ที่สุดแล้ว มันก็ยังที่จะอธิบายถึง ความสามารถ

ของสมองที่เพิ่มขึ้น การคิดและรับรู้ได้ชัดเจนขึ้น ในขณะที่สมองไม่สมบูรณ์ ไม่ว่าจากยาสลบ หรือจากภาวะหัวใจวาย.

แม้ว่าประสบการณ์เฉียดตาย อาจเป็นเงื่อนจําของการแยกกันระหว่างจิตกับสมอง หรืออาจเป็นหลักฐานสำคัญของชีวิตหลังความตาย แต่สิ่งที่น่าสนใจที่สุดเกี่ยวกับประสบการณ์เฉียดตาย ก็คือ ผลจากการผ่านประสบการณ์เฉียดตาย. ผู้ที่ผ่านประสบการณ์เฉียดตายจะมีการเปลี่ยนแปลงในเชิงความเชื่อ ทัศนคติ ค่านิยม ได้แก่ มีความเชื่อและครรภาระในเรื่องของจิตวิญญาณมากขึ้น (increase in spirituality), มีความเป็นห่วงเป็นใยมีเมตตาต่อผู้อื่นมากขึ้น (increase in sense of concern/compassion for others), ตระหนักรู้ในค่าของชีวิตมากขึ้น (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), มีความสนใจในทรัพย์สินลดลง (decreased interest in material possession), ลดความสนใจในสถานะอำนาจ เกียรติ และชื่อเสียงลง (decreased interest in status, power, prestige, and fame), ลดความสนใจในการแก่งแย่งชิงเด่นลง (decreased interest in competition).

ถึงตรงนี้ อาจทำให้เกิดคำถามว่า ถ้าผู้ผ่านประสบการณ์เฉียดตายไม่กลัวตาย และพบว่าความตายนั้นเป็นสุขและสวยงาม ทำไมเขามิ่งฆ่าตัวตายไปเลย แต่กลับรักและชื่นชมคุณค่าของชีวิต และใช้ชีวิตอย่างมีความหมาย. คำถามนี้ ทัคเกอร์ เกรร์สัน และคณไม่ได้อภิปรายไว้. แต่หากลองใคร่ครวญพิจารณาด้วยตนเองแล้ว จะพบว่าชีวิตคนนั้นแปลก. คนที่เห็นความตาย กลับไม่กลัวตาย. คนที่กลัวตาย ไม่เคยเห็นความตาย. คนไม่กลัวความตาย กลับเข้าใจชีวิต ใช้ชีวิตได้ ใช้ชีวิตอย่างมีคุณค่า. แต่คนที่ว่าไปที่ส่วนใหญ่กลัวความตาย หลายคนเลือกใช้ชีวิตทั้งเปล่าไป. หลายคนเลือกเล่นโทรศัพท์มือถือ แทนการมีปฏิสัมพันธ์กับคนรอบข้าง. หลายคนเลือกใช้ชีวิตเห็นแก่ตัว เลือกเป้าหมายชีวิตเป็นความร่าเรว สถานะ ชื่อเสียง ตอบสนองต่ออัตตาที่ขยายไม่รู้จบ แล้วเริ่มนั่นว่า ความสำเร็จ. หลายคนทึ่งคุณค่าของชีวิต ทึ่งความสงบสุข เพื่อใช้ชีวิตที่ฟุ่มเฟือ. หลายคนทึ่งความกล้าที่จะทำในสิ่งที่ถูกต้อง ทึ่งความเชื่อสัตย์มั่นคงที่จะยืนหยัดในหลักการที่อ้าง ทึ่งเมตตา ทึ่งปัญญา อันเป็นคุณธรรมอันสูงสุด ทั้งโดยรู้ตัวและไม่รู้ตัว. หลายคนกลัวความตาย แต่กลับไม่เคยใช้ชีวิตให้มีคุณค่าเลย. ประเด็นนี้ก็เป็นอีกปริศนาของชีวิตที่ยากจะอธิบาย และข้อสังเกตนี้ได้ถูกกล่าวไว้อย่างดรงโดยท่านดาไลลามาองค์ที่สิบสี่ แห่งริบูดังนี้.

“[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 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

“สิ่งที่ทำให้อาตามาแปลกใจที่สุดคือ] คน เพราะว่า คนสละสุขภาพไปเพื่อหาเงิน แล้วทีหลังก็สละเงินไปเพื่อฟื้นฟูสุขภาพ และคนก็มัวแต่กังวลกับอนาคต จนไม่มีความสุขทั้งปัจจุบัน หลังคือเขามิ่งได้อยู่ในปัจจุบันไม่ได้อยู่ในอนาคต เขาใช้ชีวิตอยู่เหมือนกับว่าเขาจะไม่มีวันตาย แล้วก็ตายไปแบบบ้ม่ำโดยไม่เคยมีชีวิตจริง ๆ.”

—เทนzin กิยันโซ่, ดาไลลามาที่สิบสี่

7.3 อภิธานศัพท์

การตรวจจับวัตถุในภาพ (object detection): ภาระกิจการระบุชนิดของวัตถุและตำแหน่งในภาพ
โอลี (YOLO): แบบจำลองที่สำคัญในการตรวจจับวัตถุในภาพ ซึ่งมีแนวคิดที่สำคัญคือการครอบปัญหา เป็นงานการหาค่าตัดตอน และช่วยลดขั้นตอนการทำงานที่ซับซ้อนลงได้ ส่งผลให้แบบจำลอง

สามารถทำงานได้รวดเร็ว และการแก้ไขปรับปรุงก็ทำได้สะดวก

กล่องสมอ (anchor box): เทคนิคที่ยอมให้มีการทายวัตถุในภาพที่มีตำแหน่งซ้อนทับกันได้ โดยใช้กลไกของรูปร่างและขนาดเริ่มต้นที่ต่างกันของกล่องขอบเขต เพื่อกำหนดความรับผิดชอบต่อวัตถุ คล้ายการปักสมอของแต่ละกล่องขอบเขต ว่ากล่องใดจะรับผิดชอบขนาดหรือรูปทรงคร่าว ๆ แบบใด

โครงข่ายปรับกษ์เชิงสร้าง (Generative Adversarial Networks คำย่อ GANs): กลไกการฝึกโครงข่ายสองโครงข่าย โดยฝึกในลักษณะที่ทั้งสองโครงข่ายมีเป้าหมายขัดแย้งกัน. โครงข่ายหนึ่งเรียกว่า โครงข่ายก่อกำเนิด ทำหน้าที่สร้างจุดข้อมูลขึ้นมา เลียนแบบจุดข้อมูลจริง ในขณะที่อีกโครงข่ายหนึ่ง เรียกว่า โครงข่ายแบ่งแยก ทำหน้าที่ตรวจสอบ ว่าจุดข้อมูลที่เห็นถูกสุมจากชุดข้อมูลจริง หรือถูกสร้างขึ้น. โครงข่ายก่อกำเนิด มีเป้าหมายเป็นการสร้างจุดข้อมูลเลียนแบบให้เหมือนข้อมูลจริง จนโครงข่ายแบ่งแยกจำแนกไม่ออกร. โครงข่ายแบ่งแยก มีเป้าหมายเป็นการจำแนกจุดข้อมูลได้ถูกต้องมากที่สุด

โครงข่ายแบ่งแยก (discriminator): โครงข่ายหนึ่งในกลไกการฝึกแบบปรับกษ์ ทำหน้าที่จำแนกจุดข้อมูลที่เห็นว่า ถูกสุมจากชุดข้อมูลจริง หรือถูกสร้างขึ้น

โครงข่ายก่อกำเนิด (generator): โครงข่ายหนึ่งในกลไกการฝึกแบบปรับกษ์ ทำหน้าที่สร้างจุดข้อมูล เลียนแบบจุดข้อมูลจริงจนโครงข่ายแบ่งแยกจำแนกได้แยกตัวกัน

พีชคณิตเวกเตอร์ (Vector arithmetic): สำหรับโครงข่ายปรับกษ์เชิงสร้าง พีชคณิตเวกเตอร์ อ้างถึงปฏิบัติการเชิงเส้นที่ทำกับเวกเตอร์ลักษณะซ่อนเร้น แล้วนำเวกเตอร์ผลลัพธ์ไปเข้าโครงข่ายก่อกำเนิด เพื่อสร้างจุดข้อมูลขึ้นมา

ลักษณะซ่อนเร้น (latent representation): ลักษณะของข้อมูล ที่ไม่ได้แสดง หรือกำหนดโดยอย่างชัดแจ้ง. ในบริบทของโครงข่ายปรับกษ์เชิงสร้าง หมายถึง ค่าของเวกเตอร์ที่ใช้เป็นอินพุตของโครงข่ายก่อกำเนิด

ปริภูมิซ่อนเร้น (latent space): หรือปริภูมิตัวแทน (representation space) ปริภูมิของลักษณะซ่อนเร้น

การพังทลายของภาวะ (mode collapse): สถานการณ์ที่โครงข่ายก่อกำเนิดสร้างจุดข้อมูลคล้าย ๆ กัน แม้ว่าจะรับอินพุตที่ต่างกัน

convolutional (fractionally-strided convolution): หรืออาจเรียก convolutional (transposed convolution) การดำเนิน convolution ที่มีความยาวไม่เท่ากัน ด้วยการเปลี่ยนแปลงอินพุตและค่าน้ำหนักของฟิลเตอร์เป็นเมทริกซ์ โดยจัดรูปเมทริกซ์ทั้งสองให้ถูกต้อง แล้วทำการคูณเมทริกซ์อินพุตเข้ากับการลับเปลี่ยนของเมทริกซ์ค่าน้ำหนัก. หากเลือกอภิมานพารามิเตอร์ได้ถูกต้อง การดำเนินการเช่นนี้ อาจมองเสมอเป็นการทำ convolution ที่ใช้ขนาดก้าวย่างเล็กกว่าหนึ่ง (เหมือนการเติมพิกเซลค่าศูนย์เข้าไประหว่างพิกเซลของอินพุต ในกรณีข้อมูลภาพ)

การถอด convolution (deconvolution): การถอด convolution มีหลายความหมาย. ในขณะที่บาง

ครั้ง การถอดถอนโวลุชั่นอาจหมายถึงถอนโวลุชั่นก้าวเศษ แต่ความหมายที่ถูกยอมรับอย่างกว้างขวาง คือการถอดค่าพารามิเตอร์ของโครงข่ายคอนโวลูชันย้อนกลับ เพื่อศึกษาลักษณะการทำงานของโครงข่ายคอนโวลูชัน ว่า ฟิลเตอร์แต่ละตัวที่ใช้ในโครงข่ายคอนโวลูชัน ได้เรียนรู้เพื่อจะตรวจจับลักษณะรูปแบบเช่นไร

การจับคู่ลักษณะสำคัญ (feature matching): ในบริบทโครงข่ายประปักษ์เชิงสร้าง การจับคู่ลักษณะสำคัญ หมายถึงการดัดแปลงฟังก์ชันจุดประสงค์ของโครงข่ายก่อกำเนิด โดย แทนที่เป้าหมายการทำให้โครงข่ายแบ่งแยกทรายผิดมากที่สุด ด้วยเป้าหมายการทำให้ความต่างระหว่าง ค่าเฉลี่ยของลักษณะสำคัญ เมื่อเห็นจุดข้อมูลที่สร้างขึ้น มีค่าน้อยที่สุด. ตัวอย่าง เช่น การใช้ฟังก์ชันสูญเสีย $\text{Loss}_G = \|E_{\mathbf{X}}[\mathbf{f}(\mathbf{X})] - E_{\mathbf{z}}[\mathbf{f}(\mathcal{G}(\mathbf{z}))]\|^2$ เมื่อ \mathbf{X} คือข้อมูลจริง และ \mathbf{z} แทนเวกเตอร์ค่าสุ่ม ส่วน $\mathcal{G}(\cdot)$ คือการคำนวณของโครงข่ายก่อกำเนิด และ $\mathbf{f}(\cdot)$ คือลักษณะสำคัญที่ได้จากโครงข่ายแบ่งแยก

การแยกแยะหมู่เล็ก (minibatch discrimination): การเพิ่มสัญญาณข้อมูลที่ช่วยบอกความแตกต่าง ระหว่างจุดข้อมูลใด ๆ กับจุดข้อมูลอื่น ๆ ภายในหมู่เล็กเดียวกัน เพื่อบรรเทาปัญหา การพังทลายของภาวะ

การทำฉลากරบ린 (label smoothing): การปรับค่าเป้าหมายของฉลากเฉลย เพื่อบรรเทาปัญหาที่แบบจำลองมีความมั่นใจสูงเกินไป

การทำหมู่เล็กเสมือนจริง (virtual minibatch): การใช้หมู่อ้างอิงที่เลือกมาก่อนกระบวนการฝึก เพื่อใช้การคำนวณแบบอร์ม 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

“ความ เมตตา ปราณี เพียง ครั้ง ก็
หยั่ง ราก ไป ทุก ทิศทาง และ ราก ก็ จะ
งอก งาม ออ กมา เป็น ต้น ใหม่.”

—เอมีเลีย แอร์ฮาร์ต

แบบฝึกหัด 7.1

จะเลือกการประยุกต์ใช้โครงข่าย convolutional neural network ที่เกี่ยวข้อง โดยให้เลือก บทความวิจัยที่เกี่ยวข้องไม่น้อยกว่า 20 บทความ และสำหรับแต่ละบทความ ให้อภิรายถึง จุดประสงค์ ความคาดหมาย ปัญหาที่ต้องการแก้ ความท้าทายที่เกี่ยวข้อง วิธีการที่นำเสนอ และผลลัพธ์ รวมถึง อภิรายจุดเด่น และประเด็นอื่น ๆ ที่เห็นว่า่าน่าสนใจในบทความ.

นอกจากนี้ จะอภิรายความสัมพันธ์กับการประยุกต์แบบอื่นที่มีลักษณะใกล้เคียงกัน (อาจต้อง ทำการศึกษาวรรณกรรมเพิ่มเติม ให้ฝึกการศึกษาวรรณกรรมอย่างกว้างขวาง).

ตัวอย่าง เช่น หากเลือกการรู้จำใบหน้า (*face recognition*) อาจอภิรายความสัมพันธ์ ความ เหมือน ความต่าง กับการประยุกต์ใช้สำหรับ การจำแนกชนิดวัตถุ ในภาพ (*image classification*) หรือการพิสูจน์ยืนยันใบหน้า (*face verification*) หรือการรู้จำอารมณ์จากใบหน้า (*facial expression recognition*) เป็นต้น.

แบบฝึกหัด 7.2

จะเลือกบทความวิจัยในแบบฝึกหัด 7.1 มา 5 บทความ และสำหรับแต่ละบทความ (นอกจากประเด็นในแบบฝึกหัดที่ 7.1) ให้อภิรายถึง ข้อมูล วิธีการปฏิบัติ การทดลอง และวิธีการประเมินผล.

แบบฝึกหัด 7.3

จากแบบฝึกหัดที่ 7.2 จะศึกษาวิธีการประยุกต์ใช้ และลงมือปฏิบัติ ทดลอง และเปรียบเทียบ ผลที่ได้กับผลที่รายงานในวรรณกรรม. ในการลงมือปฏิบัติ อาจปรับลดความยากของปัญหาลงได้ตาม ความเหมาะสม รวมถึงอาจศึกษาวิธีการปฏิบัติและโปรแกรมจากอินเตอร์เน็ต

ตัวอย่าง หากเลือกการรู้จำใบหน้า และสนใจ FaceNet[179] จะใช้คำค้นหา เช่น “facenet code” และอาจเลือกชุดข้อมูลที่ง่ายขึ้น หรือเลือกข้อมูลขนาดเล็กลง หรือใช้แบบจำลองที่เล็กลง เพื่อ ให้การฝึกทำได้รวดเร็วขึ้น.

แบบฝึกหัด 7.4

จงบททวนเรื่องโครงข่ายประបกษ์เชิงสร้าง (และอาจศึกษาเพิ่มเติม ถ้าจำเป็น) และอภิรายถึงแนวทาง วิธี หรือกลไก เพื่อจะอนุมานการแยกแยะร่วม $p(\mathbf{X}, \mathbf{C})$ เมื่อ \mathbf{X} คือข้อมูลต้น เช่น ภาพ และ \mathbf{C}

คือข้อมูลตาม เช่น ประเภทของวัตถุในภาพ โดยอาศัยแนวทางของโครงข่ายประปักษ์เชิงสร้าง การอภิปราย อาจเริ่มจากข้อคิดเห็นหรือคำถาม เช่น หากโครงข่ายประปักษ์เชิงสร้าง สามารถเรียนรู้การแยกแยะ $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$ ได้แล้ว และในเมื่อ \mathbf{z} ก็สู่สร้างขึ้นมาเอง (อาจจะจากสุ่มจากการแจกแจงเอกรูป หรือการแจกแจงเก้าส์เซียน) ส่วนเงื่อนไขหรือข้อมูลตาม \mathbf{C} ซึ่งมักอยู่ในปริภูมิที่มีจำนวนมิติไม่มาก ก็อาจสามารถประมาณการแยกแยะจากข้อมูลที่มีได้ไม่ยากนัก ดังนั้น จากการแจกแจง $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$ เราก็จะสามารถใช้ทรุษฎีของเบล์ เพื่ออนุมานการแยกแยะร่วม $p(\mathbf{X}, \mathbf{C})$ ได้ ทำให้การหาอนุมานการแยกแยะร่วม $p(\mathbf{X}, \mathbf{C})$ หรือแม้แต่การหา $p(\mathbf{X})$ เมื่อ \mathbf{X} เป็นภาพ เช่น ภาพถ่ายทิวทัศน์ทั่วไป ถึงเป็นปัญหาที่ยากมาก⁸ หากเป็นไปได้?

ตั้งกลุ่ม สามคำถามและอภิปรายข้อคิดเห็นลักษณะเช่นนี้ ความท้าทาย ความเสี่ยง แนวทางและกลไกที่จะลดหรือบรรเทาปัญหาและความเสี่ยงต่าง ๆ ยกตัวอย่าง หรือหากเหมาะสม อาจจะลองออกแบบการทดลองเล็ก ๆ ง่าย ๆ เพื่อพิสูจน์ ยืนยัน หรือหักล้าง.

แบบฝึกหัด 7.5

พิจารณาข้ออภิปรายถึงวิธีการทำฉลาก rab/rain (label smoothing) ดังนี้ วิธีการทำฉลาก rab/rain[196] ปรับค่าเป้าหมายของฉลากเป็น $q_k = (1 - \varepsilon)y_k + \varepsilon p(k)$ เมื่อ y_k คือค่าฉลากเฉลยในรูปหนึ่งร้อนของประเภท k^{th} และ $p(k)$ คือการแจกแจงของข้อมูลชนิด k^{th} . สังเกตว่า วิธีการทำฉลาก rab/rain ปรับที่ค่าเป้าหมายของฉลากเฉลย ไม่ได้แก้ไขการคำนวนฟังก์ชันกระตุ้นในแบบจำลอง. หากพิจารณาประเดิมนี้ร่วมกับฟังก์ชันสูญเสียสำหรับภาระกิจการจำแนกประเภทแบบหลายกลุ่ม ซึ่งคือ $\text{Loss} = -\sum_k y_k \log \hat{y}_k$ เมื่อ y_k คือค่าเป้าหมายเฉลย และ \hat{y}_k คือค่าท่านาย จะพบว่า กรณีไม่ทำฉลาก rab/rain (กรณีดั้งเดิม) ฟังก์ชันสูญเสียสามารถคำนวนโดย $\text{Loss} = -\log \hat{y}_{k^*}$ เมื่อ k^* คือฉลากของประเภทที่เฉลย เพราะ $y_{k^*} = 1$ และ $y_{k \neq k^*} = 0$.

แต่กรณีการทำฉลาก rab/rain ฟังก์ชันสูญเสียไม่สามารถย่อรูปดังข้างต้นได้ และหากคำนวน $\text{Loss} = -\sum_k y_k \log \hat{y}_k$ 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 แล้วอาจเกิดปัญหาการคำนวนเชิงเลขได้. ดังเช่น กรณีที่ \hat{y}_k ตัวใดตัวหนึ่งมีค่าใกล้กับศูนย์มาก ๆ ($\log 0 \rightarrow -\infty$) ซึ่งอาจจะทำให้การคำนวนไม่มีเสถียรภาพ. ปัญหานี้ แม้จะเกิดยากเนื่องจากแบบจำลองมีแนวโน้มที่จะถูกฝึกให้ \hat{y}_k ปรับเข้าหาเป้าหมาย เช่น $\hat{y}_k \rightarrow \frac{\varepsilon}{K}$ แล้วค่า ε ไม่เล็กจนเกินไป. แต่การปรับเปลี่ยนนี้ ก็เพิ่มความเสี่ยงขึ้นมากจากการณีดั้งเดิม (ที่มีแต่ $\log \hat{y}_{k^*}$ ซึ่ง $\hat{y}_{k^*} \rightarrow (1 - \varepsilon + \frac{\varepsilon}{K})$ ที่มีค่ามากใกล้ ๆ หนึ่ง).

นอกจากนี้ อีกประเดิมนี้สำหรับการใช้วิธีการทำฉลาก rab/rain ในทางปฏิบัติ หากการทำฉลาก rab/rain ถูกนำไปเขียนโปรแกรมโดยไม่ระมัดระวัง เช่น อาจอาศัยโปรแกรมหรือโครงสร้างเดิมจากฟังก์ชันสูญเสีย ซึ่งคำนวน $\text{Loss} = -\log \hat{y}_{k^*}$ แทน $\text{Loss} = -\sum_k y_k \log \hat{y}_k$ และ การทำฉลาก rab

⁸ คำใบ้ คำกล่าวว่าแบบจำลองเรียนรู้การแยกแยะ $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$ กับการที่แบบจำลองสามารถให้ค่า $p(\mathbf{X}|\mathbf{C}, \mathbf{z})$ ออกมาได้ นั้น ต่างกัน. สิ่งที่โครงข่ายประปักษ์เชิงสร้างให้ออกมาจริง ๆ คืออะไร? การแจกแจง (?) ความน่าจะเป็น (?) หรือเพียงค่าคาดหมาย $E[\mathbf{X}|\mathbf{C}, \mathbf{z}]$ หรืออะไร?

รีน อาจผิดเพี้ยนจากแนวคิดดังเดิมได้ 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 $\text{Loss} = -(1 - \varepsilon + \frac{\varepsilon}{K}) \log \hat{y}_{k^*}$ ซึ่งผิดเพี้ยนไปจากแนวคิดของการทำclassificationที่เชื่อถือและคงจะ[196]เสนอ (แต่อาจจะทำงานได้ และอาจจะไปคล้ายกับแนวคิดของการทำclassificationทางเดียว)

จึงศึกษาโปรแกรมการทำclassificationที่ค้นหาได้จากอินเตอร์เน็ต ทดลองใช้และสังเกตการทำงานของวิธีการทำclassification วิเคราะห์ และให้ข้อคิดเห็นเพิ่มเติมจากข้ออภิรายข้างต้น (อาจเห็นด้วย เห็นแย้ง หรือเห็นต่าง) พร้อมให้เหตุผล และอาจยกตัวอย่างประกอบ เพื่อสนับสนุน รวมถึงอภิรายสถานการณ์ต่าง ๆ ว่าหากเกิดขึ้นจริง จะมีผลดี ผลเสียอย่างไรบ้าง และสำหรับผลเสียจะมีวิธีจัดการ แก้ หรือบรรเทาปัญหาอย่างไรบ้าง

แบบฝึกหัด 7.6

จากแบบฝึกหัด 7.5 ที่อภิรายกรณีการจำแนกกลุ่ม จงอภิรายประเด็นข้อดี ข้อเสีย โอกาส และ ความเสี่ยง ในทางปฏิบัติ เมื่อนำวิธีการทำclassificationไปใช้ ในกรณีการจำแนกค่าทวิภาค (binary classification) รวมถึงศึกษางานของคณะของเซลลิมนัส[173] สำหรับเหตุผลที่เลือกใช้การทำclassification 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

รูปแบบของข้อมูลในเนื้อหาที่ผ่าน ๆ มา เป็นลักษณะที่เป็นอิสระในตัวเอง นั่นคือ แต่ละจุดข้อมูลมีความหมายสมบูรณ์แบบในตัวเอง หรือหากจะเจาะจงลงไป อาจกล่าวว่า ที่ผ่านมา แต่ละจุดข้อมูล เป็นอิสระต่อกันและมีการแจกแจงเหมือนกัน ที่เรียกว่า ไอ.ไอ.ดี. (*independent and identically distributed* คำย่อ *i.i.d.*). อย่างไรก็ตาม มีข้อมูลหลายประเภทที่จุดข้อมูลต่าง ๆ มีความสัมพันธ์ระหว่างกัน. เนื้อหาในบทนี้อภิปรายแนวทางและแบบจำลอง ที่ออกแบบมาสำหรับการทำงานอย่างมีประสิทธิภาพกับข้อมูลเชิงลำดับ.

“ความล้มเหลวมาก็แต่เฉพาะตอนที่เราลืม อุดมการณ์ เป้าหมาย และหลักการของเรา.”

— Jawaharlal Nehru

8.1 ข้อมูลเชิงลำดับ

ข้อมูลประเภทที่จุดข้อมูลมีความสัมพันธ์เชิงลำดับระหว่างกัน จะเรียกว่า **ข้อมูลเชิงลำดับ** (*sequential data*). นั่นคือ แม้ว่าจุดข้อมูลจะมีค่าเหมือนกัน แต่หากลำดับที่ปรากฏต่างกัน ก็อาจทำให้ความหมายต่างกัน หรือแม้จุดข้อมูลหนึ่งจะมีค่าเท่าเดิมและปรากฏที่ตำแหน่งเดิม แต่หากจุดข้อมูลอื่น ๆ ในลำดับเปลี่ยนค่า ก็อาจจะทำให้ความหมายนั้นเปลี่ยนไปได้.

ข้อมูลหลากหลายชนิด เป็นข้อมูลเชิงลำดับ และงานการรู้จำรูปแบบกับข้อมูลเชิงลำดับ ก็มีหลากหลายประเภท เช่น การทำนายข้อมูลทางการเงิน (financial data prediction) ซึ่งอาจรับอินพุตเป็นลำดับของราคากำลังของหุ้นในวันก่อน ๆ และทำนายราคากำลังของวันถัดไป, การรู้จำเสียงพูด (speech recognition) ที่รับอินพุตเป็นสัญญาณเสียง (ลำดับค่าแอมพลิจูดต่อเวลา) และให้อาต์พุตเป็นลำดับของคำ, ระบบแต่งเพลงอัตโนมัติ (music generation) ที่อาจรับอินพุตเป็นประเภทของเพลง และให้อาต์พุตเป็นลำดับของโน้ตดนตรี, การรู้จำรูปแบบสัญญาณคลื่นไฟฟ้าหัวใจ (ECG pattern recognition) ที่รับอินพุตเป็นสัญญาณคลื่นไฟฟ้าหัวใจ (ลำดับแอมพลิจูดหลาย ๆ ช่องสัญญาณต่อเวลา) และอาจจะ

ให้เอาร์พุตเป็นค่าระบุว่า ปกติหรือผิดปกติ หรืออาจจะบุตคำแห่งและชนิดที่ผิดปกติอกรมาด้วย, ระบบวิเคราะห์ลำดับดีเอ็นเอ (DNA sequence analysis) ที่รับอินพุตเป็นลำดับของชนิดฐานนิวคลีโอไทด์ และอาจจะให้เอาร์พุตเป็นตำแหน่งต่าง ๆ ในลำดับที่สัมพันธ์กับโปรตีนที่สนใจ, การรู้จำกิจกรรมจากวีดีโอ (video activity recognition) ที่รับอินพุตเป็นข้อมูลวีดีโอ (ลำดับของภาพต่าง ๆ ตามเวลา) และให้เอาร์พุตเป็นฉลากของกิจกรรม, การจำแนกอารมณ์ (sentiment classification) ที่อาจรับอินพุตเป็นข้อความ (ลำดับของคำต่าง ๆ) และให้เอาร์พุตเป็นคะแนนประเมินความพ่อใจ, การแปลภาษาอัตโนมัติ (machine translation) ที่รับอินพุตเป็นข้อความในภาษาหนึ่ง (ลำดับของคำในภาษาต้นทาง) และให้เอาร์พุตเป็นข้อความในอีกภาษาหนึ่ง (ลำดับของคำในภาษาเป้าหมาย). สำหรับข้อมูลเชิงลำดับบางชนิด ตัวลำดับอาจเป็นตัวแทนของเวลา เช่น สัญญาณเสียง หรืออาจเป็นเพียงลำดับที่ไม่ได้เกี่ยวกับเวลา ก็ได้ เช่น ลำดับของคำต่าง ๆ ในข้อความ. อย่างไรก็ตาม เพื่อความสะดวก เนื้อหาในบทนี้อาจใช้คำว่า เวลา ใน การอ้างถึงลำดับ. รูปที่ 8.1 แสดงตัวอย่างข้อมูลเชิงลำดับ และตัวอย่างภาระกิจการรู้จำรูปแบบที่เกี่ยวข้อง.

หากจำแนกภารกิจออกตามลักษณะอินพุตและเอาต์พุต การรู้จำรูปแบบเชิงลำดับ อาจจำแนกได้ดังนี้. (1) ประเภทแรกคือ ภารกิจที่อินพุตเป็นข้อมูลลำดับ $\{\mathbf{x}_n\}_{n=1,\dots,N}$ เมื่อ N คือจำนวนจุดข้อมูลทั้งหมดในลำดับ แต่เอาต์พุตไม่ได้เป็นข้อมูลลำดับ $y \in \mathbb{R}^K$ เมื่อ K คือจำนวนมิติของเอาต์พุต. ในรูปที่ 8.1 ตัวอย่างในกลุ่มนี้คือ การทำนายข้อมูลทางการเงิน, การรู้จำรูปแบบสัญญาณคลื่นไฟฟ้าหัวใจ, การรู้จำกิจกรรมจากวีดีโอ และการจำแนกอารมณ์. ระบบวิเคราะห์ลำดับดีเอ็นเอ ก็อาจจัดอยู่ในกลุ่มนี้ หากให้เอาต์พุตออกมานเป็นค่าดัชนีของจุดเริ่มต้นและจุดจบของส่วนลำดับที่สนใจ. (2) ประเภทสองคือ ภารกิจที่อินพุตเป็นข้อมูลลำดับ $\{\mathbf{x}_n\}_{n=1,\dots,N}$ และเอาต์พุตก็เป็นข้อมูลลำดับ $\{\mathbf{y}_n\}_{n=1,\dots,N}$ โดยทั้งสองลำดับมีจำนวนจุดข้อมูลในลำดับเท่ากัน. รูปที่ 8.1 ไม่ได้แสดงตัวอย่างของภารกิจในกลุ่มนี้. ตัวอย่างภารกิจในกลุ่มนี้ ได้แก่ การระบุหมวดคำ (Part-Of-Speech Tagging) ที่วิเคราะห์ข้อความแล้วระบุหมวดคำของคำทุกคำในข้อความ ว่าอยู่ในหมวดคำใดในกลุ่ม (ซึ่งมักประกอบด้วย คำนาม, คำกริยา, คำคุณศัพท์, คำกริยาवิเศษณ์, คำบุพบท, คำสัน്ധان, คำสรรพนาม, คำอุทาน และคำนำหน้านาม สำหรับภารกิจการระบุหมวดคำในภาษาอังกฤษ) และการรู้จำชื่อเฉพาะ (Named-Entity Recognition) ที่ต้องการระบุว่าคำไหนบ้างในข้อความที่เป็นชื่อเฉพาะ และเป็นชื่อเฉพาะของสิ่งประเภทใด (ซึ่งมักกำหนดประเภทที่สนใจไว้ เช่น ชื่อคน, ชื่องค์กร หน่วยงาน หรือบริษัท, ชื่อตราสินค้า, ชื่อสถานที่, เวลา, ปริมาณหรือจำนวน) เป็นต้น. (3) ประเภทสามคือ ภารกิจที่อินพุตเป็นข้อมูลลำดับ $\{\mathbf{x}_n\}_{n=1,\dots,N}$ และเอาต์พุตก็เป็นข้อมูลลำดับ $\{\mathbf{y}_n\}_{n=1,\dots,M}$ เมื่อ M เป็นจำนวนจุดข้อมูลในลำดับของเอาต์พุต แต่ลำดับทั้งสองอาจมีจำนวนจุดข้อมูลในลำดับไม่เท่ากัน. ตัวอย่างที่แสดงในรูปที่ 8.1 คือ การแปลภาษาอัตโนมัติ ที่ข้อความของภาษาต้นทาง อาจมีจำนวนคำต่างจากข้อความของภาษาปลายทาง. (4) ประเภทสี่คือ ภารกิจที่อินพุตไม่ใช่ข้อมูลลำดับ แต่เอาต์พุตเป็นข้อมูลลำดับ. ตัวอย่างที่แสดงในรูปที่ 8.1 คือ ระบบแต่งเพลงอัตโนมัติ ที่อินพุตอาจจะเป็นເວກເຕ່ອງ หรืออาจจะเป็นค่าสเกลาร์ แต่ให้เอาต์พุตออกมานเป็นลำดับของโน๊ตดนตรี. สุดท้าย หากภารกิจที่ทั้งอินพุตและเอาต์พุตไม่ใช่ข้อมูลลำดับ

ดับ ภาระกิจนี้ไม่จัดอยู่ในภาระรูปแบบเชิงลำดับ และโดยทั่วไป ภาระกิจประเภทนี้จะสามารถดำเนินการได้โดยวิธีการต่าง ๆ ที่ได้อภิรายไปในบทก่อน ๆ.

