

# การเรียนรู้ของเครื่องเบื้องต้น

## Introduction to Machine Learning

ธีชพงศ์ กตัญญูกล

29 มีนาคม พ.ศ. 2560



# สารบัญ

สารบัญ	1
<b>1 บทนำ</b>	<b>7</b>
1.1 การเรียนรู้ของเครื่องคืออะไร . . . . .	9
1.2 ภาพรวมของการเรียนรู้ของเครื่อง . . . . .	9
1.3 กิจกรรมเชิงปฏิบัติ . . . . .	14
1.4 แบบฝึกหัด . . . . .	24
<b>2 การหาค่าดีที่สุด</b>	<b>31</b>
2.1 การหาค่าดีที่สุดพื้นฐาน . . . . .	31
2.2 เงื่อนไขของค่าทำน้อยที่สุดท้องถิ่น . . . . .	33
2.3 การค้นหาแบ่งช่วงทองคำ . . . . .	40
2.4 วิธีลงเกรเดินต์ . . . . .	45
2.5 วิธีลงชันที่สุด . . . . .	48
2.6 แบบฝึกหัด . . . . .	49
<b>3 พื้นฐานสำหรับการเรียนรู้ของเครื่อง</b>	<b>57</b>
3.1 ตัวอย่างการหาค่าลดตอนมิติเดียวด้วยพังชันพหุนาม . . . . .	57
3.2 การเลือกโมเดล . . . . .	65
3.3 ความน่าจะเป็น . . . . .	72
3.4 แบบฝึกหัด . . . . .	80
<b>4 ไมเดลเชิงเส้น</b>	<b>85</b>
4.1 การหาค่าลดตอนโดยเชิงเส้น . . . . .	85
4.2 เรกวลาไรเซชัน . . . . .	90
4.3 การจำแนกประเภทด้วยโลจิสติกตลด้อย . . . . .	91
4.4 แบบฝึกหัด . . . . .	101

<b>5 โครงข่ายประสานเที่ยม</b>	<b>105</b>
5.1 เพอร์เซปตรอนหลายชั้น . . . . .	105
5.2 โครงข่ายประสานเที่ยมแบบจ่ายไปข้างหน้า . . . . .	114
5.3 การฝึกโครงข่าย . . . . .	119
5.4 การแพร่กระจายย้อนกลับ . . . . .	123
5.5 แบบฝึกหัด . . . . .	128
<b>6 การประยุกต์ใช้โครงข่ายประสานเที่ยม</b>	<b>131</b>
6.1 ตัวอย่างง่ายๆ . . . . .	133
6.2 ตัวอย่างการหาค่าลดตอน . . . . .	138
6.3 ตัวอย่างการจำแนกประเภท . . . . .	140
6.4 ตัวอย่างการจำแนกประเภทแบบหลายกลุ่ม . . . . .	145
6.5 าร์โคด . . . . .	148
6.6 แบบฝึกหัด . . . . .	179
<b>7 การฝึกที่มีประสิทธิภาพและคำแนะนำเพิ่มเติม</b>	<b>183</b>
7.1 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีลงเกรเดียนต์กับโมเมนตัม . . . . .	184
7.2 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีบีอฟจีเอส . . . . .	186
7.3 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีอสซีจี . . . . .	191
7.4 คำแนะนำเพิ่มเติม . . . . .	209
7.5 แบบฝึกหัด . . . . .	216
<b>บรรณานุกรม</b>	<b>219</b>

## สัญญาณลักษณ์

ศาสตร์การเรียนรู้ของเครื่องอาศัยพื้นฐานทางคณิตศาสตร์ ดังนั้นเพื่อให้ลดความซับซ้อนของตัวแปรคณิตศาสตร์ ที่ใช้ สัญญาณลักษณ์ของตัวแปรคณิตศาสตร์จะใช้ตามแนวทางดังตารางข้างล่าง ยกเว้นแต่จะระบุเป็นอื่น

ชนิดตัวแปร	แบบอักษร	ตัวอย่างอักษรลาติน	ตัวอย่างอักษรกรีก
สเกลาร์	พิมพ์เล็กธรรมดา	$x$	$\phi$
เวคเตอร์	พิมพ์เล็กตัวหนา	$\mathbf{x}$	$\boldsymbol{\phi}$
เมตริกซ์	พิมพ์ใหญ่ตัวหนา	$\mathbf{X}$	$\boldsymbol{\Phi}$

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

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

เวคเตอร์  $\mathbf{x} \in \mathbb{R}^4$  เช่น  $\mathbf{x} = [10.0 \ 0.75 \ -44.6 \ 1203.8]^T$

(หมายเหตุ ถ้าไม่ระบุเป็นอย่างอื่น เวคเตอร์จะหมายถึงเวคเตอร์แบบคอลัมน์)

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

ตัวอักษรภาษาอังกฤษทั่วไปจะใช้รูปแบบ เช่น  $x, y, z$ . รูปแบบสำหรับโปรแกรมคอมพิวเตอร์ ตัวแปรที่อ้างถึงตัวแปรจากโปรแกรมคอมพิวเตอร์ จะใช้รูปแบบ เช่น  $x, y, z$  โดยตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ขึ้น กับชื่อตัวแปรในโปรแกรม (ไม่เกี่ยวข้องกับโครงสร้างชนิดข้อมูลของตัวแปร) ดังนี้

รูปแบบที่ 1

```
## Radial Basis Function
rbf <- function(x, gamma=0.1){
  return exp(-gamma*dist(x))
}
```

หรือ รูปแบบที่ 2 (รูปแบบนี้ บางครั้งอาจแสดงเลขบรรทัดออกมากด้วย)

---

```
## Radial Basis Function
rbf <- function(x, gamma=0.1){
  exp(-gamma*dist(x))
}
```

---

“หนังสือ เป็นเสมือนคลังที่รวบรวมเรื่องราว ความรู้ ความคิด วิทยาการทุกด้านทุกอย่าง ซึ่งมันนุชย์ได้เรียนรู้ ได้คิดอ่าน และเพียรพยายามบันทึกภาษาไว้ด้วยลายลักษณ์อักษร หนังสือแพร่ไปถึงที่ใด ความรู้ความคิดก็แพร่ไปถึงที่นั่น หนังสือจึงเป็นสิ่งมีค่า และมีประโยชน์ที่จะประมาณไม่ได้ในแต่ที่เป็นบ่อเกิดการเรียนรู้ของมนุษย์”

—พระราชดำรัสของพระบาทสมเด็จพระเจ้าอยู่หัวรัชกาลที่เก้า

## คำนำ

หนังสือการเรียนรู้ของเครื่องเบื้องต้นเล่มนี้ อธิบายเนื้อหาที่เกี่ยวข้องกับวิชาการการเรียนรู้ของเครื่อง โดยเฉพาะโครงข่ายประสานเที่ยม. โดยมีวัตถุประสงค์ที่จะนำเสนอ ภาพรวม พื้นฐาน การประยุกต์ใช้ งานทำทุกวี่ชั่ง อธิบายด้วยสมการคณิตศาสตร์ไปเยี่ยนโปรแกรม รวมไปถึงการพัฒนาที่น่าสนใจ และแรงบันดาลใจที่เกี่ยวข้อง. อย่างไรก็ตาม แม้จุดประสงค์หลักอย่างหนึ่งคือการนำเสนอภาพรวมของศาสตร์ แต่เนื่องจากความกว้างและลึก รวมถึงความก้าวหน้าที่เติบโตอย่างต่อเนื่องและรวดเร็วของศาสตร์นี้ การจะครอบคลุมเนื้อหาทั้งหมดเป็นไปไม่ได้เลย. ดังนั้นหนังสือเล่มนี้จัดทำเนื้อหาโดยยึดแนวคิดในการวางแผนตัวเพียงเป็นจุดเริ่มต้น ให้ผู้อ่านได้พอเข้าใจและเห็นคุณค่าของวิชาการการเรียนรู้ของเครื่องและศาสตร์พื้นฐานเบื้องหลัง รวมถึงการให้ผู้อ่านได้มีตัวอย่างโปรแกรมที่สามารถนำไปทดลองปฏิบัติได้ด้วยตนเอง ไปจนถึงอภิปรายข้อสังเกตุและประเด็นที่สำคัญ นำเสนอเกร็ดความรู้ ให้แบบฝึกหัด และอ่านวิเคราะห์ความหลากหลายในกรณีที่ผู้อ่านต้องการแหล่งค้นคว้าเพิ่มเติม. แนวคิดของลำดับเนื้อหาดังกล่าวข้างต้นนี้ ออกแบบมาเพื่อใช้ประกอบการเรียนการสอน วิชาโครงข่ายประสานเที่ยม วิชาการเรียนรู้ของเครื่อง และวิชาการรู้จำรูปแบบและการตรวจจับภาพวัตถุ ระดับปริญญาตรีและบัณฑิตศึกษา ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

การเรียนรู้ของเครื่อง มีพื้นฐานมาจากหลักหลาຍศาสตร์ เช่น การหาค่าดีที่สุด ความน่าจะเป็น หนังสือเริ่มด้วยการอธิบายภาพรวมของการเรียนรู้ของเครื่อง ตามด้วยการอธิบายศาสตร์พื้นฐานบางส่วน เพื่อให้ช่วยผู้อ่านสามารถทำความเข้าใจทุกๆส่วนได้ดียิ่งขึ้น และวิธีการเรียนรู้ของเครื่องอย่างง่าย (โมเดลเชิงเส้น) ก่อนจะอธิบายวิธีโครงข่ายประสานเที่ยม (ซึ่งเป็น หนึ่งในศาสตร์และศิลป์ของการเรียนรู้ของเครื่อง) ไปจนถึงการนำวิธีโครงข่ายประสานเที่ยมไปประยุกต์ใช้งาน

เนื่องจากเนื้อหาของหนังสือเกี่ยวข้องกับคณิตศาสตร์และมีคำพิพากษาจำนวนมาก รวมทั้งหนังสือเล่มนี้จัดเตรียมด้วยโปรแกรมเลเท็กซ์. ดังนั้น เพื่อช่วยให้เนื้อหาอ่านง่ายขึ้น และเพื่อช่วยการตัดประยุกต์ของเลเท็กซ์ รูปแบบการเขียนอาจจะมีการเน้นรูคตอนมากกว่างานเขียนภาษาไทยโดยทั่วไป และเพื่อลดความสับสนจากการรูคตอน ผู้เขียนใช้มหัพภาคเพื่อช่วยบอกการจับประยุกต์ รวมถึงบางครั้งผู้เขียนใช้ฟอนต์ตัวอักษรเพื่อเน้นคำศัพท์หรือกลุ่มคำให้ชัดเจนขึ้น ตัวอย่างเช่น “วิธีที่ดีที่สุดในการเรียนรู้คณิตศาสตร์การเรียนรู้ของเครื่องก็คือการลองลงมือทำ.” ทั้งนี้ผู้เขียนต้องขออภัยสำหรับประเด็นดังกล่าวด้วย.

การเรียบเรียงจัดทำเนื้อหาของการเรียนรู้ของเครื่องเบื้องต้นเล่มนี้ ได้รับอิทธิพลหลักจากตำราการรู้จำรูปแบบและการเรียนรู้ของเครื่อง<sup>[1]</sup> และ วิดีทัศน์สอนวิชาการเรียนรู้ของเครื่อง<sup>[58]</sup> โดยอาจมีแหล่งอื่นๆเพิ่มเติมอีกตามเนื้อหาเฉพาะ เช่น การหาค่าตีที่สุดได้รับอิทธิพลหลักจากตำราการหาค่าตีที่สุดเบื้องต้น<sup>[18]</sup>.



# บทที่ 1

## บทนำ

“Adapt or perish, now as ever, is nature’s inexorable imperative.”

—H. G. Wells

“ปรับตัว หรือ สูญพันธุ์ เป็นความจำเป็นของธรรมชาติที่ไม่อาจหลีกเลี่ยงได้ ทั้งตอนนี้แขก เช่นตลาดมา” —อช จี เวลส์

วิธีการเรียนรู้ของเครื่องถูกประยุกต์ใช้อย่างกว้างขวางในวงการธุรกิจ อุตสาหกรรม การทหาร วงการวิทยาศาสตร์ บันเทิง รวมถึงการประยุกต์ใช้ชีวิตประจำวัน ตัวอย่างเช่น ลักษณะงานที่เป็นการทำเหมืองข้อมูล การตรวจสอบหารูปแบบการใช้บัตรเครดิตที่ผิดปกติ[64] ซึ่งอาจเนื่องมาจากการที่บัตรถูกขโมยไป การบริหารการลงทุนทางการเงิน[75] งานแอพพลิเคชันที่ไม่สามารถโปรแกรมต่างๆได้ (หรือ ทำได้ยากมาก) เช่น ระบบอ่านลายมือเขียน[46] การควบคุมເຄີຍໂຄປ່ອຕ່ອງຮັນກິບ[19] การควบคุมหุ่นยนต์ที่มีการเครื่องไหวที่ซับซ้อน[4] การบริหารจัดการทรัพยากรน้ำ[16] การปรับตั้งค่าของເວຼົວໜ້ວມະໜືນ[63] การพัฒนาระบบติดตามข้อมูลล่องโทรทัศน์[13] ระบบตรวจสอบการสั่นสะเทือนของแผ่นดินไหว[67] การระบุหารสีແກມม่าจากข้อมูลกล้องໂທທັນ[35] ระบบตรวจสอบการสั่นสะเทือนของแผ่นดินไหว[67] การหารูปแบบในข้อมูลชีวสารสนเทศ[43] การแปลภาษาอัตโนมัติ[21] ระบบຽຸຈຳຄຳພຸດ[71] ระบบຽຸຈຳຄວາມກ້າວໜ້າຂອງຄອർດດັດຕະໂຮງ[81] ໃຊ້ກັບຈານຄືລປະ[22] ກີ່ພາ[35] ระบบຽຸຈຳໃບໜ້າ[7] ระบบตรวจสอบความผิดปกติของสัญญาณคลื่นไฟฟ้าหัวใจ[47] การแยกอีเมล์ที่เป็นสแปม[11] ระบบแนะนำหนังสือ เพลง ວິດີໂອ หรือສິນຄ້າ[28] การจำแนกหรือระบุหัวข้อสำหรับข้อความ[12] หรือ ແມ່ແຕ່ເປີ່ມປະສິດທິພາພຂອງງານຂອງระบบควบคุม ระบบตัดสินใจ ที่ซับซ้อน ระบบควบคุมการระบายอากาศ-เครื่องทำความร้อน-เครื่องปรับอากาศ[5] ระบบควบคุมสິນຄ້າຄົງຄັ້ງ[40, 41, 38, 39] ระบบควบคุมการຈາຈອງ[17] เป็นต้น

การทำเหมืองข้อมูล (Data Mining) หรือ บางครั้งเรียกว่า การค้นหาความรู้ในฐานข้อมูล (Knowledge Discovery in Databases) หมายถึง กระบวนการค้นหารูปแบบหรือความสัมพันธ์ที่น่าสนใจและมีประโยชน์ในข้อมูลขนาดใหญ่ (จาก Encyclopedia Britannica <https://global.britannica.com/technology/data-mining> สืบคัน 9 สิงหาคม 2559) การทำเหมืองข้อมูล เน้นที่การค้นหารูปแบบที่น่าสนใจ ซึ่งแม้การทำเหมืองข้อมูลอาจจะใช้เทคนิคที่จัดเป็นวิธีการเรียนรู้ของเครื่อง

เช่น การหากฎความสัมพันธ์ (Association Rules) การแบ่งกลุ่มข้อมูล (Cluster Analysis) การจำแนกข้อมูล (Classification) เป็นต้น. แต่ในทางปฏิบัติ บอยครังที่ในกระบวนการที่สมบูรณ์ของการทำเหมืองข้อมูลจะต้องอาศัยการทำงานของมนุษย์ เช่น กระบวนการอาจมีการใช้มนุษย์ เพื่อตรวจสอบกลั้นกรองผลลัพธ์ที่ได้จากการเรียนรู้ของเครื่อง หรือแม้แต่กระบวนการทำการทำเหมืองข้อมูล อาจใช้เพียงการทำงานของมนุษย์ โดยเขียนภาษาสอบถามจากฐานข้อมูล เพื่อค้นหารูปแบบที่น่าสนใจ โดยไม่ต้องอาศัยวิธีของการเรียนรู้ของเครื่องเลยก็ได้. ในขณะที่มุ่งมองทั่วไปคือการทำเหมืองข้อมูลใช้วิธีจากการเรียนรู้ของเครื่อง การเรียนรู้ของเครื่องเองก็อาจจะถูกสร้างขึ้นได้ โดยอาศัยข้อมูลและรูปแบบที่ถูกค้นพบโดยการทำเหมือนข้อมูลได้ เช่นกัน

การเรียนรู้ของเครื่อง (Machine Learning) จัดเป็นศาสตร์แขนงหนึ่งของปัญญาประดิษฐ์. ความสำเร็จที่สำคัญในวงการปัญญาประดิษฐ์หลายอย่าง ก็อาศัยศาสตร์การเรียนรู้ของเครื่อง เช่น โปรแกรมเล่นเกมส์แบบแคมมอน[76] และ โปรแกรมเล่นหมากล้อม[73] ที่สามารถเล่นได้ระดับสูงสุดเมื่อเทียบกับมนุษย์ หรือ ไอปีเอ็มวัตสันที่สามารถชนะมนุษย์ได้ในเกมส์ตอบคำถามเจ็บพาดี[1]. ประสิทธิผลของการเรียนรู้ของเครื่อง และศักยภาพของศาสตร์นี้ทำให้มีการศึกษาวิจัยอย่างกว้างขวางและกระตือรือร้น. ศาสตร์และศิลป์ของการเรียนรู้ของเครื่อง จึงมีการพัฒนาอย่างมีนัยสำคัญอย่างต่อเนื่อง. DARPA หรือองค์กรโครงการวิจัยชั้นสูงทางการป้องกันประเทศของสหรัฐอเมริกา (Defense Advanced Research Projects Agency) ซึ่งอยู่เบื้องหลังเทคโนโลยีหลายอย่างที่มีผลกระทบต่อเศรษฐกิจและสังคมของทั่วโลก เช่น เทคโนโลยีอินเตอร์เน็ต ก็ให้ความสำคัญและสนับสนุนการวิจัยและพัฒนาศาสตร์และศิลป์ของเทคโนโลยีการเรียนรู้ของเครื่องอย่างกว้างขวาง ดังสะท้อนออกมายังวิศวกรรมศาสตร์และศิลป์ของเทคโนโลยีพิชเชอร์ ที่กล่าวผ่านเวปไซต์ของ DARPA ที่ลงข่าวเมื่อ 19 มี.ค. พ.ศ. 2556 ว่า เป้าหมายของ DARPA คือโครงการการเรียนรู้ของเครื่องในอนาคต ที่สามารถสร้างโปรแกรมการเรียนรู้ของเครื่องที่มีประโยชน์ได้เอง โดยที่ไม่จำเป็นต้องมีมนุษย์ช่วยในการกระบวนการเรียนรู้ ไม่ว่าจะเพื่อความชำนาญเฉพาะเรื่อง หรือ เพื่อการสร้างโปรแกรมการเรียนรู้ของเครื่อง<sup>1</sup>

ปัญญาประดิษฐ์ (Artificial Intelligence 俗稱 AI) เป็นศาสตร์ของการออกแบบโปรแกรมคอมพิวเตอร์ที่มีเหตุมิผลเพื่อภารกิจเป้าหมาย โดยที่โปรแกรมนั้นจะเลือกการกระทำที่ช่วยให้ภารกิจมีโอกาสสำเร็จมากที่สุด บนพื้นฐานของสถานะการณ์ที่รับรู้และความรู้เดิมที่ได้รับ แม้จะมีความไม่แน่นอนเกี่ยวข้องอยู่

นอร์วิก และ รัสเซล[69]ได้ยกตัวอย่างศาสตร์ต่างๆ ที่ จัดอยู่ภายใต้ความหมายของปัญญาประดิษฐ์ ได้แก่ ศาสตร์การประมวลผลภาษาธรรมชาติ (Natural Language Processing) ศาสตร์การแทนความรู้ (Knowledge Representation) ศาสตร์คอมพิวเตอร์วิทัศน์ (Computer Vision) ศาสตร์วิทยาการหุ่นยนต์ (Robotics) และ ศาสตร์การเรียนรู้ของเครื่อง เป็นต้น. นอกจากศาสตร์ดังกล่าวข้างต้นนี้ ศาสตร์ปัญญาประดิษฐ์ก็ยังเกี่ยวข้องสัมพันธ์กับตรรกศาสตร์ ศาสตร์การทำค่าดีที่สุด วิศวกรรมความรู้ ศาสตร์การจัดการความไม่แน่นอนซึ่งรวมถึงสถิติศาสตร์ เป็นอย่างมาก ดังเห็นได้จากคำนิยามของปัญญาประดิษฐ์เอง

<sup>1</sup>ข้อมูลจาก <http://www.darpa.mil/NewsEvents/Releases/2013/03/19a.aspx>, สืบค้น 5 ก.ย. 2556

## 1.1 การเรียนรู้ของเครื่องคืออะไร

ในปี ค.ศ. 1959 อาร์เตอร์ ชาบูเอล เอียนโปรแกรม ให้คอมพิวเตอร์เล่นหมากซอร์ส[70] แต่ชาบูเอลเองเล่นหมากซอร์สไม่เก่งเลย. ดังนั้นแทนที่ชาบูเอลจะโปรแกรมสั่งคอมพิวเตอร์ว่าควรจะเดินมากอย่างไร แต่ชาบูเอลกลับโปรแกรมให้คอมพิวเตอร์เล่นแข่งกันเอง และโปรแกรมให้คอมพิวเตอร์เก็บผลว่า ตำแหน่งของหมากอย่างไรที่เป็นตำแหน่งที่ดี ซึ่งนำไปสู่ชัยชนะ หรือตำแหน่งไหนเป็นตำแหน่งไม่ดี และมักจะทำให้แพ้แล้วให้โปรแกรมเลือกเดินหมากตามผลที่เก็บนั้น. หลังจากชาบูเอลให้โปรแกรมเล่นแข่งกันเองหลายหมื่นกระดาน โปรแกรมเล่นหมากซอร์สของชาบูเอลก็สามารถเล่นหมากซอร์สได้ดีมาก และเล่นได้ดีกว่าตัวของชาบูเอลเอง. ณ ตอนนั้น วิธีการสร้างโปรแกรมเล่นหมากซอร์สของชาบูเอลเป็นแนวทางใหม่มาก และก็ให้ผลลัพธ์ที่ดีอย่างมาก ซึ่งได้เปิดเผยถึงศักยภาพของแนวคิดนี้

อาร์เตอร์ ชาบูเอล ได้นิยามการเรียนรู้ของเครื่อง ไว้ว่า การเรียนรู้ของเครื่องคือการทำให้คอมพิวเตอร์ มีความสามารถที่จะเรียนรู้ได้ โดยที่ไม่ต้องเขียนโปรแกรมวิธีการทำ trig. ทอม มิทเซล นักวิจัยชั้นนำทางด้านการเรียนรู้ของเครื่อง ช่วยขยายความโดยการให้นิยามไว้ว่า โปรแกรมคอมพิวเตอร์จะเรียกว่า มีการเรียนรู้จากประสบการณ์  $E$  ซึ่งเกี่ยวข้องกับภารกิจ  $T$  และสมรรถนะ  $P$  ก็ต่อเมื่อสมรรถนะของการทำภารกิจ  $T$  ที่วัดด้วย  $P$  ปรับปรุงขึ้นได้จากการประสบการณ์  $E$  [53]

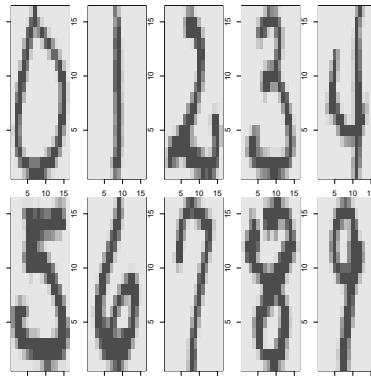
คำนิยามนี้ค่อนข้างจะเป็นทางการและมีนัยทางรูปธรรมอยู่มาก. จากตัวอย่าง โปรแกรมเล่นหมากซอร์สของชาบูเอล ประสบการณ์  $E$  คือการเดินหมากและผลที่โปรแกรมเล่นแข่งกันเอง ภารกิจ  $T$  คือการเล่นหมากซอร์ส และสมรรถนะ  $P$  คือความน่าจะเป็นที่โปรแกรมจะเล่นชนะ

ตัวอย่างที่สอง โปรแกรมเลือกหัวข้อสำหรับข้อความ[12] ประสบการณ์  $E$  คือการลองเลือกคำในข้อความไปเปรียบเทียบกับเนื้อหาในข้อความอื่นๆ ภารกิจ  $T$  คือการเลือกคำในข้อความมาเป็นหัวข้อ และสมรรถนะ  $P$  คือความน่าจะเป็นที่คำที่เลือกมาจะเป็นตัวแทนเนื้อหาของข้อความ

ตัวอย่างที่สาม โปรแกรมรู้จำลายมือแยกตัวเลข เช่น การแยกແยะรูปลายมือเขียนของตัวเลขต่างๆ ดังแสดงในรูปที่ 1.1. นั่นคือ การแยกແยะออกมาว่า แต่ละรูปภาพเป็นรูปภาพของตัวเลขอะไร. ประสบการณ์  $E$  คือการดูตัวอย่างรูปภาพตัวเลขที่เป็นลายมือเขียนและฉะลยตัวเลขที่อ่านออกมาน. ภารกิจ  $T$  คือการจำแนกແยกรูปภาพลายมือเขียน ว่าเป็นรูปภาพของเลขอะไรระหว่าง 0 ถึง 9. และสมรรถนะ  $P$  คืออัตราส่วนจำนวนรูปภาพที่ถูกจำแนกได้ถูกต้องต่อจำนวนรูปภาพทั้งหมด.

## 1.2 ภาพรวมของการเรียนรู้ของเครื่อง

ตัวอย่างการรู้จำลายมือแยกตัวเลข. รูปที่ 1.1 แสดงตัวอย่างรูปภาพที่ต้องการโปรแกรมรู้จำลายมือ เพื่อแยกรูปออกตามตัวเลขที่ภาพแสดง โดยมีรูปภาพของตัวเลขตั้งแต่ 0 ถึง 9. รูปภาพแต่ละรูปเป็นภาพขาวดำ (grayscale) มีขนาด  $16 \times 16$  พิกเซล (pixels) และค่าของแต่ละพิกเซลแทนตัวเลขจำนวนจริง. ดังนั้น รูปหนึ่งสามารถแทนได้ด้วยตัวแปรเวกเตอร์ของจำนวนจริงขนาด  $256 (= 16 \times 16)$  หรือคือสามารถ



รูปที่ 1.1: ตัวอย่างรูปตัวเลขจากกลุ่มมือเขียน

กำหนด  $\mathbf{x} \in \mathbb{R}^{256}$  แทนรูปหนึ่งรูป. ในทางปฏิบัติ เราไม่สามารถเขียนโปรแกรมนี้จากกฎตายตัวได้ หรือถ้าได้ก็อาจจะได้ผลการทำงานที่แย่มากหรือทำได้ยากมากๆ.

แต่โปรแกรมรู้ว่าจำเลยมือแยกตัวเลขสามารถเขียนขึ้นได้อย่างมีประสิทธิภาพด้วยแนวทางของศาสตร์การเรียนรู้ของเครื่อง. จุดประสงค์ที่ต้องการคือโปรแกรมที่จะบอกได้ว่ารูปภาพเป็นรูปภาพของเลขใด หรือกล่าวอีกอย่างคือ การหาค่า  $y \in \{0, 1, 2, \dots, 9\}$  ที่เหมาะสม จากรูปภาพ  $\mathbf{x}$ .

แนวทางของการเรียนรู้ของเครื่องคือจะใช้โมเดลทางคณิตศาสตร์ในการหาค่าเอาท์พุต  $y$  จาก อินพุต  $\mathbf{x}$  โดย ตัวโมเดลจะถูกควบคุมด้วยพารามิเตอร์  $\boldsymbol{\theta}$ . โมเดล  $f : \mathbf{x}, \boldsymbol{\theta} \mapsto y$  จะเขียนได้เป็น  $y = f(\mathbf{x} | \boldsymbol{\theta})$  โดย  $y$  คือเอาท์พุต (หรืออาจจะเรียกว่าตอบหรือผลลัพธ์ของกลุ่ม) และ  $\mathbf{x}$  คืออินพุต หรือเวกเตอร์ค่าของพิกเซล และ  $\boldsymbol{\theta}$  คือพารามิเตอร์ที่สามารถปรับพฤติกรรมการทำงานของโมเดลให้เป็นไปในทางที่ต้องการได้.

โมเดลทางคณิตศาสตร์นี้ ในทางทฤษฎีแล้ว บางโมเดล มีความยืดหยุ่นสูงมาก (เช่น โมเดลโครงข่ายประสาทเทียม) จะสามารถปรับตัวเป็นฟังชันอะไรก็ได้ ขึ้นกับการปรับค่าพารามิเตอร์  $\boldsymbol{\theta}$ . ในการปรับหาค่าพารามิเตอร์  $\boldsymbol{\theta}$  ที่เหมาะสม ผู้สร้างโมเดลจะใช้ตัวอย่างของรูปภาพตัวเลข  $N$  ภาพ แทนด้วยตัวแปร  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$  พร้อมฉลากเฉลย  $N$  ฉลาก แทนด้วยตัวแปรเมตริกซ์ขนาด  $1 \times N$  นั่นคือ  $\mathbf{T} = [t_1, \dots, t_N]$ . ฉลากเฉลยนี้อาจได้มาจากการใช้หักนดูรูปภาพและระบุฉลากที่ถูกต้องของแต่ละภาพไว้. ข้อมูล  $\mathbf{X}$  และ  $\mathbf{T}$  นี้จะถูกเรียกว่า “ข้อมูลชุดฝึกหัด” (training dataset).

อัลกอริทึมการเรียนรู้ของเครื่องจะช่วยหาค่าที่เหมาะสมของพารามิเตอร์  $\boldsymbol{\theta}^*$  ออกมาน. ขั้นตอนในการใช้ข้อมูลชุดฝึกหัดเพื่อปรับหาค่าพารามิเตอร์จะเรียกว่า การฝึกหัด (training) หรือการเรียนรู้ (learning). ผลลัพธ์ที่ได้จากการเรียนรู้คือโมเดลที่พร้อมใช้งาน  $f^*(\mathbf{x}) = f(\mathbf{x} | \boldsymbol{\theta}^*)$  ที่สามารถใช้หาค่าฉลากของรูปภาพได้. นั่นคือ สำหรับรูปภาพ  $\mathbf{x}'$  โมเดลจะหาค่า  $y' = f^*(\mathbf{x}')$ .

**การประเมินผล.** โมเดลที่ดีจะต้องสามารถหาค่าฉลากของรูปภาพให้ได้. รูปภาพใหม่ที่กล่าวถึงนี้ คือรูปภาพที่ไม่ได้ถูกใช้ในขั้นตอนการฝึกโมเดล. การประเมินผลการทำงานของโมเดลจะใช้ข้อมูลอีกชุด โดยที่ข้อมูลชุดนี้จะต้องไม่ได้ถูกใช้ในการฝึกหัด. ข้อมูลชุดนี้เรียกว่า “ข้อมูลชุดทดสอบ” (test dataset). ความสามารถที่โมเดลหาค่าหรือระบุฉลากของข้อมูลใหม่ได้ดี เรียกว่า คุณสมบัติความทั่วไป หรือคุณสมบัติเจนเนอ

รอลไลเซชัน (Generalization). เราต้องการโมเดลที่มีคุณสมบัติความทั่วไปที่ดี. นั่นคือ โมเดลสามารถระบุผลลัพธ์ได้ถูกต้อง แม้ว่ารูปภาพนั้นจะเป็นรูปใหม่ที่ไม่เคยเห็นมาก่อน. จากตัวอย่างการรู้จำลายมือแยกตัวเลข ข้างต้น แม้จะเป็นตัวอย่างง่ายๆ แต่ภาพรวมและหลักการที่สำคัญหลายอย่างที่อภิปรายไปนั้น ก็ครอบคลุมแนวทางของการเรียนรู้ของเครื่องโดยทั่วไป.

หัวข้อ 3.1 จะอภิปรายฟังชันโพลีโนเมียล ที่เป็นโมเดลที่มีรูปแบบทางคณิตศาสตร์ที่ไม่ซับซ้อน ซึ่งน่าจะช่วยให้ผู้อ่านเข้าใจแนวคิดและภาพรวมของการสร้างโมเดลและการปรับหาค่าพารามิเตอร์ได้ดียิ่งขึ้น. บทที่ 5 จะอภิปรายถึงโครงข่ายประสาทเทียม ซึ่งเป็นโมเดลที่มีความซับซ้อนมาก และจัดเป็นหนึ่งในศาสตร์และศิลป์ของวิชาการเรียนรู้ของเครื่อง. บทที่ 6 จะสาธิตการประยุกต์ใช้งานโครงข่ายประสาทเทียม รวมถึงตัวอย่างงานการรู้จำลายมือแยกตัวเลขนี้ เพื่อให้ผู้อ่านได้เห็นภาพโดยสมบูรณ์.

**การเตรียมและปรับข้อมูลก่อนและหลัง.** ในทางปฏิบัติแล้ว ส่วนใหญ่ อินพุต  $\mathbf{x}$  นักจะถูกเตรียมหรือปรับปรุงเบื้องต้นก่อน เพื่อเปลี่ยนไปอยู่ในปริภูมิของตัวแปร (Space of Variables) ที่โมเดลจะสามารถทำงานได้ดีขึ้น. ตัวอย่างเช่น อินพุตของโปรแกรมรู้จำลายมือแยกตัวเลข อาจจะถูกปรับให้ ขนาดภาพของตัวเลขแต่ละตัวมีความสูงและความกว้างพอดีกันก่อน ซึ่งจะช่วยทำให้สร้างโมเดลเพื่อแยกตัวเลขได้ดีขึ้น. ขั้นตอนในการเตรียมข้อมูลเบื้องต้นนี้ (pre-processing) บางครั้งจะรวมขั้นตอนที่เรียกว่า “การแยกลักษณะสำคัญ” (feature extraction) เข้าไปด้วย (ดู [42] สำหรับคำอธิบาย และตัวอย่างการแยกลักษณะสำคัญ). ในบางกรณีการปรับข้อมูลภายหลัง (post-processing) ก็อาจถูกนำมาใช้ เพื่อเพิ่มหรือปรับปรุงคุณภาพของเอาท์พุตจากโมเดลได้ เช่น การถอดรหัสแบบหนึ่งไปเค (1-to-K decoder) เพื่อปรับเอาท์พุตจากโมเดลจำแนกกลุ่มให้อยู่ในรูปแบบที่ต้องการ (ดูหัวข้อ 4.3.3 สำหรับรายละเอียด)

**ประเภทของการเรียนรู้ของเครื่อง.** หากมองจากมุมของประสบการณ์  $E$  ที่คอมพิวเตอร์ใช้ปรับปรุงการทำงาน โปรแกรมรู้จำลายมือแยกตัวเลขต้องการประสบการณ์ ซึ่งคือตัวอย่างอินพุต  $\mathbf{X}$  และตัวอย่างเอาท์พุต  $\mathbf{T}$ . การเรียนรู้ของเครื่องที่เกี่ยวข้องกับประสบการณ์เช่นนี้ จะเรียกว่า “การเรียนรู้แบบมีผู้ช่วยสอน” (Supervised Learning). เมื่อมองจากลักษณะของเอาท์พุต ปัญหาการรู้จำลายมือแยกตัวเลขมีเอาท์พุตเป็นคลาสของกลุ่ม ซึ่งจำนวนกลุ่มนี้จำนวนนั้นแน่นอน เช่น 10 กลุ่ม ตั้งแต่ ‘0’ ถึง ‘9’. ปัญหาแบบนี้จัดเป็นชนิดปัญหาของการจำแนกประเภท (Classification). แต่หากเอาท์พุตมีลักษณะเป็นเลขจำนวนจริง เช่น การคำนวณน้ำฝน การคำนวณปริมาณแร่ธาตุในดิน การคำนวณปริมาณน้ำยางจากต้นยางที่ปลูกในสภาพต่างๆ การคำนวณแรงที่เกิดกับใบพัดลักษณะต่างๆ ของกังหันลม การคำนวณมูลค่าการซื้อขายหลักทรัพย์ ปัญหาในลักษณะแบบนี้จะจัดเป็นชนิดปัญหาของการหาค่าคงอยู่ (Regression). บทที่ 4 และ 5 จะอภิปรายถึงการเรียนรู้แบบมีผู้ช่วยสอน ทั้งสองแบบ.