ข้อมูลเชิงลำดับ อาจแบ่งออกได้เป็นสองประเภท คือ ข้อมูลเชิงลำดับแบบคงที่ กับ ข้อมูลเชิงลำดับแบบไม่คงที่. ข้อมูลเชิงลำดับแบบคงที่ (stationary sequential data) ที่แม่ค่าต่าง ๆ ของจุดข้อมูล อาจผันเปลี่ยนไปตามเวลา แต่การแจกแจงเบื้องหลังลำดับข้อมูลนั้นคงที่ ไม่ได้มีการเปลี่ยนแปลงตามเวลา. ส่วนกรณีของข้อมูลเชิงลำดับแบบไม่คงที่ (nonstationary sequential data) การแจกแจงเบื้องหลัง ลำดับข้อมูลนั้นมีการเปลี่ยนแปลงตามเวลาด้วย. หากอธิบายง่าย ๆ อาจเปรียบเทียบจาก ตัวอย่างของ ข้อมูลปริมาณน้ำฝนในแต่ละวัน ตลอดหลาย ๆ ปี โดยที่ระยะเวลาเหล่านั้นฟ้าฝนตกต่อเนื่องตามฤดูกาล เป็นตัวอย่างของข้อมูลเชิงลำดับแบบคงที่. ส่วนตัวอย่างของข้อมูลปริมาณน้ำฝนในแต่ละวัน ในหลาย ๆ ปีให้หลังมาเนี้ย ซึ่งสังเกตได้ชัดว่าฟ้าฝนตกผิดเพี้ยนจากฤดูกาลที่คุ้นเคย. การเปลี่ยนแปลงเกิดที่ตัว ของฤดูกาลเองอย่างมาก ที่อาจจะเกิดจากหลาย ๆ สาเหตุรวมถึงภาวะโลกร้อนสภาพภูมิอากาศเปลี่ยน- แปลง. กรณีหลังนี้ เป็นตัวอย่างของข้อมูลเชิงลำดับแบบไม่คงที่ ซึ่งการเปลี่ยนแปลงตามเวลาของข้อมูล เกิดจากการบวนการเบื้องหลังที่มีการเปลี่ยนแปลงตัวกระบวนการไปตามเวลาด้วย. เนื้อหาในบทนี้ใช้ สมมติฐานของกรณีข้อมูลเชิงลำดับแบบคงที่เป็นหลัก ยกเว้นแต่จะระบุเป็นอื่น.

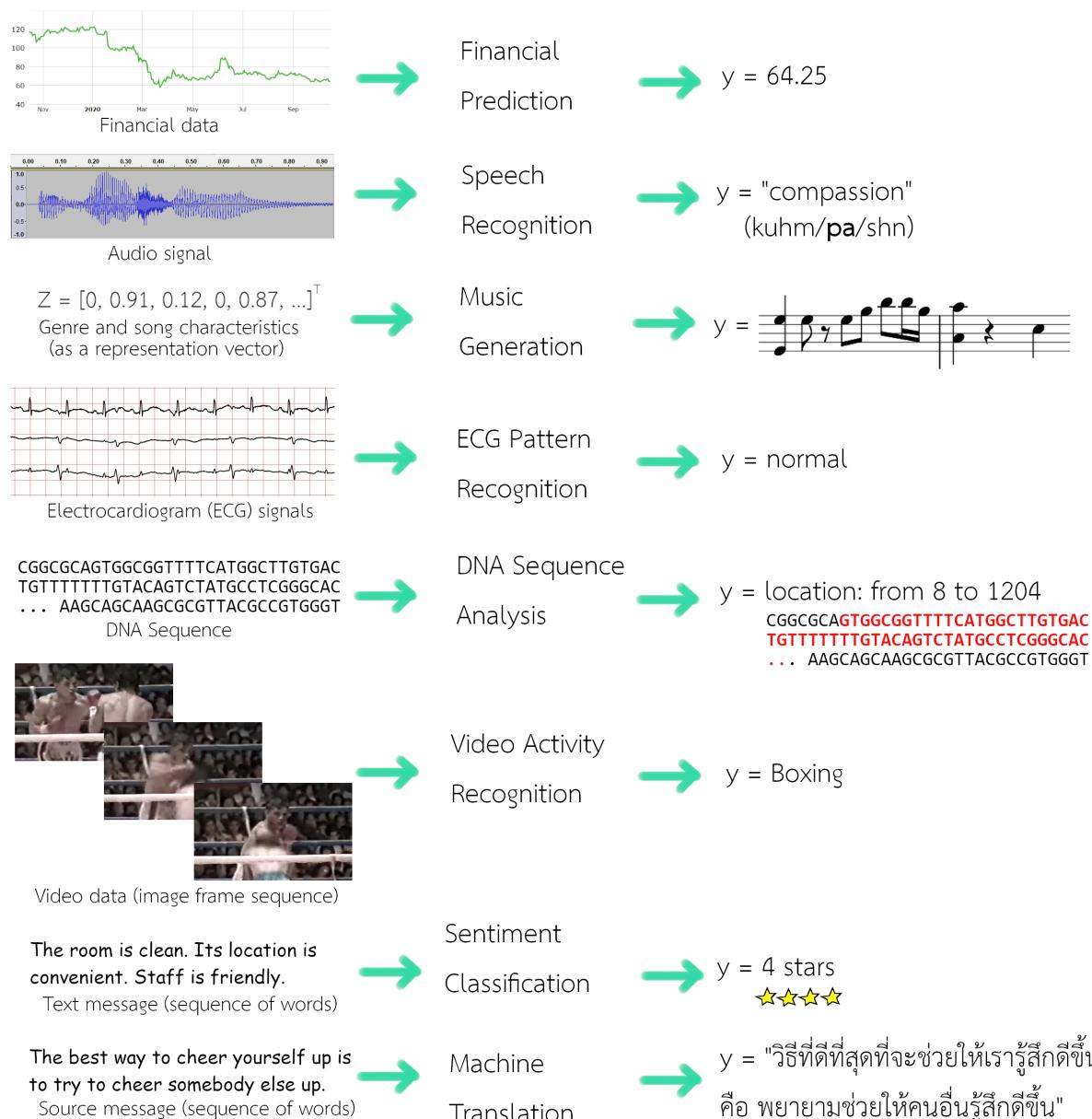
การสร้างแบบจำลองสำหรับข้อมูลเชิงลำดับ สามารถทำได้หลายแนวทาง. หัวข้อ 8.2 และ 8.3 อภิรายแนวทางการใช้ความน่าจะเป็นด้วยแบบจำลองมาร์คอฟ และแบบจำลองมาร์คอฟซ่อนเร้น. แนวทางของโครงข่ายประสาทเทียนกลับ จะถูกอภิรายในหัวข้อ 9.2.

8.2 แบบจำลองมาร์คอฟ

การสร้างแบบจำลองสำหรับข้อมูลเชิงลำดับ โดยรองรับความสัมพันธ์เชิงลำดับระหว่างลำดับต่าง ๆ อย่างสมบูรณ์ อาจทำให้การคำนวณทำได้ยาก. ตัวอย่างเช่น สำหรับการกิจการทำนายค่าในอนาคต นั่นคือ การทำนายจุดข้อมูลลำดับต่อไป \mathbf{x}_{T+1} ของข้อมูลชุดลำดับ $\{\mathbf{x}_t\}_{t=1,\dots,T}$. การใช้แบบจำลอง ความน่าจะเป็น¹ ด้วย $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ อาจอาศัยทฤษฎีของเบส (หัวข้อ 2.2) เพื่อประมาณค่าใน อนาคตด้วย $p(\mathbf{x}_{T+1} | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) = p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T, \mathbf{x}_{T+1}) / p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$. ค่า $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ เอง ก็ประมาณได้ยากในทางปฏิบัติ และถึงแม้จะรู้ แต่ค่าของ $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T, \mathbf{x}_{T+1})$ ยิ่งยากกว่าที่จะประมาณ หากยังใช้แบบจำลองที่รองรับความสัมพันธ์เชิงลำดับระหว่าง ลำดับต่าง ๆ อย่างสมบูรณ์.

แนวทางแก้ปัญหาเชิงคำนวณหลาย ๆ ครั้ง ที่การคำนวณค่าอย่างแม่นยำ (exact calculation) ไม่สามารถทำได้หรือทำได้ยาก การประมาณหรือการเพิ่มเงื่อนไขที่สมเหตุสมผล มักถูกนำมาใช้. วิธี หนึ่งคือ เลือกการประมาณที่สุดขอบ เช่น การประมาณโดยไม่สนใจความสัมพันธ์เชิงลำดับเลย (ใช้สม- มนติฐาน ไอ.ไอ.ดี.) ซึ่งวิธีนี้ทำให้เราสามารถเลือกแบบจำลองต่าง ๆ ที่ไม่มีความสามารถเชิงลำดับ เช่น

¹ หรือความหนาแน่นความน่าจะเป็น กรณีตัวแปรสุ่มต่อเนื่อง



รูปที่ 8.1 ตัวอย่างการรู้จำรูปแบบเชิงลำดับ. ภารกิจ (จากแคว้นลงล่าง) ได้แก่ การทำนายทางการเงิน อินพุตเป็นราคากิตติมัตต์ต่อวัน, การรู้จำเสียงพูด อินพุตเป็นสัญญาณเสียง ที่เป็นข้อมูลแอมพลิจูดต่อเวลา, ระบบแต่งเพลงอัตโนมัติ เอาต์พุตเป็นลำดับของโน้ตดนตรี (โดยอินพุตอาจเป็นค่าสเกลาร์หรือเวกเตอร์ ที่เป็นตัวแทนระบุลักษณะของเพลง), การรู้จำรูปแบบสัญญาณคลื่นไฟฟ้าหัวใจ อินพุตเป็นลำดับแอมพลิจูดหลาย ๆ ช่องสัญญาณต่อเวลา, การวิเคราะห์ลำดับเดือน蛾 อินพุตเป็นลำดับของชนิดฐานนิวเคลียร์, การรู้จำกิจกรรมจากวีดีโอ อินพุตเป็นภาพต่อเวลา, การจำแนกอารมณ์ อินพุตเป็นลำดับคำ และการแปลงภาษาอัตโนมัติ ที่ทั้งอินพุตและเอาต์พุตต่างก็เป็นข้อมูลลำดับของคำ. สังเกตภารกิจอาจเกี่ยวข้องกับข้อมูลเชิงลำดับ โดยรับอินพุตเป็นข้อมูลเชิงลำดับ หรือให้เอาต์พุตอكمabeineข้อมูลเชิงลำดับ หรือทั้งสองอย่าง. ข้อมูลเชิงลำดับอาจมีลำดับที่มีความหมายตามลำดับเวลาจริง ๆ เช่น ลำดับราคาปิดต่อวัน, ลำดับแอมพลิจูดต่อเวลา, ลำดับโน้ตดนตรีต่อเวลา และลำดับภาพต่อเวลา หรืออาจมีลำดับที่ไม่ได้มีความหมายกับเวลา เช่น ลำดับของชนิดฐานนิวเคลียร์ และลำดับของคำในข้อความ.

โครงข่ายประสาทเทียม (บทที่ 3) มาใช้ก็ได้. แต่การประมาณที่สุดของเรื่องนี้ เท่ากับเราทิ้งสารสนเทศความสัมพันธ์เชิงลำดับไปทั้งหมดเลย. เราไม่สามารถใช้ประโยชน์จากสารสนเทศเชิงลำดับได้เลย.

ยังไงนี่มาบ้าง 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 τ เป็นจำนวนจุดข้อมูลในลำดับก่อนหน้าที่แบบจำลองมาร์คอฟเลือกว่ามีความสัมพันธ์. อภิมานพารามิเตอร์ τ 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. รูปที่ 8.2 แสดงเงื่อนไขมาร์คอฟเปรียบเทียบกับสมมติฐานอื่น ๆ. เนื่องจากแบบจำลองมาร์คอฟอันดับหนึ่งเป็นที่นิยมเป็นอย่างมากทำให้หลาย ๆ ครั้งการอ้างถึงแบบจำลองมาร์คอฟ หมายถึงแบบจำลองมาร์คอฟอันดับหนึ่ง. เพื่อความกระชับ เนื้อหาต่อไปนี้จะอ้างถึงแบบจำลองมาร์คอฟอันดับหนึ่งว่า แบบจำลองมาร์คอฟ ยกเว้นแต่จะระบุเป็นอื่น.

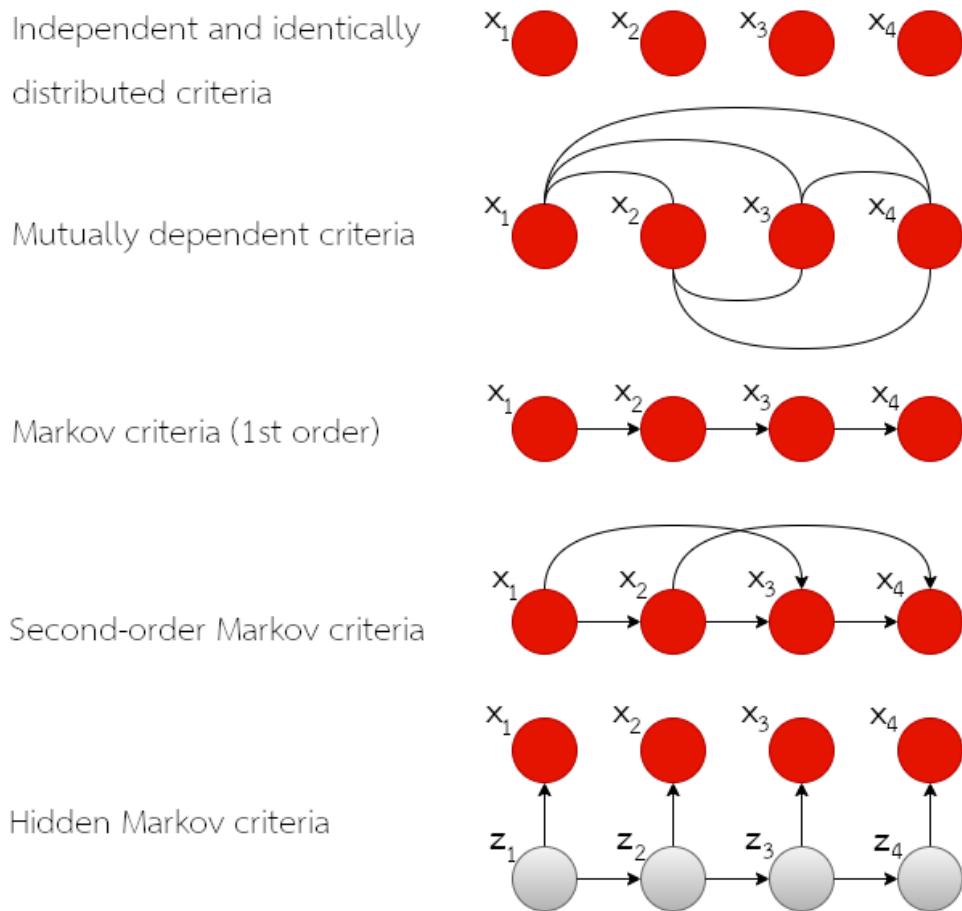
ด้วยแบบจำลองมาร์คอฟ การแจกแจงร่วมของลำดับข้อมูลสามารถวิเคราะห์ได้จาก

$$p(\mathbf{x}_1, \dots, \mathbf{x}_T) = p(\mathbf{x}_1) \cdot \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}). \quad (8.2)$$

การแจกแจงแบบมีเงื่อนไข $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ สามารถถูกกำหนดให้มีค่าเท่ากันทุก ๆ ตัวนี้ลำดับ t ตามสมมติฐานข้อมูลเชิงลำดับแบบคงที่. การกำหนดเช่นนี้ ช่วยให้เราสามารถใช้ค่าพารามิเตอร์ร่วมกันได้² ซึ่งมีผลช่วยลดความซับซ้อนของการคำนวณ และช่วยลดปริมาณข้อมูลที่ต้องการสำหรับการฝึกแบบจำลองด้วย. แบบจำลองมาร์คอฟ ช่วยให้การประมาณการแจกแจงร่วมของข้อมูลลำดับคำนวณได้สะดวกขึ้น.

แต่อย่างไรก็ตาม เงื่อนไขการขึ้นอยู่กับค่าจุดข้อมูลก่อนหนึ่งเพียงหนึ่งลำดับ ก็เป็นปัจจัยจำกัดความสามารถของแบบจำลองมาร์คอฟลงด้วย. เราอาจจะเพิ่มอันดับ (เพิ่มค่า τ) ของแบบจำลองมาร์คอฟ

² การใช้ค่าพารามิเตอร์ร่วมกัน (parameter sharing) เป็นปัจจัยที่สำคัญที่ช่วยให้แบบจำลองหลาย ๆ ชนิดสามารถทำงานได้ดี. ตัวอย่างเช่น โครงข่ายคอนโวลูชัน (บทที่ 6) มีการใช้ชั้นคำนวนคอนโวลูชัน ซึ่งอาศัยการเชื่อมต่อท้องถิ่น ที่ใช้ค่าพารามิเตอร์ร่วมกัน. เนื่องจากสมมติฐานของการข้ารูปแบบเชิงพื้นที่สมเหตุสมผลกับข้อมูลรูปภาพ โครงข่ายคอนโวลูชัน จึงสามารถทำงานได้ดีกับภาระกิจคอมพิวเตอร์วิวัฒนาต่าง ๆ.



รูปที่ 8.2 ตัวอย่างสมมติฐานแบบต่าง ๆ ของความสัมพันธ์ระหว่างจุดข้อมูล. แต่ละภาพ แสดงจุดข้อมูลสี่จุด. วงกลมเทibusสีแดง แทนจุดข้อมูล และตัวแปร x_i สำหรับ $i = 1, \dots, 4$ แทนค่าของจุดข้อมูล. ภาพบนสุด แสดงสมมติฐาน*ไอ.ไอ.ดี.* ที่ถือว่า ค่าของจุดข้อมูลไม่มีความสัมพันธ์ระหว่างกันเลย (แต่ค่าของจุดข้อมูลทุกจุดมาจากการแจกแจงเดียวกัน). ภาพที่สองจากบน แสดงสมมติฐานว่า ทุกจุดข้อมูลมีความสัมพันธ์ร่วมกัน (เกี่ยวพันกันหมด ไม่ว่ากับจุดข้อมูลลำดับก่อนหน้า หรือกับจุดข้อมูลลำดับหลัง). เส้นเชื่อมไม่มีหัวลูกศร แสดงความสัมพันธ์สองทาง. ภาพที่สามจากบน แสดงสมมติฐานมาร์คอฟ(อันดับหนึ่ง). เส้นเชื่อมมีหัวลูกศร แสดงความสัมพันธ์ทางเดียว นั่นคือ ค่าของจุดข้อมูลต้นทางของเส้นเชื่อม ส่งผลต่อค่าของจุดข้อมูลปลายทาง (ที่หัวลูกศรชี้). ภาพที่สี่จากบน แสดงสมมติฐานมาร์คอฟอันดับสอง. ภาพล่างสุด แสดงสมมติฐานมาร์คอฟซ่อนเร้น ที่ว่า ค่าของจุดข้อมูล x_i ขึ้นอยู่กับค่าของสถานะ z_i และค่าของสถานะ z_i ได้รับอิทธิพลมาจากค่าของสถานะก่อนหน้า z_{i-1} . ค่าของสถานะ z_i อาจสามารถสังเกตได้โดยตรง อาจมีความหมายซับเจกต์ หรืออาจจะไม่สามารถสังเกตได้โดยตรง หรืออาจเป็นสถานะที่สมมติขึ้นมาก็ได้.

ขึ้น ซึ่งก็จะอาจจะช่วยบรรเทาข้อจำกัดลงได้บ้าง ในเรื่องความสามารถที่จะเขียนໂ;yic ความสัมพันธ์เชิงลำดับระยะที่ยาวขึ้น. แต่แนวทางนี้ กลับส่งผลลบต่อประสิทธิภาพการคำนวณเป็นอย่างมาก (ศึกษารายละเอียดเพิ่มเติมจาก [15, §13.1]).

เพื่อแบบจำลองจะไม่ถูกจำกัดจำนวนลำดับย้อนหลังที่สัมพันธ์กันจากเงื่อนไขของมาร์คอฟ และก็ยังสามารถคำนวณได้อย่างมีประสิทธิภาพ เราสามารถดัดแปลงแบบจำลองได้โดยกำหนดให้ จุดข้อมูล \mathbf{x}_t สัมพันธ์กับสถานะ \mathbf{z}_t และสถานะ \mathbf{z}_t เป็นไปตามเงื่อนไขของมาร์คอฟ. เงื่อนไขเข่นี้ ช่วยให้แบบจำลองมีความยืดหยุ่น สามารถรองรับความสัมพันธ์ของจุดข้อมูลย้อนหลังกีลำดับก็ได้ โดยผ่านกลไกของตัวแปรสถานะ และยังคงรักษาการคำนวณที่มีประสิทธิภาพไว้ได้.

ค่าของสถานะ \mathbf{z}_t อาจมีความหมายซัดเจนหรือไม่ก็ได้ และอาจสามารถสังเกตได้โดยตรงหรือไม่ก็ได้. ดังนั้น สถานะ \mathbf{z}_t จึงถูกเรียกว่า **สถานะซ่อนเร้น** (*latent state*) หรือตัวแปรซ่อนเร้น (*latent variable*) และเงื่อนไขนี้ เรียกว่า **เงื่อนไขมาร์คอฟซ่อนเร้น** (*hidden Markov criteria*). บางครั้ง เพื่อลดความสับสน ตัวแปรจุดข้อมูล \mathbf{x}_t อาจถูกเรียกว่า **ตัวแปรที่ถูกสังเกต** (*observed variable*). รูปที่ 8.2 แสดงเงื่อนไขมาร์คอฟซ่อนเร้น (ภาพสุดท้าย) เปรียบเทียบกับสมมติฐานอื่น ๆ.

ค่าของสถานะซ่อนเร้น \mathbf{z}_t อาจจะเป็นค่าชนิดเดียวกับค่าของจุดข้อมูล \mathbf{x}_t ก็ได้ หรืออาจจะต่างกัน ก็ได้ เช่น ค่าของสถานะอาจเป็นค่าวิญญาณที่มีจำนวนจำกัด แต่ค่าของจุดข้อมูล \mathbf{x}_t อาจจะเป็นค่าวิญญาณที่มีจำนวนจำกัด หรือค่าวิญญาณที่มีจำนวนไม่จำกัด หรือค่าต่อเนื่องก็ได้. จำนวนมิติของสถานะซ่อนเร้น อาจ จะเท่ากับ 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 การแจกแจงร่วมของลำดับชุดข้อมูลและชุดสถานะ 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*).

แบบจำลองปริภูมิสถานะที่สำคัญได้แก่ แบบจำลองมาร์คอฟซ่อนเร้น ที่จะจะสำหรับกรณีสถานะซ่อนเร้นเป็นตัวแปรวิญญาณ และแบบจำลองพลวัตเชิงเส้น (*linear dynamic model* หรือ *linear dynamic system*) ที่จะจะสำหรับกรณีที่สถานะซ่อนเร้นและตัวแปรที่ถูกสังเกต เป็นตัวแปรต่อเนื่อง ที่มีการแจกแจงเกาส์เชียน.

8.3 แบบจำลองมาร์คอฟซ่อนเร้น

แบบจำลองมาร์คอฟซ่อนเร้น³ (*hidden Markov model*) เป็นแบบจำลองสำหรับข้อมูลเชิงลำดับ ที่ใช้เงื่อนไขมาร์คอฟซ่อนเร้น สำหรับกรณีที่สถานะซ่อนเร้นเป็นตัวแปรวิญญาณ ที่ค่าวิญญาณมีจำนวนจำกัด. เมื่อ

³ เนื้อหาในส่วนของแบบจำลองมาร์คอฟซ่อนเร้น ได้รับอิทธิพลหลัก ๆ จาก [15].

พิจารณาสมการ 8.3 ด้วยสมมติฐานสถานะซ่อนเร้นวิถุ 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 π_k เป็นความน่าจะเป็นที่ระบบจะเริ่มต้นด้วยสถานะ k . ค่าของ π_k ต่าง ๆ เป็นพารามิเตอร์ของแบบจำลอง (กระบวนการหาค่าพารามิเตอร์เหล่านี้จะอภิปรายในหัวข้อ 8.3.)

ความน่าจะเป็น $p(z_t | z_{t-1})$ ที่เรียกว่า ความน่าจะเป็นของการเปลี่ยนสถานะ (transition probabilities) สามารถแทนด้วยเมตริกซ์ \mathbf{A} ที่ส่วนประกอบ A_{ij} แทนความน่าจะเป็นของการเปลี่ยนสถานะจากสถานะ i ไปเป็นสถานะ j . เมตริกซ์ \mathbf{A} อาจถูกเรียกว่า เมตริกซ์การเปลี่ยนสถานะ (transition matrix).

เพื่อความสะดวก สถานะซ่อนเร้นได้ K สถานะ สามารถแสดงด้วยรหัสหนึ่งร้อน (one-hot coding ดูหัวข้อ 3.3) ซึ่งคือ สถานะซ่อนเร้น $\mathbf{z}_t = [z_{t,1}, \dots, z_{t,K}]^T \in \{0, 1\}^K$ 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 ด้วยสมมติฐานข้อมูล เชิงลำดับแบบคงที่ ความน่าจะเป็นของการเปลี่ยนสถานะ $p(\mathbf{z}_t | \mathbf{z}_{t-1})$ จะถูกแทนด้วยเมตริกซ์การเปลี่ยนสถานะ \mathbf{A} เมื่อกันสำหรับทุก ๆ ค่าของลำดับ t . นอกจากนั้น 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 $\sum_{j=1}^K A_{ij} = 1$.

เช่นเดียวกับพารามิเตอร์ $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]^T$ เมตริกซ์ \mathbf{A} ก็เป็นพารามิเตอร์ของแบบจำลอง. เราสามารถเขียนฟังก์ชันการแจกแจงเริ่มต้น และฟังก์ชันการแจกแจงการเปลี่ยนสถานะ โดยเน้นพารามิเตอร์เหล่านี้ได้⁴

$$p(\mathbf{z}_1 | \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_{1k}}. \quad (8.4)$$

$$p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{A}) = \prod_{j=1}^K \prod_{i=1}^K A_{ij}^{z_{t-1,i} \cdot z_{t,j}}. \quad (8.5)$$

ตัวอย่างเช่น สำหรับระบบที่มีสามสถานะซ่อนเร้น การคำนวณฟังก์ชันการแจกแจงเริ่มต้น สำหรับสถานะที่หนึ่ง ($\mathbf{z}_1 = [1, 0, 0]^T$) ทำโดย $p(\mathbf{z}_1 = [1, 0, 0]^T | \boldsymbol{\pi}) = \pi_1^1 \cdot \pi_2^0 \cdot \pi_3^0 = \pi_1$.

สุดท้ายฟังก์ชันการแจกแจงแบบมีเงื่อนไข สำหรับตัวแปรที่ถูกสังเกต $p(\mathbf{x}_t | \mathbf{z}_t)$ มักถูกเรียกว่า ความน่าจะเป็นของการปล่อย (emission probabilities). เช่นเดียวกัน ด้วยสมมติฐานข้อมูล เชิงลำดับแบบคงที่ ความน่าจะเป็นของการปล่อย ไม่ขึ้นกับดัชนีลำดับ t . เราอาจเขียนความน่าจะเป็นของการปล่อยด้วย $p(\mathbf{x} | \mathbf{z})$ โดยละเอียดนี้ลำดับออกได้.

ความน่าจะเป็นของการปล่อย อาจถูกนิยามด้วยการแจกแจงที่เหมาะสมกับลักษณะของตัวแปรที่ถูกสังเกต เช่น หากตัวแปรที่ถูกสังเกตเป็นค่าวิถุ เราอาจประมาณความน่าจะเป็นของการปล่อย ด้วย

⁴วิธีการเขียนเช่นนี้ เป็นการเขียนจากมุมมองคณิตศาสตร์เพื่อให้คำนิยามต่าง ๆ สมบูรณ์. การนำพจน์ต่าง ๆ เหล่านี้ไปใช้ในโปรแกรม อาจดำเนินการต่างไป เพื่อให้คอมพิวเตอร์สามารถคำนวณได้อย่างมีประสิทธิภาพ.

เมทริกซ์แจกแจง Φ ที่ส่วนประกอบ ϕ_{kd} ระบุค่าความน่าจะเป็นที่ตัวแปรที่ k กลังเกตจะเป็นชนิดที่ d^{th} เมื่อสถานะช่อนเร้นเป็นสถานะที่ k^{th} . นั่นคือ $p(\mathbf{x}|\mathbf{z}, \Phi) = \prod_d \prod_k \phi_{kd}^{z_k \cdot x_d}$ เมื่อ $\phi_{kd} \equiv p(x_d = 1|z_k = 1)$ และทั้ง \mathbf{x} และ \mathbf{z} แสดงด้วยรหัสหนึ่งร้อน.

หากตัวแปรที่ k กลังเกตเป็นค่าต่อเนื่อง เราอาจเลือกการแจกแจงเกาล์เซียน สำหรับประมาณความน่าจะเป็นของการปล่อย. 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 เป็นจำนวนมิติของเวกเตอร์ \mathbf{x} และ $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,\dots,K}$ เป็นพารามิเตอร์ของแบบจำลอง. หมายเหตุ ค่า π ในที่นี้หมายถึง ค่าคงที่เพ ($\pi \approx 3.1416$) ซึ่งต่างจาก π_k (เช่นในสมการ 8.4) ที่เป็นพารามิเตอร์ของแบบจำลอง. สังเกตว่า ถึงแม้แบบจำลองมาร์คอฟช่อนเร้น จะกำหนดให้สถานะช่อนเร้น \mathbf{z} เป็นตัวแปรค่าวิญญาต แต่ตัวแปรที่ k กลังเกต \mathbf{x} อาจจะเป็นค่าวิญญาต หรือเป็นค่าต่อเนื่องก็ได้.

เพื่อความสะดวก ต่อจากนี้ พารามิเตอร์ของฟังก์ชันประมาณความน่าจะเป็นของการปล่อย จะถูกจัดถึงโดยรวมด้วยสัญกรณ์ $\boldsymbol{\phi}$ ไม่ว่าจะเลือกใช้การแจกแจงแบบใด และ $\boldsymbol{\phi}_k$ จะหมายถึงชุดพารามิเตอร์ของการแจกแจง ของสถานะช่อนเร้น k^{th} เช่น กรณีเมทริกซ์แจกแจง $\boldsymbol{\phi}_k \equiv \prod_d \phi_{kd}^{x_d}$ หรือกรณีเกาล์เซียน $\boldsymbol{\phi}_k \equiv \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. นั่นคือ ฟังก์ชันประมาณความน่าจะเป็นของการปล่อย หรือเรียกว่า ๆ ว่า ฟังก์ชันการปล่อย (emission function) จะใช้สัญกรณ์ $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\phi})$ โดย $\boldsymbol{\phi}$ หมายถึงพารามิเตอร์ของแบบจำลองที่ใช้ (อาจจะหมายถึง เมทริกซ์ Φ หรือเซต $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1,\dots,K}$ หรือชุดพารามิเตอร์อื่น ๆ ตามการแจกแจงที่เลือก). ฟังก์ชันการปล่อย อาจเขียนได้ โดยทั่วไปเป็น

$$p(\mathbf{x}_t|\mathbf{z}_t, \boldsymbol{\phi}) = \prod_{k=1}^K (p(\mathbf{x}_t|\boldsymbol{\phi}_k))^{z_{t,k}}. \quad (8.6)$$

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 โดยเฉพาะทฤษฎีของเบส (หัวข้อ 2.2) เราสามารถทำการอนุมานต่าง ๆ ได้หากรู้ค่าของชุดพารามิเตอร์ $\boldsymbol{\theta}$ เช่น กรณีการทำนายข้อมูลการเงิน การทายค่าลำดับต่อไป $\hat{\mathbf{x}}_{T+1}$ ก็สา-

มารถอนุมานจาก⁵ $\hat{\mathbf{x}}_{T+1} \approx E[\mathbf{x}_{T+1} | \mathbf{X}, \boldsymbol{\theta}] = \sum_d \sum_k \sum_j \mathbf{x}_{T+1,d} \cdot p(\mathbf{x}_{T+1,d} | z_{T+1,k}, \boldsymbol{\phi}) \cdot p(z_{T+1,k} | z_{T,j}, \mathbf{A}) \cdot p((z_{T,j} | \mathbf{X}, \boldsymbol{\theta})$ และ $p((z_T | \mathbf{X}, \boldsymbol{\theta}) = \sum_{\{z_1, \dots, z_{T-1}\}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})$ 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 $\boldsymbol{\theta}$ สามารถทำได้ด้วยวิธีค่าฟังก์ชันความเป็นสูงสุด (maximum likelihood ดูแบบฝึกหัด 5.18). แนวคิดของวิธีค่าฟังก์ชันควรจะเป็นสูงสุด คือ การหาค่าของชุดพารามิเตอร์ ที่ทำให้ค่าความน่าจะเป็นภายหลัง (posterior) มีค่ามากกว่าที่สุด โดยความน่าจะเป็นภายหลัง คือค่าความน่าจะเป็นที่คำนวณด้วยค่าต่าง 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 $p(\mathbf{X} | \boldsymbol{\theta})$ จะหาได้โดยการ слายปัจจัย (marginalization)

$$p(\mathbf{X} | \boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}). \quad (8.8)$$

ในทางปฏิบัติ การคำนวณสมการ 8.8 โดยตรง ทำได้ยาก เพราะว่า การ слายปัจจัย อาศัยการบวกทุกค่าที่เป็นไปได้ของชุดลำดับสถานะซ่อนเร้น $\mathbf{Z} = \{z_1, \dots, z_T\}$ และ ที่แต่ละลำดับ สถานะซ่อนเร้นก็มีโอกาสเป็นไปได้หลายค่า. ในที่นี้ กำหนดให้ ค่าของสถานะซ่อนเร้นที่เป็นไปได้มีจำนวนเป็น K ค่า. ด้วยความยาวของลำดับเป็น T ทำให้มีชุดค่าของลำดับสถานะซ่อนเร้นที่เป็นไปได้เท่ากับ K^T ชุด. จำนวนพจน์ที่ต้องทำการบวก จะเพิ่มขึ้นแบบขี้กำลังตามความยาวของชุดลำดับ.

แนวทางหนึ่งที่นิยมสำหรับแบบจำลองมาร์คอฟซ่อนเร้น คือ ขั้นตอนวิธีอีเม็ม (expectation-maximization algorithm คำย่อ EM algorithm) ซึ่งเป็นขั้นตอนวิธีทั่ว ๆ ไปสำหรับการหาค่าพารามิเตอร์ของแบบจำลอง โดยมีนัยของความน่าจะเป็นประกอบ. กล่าวโดยคร่าว 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 เพื่อประเมินค่าคาดหมายของลอกการทึบของค่าฟังก์ชันควรจะเป็นของข้อมูล

$$\varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \quad (8.9)$$

เมื่อ $\boldsymbol{\theta}_0$ เป็นค่าพารามิเตอร์ ณ ปัจจุบัน ส่วน $\boldsymbol{\theta}$ เป็นตัวแปรของค่าพารามิเตอร์ (ที่ต้องการจะปรับปรุงใหม่). การประเมินพังก์ชันควรจะเป็น $\varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0)$ จะใช้ในการบวนการหาค่าพารามิเตอร์ที่ดีที่สุด. หลัง

⁵ ตัวอย่างนี้ ต้องการแสดงในเห็นคร่าว ๆ เท่านั้นว่า การแยกแจงร่วมสามารถนำไปใช้ในการอนุมานต่าง ๆ ได้อย่างไร อย่างน้อยในทางทฤษฎี. การอนุมานในทางปฏิบัติ อาจต้องการขั้นตอนวิธีคำนวณที่มีประสิทธิภาพมากกว่าการประยุกต์ใช้ทฤษฎีความน่าจะเป็นตรง ๆ ดังที่แสดงในตัวอย่างนี้.