หากประสบการณ์  $E$  ไม่ได้ให้ตัวอย่างเอาท์พุตมาให้ด้วย การเรียนรู้ของเครื่องแบบนี้จะเรียกว่า “การเรียนรู้แบบไม่มีผู้ช่วยสอน” (Unsupervised Learning). ปัญหาประเภทนี้ มีหลายชนิด เช่นการจัดกลุ่มข้อมูล (Clustering) ซึ่งคือการจัดของหรือข้อมูลที่มีลักษณะคล้ายกันให้อยู่กลุ่มเดียวกัน การประมาณความหนาแน่นของข้อมูล (Density Estimation) หรือการลดมิติของข้อมูล (Dimension Reduction) เป็นต้น

นอกจากนี้ยังมีการเรียนรู้ของเครื่องที่ลักษณะของประสบการณ์ *E* ต่างจากสองกลุ่มข้างต้น เช่น การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning), การเรียนรู้แบบกึ่งมีผู้ช่วยสอน (Semi-Supervised Learning), การเรียนรู้ของเครื่องที่ใช้ในระบบแนะนำสินค้าอัตโนมัติ (recommender systems) เป็นต้น. การเรียนรู้แบบเสริมกำลังจะใช้กับปัญหาเชิงลำดับเวลา ที่คอมพิวเตอร์จะเลือกการกระทำที่เหมาะสมกับสถานะในแต่ละคาบเวลา เพื่อที่จะให้ผลรวมของรางวัลในแต่ละคาบเวลามากที่สุด. ลักษณะของปัญหาที่การเรียนรู้แบบเสริมกำลังทำงานคือ คอมพิวเตอร์ไม่มีตัวอย่างของการกระทำที่มั่นคงจะเลือก แต่มันจะค้นหาการกระทำที่เหมาะสมกับสถานะโดยการลองผิดลองถูก เพื่อที่จะเรียนรู้ผลของการกระทำนั้น ในขณะที่ พยายามจะให้ผลรวมของรางวัลมากที่สุดด้วย. ระบบการเรียนรู้แบบเสริมกำลังที่ดีจะต้องสร้างสมดุล ระหว่างการเลือกการกระทำ เพื่อที่จะได้ผลรวมรางวัลตีสุด กับการเลือกการกระทำเพื่อการเรียนรู้. การประเด็นเรื่องความสมดุลนี้เรียกว่า ประเด็นของการใช้งานและการเรียนรู้ (issue of exploitation and exploration). ลักษณะที่เด่นชัดอีกอย่างหนึ่งของการเรียนรู้แบบเสริมกำลัง คือการที่ระบบมีปฏิสัมพันธ์สั่ง แวดล้อม ผลของการกระทำที่ระบบเลือกมีผลต่อสิ่งที่ระบบจะเรียนรู้ (ดู [40] หรือ [74] สำหรับรายละเอียด)

บทที่ 2 อภิปรายศาสตร์การหาค่าดีที่สุดเบื้องต้น ซึ่งศาสตร์การหาค่าดีที่สุดเป็นพื้นฐานที่สำคัญ สำหรับอธิบายกลไกการทำงานของวิชาการเรียนรู้เครื่อง. บทที่ 3 อภิปรายตัวอย่าง่ายๆที่จะช่วยให้เห็นภาพของการเรียนรู้ของเครื่อง การประเมินผล และการเลือกโมเดล เพื่อความเข้าใจก่อนที่จะศึกษาโมเดลที่ซับซ้อนขึ้น บทที่ 4 อภิปรายโมเดลเชิงเส้น ซึ่งมีรูปแบบคณิตศาสตร์ที่เข้าใจง่ายไม่ซับซ้อน. บทที่ 5 และ 6 อภิปรายโมเดลโครงข่ายประสาทเทียม และการนำไปประยุกต์ใช้งาน. บทที่ 7 อภิปรายเทคนิคและแนวทางในการนำโครงข่ายประสาทเทียมไปใช้ในทางปฏิบัติ. บทที่ ?? อภิปรายทิศทางการพัฒนาในปัจจุบันของโครงข่ายประสาทเทียม.

**เก้าอี้ความรู้ สติปัญญาของลิง** ปี ค.ศ. 2008 สถานีโทรทัศน์พีบีเอสของสหรัฐอเมริกาออกอากาศรายการโนวา เกี่ยวกับสติปัญญาของลิงไม่มีทาง เรื่อง “Ape Genius” รายงานนำเสนองานศึกษาสติปัญญาของลิงชิมแปนซีและลิงโอบโนโลหะงาน (ลิงชิมแปนซีและลิงโอบโนโล มีพันธุกรรมต่างจากมนุษย์แค่ประมาณ 1.2% (มนุษย์แต่ละคนมีพันธุกรรมแตกต่างกันประมาณ 0.1% จาก <http://humanorigins.si.edu/evidence/genetics> สืบคัน 12 สิงหาคม 2559) รายการดำเนินการ โดยเป้าหมายคือ เพื่อหาคำตอบว่า ลักษณะของสติปัญญาแห่งมุ่งมาดที่ต่างกัน และทำให้ชิมแปนซีและโอบโนโลไม่สามารถพัฒนาขึ้นมาสร้างอารยธรรม เช่นเดียวกับที่มนุษย์ทำได้.

ความสามารถในการสร้างและใช้เครื่องมือ. มีหลักฐานชัดเจนว่า ลิงชิมแปนซีมีการสร้างและใช้เครื่องมือ เช่น การสังเกตุของเงิน ภูดถอด ที่พับลิงชิมแปนซีในแทนซาเนียใช้กิ่งไม้ในการล้อมมากิน และชิล พริตซ์ที่พับลิงชิมแปนซีในป่าไฟกอลีในประเทศเซเนกัล ที่สร้างหอกจากกิ่งไม้และใช้เป็นเครื่องมือในการล่าหาอาหาร

ความสามารถในการทำงานร่วมกัน. มีหลักฐานหลายอย่างแสดงให้เห็นพฤติกรรมในลักษณะการอุกลาร่วมกันของลิงชิมแปนซี และมีการทดลองของสถาบันวิจัยลิงใหญ่เมือง (Great Ape Research Institute) ของญี่ปุ่น ที่พบร่องรอยชิมแปนซีมีความสามารถในการทำงานร่วมกัน มีความสามารถในการขอความช่วยเหลือ และก็สามารถให้ความช่วยเหลือมนุษย์ได้เวลาที่ถูกร้องขอ

ความสามารถในการแก้ปัญหา. การศึกษาหนึ่งทดลอง โดยใส่เมล็ดถั่วไว้ในหลอดยาวที่ลิงไม่สามารถจะล้วงเข้าไปหยิบได้ และตัวหลอดกี้ดีดติดกับกรงแน่นจนลิงไม่สามารถขยับได้. ลิงใช้เวลาพักหนึ่ง ก่อนจะพบริจิกับปัญหา. มันไปที่บ่อน้ำในกรง อม

น้ำแล้วมาพ่นใส่หลอด แล้วอาหารก็ลอดขึ้นบนน้ำ มันเดินน้ำเข้าไปจนอาหารลอยอยู่ในระดับที่เอื้อมถึงได้ สิ่งนี้แสดงถึงความสามารถในการแก้ปัญหาของลิง.

ความสามารถในการเลียนแบบ. ทีมของนักจิตวิทยาแอนดรู วิทเทนต้องการทดสอบความสามารถในการเลียนแบบของลิง. ทีมสร้างเครื่องกลิกที่ลิงจะต้องทำ 2 ขั้นตอนได้แก่ หมุนจานให้พอดีซองและโยกคันโยก เพื่อจะได้กินอาหาร และนำเครื่องไปทดสอบกับตัวลิง. ลิงไม่สามารถจะหาวิธีทำนี้ได้อะไร. แต่ทีมงานค่อยๆสอนลิงขั้นมาตัวหนึ่ง จากนั้นลองให้ลิงตัวอื่นคลิงตัวนี้ทำงาน แล้วสังเกตุว่าลิงตัวอื่นๆสามารถเลียนแบบ เพื่อทำงานสองขั้นตอนนี้ได้อย่างง่ายดาย.

ความสามารถทางตัวเลข. เททธูริ มัตซูชา瓦แห่งมหาวิทยาลัยเกียวโต นำเสนอผลการทดสอบลิงชิมแปนซีชีโอ ที่แสดงความสามารถทางตัวเลข ใน การเข้าใจความหมายของตัวเลขารบิก และยังสามารถรู้ลำดับของตัวเลขได้

ความสามารถทางภาษาและการสื่อสาร. ลิงโบโนโบซึ่อคานซี เรียนรู้ภาษาอังกฤษได้เอง โดยไม่ได้ถูกสอนโดยตรง และชูชา เวจ-รัมบากำแสดงให้เห็นว่าคานซีเข้าใจภาษาอังกฤษและสามารถทำตามคำสั่งได้อย่างถูกต้อง

**ความสามารถที่ลิงไม่มี** ความสามารถที่กล่าวมาข้างต้น เป็นความสามารถที่พบหลักฐานในลิงชิมแปนซีหรือโบโนโบ. แต่ความสามารถที่ลิงชิมแปนซีหรือโบโนโบไม่มี และเป็นปัจจัยสำคัญที่ทำให้ลิงไม่สามารถพัฒนาอารยธรรมขึ้นมาได้ เช่นว่าคือความสามารถด้านอารมณ์. ลิงชิมแปนซีมีปัญหาที่เห็นได้ชัดเจน คือปัญหาด้านอารมณ์ ทั้งการแก่งแย่งชิงดีกัน ความรุนแรง และที่สำคัญคือ การควบคุมอารมณ์ตัวเอง.

ความสามารถในการควบคุมตัวเอง. การทดลองของแซลลี บอยเซนมหาวิทยาลัยรัฐโอไฮโอ แสดงให้เห็นโดยให้ลิงเลือกงานอาหารระหว่างงาน 2 งานที่มีชนมอยู่ไม่เท่ากัน แต่งานที่ลิงเลือกนี่อีกไปทาง จะเป็นงานที่จะไปให้กับลิงอีกด้วย. ถ้าเป็นชนมที่อยู่บนจาน ลิงไม่สามารถจะอดใจและเอื้อมไปที่งานที่น้อยกว่าได้ มันจะเอื้อมไปที่งานที่มันเห็นอาหารมากกว่าตัดผล. แต่พอแซลลี บอยเซนเปลี่ยนจากการที่เอาชนมวลไว้ในจานให้เห็น กลับใช้ตัวเลขซึ่งลิงเข้าใจความหมาย วางไว้แทน. ลิงสามารถเรียนรู้ที่จะเอื้อมไปที่งานที่ตัวเลขน้อยกว่าได้. การทดลองนี้แสดงให้เห็นว่า ลิงชิมแปนซีมีปัญหานในการควบคุมอารมณ์ของตัวมันเอง. เวลาที่มันเห็นอาหารอยู่ มันไม่สามารถควบคุมตัวเพื่อเลือกทางเลือกที่ดีกว่าได้ แต่พอตัดแรงกระตุนทางอารมณ์ออก (ใช้ตัวเลขวางแผนอาหารจริง) มันสามารถเลือกทางเลือกที่ดีกว่าได้.

นอกจากความสามารถในการควบคุมตนเองแล้ว ปัจจัยสำคัญอีกสองอย่างที่รายการสรุปว่า เป็นอุปสรรคที่ทำให้สติปัญญาของลิงไม่อาจสะสม สร้างเสริมไปสู่การพัฒนาในระดับเดียวกับมนุษย์ได้ ก็คือ ความสามารถในการเรียนรู้โดยรับการถ่ายทอดจากคนอื่น (หรือลิงตัวอื่น) และความสามารถในการสอน. แม้เด็กอาจไม่ได้แสดงความสามารถในการแก้ปัญหาได้ดีเท่ากับลิงชิมแปนซี แต่เด็กๆแสดงความสามารถที่สามารถเรียนรู้จากสิ่งที่ถูกสอนได้ดีกว่า สุนัขเองก็ยังมีความสามารถในการเรียนจากการสอนของมนุษย์ได้ดีกว่าลิง.

นอกจากความสามารถในการเรียนจากการถ่ายทอด ความเต็มใจที่จะถ่ายทอด หรือความเต็มใจที่จะสอน ก็เป็นส่วนประกอบสำคัญที่ทำให้การถ่ายทอดความรู้เกิดขึ้นได้ และลิงชิมแปนซีไม่มีทั้งสององค์ประกอบนี้. อารยธรรมของมนุษย์สร้างโดยการส่งผ่านความรู้และปัญญาจากรุ่นสู่รุ่น. แม้ลิงสามารถเรียนรู้จากลิงตัวอื่นได้โดยการเลียนแบบนั้nm ก็จะช้าและตื้นเขิน. บางครั้งยังอาจมีการสูญหายไป จากการเปลี่ยนรุ่นของลิงอีกด้วย. ลิงรุ่นเก่าตายไป ลิงรุ่นใหม่อาจไม่ได้เรียนรู้สิ่งที่ลิงรุ่นเก่ารู้แล้ว หลายๆอย่างที่ลิงรุ่นเก่ารู้แล้ว เช่นวิธีการใช้เครื่องมือ อาจหายไปจากลิงรุ่นใหม่ และอาจใช้เวลาอีกนานกว่าที่ลิงรุ่นใหม่จะพบวิธีใช้เครื่องมืออีกครั้ง.

การควบคุมตัวเอง การเรียนรู้จากการถ่ายทอด และความเต็มใจที่จะสอน เป็นคุณสมบัติที่แยกมนุษย์ออกจากลิง และเป็นพื้นฐานอารยธรรมของมนุษย์.

## 1.3 กิจกรรมเชิงปฏิบัติ

การเรียนรู้ของเครื่องเป็นศาสตร์และศิลป์ของการนำทฤษฎีไปประยุกต์กับการปฏิบัติ และวิธีที่ดีที่สุดในการเรียนรู้ศาสตร์การเรียนรู้ของเครื่อง ก็คือ การลองลงมือทำ. แม้จะมีเครื่องมือที่ใช้ได้มากมาย ไม่ว่าจะเป็น แมทแลป ไฟรอน อาร์โปรเจค หรือว่าภาษาโปรแกรมทั่วๆไป เช่น ซี ซีพลัสพลัส หรือจาวา เป็นต้น หนังสือเล่มนี้เลือกวาร์โปรเจค (R Project) เป็นเครื่องมือสำหรับเสริมเนื้อหาของหนังสือนี้ บนพื้นฐานของความสามารถในการคำนวนที่ดี และประสิทธิภาพในการประมวลผลข้อมูลขนาดใหญ่ รวมถึงการที่อาร์โปรเจคติดตั้งง่าย และมีเสถียรภาพพอสมควร. นอกจากนั้น อาร์โปรเจคยังเป็นโปรแกรมรหัสเปิดที่สนับสนุน หลากหลายระบบปฏิบัติการ และมีฐานผู้ใช้งานจำนวนมาก.

ผู้อ่านสามารถดาวน์โหลดและติดตั้งโปรแกรมอาร์โปรเจคได้ตามคำแนะนำจากเวปไซต์ <https://www.r-project.org/> หลังจากติดตั้งโปรแกรมเรียบร้อยแล้ว หัวข้อ 1.3.1 แสดงตัวอย่างการใช้โปรแกรมอาร์เบื้องต้น พร้อมคำอธิบายสั้นๆ.

### 1.3.1 การใช้อาร์โปรเจคเบื้องต้น

ตัวอย่างต่อไปนี้ (แสดงข้างล่างในรูปแบบสองคอลัมน์) แสดงการบวก ลบ คูณ หาร เอาเศษ (modulus) ยกกำลัง เปรียบเทียบมากกว่า เปรียบเทียบมากกว่าหรือเท่ากับ เปรียบเทียบเท่ากับ เปรียบเทียบไม่เท่ากับ เปรียบเทียบน้อยกว่า และการใช้งานลีบ. บรรทัดที่นำหน้าด้วยเครื่องหมาย > เป็นคำสั่งที่ผู้ใช้ใส่เข้าไป ส่วนบรรทัดที่นำหน้าด้วย [1] คือคำตอบที่ได้จากการ์โปรเจค.

> 5 + 4	> 5 > 3
[1] 9	[1] TRUE
> 5 - 4	> 5 >= 3
[1] 1	[1] TRUE
> 5 * 4	> 5 == 3
[1] 20	[1] FALSE
> 5 / 4	> 5 != 3
[1] 1.25	[1] TRUE
> 5 %% 4	> 5 < 3
[1] 1	[1] FALSE
> 2^5	> (5 < 3) == TRUE
[1] 32	[1] FALSE

ตัวอย่างข้างล่างในรูปแบบสองคอลัมน์ แสดง การใช้ฟังชันสำเร็จรูป ได้แก่ pi สำหรับค่า  $\pi$ , ค่าฟังชันไซน์, ค่า  $e^1$ , ค่าล็อกการิทึมของ 0.1, 0, 10 และ  $e^5$ , การปั๊ดเศษเป็นจำนวนเต็ม และเป็นทศนิยม 1 ตำแหน่ง และการดูเวลาของระบบ. สังเกตุ มหัพภาค (เครื่องหมายจุด) สำหรับอาร์โปรเจค เป็นแค่ตัว

อักขระธรรมดายังไม่ได้มีความหมายพิเศษ. สำหรับอาร์ໂປຣເຈກ ມັກກາຟໄມ່ໃຈ່  
ເຄື່ອງໝາຍບ່າງໜີ້ແອທທີ່ບົວດີ (attribute) ອີ່ເມຮອດ (method) ຂອງການເຂົ້າໃນໂປຣແກຣມເຊິ່ງວັດຖຸ

```
> pi                               > log(10)
[1] 3.141593                      [1] 2.302585
> sin(pi/2)                        > log(exp(5))
[1] 1                                [1] 5
> exp(1)                           > round(545.32)
[1] 2.718282                      [1] 545
> log(0.1)                          > round(545.32,1)
[1] -2.302585                     [1] 545.3
> log(0)                            > Sys.time()
[1] -Inf                            [1] "2013-09-05 10:59:32 ICT"
```

ຕ້ວຍຢ່າງຂ້າງລ່າງໃນຮູບແບບສອງຄອລັມນີ້ ແສດກາຣໃຫ້ຄ່າຕ້ວແປ ກາຣເຮີກຕ້ວແປ ກາຣເຮີກຕ້ວແປທີ່ຍັງໄມ່ໄດ້ປະກາສ. ສັງເກັດ (1) ຕ້ວໃຫ້ຕ້ວເລີກໄມ່ເໜືອນກັນ (2) ເທຩມ `my.x` ເປັນແຄ່ຂໍ້ຕ້ວແປ ເຊັ່ນເດີຍກັບ ກາພາ  
ໆທີ່ມັກໃຊ້ຊື່ດຳລ່າງແຍກຄໍາ ເຊັ່ນ `my_x` ຢ້າອີກຮັ້ງທັພກາດ(ເຄື່ອງໝາຍຈຸດ)ໄມ່ໄດ້ມີຄວາມໝາຍພິເສດສຳຮັບອາຣ໌  
ໂປຣເຈກ (3) ອາຣີໂປຣເຈກໃຊ້ = ອີ່ໂປຣເຈກໃຊ້ <- ເປັນຕ້ວບວິບຕິກາຣໃຫ້ຄ່າ (assignment operator).

```
> x <- 5                           > x <- x + 8
> x                               > x
[1] 5                                [1] 16
> X                               > x = 4
Error: object 'X' not found          > x
> X <- 8                           [1] 4
> X                               > my.x <- 23
[1] 8                                > my.x
> x                               [1] 23
[1] 5
```

ຕ້ວຍຢ່າງການນິຍາມພິ້ນໜີ້. ສັງເກັດ (1) ດ້ວຍໃນຕ້ວພິ້ນໜີ້ໄມ່ມີກາຣເຮີກ `return` ອາຣີໂປຣເຈກຈະເອົາຄ່າທີ່  
ໄດ້ຈາກຄໍາສັ່ງບຣທັດສຸດທ້າຍຕ້ວພິ້ນໜີ້ອອກມາຕອບ (ດູ `f0` ເທີບກັບ `my.f0`) (2) ດ້ວຍໃກ່ຂໍ້ພິ້ນໜີ້ໂດຍໄມ່ໄສ  
ວັງເລີບຕາມໜີ້ ອາຣີໂປຣເຈກຈະພິມພົດຂອງພິ້ນໜີ້ອອກມາ (ດູ `f2`) (3) ອາຣີໂປຣເຈກອຸ່ນໝາຕີໃຫ້ສາມາດໄສ່ອາຣ໌  
ກູມເນັດຕາມລຳດັບ ອີ່ໂປຣເຈກໃສ່ຂໍ້ເຂົ້າໄປໄດ້ (ດູກາຣເຮີກໃຊ້ `f2`).

```
> f0 <- function(){ 1; 5; 9; }
```

```

> f0()
[1] 9
> f0 <- function(){ 1; 5; 9; 3}
> f0()
[1] 3
> my.f0 <- function(){ 1; return(5); 9; 3}
> my.f0()
[1] 5
> f2 <- function(a, b){ a - b }
> f2
function(a, b){ a - b }
> f2(5, 9)
[1] -4
> f2(a=5, b=9)
[1] -4
> f2(b=5, a=9)
[1] 4

```

ตัวอย่างการดูคำอธิบายและการหาฟังชันที่ต้องการ. สังเกตุและเปรียบเทียบผลจากการรันสองคำสั่งข้างล่าง.

```

> help(seq)
> ??seq

```

### 1.3.2 ตัวแปรหลายค่าและเมตริกซ์

ตัวอย่างการกำหนดค่าให้กับตัวแปรหลายค่าและตัวแปรเมตริกซ์. สังเกตุ (1) วิธีการสร้างตัวแปรหลายค่าและผลที่ได้ (2) การสร้างเมตริกซ์ของอาร์โปรเจค จะเรียกตัวเลขตามคอลัมน์โดยดีฟอลต์ ถ้าอยากรีใช้เรียกตามแถวต้องระบุ `byrow=T` หรือ `byrow=TRUE`.

```

> seq(0, 1, len=5)
[1] 0.00 0.25 0.50 0.75 1.00
> seq(0, 1, by=0.2)
[1] 0.0 0.2 0.4 0.6 0.8 1.0
> x <- seq(-1, 1, len=5)
> x

```

```
[1] -1.0 -0.5  0.0  0.5  1.0
> 1:5
[1] 1 2 3 4 5
> seq(1, 5)
[1] 1 2 3 4 5
> x <- 1:3
> x
[1] 1 2 3
> y <- seq(0, 0, len=3)
> y
[1] 0 0 0
> m1 <- matrix(0, 2, 3)
> m1
[,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
> m2 <- matrix(8, 2, 3)
> m2
[,1] [,2] [,3]
[1,]    8    8    8
[2,]    8    8    8
> m1 <- matrix(1:6, 2, 3)
> m1
[,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> m1 <- matrix(c(24, 6, 2475, 4, 7, 1776), 2, 3)
> m1
[,1] [,2] [,3]
[1,]   24  2475     7
[2,]     6     4 1776
> m1 <- matrix(c(24, 6, 2475, 4, 7, 1776), 2, 3, byrow=T)
> m1
[,1] [,2] [,3]
```

```
[1,] 24 6 2475
[2,] 4 7 1776
```

ตัวอย่างการอ้างค่าตัวแปรหลายค่าและเมตริกซ์. สังเกตุ (1) การใช้ฟังชัน `dim` ในการดูขนาดของเมตริกซ์ (2) ถ้าแยกส่วนของเมตริกซ์ออกมา แล้วส่วนที่แยกออกมานี้เป็นแต่แค่เดียวหรือหลักเดียว อาร์เพรเจคจะเปลี่ยนเมตริกซ์เป็นตัวแปรหลายค่า (ดูผล `y[-2,]` เปรียบเทียบกับ `y[-2,,drop=F]`). ถ้าผู้ใช้อยากจะให้ผลลัพธ์คงชนิดเป็นเมตริกซ์ ต้องระบุอาร์กูเมนต์ `drop=F` หรือ `drop=FALSE` เข้าไป.

```
> x <- seq(1,10, by=2)
> x
[1] 1 3 5 7 9
> x[4]
[1] 7
> x[-4]
[1] 1 3 5 9
> x[2:4]
[1] 3 5 7
> x[-2:-4]
[1] 1 9
> x[c(1,4)]
[1] 1 7
> y <- matrix(seq(1,10, len=6), 2,3, byrow=T)
> y
[,1] [,2] [,3]
[1,] 1.0 2.8 4.6
[2,] 6.4 8.2 10.0
> y[2,3]
[1] 10
> y[2,]
[1] 6.4 8.2 10.0
> y[,2]
[1] 2.8 8.2
> y[,2:3]
[,1] [,2]
[1,] 2.8 4.6
```

```
[2,] 8.2 10.0
> y[,c(1,3)]
 [,1] [,2]
[1,] 1.0 4.6
[2,] 6.4 10.0
> dim(y)
[1] 2 3
> y[,-1]
 [,1] [,2]
[1,] 2.8 4.6
[2,] 8.2 10.0
> dim(y[,-1])
[1] 2 2
> y[-2,]
[1] 1.0 2.8 4.6
> dim(y[-2,])
NULL
> length(y[-2,])
[1] 3
> class(y)
[1] "matrix"
> class(y[-2,])
[1] "numeric"
> y[-2,,drop=F]
 [,1] [,2] [,3]
[1,] 1 2.8 4.6
> class(y[-2,,drop=F])
[1] "matrix"
> y[,-1:-2]
[1] 4.6 10.0
> class(y[,-1:-2])
[1] "numeric"
```

ตัวอย่างการบวกเมทริกซ์ การคูณสเกลาร์กับเมทริกซ์ การลบเมทริกซ์ การคูณแบบตัวต่อตัว (Element-

Wise Multiplication) การทำเมตริกซ์ทรานส์โพร์ต (Matrix Transpost) การคูณเมตริกซ์ การทำเมตริกซ์ อินเวอร์ส (Matrix Inverse) และการสร้างเมตริกซ์ลั่นทแยง (Diagonal Matrix).

```
> y
 [,1] [,2] [,3]
[1,] 1.0 2.8 4.6
[2,] 6.4 8.2 10.0
> y + y
 [,1] [,2] [,3]
[1,] 2.0 5.6 9.2
[2,] 12.8 16.4 20.0
> 2*y
 [,1] [,2] [,3]
[1,] 2.0 5.6 9.2
[2,] 12.8 16.4 20.0
> 2*y - y
 [,1] [,2] [,3]
[1,] 1.0 2.8 4.6
[2,] 6.4 8.2 10.0
> y * y
 [,1] [,2] [,3]
[1,] 1.00 7.84 21.16
[2,] 40.96 67.24 100.00
> y %*% y
Error in y %*% y : non-conformable arguments
> t(y)
 [,1] [,2]
[1,] 1.0 6.4
[2,] 2.8 8.2
[3,] 4.6 10.0
> y %*% t(y)
 [,1] [,2]
[1,] 30.00 75.36
[2,] 75.36 208.20
```

```

> t(y) %*% y
      [,1]  [,2]  [,3]
[1,] 41.96 55.28 68.60
[2,] 55.28 75.08 94.88
[3,] 68.60 94.88 121.16
> x <- matrix(seq(1,9, len=4), 2, 2)
> x
      [,1]  [,2]
[1,] 1.000000 6.333333
[2,] 3.666667 9.000000
> solve(x)
      [,1]  [,2]
[1,] -0.6328125 0.4453125
[2,]  0.2578125 -0.0703125
> solve(x) %*% x
      [,1]  [,2]
[1,] 1.000000e+00 -3.885781e-16
[2,] -3.585999e-17 1.000000e+00
> z <- diag(4)
> z
      [,1]  [,2]  [,3]  [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1

```

### 1.3.3 การเขียนโปรแกรมด้วยอาร์สคริปต์

ตัวอย่างที่ผ่านมาเป็นการใช้อาร์โปราเจคในลักษณะของรันคำสั่งเดียวๆ การใช้งานอาร์โปราเจคสามารถที่จะรวมคำสั่งเดียวๆเพื่อเขียนเป็นโปรแกรมได้ เช่น ตัวอย่างการใช้คำสั่ง `if` ในการควบคุมลำดับของโปรแกรม.

```

> a <- 3
>
> if(a > 2){

```

```

+
+      cat('a is greater than 2.\n')
+
a is greater than 2.

> a <- 1
>
> if(a > 2){
+
+      cat('a is greater than 2.\n')
+

```

ตัวอย่างการใช้ `if ... else`.

```

> a <- 1
>
> if(a > 2){
+
+      cat('a is greater than 2.\n')
+
+ }else{
+      cat('a is not greater than 2.\n')
+
}
```

a is not greater than 2.

ตัวอย่างการใช้ `for`. สังเกตุการใช้คำสั่ง `paste` เพื่อรวมข้อความกับค่าตัวแปร.

```

> a <- 0
>
> for(i in 1:4){
+
+      cat(paste('a = ', a, '\n'))
+
+      a <- a + 1
+
}
```

a = 0  
a = 1  
a = 2  
a = 3

ตัวอย่างการใช้คำสั่ง `for` ร่วมกับ `break`. สังเกตุการทำงาน และ ลองรัน ??control เพื่อศึกษาคำสั่งอื่นๆที่ใช้ควบคุมลำดับของโปรแกรม เช่น `while` และ `repeat` เป็นต้น.

```
> a <- 0
>
> for(i in 1:4){
+
+   cat(paste('a = ', a))
+
+   a <- a + 1
+
+   if(a > 2) break;
+
+   cat('; done updating a\n')
+
+ }
a = 0; done updating a
a = 1; done updating a
a = 2
```

การโหลดชอร์สโค๊ดและเรียกไลบรารี ผู้ใช้สามารถเขียนโปรแกรมเป็นสคริปต์ไว้แล้วเรียกใช้ภายหลังได้ เช่น ผู้ใช้อาจเขียนโปรแกรมข้างล่างนี้แล้วบันทึก เป็นไฟล์ชื่อ `nparks.r`.

```
list.parks <- function(i=NULL){
  parks.to.see <- c('Khao yai', 'Keang Krachan', 'Pha tam')

  N <- length(parks.to.see)

  if(is.null(i)){
    i <- 1:N
  }
  for(p in i){
    cat(paste('park: ', parks.to.see[p], '\n'))
  }
}

list.parks()
```

หลังจากบันทึกไฟล์ข้างต้นแล้ว ที่หน้าต่างส่วนรับคำสั่ง (Command Session) ของอาร์ໂປຣເຈກ ให้ทดลองเรียก `source('nparks.r')`. สังเกตุผลและวิธีการรันสคริปต์ของอาร์ໂປຣເຈກ. ถ้าอาร์ໂປຣເຈກหาไฟล์ไม่เจอ ให้ลองตรวจสอบดูว่าไฟล์ที่บันทึกได้บันทึกอยู่ที่ใดเรียกหอรีเดียวกับไฟรีกหอรีที่อาร์ໂປຣເຈກทำงานอยู่หรือไม่ ให้ลองศึกษาวิธีการใช้คำสั่ง `getwd` คำสั่ง `setwd` รวมถึงวิธีการใส่เส้นทางที่อยู่ (path) หน้าชื่อไฟล์สำหรับการใช้คำสั่ง `source` เพื่อระบุที่อยู่ของไฟล์.

อาร์ໂປຣເຈກมีไลบรารีให้เลือกใช้มากมาย. ไลบรารีของอาร์ໂປຣເຈກ มีลักษณะเดียวกับ library ในภาษา C, package ใน Java, หรือ toolbox ใน Matlab. การเรียกใช้ไลบรารีก็เพียงแค่เรียก `library(...)` โดยใส่ชื่อไลบรารีที่ต้องการเข้าไป เช่น ลอง `library(MASS)` และศึกษาการใช้คำสั่ง `mvrnorm` ซึ่งเป็นคำสั่งในไลบรารี MASS. ถ้าโหลดไลบรารีไม่ได้ อาจเป็นเพราะเรายังไม่ได้ติดตั้ง ไลบรารี MASS. ดูวิธีการติดตั้งไลบรารีโดยเรียก `?install.packages`.

**การบันทึกข้อมูลและนำเข้าข้อมูล** ตัวอย่างข้างล่างแสดงการบันทึกค่าตัวแปร x. สังเกตการทำงานและศึกษาคำสั่ง `runif`.

```
> x <- runif(3)
> x
[1] 0.3249652 0.7399444 0.9919758
> save(x, file='savX.RData')
```

ตอนนี้หากดูในไฟรีกหอรีที่ทำงานอยู่ จะพบว่ามีไฟล์ `savX.RData` ปรากฏขึ้นมา. การนำเข้าค่าจากไฟล์ที่บันทึกไว้ ก็เพียงเรียก

```
load('savX.RData')
```

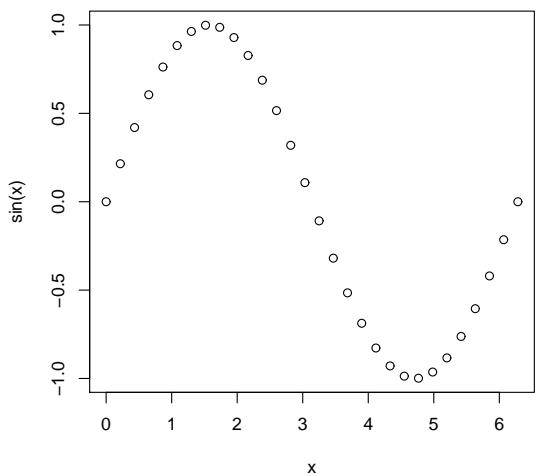
**การวาดกราฟ** โปรแกรมข้างล่างนี้แสดงการวาดกราฟดังแสดงในรูปที่ 1.2.

```
> x <- seq(0, 2*pi, len=30)
> plot(x, sin(x))
```

หากผู้ใช้ต้องการวาดกราฟเป็นเส้นสามารถกำหนดได้โดย การระบุ `type='l'` เข้าไป เป็น `plot(x, sin(x), type='l')` ลองศึกษาการทำงานของ `plot` (โดยลอง `help(plot)`).

## 1.4 แบบฝึกหัด

1. จงสืบค้นจากอินเตอร์เนตเพื่อหาตัวอย่างการประยุกต์ใช้การเรียนรู้ของเครื่องอย่างน้อย 3 ตัวอย่าง.