จากได้ค่าพารามิเตอร์ชุดใหม่แล้ว จึงวนซ้ำคำนวณในลักษณะเช่นนี้ต่อไปจนค่าต่าง ๆ ลู่เข้า หรือจนกว่าจะเป็นไปตามเงื่อนไขการจบการคำนวณ.

ขั้นตอนวิธีอีเม็ม. กล่าวโดยละเอียดแล้ว ด้วยค่าพารามิเตอร์ $\boldsymbol{\theta}_0$ ขั้นตอนวิธีอีเม็ม เริ่มที่ขั้นตอนการหาค่าคาดหมายใช้ค่าพารามิเตอร์นี้ในการคำนวณค่าคาดหมายของสถานะช่องเรื่น.

เพื่อความสะดวก นิยามความน่าจะเป็นภายหลังของสถานะช่องเรื่นด้วย \mathbf{q}_t และนิยามความน่าจะเป็นภายหลังร่วมด้วย $\mathbf{R}^{(t-1,t)}$. นั่นคือ

$$\mathbf{q}_t \equiv p(\mathbf{z}_t | \mathbf{X}, \boldsymbol{\theta}_0) \quad (8.10)$$

$$\mathbf{R}^{(t-1,t)} \equiv p(\mathbf{z}_{t-1}, \mathbf{z}_t | \mathbf{X}, \boldsymbol{\theta}_0) \quad (8.11)$$

โดย สำหรับแต่ละดัชนีลำดับ 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} เป็นตัวแปรค่าทวีภาค นั่นทำให้ ค่าคาดหมายของสถานะช่องเรื่น $E[z_{tk}] = p(z_{tk} = 1 | \mathbf{X}, \boldsymbol{\theta}_0) \cdot 1 + p(z_{tk} = 0 | \mathbf{X}, \boldsymbol{\theta}_0) \cdot 0 = p(z_{tk} = 1 | \mathbf{X}, \boldsymbol{\theta}_0) = q_{tk}$. 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 $\mathbf{R}^{(t-1,t)}$ จึงเทียบเท่ากับการหาค่าคาดหมาย. การคำนวณในขั้นตอนนี้ จึงถูกเรียกว่าเป็น ขั้นตอนการหาค่าคาดหมาย.

ด้วยความน่าจะเป็นภายหลังทั้ง \mathbf{q}_t และ $\mathbf{R}^{(t-1,t)}$ และนำแบบจำลองมาร์คอฟช่องเรื่น ในสมการ 8.7 มาประกอบ เราจะได้พังก์ชันควรจะเป็น (สมการ 8.9) ว่า

$$\begin{aligned} \varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) &= \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln \left\{ p(\mathbf{z}_1 | \boldsymbol{\pi}) \cdot \left(\prod_{t=2}^T p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{A}) \right) \cdot \prod_{\tau=1}^T p(\mathbf{x}_{\tau} | \mathbf{z}_{\tau}, \boldsymbol{\phi}) \right\} \\ &= \sum_{k=1}^K q_{1k} \cdot \ln p(\mathbf{z}_1 | \boldsymbol{\pi}) + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{A}) \\ &\quad + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(\mathbf{x}_{\tau} | \mathbf{z}_{\tau}, \boldsymbol{\phi}) \\ &= \sum_{k=1}^K q_{1k} \cdot \ln \pi_k + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln A_{jk} + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(\mathbf{x}_{\tau} | \boldsymbol{\phi}_k) \end{aligned} \quad (8.12)$$

โดย $\boldsymbol{\pi}$ และ \mathbf{A} เป็นค่าความน่าจะเป็นเริ่มต้น และความน่าจะเป็นของการเปลี่ยนสถานะ ตามลำดับ. ส่วน $p(\mathbf{x}_{\tau} | \boldsymbol{\phi}_k)$ คือความน่าจะเป็นของการปล่อยของสถานะที่ k^{th} .

ขั้นตอนการหาค่าตัวทำมากที่สุด หาค่าของพารามิเตอร์ $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$ จากค่าที่ทำให้พังก์ชันควรจะเป็น (สมการ 8.12) มีค่ามากที่สุด (โดยค่า \mathbf{q}_t และ $\mathbf{R}^{(t-1,t)}$ จะใช้ค่าที่ได้จากขั้นตอนการหาค่า

คาดหมาย และคิดเห็นเป็นค่าคงที่). ค่าพารามิเตอร์ $\boldsymbol{\pi}$ และ \mathbf{A} หาได้จากค่าทำงานที่สุดของฟังก์ชัน ควรจะเป็น ประกอบกับเงื่อนไขของพารามิเตอร์ ได้แก่ $\sum_k \pi_k = 1$ และ $\sum_k A_{jk} = 1$ และได้ผลลัพธ์ (ดูแบบฝึกหัด 8.2) คือ

$$\pi_k = \frac{q_{1k}}{\sum_{j=1}^K q_{1j}} \quad (8.13)$$

$$A_{jk} = \frac{\sum_{t=2}^T R_{jk}^{(t-1,t)}}{\sum_{l=1}^K \sum_{t=2}^T R_{jl}^{(t-1,t)}}. \quad (8.14)$$

สำหรับพารามิเตอร์ $\boldsymbol{\phi}$ การหาค่ากีสามารถทำได้ในทำนองเดียวกัน เพียงแต่แบบจำลองมาร์คอฟซ่อนรseen เปิดกว้างสำหรับการเลือกใช้ฟังก์ชันการปล่อย $p(\mathbf{x}_t | \boldsymbol{\phi}_k)$ ได้หลายแบบ.

หากเลือกฟังก์ชันการปล่อยเกาส์เซียน นั่นคือ $p(\mathbf{x}_t | \boldsymbol{\phi}_k) \equiv \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. เมื่อทำการหาค่าตัวทำงานที่สุด 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 ϕ_{kd} แทนค่าความน่าจะเป็นที่จะพบตัวแปรที่ถูกสังเกต \mathbf{x}_t เป็นชนิด d^{th} หากสถานะซ่อนรseen เป็นชนิด 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 $\sum_{d=1}^D x_{td} = 1$. เมื่อทำการหาค่าตัวทำงานที่สุด 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 $\boldsymbol{\phi}$ อาจกำหนดเป็นสมैือนพารามิเตอร์อีกส่วน และใช้กระบวนการเรียนรู้จากข้อมูลเพื่อช่วยกำหนดค่าได้.

ขั้นตอนวิธีอีเม็ม มีคุณสมบัติความถูกต้องและคุณสมบัติการถูกลื้เข้าที่ดี (ดู [188] เพิ่มเติม). อย่างไรก็ตาม ในขั้นตอนการหาค่าคาดหมาย การประมาณค่าของค่าความน่าจะเป็นภายหลัง \mathbf{q}_t และ $\mathbf{R}^{(t-1,t)}$ แม้อาจสามารถทำได้โดยวิธีการลสายปัจจัย 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.

เฉพาะการคำนวณ $\sum_{\{z_1, \dots, z_{t-1}, z_t+1, \dots, z_T\}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0)$ เองอย่างเดียว ก็เท่ากับต้องบวกจนทั้งหมด $(T-1) \cdot K$ พจน์เข้าด้วยกัน และค่าของแต่ละพจน์ ก็ต้องผ่านการคำนวณต่าง ๆ ดังอภิราย. ดังนั้น เมื่อต้องนำมาใช้กับขั้นตอนวิธีอีเม็ม ซึ่งต้องคำนวณค่าความน่าจะเป็นภายหลังใหม่ทุก ๆ สมัยฝึกการประมาณค่า \mathbf{q}_t และ $\mathbf{R}^{(t-1,t)}$ ด้วยวิธีนี้ จึงไม่เหมาะสมที่จะนำมาใช้ได้ในทางปฏิบัติ.

ปัญหาการคำนวณค่าความน่าจะเป็นภายหลังอย่างมีประสิทธิภาพ ถูกบรรเทาด้วยขั้นตอนวิธีไปข้างหน้ากับถอยกลับ ที่จะอภิรายต่อไปในหัวข้อ 8.3.

ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ

ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ (forward-backward algorithm) เป็นขั้นตอนวิธี ที่ใช้คำนวณค่าของความน่าจะเป็นภายหลังที่ใช้ในวิธีอีเม็ม ได้อย่างมีประสิทธิภาพ. จริง ๆ แล้ว ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ มีการศึกษาอย่างกว้าง และวิธีดำเนินการก็มีอยู่หลายแบบ หัวข้อนี้จะอภิรายแบบหนึ่งที่มีการใช้อย่างกว้างขวาง[15] เรียกว่า ขั้นตอนวิธีแอลfa-บีตา (alpha-beta algorithm).

เนื่องจาก หัวข้อนี้พิจารณาค่าพารามิเตอร์ $\boldsymbol{\theta}_0$ เป็นเหมือนค่าคงที่ ดังนั้นเงื่อนไข $\boldsymbol{\theta}_0$ จะถูกละไว้ในฐานที่เข้าใจ เพื่อความกระชับ.

ด้วยเงื่อนไขของแบบจำลองมาร์คอฟช่องเรื่อง 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 และกฎผลคุณ 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 จัดรูปการคำนวน $\alpha(\mathbf{z}_t)$ และ $\beta(\mathbf{z}_t)$ ให้อยู่ในรูปที่สามารถคำนวนได้อย่างมีประสิทธิภาพ โดยอาศัยความลัมพันธ์แบบเรียกซ้ำ (recursive relation). การคำนวนค่า $\alpha(\mathbf{z}_t)$ สามารถจัดรูปใหม่ได้ดังนี้

$$\begin{aligned} \alpha(\mathbf{z}_t) &= p(\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{z}_t) \\ &= p(\mathbf{x}_1, \dots, \mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_t) \cdot p(\mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_t) \\ &= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_{t-1}, \mathbf{z}_t) \end{aligned}$$

$$\begin{aligned}
&= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_t | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_{t-1}) \\
&= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1} | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_{t-1}) \\
&= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1}) \\
&= p(\mathbf{x}_t | \mathbf{z}_t) \cdot \sum_{\mathbf{z}_{t-1}} \alpha(\mathbf{z}_{t-1}) \cdot p(\mathbf{z}_t | \mathbf{z}_{t-1})
\end{aligned} \tag{8.31}$$

สำหรับ $t = 2, \dots, T$.

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{z}_1) \cdot p(\mathbf{x}_1 | \mathbf{z}_1) = \prod_{k=1}^K (\pi_k \cdot p(\mathbf{x}_1 | \boldsymbol{\phi}_k))^{\mathbf{z}_{1k}}. \tag{8.32}$$

นั่นคือ $\alpha(z_{1k}) = \pi_k \cdot p(\mathbf{x}_1 | \boldsymbol{\phi}_k)$.

การคำนวณเริ่มจากลำดับเวลาแรก แล้วค่อยๆ คำนวณขึ้นไปทีละลำดับ. การคำนวณสมการ 8.31 ทำการบวก K พจน์ สำหรับแต่ละสถานะซ่อนเร้น ซึ่งสถานะซ่อนเร้นมีจำนวนทั้งหมด K สถานะ. ดังนั้น ที่แต่ละลำดับเวลา 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

$$p(\mathbf{z}_T | \mathbf{X}) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_T, \mathbf{z}_T) \cdot \beta(\mathbf{z}_T)}{p(\mathbf{X})} = \frac{p(\mathbf{X}, \mathbf{z}_T)}{p(\mathbf{X})} \cdot \beta(\mathbf{z}_T) \tag{8.34}$$

และจากกฎคุณ ซึ่ง ณ ที่นี่ คือ $\frac{p(\mathbf{X}, \mathbf{z}_T)}{p(\mathbf{X})} = p(\mathbf{z}_T | \mathbf{X})$ ดังนั้น ค่าของ $\beta(z_T)$ 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 แล้วขยับลำดับไปข้างหน้า การคำนวณ $\beta(\mathbf{z}_t)$ เริ่มจากลำดับเวลาท้ายสุด แล้วขยับย้อนถอยหลังมาเรื่อย 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 $\beta(\mathbf{z}_t)$ ที่ได้ ไปใช้ในขั้นตอนการหาค่าตัวทำมากที่สุดโดยตรงเลยก็ได้ 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 $p(\mathbf{X})$. ค่าดัชนีลำดับที่สะดวกในกรณีนี้ คือ $t = T$ ซึ่งจะได้

$$p(\mathbf{X}) = \sum_{\mathbf{z}_T} \alpha(\mathbf{z}_T) \quad (8.37)$$

เพราะว่า $\beta(z_{Tk}) = 1$ สำหรับทุก ๆ ค่าของ k .

สังเกตว่า ค่า $p(\mathbf{X})$ อาจหาได้โดยการ слายปัจจัย $p(\mathbf{X}) = \sum_{\mathbf{Z}} p(\mathbf{Z}, \mathbf{X})$ แต่การทำเช่นนั้น เท่ากับการบวกของ K^T พจน์ ซึ่งแต่ละพจน์ต้องประเมินค่า $p(\mathbf{Z}, \mathbf{X})$ เปรียบเทียบกับสมการ 8.37 ซึ่งเท่ากับการบวกของ K พจน์เท่านั้น. การจัดรูปการคำนวณในสมการ 8.37 จึงเปลี่ยนการคำนวณ ที่ปริมาณเป็นสัดส่วนเติบโตแบบซึ่งกำลัง มาเป็นสัดส่วนแบบเชิงเส้น ลดการคำนวณลงได้มากาล โดยเฉพาะกับชุดลำดับข้อมูลยาว ๆ.

การอนุมานข้อมูลด้วยแบบจำลองมาร์คอฟช่อนเร้น. แบบจำลองมาร์คอฟช่อนเร้น สามารถนำประยุกต์ใช้ได้กว้างขวางในการอนุมานต่าง ๆ. ตัวอย่างหนึ่งที่สำคัญ คือ กรณีการอนุมานจุดข้อมูลต่อไปในชุดลำดับ เช่น ในกรณีการทำนายทางการเงิน (ภาพบนสุด รูปที่ 8.1 ที่อินพุตเป็นชุดลำดับของราคาปิดต่อวัน จากหลาย ๆ วันที่ผ่านมา และเอาต์พุตคือค่าทำนายราคาปิดของวันปัจจุบัน). นั่นคือ ด้วยข้อมูลชุดลำดับที่ถูกสังเกตมา $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ เราต้องการทำนายจุดข้อมูล \mathbf{x}_{T+1} . ด้วยเงื่อนไข มาร์คอฟและทฤษฎีของเบส์ การอนุมานอาจทำผ่านค่าความน่าจะเป็น 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.

ข้อจำกัดของแบบจำลองมาร์คอฟช่อนเร้น. แบบจำลองมาร์คอฟช่อนเร้น เป็นแบบจำลองเชิงความน่าจะเป็น แม้จะมีศักยภาพสูง แต่ปัจจุบัน การประยุกต์ใช้กับภาระกิจการรู้จำรูปแบบเชิงลำดับที่ซับซ้อน เช่น การรู้จำเสียงพูด และการประมวลผลภาษาธรรมชาติ นิยมใช้แนวทางของโครงข่ายประสาท เวียนกลับ (หัวข้อ 9.2) หากกว่า เพราะว่า โครงข่ายประสาทเวียนกลับสามารถทำภาระกิจดังกล่าวได้ดีกว่ามาก. ปัจจัยที่เป็นสาเหตุของความสามารถที่จำกัดของแบบจำลองมาร์คอฟช่อนเร้น อาจได้แก่ จำนวนสถานะช่อนเร้น ถูกจำกัด หรือการที่ต้องกำหนด ระบุจำนวนสถานะอย่างชัดเจนในตัวแบบจำลอง, การเรียนรู้ความสัมพันธ์ระยะยาว ต้องการอาศัยการทำผ่านตัวแปรช่อนเร้น ซึ่งถูกจำกัดจำนวนสถานะไว้ และการใช้รหัสหนึ่งร้อน ยังทำให้ตัวแปรช่อนเร้นไม่อาจแทนความสัมพันธ์ที่ซับซ้อนอย่างมีประสิทธิภาพได้ คือ ไม่สามารถระบุแยกความสัมพันธ์เป็นองค์ประกอบบ่อยครั้งได้ และอาจจะรวมถึง การอาศัยมุมมองเชิงความน่าจะเป็น ซึ่งอยู่บนสมมติฐานของการวิเคราะห์ที่ครอบคลุมทุกรณี ที่อาจจะส่งผลในทาง

ปฏิบัติ (หากไม่นึกໄกพิเศษ เพื่อแก้ไข ชดเชย หรือบรรเทาการคำนวณที่พัฒนาจากสมมติฐานเช่นนี้⁶).

8.4 อภิธานศัพท์

ข้อมูลเชิงลำดับ (sequential data): ข้อมูลประเภทที่จุดข้อมูลมีความสัมพันธ์เชิงลำดับระหว่างกัน

แบบจำลองมาร์คอฟ (Markov model): แบบจำลองความสัมพันธ์เชิงลำดับของข้อมูล ที่ใช้เงื่อนไข มาร์คอฟ ซึ่งจำกัดให้จุดข้อมูลมีความความสัมพันธ์เชิงความไม่เป็นอิสระต่อกันแบบมีเงื่อนไข (*conditional dependence*) ระหว่างกันได้ เฉพาะกับจุดข้อมูลลำดับที่ผ่านมา ตามจำนวน ลำดับที่กำหนด.

แบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov model คำย่อ HMM): แบบจำลองสำหรับข้อมูล เชิงลำดับ ที่บรรยายความสัมพันธ์เชิงลำดับของค่าข้อมูล ผ่านตัวแปรซ่อนเร้น และใช้เงื่อนไข มาร์คอฟกับตัวแปรซ่อนเร้น.

ตัวแปรที่สังเกตได้ (observable variable): ค่าจุดข้อมูล ซึ่งเป็นค่าที่สามารถรู้และแน่นอนได้ **สถานะซ่อนเร้น (latent state) หรือ ตัวแปรซ่อนเร้น (latent variable):** ค่าจุดข้อมูลที่สมมติขึ้น หรือเชื่อว่ามีอยู่ อาจจะสามารถรู้ค่าແเนื่องนondได้ หรืออาจจะไม่สามารถรู้ค่าได้เลย และอาจมีนัย ความหมายจริงก็ได้ หรืออาจจะไม่ได้มีนัยความหมายที่ชัดเจนก็ได้.

ค่าความน่าจะเป็นเริ่มต้น (initial probabilities): ความน่าจะเป็นของสถานะต่าง ๆ ของจุดข้อมูล ที่ลำดับแรกสุด

ความน่าจะเป็นของการเปลี่ยนสถานะ (transition probabilities): ความน่าจะเป็นของสถานะต่าง ๆ ของจุดข้อมูลที่ลำดับถัดไป เมื่อจุดข้อมูลลำดับปัจจุบันมีสถานะดังระบุ.

ความน่าจะเป็นของการปล่อย (emission probabilities): ความน่าจะเป็นของค่าจุดข้อมูลที่สัง- กетеได้ เมื่อสถานะซ่อนเร้นมีค่าดังระบุ

ฟังก์ชันควรจะเป็น (likelihood function): ฟังก์ชันของตัวแปรที่สนใจ ที่คำนวณค่าความน่าจะเป็น ด้วยค่าข้อมูลที่สังเกตได้

ขั้นตอนวิธีอิเม็ม (expectation-maximization algorithm คำย่อ EM algorithm): ขั้นตอนวิธี ที่นำไป สำหรับการหาค่าพารามิเตอร์ ของแบบจำลองเชิงความน่าจะเป็น โดยอาศัยขั้นตอนการ คำนวณค่าคาดหมาย เมื่อกำหนดค่าพารามิเตอร์ที่สนใจเป็นค่าคงที่ สลับกับขั้นตอนการหาค่า มากที่สุด ที่ใช้ค่าคาดหมายที่ได้ประเมินค่าฟังก์ชันควรจะเป็น สำหรับการหาค่าพารามิเตอร์ที่ ทำให้ฟังก์ชันควรจะเป็นมีค่ามากที่สุด.

⁶งานวิจัย [134, 136] ได้ชี้ให้เห็นปัญหาของสมมติฐานว่าการวิเคราะห์ครอบคลุมทุกรณี (*assumption of all-inclusiveness*) และได้เสนอกลไกบรรเทา สำหรับกรณีโครงข่ายประชาที่ยังจำแนกประเภท.

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

“ชีวิตเป็นโอกาส ใช้ประโยชน์จากมัน. ชีวิต เป็นความสวยงาม ชื่นชมมัน. ชีวิตเป็นความฝัน ทำมันให้เป็นจริง. ชีวิตเป็นความท้าทาย ต้อนรับ มัน. ชีวิตเป็นหน้าที่ ทำมันให้สมบูรณ์. ชีวิตเป็น เกมส์ เล่นมัน. ชีวิตเป็นคำสัญญา รักษา มัน. ชีวิต เป็นความเคร้า ผ่านมันให้ได้. ชีวิตเป็นเพลย ร้อง มัน. ชีวิตเป็นการดื่นرن ยอมรับมัน. ชีวิตเป็นโศก- นาฏกรรม เผชิญหน้ามัน. ชีวิตเป็นความท้าทาย กล้าผจญมัน. ชีวิตเป็นโชค ทำมันให้เกิด. ชีวิตมีค่า มาก อย่าทำลายมัน. ชีวิตคือชีวิต สู้เพื่อมัน.”

—แม่เทเรซา

แบบฝึกหัด 8.1

จากหัวข้อ 8.3 (สมการ 8.12) จงแสดงให้เห็นว่า

$$\begin{aligned} \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) \cdot \ln \left\{ p(z_1|\boldsymbol{\pi}) \cdot \left(\prod_{t=2}^T p(z_t|z_{t-1}, \mathbf{A}) \right) \cdot \prod_{\tau=1}^T p(x_{\tau}|z_{\tau}, \boldsymbol{\phi}) \right\} \\ = \sum_{k=1}^K q_{1k} \cdot \ln p(z_1|\boldsymbol{\pi}) + \sum_{j=1}^K \sum_{k=1}^K \sum_{t=2}^T R_{j,k}^{(t-1,t)} \cdot \ln p(z_t|z_{t-1}, \mathbf{A}) \\ + \sum_{k=1}^K \sum_{\tau=1}^T q_{\tau k} \cdot \ln p(x_{\tau}|z_{\tau}, \boldsymbol{\phi}) \end{aligned}$$

เมื่อ $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_0) = p(z_1, \dots, z_T|\mathbf{X}, \boldsymbol{\theta}_0)$ โดย z_t แสดงด้วยรหัสหนึ่งร้อน และ $q_{tk} \equiv p(z_{tk} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$ กับ $R_{j,k}^{(t-1,t)} \equiv p(z_{t-1,j} = 1, z_{t,k} = 1|\mathbf{X}, \boldsymbol{\theta}_0)$. สำหรับ แบบจำลองนี้ อาศัยเงื่อนไข marrow.

แบบฝึกหัด 8.2

จงแสดงให้เห็นว่า ค่าของ π_k และค่าของ A_{jk} ดังระบุในสมการ 8.13 และ 8.14 จะทำให้方程式 ชั้นควรจะเป็น (สมการ 8.12) มีค่าสูงสุด ขณะที่ยังรักษาเงื่อนไขความน่าจะเป็น $\sum_k \pi_k = 1$ และ $\sum_k A_{jk} = 1$ ไว้ได้.

คำใบ้ พังก์ชันจุดประสงค์ที่รวมเงื่อนไขแล้ว⁷ จะเป็นดังสมการ 8.39.

$$\mathcal{G}(\boldsymbol{\theta}) = \varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0) - \lambda_1 \left(\sum_k \pi_k - 1 \right) - \lambda_2 \left(\sum_k A_{jk} - 1 \right) \quad (8.39)$$

เมื่อ $\lambda_1 \geq 0$ และ $\lambda_2 \geq 0$ เป็น参数น้ำหนัก หรือรากฐานของค่าอนุพันธ์ $\partial \mathcal{G}(\boldsymbol{\theta}) / \partial \pi_k = 0$. นั่นคือ

$$\begin{aligned} \frac{\partial \varepsilon(\boldsymbol{\theta}, \boldsymbol{\theta}_0)}{\partial \pi_k} - \lambda_1 \frac{\partial (\sum_j \pi_j - 1)}{\partial \pi_k} &= 0 \\ \frac{\partial \sum_j q_{1j} \cdot \ln \pi_j}{\partial \pi_k} - \lambda_1 &= 0 \\ \frac{q_{1k}}{\pi_k} &= \lambda_1 \\ \pi_k &= \frac{q_{1k}}{\lambda_1}. \end{aligned} \quad (8.40)$$

ด้วยเงื่อนไข $\sum_j \pi_j = 1$ เมื่อแทนสมการ 8.40 ลงไปจะได้

$$\begin{aligned} \sum_j \frac{q_{1j}}{\lambda_1} &= 1 \\ \lambda_1 &= \sum_j q_{1j}. \end{aligned} \quad (8.41)$$

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 และสร้างระบบการจำแนกอัตโนมัติ พร้อมประเมินผล.

แบบฝึกหัด 8.5

จงศึกษาศาสตร์การจำลองแบบ โดยเฉพาะสำหรับข้อมูลชุดลำดับ ในเชิงกว้าง ถึงแบบจำลอง ขั้นตอนวิธี และกลไกที่เป็นศาสตร์และศิลป์ หรือคิดว่า่น่าสนใจ รวมไปถึง ปัจจัยหรือประเด็นที่ควรใส่ใจ

⁷พังก์ชันจุดประสงค์ที่รวมเงื่อนไข อาจต้องแบ่งอีก เช่น แบบที่ใช้วิธีการลงโทษ (หัวข้อ 2.3) ได้ แต่การวิเคราะห์อาจจะซับซ้อนขึ้น.

การประยุกต์ใช้เด่น ๆ และการกิต่าง ๆ และอภิปรายโอกาสการประยุกต์ใช้ต่าง ๆ และความท้าทายต่าง ๆ ในงานวิจัย แล้วสรุปและให้ความเห็น.

แบบฝึกหัด 8.6

จากแบบฝึกหัด 8.5 จะเลือกประเด็น แบบจำลอง ขั้นตอนวิธี หรือกลไก ที่สนใจ แล้วศึกษาเรื่องดังกล่าว ตั้งคำถามที่เกี่ยวข้อง ดำเนินการหาคำตอบ และสรุปผล.

หมายเหตุ การตั้งคำถาม ควรเป็นคำถามปลายเปิด ซึ่งจะนำไปสู่คำตอบที่น่าสนใจ เช่น หากสนใจโครงสร้างของแบบจำลองความจำระยะสั้น แทนที่จะตั้งคำถามว่า “หากตัดประตูลีมออกไปแล้ว แบบจำลองจะยังทำงานได้หรือไม่?” ซึ่งคำตอบจะเป็น แค่ ใช่หรือไม่ใช่. คำถามแบบนี้ ไม่น่าสนใจ. คำถามที่ดีกว่า อาจจะเป็น “ประตูลีม ช่วยการทำงานในกรณีกับข้อมูลลักษณะแบบไหน และช่วยได้มากน้อยเท่าไรในแต่ละกรณี เมื่อวัดผลโดยสมบูรณ์ และเมื่อเปรียบเทียบกับประตูลีน อีก?” ซึ่งเป็นคำถามปลายเปิด และจะนำไปสู่คำตอบที่เดาได้ยาก มีความลึก น่าสนใจ และตัวคำตอบเอง ก็จะมีประโยชน์มากกว่าด้วย. กว่าทั้ง ๆ ไป คือ หากคำถามใด สามารถตอบได้เลย โดยไม่ต้องทำการศึกษาเพิ่มเติม หรือศึกษาเพียงเล็กน้อย หรือ เดาคำตอบได้ง่าย ๆ คำถามนั้นไม่น่าสนใจ.

บทที่ 9

การรู้จักรูปแบบเชิงลำดับในโลกการประมวลผลภาษาธรรมชาติ

``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 ของภาษามนุษย์ไม่ได้อยู่ที่ความสามารถในการแปลงสารสนเทศเรื่องของคนกับสิงโต. แต่มันคือความสามารถในการแปลงสารสนเทศเกี่ยวกับสิ่งที่ไม่ได้มีอยู่จริงเลย.”

—ญาล โนอาห์ หารารี

ภาษา เป็นแบบจำลองคร่าว ๆ ของความคิด และเป็นตัวแทนที่หยาบมาก ๆ สำหรับบรรยายโลกและอธิบายความเป็นจริง. เพียงแต่ มันยังคงเป็นเครื่องมือที่ดีที่สุดอย่างหนึ่งเท่าที่เรามี สำหรับการถ่ายทอดความคิดและสื่อสารเรื่องราว. แฮร์มันน์ เอสเซอ กล่าวว่า “ทุก ๆ อย่างที่คิด และแสดงออก มาเป็นคำพูด จะลำเอียงไปข้างเดียว ครึ่งเดียวของความจริง 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 ภาษาอังกฤษ ภาษาจีน ภาษาสเปน. ภาษาธรรมชาติ¹ มีการเกิด การพัฒนา การวิวัฒนาการตามธรรมชาติ. วิวัฒนาการของภาษาเกิดจากคนจำนวนมาก และผ่านผู้คนหลายรุ่น แม้ว่าอาจมีบางครั้งที่ได้รับการควบคุม ปรับปรุง ผ่านกลุ่ม

¹ คำอธิบายในส่วนนี้ ได้รับอิทธิพลหลัก ๆ จาก [58]

คนจำนวนน้อย ๆ ที่มีอำนาจหรือที่ได้รับมอบหมายบ้าง. ส่วนภาษาโปรแกรม เป็นภาษาที่ออกแบบจากคนหรือกลุ่มคน (จำนวนไม่มาก) เพื่อใช้สั่งงานคอมพิวเตอร์. ตัวอย่างภาษาโปรแกรม เช่น ภาษาซี ภาษาซีเพลสเพลส ภาษาจาวา ภาษาอาร์ ภาษาไฟรอน. ภาษาโปรแกรม จะมีไวยากรณ์ที่ตายตัว ใช้ควบคุมโครงสร้างของคำสั่งต่าง ๆ. ขณะที่ไวยากรณ์ของภาษาธรรมชาติ มักจะยืดหยุ่น และมีข้อยกเว้นอยู่มาก.

ไวยากรณ์ (syntax) คือกฎเกณฑ์ที่เกี่ยวกับโทเค็น และโครงสร้าง. **โทเค็น (token)** เป็นหน่วยพื้นฐานของภาษาที่มีความหมาย เช่น ในภาษาธรรมชาติ โทเค็น หมายถึง คำ. ในภาษาโปรแกรม โทเค็น หมายถึง คำ, ตัวแปร, ค่าคงที่, นิพจน์, ฟังก์ชัน, ออปเจ็ค, เมท็อด เป็นต้น. โครงสร้างไวยากรณ์ คือการนำโทเค็นไปประกอบกันเพื่อสื่อความหมาย. ความสัมพันธ์ระหว่างโทเค็นและโครงสร้างไวยากรณ์ อาจแสดงได้ด้วยตัวอย่าง เช่น ในภาษาอังกฤษ “This is an English sentence.” มีการใช้โทเค็นที่ไม่ถูกต้อง. ส่วน “is.sentence ThisEnglish an” แม้จะใช้โทเค็นที่ถูกต้องทั้งหมด แต่เป็นการประกอบกันที่ไม่ถูกต้องตามไวยากรณ์ภาษาอังกฤษ. ประโยชน์คือ “This is an English sentence.” ถูกไวยากรณ์ภาษาอังกฤษ (โทเค็นถูกต้องทั้งหมด และประกอบกันเป็นโครงสร้างที่ถูกต้อง). การวิเคราะห์โครงสร้างไวยากรณ์ของข้อความหรือประโยค จะเรียกว่า **การแยกส่วน (parsing)**. เวลาที่เราอ่านข้อความต่าง ๆ เราทำการแยกส่วน เพื่อเข้าใจรูปประโยค ประกอบการทำเข้าใจความหมายของข้อความ.

นอกจากที่ภาษาโปรแกรมมีไวยากรณ์ที่ตายตัวมากกว่าภาษาธรรมชาติแล้ว ยังมีประเด็นที่แตกต่างกันดังนี้ (1) **ความกำกวມ (ambiguity)** ที่ภาษาธรรมชาติมักอาศัยบริบทและสามัญสำนึกประกอบในการทำความเข้าใจข้อความ ในขณะที่ภาษาโปรแกรม ถูกออกแบบให้มีความชัดเจนโดยสมบูรณ์ ตีความได้อย่างเดียว ไม่มีความกำกวມเลย, (2) **ความซ้ำซ้อน (redundancy)** ที่พบได้บ่อย ๆ ในภาษาธรรมชาติ แต่ภาษาโปรแกรมจะกระชับและไม่ซ้ำซ้อน, (3) **ความตรงตามตัวอักษร (literalness)** ที่ภาษาโปรแกรมบอกความหมายที่เจาะจง ตรงตามตัวอักษร ในขณะที่ภาษาธรรมชาติ มีการใช้ส่วนวน โวหาร คำเบรียบ เปรย.

ด้วยความแตกต่างระหว่างภาษาธรรมชาติและภาษาโปรแกรม ทำให้การประมวลผลภาษาธรรมชาติต้องการเครื่องมือ แนวทาง และกลไกเฉพาะ ที่นอกเหนือไปจากการยึดมานาจวิธีการต่าง ๆ ในการประมวลผลโปรแกรม.

การประมวลผลภาษาธรรมชาติ (Natural Language Processing คำย่อ NLP) เป็นศาสตร์ที่ใช้วิธีการต่าง ๆ เพื่อให้คอมพิวเตอร์สามารถนำข้อความในภาษาธรรมชาติไปประมวลผล และให้ผลลัพธ์ตามจุดประสงค์ของการกิจที่ต้องการ.