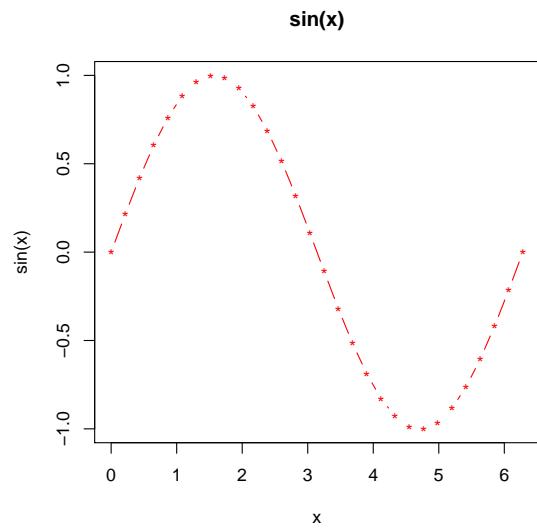
รูปที่ 1.2: ตัวอย่างจากคำสั่ง `plot`

2. จากนิยามการเรียนรู้ของเครื่อง จระบบประสนับการณ์  $E$ , งาน  $T$ , และตัววัดสมรรถนะ  $P$  ของตัวอย่าง จากข้อ 1.
3. จะเขียนฟังชันเพื่อคำนวณค่าดอกเบี้ยทบทั้น โดยรับอาร์กูเมนต์ 3 ค่า คือเงินต้น ดอกเบี้ยต่อปี (0 ถึง 100%) และจำนวนปี เช่น `compound(500000, 7, 20)` เพื่อคำนวณว่าเงินต้น 500,000 บาท คิดดอกเบี้ยที่ 7% ต่อปีทบทั้น เป็นเวลา 20 ปี พอกบปีที่ 20 ยอดเงินรวมจะเป็นเท่าไร.
4. จะเขียนโปรแกรมโดยใช้ฟังชัน `plot` เพื่อให้ได้กราฟดังแสดงในรูป 1.3. สังเกตุการจัดรูปแบบ สี ลักษณะเส้น และข้อกราฟ.
5. จะศึกษาคำสั่ง `par`, `lines`, `legend` และวิธีการดังแสดงในรูป 1.4. สังเกตุทั้งสองกราฟอยู่ในรูปเดียวกัน กราฟทางซ้ายมือแกน  $y$  มีค่าระหว่าง  $-2$  ถึง  $2$ , ระบุชื่อของแกน  $y$  และ แกน  $x$  ตามรูป 1.4.
6. โปรแกรมข้างล่างนี้ใช้รูป 1.5.

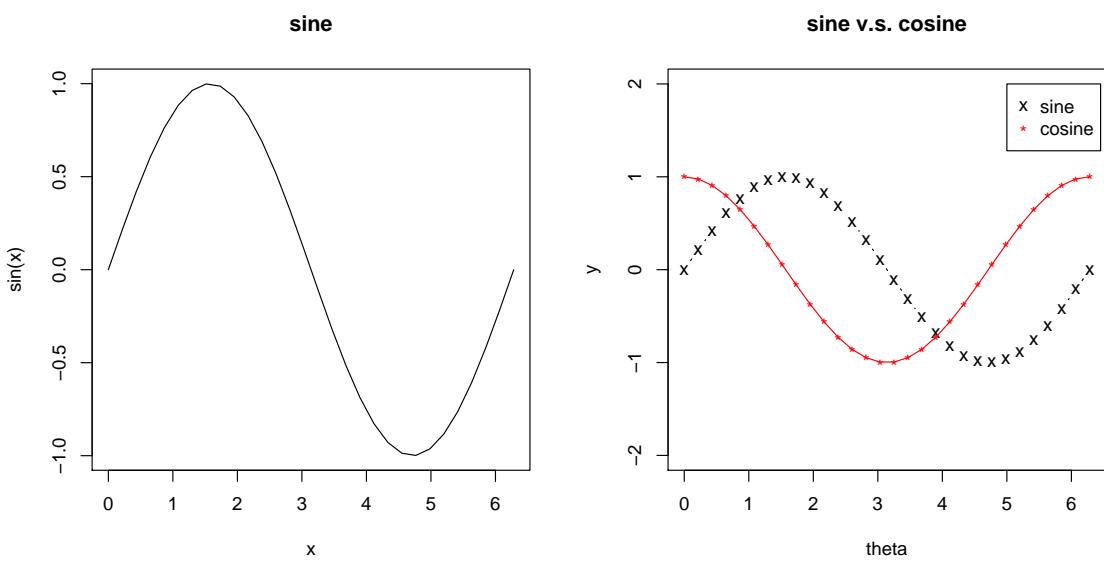
```
> x <- seq(0,100,len=5)
> plot(x, sin(x), type='l')
```

จงวิเคราะห์และอธิบายว่าทำไม่รูปที่ได้ไม่เห็นเป็นรูปโดยคึ่งขึ้นลงเหมือนรูปปั้นที่คุณเคย.

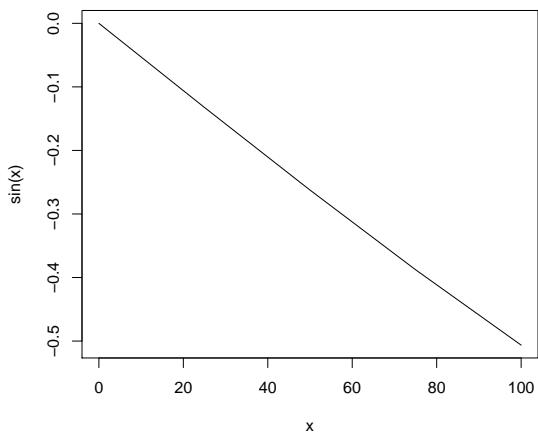
7. โปรแกรมข้างล่างนี้ใช้รูป 1.6.



รูปที่ 1.3: แบบฝึกหัด plot ข้อ 4



รูปที่ 1.4: แบบฝึกหัด plot ข้อ 5



รูปที่ 1.5: แบบฝึกหัดวิเคราะห์ข้อ 6

รายการ 1.1: โค้ดสำหรับรูปแบบฝึกหัดข้อ 7

---

```

1 x <- seq(-0.5, 0.5, len=50)
2
3 plot(x, 10*exp(-x^2)-8.8, type='l', ylab='y',
4      main='10*exp(-x^2)-8.8 v.s. sin(x)')
5 lines(x, sin(x), col='red', lty=2)
6 legend(0, -0.5, c('10*exp(-x^2)-8.8', 'sin(x)'),
7        lty=c(1,2), col=c('black', 'red'))

```

---

จงวิเคราะห์และอธิบายว่าทำไม่ให้กราฟเส้นประ ซึ่งเป็นกราฟของฟังชันไซน์จึงไม่เป็นรูปโค้งขึ้นลงเหมือนรูปไซน์ที่คุณเคย.

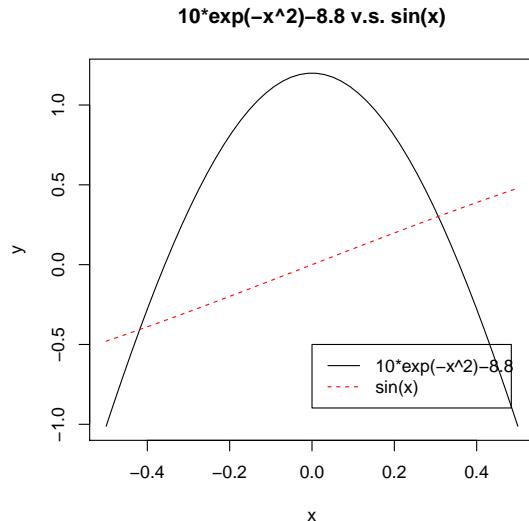
8. จากโปรแกรมและผลการรันดังแสดงข้างล่างนี้ จงอภิรายว่าทำไม่มี  $x$  บางตัวไม่เท่ากับ  $7 \cdot y$  ซึ่ง  $y = x/7$ .

```

> x <- seq(1,10, len=20)
> y <- x/7
> x == 7*y
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[9] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE
[17] TRUE TRUE TRUE TRUE

```

จงวิเคราะห์และอธิบายผลของ  $x == 7*(x/7)$  กับ  $x == 7*x/7$  ประกอบพร้อมอภิรายการประยุกต์ใช้ประเด็นที่ได้เรียนรู้นี้กับสถานการณ์ที่อาจจะเกิดขึ้น รวมถึงความเสี่ยงและโอกาส.



รูปที่ 1.6: แบบฝึกหัดวิเคราะห์ข้อ 7

### 9. โปรแกรมข้างล่างนี้วาดรูป 1.7.

รายการ 1.2: โปรแกรมสำหรับรูปแบบฝึกหัดข้อ 9

---

```

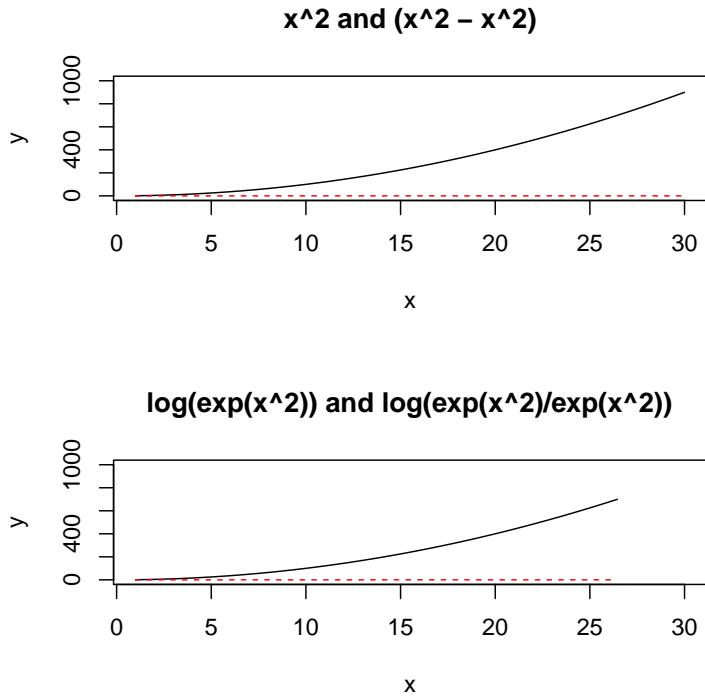
1 x ← seq(1,30, len=50)
2
3 par(mfrow=c(2,1))
4 plot(x, x^2, type='l', ylab='y', ylim=c(0,1000),
5   main='x^2 and (x^2 - x^2)')
6 lines(x, x^2 - x^2, col=2, lty=2)
7
8 plot(x, log(exp(x^2)), type='l', ylab='y', ylim=c(0,1000),
9   main='log(exp(x^2)) and log(exp(x^2)/exp(x^2))')
10 lines(x, log(exp(x^2)/exp(x^2)), col=2, lty=2)

```

---

รูปบนเป็นกราฟของ  $x^2$  (เส้นทึบสีดำ) และ  $x^2 - x^2$  (เส้นประสีแดง). รูปล่างเป็นกราฟของ  $\log(\exp(x^2))$  (เส้นทึบสีดำ) และ  $\log(\exp(x^2)/\exp(x^2))$  (เส้นประสีแดง). จากความรู้คณิตศาสตร์ จะได้  $\log(\exp(x^2)) = x^2$  และ  $\log(\exp(x^2)/\exp(x^2)) = x^2 - x^2$  แต่ทำไมกราฟรูปบนจึงต่างจากรูปล่าง (รูปบนแสดงค่าไปจนถึง  $x = 30$  แต่รูปล่างไม่มีถึง). ลองอภิปราย.

10. จงศึกษาผลจากการคูณกันของเวกเตอร์ทิศทางต่างๆ. จากโปรแกรมอาจเรียกว่า “แบบฝึกหัด” ที่แสดงผลการคูณกันของเวกเตอร์ 2 เวกเตอร์ นั่นคือ  $vq1$  กับ  $v2$  โดย  $vq1$  จะอยู่ที่เดิม แต่  $v2$  จะเปลี่ยนทิศทางไป (20 ค่ารอบทิศทาง ตั้งแต่  $\frac{1}{10}\pi$  จนถึง  $2\pi$ ). โปรแกรมจะวาดรูปประกอบมา 2 รูป แต่ละรูปแสดง 10 ภาพ ย่ออย่างทั้งหมด 20 ภาพย่ออย่าง (แต่ละภาพย่ออย่างแทนแต่ละทิศทางของ  $v2$ ). สังเกตผลจากการคูณที่ระบุไว้ในแต่ละภาพ และทดลองเปลี่ยนค่าองศาของ  $vq1$  ให้เป็นค่าอื่นๆ (เปลี่ยนค่าของ  $\theta_1$  ที่ตอนนี้ระบุเป็น



รูปที่ 1.7: แบบฝึกหัดวิเคราะห์ข้อ 9

$6\pi/10)$  สรุปผลจากการคูณเวกเตอร์ทิศทางต่างๆ และตอบคำถาม ก.-จ. (เวกเตอร์ทุกตัวมีขนาดมากกว่า 0)

รายการ 1.3: โปรแกรมวัดรูปสำหรับแบบฝึกหัดข้อ 10

---

```

1 theta1 <- 6*pi/10
2 vq1 <- matrix(c(cos(theta1), sin(theta1)), 2,1)
3
4 par(mfrow=c(2,5))
5 par(mar=c(2,2,2,1))
6
7 for(i in 1:20){
8
9   plot(0, 0, ylim=c(-1.5,1.5), xlim=c(-1.5,1.5),
10     xlab='x', ylab='y',
11     main=paste('v2 of ', round(i/10,2), ' pi'))
12
13   lines(c(cos(theta1-pi/2), cos(theta1+pi/2)),
14     c(sin(theta1-pi/2), sin(theta1+pi/2)),
15     col='gray', lty=2)
16
17   v2 <- matrix(c(cos(i/10*pi), sin(i/10*pi)), 2,1)

```

```

18
19 vs ← t(vq1) %*% v2
20
21 lines(c(0, vq1[1]), c(0, vq1[2]), col='red')
22 text(vq1[1]+0.2, vq1[2]+0.2,
23   paste(round(theta1/pi,2), 'pi'))
24 lines(c(0, v2[1]), c(0, v2[2]), col='blue')
25 text(0,-0.5, paste('v1*v2=', round(vs,2)))
26
27 if(i == 10){
28   x11()
29   par(mfrow=c(2,5))
30   par(mar=c(2,2,2,1))
31 }
32 }
```

---

- ก. ถ้าเวกเตอร์ A คูณกับเวกเตอร์ B ได้ผลเป็นบวก และเวกเตอร์ A ชี้ไปทิศทางสองนาฬิกา (เทียบเท่า 30 องศาจากแกน x) สิ่งนี้บอกอะไรได้บ้างเกี่ยวกับทิศทางของเวกเตอร์ B
- ข. ถ้าเวกเตอร์ A จากข้อ ก. ขนาดกับเวกเตอร์ C ผลคูณของเวกเตอร์ A กับ C ผลคูณนี้จะเป็นอะไรได้บ้าง และเป็นอะไรไม่ได้บ้าง
- ค. ถ้าเวกเตอร์ A จากข้อ ก. ตั้งฉากกับเวกเตอร์ D ผลคูณของเวกเตอร์ A กับ D ผลคูณนี้จะเป็นอะไรได้บ้าง และเป็นอะไรไม่ได้บ้าง
- ง. ถ้าเวกเตอร์ A จากข้อ ก. คูณกับเวกเตอร์ E ได้ผลเป็นลบ สิ่งนี้บอกอะไรได้บ้างเกี่ยวกับทิศทางของเวกเตอร์ E
- จ. ถ้าหากต้องการผลคูณเวกเตอร์ A จากข้อ ก. กับเวกเตอร์ F ให้ได้ผลเป็นบวกและมีค่ามากที่สุด เวกเตอร์ F ควรมีทิศทางอย่างไร

## บทที่ 2

### การหาค่าดีที่สุด

“... if we’re facing in the right direction,  
all we have to do is keep on walking.”  
—Joseph Goldstein’s The Experience of Insight

“... ถ้าเรามุ่งหน้าไปในทิศทางที่ถูกต้อง  
สิ่งที่เราต้องทำก็แค่เดินไปเรื่อยๆ”  
—ประสบการณ์แห่งความเข้าใจที่ลึกซึ้ง โจเซฟ โกล์ดส్ไตน์

การเรียนรู้ของเครื่องในมุมมองหนึ่ง ก็คือการสร้างโมเดลและการหาค่าพารามิเตอร์ที่ดีที่สุดให้กับโมเดลโดยอาศัยประสบการณ์ที่เกี่ยวข้องช่วย. การหาค่าพารามิเตอร์ที่ดีที่สุดสามารถใช้เทคนิคและความรู้จากศาสตร์และศิลป์ของวิชาการหาค่าดีที่สุด (Optimization) มาช่วยได้. วิชาการหาค่าดีที่สุดมีรายละเอียดมาก และด้วยเนื้อหาของวิชาการหาค่าดีที่สุดเองก็มากพอที่จะเป็นตำราของตัวเองได้. บทนี้จะถกถึงเฉพาะพื้นฐานบางส่วนของศาสตร์การหาค่าดีที่สุดที่พожะช่วยให้เข้าใจกระบวนการเกี่ยวข้องกับศาสตร์การเรียนรู้ของเครื่องบ้างเท่านั้น.

#### 2.1 การหาค่าดีที่สุดพื้นฐาน

การหาค่าดีที่สุด (Optimization) คือการเลือกค่าของปัจจัย (แทนด้วยตัวแปร) ที่มีผลให้เป้าหมาย (แทนด้วยฟังชันของตัวแปร) มีค่าน้อย หรือมากที่สุด. ปัจจัยที่ต้องการเลือก จะเรียกว่า ตัวแปรตัดสินใจ (Decision Variable) และ เป้าหมาย จะเรียกว่า ฟังชันจุดประสงค์ (Objective Function). ตัวอย่างเช่น การเลือกเลือกค่าปัจจัยอุณหภูมิ แทนด้วยตัวแปร  $x$  เพื่ออบ麝เขือเทศได้อร่อยที่สุด โดยวัดจากปริมาณน้ำตาลที่ได้มากที่สุด โดยฟังชัน  $h$  แสดงความสัมพันธ์ระหว่างอุณหภูมิกับปริมาณน้ำตาลที่ได้จากการอบ麝เขือเทศ. ตัวแปร  $x$  คือตัวแปรตัดสินใจ และฟังชัน  $h$  คือฟังชันจุดประสงค์. กรณีนี้คือ การหาค่า  $x$  ที่ทำให้ได้ค่าฟังชัน  $h$  มากที่สุด.

ปัญหาที่เป็นการหาค่าที่ทำให้เป้าหมายมีค่ามากที่สุด เรียกรวมๆว่า ปัญหาค่ามากที่สุด (Maximization Problem). ตัวอย่างการเลือกอุณหภูมิการอบมะเขือเทศเพื่อให้ได้ปริมาณน้ำตาลสูงสุด เป็นปัญหาค่ามากที่สุด. การเลือกชนิดการลงทุนเพื่อให้ได้ผลตอบแทนมากที่สุด การเลือกความเร็วของรถเพื่อวิ่งได้ระยะทางไกลที่สุด (ระยะทางมากที่สุด) สำหรับน้ำมัน 1 ถัง เหล่านี้เป็นตัวอย่างของปัญหาค่ามากที่สุด.

นำองเดียวกัน ปัญหาที่เป็นการหาค่าที่ทำให้เป้าหมายมีค่าน้อยที่สุด เรียกรวมๆว่า ปัญหาค่าน้อยที่สุด (Minimization Problem). ตัวอย่างเช่น การเลือกเส้นทางขับรถจากอนแก่นไปร้อยเอ็ด โดยใช้เวลาเดินทางให้น้อยที่สุด.

ในทางคณิตศาสตร์ ปัญหาค่าน้อยที่สุดกับปัญหาค่ามากที่สุด สามารถแปลงไปมาระหว่างกันได้. นั่นคือ การหาค่า  $x$  ที่ทำให้  $h(x)$  มีค่ามากที่สุด จะเทียบเท่ากับ การหาค่า  $x$  ที่ทำให้  $-h(x)$  มีค่าน้อยที่สุด (ดูแบบฝึกหัดการหาค่าดีที่สุด). ดังนั้น เพื่อความสะดวก ปัญหาการหาค่าดีที่สุด ไม่ว่าจะเป็นปัญหาค่าน้อยที่สุดหรือปัญหาค่ามากที่สุด ก็สามารถเขียนให้อยู่ในรูปปัญหาค่าน้อยที่สุด ได้ดังนี้

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && g(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega \end{aligned} \quad (2.1)$$

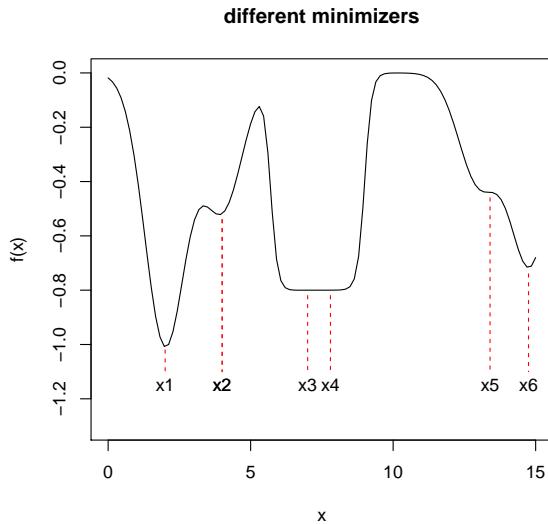
โดย  $\mathbf{x}$  คือตัวแปรตัดสินใจ (ซึ่งอาจมีหลายมิติ  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ ),  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  เป็นฟังชันจุดประสงค์ (หรือ Cost Function), และเซต  $\Omega$  เป็นซับเซตของ  $\mathbb{R}^n$  ที่ระบุค่าของตัวแปรในช่วงที่สนใจ หรือยอมรับได้ เรียกว่า เซตข้อจำกัด (Constraint Set หรือ Feasible Set). ถ้า  $\Omega = \mathbb{R}^n$  (หรือไม่มีข้อจำกัดของตัวแปรตัดสินใจ) จะเรียกปัญหาแบบนี้ว่า ปัญหาการหาค่าดีที่สุดแบบไม่มีข้อจำกัด (Unconstrained Optimization Problem). เพื่อความสะดวก ค่าตัวแปรตัดสินใจที่ทำให้ฟังชันเป้าหมายมีค่าน้อยที่สุด จะเรียกว่า ค่าทำน้อยที่สุด (Minimizer) และนิยามใช้ตัวยกดอกจันทร์ตามตัวแปร เช่น  $\mathbf{x}^*$  เพื่อระบุว่า กำลังพูดถึงค่าทำน้อยที่สุด.

เมื่อพูดถึงการหาค่าน้อยที่สุด สิ่งที่ต้องการคือ ค่าทำน้อยที่สุดและค่าฟังชันเป้าหมายของมัน (ค่าน้อยที่สุด). ค่าทำน้อยที่สุด มี 2 ประเภท ได้แก่ ค่าทำน้อยที่สุดท้องถิ่น และ ค่าทำน้อยที่สุดทั่วหมด.

นิยามของค่าทำน้อยที่สุดท้องถิ่น. สมมติ  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  เป็นฟังชันค่าจริงที่นิยามสำหรับเซต  $\Omega \subset \mathbb{R}^n$ . จุด  $\mathbf{x}^* \in \Omega$  เป็นค่าทำน้อยที่สุดท้องถิ่น (Local Minimizer) ของ  $g$  บนเซต  $\Omega$  ถ้ามีค่า  $\varepsilon > 0$  ที่  $g(\mathbf{x}) \geq f(\mathbf{x}^*)$  สำหรับ ทุกค่า  $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$  และ  $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ .

นิยามของค่าทำน้อยที่สุดทั่วหมด. จุด  $\mathbf{x}^* \in \Omega$  เป็นค่าทำน้อยที่สุดทั่วหมด (Global Minimizer) ของ ฟังชัน  $g$  บนเซต  $\Omega$  ถ้า  $g(\mathbf{x}) \geq g(\mathbf{x}^*)$  สำหรับทุก  $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$ .

จากนิยามข้างต้น กล่าวง่ายๆคือ ค่าทำน้อยที่สุดท้องถิ่นคือค่าตัวแปรตัดสินใจที่ให้ค่าเป้าหมายน้อยกว่าค่าเป้าหมายบริเวณรอบๆ (ท้องถิ่น). ส่วน ค่าทำน้อยที่สุดทั่วหมดคือค่าตัวแปรตัดสินใจที่ให้ค่าเป้าหมายน้อยกว่าค่าเป้าหมายของทุกๆค่าตัวแปรตัดสินใจที่เป็นไปได้ (ทั่วทั้งหมด). ดังนั้น ค่าทำน้อยที่สุดทั่ว



รูปที่ 2.1: ค่าทำน้อยที่สุดต่างๆ

หมาย ก็จะเป็นค่าทำน้อยที่สุดท้องถิ่นด้วยเสมอ. รูป 2.1 แสดงค่าทำน้อยที่สุดต่างๆ โดย  $x_1$  เป็นทั้งค่าทำน้อยที่สุดทั่วหมดและค่าทำน้อยที่สุดท้องถิ่น ส่วน  $x_2$  ถึง  $x_6$  เป็นค่าทำน้อยที่สุดท้องถิ่น.

## 2.2 เจื่อนไขของค่าทำน้อยที่สุดท้องถิ่น

จากฟังชันเป้าหมาย  $g : \mathbb{R}^n \mapsto \mathbb{R}$  อนุพันธ์อันดับหนึ่งของฟังชันเป้าหมาย เขียนย่อเป็น  $Dg$ , คือ

$$Dg = \left[ \frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \dots, \frac{\partial g}{\partial x_n} \right] \quad (2.2)$$

และเกรเดียนต์ (Gradient)  $\nabla g = (Dg)^T$ .

อนุพันธ์อันดับสองของฟังชันเป้าหมาย เรียกว่า เอเชียน (Hessian of  $g$ ) คือ

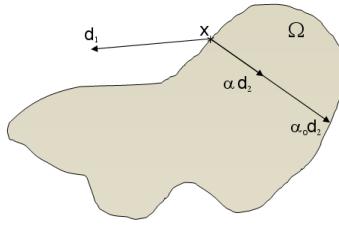
$$\mathbf{G}(\mathbf{x}) = D^2g(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 g}{\partial x_n x_1}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial^2 g}{\partial x_1 x_n}(\mathbf{x}) & \cdots & \frac{\partial^2 g}{\partial x_n^2}(\mathbf{x}) \end{bmatrix} \quad (2.3)$$

ตัวอย่าง. เช่น ถ้า  $g(\mathbf{x}) = 2x_1 + 18x_2 + x_1x_2 - x_1^2 - 4x_2^2$ , ตั้งนั้น

$$\nabla g(\mathbf{x}) = \begin{bmatrix} \frac{\partial g}{\partial x_1}(\mathbf{x}) \\ \frac{\partial g}{\partial x_2}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 2 + x_2 - 2x_1 \\ 18 + x_1 - 8x_2 \end{bmatrix}$$

และ

$$\mathbf{G}(\mathbf{x}) = D^2g(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2}(\mathbf{x}) & \frac{\partial^2 g}{\partial x_2 x_1}(\mathbf{x}) \\ \frac{\partial^2 g}{\partial x_1 x_2}(\mathbf{x}) & \frac{\partial^2 g}{\partial x_2^2}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & -8 \end{bmatrix}$$



รูปที่ 2.2: รูปประกอบช่วยอธิบายทิศทางซึ่งเป็นไปได้. ทิศทาง  $d_1$  ไม่ใช่ทิศทางซึ่งเป็นไปได้ ทิศทาง  $d_2$  คือทิศทางซึ่งเป็นไปได้ที่  $x$

□

จากเซตข้อจำกัด  $\Omega$  ค่าน้อยที่สุดอาจจะอยู่ภายใน  $\Omega$  (กรณีเรียกว่า กรณีภายใน, Interior Case) หรืออาจจะอยู่ที่ขอบของ  $\Omega$  ก็ได้ (กรณีขอบเขต, Boundary Case).

เพื่อความสะดวกการศึกษาทั้งสองกรณี โดยเฉพาะกรณีขอบเขต เราจะนำแนวคิดของทิศทางซึ่งเป็นไปได้ (Feasible Direction) เข้ามา.

**นิยามทิศทางซึ่งเป็นไปได้.** เวกเตอร์  $\mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}$  เป็นทิศทางซึ่งเป็นไปได้ (Feasible Direction) ที่  $\mathbf{x} \in \Omega$  ก็ต่อเมื่อมีค่า  $\alpha_0 > 0$  ที่ทำให้  $\mathbf{x} + \alpha\mathbf{d} \in \Omega$  สำหรับทุกๆ ค่าของ  $\alpha \in [0, \alpha_0]$ . ดูรูป 2.2 และ 2.3 ประกอบ.

จากนิยามทิศทางซึ่งเป็นไปได้ เพื่อศึกษาคุณสมบัติหรือเงื่อนไขที่สามารถใช้ช่วยในการแก้ปัญหาการหาค่าน้อยที่สุด พิจารณาการวิเคราะห์ต่อไปนี้.

ถ้าให้ พังชั่นค่าจริง  $g : \mathbb{R}^n \mapsto \mathbb{R}$  และ  $\mathbf{d}$  เป็นทิศทางซึ่งเป็นไปได้ ที่  $\mathbf{x} \in \Omega$  และอนุพันธ์เชิงทิศทาง (Directional Derivative) ของ  $g$  ในทิศทาง  $\mathbf{d}$  ซึ่งเขียนย่อด้วย  $\partial g / \partial \mathbf{d}$ , คือ

$$\frac{\partial g}{\partial \mathbf{d}}(\mathbf{x}) = \lim_{\alpha \rightarrow 0} \frac{g(\mathbf{x} + \alpha\mathbf{d}) - g(\mathbf{x})}{\alpha}. \quad (2.4)$$

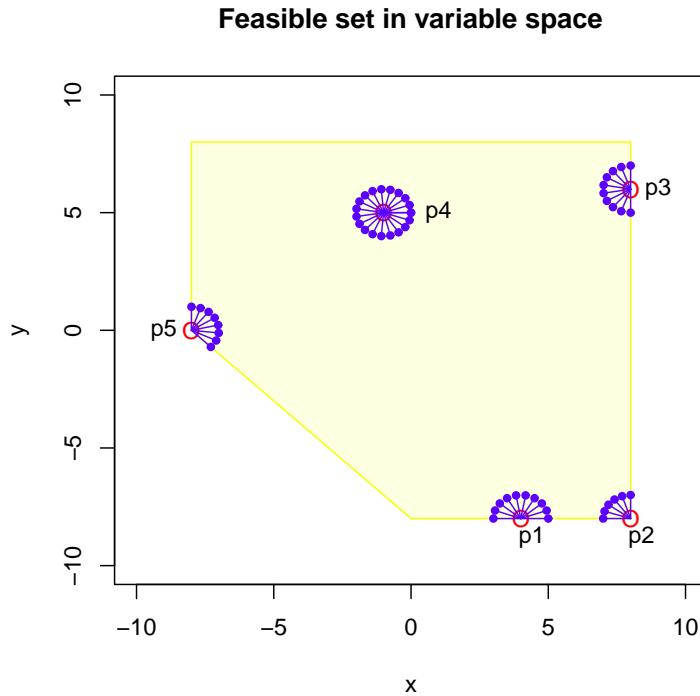
ถ้า  $\|\mathbf{d}\| = 1$  และ ค่า  $\partial g / \partial \mathbf{d}$  ก็คือ อัตราการเพิ่มของค่าพังชั่น  $g$  ที่  $\mathbf{x}$  ในทิศทางของ  $\mathbf{d}$  นั่นเอง.

สำหรับการคำนวณค่า  $\partial g / \partial \mathbf{d}$  สมมติว่ามีค่า  $\mathbf{x}$  กับ  $\mathbf{d}$  อยู่ ดังนั้น  $g(\mathbf{x} + \alpha\mathbf{d})$  เป็นพังชั่นของ  $\alpha$  หรือ

$$\frac{\partial g}{\partial \mathbf{d}}(\mathbf{x}) = \left. \frac{d}{d\alpha} g(\mathbf{x} + \alpha\mathbf{d}) \right|_{\alpha=0}$$

เมื่อใช้กฎลูกโซ่ (Chain Rule) เข้าไป จะได้

$$\frac{dg(\mathbf{x} + \alpha\mathbf{d})}{d\alpha} = \frac{dg(\mathbf{x} + \alpha\mathbf{d})}{d(\mathbf{x} + \alpha\mathbf{d})} \cdot \frac{d(\mathbf{x} + \alpha\mathbf{d})}{d\alpha} = (\nabla g(\mathbf{x})^T) \cdot (\mathbf{d}) = \mathbf{d}^T \nabla g(\mathbf{x}).$$



รูปที่ 2.3: ทิศทางซึ่งเป็นไปได้ที่จุดต่างๆ 5 จุด ได้แก่ P1 ถึง P5 ภายในเขตข้อจำกัด (ภายในพื้นที่แรเงา). เส้นเล็กที่วัดอุกมาจากแต่ละจุด คือ ตัวอย่างของทิศทางซึ่งเป็นไปได้ต่างๆ ที่จุดนั้น เช่นที่จุด P1 ที่อยู่ชิดขอบล่างของเขตข้อจำกัด มีทิศทางซึ่งเป็นไปได้ต่างๆ ได้ตั้งแต่ทิศทางสามานาพิกาทวนเข็มปีกนกถึงเก้านาพิกา). สังเกตุทิศทางซึ่งเป็นไปได้ต่างๆ ที่จุดที่อยู่ขอบของเขตข้อจำกัด (อาทิ จุด P1, P2, P3, P5) เทียบกับ จุด P4 ที่อยู่ภายนอกในเขตข้อจำกัด. หมายเหตุ เพื่อความสะดวกหัวเรกเตอร์วัดด้วยจุดแทนลูกศร