ภาระกิจของการประมวลผลภาษาธรรมชาตินั้นมีหลากหลายมาก เช่น ระบบตรวจสอบภาษา (spelling and grammar correction), ระบบช่วยจบคำอัตโนมัติ (autocompletion), ระบบตอบคำถามอัตโนมัติ (question answering system), การสรุปข้อความ (text summarization), แชทบอต (chatbot), การจำแนกอารมณ์ (sentiment classification), ระบบแปลภาษาอัตโนมัติ (machine

translation), การค้นหาเนื้อหาในเอกสาร (content searching), การตรวจสอบการลอกเลียนวรรณกรรม (plagiarism detection) และการสร้างข้อความอัตโนมัติ (text generation). ลักษณะเฉพาะของภาษาของ มีส่วนอย่างมากต่อความต้องการและความจำเป็นของการกิจต่าง ๆ เช่น ภาษาอังกฤษมีขอบเขตคำและขอบเขตประโยคที่ชัดเจน. การตัดคำ (word segmentation) และตัดประโยค (sentence segmentation) ในภาษาอังกฤษทำได้ง่ายมาก เมื่อเทียบกับภาษาไทย. ดังนั้นในขณะที่ระบบอัตโนมัติสำหรับการตัดคำและการตัดประโยคในภาษาอังกฤษมีความสมบูรณ์เต็มที่และพร้อมใช้งาน ความสามารถของการตัดคำและการตัดประโยคในภาษาไทย กลับอยู่ในระดับเริ่มต้น และยังต้องการการพัฒนาอีกมาก. การตัดคำและการตัดประโยค นอกจากจะใช้ประกอบการจัดแสดงหน้าเอกสาร (ในการตัดคำขึ้นบรรทัดใหม่) การตัดคำและการตัดประโยค จัดเป็นภาระกิจพื้นฐานของการประมวลผลภาษาธรรมชาติ ที่จะช่วยให้งานที่มีความซับซ้อนอื่น ๆ สามารถประมวลผลต่อไปได้อย่างมีประสิทธิภาพ.

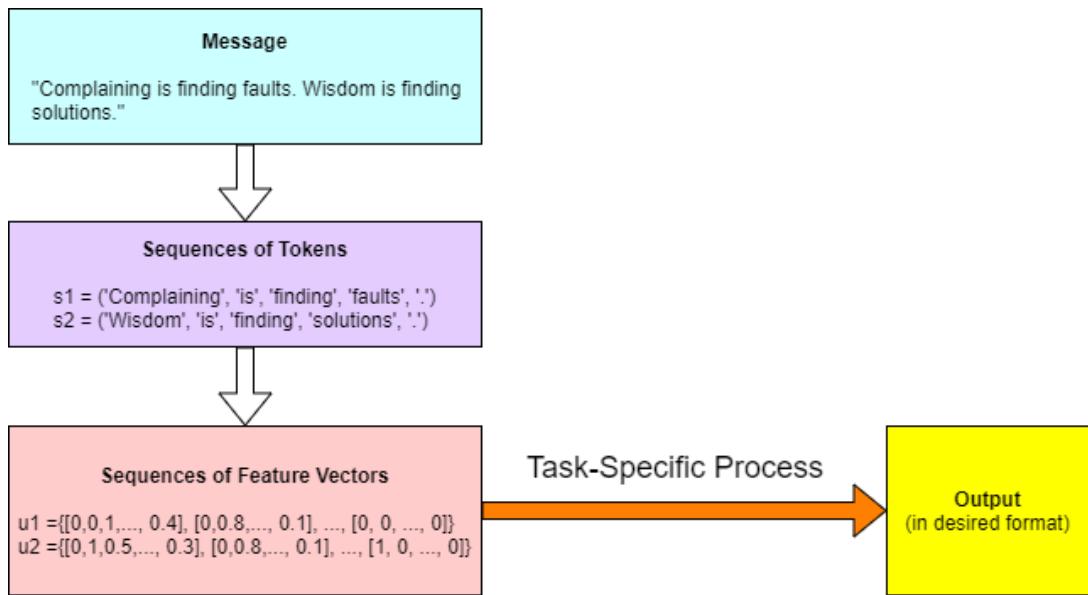
ภาพรวมของการประมวลผลภาษาธรรมชาติ โดยเฉพาะภาษาอังกฤษ² คือ อินพุตที่ข้อความ จะถูกแปลงเป็นชุดลำดับของໂທເຄີນ ซึ่งแต่ละໂທເຄີນเป็นคำ. จากนั้นแต่ละໂທເຄີນ จะถูกแปลงเป็นເວກເຕັອງລักษณะสำคัญ ซึ่งเป็นເວກເຕັອງຂອງค่าต่าง ๆ ที่เป็นตัวเลข ก่อนจะเข้ากระบวนการประมวลผลตามแต่ภาระกิจ. รูปที่ 9.1 แสดงแนวทางการประมวลผลภาษาธรรมชาติ โดยทั่วไป ที่แปลงข้อความภาษาธรรมชาติ ไปเป็นชุดลำดับต่าง ๆ ของค่าເວກເຕັອງລักษณะสำคัญ ก่อนจะเข้าประมวลผล. แนวทางเช่นนี้ ทำให้สามารถใช้แบบจำลองเชิงลำดับต่าง ๆ ที่ทำงานกับข้อมูลที่มีค่าเป็นตัวเลข มาช่วยการประมวลผล ตามแต่ภาระกิจได้.

รูปที่ 9.2 แสดงลักษณะภารกิจการระบุหมวดคำ (Part-Of-Speech Tagging) ที่รับอินพุตเป็นข้อความ (ลำดับของคำ) และให้อาต์พุต ออกมารูปเป็นลำดับของหมวดคำ โดยลำดับของอาต์พุตจะสอดคล้องกับลำดับของอินพุต. ในภาพ ข้อความอินพุตถูกแบ่งออกเป็นໂທເຄີນ ซึ่ง ณ ที่นี่ ແຕ່ລະໂທເຄີນคือคำ และอาต์พุต กີ່ເປັນຊຸດลำดับข้อมูล ທີ່ແຕ່ລະຈຸດຂໍ້ມູນຈະສອດຄລັອງກັບແຕ່ລະໂທເຄີນ.

9.2 โครงข่ายประสาทเวียนกลับ

โครงข่ายประสาทเวียนกลับ (Recurrent Neural Network คำย่อ RNN) เป็นแบบจำลองโครงข่ายประสาทเทียม ที่อินพุตของแต่ละหน่วยย่อย นอกจากจะเป็นค่าของอาต์พุตจากหน่วยย่อยในชั้นคำนวนก่อนหน้าแล้ว ยังสามารถเป็นค่าของอาต์พุตของหน่วยย่อยในชั้นคำนวนเดียวกัน สำหรับจุดข้อมูลลำดับก่อนหน้าได้. เช่นเดียวกับการวิเคราะห์การคำนวนของโครงข่ายเป็นชั้นคำนวน ดังอภิปรายในบทที่ 6 โครงข่ายประสาทเวียนกลับ ก็สามารถมองเป็นการประกอบกันของชั้นคำนวนเวียนกลับ (recurrent layer) ได้.

²ภาษาไทยมีลักษณะเฉพาะหลายอย่าง โดยเฉพาะความคลุมเครือของขอบเขตคำและขอบเขตประโยค ทำให้อาจต้องการความคิดสร้างสรรค์ และกรอบวิธีคิดใหม่ ที่ต่างจากภาพรวมที่อภิปรายนี้.



รูปที่ 9.1 ภาพรวมของการประมวลผลภาษาธรรมชาติ. กล่องแสดงตัวอย่างลักษณะของข้อมูล และลูกศรแทนกระบวนการแปลงข้อมูล. ข้อความภาษาธรรมชาติ จะผ่านขั้นตอนต่าง ๆ เพื่อแปลงไปเป็นชุดลำดับของค่าเวกเตอร์ลักษณะสำคัญ ก่อนจะเข้าประมวลผลตามภาระกิจที่ต้องการ.

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		

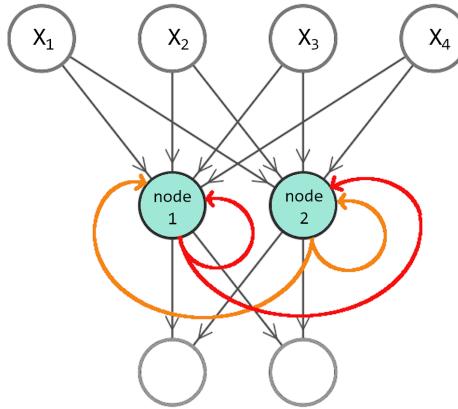
รูปที่ 9.2 ตัวอย่างอินพุตເອົາຕົກພຸດຂອງການກິຈກາຮະບູມວັດຄຳ. ອິນພຸດເປັນ ຄຳພຸດຂອງບຽນລື ແລະ ເອົາຕົກພຸດ ເປັນຕົວອ່າງ ພຶລື້ພົບຈາກຮະບາກກາຮະບູມວັດຄຳ. ບຣະທິດສຸດທ້າຍ ແສດງເຂົລຍ.

การคำนวณของหน่วยย่อในชั้นคำนวณเวียนกลับที่ q^{th} อาจเขียนได้ดังนี้

$$a_j^{(q)}(t) = \sum_{i=1}^D w_{ji}^{(q)} \cdot z_i^{(q-1)}(t) + \sum_{m=1}^M v_{jm}^{(q)} \cdot z_m^{(q)}(t-1) + b_j^{(q)} \quad (9.1)$$

$$z_j^{(q)}(t) = h(a_j^{(q)}(t)) \quad (9.2)$$

เมื่อ $a_j^{(q)}(t)$ คือค่าตัวกระตุนของหน่วยย่อที่ j^{th} สำหรับจุดข้อมูลลำดับที่ t^{th} . ตัวแปร $z_j^{(q)}(t)$ คือผลการกระตุน หรือบางครั้งอาจเรียก ว่าเป็น สถานะซ่อน ของหน่วยที่ j^{th} ในชั้นคำนวน q^{th} สำหรับจุดข้อมูลลำดับเวลา t^{th} โดย D คือจำนวนหน่วยย่อในชั้น $(q-1)^{th}$ และ M คือจำนวนหน่วยย่อในชั้น q^{th} . ตัวแปร $w_{jd}^{(q)}$ เป็นค่าน้ำหนักของการเชื่อมต่อระหว่างหน่วยที่ d^{th} ของชั้น $(q-1)^{th}$ กับหน่วยที่ j^{th} ของชั้นคำนวน q^{th} . ตัวแปร $v_{jm}^{(q)}$ เป็นค่าน้ำหนักของการเชื่อมต่อของหน่วยที่ m^{th} เวียนกลับมาเข้าหน่วยที่ j^{th} ของชั้นคำนวนเดียวกัน. ส่วน $b_j^{(q)}$ คือค่าไบอสของหน่วยที่ j^{th} และ $h(\cdot)$ คือฟังก์ชันกระตุน.



รูปที่ 9.3 ตัวอย่างโครงข่ายประสาทเวียนกลับ โดยเน้นเส้นทางข้อมูลป้อนเวียนกลับ. โครงข่ายประกอบด้วยสามชั้นคำนวณโดยชั้นอินพุต (อยู่บนสุด) มีสีหน่วย (รับอินพุตสี่มิติ ได้แก่ x_1 ถึง x_4) ชั้นที่สอง เป็นชั้นเวียนกลับ มีสองหน่วย (แสดงด้วยวงกลมสีฟ้าเขียว) และชั้นที่สาม (อยู่ล่างสุด). เส้นทางการส่งข้อมูลเวียนกลับ แสดงด้วย เส้นสีแดง (ค่าเวียนกลับจากหน่วยแรก) และเส้นสีส้ม (ค่าเวียนกลับจากหน่วยที่สอง).

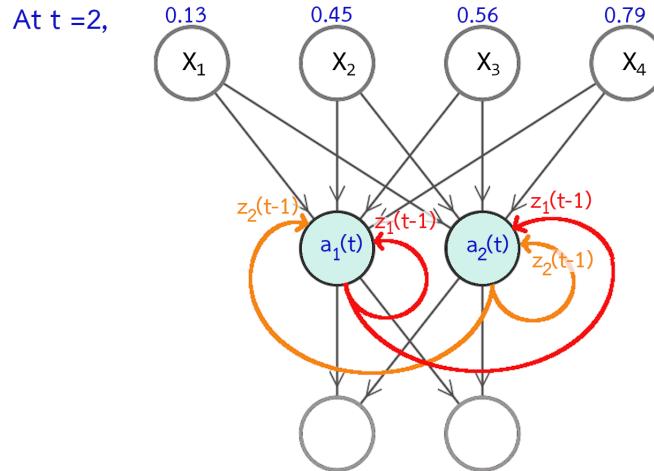
เมื่อเปรียบเทียบสมการ 9.1 กับสมการ 3.16 ซึ่งเป็นการคำนวณของโครงข่ายแพร่กระจายไปข้างหน้า จะเห็นว่าจุดต่างที่สำคัญ คือ พจน์ $\sum_{m=1}^M v_{jm}^{(q)} \cdot z_m^{(q)}(t-1)$ ซึ่งเป็นการนำผลการกระตุ้นที่ลำดับเวลา ก่อนเข้ามาคำนวณด้วย. รูปที่ 9.3 แสดงตัวอย่างโครงสร้างการเข้ามาร่วมต่อของโครงข่ายประสาทเวียนกลับ ที่อินพุตมีสี่มิติ และเอาต์พุตมีสองมิติ โดยชั้นคำนวณที่สอง ซึ่งเป็นชั้นเวียนกลับ มีหน่วยย่อยสองหน่วย. รูปที่ 9.4 แสดงตัวอย่างโครงข่ายประสาทเวียนกลับ พิริมตัวอย่างชุดข้อมูลลำดับ และตัวแปรที่สำคัญ.

รูปที่ 9.3 แสดงโครงข่ายประสาทเวียนกลับ โดยเน้นการแสดงโครงสร้าง. อย่างไรก็ตาม หากชั้นเวียนกลับมีจำนวนหน่วยมาก ๆ การเขียนแผนภาพเช่นนี้ จะดูยุ่งเหยิงมาก (แต่ละหน่วยส่งค่าเวียนกลับไปให้ทุก ๆ หน่วยในชั้น). บ่อยครั้ง แผนภาพโครงข่ายประสาทเวียนกลับ จึงมักถูกแสดงโดยใช้วงกลมแค่หนึ่งวงแทนชั้นคำนวณทั้งชั้น (ไม่ว่าภายในชั้นจะใช้จำนวนหน่วยเท่าใด) ดังแสดงในรูปที่ 9.5. นอกจากนั้น ในบางสถานการณ์ การใช้แผนภาพคลี่ลำดับ (unfolding diagram) ที่แสดงข้อมูลการเวียนกลับ ด้วยการกระจายออกตามลำดับเวลา อาจช่วยให้เข้าใจแนวคิดได้ดีกว่า. แผนภาพคลี่ลำดับอาจแสดงดังรูปที่ 9.6.

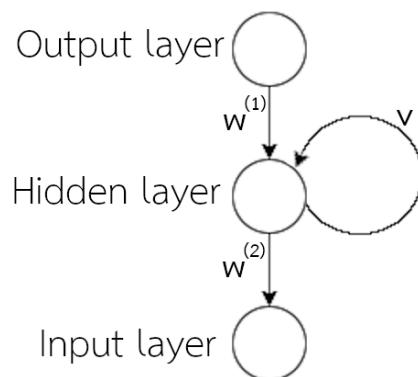
จากแผนภาพคลี่ลำดับ ในรูปที่ 9.6 สังเกต (1) ทุก ๆ ลำดับเวลา การคำนวณใช้ค่าน้ำหนักชุดเดียวกัน (ที่เวลา t ต่าง ๆ ใช้ค่า $w^{(1)}, w^{(2)}$ และ v เมื่อนั้น) (2) ผลการกระตุ้นของจุดข้อมูลลำดับเวลาใด ๆ $z(t)$ จะส่งผลต่อเอาต์พุตผ่านหลายเส้นทาง (เส้นทางตรง ส่งผลต่อ $y(t)$ และเส้นทางเวียนกลับเอาต์พุตอื่น ๆ หลังจากลำดับเวลาันนั้น ๆ ผ่านเส้นทางการเวียนกลับ).

เกรเดียนต์ของชั้นเวียนกลับ. ในลักษณะเดียวกับโครงข่ายแพร่กระจายไปข้างหน้าและโครงข่ายคอนโวโลชัน การฝึกโครงข่ายประสาทเวียนกลับ สามารถทำได้โดยปรับค่าน้ำหนักต่าง ๆ โดยอาศัยการแพร่กระจายย้อนกลับ ซึ่งทำการคำนวณค่าเกรเดียนต์เป็นชั้น ๆ.

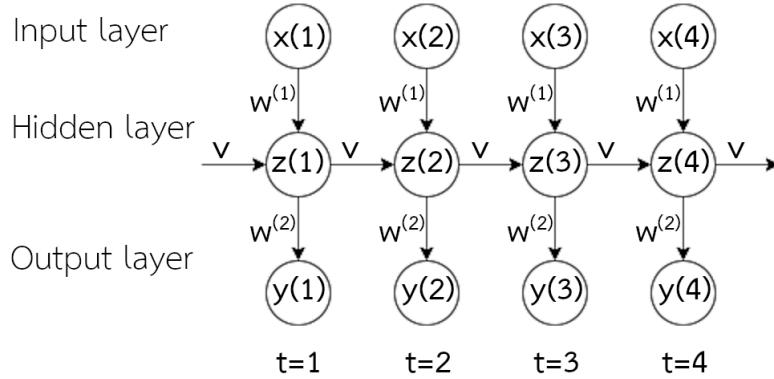
input sequence = { $\underbrace{[0.8, 0.2, 0.1, 0]}_{\mathbf{x}(t=1)}, \underbrace{[0.13, 0.45, 0.56, 0.79]}_{\mathbf{x}(t=2)}, \underbrace{[-0.28, 1.12, 2.3, 1.11]}_{\mathbf{x}(t=3)}, \dots \}$



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 สำหรับค่าหนักการแพร่กระจายไปข้างหน้า ของชั้นคำนวณที่หนึ่ง กับของชั้นคำนวณที่สอง และค่าน้ำหนักเวียนกลับ (ของชั้นคำนวณที่หนึ่ง แต่ตัวยกฤกษ์ไว้ เพื่อความกระชับ).



รูปที่ 9.6 แผนภาพคลื่นลำดับของโครงข่ายประสาทเวียนกลับ.

การแพร่กระจายย้อนกลับ สำหรับชั้นคำนวนเวียนกลับ สามารถทำได้หลาย ๆ วิธี[78] ไม่ว่าจะเป็น การเรียนรู้เวียนกลับเวลาจริง (real time recurrent learning[167]) หรือการแพร่กระจายย้อนกลับผ่านเวลา (backpropagation through time[218, 215] ค่าย่อ BPTT) โดย เกรฟซ์[78] ให้ความเห็นว่า การแพร่กระจายย้อนกลับผ่านเวลา เข้าใจได้ง่ายกว่า และสามารถคำนวนได้อย่างมีประสิทธิภาพมากกว่า. เกรเดียนต์ของชั้นเวียนกลับ ดังที่จะอธิบายต่อไปนี้ ใช้แนวทางของการแพร่กระจายย้อนกลับผ่านเวลา เช่นเดียวกับการอธิบายของเกรฟซ์[78].

ทำงานเดียวกัน กำหนดให้ E เป็นฟังก์ชันค่าผิดพลาด 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 (ผ่านชั้นคำนวนต่อไป) และ (2) ผ่านการเวียนกลับ (ผ่านชั้นคำนวนเดิม แต่สำหรับลำดับเวลาถัดไป). ดังนั้น ด้วยกฎลูกโซ่ เราจะได้

$$\begin{aligned} \frac{\partial E}{\partial z_j^{(q)}(t)} &= \sum_k \frac{\partial L}{\partial a_k^{(q+1)}(t)} \cdot \frac{\partial a_k^{(q+1)}(t)}{\partial z_j^{(q)}(t)} + \sum_m \frac{\partial L}{\partial a_m^{(q)}(t+1)} \cdot \frac{\partial a_m^{(q)}(t+1)}{\partial z_j^{(q)}(t)} \\ &= \sum_k \delta_k^{(q+1)}(t) \cdot w_{kj}^{(q+1)} + \sum_m \delta_m^{(q)}(t+1) \cdot v_{mj}^{(q)}. \end{aligned} \quad (9.5)$$

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 เมื่อพิจารณาเกรเดียนต์ต่อค่าน้ำหนักต่าง 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

$$\frac{\partial E}{\partial v_{jm}^{(q)}} = \sum_t \frac{\partial E}{\partial a_j^{(q)}(t)} \cdot \frac{\partial a_j(t)^{(q)}}{\partial v_{jm}^{(q)}} \quad (9.10)$$

$$= \sum_t \delta_j^{(q)}(t) \cdot z_m^{(q)}(t-1) \quad (9.11)$$

เช่นเดียวกับค่าน้ำหนัก ค่าไบอส $b_j^{(q)}$ 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 พังก์ชันค่าผิดพลาด 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} . หากลักษณะภาระกิจถูกตีกรอบเป็นการหาค่าลดตอนย เรายาจะกำหนด

$$E_k(t) = \frac{\mathcal{M}(t)}{2} \cdot (\hat{y}_k(t) - y_k(t))^2 \quad (9.14)$$

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 แต่สำหรับบริบทของการอนุมานข้อมูลเชิงลำดับกลไกนี้มีความสำคัญอย่างมาก. ภาระกิจการอนุมานข้อมูลเชิงลำดับ มีหลากหลายประเภท (หัวข้อ 8.1). ภาระกิจบางประเภท อาจต้องการการนำมายค่าสำหรับทุก ๆ ลำดับเวลา (เช่น ระบบตรวจสอบการshedule ที่ต้องให้ค่านำมายค่าสำหรับทุกห้องผู้พิเศษ) ภาระกิจบางประเภท อาจต้องการการนำมายค่า แค่บางลำดับเวลา (เช่น การจำแนกอารมณ์ ที่อาจจะให้ค่านำมายค่าของมาเฉพาะที่ดัชนีลำดับสุดท้ายเท่านั้น) การใช้กลไกหน้ากาก ช่วยกำหนดดัชนีลำดับที่มีผลจริง ๆ ($\mathcal{M}(t) = 1$ เฉพาะดัชนีลำดับ t ที่มีค่าเฉลย ส่วนอกนั้นให้ $\mathcal{M}(t) = 0$) จะช่วยให้การทำงานกับข้อมูลลำดับมีดียุ่งและสะดวกมากขึ้น.

เมื่อพิจารณากราฟเดินต์ ด้วยสมการ 9.13 และ 9.14 สำหรับ กรณีการหาค่าลดลง 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 เมโล

เกร็ดความรู้ ความเมตตา. ปัญญาและเมตตาเป็นคุณค่าสูงสุดของมนุษย์. ปัญญา คือ ความรู้ในเรื่องร่างกายตามความเป็นจริง ครอบคลุมถึงความสามารถในการคิด วิเคราะห์ สังเคราะห์ แก้ปัญหา พัฒนา ตระหนักรู้ โดยใช้ความรู้ ประสบการณ์ ความเข้าใจ สามัญสำนึก และมุมมองที่หลากหลายครบถ้วน. แฮร์มันน์ เฮสเซอ กล่าวว่า “ความรู้สามารถสื่อสารกันได้ แต่ไม่ใช่ปัญญา. เราหาปัญญาได้ เราใช้ชีวิตอยู่กับปัญญาได้ เราป้องกันตัวเองจากภัยนตรายด้วยปัญญาได้ เราทำสิ่งมหัศจรรย์ด้วยปัญญาได้ แต่เราสื่อสารปัญญาออกไปไม่ได้ 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.”) ผู้คนและสังคมชื่นชมและยกย่องปัญญา แม้หลายครั้งอาจจะสับสนระหว่างปัญญา ความรู้ และความฉลาด.

เมตตา คือ ความปรารถนาให้ชีวิตต่าง ๆ เป็นสุข ซึ่งรวมทั้งชีวิตสัตว์ ชีวิตคนอื่น และชีวิตของตัวเราเองด้วย. ในความหมายกว้าง ๆ แล้ว ความหมายของเมตตา ยังครอบคลุมไปถึงความปรารถนาให้ชีวิตพัฒนา (กรุณ), ความยินดีเมื่อชีวิตเป็นสุข (มุทิตา) และในบางครั้งก็อาจหมายรวมถึงการปล่อยวาง (อุเบกขา) ด้วย. สำหรับเมตตาแล้ว แม้จะเป็นหนึ่งในสองคุณค่าสูงสุดคู่กับปัญญา แต่สังคมดูเหมือนจะชื่นชมและยกย่องเมตตาบันอย่างเกินไป โดยเฉพาะเมื่อเปรียบเทียบกับระดับการยกย่องปัญญา. (ดูจากปรัชญาของโรงเรียนและมหาวิทยาลัยต่าง ๆ ทั้งในและต่างประเทศ เป็นตัวอย่าง.)

หมายเหตุ คุณธรรม นั้นอ้างถึงความดี ซึ่งครอบคลุมความหมายกว้าง ๆ และบ่อຍครั้งที่ถูกตีความผ่านค่านิยมของสังคมหรือกลุ่มคน. แม้บางครั้งอาจมองว่า คุณธรรมครอบคลุมถึงความเมตตาด้วย แต่เนื่องจากคุณธรรมถูกตีความผ่านค่า

นิยมของสังคม ความหมายของคุณธรรมจึงขึ้นกับบริบทเป็นอย่างมาก. ตัวอย่างเช่น[217, 159] คุณธรรมตามค่านิยมของกรีกโบราณ ตามแนวคิดของเพลโต คือ ความรอบคอบ, ความกล้าหาญ, การรู้จักระบับยังไง และความยุติธรรม. คุณธรรมตามค่านิยมของชาวนิโธ (บุชido) คือ ความซื่อสัตย์และยุติธรรม, ความกล้าหาญ, ความเมตตา, ความเคารพให้เกียรติกันและกัน, สัจจะว่าจ่า, เกียรติและศักดิ์ศรี, หน้าที่และความภักดี และการระงับอารมณ์ การควบคุมตัวเอง. คุณธรรมตามค่านิยมจีนดั้งเดิม คือ ความเมตตา, ความประยัคต์มั่นคง, ความอ่อนน้อมถ่อมตน และความกดดัน. คุณธรรมแก่นตามแนวคิดจิตวิทยาคุกใหม่ คือ ปัญญาและความรู้ (ความอยากรู้อยากเห็น, ความคิดสร้างสรรค์, การเปิดกว้างทางความคิด, การรักที่จะเรียนรู้ และการมีมุ่งมองที่หลากหลาย), ความกล้าหาญ (ความอาจหาญในการเผชิญความเสี่ยงหรืออันตราย, ความมุ่ມานะอุตสาหะ, ความซื่อสัตย์มั่นคง และความกระตือรือร้น), มนุษยธรรม (ความรัก, เมตตา และความฉลาดทางสังคม), ความเป็นธรรม (ความรับผิดชอบทางสังคม, ความยุติธรรม และความเป็นผู้นำ), การควบคุมอารมณ์ (การให้อภัย, ความอ่อนน้อมถ่อมตน, ความรอบคอบ และการควบคุมตนเอง) และอุต্তरภาพ (การเข้มแข็งในความงามของสิ่งรอบตัวและความดีของผู้คน, ความสำนึกเห็นค่าและรู้คุณ, ความหวัง, อารมณ์ขันและความขี้เล่น และศรัทธาหรือความแกร่งทางจิตวิญญาณ)

การจะพัฒนาปัญญาของนั้น ถ้าหากขาดเมตตาแล้ว ปัญญาจะพัฒนาไปได้อย่างจำกัดมาก (หากจะยังพัฒนาต่อไปได้) เพราะความรู้ในเรื่องราวดามความเป็นจริง จะสมบูรณ์ได้อย่างไร หากขาดความเห็นใจเข้าใจชีวิตอื่น. นอกจากนั้น เนื่องเดียว กับที่ ผลป้อนกลับลบ (negative feedback) จะช่วยให้ระบบทางวิศวกรรมมีเสถียรภาพที่ดี และทนทานต่อสภาพการใช้งานที่หลากหลายมากกว่า เมตตาเป็นเสมือนกับกลไกผลป้อนกลับของชีวิต. ลองจินตนาการถูว่า หากเราเป็นผู้น้อยอ่อนประสาการณ์ ผู้คนสามารถว่ากล่าวตักเตือน ให้คำแนะนำกับเราได้. แต่หากเราเป็นผู้ยิ่งใหญ่ที่สูงด้วยวัยรุ่น ด้วยคุณภาพ ด้วยชื่อเสียง ด้วยเงินทอง ด้วยอำนาจ โดยไม่มีเกณฑ์ใดที่บังคับให้เราต้องฟังใคร และหากไม่มีความจำเป็นต้องฟังใคร จะมีอะไรที่ทำให้เราต้องฟังคนอื่น? ณ ตอนนั้น มีเพียงความเมตตาความเห็นอกเห็นใจเท่านั้น ที่จะเป็นเสมือนช่องทางที่ยังจะเปิดรับฟังอยู่เสมอ ไม่ว่าซองทางอื่น ๆ อาจจะถูกปิดไปแล้ว ปิดไปด้วยความสูงส่งของอำนาจ เกียรติยศ ศักดิ์ศรี ชื่อเสียง เงินทอง. เราต้องการกลไกผลป้อนกลับลบ เพื่อเสถียรภาพที่ดีของสังคมและของตัวเราเอง เพื่อที่จะยังสามารถรับฟังคำตักเตือน คำแนะนำ ความเห็นต่าง ๆ ได้อยู่เสมอ. บ่อยครั้งที่เมตตาอาจช่วยให้ เราสามารถสังเกตและรับรู้ถึงความรู้สึกของผู้คนได้ ก่อนที่เขาจะต้องเอ่ยปากด้วยคำ.

“Kindness in words creates confidence.

Kindness in thinking creates profoundness.

Kindness in giving creates love.”

--Lao Tzu

“ความเมตตาในคำพูด สร้างความมั่นใจ.

ความเมตตาในความคิด สร้างความลึกซึ้ง.

ความเมตตาในการให้ สร้างความรัก”

—เล่าจื้อ

การขาดเมตตา ไม่ได้มีผลเฉพาะแค่ต่อการจำกัดปัญญา, ต่อการขาดระบบป้อนกลับ และต่อการลดประสิทธิภาพในการสื่อสารเท่านั้น. เมตตาเป็นกลไกสำคัญในการลดและควบคุมอัตตา. อัตตา (ego) หรือ มโนคติของตัวตน (concept of self) เป็นแนวโน้มและพฤติกรรมการยึดติดกับสิ่งที่จิตใช้เป็นตัวแทนของตัวตน เป็นการยึดติดในความรู้สึกเป็นเจ้าของ. จากล่าวโดยรวมได้ว่า เรายุคคนมีอัตตาอยู่ (ยกเว้นบุคคล เช่น อริยบุคคล ซึ่งเป็นผู้ไม่มีอัตตา) เพียงแต่ว่า โดยส่วนใหญ่แล้ว ขนาดของอัตตาของเรามาได้ให้ใหญ่จนรบกวนการดำเนินชีวิตมากจนเกินไป. อย่างไรก็ตาม คนบางคนอาจมีอัตตาที่ใหญ่มาก ๆ และอาจใหญ่มากจนเข้าข่ายของโรคหลงตัวเอง.

โรคหลงตัวเอง (ความผิดปกติทางบุคคลิกภาพแบบหลงตัวเอง ซึ่งภาษาอังกฤษคือ Narcissistic Personality Disorder คำย่อ NPD. เนื้อหาหลัก ๆ ในส่วนนี้ เรียบเรียงจาก [41]) คือ สภาพจิต ที่ผู้ป่วยรู้สึกว่าตัวเองเป็นคนสำคัญมาก, ชอบให้คนมาสนใจและชื่นชมมาก ๆ, มีปัญหาความสัมพันธ์กับคนในครอบครัว และขาดความเห็นอกเห็นใจผู้อื่น. ภายนอก ผู้ป่วยอาจดูเป็นคนที่มีความมั่นใจในตัวเองสูงมาก แต่ภายในแล้ว ผู้ป่วยมีความนับถือตัวเองในระดับที่ประบางมาก และทนไม่ได้กับการถูกวิพากษ์วิจารณ์.

สัญญาณและการของโรค ได้แก่ คิดว่าตัวเองสำคัญมาก (มากเกินกว่าความเป็นจริง), คิดว่าตัวเองสมควรจะถูกยกย่อง และต้องการถูกชื่นชมอยู่ตลอดเวลา, คิดว่าตัวเองต้องถูกยอมรับว่าเหนือกว่าคนอื่น ๆ โดยไม่ได้มีหลักฐานรูปธรรมรองรับ, โ้อวัดความสำเร็จ พรสวรรค์ และความสามารถ, หมกมุ่นและผึ่งเพื่อกับการประสบความสำเร็จ อำนาจ ความเฉลี่ยวฉลาด ความ爽快 หรือคุ้ครองที่สมบูรณ์แบบ, เชื่อว่าตัวเองดีกว่าคนอื่น ๆ และควรจะได้คุบหาสมาคมกับคนพิเศษในระดับเดียวกัน, จงพูดอยู่คนเดียวในวงสนทน และการถูกถูกคนอื่นที่คิดว่าต่ำต้อยกว่า, เอาเปรียบคนอื่น เพื่อให้ได้สิ่งที่ตนต้องการ, ไม่สามารถหรือไม่ยอมที่จะรับรู้ถึงความต้องการหรือความรู้สึกของคนอื่น, อิจฉาคนอื่น ๆ หรือคิดว่าคนอื่น ๆ อิจฉาตัวเอง, ก้าวร้าว หรือหยิ่งโส ดูไม่จริงใจ ขี้มี แลงเสแรง, ยืนกรานที่จะได้สิ่งที่ดีที่สุด เช่น รถที่ดีที่สุด ที่ทำงานที่ดีที่สุด, ไม่สามารถยอมรับการถูกวิพากษ์วิจารณ์ได้, หงุดหงิดหรือโกรธ หากไม่ได้รับการต้อนรับปฏิบัติเป็นพิเศษ, มีปัญหาการควบคุมอารมณ์, มีปัญหาการจัดการกับความเครียด, มีปัญหาการปรับตัวกับการเปลี่ยนแปลง, รู้สึกเครียดและไม่สงบอารมณ์ เวลาไม่ได้ดังใจ และแอบรู้สึกว่าไม่มั่นคง อ่อนแอ อาย อดสูญหายหน้า.