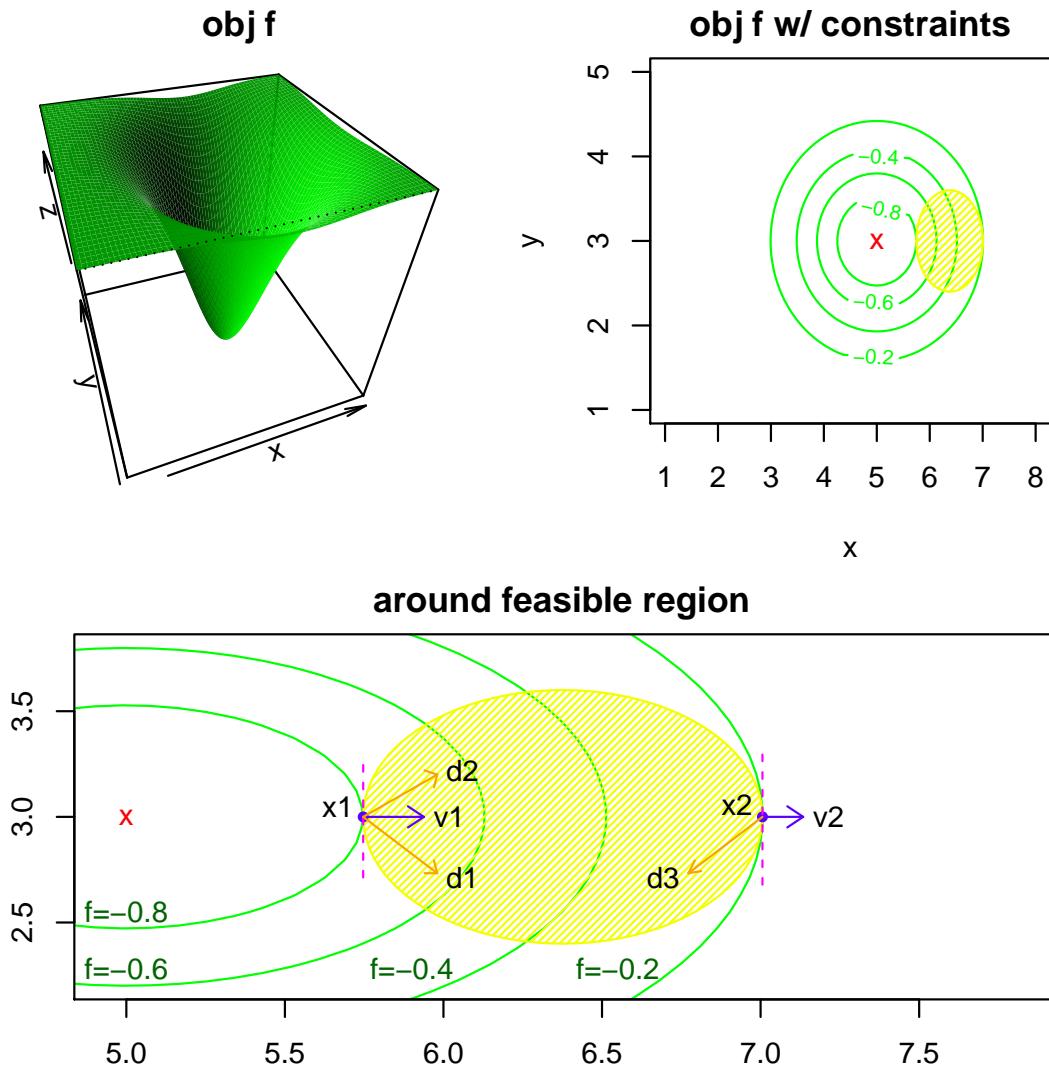
**First-Order Necessary Condition (FONC).** เจื่อนใจ เป็นอันดับแรก (ของตัวทำต่ำสุด) คือ ให้  $\Omega \subset \mathbb{R}^n$  และ  $g \in \mathcal{C}^1$  เป็นฟังชันค่าจริงบนเซต  $\Omega$  แล้ว ถ้า  $\mathbf{x}^*$  เป็นตัวทำต่ำสุดท้องถิ่นของฟังชัน  $g$  บนเซต  $\Omega$  แล้ว สำหรับแต่ละทิศทางซึ่งเป็นไปได้  $\mathbf{d}$  ที่  $\mathbf{x}^*$  ต้องได้ว่า อนุพันธ์เชิงทิศทางที่จุดนั้นต้องมากกว่าหรือเท่ากับศูนย์. นั่นคือ

$$\mathbf{d}^T \nabla g(\mathbf{x}^*) \geq 0 \quad (2.5)$$

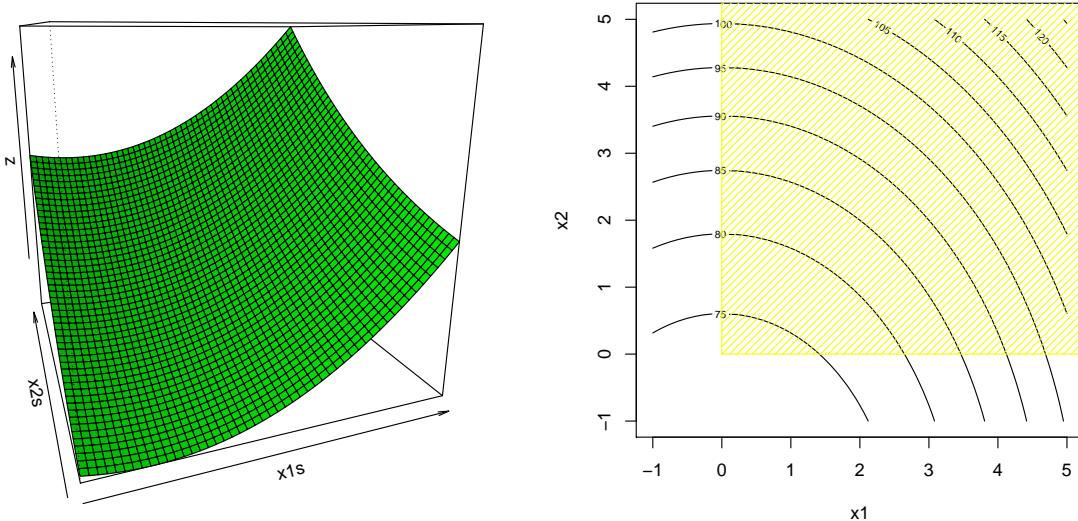
สำหรับทุกๆ  $\mathbf{d}$ . หมายเหตุ  $g \in \mathcal{C}^1$  หมายถึง ฟังชัน  $g$  เป็นฟังชันค่าต่อเนื่อง และมีอนุพันธ์อันดับหนึ่งทุกจุดบนเซตจำกัด.

รูป 2.4 (รูปล่าง) แสดงค่าทำน้อยที่สุดที่จุด  $x_1$  (ที่ขอบซ้ายของเขตข้อจำกัด) พร้อมเกรเดียนต์ ( $v_1$ ) และตัวอย่างทิศทางซึ่งเป็นไปได้ ( $d_1$  กับ  $d_2$ ) ซึ่งเมื่อตรวจสอบสมบัติจะได้ตามเจื่อนใจ เป็นอันดับแรก (FONC).

เมื่อพิจารณาเบริญบที่ยบกับจุด  $x_2$  ที่ขอบขวาของเขตข้อจำกัด ที่จุด  $x_2$  มีเกรเดียนต์ ( $v_2$ ) และตัวอย่างทิศทางซึ่งเป็นไปได้ ( $d_3$ ), จุด  $x_2$  ไม่ผ่านเจื่อนใจ เป็นอันดับแรก เพราะมีทิศทางซึ่งเป็นไปได้  $d_3$  ที่ทำให้  $\mathbf{d}_3^T \nabla f(\mathbf{x}_2) < 0$  (สังเกตุ ทิศทางของเวกเตอร์  $v_2$  และ  $d_3$ ). เมื่อไม่ผ่านเจื่อนใจ เป็นอันดับแรก จึงมั่นใจได้ว่า จุด  $x_2$  ไม่ใช่ค่าทำน้อยที่สุด.



รูปที่ 2.4: รูปบนซ้ายแสดงฟังชันเป้าหมายที่วัดแบบสามมิติ. รูปบนขวาแสดงค่าฟังชันเป้าหมายที่วัดแบบสองตัวแปร พร้อมแสดงเขตข้อจำกัด (พื้นที่ภายในวงกลมพื้นลาย), จุดที่ฟังชันเป้าหมายมีค่าน้อยที่สุด (แต่อยู่นอกเขตข้อจำกัด). รูปล่างแสดงภาพขยายของรูปบนขวา โดยขยายเพื่อเน้นบริเวณเขตข้อจำกัด. ที่จุด  $x_1$  เป็นค่าทำน้อยที่สุด. สังเกตความสัมพันธ์ระหว่างเกรเดียนต์ (เวกเตอร์  $v$ ) กับทิศทางซึ่งเป็นไปได้ (เวกเตอร์  $d$ ) เปรียบเทียบความสัมพันธ์นี้ระหว่างจุด  $x_1$  (ค่าทำน้อยที่สุด) กับ จุด  $x_2$  ที่ขอบของเขตข้อจำกัดอิกต้าน



รูปที่ 2.5: รูปซ้ายแสดงฟังชันเป้าหมายที่วัดแบบสามมิติ. รูปบนขวาแสดงค่าฟังชันเป้าหมายที่วัดแบบสองทั่วพร้อมเขตข้อจำกัด (พื้นที่เรցา)

เงื่อนไขจำเป็นอันดับแรกสำหรับกรณีภายใน. หาก  $\Omega \subset \mathbb{R}^n$  และ  $g \in \mathcal{C}^1$  เป็นฟังชันค่าจริงบนเขต  $\Omega$  แล้ว ถ้า  $\mathbf{x}^*$  เป็นตัวทำต่ำสุดของฟังชัน  $g$  บนเขต  $\Omega$  และ ถ้า  $\mathbf{x}^*$  เป็นจุดที่อยู่ภายในของเขต  $\Omega$  แล้วดังนี้

$$\nabla g(\mathbf{x}^*) = \mathbf{0}. \quad (2.6)$$

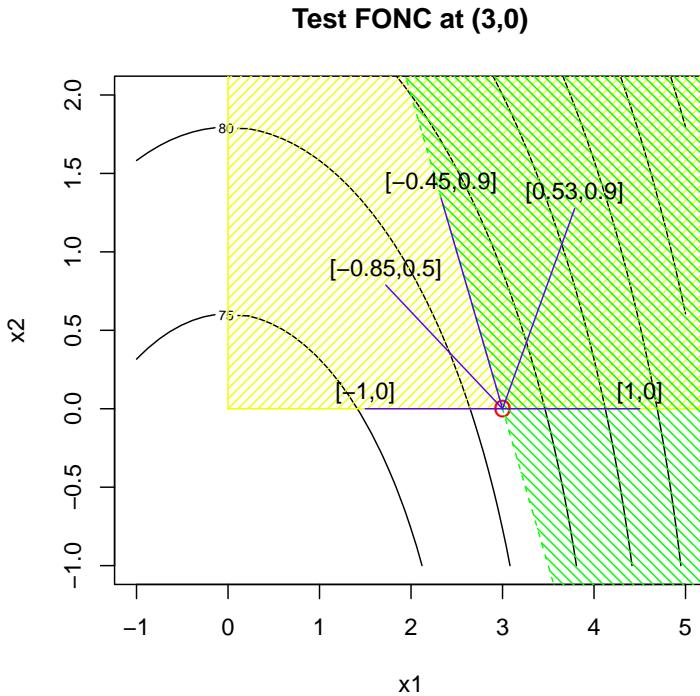
ตัวอย่าง. ปัญหาการค่าต่ำสุด

$$\begin{aligned} & \text{minimize}_{x_1, x_2} \quad g(x_1, x_2) = x_1^2 + 0.5x_2^2 + 3x_2 + 73 \\ & \text{subject to} \quad x_1, x_2 \geq 0. \end{aligned}$$

รูป 2.5 แสดงค่าฟังชันเป้าหมายพร้อมขอบเขตของเขตข้อจำกัด. ลองตรวจสอบเงื่อนไขจำเป็นอันดับแรกของจุด  $(4, 3), (3, 0), (0, 0)$  ดูว่าเป็นอย่างไรบ้าง. ค่าเกรเดียนต์ คือ  $\nabla g(\mathbf{x}) = [2x_1, x_2 + 3]^T$ .

- ที่จุด  $(4, 3)$ ,  $\nabla g(\mathbf{x} = [4, 3]^T) = [2x_1, x_2 + 3]^T = [8, 6]^T$  และ ที่จุด  $(4, 3)$  (เป็นจุดภายใน, interior point),  $\nabla g(\mathbf{x}) = [8, 6]^T \neq \mathbf{0}$  ดังนั้นจุดนี้จึงไม่ผ่านเงื่อนไขจำเป็นอันดับแรก.
- ที่จุด  $(3, 0)$ ,  $\nabla g(\mathbf{x} = [3, 0]^T) = [6, 3]^T$ , ดังนั้น  $\mathbf{d}^T \nabla g(\mathbf{x}) = 6d_1 + 3d_2$  โดย  $\mathbf{d} = [d_1, d_2]^T$ . ถ้าจะผ่านเงื่อนไขจำเป็นอันดับแรก ค่าของ  $6d_1 + 3d_2 \geq 0$  สำหรับทุกค่าของ  $\mathbf{d}$  ที่จุด  $(3, 0)$  หรือ  $6d_1 + 3d_2 \geq 0 \equiv d_1 \geq -d_2/2$ . แต่พอบว่า มีทิศทางซึ่งเป็นไปได้ที่จุด  $(3, 0)$  เช่น  $\mathbf{d} = [-1, 0]^T$  จะทำให้  $\mathbf{d}^T \nabla f = -6 < 0$  ทำให้ จุด  $(3, 0)$  ไม่ผ่านเงื่อนไขจำเป็นอันดับแรก ดูรูป 2.6 ประกอบ.
- ที่จุด  $(0, 0)$ ,  $\nabla f(\mathbf{x} = [0, 0]^T) = [0, 3]^T$ , ดังนั้น  $\mathbf{d}^T \nabla f(\mathbf{x}) = 3d_2$ , ซึ่งที่จุด  $(0, 0)$ ,  $d_2 \geq 0$  จึงทำให้  $\mathbf{d}^T \nabla f(\mathbf{x}) \geq 0$  สำหรับทุกๆ ทิศทางซึ่งเป็นไปได้. จุด  $(0, 0)$  จึงผ่านเงื่อนไขจำเป็นอันดับแรก.

□



รูปที่ 2.6: ตัวอย่าง ทิศทางซึ่งเป็นไปได้ที่จุด  $(3,0)$ . ทิศทาง เช่น  $[-1,0]$  และ  $[-0.85, 0.5]$  จะทำให้  $\mathbf{d}^T \nabla f < 0$ .

**Second-Order Necessary Condition (SONC).** เงื่อนไขจำเป็นอันดับสอง (ของตัวทำต่ำสุด) คือ ให้  $\Omega \subset \mathbb{R}^n$  และ  $g \in \mathcal{C}^2$  เป็นฟังชันค่าจริงบนเซต  $\Omega$ , ให้  $\mathbf{x}^*$  เป็นตัวทำต่ำสุดท้องถิ่นของฟังชัน  $g$  บนเซต  $\Omega$ , สำหรับทุกๆ ทิศทางซึ่งเป็นไปได้  $\mathbf{d}$  ที่  $\mathbf{x}^*$ , ถ้า  $\mathbf{d}^T \nabla g(\mathbf{x}^*) = 0$ , และ

$$\mathbf{d}^T \mathbf{G}(\mathbf{x}^*) \mathbf{d} \geq 0, \quad (2.7)$$

เมื่อ  $\mathbf{G}$  เป็น矩阵ของ  $g$ . หมายเหตุ  $g \in \mathcal{C}^2$  หมายถึง ฟังชัน  $g$  เป็นฟังชันค่าต่อเนื่อง และมีอนุพันธ์ อันดับสองทุกจุดบนเซตจำกัด (สามารถหาเรียง  $\mathbf{G}$  ได้).

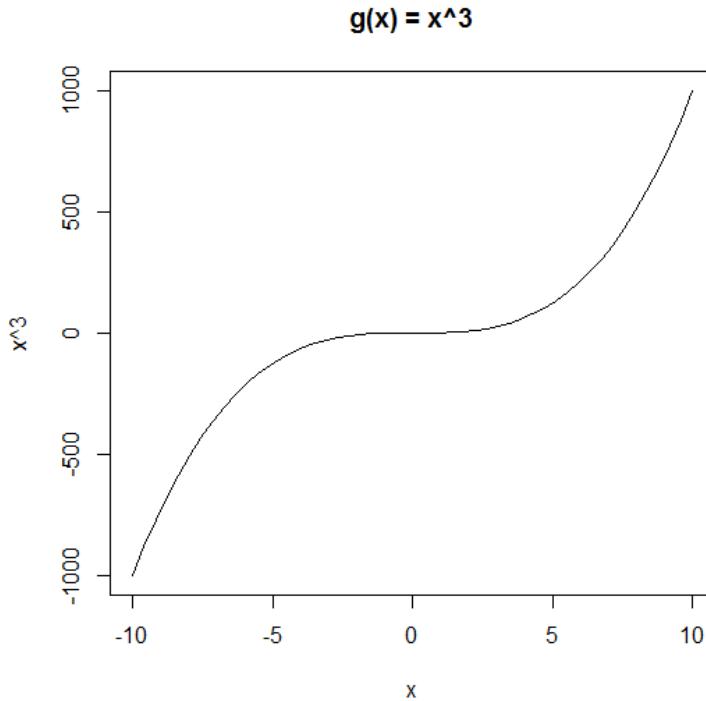
เงื่อนไขจำเป็นอันดับสองสำหรับกรณีภายใน. ให้  $\mathbf{x}^*$  เป็นจุดที่อยู่ภายในเซตจำกัด  $\Omega \subset \mathbb{R}^n$ , ถ้า  $\mathbf{x}^*$  เป็นตัวทำน้อยที่สุดท้องถิ่นของฟังชันเป้าหมาย  $g : \Omega \rightarrow \mathbb{R}, g \in \mathcal{C}^2$ , และ

$$\nabla g(\mathbf{x}^*) = \mathbf{0},$$

และ  $\mathbf{G}(\mathbf{x}^*)$  เป็นบวกกึ่งแหนบอน (Positive Semi-Definite,  $\mathbf{G}(\mathbf{x}^*) \geq 0$ ) ซึ่งก็คือ

$$\mathbf{d}^T \mathbf{G}(\mathbf{x}^*) \mathbf{d} \geq 0, \text{ for all } \mathbf{d} \in \mathbb{R}^n.$$

ตัวอย่าง. ฟังชันเป้าหมาย  $g(x) = x^3$ , ที่  $x = 0$  ค่า  $g'(0) = 0$  และ  $g''(0) = 0$  ดังนั้น ที่จุด  $x = 0$  ผ่านทั้งเงื่อนไขจำเป็น อันดับหนึ่งและอันดับสอง. แต่  $x = 0$  ไม่ใช่ค่าทำน้อยที่สุด ดังแสดงในรูป 2.7. □

รูปที่ 2.7: ตัวอย่าง ค่าฟังชัน  $g(x) = x^3$ 

ตัวอย่าง. พังชันเป้าหมาย  $g(\mathbf{x}) = x_1^2 - x_2^2$  มีกรเดียนต์  $\nabla g(\mathbf{x}) = [2x_1, -2x_2]^T$  และ เอเชียน

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}.$$

ที่  $\mathbf{x} = [0, 0]^T$ ,  $\nabla g(\mathbf{x}) = \mathbf{0}$  ซึ่งผ่านเงื่อนไขจำเป็นอันดับหนึ่ง แต่ค่าเอเชียนไม่ได้เป็นบวกก็ยังแหน่งอน เนื่องจาก มีพิเศษทางซึ่งเป็นไปได้ ที่ทำให้  $\mathbf{d}^T \mathbf{G} \mathbf{d} < 0$  เช่น  $\mathbf{d} = [0, 1]^T$ . จุด  $\mathbf{x} = [0, 0]^T$  จึงไม่ผ่านเงื่อนไขจำเป็นอันดับสอง ซึ่งยืนยันว่าจุดนี้ไม่ใช่ตัวทำน้อยที่สุด. รูป 2.8 แสดงค่าฟังชันเป้าหมายของตัวอย่างนี้.