ผู้ป่วยโรคหลงตัวเอง นอกจากสร้างความทุกข์ให้กับคนอื่นแล้ว โรคอาจส่งผลกระทบกับตัวผู้ป่วยเอง ได้แก่ ปัญหาความสัมพันธ์ในครอบครัว, ปัญหาที่โรงเรียน หรือที่ทำงาน, ปัญหาภาวะซึมเศร้าและวิตกกังวล, ปัญหาสุขภาพทางกาย, ปัญหาการใช้ยาเสพติด หรือการดื่มสุรา และพฤติกรรมการฆ่าตัวตาย. คำแนะนำจากเมโยคลินิก[41] สำหรับผู้ป่วยโรคหลงตัวเอง คือ การเข้าพบแพทย์. แต่โดยส่วนใหญ่แล้ว ผู้มีความผิดปกติทางบุคลิกภาพ รวมถึงผู้ป่วยโรคหลงตัวเอง มักไม่คิดว่าตัวเองป่วย และมักไม่ยอมเข้ารับการรักษา.

การลดอัตตา. ผู้ที่ป่วยแล้ว การเข้าพบแพทย์น่าจะดีที่สุด แต่สำหรับ คนทั่วไป ที่อาจต้องการลดหรือควบคุมอัตตา อาจทำได้ด้วยการพัฒนาเมตตาขึ้น. การพัฒนาเมตตา อาจทำโดย ฝึกให้อภัยคนอื่น ให้อภัยตัวเอง และปล่อยวางบ้าง, ฝึกยอมรับความจริง ฝึกพูดความจริง และฝึกที่จะเปิดใจกว้างยอมรับความคิดความเห็นที่หลากหลาย, ฝึกยอมรับความผิดของตัวเอง, ฝึกลดหรือละความรู้สึกที่จะควบคุมทุกสิ่งทุกอย่างลง, หาเวลาอยู่เงียบ ๆ สงบ ๆ คนเดียวบ้าง, ฝึกชื่นชมความงามของสิ่งรอบตัว และมองเห็นความดีของคนอื่น ๆ, ฝึกระลึกถึงบุญคุณหรือสิ่งดี ๆ ที่คนอื่น ๆ ทำให้เรา, ฝึกช่วยเหลือคนอื่นบ้าง, ฝึกทำดีกับคนและหน้าบ้าง, ลองเป็นจิตอาสาบ้าง, ฝึกพูดสิ่งดี ๆ ให้กำลังใจคนอื่น, ลด ละ เลิกการวิจารณ์คนอื่นและการเบรียบเที่ยบคน, ฝึกที่จะไม่บ่น ไม่เสียดสี ไม่ประชดประชัน, ฝึกมองโลกในแง่ดี, ฝึกทักษะพูดอย่างยิ้มเย้มแจ่มใส, ฝึกที่จะช่วยคนที่เดือดร้อนบ้าง หากมีโอกาส, ฝึกที่จะถ่อมตัว, ฝึกที่จะไม่พูดโอ้อวด รวมถึงลดหรือเลิกการโอ้อวด ผ่านสื่อสังคมออนไลน์, ฝึกที่จะปล่อยให้คนอื่นได้รับความสนใจ ได้รับการชื่นชม, ฝึกสามารถอย่างสม่ำเสมอ, แผ่เมตตาหรืออวยพรให้สรรพชีวิตอย่างสม่ำเสมอ, แผ่เมตตาให้กับคนที่เราไม่ชอบหรือคนที่เราโกรธ, พยายามมีสติรู้สึกอารมณ์ที่เข้ามาในใจ, พยายามควบคุมอารมณ์ และศึกษาพัฒนาตนเองด้านจิตวิญญาณบ้าง.

อัตตา มีลักษณะที่แพลง. นั่นคือ ถ้าเราขอบคิดว่า เราดีกว่าคนอื่น นี่คืออัตตาสูง. ถ้าเราขอบคิดว่า เราแย่กว่าคนอื่น นี่ก็คืออัตตาสูง. ตราชที่เรายังคงมุ่งกับตัวเราเป็นสำคัญ นั่นคืออัตตาสูง. สิ่งที่จะลดอัตตาได้ คือเมตตา (ภาพของสรรพชีวิตมีความสุข เราอาจจะยังอยู่ในภาพ แต่ไม่ได้เด่นอีกต่อไปแล้ว).

ไม่ได้รักษาความผิดปกติ แต่ดูแลส่วนที่ปกติ. สำหรับการรักษาผู้ป่วยอาการจิตเวช มีเรื่องเล่าที่น่าสนใจจากอาจารย์พรหม (Ajahn Brahm) ซึ่งเป็นพระนักเทคโนโลยี นักบรรยาย และนักเขียนที่ได้รับการยอมรับนับถืออย่างกว้างขวาง ที่ท่านเคยถามเจ้าหน้าที่ในโรงพยาบาลจิตเวชแห่งหนึ่งว่า เขา.rักษาความผิดปกติทางจิตอย่างไร เจ้าหน้าที่ตอบว่า เขายังไม่ได้รักษาส่วนที่ผิดปกติ เขายังรักษาส่วนที่ดี.

ผู้ป่วยจิตเวช ไม่ได้แสดงอาการผิดปกติอย่างมาตลอดเวลา. ผู้ป่วยหลายคน ส่วนใหญ่ก็ปกติดี เพียงแค่มีช่วงเวลาที่เกิดอาการผิดปกติทางจิตขึ้นมาเท่านั้น. สิ่งที่เจ้าหน้าที่จิตเวชทำ คือ พยายามรักษา ส่งเสริม ดูแล ให้ช่วงเวลาที่ดีอยู่ได้ยาวนานขึ้น ดูแลให้ส่วนที่ปกติเติบโตขึ้น แล้วช่วงเวลาที่ผู้ป่วยเป็นปกติ จะยาวนานขึ้น และทำให้ช่วงเวลาผิดปกติสั้นลงไปเอง. ความปกติถูกดูแล ถูกให้ความสำคัญ จนมันอยู่ได้นานขึ้น แข็งแรงมากขึ้น ส่วนความผิดปกติจะเกิดน้อยลงและเบาลงเอง. แนวทางนี้ไม่ใช่ได้เฉพาะกับผู้ป่วยจิตเวชหรอก ในตัวคนเรา ในชุมชน หรือในสังคมก็เช่นกัน 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 ใส่ใจกับมัน 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 การฝึกโครงข่ายประสาทเวียนกลับยังมีปัญหาการเลื่อนหายของเกรเดียนต์ และปัญหาการระเบิดของเกรเดียนต์. การเวียนกลับย้อนลำดับเวลา ให้ผลคล้ายการแพร่กระจายย้อนกลับผ่านชั้นคำนวณต่าง ๆ ของโครงข่ายประสาทเชิงลึก (ดูแผนภาพคลื่นลำดับ เช่น รูปที่ 9.6 ประกอบ) แต่จุดต่างที่สำคัญคือ เมื่อย้อนกลับผ่านชั้นคำนวณ ค่าน้ำหนักของชั้นคำนวณแต่ละชั้น เป็นอิสระต่อกัน แต่เมื่อย้อนกลับผ่านลำดับเวลา ค่าน้ำหนักที่ลำดับเวลาต่าง ๆ เป็นชุดเดียวกัน. กลไกการเวียนกลับ ส่งผลต่อเสถียรภาพของการคำนวณค่าเกรเดียนต์ ซึ่งบางครั้งเกิดปัญหาในลักษณะการเลื่อนหายของเกรเดียนต์ ที่เกรเดียนต์มีค่าลดลงอย่างมาก เมื่อเวียนกลับย้อนลำดับเวลา จนไม่สามารถเชื่อมโยงความสัมพันธ์ระยะยาวได้. แต่บางครั้ง การฝึกโครงข่ายประสาทเวียนกลับ อาจเห็นปัญหาในลักษณะของการระเบิดของเกรเดียนต์. **ปัญหาการระเบิดของเกรเดียนต์ (exploding gradient problem)** ที่พบกับการฝึกโครงข่ายประสาทเวียนกลับ คือ การที่เกรเดียนต์ มีค่าเพิ่มขึ้นอย่างมาก เมื่อเวียนกลับย้อนลำดับเวลา จนทำให้การคำนวณเสียเสถียรภาพ และการฝึกล้มเหลวในที่สุด.

ปัญหาการเลื่อนหายของเกรเดียนต์ ในโครงข่ายประสาทเวียนกลับ สามารถบรรเทาลงได้ด้วย กลไกต่าง ๆ เช่นที่เป็นส่วนประกอบของแบบจำลองความจำระยะลื้นที่ยาว (หัวข้อ 9.4). ส่วนปัญหาการระเบิดของเกรเดียนต์ สามารถบรรเทาลงได้ด้วย ๆ ด้วยการเลิมเกรเดียนต์.

การเลิมเกรเดียนต์ (gradient clipping) เป็น กลไกง่ายในการลดขนาดเกรเดียนต์ลง ให้อยู่ในระดับที่การคำนวณจะยังสามารถทำต่อไปได้โดยมีเสถียรภาพ. พาสคานูและคณะ[151] ปรับขนาดของ

เกรเดียนต์ลงให้ไม่เกินค่าที่กำหนด 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 τ คือ ค่าขีด界ที่กำหนด. ค่าขีด界 τ สามารถเลือกได้ง่าย ๆ เพียงเป็นค่าที่ไม่มากเกินไปที่จะทำให้ระบบเสียเสถียรภาพเท่านั้น เช่น อาจเลือกให้ $\tau = 1$ เมื่ออนที่พารามิเตอร์และคณะใช้ในการทดลองก็ได้.

พารามิเตอร์และคณะ ใช้วิธีปรับลงขนาดของเกรเดียนต์ทั้งเวคเตอร์ ทำให้แม้ลดขนาดของเวคเตอร์ลงแต่ทิศทางของเกรเดียนต์ยังคงเดิม.

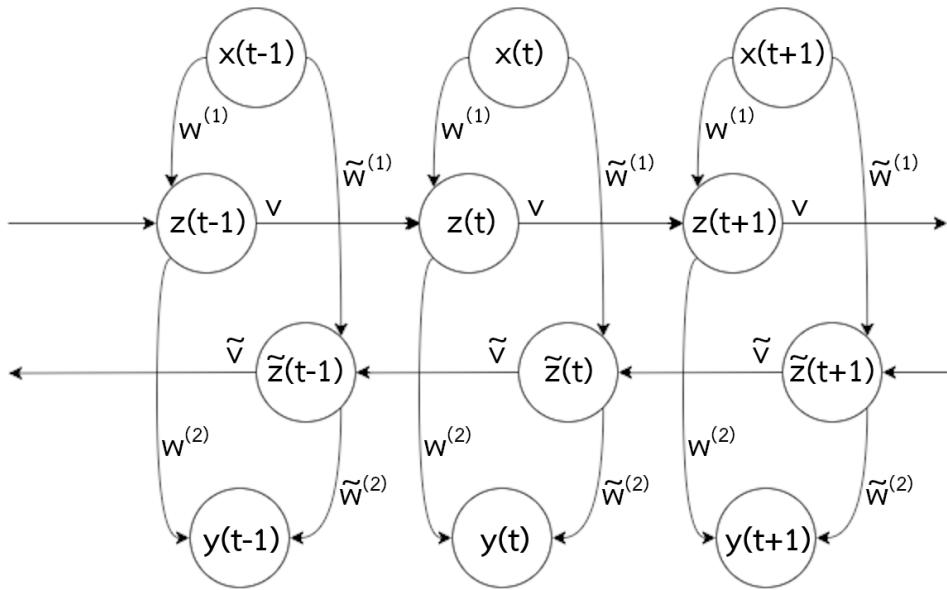
9.3 โครงข่ายประสาทเวียนกลับสองทาง

โครงข่ายประสาทเวียนกลับ นำจุดข้อมูลลำดับก่อนหน้ามาร่วมพิจารณาผลการทำนายที่ลำดับเวลาปัจจุบัน ช่วยให้เราสามารถสร้างแบบจำลองความสัมพันธ์ของจุดข้อมูลลำดับปัจจุบัน กับจุดข้อมูลต่าง ๆ ในลำดับก่อนหน้าได้. อย่างไรก็ตาม การกิจกับข้อมูลเชิงลำดับหลายอย่าง อาจต้องการจำลองความสัมพันธ์ระหว่างจุดข้อมูลลำดับปัจจุบันกับจุดข้อมูลในลำดับหลัง ๆ เช่น กรณีภารกิจการระบุหมวดคำในรูปที่ 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" ทำหน้าที่เป็นสรรพนามแต่ถ้าสำหรับ "... if one day you can let it go." คำว่า "one" ทำหน้าที่เป็นตัวเลข. โครงข่ายประสาทเวียนกลับ ที่อาศัยเฉพาะแต่ความสัมพันธ์กับลำดับที่ผ่านมา ไม่อาจแก้ปัญหาลักษณะนี้ได.

วิธีบรรเทาปัญหาลักษณะนี้อย่างง่าย ๆ คือ การใช้กลไกหน้าต่างเวลา (time-window) ที่จับกลุ่มトイเค็นหลาย ๆ トイเค็นรวมกันเป็นจุดข้อมูลแต่ละจุด สำหรับโครงข่ายประสาทเวียนกลับ. อย่างไรก็ตาม แนวทางการใช้กลไกหน้าต่างเวลาที่อาศัยกรอบหน้าต่างเวลา ที่มีความยาวคงที่ ทำให้จำกัดความสัมพันธ์ระยะยาวระหว่างจุดข้อมูล. อีกแนวทางง่าย ๆ คือ การหน่วงเวลาระหว่างจุดข้อมูลลำดับของอินพุต กับจุดข้อมูลลำดับของเอาต์พุต แต่แนวทางนี้ยังต้องอาศัยการเลือกระยะเวลาหน่วงที่เหมาะสม.

แนวทางหนึ่ง ที่ถูกออกแบบและพบร่วมกับ[78] ใช้ได้ สำหรับกรณีเช่นนี้ คือ โครงข่ายประสาทเวียนกลับสองทาง (bidirectional recurrent neural networks[180]). กลไกที่สำคัญของโครงข่ายประสาทเวียนกลับสองทาง คือ เพิ่มชั้นคำนวนเวียนกลับที่รับชุดลำดับที่เรียงกลับหลัง. การคำนวนเอาต์พุตสุดท้ายของโครงข่าย จะรอนจนกว่าชั้นคำนวนเวียนกลับ (ทั้งชั้นที่รับชุดลำดับเรียงหน้าไปหลัง และชั้นที่รับชุดลำดับเรียงหลังไปหน้า) จะได้ประมาณผลครบถ้วนจุดข้อมูลในชุดลำดับก่อน.

รูปที่ 9.7 แสดงแผนภาพคลื่นลำดับของโครงข่ายประสาทเวียนกลับสองทาง. การคำนวนของหน่วย



รูปที่ 9.7 แผนภาพคลื่นลำดับของโครงข่ายประสาทเวียนกลับสองทาง.

ย่อในชั้นคำนวณเวียนกลับทิศทางย้อนกลับที่ 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 ของหน่วยย่อที่ j^{th} ในชั้นคำนวณ q^{th} สำหรับจุดข้อมูลลำดับที่ t^{th} โดย \tilde{D} คือจำนวนหน่วยย่อในชั้น $(q-1)^{th}$ และ \tilde{M} คือจำนวนหน่วยย่อในชั้น q^{th} . ตัวแปร $\tilde{w}_{jd}^{(q)}$ เป็นค่าน้ำหนักของการเชื่อมต่อระหว่างหน่วยที่ d^{th} ของชั้น $(q-1)^{th}$ กับหน่วยที่ j^{th} ของชั้นคำนวณ q^{th} . ตัวแปร $\tilde{v}_{jm}^{(q)}$ เป็นค่าน้ำหนักของการเชื่อมต่อของหน่วยที่ m^{th} เวียนกลับมาเข้าหน่วยที่ j^{th} ของชั้นคำนวณเดียวกัน, $\tilde{b}_j^{(q)}$ คือค่าไบอสของหน่วยที่ j^{th} และ $h(\cdot)$ คือฟังก์ชันกระตุ้น.

สังเกตสมการ 9.17 เปรียบเทียบกับสมการ 9.1 ซึ่งเป็นการคำนวณค่าตัวกระตุ้น ในชั้นคำนวณ เวียนกลับ ทิศทางปกติ (ทิศทางไปข้างหน้า) จะเห็นว่า จุดสำคัญคือ การที่ค่าตัวกระตุ้น ในชั้นคำนวณ เวียนกลับ ทิศทางย้อนกลับ ได้รับอิทธิพลจาก ผลการกระตุ้นของลำดับเวลาอนาคต $\tilde{z}_m^{(q)}(t+1)$.

การคำนวณของหน่วยย่อในชั้นรวมผลของทั้งสองทิศทาง (เช่น ชั้นเอาร์พุต ที่คำนวณค่า \mathbf{y} ในรูปที่ 9.7) ก็สามารถดำเนินการได้ เช่นเดียวกับการคำนวณหน่วยย่อในชั้นคำนวณเชื่อมต่อเต็มที่ทั่ว 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 ค่าตัวกระตุนและผลการกระตุน ของชั้นที่รวมผลจากการคำนวณเวียนกลับทั้งสองทิศทาง โดย $z_i^{(q-1)}(t)$ และ $\tilde{z}_i^{(q-1)}(t)$ คือผลการกระตุน จากชั้นเวียนกลับทิศทางไปข้างหน้า และทิศทางย้อนกลับ ตามลำดับ. ส่วน $w_{ji}^{(q)}$, $\tilde{w}_{ji}^{(q)}$ และ $\hat{b}_j^{(q)}$ คือพารามิเตอร์ของชั้นคำนวณ.

เพื่อให้การคำนวณค่าเออตพุตของโครงข่ายประสาทเวียนกลับสองทาง เป็นไปโดยเรียบร้อย การคำนวณ (การแพร่กระจายไปข้างหน้า) ดำเนินการตามลำดับดังนี้

- คำนวณค่าผลการกระตุน จากชั้นเวียนกลับทิศทางไปข้างหน้า โดยคำนวณตามลำดับเวลาจาก $t = 1$ ไป $t = T$. นั่นคือ คำนวณค่า $z_i^{(q-1)}(t)$ สำหรับ $t = 1, \dots, T$ ตามลำดับ.
- คำนวณค่าผลการกระตุน 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 (ลำดับได้กีดี).

การฝึกโครงข่ายประสาทเวียนกลับสองทาง ก็สามารถทำได้ในลักษณะเดียวกับการฝึกโครงข่ายประสาทเวียนกลับ เพียงมีความซับซ้อนเพิ่มขึ้น เนื่องจาก (1) การเวียนกลับมีสองทิศทาง และ (2) การเวียนกลับทั้งสองทิศทาง เปรียบเสมือนส่วนประกอบในชั้นคำนวณเดียวกัน เพราะรับอินพุตจากชั้นเดียวกัน และให้อเอตพุตออกไปที่ชั้นเดียวกัน.

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 $\tilde{z}_k^{(q+1)}(t)$ สำหรับทุก ๆ k และทุก ๆ t .
(สมการ 9.19 และ 9.20 ถ้าชั้น $(q+1)^{th}$ เป็นชั้นเชื่อมต่อเต็มที่)
- คำนวณชั้นต่อ ๆ ไป จนได้อาร์พุตสุดท้าย

- (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 (แต่การคำนวณต้อง
ทำจาก $t = T$ 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 (การคำนวณต้องทำ
จาก $t = 1$ 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 $(q - 1)^{th}$ เป็นชั้นเชื่อมต่อเต็มที่ แล้ว

$$\begin{aligned}\hat{\delta}_i^{(q-1)}(t) &\equiv \frac{\partial E}{\partial \hat{a}_i^{(q-1)}(t)} \\ &= h'(\hat{a}_i^{(q-1)}(t)) \cdot \left(\sum_j \delta_j^{(q)}(t) \cdot w_{ji}^{(q)} + \sum_j \tilde{\delta}_j^{(q)}(t) \cdot \tilde{w}_{ji}^{(q)} \right)\end{aligned}\quad (9.25)$$

- (3) ปรับค่าหน้ากากตามเกรเดียนต์ที่คำนวณได้

โครงข่ายประสาทเวียนกลับสองทาง มีความสามารถในการเชื่อมความสัมพันธ์ ทั้งความสัมพันธ์ กับลำดับในอดีต และลำดับในอนาคต. อย่างไรก็ตาม การใช้งานโครงข่ายประสาทเวียนกลับสองทาง เหมาะกับ (1) ลักษณะชุดข้อมูลที่ต้องการจำลองความสัมพันธ์กับลำดับในอนาคต และ (2) ภารกิจที่ ต้องการเอาต์พุต หลังจากได้เห็นชุดข้อมูลครบทุกลำดับแล้ว. หากลักษณะชุดข้อมูลไม่ได้ต้องการความ สัมพันธ์กับลำดับในอนาคต การคำนวณที่เพิ่มขึ้นของโครงข่ายประสาทเวียนกลับสองทาง จะกลายเป็น ภาระที่ไม่จำเป็น. หากภารกิจต้องการเอาต์พุตก่อนที่แบบจำลองจะได้เห็นข้อมูลครบลำดับ โครงข่าย ประสาทเวียนกลับสองทาง (ในรูปแบบดังเดิมนี้) จะไม่เหมาะสมกับภารกิจนั้น และอาจพิจารณาทาง เลือกอื่น เช่น กลไกหน้าต่างเวลา หรือ การหน่วงเวลาระหว่างอินพุตและเอาต์พุต.

9.4 แบบจำลองความจำระยะสั้นที่ยาว

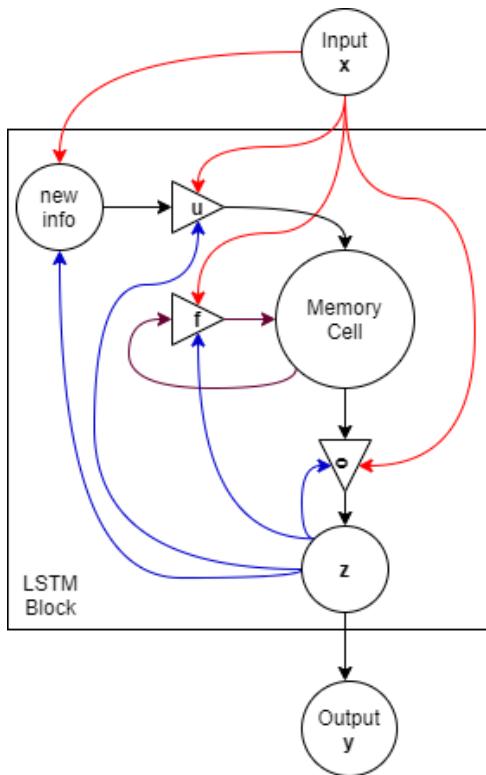
ดังข้อดีข้อเสียของโครงข่ายประสาทเวียนกลับ ที่ได้อธิบายไปว่า โครงข่ายประสาทเวียนกลับ มีปัญหา การเลื่อนหายของเกรเดียนต์ ที่ทำให้โครงข่ายประสาทเวียนกลับ ไม่สามารถเชื่อมโยงความสัมพันธ์ ระยะยาว (ความสัมพันธ์ระหว่างจุดข้อมูลที่ลำดับต่างกันมาก).

ปัญหาเรื่องนี้ นำไปสู่การพัฒนาวิธีแก้ต่าง ๆ มากมาย และหนึ่งในนั้น คือ แบบจำลองความจำระยะ สั้นที่ยาว ที่ปัจจุบัน ได้รับการยอมรับอย่างกว้างขวาง.

แบบจำลองความจำระยะสั้นที่ยาว (*long short-term memory model*[88, 89, 71, 72, 79] ที่มักย่อ LSTM) คือ โครงข่ายประสาทเวียนกลับ ที่เพิ่มกลไกควบคุมความจำค่าใหม่ ควบคุมการลืม ค่าเก่า และควบคุมการระลึกความจำไปใช้ อย่างชัดเจน. รูปที่ 9.8 แสดงโครงสร้างของแบบจำลอง ความจำระยะสั้นที่ยาว. แบบจำลองความจำระยะสั้นที่ยาว มีกลไกภายในหน่วยคำนวณย่อยที่ซับซ้อน กว่า หน่วยย่อยของโครงข่ายประสาทเทียมทั่วไปอยู่มาก ดังนั้น เพื่อกันการสับสน หน่วยคำนวณย่อย ที่ครอบคลุมแนวคิดแบบจำลองความจำระยะสั้นที่ยาว จะเรียกว่า บล็อกความจำ (*LSTM block*).

ภายในบล็อกความจำ มีหน่วยความจำ มีหน่วยความจำ เรียกว่า **เซลล์** (cell) หรือ **เซลล์ความจำ** (*memory cell*).

กลไกการเก็บความจำของเซลล์นี้ คือ จุดเด่นของแบบจำลองความจำระยะสั้นที่ยาว ซึ่งบรรยายดัง สมการ 9.26 ถึง 9.31.



รูปที่ 9.8 แผนภาพโครงสร้างบล็อกความจำของแบบจำลองความจำระยะสั้นที่ยาว. วงกลม แทนหน่วยคำนวณ ดังเช่น แผนภาพโครงข่ายประสาทเทียมอื่น ๆ. สามเหลี่ยม แทนกลไกของประตุคุบคุม ที่ควบคุมการให้ผลของสารสนเทศ โดยการปิดเปิดประตุ ขึ้นกับสัญญาณจากอินพุตเวลาปัจจุบัน และผลลัพธ์ที่ผ่านมา. ดูแผนภาพคลี่ลำดับ (รูปที่ 9.9) ประกอบ.

เมื่อ ชุดลำดับอินพุต คือ $[x(1), \dots, x(T)]$ การคำนวณของบล็อกความจำ ดำเนินการดังนี้

$$\mathbf{f}(t) = \sigma(\mathbf{W}_f \cdot [z(t-1); x(t)] + \mathbf{b}_f) \quad (9.26)$$

$$\mathbf{u}(t) = \sigma(\mathbf{W}_u \cdot [z(t-1); x(t)] + \mathbf{b}_u) \quad (9.27)$$

$$\mathbf{o}(t) = \sigma(\mathbf{W}_o \cdot [z(t-1); x(t)] + \mathbf{b}_o) \quad (9.28)$$

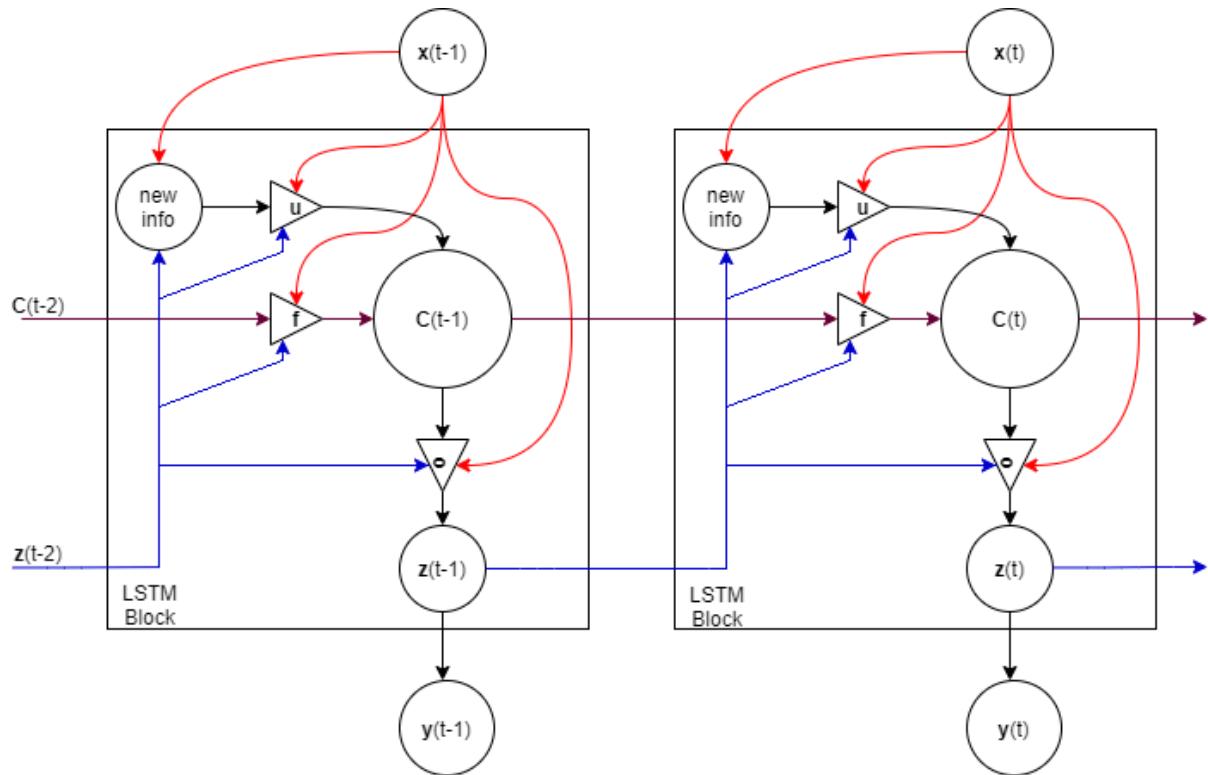
$$\tilde{\mathbf{c}}(t) = \tanh(\mathbf{W}_c \cdot [z(t-1); x(t)] + \mathbf{b}_c) \quad (9.29)$$

$$\mathbf{c}(t) = \mathbf{f}(t) \odot \mathbf{c}(t-1) + \mathbf{u}(t) \odot \tilde{\mathbf{c}}(t) \quad (9.30)$$

$$z(t) = \mathbf{o}_t \odot \tanh(\mathbf{c}(t)) \quad (9.31)$$

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 $\mathbf{v}^{(2)} = [v_1^{(2)}, \dots, v_N^{(2)}]^T$ และ $[\mathbf{v}^{(1)}; \mathbf{v}^{(2)}] = [v_1^{(1)}, \dots, v_M^{(1)}, v_1^{(2)}, \dots, v_N^{(2)}]^T$. 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 $\mathbf{o}(t)$ ทำหน้าที่เป็นสมีองประตุควบคุมการให้ผลของสารสนเทศ. ประตุลืม (forget gate) และประตุรับค่าใหม่ (update gate) ซึ่งคือ $\mathbf{f}(t)$ และ $\mathbf{u}(t)$ ตาม



รูปที่ 9.9 แผนภาพคลื่ล้ำดับของแบบจำลองความจำระยะสั้นที่ยาว สำหรับสองลำดับเวลา.

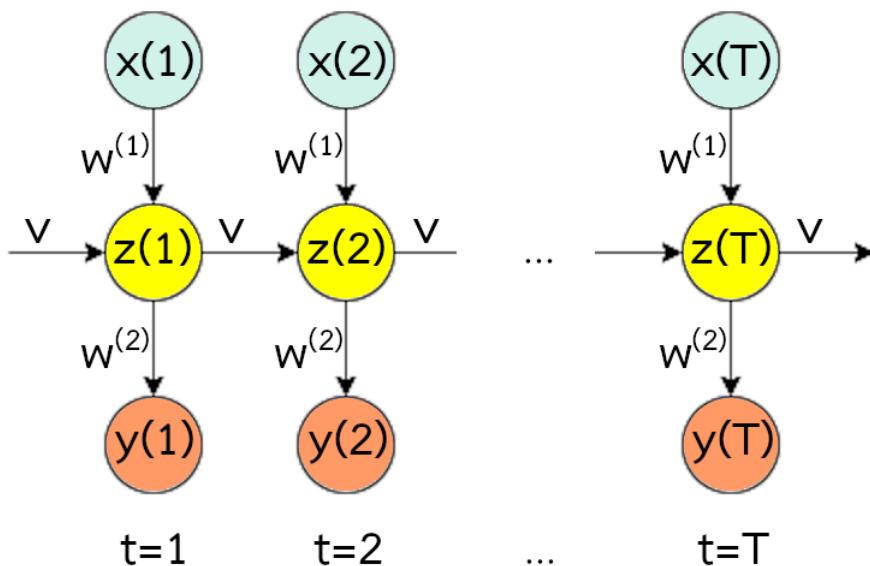
ลำดับ ควบคุมว่า จะให้เซลล์รับสารสนเทศใหม่ $\tilde{c}(t)$ หรือคงความจำเดิม $c(t-1)$. ประตูผลลัพธ์ (output gate) $o(t)$ ควบคุมว่า ควรจะระลึกความจำอกรมาหรือไม่. ค่าอินพุต $x(t)$ และค่าผลลัพธ์ที่ผ่านมา $z(t-1)$ ควบคุมการทำงานของประตูต่าง ๆ. รูปที่ 9.9 แสดงแผนภาพคลื่ล้ำดับของบล็อกความจำ.

นอกจากนี้ เพื่อปรับการควบคุมประตูได้แม่นยำขึ้น โครงสร้างของบล็อกความจำ อาจมีกลไกช่องแอบมอง (peephole connections^[72]) เพิ่มเข้ามาด้วย. กลไกของช่องแอบมอง จะเพิ่มค่าของความจำที่ผ่านมา เข้ามาเป็นส่วนในการควบคุมประตูต่าง ๆ ด้วย ดังสมการ 9.32 ถึง 9.34.

$$f(t) = \sigma(\mathbf{W}_f \cdot [z(t-1); x(t); c(t-1)] + \mathbf{b}_f) \quad (9.32)$$

$$u(t) = \sigma(\mathbf{W}_u \cdot [z(t-1); x(t); c(t-1)] + \mathbf{b}_u) \quad (9.33)$$

$$o(t) = \sigma(\mathbf{W}_o \cdot [z(t-1); x(t); c(t-1)] + \mathbf{b}_o). \quad (9.34)$$



รูปที่ 9.10 แผนภาพคลื่นลำดับของโครงข่ายภาษาที่เวียนกลับ สำหรับกรณีที่ห้องอินพุตและเอาต์พุตเป็นชุดลำดับ และมีจำนวนลำดับเท่ากัน.

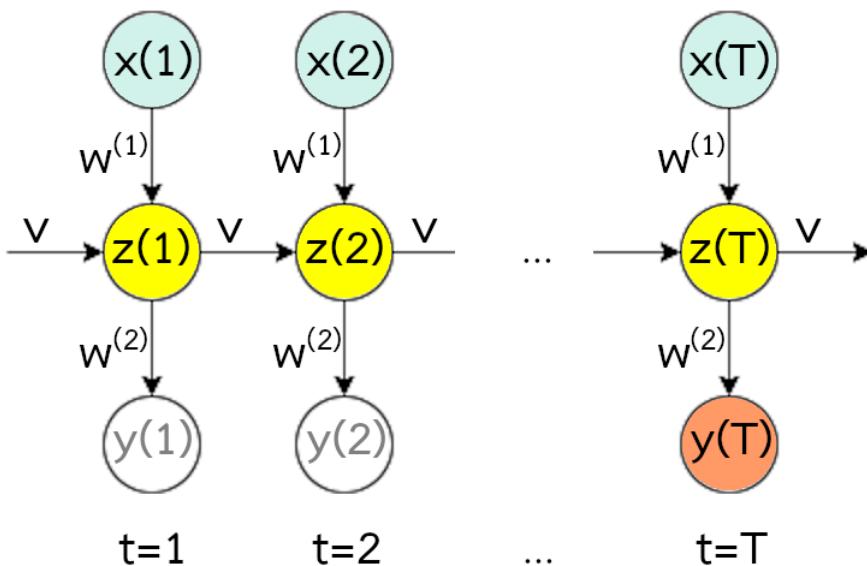
9.5 การใช้งานโครงข่ายภาษาที่เวียนกลับ

โครงข่ายภาษาที่เวียนกลับ³ สามารถใช้เป็นแบบจำลองในการรู้จำรูปแบบเชิงลำดับแบบต่าง ๆ ดังอภิรายในหัวข้อ 8.1. ตัวอย่างเช่น กรณีที่ห้องอินพุตและเอาต์พุตเป็นชุดลำดับ และมีจำนวนลำดับเท่ากัน เช่น การระบุหมวดคำ (ดูแบบฝึกหัด 9.7) โครงข่ายภาษาที่เวียนกลับ สามารถนำมาใช้ในกรณีนี้ได้อย่างตรงไปตรงมา. กระบวนการฝึก อาจกำหนดให้ $\mathcal{M}(t) = 1$ สำหรับทุก ๆ ลำดับเวลา t . แผนภาพคลื่นลำดับ สำหรับกรณีนี้ แสดงในรูปที่ 9.10.

กรณีการจำแนกลำดับ เช่น การจำแนกอารมณ์ (ดูแบบฝึกหัด 9.6) ที่อินพุตเป็นข้อมูลชุดลำดับ แต่เอาต์พุตไม่ได้เป็นชุดลำดับ นั่นคือ อินพุต $\mathbf{X} = \{\mathbf{x}(1), \dots, \mathbf{x}(T)\}$ และเอาต์พุต \mathbf{y} . กรณีนี้อาจดำเนินการโดยกำหนดให้ ค่าหมายที่ลำดับเวลาสุดท้าย เป็นเอาต์พุต และค่าหมายต่าง ๆ ที่ลำดับเวลาอื่น ๆ ไม่มีความสำคัญ. กระบวนการฝึก ในกรณีการจำแนกลำดับ ก็สามารถทำได้สะดวก โดยการใช้กลไกหน้าหาก ที่กำหนดให้ $\mathcal{M}(T) = 1$ และ $\mathcal{M}(t \neq T) = 0$. รูปที่ 9.11 แสดงแผนภาพคลื่นลำดับ สำหรับกรณีการจำแนกลำดับ.

กรณีที่เอาต์พุตเป็นชุดลำดับ แต่อินพุตไม่ใช่ เช่น ระบบแต่งเพลงอัตโนมัติ (ดูแบบฝึกหัด 8.4) โครงข่ายภาษาที่เวียนกลับ ก็อาจสามารถนำมาประยุกต์ใช้ได้โดยจัดโครงสร้างดังแสดงในรูปที่ 9.12. การจัดโครงสร้างที่นิยม สำหรับกรณีเช่นนี้ คือ ใช้อินพุตของระบบ เป็นอินพุตที่ลำดับเวลาแรกสุดของโครงข่ายภาษาที่เวียนกลับ และหลังจากนั้น ใช้เอาต์พุตที่คำนวณได้ มาเป็นอินพุตสำหรับลำดับเวลาถัดไป. การฝึกโครงข่ายภาษาที่เวียนกลับ สำหรับกรณีเช่นนี้ ซึ่งเป็นลักษณะของการใช้งานแบบจำลอง

³ ณ ที่นี่ โครงข่ายภาษาที่เวียนกลับ จะหมายรวมถึง โครงข่ายภาษาที่เวียนกลับทุก ๆ ชนิด ซึ่งรวมถึง แบบจำลองความจำระยะสั้นที่ยาว และหน่วยเวียนกลับมีประตุ ด้วย.



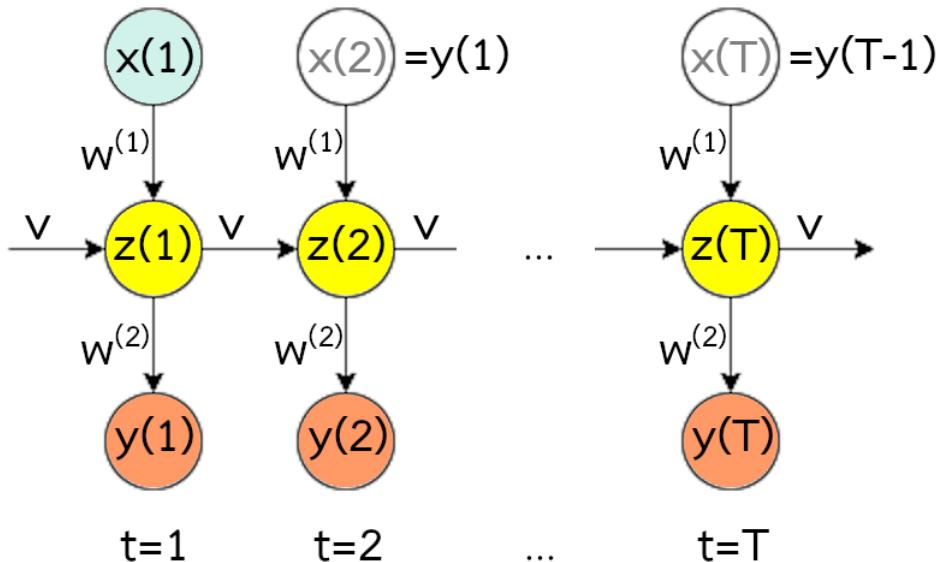
รูปที่ 9.11 แผนภาพคลื่นล้ำดับ กรณีการจำแนกลำดับ. จุดสำคัญอยู่ที่ การเลือกเฉพาะเอาต์พุตที่ต้องการไปใช้ (ค่าที่ทำนาย $\hat{y} = y(T)$). แม้ค่าแบบจำลองจะยังคงให้อาต์พุตอื่น ๆ $y(t \neq T)$ ออกมามาด้วย เพียงแต่เอาต์พุตเหล่านี้ไม่ได้ถูกนำไปใช้ทำงานໄร (แผนภาพใช้สีจาง เพื่อสื่อถึงการปล่อยค่าอาต์พุตเหล่านี้ทิ้ง).

ก่อกำเนิด ก็สามารถใช้แนวทางของงานแบบจำลองก่อกำเนิด เช่น แนวทางของโครงข่ายปรับปักรชีวิตริบส์ สร้างได้ (ดูหัวข้อ 7.2 ประกอบ).

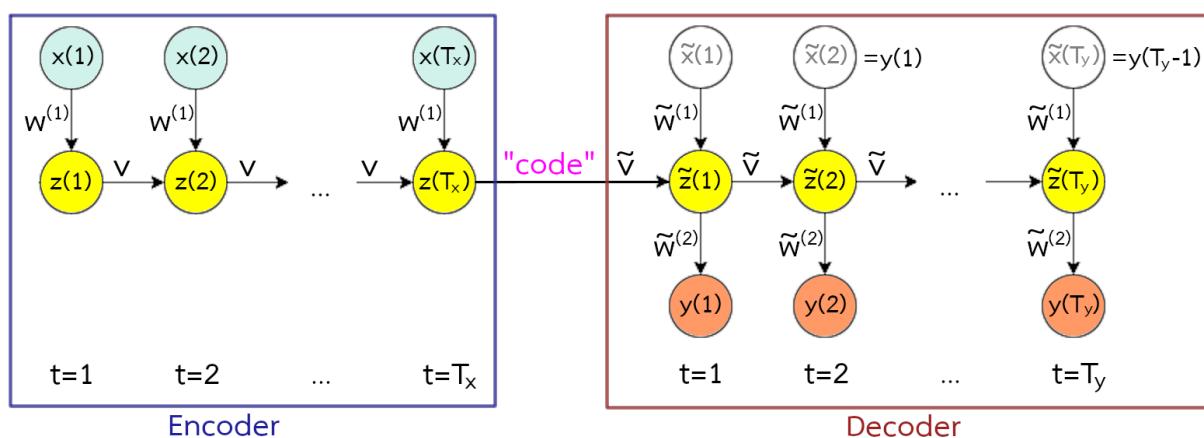
สุดท้าย กรณีที่ทั้งอินพุตและอาต์พุตเป็นชุดลำดับ แต่มีจำนวนลำดับอาจไม่เท่ากัน เช่น การกิจกรรมแปลงภาษาอัตโนมัติ ที่จำนวนคำในประโยคของภาษาต้นทาง อาจไม่เท่ากับจำนวนคำในประโยคของภาษาเป้าหมาย. กรณีนี้ บอยครั้งอาจถูกอ้างถึงเป็น ภารกิจจำลองแบบชุดลำดับเป็นชุดลำดับ (*sequence-to-sequence modeling task*) เป็น สถานการณ์ที่ท้าทายอย่างมาก โดยเฉพาะ เมื่อการทำนายชุดลำดับของอาต์พุต จำเป็นต้องเห็นอินพุตครบถ้วนลำดับก่อน. แนวทางหนึ่ง คือการใช้สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส (*encoder-decoder architecture*[38]) หรือบางครั้งอาจเรียก สถาปัตยกรรมแปลงชุดลำดับไปชุดลำดับ (*sequence-to-sequence architecture*[193]) ดังแสดงในรูปที่ 9.13 (ดูแบบฝึกหัด 9.5 เพิ่มเติม).

สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส คล้ายการรวมแบบจำลองสองตัว. ตัวแรก สำหรับชุดลำดับอินพุต (เรียก ตัวเข้ารหัส) และตัวที่สอง สำหรับชุดลำดับอาต์พุต (เรียก ตัวถอดรหัส). สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส อาศัยกลไกของรหัสเนื้อความ (*code*) หรืออาจเรียก บริบท (*context*) ในการสรุปความหมายของชุดลำดับอินพุตไว้ แล้วค่อยถอดรหัสเนื้อความออกมาเป็นอีกชุดลำดับ.

โดยทั่วไป ผลการกระตุ้นลำดับสุดท้ายของตัวเข้ารหัส เช่น $\mathbf{z}(T_x)$ ในรูปที่ 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 $\tilde{\mathbf{x}}(1)$ อาจ



รูปที่ 9.12 แผนภาพคลื่ลำดับของโครงข่ายประสาทเวียนกลับ ที่นิยมใช้สำหรับกรณีที่เอาต์พุตเป็นชุดลำดับ แต่อินพุตไม่ใช่.
อินพุตจริงของระบบ \tilde{x} จะถูกนำเข้าเป็นจุดข้อมูลที่ลำดับเวลาแรก นั่นคือ $x(1) = \tilde{x}$ และหลังจากลำดับเวลาแรก เอาต์พุตที่คำนวณได้ จะถูกนำเป็นอินพุตสำหรับลำดับเวลาถัดไป. ภาพแสดง $x(2)$ ถึง $x(T)$ ด้วยสีจาง เพื่อสื่อถึงว่า ค่าอินพุตที่ลำดับเวลาเหล่านี้ ไม่ใช้อินพุตของระบบ แต่เป็นค่าที่ถูกสร้างขึ้นในกระบวนการ.



รูปที่ 9.13 แผนภาพคลื่ลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส. สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส ใช้โครงข่ายประสาทเวียนกลับสองตัว ตัวหนึ่งทำงานที่เข้ารหัส (encoder แบบจำลองทางซ้ายในภาพ) อีกตัวหนึ่งทำงานที่ถอดรหัส (decoder แบบจำลองทางขวาในภาพ). ตัวเข้ารหัส สรุปเนื้อความของชุดลำดับอินพุต ไว้เป็นรหัสเนื้อความ แล้วส่งรหัสเนื้อความนี้ไปให้ตัวถอดรหัส เพื่อถอดออกมานาเป็นชุดลำดับเอาต์พุต.

กำหนดด้วยค่า **0**. ดูแบบฝึกหัด 9.5 เพิ่มเติม สำหรับการอภิปรายโครงสร้างของสถาปัตยกรรมตัวเข้ารหัสตัวอุดรหัสแบบอื่น.

โดยทั่วไป ขนาดสถานะภายในของตัวอุดรหัส ไม่เล็กกว่าขนาดสถานะภายในของตัวเข้ารหัส นั่นคือ $|z(t)| \leq |\tilde{z}(t)|$. การกำหนดขนาดสถานะภายในของตัวอุดรหัส ให้ใหญ่กว่าขนาดรหัส ช่วยให้ตัวอุดรหัสประเมินมีความจำเพลือพอที่จะใช้งานอื่น ๆ ได้ (เช่น อาจจะเก็บสถานะของการทำงาน ณ ลำดับเวลาปัจจุบัน).

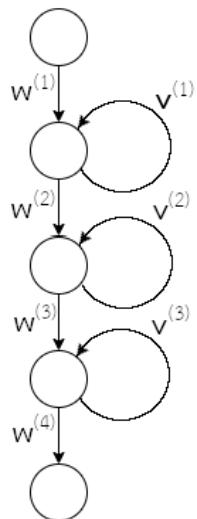
แม้สถาปัตยกรรมตัวเข้ารหัสตัวอุดรหัส จะสามารถช่วยให้ชุดลำดับของເອົ້າພຸດມีจำนวนลำดับที่เป็นอิสระจากจำนวนลำดับของອິນພຸດ และยังช่วยให้การคำนายนຸດลำดับของເອົ້າພຸດ ได้ເຫັນອິນພຸດຄຽບທຸກລຳດັບກ່ອນ ແຕ່สถาปัตยกรรมตัวเข้ารหัสตัวอุดรหัส ໃຊ້ກລິກຂອງຮັບສິນ ໃນການສັງເກົນຄວາມໝາຍສຽງຈາກອິນພຸດ ໄປສ້າງອິນພຸດລຳດັບເອົ້າພຸດ. ດັ່ງນັ້ນ ขนาดຂອງຮັບສິນ ມີຜລໂດຍຕຽບຕ່ອງຄວາມສາມາດ ໃນການແທນຄວາມໝາຍ. ໃນການນີ້ຂອງຮັບສິນ ເປົ້າໄດ້ການແປລກາຍາອັດໂນມັຕີ ຂະນຸມຂອງຮັບສິນທີ່ເລີກເກີນໄປ ຈະສັງຜລລົບຕ່ອງຄຸນກາພກແປລໂຍ່ງໜັດເຈັນ ໂດຍເຂົາມະວິທີກັບການແປລປະໂຍຄຍາວ ພ[37, 9] (ดูแบบฝึกหัด 9.9 ເພີ່ມເຕີມ. ທັງໝົດ 9.6 ອົງປະກາດໄກຄວາມໄສ່ໃຈ ຈຶ່ງພັດນາມາເພື່ອແກ້ໜ້ອຈຳກັດນີ້).

การจัดโครงสร้างเชิงลືກຂອງໂຄຮູງຂ່າຍປະສາທວີຢັນກັບ. ໂຄຮູງຂ່າຍປະສາທວີຢັນກັບ (ສາມາດ 9.1 ແລະ 9.2) ສາມາດຖືກຈັດໂຄຮູງຂ່າຍປະສາທວີຢັນກັບໄດ້ (ຮູບທີ 9.14). ອີ່ຍໍາໄຮກ້ຕາມ ຜູ້ເຂົ້າວ່າຍຸດສາຕ່າງໆ ເຊິ່ງຮັບສິນ ຮູ່ອົງເຄື່ອງ ແອນດຽວ່າ ວິຊ້[140] ໃຫ້ຂໍ້ສັງເກດວ່າ ແມ່ປັຈຸບັນ ໂຄຮູງຂ່າຍປະເທີມເຊີງລືກຈາມມີຈຳນວນຂັ້ນ ຄຳນວນເປັນຫລັກຮ້ອຍຂັ້ນໄດ້ (ເຊັ່ນ ເຮັດວຽກ[85]) ແຕ່ສໍາທັບໂຄຮູງຂ່າຍປະສາທວີຢັນກັບ ໂດຍເຂົາມະວິທີ່ຂັ້ນ ຄຳນວນເວີຍນັກັບ ມັກຕ່ອກັນໄມ່ເກີນສາມຂັ້ນ⁴ ເຫຼຸດຜລສ່ວນໜຶ່ງ ຈາກເປັນພຣະກລິກໄກການເວີຍນັກັບໄດ້ເພີ່ມຄວາມສາມາດຂອງແບບຈຳລອງຂັ້ນຍ່າງມາກ (ຮວມເຖິງເພີ່ມຄວາມຕ້ອງການການຄຳນວນຂັ້ນຍ່າງມහາສາລ ໂດຍເຂົາມະວິກະບວນການຝຶກ). ແຕ່ໂຄຮູງຂ່າຍປະສົງລືກມັກພບ ອີ່ການໃຊ້ໂຄຮູງຂ່າຍປະສາທວີຢັນກັບ ທີ່ມີຂັ້ນເວີຍນັກັບ (ຫົ່ງເຖິງສາມຂັ້ນ) ແລ້ວຕ່ອງດ້ວຍຂັ້ນຄຳນວນ (ທີ່ໄມ່ມີກລິກໄກເວີຍນັກັບ) ລາຍ ພ ຂັ້ນ (ຮູບທີ 9.15).

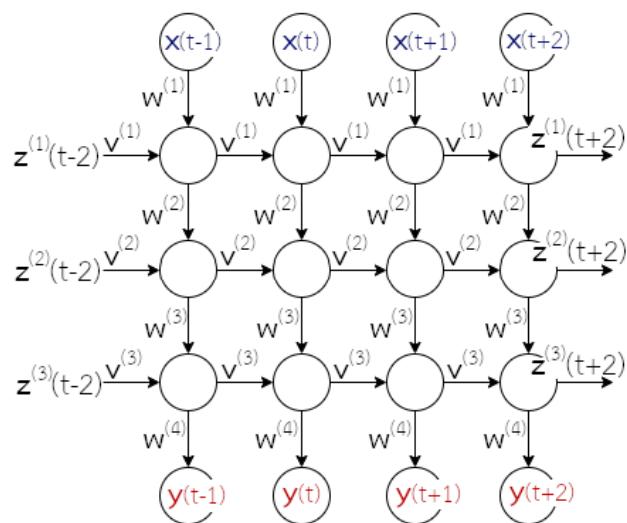
9.6 กลไกความใส่ใจ

สำหรับการกิจจำลองแบบชຸດລຳດັບເປັນຫຼຸດລຳດັບ ເຊັ່ນ ການແປລກາຍາອັດໂນມັຕີ (machine translation), ການສຽບຂໍ້ຄວາມ (text summarization), ແຈບອຕ (chatbot) ແລະ ຮະບບຕອບຄຳຄາມ (question answering) ສາມາດຮັບສິນຕົວເຂົ້າຮັບສິນຕົວອິນພຸດ ສາມາດການໄດ້ດີ ແຕ່ຄວາມສາມາດຂອງຮັບສິນ

⁴ຂໍ້ສັງເກດນີ້ ຕັ້ງຂັ້ນຈາກສາພສີ່ແວດລ້ອມ ແລະ ຄວາມນີ້ຍືມໃນປັຈຸບັນ. ປະວັດທີແລະ ວິວວັດນາກາຮ່ອງໂຄຮູງຂ່າຍປະສາທວີຢັນ ເຊັ່ນ ການໃຊ້ເພົ່າເຫຼຸດຕອນຫລາຍຂັ້ນ (ບທທີ 3) ໃນຊ່ວງເວລາກ່ອນປີ ປ.ສ. 2012 (ທີ່ໂຄຮູງຂ່າຍປະສາທວີຢັນເຊີງລືກເຮີມໄດ້ຮັບຄວາມສຸນໃຈຍ່າງກວ້າງຂວາງ) ກີ່ມີຄວາມເຝື່ອວ່າ ໄນມີຄວາມຈຳເປັນທີ່ຈະຕ້ອງໃຊ້ໂຄຮູງຂ່າຍທີ່ມີຈຳນວນຂັ້ນຄຳນວນເກີນກ່າວສອງຂັ້ນ. ດັ່ງນັ້ນ ຂໍ້ສັງເກດນີ້ ຜູ້ເຂົ້າມະວິກະບວນການຝຶກ ເພື່ອຜູ້ອ່ານຈະໄດ້ພົບເຫັນການນຳໄປໃຊ້ງານ ແຕ່ທີ່ວ່າງວ່າ ຜູ້ອ່ານຈະໄມ່ຍື້ດີຕິດຈຸນເກີນໄປ ດັ່ງທີ່ປະວັດສາຕ່າງໆ ໄດ້ໄວ້ຕອດມາ. ນັ້ກຮຽມຄວາມກ້າວໜ້າ ພັດນາກາທີ່ສຳຄັນ (breakthrough) ເກີດຈາກການປລ່ອຍວາງ ມາກກວ່າກາຍີດຕິດ.

Structural diagram
of a deep RNN

Unfolding diagram of a deep RNN



รูปที่ 9.14 โครงสร้างและแผนภาพคลื่นลำดับของโครงข่ายประสาทเวียนกลับแบบลึก. ตัวอย่างนี้ แสดงการใช้ชั้นเวียนกลับสามชั้นต่อ กัน.

ลงอย่างมาก เมื่อชุดลำดับอินพุตมีความยาวมาก ๆ. คณะของบาร์ดโน[9] เชื่อว่าข้อจำกัดนี้ เกิดจากปัญหาของขวดที่การใช้รหัสเนื้อความ ซึ่งมีความยาวจำกัด และได้เสนอวิธีการแก้ด้วยกลไกความไม่ใช่.

แนวคิดของกลไกความไม่ใช่ ผ่านจากการกิจกรรมแปลภาษาอัตโนมัติ ซึ่งสังเกตว่า เวลาที่คนแปลภาษา แม้จะเป็นการแปลจากประโยคยาว ๆ ในภาษาต้นทาง แต่เวลาแปล ถอดความออกมานะเป็นคำ ๆ ในภาษาปลายทาง ผู้แปล แม้จะเห็นห้องประโยค แต่เวลาพิจารณาเลือกคำ แต่ละคำ ที่จะใช้สำหรับประโยคปลายทาง ผู้แปลจะใส่ใจกับเฉพาะบางส่วนของประโยคต้นทางเท่านั้น.

ตัวอย่างเช่น ประโยคต้นทาง (ภาษาอังกฤษ) คือ “Knowing that what is cannot be undone —because it already is — you say yes to what is or accept what isn’t.”⁵

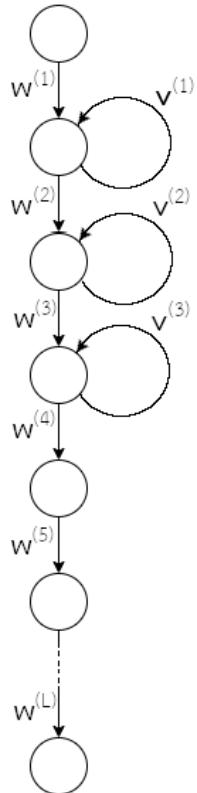
ประโยคปลายทาง (ภาษาไทย) คือ “การรู้ว่า สิ่งที่เป็น ไม่สามารถเปลี่ยนแปลงได้ (เพราะว่ามันเป็นไปแล้ว) คุณยอมรับกับ สิ่งที่เป็น หรือยอมรับ สิ่งที่ไม่ได้เป็น.”

เอาร์พุต “การรู้ว่า” อาจมาจากโทเค็น “Knowing” และโทเค็น “that” เป็นหลัก โดยส่วนอื่น ๆ ของประโยคต้นทางมีความเกี่ยวข้องต่ำ.

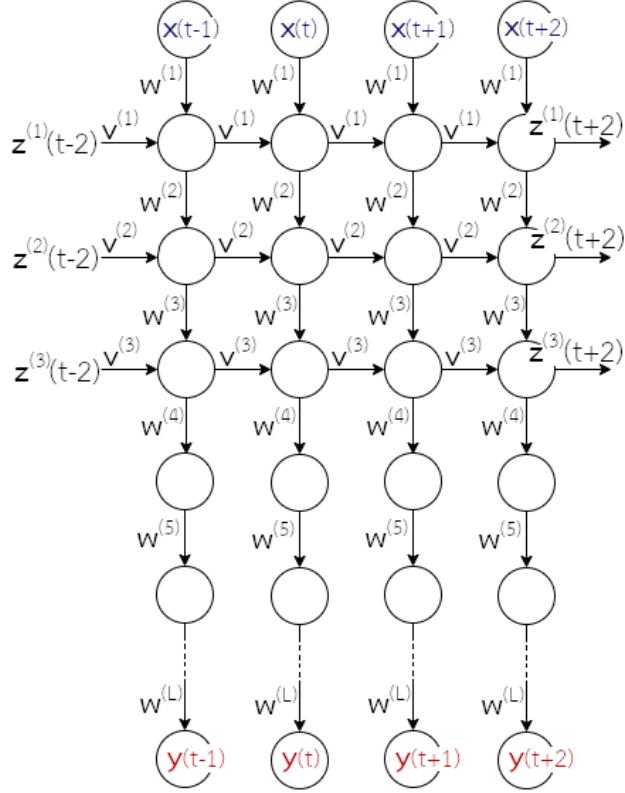
การทำงานของกลไกความไม่ใช่. แทนที่จะอาศัยการส่งเนื้อความจากชุดลำดับอินพุต ไปชุดลำดับเอาร์พุต ผ่านรหัสเนื้อความที่มีความยาวจำกัด และให้รหัสกับตัวถอดรหัสในการจัดเรียงเอาร์พุต เอง กลไกความไม่ใช่ (attention mechanism) กำหนดบริบทสำหรับแต่ละลำดับ ให้กับตัวถอดรหัสโดยตรง.

⁵ จาก Eckhart Tolle, The Power of Now.

Structural diagram of a deep RNN with non-recurrent layers



Unfolding diagram of a deep RNN with non-recurrent layers



รูปที่ 9.15 โครงสร้างและแผนภาพคลื่นลำดับของโครงข่ายประสาทเวียนกลับแบบลีก ที่ใช้ชั้นคำนวนไม่เรียนกลับจำนวนมาก.

ด้วยชุดลำดับอนพุต $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(T_x)]$ สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส คำนวนรหัสเนื้อความ

$$\mathbf{c} = e'(\mathbf{z}(1), \dots, \mathbf{z}(T_x)) \quad (9.35)$$

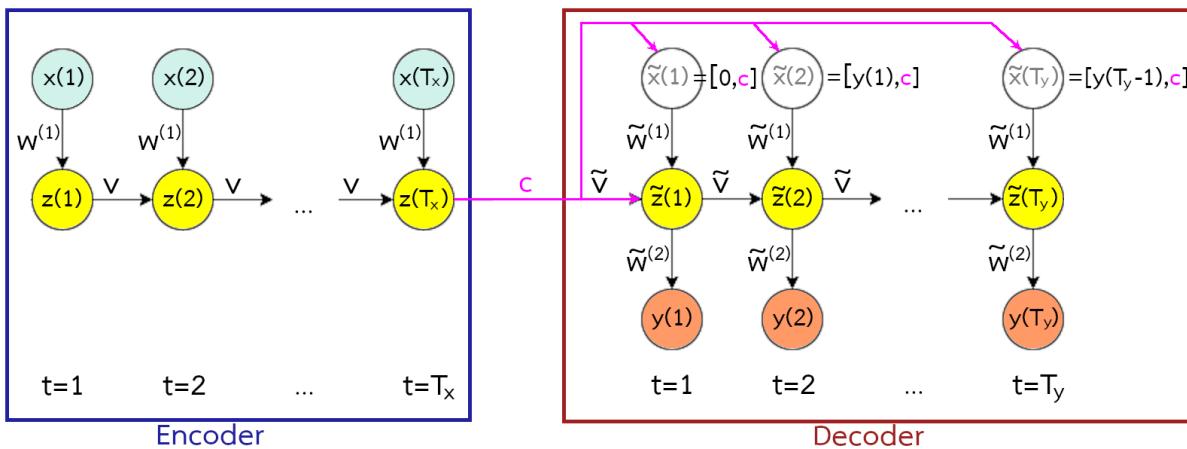
เมื่อ

$$\mathbf{z}(t) = e(\mathbf{z}(t-1), \mathbf{x}(t), \mathbf{z}(t+1)) \quad (9.36)$$

โดย $e'(\cdot)$ และ $e(\cdot)$ เป็นฟังก์ชันของตัวเข้ารหัส ที่ทำหน้าที่สรุป และจำลองแบบเชิงลำดับ. ตัวอย่าง เช่น $e'(\mathbf{z}(1), \dots, \mathbf{z}(T_x)) = \mathbf{z}(T_x)$ และ $e(\cdot)$ เป็นโครงข่ายประสาทเวียนกลับสองทาง.

ในสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัสแบบดั้งเดิม (รูปที่ 9.16) ตัวถอดรหัสรับรหัสเนื้อความ \mathbf{c} และระบบประมาณความน่าจะเป็นของเอกสารพุตที่ลำดับ t ด้วย

$$p(\mathbf{y}(t) | \{\mathbf{y}(1), \dots, \mathbf{y}(t-1)\}, \mathbf{X}) \approx \mathcal{D}(\mathbf{y}(t-1), \tilde{\mathbf{z}}(t), \mathbf{c}) \quad (9.37)$$



รูปที่ 9.16 สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส ที่ใช้รหัสเนื้อความประกอบอินพุตของตัวถอดรหัส.

เมื่อ $\mathcal{D}(\cdot)$ เป็นฟังก์ชันที่อนุมานความน่าจะเป็นของเอาร์พุต⁶ และ $\tilde{\mathbf{z}}(t)$ เป็นสถานะซ่อน ของโครงข่ายประสาทเวียนกลับ ที่ลำดับเวลา t .

จากสมการ 9.37 เราจะเห็นว่า ตัวถอดรหัสรับรู้ชุดลำดับอินพุต \mathbf{X} ผ่านรหัสเนื้อความ \mathbf{c} ตลอดการประมาณเอาร์พุตทุก ๆ ลำดับ. ดังนั้น ชุดลำดับอินพุตที่มีความยาวมาก อาจจะยืดเยียดสารสนเทศจำนวนมากลงไบในรหัสเนื้อความ \mathbf{c} ซึ่งแม้โดยแนวคิด รหัสเนื้อความไม่ได้ถูกจำกัดขนาด แต่ในทางปฏิบัติ การนำแนวคิดสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัสไปใช้งาน จะกำหนดขนาดของรหัสเนื้อความนี้ และขนาดของรหัสเนื้อความที่จำกัด อาจทำให้เกิดปัญหาอย่างชุดลำดับยาว 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 . ค่าสถานะซ่อนคำนวณได้จาก

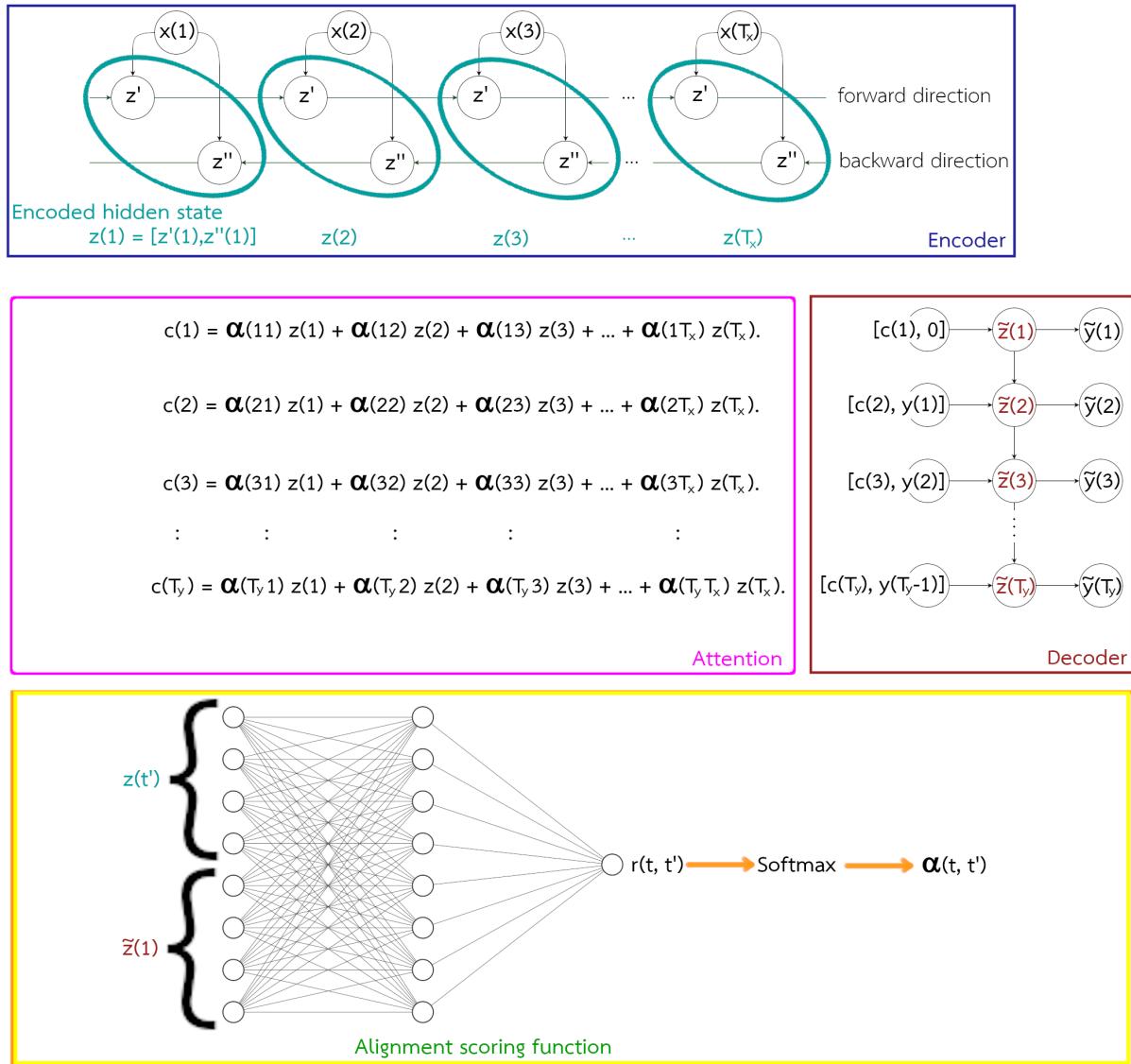
$$\tilde{\mathbf{z}}(t) = d(\tilde{\mathbf{z}}(t-1), \mathbf{y}(t-1), \mathbf{c}_t) \quad (9.39)$$

เมื่อ $d(\cdot)$ เป็นส่วนของตัวถอดรหัสที่ทำหน้าที่จำลองแบบเชิงลำดับ⁷.

บริบท \mathbf{c}_t ควรจะขึ้นกับชุดลำดับของเนื้อความย่ออย $\{\mathbf{z}(1), \dots, \mathbf{z}(T_x)\}$ ที่ตัวเข้ารหัสได้ไว้เคราะห์มาจากชุดลำดับอินพุต. เนื่องจากตัวเข้ารหัสเป็นโครงข่ายประสาทเวียนกลับสองทาง แต่ละเนื้อความ

⁶นิพจน์นี้ (บรรยายตามคณของบทที่ 9) ต้องการสืบถึงความสัมพันธ์ระหว่างรหัสเนื้อความกับการแทนใจความของลำดับอินพุต 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 $\tilde{\mathbf{z}}(t)$ มาก่อนมาจากการคำนวณได้จาก $\tilde{\mathbf{z}}(t) = d(\tilde{\mathbf{z}}(t-1), \mathbf{y}(t-1), \mathbf{c})$ เมื่อ $d(\cdot)$ เป็นส่วนของตัวถอดรหัส.

⁷เนื่องจาก $d(\cdot)$ ต้องให้ผลลัพธ์ออกมาทุก ๆ ลำดับ ก่อนที่จะจบลำดับ ซึ่งแม้แต่การจบลำดับก็อาจจะถูกกำหนดด้วยค่าของผลลัพธ์ที่ออกมานั้น $d(\cdot)$ อาจเป็นโครงข่ายประสาทเวียนกลับ แต่ไม่ใช่โครงข่ายประสาทเวียนกลับสองทาง.



ຮູບທີ 9.17 ແຜນການແສດງໂຄຮງສຮ້າງ ເນື່ອໃຫ້ກລົມໄກຄວາມໃສ່ຈີ. ດ່ານ້ຳໜັກຄວາມໃສ່ຈີ $\alpha_{t,t'}$ ຈະຖືກຄຳນວນທຸກ ຈົດ ອອບຂອງລຳດັບເອົາຕົວ t . ແມ່ວ່າ ດ່ານ້ຳໜັກຄວາມໃສ່ຈີ ໃນກາພ ອາຈດູ້ເໜືອນເມທຣິກ໌ ແຕ່ $\alpha_{t,t'}$ ຖືກຄຳນວນຈາກຄ່າຄະແນນ $r_{t,t'}$ ທີ່ເປັນພັກໜັນຂອງຄ່າສຕານະໜ່ອນ ໄນໄດ້ເຊື້ນກັບລຳດັບ t ແລະ t' ໂດຍຕຽງ. ດ່ານ້ຳຄະແນນ $r_{t,t'}$ ອາຈໄດ້ຈາກໂຄຮງຂ່າຍປະສາທເຫີມ ດັ່ງແສດງໃນກາພ (ພັກໜັນຄະແນນກາຈັດເຮືອງຕຳແໜ່ງ ຍ້າງວິທີ alignment scoring function).

โดย $\mathbf{z}(t)$ จึงถูกสรุปมาจากการอินพุตทั้งชุดลำดับ โดยเน้นลำดับต่าง ๆ บริเวณรอบ ๆ ลำดับ t (ของชุดลำดับอินพุต).

คณะของบ้าห์ดาโน [9] กำหนดให้บริบทของเอาร์พุตที่ลำดับเวลา t เป็นผลรวมตามน้ำหนักของเนื้อความย่ออยู่ต่าง 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. และเพื่อให้การคำนวณมีเสถียรภาพ ค่าน้ำหนักความใส่ใจ $\alpha_{t,t'}$ 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' ควรมีผลกับเอาร์พุตลำดับ t มากน้อยขนาดไหน. แบบจำลองการจัดเรียงตำแหน่ง ควรขึ้นอยู่กับสถานะช่องจากคู่ลำดับอินพุตและเอาร์พุต ดังนี้

$$r_{t,t'} = f(\tilde{\mathbf{z}}(t-1), \mathbf{z}(t')) \quad (9.42)$$

โดย $f(\cdot)$ เป็นฟังก์ชันคำนวณคะแนนความสัมพันธ์ระหว่างลำดับ t' ของอินพุต กับลำดับ t ของเอาร์พุต. สมการ 9.42 ใช้ $\tilde{\mathbf{z}}(t-1)$ แทนที่จะเป็น $\mathbf{z}(t)$ เพราะว่า $\tilde{\mathbf{z}}(t-1)$ คือสถานะช่องล่าสุดของตัวถอดรหัส (ดูลำดับการคำนวณประกอบ). ฟังก์ชันคะแนน $f(\cdot)$ ที่สามารถทำได้จากโครงข่ายประสาทเทียม โดยทำการฝึกโครงข่ายไปพร้อม ๆ กับแบบจำลองส่วนอื่น ๆ ของระบบ.

สังเกต ฟังก์ชัน $f(\cdot)$ คำนวณคะแนน โดยใช้ค่าเวคเตอร์จากสถานะช่อง ไม่ได้ใช้ค่าดัชนี t หรือ t' โดยตรง. ดังนั้น (1) ไม่ต้องกังวลเรื่องความยาวของลำดับ และ (2) การเชื่อมโยงระหว่างลำดับอินพุตและลำดับเอาร์พุต ทำผ่านสถานะช่อง ไม่ได้ขึ้นกับตำแหน่งสัมบูรณ์.

รูปที่ 9.17 แสดงโครงสร้าง เมื่อใช้กลไกความใส่ใจ. สถานะช่องของตัวเข้ารหัสทุก ๆ ลำดับเวลา จะถูกนำไปประกอบเป็นบริบท โดยสถานะช่องที่ลำดับเวลาใด จะถูกผสมเข้าไปมากน้อยเท่าไร ขึ้นกับน้ำหนักความใส่ใจ. สถานะช่องของตัวเข้ารหัส $\mathbf{z}(t) = [\mathbf{z}'(t); \mathbf{z}''(t)]$ เมื่อ $\mathbf{z}'(t)$ และ $\mathbf{z}''(t)$ คือผลกระทบตุนในทิศทางไปข้างหน้า และกลับหลัง ตามลำดับ.

ด้วยกลไกความใส่ใจ ไม่ว่าชุดลำดับอินพุตจะยาวเท่าไร แต่ละลำดับเวลาของเอาร์พุต จะเสนือ่อนมองเห็นทั้งชุดลำดับอินพุต เพียงแต่เน้นความหมายจากเฉพาะส่วนที่เกี่ยวข้องเท่านั้น ซึ่งแม้ขนาดของบริบทที่ลำดับ \mathbf{c}_t จะไม่ได้ใหญ่ไปกว่าขนาดของรหัสเนื้อความ \mathbf{c} แต่การที่ค่าของบริบทเปลี่ยนไป

ตามลำดับของเอาร์พุตได้ ช่วยแก้ปัญหาคอขวดของการกิจจำลองแบบชุดลำดับเป็นชุดลำดับได้อย่างดี. ปัจจุบัน กลไกความใส่ใจ เป็นศาสตร์และศิลป์ของการเรียนรู้เชิงลึก โดยเฉพาะการกิจเกี่ยวกับการประมวลผลภาษาธรรมชาติ มีการประยุกต์ใช้อย่างกว้างขวาง (รวมถึงการกิจนอกเหนือจากการประมวลผลภาษาธรรมชาติทั่วไปด้วย เช่น [221]) และเป็นแรงบันดาลใจให้เกิดการพัฒนาอย่างต่อเนื่อง จนเป็นแนวทางของตัวแปลง (transformer[203, 40, 24]).

9.7 อภิธานศัพท์

โครงข่ายประสาทเวียนกลับ (Recurrent Neural Network คำย่อ RNN): โครงข่ายประสาทเทียม ที่โครงสร้างการคำนวนมีการเชื่อมต่อ ที่นำค่าที่คำนวนแล้วในลำดับเวลา ก่อนกลับเข้ามาคำนวนในลำดับเวลาปัจจุบันด้วย.

แผนภาพคลี่ลำดับ (unfolding diagram): แผนภาพโครงสร้างของโครงข่ายประสาทเทียม ที่รายจายการแสดงการเวียนกลับเชิงลำดับเวลา ออกเป็นลักษณะเดียวกับโครงสร้างตระหง่านภาษาพ. แผนภาพคลี่ลำดับ นิยมใช้แสดงการเชื่อมต่อของโครงข่ายประสาทเวียนกลับ.

การแพร่กระจายย้อนกลับผ่านเวลา (backpropagation through time คำย่อ BPTT): ขั้นตอนวิธี การคำนวนเกรเดียนต์ สำหรับโครงข่ายประสาทเวียนกลับ.

ปัญหาการระเบิดของเกรเดียนต์ (exploding gradient problem): ปัญหาที่อาจพบกับการฝึกโครงข่ายประสาทเวียนกลับ ที่เกรเดียนต์มีค่าเพิ่มขึ้นอย่างมาก เมื่อเวียนกลับย้อนกลับลำดับเวลา.

การเล้มเกรเดียนต์ (gradient clipping): วิธีแก้ปัญหาการระเบิดของเกรเดียนต์ ด้วยวิธีจำกัดขนาดของ เกรเดียนต์ ที่จะใช้คำนวนปรับค่าน้ำหนัก.

โครงข่ายประสาทเวียนกลับสองทาง (bidirectional recurrent neural network คำย่อ BRNN): โครงข่ายประสาทเวียนกลับ ที่ใช้นำค่าสถานะทั้งในลำดับก่อนหน้า (ทิศทางปกติ) และลำดับหลัง (ทิศทางย้อนกลับ) เวียนมาคำนวนผลลัพธ์ในลำดับปัจจุบัน.

แบบจำลองความจำระยะสั้นที่ยาว (long short-term memory model คำย่อ LSTM): แบบจำลอง โครงข่ายประสาทเวียนกลับ ที่มีการใช้กลไกของประตุความคุมการปรับเปลี่ยนค่าสถานะ และค่าผลลัพธ์ของหน่วยคำนวน เพื่อช่วยเพิ่มประสิทธิผลในการรู้จำความสัมพันธ์ระยะยาว ของข้อมูล.

ช่องแออบมอง (peephole connections): กลไกเพิ่มเติม สำหรับแบบจำลองความจำระยะสั้นที่ยาว เพื่อช่วยให้การควบคุมเปลี่ยนค่าสถานะและค่าผลลัพธ์ของหน่วยคำนวน ทำได้แม่นยำมากขึ้น โดยนำค่าสถานะเข้าไปเป็นส่วนหนึ่งในการพิจารณาการเปิดปิดของประตุด้วย.

สถาปัตยกรรมตัวเข้ารหัสตัวคอดรหัส (encoder-decoder architecture): โครงสร้างการต่อเชื่อม ที่ใช้โครงข่ายประสาทเวียนกลับสองตัวต่อกัน โดยให้ตัวหนึ่งประมวลผลชุดลำดับอินพุต และอีกตัวประมวลผลชุดลำดับเอาร์พุต สำหรับการกิจจำลองแบบชุดลำดับเป็นชุดลำดับ.

การประมวลผลภาษาธรรมชาติ (Natural Language Processing คำย่อ NLP): ศาสตร์ที่ใช้วิธีการต่าง ๆ เพื่อให้คอมพิวเตอร์สามารถนำข้อมูลความในภาษาธรรมชาติไปประมวลผล และให้ผลลัพธ์ตามจุดประสงค์ของการกิจที่ต้องการ.

โทเค็น (token): หน่วยพื้นฐานของภาษาที่มีความหมาย เช่น โทเค็น อาจหมายถึง คำ.

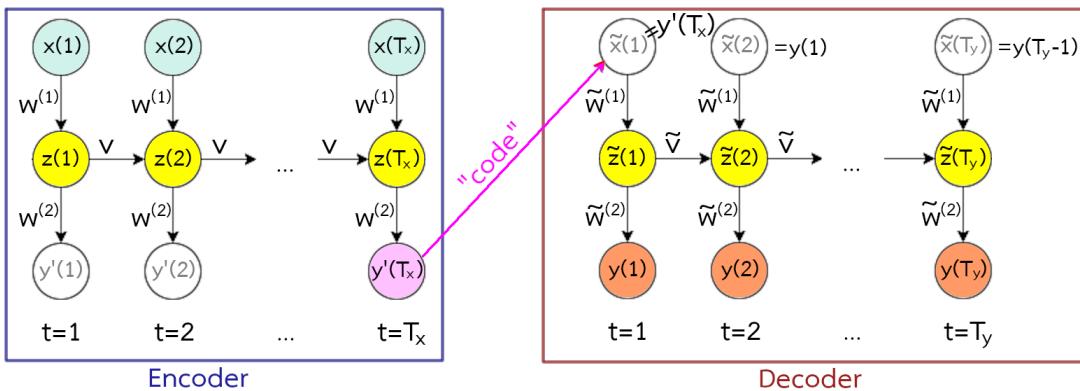
ไวยากรณ์ (syntax): กฎเกณฑ์ที่เกี่ยวกับโทเค็น และโครงสร้าง.

การแยกส่วน (parsing): การวิเคราะห์โครงสร้างไวยากรณ์ของข้อความหรือประโยค.

การระบุหมวดคำ (Part-Of-Speech Tagging): การกิจที่ระบุว่า คำต่าง ๆ ในข้อความ แต่ละคำอยู่ในหมวดคำใด จากหมวดคำ เช่น นาม สรรพนาม กริยา วิเศษณ์ คุณศัพท์ สันธาน บุพบท อุทาน.

การกิจจำลองแบบชุดลำดับเป็นชุดลำดับ (sequence-to-sequence modeling task): การกิจที่ทั้งอินพุตและเอาต์พุตเป็นชุดลำดับ แต่จำนวนลำดับของชุดอินพุต อาจไม่เท่ากับจำนวนลำดับในชุดเอาต์พุต

กลไกความสนใจ (attention mechanism): กลไกสำหรับการกิจจำลองแบบชุดลำดับเป็นชุดลำดับ เพื่อช่วยบรรเทาปัญหา เมื่อทำงานกับชุดลำดับที่ยาว.



รูปที่ 9.18 แผนภาพคลื่นลำดับของสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส. สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส โดยใช้ เอกลักษณ์จากตัวเข้ารหัส 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. ถ้าคุณคุยกับ เขายาด้วยภาษาของเขา สิ่งที่คุณพูดจะเข้าไปใน ใจเขา.”

—เนลสัน เมนเดลา

แบบฝึกหัด 9.1

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายประสาทเทียม เช่น เพอร์เซptron หลักยั่งยืน ที่ไม่มีกลไกการเรียนกลับ กับข้อมูลเชิงลำดับ พิริมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่าย ประสาทเทียมเรียนกลับ.

แบบฝึกหัด 9.2

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายคอนโวลูชัน ที่ไม่มีกลไกการเรียนกลับ กับข้อมูลเชิงลำดับ พิริมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่ายประสาทเทียมเรียนกลับ.

แบบฝึกหัด 9.3

จงอภิปรายถึง (1) แนวทางต่าง ๆ เพื่อประยุกต์ใช้โครงข่ายประสาทเรียนกลับ กับข้อมูลที่มีความ สัมพันธ์เชิงท้องถิ่นหลักมิติ เช่น ข้อมูลภาพ พร้อมอภิปรายข้อดีและข้อเสีย สำหรับแนวทางต่าง ๆ ที่เสนอมา พร้อมยกตัวอย่างให้เห็นภาพ (2) ข้อดีและข้อเสียเปรียบเทียบกับการใช้โครงข่ายคอนโวลูชัน.

แบบฝึกหัด 9.4

รูปที่ 9.18 แสดงการจัดโครงสร้างสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส โดยใช้เอาร์พุตจากตัวเข้ารหัส นำมาเป็นอินพุตสำหรับตัวถอดรหัส

จงอภิราย ข้อดี ข้อเสีย และปัญหาของการจัดโครงสร้างแบบนี้ เปรียบเทียบแบบใช้สถานะภายใน เป็นรหัส (รูปที่ 9.13)

แบบฝึกหัด 9.5

นอกจาก การจัดโครงสร้างดังแสดงในรูปที่ 9.13 แล้วสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส อาจ ผ่านรหัสช้อความ ไปเป็นส่วนหนึ่งของอินพุตของตัวถอดรหัสได้ (รูปแบบที่สอง) หรือ แม้แต่จะผ่านรหัส ช้อความ ไปเป็นทั้งสถานะเริ่มต้น และส่วนหนึ่งของอินพุต ดังแสดงในรูปที่ 9.16 (รูปแบบที่สาม)

จงอภิราย ข้อดี ข้อเสีย การจัดโครงสร้างแบบต่าง ๆ นี้ รวมถึงอภิรายปัจจัยที่เกี่ยวข้อง สถานการณ์ที่บางรูปแบบอาจทำงานได้ดีกว่า พร้อมออกแบบการทดลอง ดำเนินการทดลอง และนำเสนอผล.

แบบฝึกหัด 9.6

จงศึกษาการกิจการจำแนกอารมณ์ แนวทางปฏิบัติ การวัดผล และข้อมูลที่นิยม และสร้างระบบ การจำแนกอารมณ์ พร้อมประเมินผล.

แบบฝึกหัด 9.7

จงศึกษาการกิจการระบุหมวดคำ แนวทางปฏิบัติ การวัดผล และข้อมูลที่นิยม และสร้างระบบการ จำแนกอารมณ์ พร้อมประเมินผล.

แบบฝึกหัด 9.8

จงศึกษาการทำงานของระบบสังเคราะห์เสียง เช่น เวฟเน็ต (Wavenet[200]) อภิรายถึงแนวทาง และวิธีที่ใช้ รวมถึงการประเมินผลและข้อมูล.

แบบฝึกหัด 9.9

จงศึกษาสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส (อาจเริ่มจากบทความที่ทรงอิทธิพล[37, 193]) ออกแบบการทดลอง เพื่อศึกษาปัจจัยความพยายามของชุดลำดับข้อมูลกับประสิทธิภาพของระบบ ดำเนินการทดลอง รายงานผล และสรุป.

แบบฝึกหัด 9.10

จงรวมกลุ่มระดมความคิด และอภิรายแนวทางที่จะแก้ปัญหาระบบแปลภาษา เพื่อแก้ปัญหาของ ขวดในสถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส

หมายเหตุ แบบฝึกหัดนี้ ต้องการฝึกการคิดเชิงวิพากษ์ และฝึกความคิดสร้างสรรค์ อีกทั้งจะทำให้เห็นคุณค่าของกลไกความใส่ใจด้วย แต่เพื่อให้แบบฝึกหัดนี้ บรรลุจุดประสงค์ได้ ควรทำแบบฝึกหัดนี้ ก่อนที่จะศึกษาเรื่องกลไกความใส่ใจ หรือหากได้ศึกษาเรื่องกลไกความใส่ใจไปแล้ว อาจจะลองเปิดใจมองหาแนวทางอื่น ๆ ที่อาจจะช่วยบรรเทาปัญหานี้ได้.

แบบฝึกหัด 9.11

จงศึกษาพัฒนาการและกลไกที่สำคัญของศาสตร์แบบจำลองชุดข้อมูลลำดับ จากโครงข่ายประสาท เกียนกลับ แบบจำลองความจำระยะสั้นที่ยาว สถาปัตยกรรมตัวเข้ารหัสตัวอัตโนมัติ กลไกความใส่ใจ และตัวแปลงชนิดต่าง ๆ (เช่น Transformer, BERT, GPT-3) สรุปประเด็น แนวทางที่สำคัญ ลักษณะ ปัญหา และการประยุกต์ใช้ รวมถึงช่วงเวลา และอภิราย เหตุผล แรงขับดันเบื้องหลังพัฒนาการเหล่านี้.

แบบฝึกหัด 9.12

จงศึกษาศาสตร์การประมวลผลภาษาธรรมชาติ ในเชิงวิเคราะห์ ถึงการกิจต่าง ๆ ที่สำคัญ และบริบท ในแง่ความต้องการของสังคม รวมถึง ความก้าวหน้าในแต่ละภารกิจ เมื่อเทียบกับจุดมุ่งหมาย และ อภิรายโอกาสการประยุกต์ใช้ต่าง ๆ ที่อาจจะนอกเหนือจากประยุกต์ใช้ดั้งเดิม และความท้าทายต่าง ๆ ในงานวิจัย เพื่อบรรลุจุดประสงค์ รวมถึงเชื่อมโยงสิ่งที่ได้เรียนรู้ ความก้าวหน้า การประยุกต์ใช้ จาก บริบทในแง่ความต้องการของสังคม.

แบบฝึกหัด 9.13

จงเลือกภารกิจการประมวลผลภาษาธรรมชาติที่สนใจ และศึกษาภารกิจ แนวทางปฏิบัติ ปัจจัย ที่เกี่ยวข้อง การวัดผล และข้อมูลที่นิยม และทดลองสร้างระบบสำหรับภารกิจที่เลือก ประเมินผล ให้ ความเห็น และสรุปสิ่งที่ได้เรียนรู้.

បរទាន់ក្រម

- [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, 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., Vaicilius, 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. សមោងແහេងពុទធស, 1 ed. ឯមវិនទ័រររមេ, 378 ត.ខេត្តពុំកែវ តិចិន្ទី ក្រុងពេទ្យ 10170, ព.ស. 2557. ແណ្តល់ដូយ ផែវ សយាមវាតា ឱក 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, 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, 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., Bretschneider, H., Merico, D., Yuen, R. K. C., Hua, Y., Gueroussov, S., Najafabadi, H. S., Hughes, T. R., Morris, Q., Barash, Y., Krainer, A. R., Jovicic, 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.
- [228] รัชพงศ์ กตัญญกุล. รายงานสรุปโครงการวิจัยการวิเคราะห์พฤติกรรมลูกค้าผ่านข้อมูลวิดีโอจากกล้องวงจรปิด. Tech. rep., คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น, 2016.

ธรรมนิภาษาไทย

- กฎของความน่าจะเป็นรวม, 48
- กฎของเบส์, 51, 426
- กฎผลคูณ, 51
- กฎผลรวม, 51
- กฎลูกโซ่ของความน่าจะเป็น, 51
- กราฟอาร์โอี, 198
- กลไกความใส่ใจ, 475, 476, 482
- กล่องขอบเขต, 226, 250
- กล่องสมอ, 423
- การกำจัดการระบุชี้ช้อน, 226, 250
- การกำจัดความชี้ช้อน, 226
- การกำหนดค่า้น้ำหนักเริ่มต้น, 139, 163, 281
- วิธีเซเวียร์, 282
- การกำหนดค่า้น้ำหนักเริ่มต้นด้วยการสุ่มโปรแกรม, 162
- การกำหนดค่า้น้ำหนักเริ่มต้นด้วยวิธีเหยี่ยนวิดโดยร์
- โปรแกรม, 164
- การกำหนดค่าเริ่มต้น, 77
- การคำนวนเมทริกซ์, 33
- การคูณเมทริกซ์, 33
- คุณสมบัติ, 33
- การคูณแบบตัวต่อตัว, 33, 470
- การค้นหายา, 201
- การจัดกลุ่มข้อมูล, 402
- เค-มีนส์, 402
- การจัดการกับข้อมูลขาดหาย, 180
- การจัดถุง, 279
- การจับคู่ลักษณะสำคัญ, 415, 424
- การจำลองแบบชุดลำดับเป็นชุดลำดับ, 473
- การจำแนกกลุ่ม, 13, 17, 140
- การจำแนกค่าทวิภาค, 140
- การจำแนกภาพ, 398
- การตอกอก, 292
- การตรวจจับภาพวัตถุ, 398
- วิธีหน้าต่างเลื่อน, 218
- โยโล', 398
- การตรวจจับวัตถุในภาพ, 397
- การตรวจหารวัตถุ, 248
- การตรวจจับวัตถุในภาพ, 422
- โยโล', 422
- การถอดถอนไวลูชั่น, 423
- การถ่ายโอนการเรียนรู้, 290
- การทดสอบนัยสำคัญ, 195
- การทำลายกราบรื่น, 417, 424, 426, 427
- การทำลายกราบรื่นทางเดียว, 417
- การทำช้ำ, 165
- การทำอร์มอลไลซ์, 142
- การทำอร์มอลไลซ์อินพุต, 168
- โปรแกรม, 169
- การทำหมู่เล็กเสมือนจริง, 417, 424
- การทำเรกูลาร์ซี, 118, 155
- ค่าน้ำหนักเสื่อม, 118
- การทำเหมืองข้อมูล, 15
- การประมวลผลภาษาธรรมชาติ, 454, 482
- การจำแนกอารมณ์, 484
- การระบุหมวดคำ, 484
- การประสานการเรียนรู้, 279
- การประเมินผล
- กราฟการทำนายกับค่าเฉลย, 172

- กราฟอร์โอดี, 198
 การทดสอบนัยสำคัญ, 195
 การระลึกกลับ, 179
 ความเที่ยงตรง, 179
 ความแม่นยำ, 178
 คะแนนเอฟ, 179
 ค่าความแม่นยำ, 117
 พื้นที่ใต้เส้นโค้ง, 198
 เมทริกซ์ความสัมสูน, 178
 การปรับละเอียด, 289
 การปรับส่วนค่าน้ำหนัก, 277
 การปรับเส้นโค้ง, 151
 ฟังก์ชันพหุนามระดับขั้นได ๆ, 154
 การปรับเส้นโค้งด้วยฟังก์ชันพหุนาม, 155
 การฝึก, 109, 130, 151
 การกำหนดค่าน้ำหนักเริ่มต้น, 139
 หมู่, 138
 ออนไลน์, 138
 โครงข่ายประสาทเทียม, 130
 การฝึกก่อน, 268, 289, 403
 การฝึกทีละหมู่เล็ก
 ขนาดของหมู่เล็ก, 274
 การฝึกแบบหมู่, 138
 การฝึกแบบหมู่เล็ก, 300
 การฝึกแบบออนไลน์, 138
 การพังทลายของภาวะ, 423
 การระบุหมวดคำ, 432, 455, 472, 482
 การระลึกกลับ, 179
 การรู้จำตัวเลขลายมือ, 10, 17, 184
 การรู้จำรูปแบบ, 4, 12, 17
 การถูเข้า, 66, 77
 การสกัดลักษณะสำคัญ, 248
 การสลับเปลี่ยน, 30
 การสร้างปัจจัย, 51, 76, 440
 การหยุดก่อนกำหนด, 144, 209
 การหาค่าดีที่สุด, 59, 61, 76
 ภาวะคู่กัน, 100
 วิธีลงเกรเดียนต์, 64
 แบบมีข้อจำกัด, 69
 การหาค่าดีที่สุดแบบมีข้อจำกัด, 69
 แนวทางการลงโทษ, 71
 แนวทางการแปลงมุมมอง, 71
 การหาค่าลดตอน, 13, 17, 131
 การหาเกรเดียนต์เชิงเลข, 210
 การอันเดอร์พิต, 300
 การเขียนโปรแกรมเชิงเลข, 83
 การเฉลี่ยตามประวัติ, 417
 การเติมเต็ม, 350
 การเติมเต็มด้วยศูนย์, 350
 การเรียนรู้, 109, 151
 การเรียนรู้การแนะนำสินค้า, 14
 การเรียนรู้ของเครื่อง, 4, 6, 17
 การเรียนรู้เชิงลึก, 16, 267, 292
 การเรียนรู้แบบบกี๋มีผู้ช่วยสอน, 14, 406
 การเรียนรู้แบบบกี๋ช่วยสอน, 13
 การเรียนรู้แบบบกี๋สอน, 17
 การเรียนรู้แบบเสริมกำลัง, 14
 การเรียนรู้แบบไม่มีผู้สอน, 14, 17, 406
 การเรียนหลักสูตร, 291
 การเลือกแบบจำลอง, 114, 157
 การเลื่อนของความแปรปรวนร่วมเกี่ยวภายใน,
 287
 การเลี้มเกรเดียนต์, 464, 481
 การแจกส่วน, 454, 482
 การแจกแจง
 ปกติ, 59

- ເກາສ්ເෂියන, 59
- ກາຣແຈກແຈງປກຕີ, 59
- ກາຣແຈກແຈງເກາສ්ເෂියන, 59
- ກາຣແຈກແຈງເອກຮູປ, 168
- ກາຣແບ່ງຂໍ້ມູນ
- ໂປຣແກຣມ, 163
- ກາຣແປລັງເຊີງເສັ້ນ, 39
- ກາຣແພ່ວ່າວ, 41
- ກາຣແພຣ່ກະຈາຍຢ້ອນກັບ, 134
- ໂປຣແກຣມ, 161
- ກາຣແພຣ່ກະຈາຍຢ້ອນກັບຜ່ານເວລາ, 459, 481
- ກາຣແຍກແຍະໜູ່ເລື້ອກ, 416, 424
- ກາຣໃສ່ສັ້ນຢູ່ນົມບານຮຽນກວນ, 418
- ຂນາດກ້າວ, 65
- ຂນາດກ້າວຍ່າງ, 351
- ຂນາດຂົບເລື່ອນ, 218, 248
- ຂນາດຂອງໜູ່ເລື້ອກ, 274
- ຂອບເຂດຂອງກາຣແບ່ງ, 235, 251
- ຂອບເຂດຕັດສິນໃຈ, 248, 251
- ຂັ້ນຕອນວິວິທີເອັມ, 440, 448
 - ແບບຈຳລອງມາຮົກພ້ອນເຮັນ, 440, 448
- ຂັ້ນຕອນວິວິທີແລ້ວພາ-ປີຕາ, 443
- ຂັ້ນຕອນວິວິທີໄປໜ້າທີ່ກັບຄອຍກັບ, 443
- ຂໍ້ຈຳກັດທີ່ທໍາງນານ, 238
- ຂໍ້ມູນ
 - ຕຽບສອບ, 145
 - ທດສອບ, 115
 - ຝຶກ, 115
- ຂໍ້ມູນຂາດໄຫຍ່
 - ປັນຫາ, 300
- ຂໍ້ມູນຊຸດຕຽບສອບ, 120
- ຂໍ້ມູນຕຽບສອບ, 145
- ຂໍ້ມູນທດສອບ, 115, 151
- ຂໍ້ມູນນຳອອກ, 17, 108
- ຂໍ້ມູນນຳເຂົາ, 17, 108
- ຂໍ້ມູນຝຶກ, 115, 151
- ຂໍ້ມູນມ້ທັດ, 16
- ຂໍ້ມູນເຊີງລຳດັບ, 431, 448
- ຂໍ້ມູນເຊີງລຳດັບແບບຄົງທີ່, 433
- ຄຣອສວາລີເດືອນ, 120
- ຄວາມໜັບໜັນຂອງແບບຈຳລອງ, 115, 152
- ຄວາມນ່າຈະເປັນ, 44, 45, 75
 - ພລລັພົງ, 75
 - ເໜັກກາຣົນ, 75
 - ແບບມືເຈື່ອນໄຂ, 50
 - ໄມ້ມີສ່ວນຮ່ວມກັນ, 48
- ຄວາມນ່າຈະເປັນກ່ອນ, 76
- ຄວາມນ່າຈະເປັນຂອງກາຣປລ່ອຍ, 438, 448
 - ແບບຈຳລອງມາຮົກພ້ອນເຮັນ, 438, 448
- ຄວາມນ່າຈະເປັນຂອງກາຣປລື່ຍນສຕານະ, 438, 448
 - ແບບຈຳລອງມາຮົກພ້ອນເຮັນ, 438, 448
- ຄວາມນ່າຈະເປັນກາຍທັງ, 76
- ຄວາມນ່າຈະເປັນຮ່ວມ, 50
- ຄວາມນ່າຈະເປັນແບບມືເຈື່ອນໄຂ, 50, 76
- ຄວາມພິດປົກທາງບຸກລິກພາພແບບໜັງຕົວເອງ, 461, 462
- ຄວາມລຳເອີ່ຍງ, 145
- ຄວາມລຳເອີ່ຍງກັບຄວາມແປປປວນ, 145
- ຄວາມລຳເອີ່ຍງສູງ, 147
- ຄວາມລຶກຂອງໂຄຮງຂ່າຍ, 267
- ຄວາມເຖິງຕຽງ, 179
- ຄວາມເປັນອີສະເຫຼັນເສັ້ນ, 36
- ຄວາມແປປປວນ, 50, 78, 145
- ຄວາມແປປປວນຮ່ວມເກີຍວ, 50
- ຄວາມແປປປວນສູງ, 147
- ຄອນໄວລູ້ໜັກ້າຍາວ, 413

- ค่อนโวโลชั่นก้าวเศษ, 413, 423
 ค่อนโวโลชั่นสลับเปลี่ยน, 423
 คะแนนเอฟ, 179
 คำสาปของมิติ, 404
 คุณธรรม, 462
 คุณลักษณะ, 125
 คุณสมบัติความทั่วไป, 114, 151
 การหยุดก่อนกำหนด, 144
 ค่าความจำเพาะ, 198
 ค่าความน่าจะเป็นเริ่มต้น, 438, 448
 แบบจำลองมาร์คอฟช่องเร้น, 438, 448
 ค่าความเที่ยงตรง/ค่าการเรียกกลับ, 229
 ค่าความแม่นยำ, 117
 ค่าความไว, 198
 ค่าคาดหมาย, 50, 76
 ค่าทำให้น้อยที่สุด, 62, 76
 ท้องถิ่น, 63
 ค่าทำให้น้อยที่สุดท้องถิ่น, 63
 ค่าน้ำหนัก, 17, 124
 ค่าน้ำหนักเสื่อม, 118
 ค่าผิดปกติ, 200
 ค่าผิดพลาดของห้าชนิดอันดับสูงสุด, 376
 ค่าผิดพลาดชุดทดสอบ, 115
 ค่าผิดพลาดชุดฝึก, 115
 ค่าฟังก์ชันควรจะเป็นสูงสุด, 323
 ค่าลอกการทีมของฟังก์ชันควรจะเป็น, 324
 ค่าเฉลย, 109, 151
 ค่าเฉลี่ยความผิดพลาดกำลังสอง, 118
 ค่าเฉลี่ยค่าประมาณความเที่ยงตรง, 229, 250
 จำนวนบวกจริง, 178
 จำนวนบวกเท็จ, 178
 จำนวนลบจริง, 178
 จำนวนลบเท็จ, 178
 จิต, 134, 418
 จุดข้อมูล, 107, 151
 จุดที่ดีที่สุด, 69
 ฉลาก, 17
 ชั้นคำนวน, 127, 129
 ชั้นซ่อน, 129
 ชั้นดีคอนโวโลชั่น, 405
 ชั้นดึงรวม, 358
 ชั้นเชื่อมต่อเต็มที่, 347
 ชั้นเอาต์พุต, 131
 ชีวิต, 418
 ชุดข้อมูล
 การจับตัวกับโปรดีน, 188
 การรู้จำภาพตัวเลขลายมือเจียน, 184
 ข้อมูลภาพເອັກຊາຣຍ໌ແຕ້ານມ, 174
 ຍອຊົ່ວ, 169
 ເອມນິສຕີ, 11, 184
 ชຸດຂໍ້ມູນໂລດສໍາເຮົາ
 MNIST, 316
 ชຸດມິຕີ, 346
 ชຸດລຳດັບມິຕີ, 346
 ช່ອງແອບມອງ, 471, 481
 ซอฟຕີແມກໜີ, 105
 ໜັກພອຣຕົວເຕອຣ໌, 235
 ໜັກພອຣຕົວເຕອຣ໌ແມ່ຫືນ, 233, 250, 251, 253,
 257, 259, 262
 ໜັກພອຣຕົວເຕອຣ໌, 251
 ປັບປຸງຫາປຸ່ມ
 ໂປຣແກຣມ, 255, 259
 ຟັງກົງໜັນເຄອຣ໌ນັລ, 251
 ຟັງກົງໜັນເຄອຣ໌ນັລເກາສ໌ເຊີຍນ, 251
 ຟັງກົງໜັນເຄອຣ໌ນັລເຊີງເສັ້ນ, 251
 ຟັງກົງໜັນເຄອຣ໌ນັລເກາສ໌ເຊີຍນ, 247

- ฟังก์ชันเครื่องเรียงเส้น, 247
 ลูกเล่นเครื่องเรียง, 251
 เก้าส์เชี่ยนเครื่องเรียง
 โปรแกรม, 263
 โปรแกรม, 262
 ซิกมอยด์, 154
 อนุพันธ์, 154
 ดีเทอร์มิเนนต์, 36
 ตัวประมาณค่าสากล, 129
 ตัวเข้ารหัสอัตโนมัติแบบเปลี่ยนแปลง, 404
 ตัวเข้าอัตโนมัติ, 225
 ตัวแปรตัดสินใจ, 61
 ตัวแปรที่ถูกสังเกต, 437
 ตัวแปรที่สังเกตได้, 448
 ตัวแปรสุ่ม, 49, 76
 ตัวแปรสุ่มต่อเนื่อง, 57, 76
 ตัวแปรสุ่มวิภาค, 76
 ทบทวน
 การหาค่าตีที่สุด, 59
 ความน่าจะเป็น, 44
 พีชคณิตเชิงเส้น, 29
 ทฤษฎีบทควรซึ่งคุณทักษะ, 97
 ทวิบของความจำเอียงกับความแปรปรวน, 147
 นอร์ม, 38, 416
 นอร์มอย่างอินพุต, 143
 บล็อกความจำ, 469
 บวกกึ่งແน่อน, 246
 บวกແน่อน, 246
 ประสบการณ์เฉียดตาย, 418
 ปริภูมิ
 การแผ่ทั่ว, 41
 ปริภูมิค่า, 31
 ปริภูมิตัวอย่าง, 45, 75
 ปริภูมิอย่าง, 41
 ปริภูมิลักษณะสำคัญ, 233, 250
 ปัญญา, 461
 ปัญญาประดิษฐ์, 15
 ปัญหาการระเบิดของเกรเดียนต์, 464, 481
 ปัญหาการเลื่อนหายของเกรเดียนต์, 269, 292,
 293, 464
 ปัญหาค่าน้อยที่สุด, 61
 ปัญหาค่ามากที่สุด, 61
 ปัญหาที่สามารถแบ่งแยกได้เชิงเส้น, 126
 ปัญหามอนตีร์โอล, 56
 ผลต่างเซต, 45
 ผลบวกผิด, 55
 ผลรวมเชิงเส้น, 36
 พารามิเตอร์, 17
 อภิมาน, 66
 พิกเซลอาร์ເອນເອນ, 403
 พีชคณิตเชิงเส้น
 ระบบสมการ, 33
 พีชคณิตเวกเตอร์, 410
 พีชคณิตเชิงเส้น, 29
 พื้นที่ได้เส้นโค้ง, 198
 พังก์ชัน
 ซอฟต์แมกซ์, 105
 ซิกมอยด์, 131
 พังก์ชันกราฟต์, 125
 ฟังก์ชันเครื่องหมายอ่อน, 282
 เรคติไฟเดลเนียร์, 292
 เรลู, 292
 พังก์ชันกราฟต์, 131
 ฟังก์ชันซอฟต์แมกซ์, 141
 ฟังก์ชันซิกมอยด์, 140
 ฟังก์ชันเอกลักษณ์, 131

- พังก์ชันการแจกแจง, 49, 76
 พังก์ชันการแจกแจงสะสม, 49
 พังก์ชันขั้นบันไดหนึ่งหน่วย, 100, 125
 พังก์ชันควรจะเป็น, 76, 448
 พังก์ชันความสูญเสีย, 62
 พังก์ชันความหนาแน่นความน่าจะเป็น, 76
 พังก์ชันค่าใช้จ่าย, 62
 พังก์ชันจำกัดแข็ง, 125
 โปรแกรม, 158
 พังก์ชันจุดประสงค์, 61, 76
 พังก์ชันซอฟต์แแมกซ์, 105, 141
 พังก์ชันซิกมอยด์, 85, 131
 พังก์ชันบางอ่อน, 80, 85
 พังก์ชันพลังงาน, 62
 พังก์ชันพหุนาม, 108, 151
 พังก์ชันพหุนามระดับขั้นได ๆ, 154
 พังก์ชันมวลความน่าจะเป็น, 49, 76
 พังก์ชันลงโทษ, 72
 พังก์ชันสูญเสีย, 76
 ครอบเงอนໂທປີ, 140
 ค่าเฉลี่ยค่าผิดพลาดกำลังสอง, 131
 อนຸພັນຮັບ, 155
 พังก์ชันสูญเสียครอบเงอนໂທປີ, 140
 พังก์ชันເກາສ්ເຊියන්ස්, 85
 พังก์ชันเครื่องหมายอ่อน, 282
 พังก์ชันເຄອງນັລ, 239, 251
 พังก์ชันເຄອງນັລເກາສ්ເຊියන, 251
 พังก์ชันເຄອງນັລເຊີງເສັ້ນ, 251
 พังก์ชันເຄອງນັລເກາສ්ເຊියන, 247, 251
 พังก์ชันເຄອງນັລເຊີງເສັ້ນ, 247, 251
 พังก์ชันແບ່ງແຍກ, 226, 248
 ພິລເຕົອຣີ, 349
 ກາຮກິຈ
- การจำแนกกลุ่ม, 140
 การจำแนกค่าทวิภาค, 140
 การหาค่าถดถอย, 131
 ກາຮກິຈຈຳລອງແບບຊຸດລຳດັບເປັນຊຸດລຳດັບ, 482
 ກາຊາຮຣມໜາຕີ, 453
 ມິຕີ, 31, 75, 125, 346
 ມິຕີປະກຸມືມີຕໍາ, 32
 ຍູ້ເນີຍນ, 45
 ຮහສຫີ້ຮ້ອນ, 140
 ຮະດັບຄ່າຂຶ້ນແບ່ງ, 196
 ຮະບບສມກາຣ, 33
 ຮະບບສມກາຣເຊີງເສັ້ນ, 78
 ຮະບບແຕ່ງເພັລງວັດໂນມັຕີ, 450
 ຮະຍະທາງຢຸກລື້ເດືອຍນ, 38
 ຮະເບີຍນ, 169
 ຮາກທີ່ສອງຂອງຄ່າເນີລີ່ຍຄວາມຜິດພລາດກຳລັງສອງ, 118
 ຮາຍງານຜລ
 ເວລານອ້ຣມ່ໄລ້ຈົດ, 144
 ຮູບແບບ, 4, 17
 ລາກຮານຈົ່ງພາຣາມີເຕົອຣີ, 72
 ລຳດັບຂັ້ນ, 31, 32, 346
 ລຳດັບຂັ້ນຂອງຄວາມຄິດ, 311
 ລຸກເລ່ນເຄອງນັລ, 246, 251
 ວິທຍາກາຣຂໍ້ມູລ, 16
 ວິທີກາຣປະມານຄວາມໜາແນ່ນແກ່ນ, 227, 250, 252, 253
 ໂປຣແກຣມ, 253
 ວິທີກາຣຝຶກແບບປປັກົງ, 406
 ວິທີກາຣລົງໂທ, 71, 80
 ວິທີຄ່າພັງກົນຄວາມຈະເປັນສູງສຸດ, 440
 ແບບຈຳລອງມາຮົກໂພ່ອນເຮັ້ນ, 440
 ວິທີຮະຈັບຄ່າໄມ່ມາກສຸດທ້ອງຄືນ, 250

- วิธีลงเกรเดียนต์, 64, 77
 โปรแกรม, 255
 วิธีลงเกรเดียนต์สโทแคสติก, 283
 วิธีหน้าต่างเลื่อน, 216, 218, 248
 วิธีเชเวียร์, 282
 สติปัญญา, 74
 สถาปัตยกรรมตัวเข้ารหัสตัวถอดรหัส, 473, 481
 สนามรบรู้, 357
 สมการกำลังสอง, 43
 สมการพุนาม, 43
 สมองมนุษย์, 112
 สมัย, 138
 สัญกรณ์จุดๆ, 30
 สัดส่วนข้อมูลไม่สมดุล, 188, 403
 สเกลเทา, 184
 สเกลาร์, 29
 ส่วนเติมเต็ม, 45
 หน่วยคำนวน, 127
 หน่วยซ่อน, 129
 หน่วยเวียนกลับมีประตุ, 472
 หมู่เด็ก, 272, 292
 อภิมานพารามิตอร์, 66, 114, 128
 อภิรานاب, 233, 251
 อัตตา, 461
 อัตราการตรวจจับได้, 198
 อัตราสัญญาณหลอก, 198
 อัตราเรียนรู้, 138
 อันเดอร์พิต, 147
 อินพุต, 17, 108
 คุณลักษณะ, 125
 มิติ, 125
 อินเตอร์เซกชัน, 45
 อิมเมจเนต, 375
 อเล็กซ์เน็ต, 375
 เกร็ด
 การค้นหายา, 201
 จิต, 134
 จิต สมอง และชีวิต, 418
 มะเร็งและยารักษา, 7
 สติปัญญา, 74
 สมองมนุษย์, 112
 เซลล์ประสาท, 122
 เมตตา, 461
 เกร็ดความรู้, 7, 74, 112, 122, 134, 201, 418,
 461
 เขตข้อมูล, 169
 เค-มีนส์, 402
 เคอร์เนล, 349
 เงื่อนไข
 ทำงาน, 98
 ไม่ทำงาน, 98
 เงื่อนไขการจบ, 67
 เขต, 44
 ยูเนียน, 45
 อินเตอร์เซกชัน, 45
 เขตย้อย, 44
 เขตว่าง, 44
 เซลล์ความจำ, 469
 เซลล์ประสาท, 122
 เทคนิคกล่องสมอ, 399
 เทนเซอร์, 32
 เนสเตอร์ฟูเมนต์, 284
 เพอร์เซปตรอน, 123
 เพอร์เซปตรอนหลายชั้น, 123
 เมตตา, 461
 เมทริกซ์, 30

- การคำนวณ, 33
 การคูณ, 33
 การบวก, 33
 ดีเทอร์มิแนนต์, 36
 เมทริกซ์การเปลี่ยนสถานะ, 438
 แบบจำลองมาร์คอฟช่องเร้น, 438
 เมทริกซ์ความสับสน, 178
 เรคติไฟเดลเนียร์, 269
 เรลู, 254, 269, 294, 300, 415
 โปรแกรม, 255
 เรลูรั่ว, 415
 เรเดียวเบซิส, 154
 อนุพันธ์, 154
 เวกเตอร์, 29
 ตั้งฉาก, 39
 เวกเตอร์ตั้งฉาก, 39
 เวกเตอร์หนึ่งหน่วย, 38
 เวคเตอร์ไซเซ็น, 129
 เวลา Narom ไลช์ต์, 144
 เส้นโค้งเรียนรู้, 145, 147
 เหตุการณ์, 45
 เอชโอดี, 221
 เอนโทรปี, 78
 เออมนิสต์, 11, 17, 184
 เอ่าต์พุต, 17, 108
 เอ่าต์พุตจริง, 109
 แบบความเชื่อมั่น, 167
 แบบอร์ม, 287, 292, 415
 โครงข่ายคอนโวลูชัน, 289
 โปรแกรม, 338
 แบบจำลอง, 12, 17, 107, 108
 การทำนาย, 12
 การอนุมาน, 12
 การแปลงค่า, 12
 แบบจำลองความจำรยะสั้นที่ยาว, 472
 ช่องแออบมอง, 471, 481
 บล็อกความจำ, 469
 เชลล์, 469
 แบบจำลองความระยะสั้นที่ยาว, 469, 481
 แบบจำลองความหนาแน่นผสม, 324
 แบบจำลองปริภูมิสถานะ, 437
 แบบจำลองมาร์คอฟ, 433, 448
 แบบจำลองมาร์คอฟช่องเร้น, 437, 448
 ขั้นตอนวิธีไปข้างหน้ากับถอยกลับ, 443
 ขั้นตอนวิธีแอลฟ่า-บีตา, 443
 ฟังก์ชันการปล่อย
 อnen นามวิยุต, 442
 เก้าส์เชียน, 442
 แบบจำลองสร้างกำเนิด, 225, 248
 แบบจำลองแบ่งแยก, 225, 248
 แผนที่ความร้อน, 227
 แผนที่ลักษณะสำคัญ, 357
 แผนภาพคลื่นลำดับ, 457, 481
 โครงสร้างวิเคราะห์พฤติกรรมลูกค้า, 213
 โครงข่ายก่อกำเนิด, 406, 423
 โครงข่ายคอนโวลูชัน, 345, 387
 โครงข่ายปรัปักษ์เชิงสร้าง, 405, 423
 การจับคู่ลักษณะสำคัญ, 415, 424
 การพังทลายของภาวะ, 413, 423
 การแยกแยะหมู่เด็ก, 416
 ปริภูมิช่องเร้น, 423
 พีชคณิตเวกเตอร์, 423
 ลักษณะช่องเร้น, 410, 423
 โครงข่ายก่อกำเนิด, 406, 423
 โครงข่ายแบ่งแยก, 406, 423
 โครงข่ายปรัปักษ์เชิงสร้างกำเนิด, 225

โครงข่ายปรัชญาเชิงสร้างแบบมีเงื่อนไข, 409

โครงข่ายประชาทเทียม, 123

การทำรัมอල์, 142

การฝึก, 130, 158

ตระกะเอ็กซ์อร์, 157

โปรแกรม, 158

การตกออก, 319–321

คลาส, 299

ชั้นสัญญาณรบกวน, 323

ไฟฟอร์ซ, 302, 306

โครงข่ายประชาทเวียนกลับ, 455, 481

สถานะซ่อน, 456

โครงข่ายประชาทเวียนกลับสองทาง, 465, 481

โครงข่ายแบ่งแยก, 406, 423

โครงข่ายแพร์ಕราจายไปข้างหน้า, 129

โนเมนตัม, 284

โภคหลงตัวเอง, 461, 462

โนนด, 127

โนนดซ่อน, 129

โอบเวอร์ฟิต, 115, 152

การทำเรกุล่าเรซ์, 118

ไบอัส, 125

ไฟฟอร์ซ, 306

ไวยากรณ์, 454, 482

ไอ.ไอ.ดี., 324, 431, 433

ไอโอယ, 226, 250, 401

ไฮเปอร์บอลิกแทนเจนต์, 154

อนุพันธ์, 154

ទ្រចនីភាសាហ៉ងករូម

- accuracy, 117
- activation
 - leaky relu, 283
 - PReLU, 283
 - relu, 269
 - sigmoid, 131
 - softsign, 282
 - tanh, 131
- activation function, 125
- active constraint, 238
- adam, 268
- adversarial training, 406
- AlexNet, 375
- alpha-beta algorithm, 443
- anchor box, 399, 423
- Approaches to exact calculation issues
 - approximation, 433
 - constraining, 433
- Area Under Curve, 198
- artificial intelligence, 15
- Artificial Neural Network, 123
- artificial neural network
 - code, 158
 - class, 299
 - dropout, 319–321
 - noise layer, 323
 - pytorch, 302, 306
 - normalization, 142
 - train, 130, 158
 - XOR, 157
- attention, 268
- attention mechanism, 475, 476, 482
- AUC, 198
- Autoencoder, 225
- backpropagation, 134
 - code, 161
- backpropagation through time, 459, 481
- bagging, 279
- batch norm, 287, 292, 415
 - code, 338
 - convolutional neural network, 289
- batch normalization, 268
- batch training, 138
- Bayes' rule, 51, 426
- Bayes' theorem, 51, 426
- bias, 125
- bias (model behavior), 145
- bias-variance, 145
- Bias/Variance Dilemma, 147
- bidirectional recurrent neural networks, 465, 481
- big data, 16
- binary classification, 140
- Biological Neurons, 122
- bounding box, 226, 250
- built-in datasets
 - MNIST, 316
- chain rule of probability, 51
- classification, 13, 17, 140
- clustering, 402

- K-means, 402
- CNN, 268, 345, 387
- colon notation, 30
- compassion, 461
- complement, 45
- Conditional Generative Adversarial Networks,
 - 409
- conditional probability, 50, 76
- confidence intervals, 167
- confusion matrix, 178
- constrained optimization, 69, 71, 80
 - penalty, 71
 - projection, 71
- constraint
 - active, 98
 - inactive, 98
- continuous random variable, 57, 76
- convergence, 66, 77
- Convolutional Neural Network, 345, 387
- cost function, 62
- covariance, 50
- cross entropy
 - code, 188
- cross entropy loss, 140
- cross-validation, 120
- cumulative distribution function, 49
- curriculum learning, 291
- curses of dimensionality, 404
- curve fitting, 151
 - degree-M polynomial, 154
- customer behavior analytic project, 213
- data
 - test, 115
- training, 115
- validation, 145
- data mining, 15
- data science, 16
- data separation
 - code, 163
- datapoint, 107, 151
- dataset
 - mammography, 174
 - MNIST, 11, 184
 - protein binding, 188
 - yacht, 169
- DCGAN, 413
- decision boundary, 248, 251
- decision variable, 61
- deconvolution, 423
- deconvolution layer, 405
- deep learning, 16, 267, 292
- Degree-M polynomial, 154
- detection rate, 198
- determinant, 36
- dimension, 31, 75, 125, 143, 346
- discrete random variable, 76
- discriminant function, 226, 248
- discriminative model, 225, 248
- discriminator, 406, 423
- distribution
 - Gaussian, 59
 - normal, 59
- distribution function, 49, 76
- drop out, 292
 - break co-adaptation, 277
 - learn features more thoroughly, 277

- robustness, 277
- dropout, 268
- drug discovery, 201
- dual problem, 101
- early stopping, 144, 209
- Eigenvalues, 42
- Eigenvectors, 42
- element-wise product, 33, 470
- emission probabilities, 438, 448
 - HMM, 438, 448
- empty set, 44
- encoder-decoder architecture, 473, 481
- energy function, 62
- ensemble learning, 279
- entropy, 78
- epoch, 138
- Euclidean distance, 38
- evaluation, 115
 - accuary, 178
 - accuracy, 117
 - Area Under Curve, 198
 - AUC, 198
 - confusion matrix, 178
 - F-score, 179
 - MSE, 118
 - overfitting, 115
 - precision, 179
 - predict-groudtruth plot, 172
 - recall, 179
 - Receiver Operating Characteristic, 198
 - RMSE, 118
 - ROC, 198
 - significance test, 195
- test error, 115
- event, 45
- expectation, 50, 76
- expectation-maximization algorithm, 440, 448
- HMM, 440, 448
- expected value, 50
- exploding gradient problem, 464, 481
- F-score, 179
- false alarm rate, 198
- false negative, 178
- false positive, 55, 178
- feature, 125
- feature extraction, 248
- feature map, 357
- feature matching, 415, 424
- feature space, 233, 250
- feedforward network, 129
- field, 169
- filter, 349
- fine tuning, 289
- forward-backward algorithm, 443
- fractionally-strided convolution, 413, 423
- fully connected layer, 347
- function
 - sigmoid, 131
 - softmax, 105
- GAN, 268, 405, 423
 - Conditional GAN, 409
 - DCGAN, 413
 - discriminator, 406
 - feature matching, 415

- generator, 406
- latent representation, 410
- latent space, 410
- minibatch discrimination, 416
- mode collapse, 413
- representation space, 410
- Gaussian distribution, 59
- gaussian function, 85
- Gaussian kernel, 247
- gaussian kernel, 251
- generalization, 114, 151
 - early stopping, 144
- Generative Adversarial Network, 225, 268,
 - 405
- Generative Adversarial Networks, 423
 - discriminator, 423
 - feature matching, 424
 - generator, 423
 - latent representation, 423
 - latent space, 423
 - mode collapse, 423
 - vector arithmetic, 423
- generative model, 225, 248
- generator, 406, 423
- gradient clipping, 464, 481
- gradient descend method
 - code, 255
- gradient descent algorithm, 64, 77
- gray scale, 184
- ground truth, 109, 151
- gru, 472
- hand-written digit recognition, 10, 184
- handling missing data, 180
- handwritten digit recognition, 17
- hard limit
 - code, 158
- hard limit function, 125
- heat map, 227
- hidden layer, 129
- Hidden Markov model, 437
- hidden node, 129
- hidden unit, 129
- hierarchy of abstraction, 311
- high bias, 147
- high variance, 147
- Histogram of Oriented Gradients, 221
- historical averaging, 417
- HMM, 448
 - emission
 - discrete multinomial variables, 442
 - gaussian variables, 442
- forward-backward algorithm, 443
 - alpha-beta algorithm, 443
- HOG, 221
- Human Brain, 112
- hyper-parameter, 66, 114, 128
- hyperbolic tangent, 131
- hyperplane, 233, 251
- i.i.d., 324, 431, 433
- image classification, 398
- ImageNet, 375
- independent and identically distributed,
 - 324, 431
- initial probabilities, 438, 448
 - HMM, 438, 448
- initialization, 77

- input, 17, 108
 - dimension, 125
 - feature, 125
- input normalization, 168
 - code, 169
- Intelligence, 74
- internal covariance shift, 287
- intersection, 45
- IoU, 226, 250, 401
- joint probability, 50
- K-means, 402
- Karush-Kuhn-Tucker theorem, 97
- KDE, 252, 253
- kernel, 239, 349
 - Kernel Density Estimation, 227, 252, 253
 - kernel density estimation, 250
 - code, 253
 - kernel function, 251
 - gaussian kernel, 251
 - linear kernel, 251
 - kernel tricks, 246, 251
- KKT, 97
- L1 norm, 416
- label, 17
 - label smoothing, 417, 424, 426, 427
- Lagrange parameter, 72
- large dataset
 - issues, 300
- latent space, 408
- latent state, 437, 448
 - HMM, 437, 448
- latent variabl
- HMM, 437
- latent variable, 410, 437, 448
- GAN, 410
- HMM, 448
- law of total probability, 48
- layer, 127, 129
- leaky relu, 283, 415
- learning, 109, 151
 - long-term synaptic potentiation, 136
 - LTP, 136
- Learning Curve, 147
- learning curve, 145
- learning rate, 138
- life, 418
- likelihood function, 76, 448
- linear algebra, 29
 - linear equations, 33
- linear combination, 36
- linear equations, 33, 78
- linear kernel, 247, 251
- linear transformation, 39
- linearly independence, 36
- linearly separable problem, 126
- local minimizer, 63
- local optimum, 167
- log likelihood, 324
- long-term synaptic potentiation, 136
- loss
 - derivative, 155
 - mean square error, 131
- loss function, 62, 76
- loving kindness, 461
- lstm, 469, 472, 481

- block, 469
- cell, 469
 - peephole, 471, 481
- lstm block, 469
- lstm cell, 469
- LTP, 136
- machine learning, 4, 6, 17
 - supervised learning, 13
 - classification, 13
 - regression, 13
- mAP, 229, 250
- margin of separation, 235, 251
- marginalization, 51, 76, 440
- Markov model, 433, 448
- matrix, 30
 - addition, 33
 - determinant, 36
 - multiplication, 33
 - operation, 33
- matrix operation, 33
- matrix product, 33
 - useful properties, 33
- maximization problem, 61
- maximum likelihood, 323, 440
 - HMM, 440
- mean Average Precision, 229, 250
- mean square error, 118
- meta-parameter, 66, 114, 128
- Mind, 134
- mind, 418
- minibatch, 268, 272, 274, 292, 300
 - batch size, 274
- minibatch discrimination, 416, 424
- minimization problem, 61
- minimizer, 62, 76
 - local, 63
- missing data, 174
- mixture density model, 324
- MLP, 123
- mlp
 - code, 158
 - train, 158
 - code, 161
- MNIST, 11, 17, 184
- mode collapse, 423
- model, 12, 17, 107, 108
 - inference, 12
 - mapping, 12
 - prediction, 12
- model complexity, 115, 152
- model selection, 114, 157
- momentum, 284
- Monty Hall Problem, 56
- MSE, 118
- multi-class classification, 140
- multi-layer perceptron, 123
- multiclass classification, 140
- music generation, 450
- Named-Entity Recognition, 432
- Narcissistic Personality Disorder, 461, 462
- natural language, 453
 - applications, 455
 - issues, 454
- Natural Language Processing, 454, 482
- NDE, 418
- near death experience, 420

- Nesterov momentum, 284
- network depth, 267
- NLP, 454, 482
 - POS tagging, 484
 - sentiment analysis, 484
- node, 127
- noise addition, 418
- non-local-maximum suppression, 250
- norm, 38, 416
- normal distribution, 59
- normalization, 142
- normalized input, 143
- normalized time, 144
- numerical gradient, 210
 - code, 211
- numerical programming, 83
- Object Detection, 398
 - YOLO, 398
- object detection, 248, 397, 422
 - sliding window, 218
 - YOLO, 422
- objective function, 61, 76
- observable variable, 448
- observed variable, 437
- one-hot coding, 140
- one-sided label smoothing, 417
- online training, 138
- optimal point, 69
- optimization, 59, 61, 76
 - constrained, 69, 71
 - duality, 100
 - dual problem, 101
 - primal problem, 101
 - gradient descent algorithm, 64
 - orthogonal vector, 39
 - outliers, 200
 - output, 17, 108
 - output activation function, 131
 - identity function, 131
 - sigmoid function, 140
 - softmax function, 141
 - output layer, 131
 - overfitting, 115, 152
 - regularization, 118
 - padding, 350
 - parameter, 17
 - hyper-, 66
 - meta-, 66
 - parameter sharing, 435
 - parsing, 454, 482
 - Part-Of-Speech Tagging, 432, 455, 472, 482
 - pattern, 4, 17
 - pattern recognition, 4, 12, 17
 - peephole, 471, 481
 - penalty function, 72
 - penalty method, 71, 80
 - perceptron, 123
 - PixelRNN, 403
 - pmf, 49
 - polynomial, 43
 - code, 156
 - train
 - code, 156
 - polynomial curve fitting, 155
 - polynomial function, 108, 151

- pooling layer, 358
positive definite, 246
positive semidefinite, 246
posterior distribution, 76
pre-training, 268, 289, 403
precision, 179
precision/recall, 229
PReLU, 283
primal problem, 101
prior probability, 76
probability, 44, 45, 75
 conditional, 50
 disjoint, 48
 joint, 50
 mass function, 49
 outcome, 75
 sum rule, 48
probability density function, 76
probability mass function, 49, 76
probablity
 event, 75
product rule, 51
pytorch, 306
quadratic, 43
quote
 Achieving the goal, 267
 Adaptation, 345
 Break down the problem, 29
 Build up knowledge, 107
 Courage, 154, 378
 Explore, 18
 Gradient Descent, 69
 Ignorance, 420
Impermanence, 464
Kindness, 425, 462
Language, 453, 483
Learn from failure, 213
Learning, 397
Life, 449
Living life, 422
Moderation, 138, 202
Natural regulation, 10
Pattern, 4
Pre-conditioning, 418
Principles, 431
principles, 78
Science, 293
Truth, 419
Value, 209, 252
Wisdom, 461
 Wisdom and Compassion, 75
radial basis, 154
 derivative, 154
random variable, 49, 76
rank, 31, 346
recall, 179
Receiver Operating Characteristic, 198
receptive field, 357
recommendation learning, 14
record, 169
rectified linear, 269, 292
Recurrent Neural Network, 455, 481
 hidden state, 456
redundancy removal, 226, 250
regression, 13, 17, 131
regularization, 118, 155

weight decay, 118
 reincarnation, 419
 reinforcement learning, 14
 ReLu, 415
 relu, 254, 269, 292, 294, 300
 code, 255
 repeat, 165
 report
 normalized time, 144
 representation space, 408
 revise
 linear algebra, 29
 optimization, 59
 probability, 44
 RMSE, 118
 RNN, 268
 ROC, 198
 root mean square error, 118
 sample space, 45, 75
 scalar, 29
 self, 461
 semi-supervised Learning, 406
 semi-supervised learning, 14
 sensitivity, 198
 sequence-to-sequence architecture, 473,
 481
 sequence-to-sequence modeling, 473, 482
 sequential data, 431, 448
 stationary, 433
 set, 44
 difference, 45
 empty, 44
 intersection, 45
 union, 45
 set difference, 45
 side story, 7, 74, 112, 122, 134, 201, 418,
 461
 CML and Cure, 7
 Compassion, 461
 drug discovery, 201
 Human Brain, 112
 Intelligence, 74
 Mind, 134
 Mind, Brain, and Life, 418
 Neurons, 122
 sigmoid, 131, 154
 derivative, 154
 sigmoid function, 85
 significance test, 195
 sliding window, 216, 218, 248
 soft optimization, 167
 softmax, 105, 141
 code, 188
 softplus function, 80, 85
 softsign, 282
 space
 span, 41
 span, 41
 specificity, 198
 state space model, 437
 stationary sequential data, 433
 step size, 65
 stochastic gradient descent, 283
 stride, 218, 248, 351
 strided convolution, 413
 subset, 44

- subspace, 41
- sum rule, 51
- supervised learning, 13, 17
- Support Vector Machine, 233, 250, 253, 257, 259, 262
 - support vector machine, 251
 - code, 262
 - gaussian kernel
 - code, 263
 - kernel function, 251
 - gaussian kernel, 251
 - linear kernel, 251
 - kernel tricks, 251
 - primal
 - code, 255, 259
 - support vectors, 251
 - support vectors, 235
 - SVM, 233, 253, 257, 259, 262
 - Gaussian kernel, 247
 - linear kernel, 247
 - symmetry breaking, 281
 - syntax, 454, 482
 - tanh, 131, 154
 - derivative, 154
 - task
 - binary classification, 140
 - multiclass classification, 140
 - regression, 131
 - tensor, 32
 - terminating condition, 67
 - test data, 115, 151
 - test error, 115
 - threshold, 196
 - thresholding, 403
 - top-5 error rate, 376
 - torchvision, 316
 - train, 130
 - artificial neural network, 130
 - backpropagation, 134
 - training, 109, 151
 - batch, 138
 - online, 138
 - weight initialization, 139
 - training data, 115, 151
 - training error, 115
 - transfer learning, 290
 - transition matrix, 438
 - HMM, 438
 - transition probabilities, 438, 448
 - HMM, 438, 448
 - transpose, 30
 - transposed convolution, 423
 - true negative, 178
 - true positive, 178
 - unbalanced data, 188, 403
 - underfit, 147
 - underfitting, 300
 - unfolding diagram, 457, 481
 - uniform distribution, 168
 - union, 45
 - unit, 127
 - unit step function, 100, 125
 - unit vector, 38
 - universal approximator, 129
 - unsupervised learning, 14, 17, 406
 - validation, 120

- validation data, 145
- vanishing gradient problem, 269, 292, 293, 464
- variance, 50, 78
- variance (model behavior), 145
- Variational Autoencoder, 404
- vector, 29
 - orthogonal, 39
- vector arithmetic, 410
- vector space, 31
- vectorization, 129
- virtual minibatch, 417, 424
- virtues, 462
- weight, 17
- weight decay, 118
- weight initialization, 139, 163, 281
 - code, 162
 - Nguyen-Widrow
 - code, 164
 - random
 - code, 162
 - Xavier, 282
- weight scaling, 277
- weights, 124
- wisdom, 461
- words of wisdom
 - Ajahn Brahm, 464
 - Al-Kindi, 419
 - Albert Einstein, 252
 - Amelia Earhart, 425
 - Anthony de Mello, 461
 - Chuck Palahniuk, 4
 - Confucius, 397
- H. G. Wells, 345
- Isaac Asimov, 420
- Isaac Newton, 107
- Jawaharlal Nehru, 431
- Johann Wolfgang von Goethe, 267
- Lao Tzu, 462
- Leo Tolstoy, 418
- Marcus Du Sautoy, 293
- Marcus Tullius Cicero, 138
- Mark Twain, 18
- Mohandas Karamchand Gandhi, 209
- Morihei Ueshiba, 213
- Mother Teresa, 449
- Nelson Mandela, 154, 483
- Paracelsus, 202
- Ralph Waldo Emerson, 78
- René Descartes, 29
- Robina Courtin, 75
- Sean B. Carroll, 10
- Tenzin Gyatso, 422
- Winston Churchill, 378
- Yuval Noah Harari, 453
- Xavier initialization, 282
- YOLO, 398, 422
- zero-padding, 350