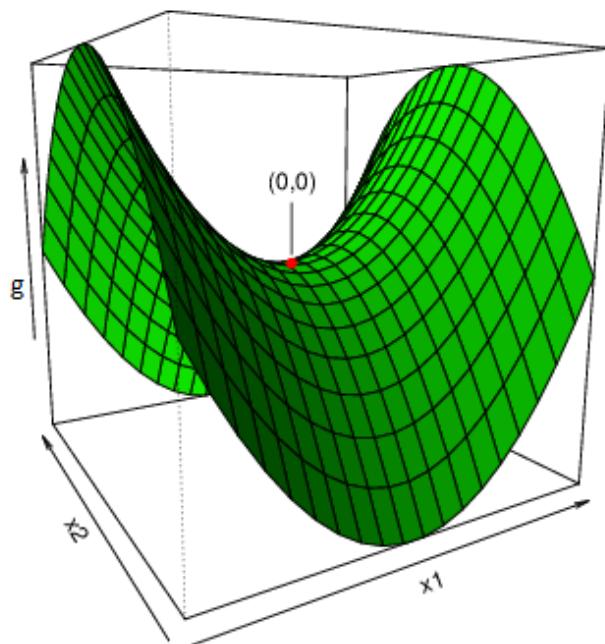
□

สังเกตว่า ถ้าจุดใดไม่ผ่านเงื่อนไขจำเป็น จุดนั้นไม่ใช่ตัวทำต่ำสุด. แต่จุดที่ผ่านเงื่อนไขจำเป็น จุดนั้นอาจจะไม่ใช่ตัวทำต่ำสุดก็ได้.

**Second-Order Sufficient Condition (SOSC), Interior Case** เงื่อนไขเพียงพออันดับสอง สำหรับกรณีภายใน คือ ให้  $g \in \mathcal{C}^2$  และ  $\mathbf{x}^*$  เป็นจุดที่อยู่ภายในเขตข้อจำกัด, ถ้า  $\nabla g(\mathbf{x}^*) = \mathbf{0}$  และ  $\mathbf{G}(\mathbf{x}^*) > 0$  แล้ว  $\mathbf{x}^*$  เป็นตัวทำน้อยสุดท้องถิ่นอย่างแท้จริง (Strict Local Minimizer) ของ  $g$ .

ตัวอย่าง. พังชันเป้าหมาย  $g(\mathbf{x}) = x_1^2 + x_2^2$ , กรเดียนต์  $\nabla g(\mathbf{x}) = [2x_1, 2x_2]^T$  และค่ากรเดียนต์ จะเท่ากับ  $\mathbf{0}$  ก็เมื่อ  $\mathbf{x} = [0, 0]^T$  เท่านั้น และเนื่องจากค่าเอเชียนเป็น

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



รูปที่ 2.8: ตัวอย่างค่าฟังชัน  $g(x) = x_1^2 - x_2^2$ . แกนตั้งเป็นค่าฟังชัน  $g(x)$ . จุด  $(0,0)$  ผ่านเงื่อนไขจำเป็นอันดับหนึ่ง แต่ไม่ผ่านเงื่อนไขจำเป็นอันดับสอง.

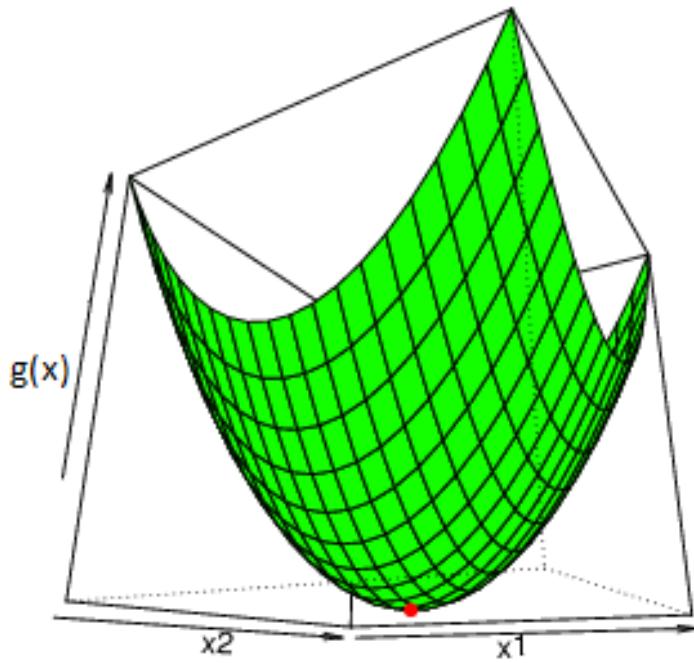
ซึ่งเป็นบวกແນื่อง ( $\mathbf{G} > 0$ ) เพราะ  $\mathbf{d}^T \mathbf{G} \mathbf{d} = 2d_1^2 + 2d_2^2 > 0$  ไม่ว่าค่า  $d_1$  หรือ  $d_2$  จะเป็นอะไร ( $d_1, d_2 \in \mathbb{R}$ ). จุด  $(0,0)$  ผ่านเงื่อนไขทั้งเงื่อนไขจำเป็นอันดับหนึ่ง อันดับสอง และเงื่อนไขเพียงพออันดับสอง, จุด  $(0,0)$  จึงเป็นตัวทำค่าน้อยที่สุดของฟังชันนี้ (รูป 2.9 แสดงฟังชันเป้าหมาย และจุด  $(0,0)$ ).  $\square$

วิธีการหาค่าดีที่สุดมีหลายวิธี เนื้อหาของการหาค่าดีที่สุดมีมาก. บทนี้อภิปรายเพียงบางวิธีเท่านั้น เพื่อผู้อ่านจะพอได้เห็นภาพคร่าวๆ ว่า ศาสตร์การหาค่าดีที่สุดมีวิธีการทำอย่างไรบ้าง.

### 2.3 การค้นหาแบ่งช่วงทองคำ

การค้นหาแบ่งช่วงทองคำ (Golden Section Search) เป็นหนึ่งในวิธีการค้นหาตามแนวเลี้ยว (Line Search) ที่ใช้หาค่าทำน้อยที่สุดสำหรับฟังชันเป้าหมาย  $g : [a_0, b_0] \rightarrow \mathbb{R}$  โดย  $[a_0, b_0] \subset \mathbb{R}$  และฟังชันเป้าหมาย  $g$  เป็นฐานนิยมเดียว (Unimodal, ซึ่งหมายความว่า ฟังชัน  $g$  มีค่าทำน้อยที่สุดท้องถิ่นค่าเดียว (มีหลุมเดียว) ดูรูป 2.10 ประกอบ)

แนวคิดของวิธีค้นหาแบ่งช่วงทองคำ คือ เริ่มต้นจากช่วง  $[a_0, b_0]$  และจะเลือกประเมินค่าของฟังชันเป้าหมายบางจุด โดยพยายามจะทำการประเมินให้น้อยที่สุด. จากจุดที่ประเมินค่า เราจะขยายช่วง  $[a_0, b_0]$  ให้แคบลง และจะทำแบบนี้ไปจนในที่สุดได้ช่วงที่แคบพอก และก็จะได้ค่าทำน้อยที่สุดตามความแม่นยำที่ต้องการ.



รูปที่ 2.9: ตัวอย่าง ค่าฟังชัน  $g(x) = x_1^2 + x_2^2$ . แกนตั้งแทน  $g(x)$ . จุด  $(0,0)$  ผ่านทั้งเงื่อนไขจำเป็นอันดับหนึ่ง อันดับสอง และเงื่อนไขเพียงพออันดับสอง

เพื่อจะขยับช่วงให้แคบลง เราต้องรู้ค่าของฟังชันระหว่างช่วงอย่างน้อย 2 ค่า เรียกว่า ค่า  $g_a$  กับค่า  $g_b$  แทนค่าฟังชันเป้าหมายที่จุด  $a_1$  กับที่จุด  $b_1$  ตามลำดับ. ดูรูป 2.11 ประกอบ. สำหรับฟังชันนี้มีเดียวกับค่าฟังชัน  $g_a < g_b$  (สองภาพบนของรูป 2.11) สิ่งที่บอกได้คือ ค่าทำน้อยที่สุดอาจอยู่ระหว่างช่วงย่อย  $[a_0, a_1]$  (กรณีภาพบนซ้าย) หรือระหว่างช่วงย่อย  $[a_1, b_1]$  (กรณีภาพบนขวา) แต่ไม่อยู่ระหว่างช่วงย่อย  $[b_1, b_0]$  แน่น ( เพราะ ถ้าค่าทำน้อยที่สุดอาจอยู่ระหว่างช่วงย่อย  $[b_1, b_0]$  ฟังชัน  $g(x)$  ก็ไม่ใช่ฐานนิยมเดียวกัน ). ดังนั้นถ้าหากรู้ว่า  $g_a < g_b$  เราสามารถทำช่วงให้แคบลงได้โดยตัดช่วงย่อย  $[b_1, b_0]$  ออก. ทำนองเดียวกัน ถ้าค่าฟังชัน  $g_a > g_b$  กับบอกได้ว่าค่าทำน้อยที่สุดไม่อยู่ระหว่างช่วงย่อย  $[a_0, a_1]$  แน่น และก็สามารถตัดช่วงย่อย  $[a_0, a_1]$  ออกได้. เราเก็บสามารถทำแบบนี้ซ้ำๆ ได้จนกว่าจะได้ช่วงที่แคบพอที่จะได้ค่าความแม่นยำที่ต้องการ.

วิธีค้นหาแบบช่วงของคำจะทำการแบ่งอย่างสมมาตร. นั่นคือ ช่วงย่อยด้านซ้ายกับช่วงย่อยด้านขวาเท่าๆ กัน แต่ไม่เกยกัน

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0) \quad (2.8)$$

โดยที่  $\rho < \frac{1}{2}$ . เงื่อนไขของ  $\rho$  มีเพื่อป้องกันไม่ให้ช่วงย่อย  $[a_0, a_1]$  เกยกับช่วงย่อย  $[b_1, b_0]$ . ดูรูป 2.12 ประกอบ.

การเลือกค่าของอัตราส่วน  $\rho$  จะเลือกค่าที่ทำให้เวลาขยับช่วงเข้าไปแล้ว สามารถซ่อนประกายด้วยการคำนวณได้บ้าง ดังแสดงในรูป 2.13 จากขั้นตอนที่ 1, เราเมื่อค่าฟังชันที่จุด  $a_1$  ซึ่งคือ  $g_a^{(1)}$  กับค่าฟังชันที่จุด

$b_1$  ซึ่งคือ  $g_b^{(1)}$  อยู่<sup>1</sup> หลังจากเปรียบเทียบแล้ว หากพบว่า  $g_a^{(1)} < g_b^{(1)}$ , สามารถยับช่วงเข้ามาได้ โดยให้  $b_0^{(2)} = b_1^{(1)}$ . ถ้าการเลือกค่าของ  $\rho$  ทำให้  $b_1^{(2)}$  ไปตรงกับ  $a_1^{(1)}$  ได้ จะช่วยทำให้ในขั้นตอนที่ 2 ต้องการคำนวณแค่ค่าพังชั้นที่  $a_1^{(2)}$  ไม่จำเป็นต้องคำนวณค่าที่  $b_1^{(2)}$  อีก เพราะว่าสามารถใช้  $g_b^{(2)} = g_a^{(1)}$  ได้เลย.

ดังนั้น เมื่อพิจารณาจาก  $\rho(b_0^{(2)} - a_0^{(2)}) = b_0^{(2)} - b_1^{(2)}$  และ การพิจารณาสัดส่วนต่างๆ ในเทอมของ  $\rho$  (ดูรูป 2.14 ประกอบ) เราจะพบว่า

$$\rho(b_1^{(1)} - a_0^{(1)}) = b_1^{(1)} - a_1^{(1)}. \quad (2.9)$$

เพื่อความกระหึ่ด สมมติ  $b_0^{(1)} - a_0^{(1)} = L_1 = 1$  ดังนั้น  $b_1^{(1)} - a_0^{(1)} = 1 - \rho$  และ  $b_1^{(1)} - a_1^{(1)} = 1 - 2\rho$  และแทนความสัมพันธ์เหล่านี้เข้าไปในสมการ 2.9 เราจะได้ว่า

$$\begin{aligned} \rho(1 - \rho) &= 1 - 2\rho \\ \rho^2 - 3\rho + 1 &= 0 \end{aligned}$$

ซึ่งเมื่อแก้สมการหาค่า  $\rho$  แล้วจะได้

$$\begin{aligned} \rho_1 &= \frac{3 + \sqrt{5}}{2} \approx 2.618 \\ \rho_2 &= \frac{3 - \sqrt{5}}{2} \approx 0.382 \end{aligned} \quad (2.10)$$

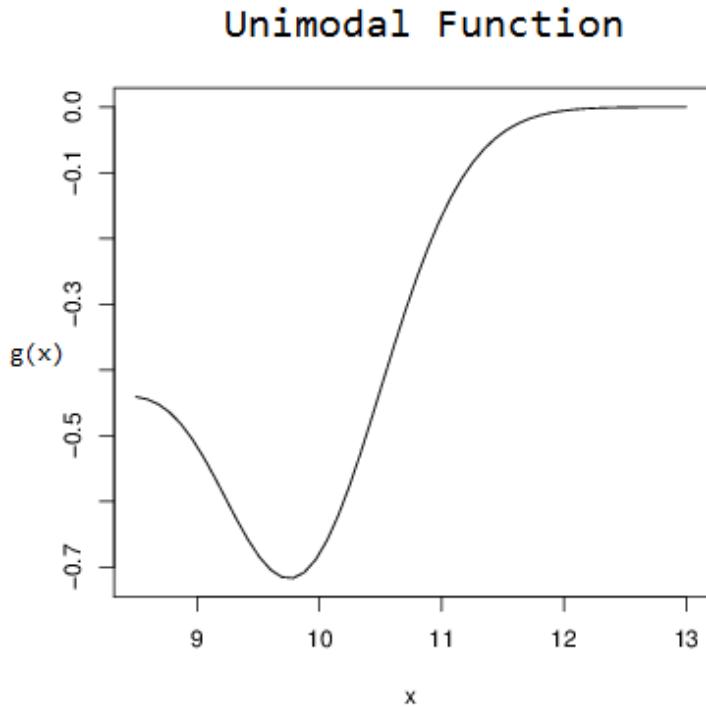
แต่จากเงื่อนไขที่ต้องการ  $\rho < 0.5$  (เพื่อไม่ให้ช่วงย่ออยู่เกยกัน), ดังนั้นจึงได้ว่า  $\rho = 0.382$ . สัดส่วนนี้จะตรงกับสิ่งที่นักเรขาคณิตกรีกโบราณศึกษา และเรียกว่า สัดส่วนทองคำ (Golden Section).

ขนาดช่วงของตัวแปรจะลดลงในอัตราส่วน  $1 - \rho \approx 0.61803$  ต่อการทำแบ่งช่วงทองคำ 1 ขั้นตอน เพราะฉะนั้นถ้าทำ  $N$  ขั้นตอน ขนาดช่วงจะแคบเป็น  $(1 - \rho)^N \approx (0.61803)^N$  เท่าของช่วงเริ่มต้น.

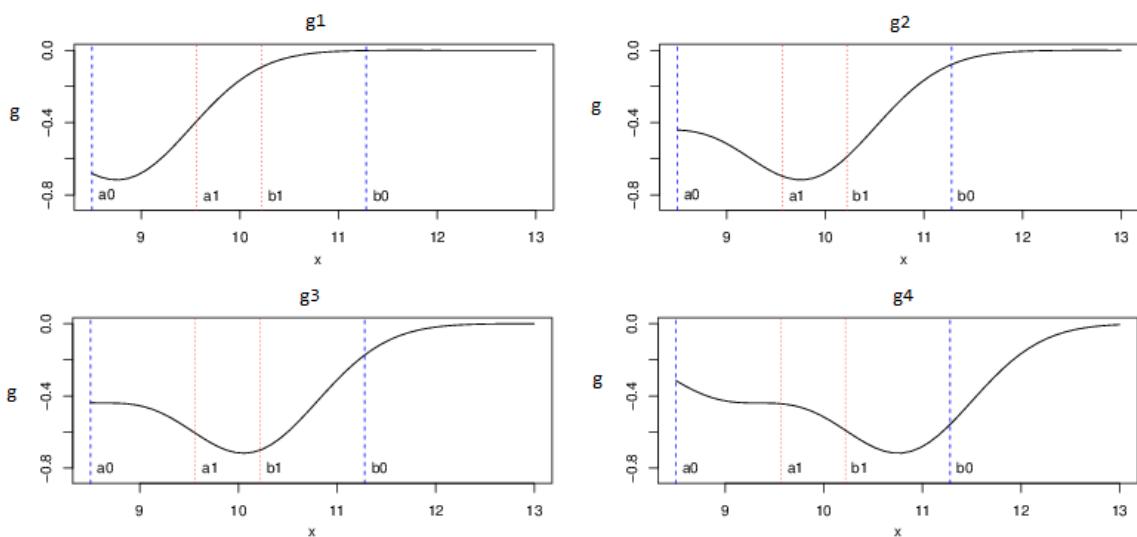
ตัวอย่าง. การใช้การค้นหาแบ่งช่วงทองคำ เพื่อหาค่าทำน้อยที่สุดของฟังชัน  $g(x) = -0.4\exp(-(x-8)^2) - 0.7\exp(-(x-9.8)^2)$  ภายในช่วง [8.5, 13]. สมมติหากต้องการความแม่นยำของคำตอบให้มีค่าน้อยกว่า 1, จะต้องทำการค้นหาแบ่งช่วงทองคำ  $N$  ขั้นตอน โดย  $(13 - 8.5) \cdot (0.61803)^N < 1$ , ซึ่งถ้า  $N = 3$ , จะได้ขนาดช่วงสุดท้ายเป็น 1.0623 และถ้า  $N = 4$ , จะได้ขนาดช่วงสุดท้ายเป็น 0.6565 เพราะฉะนั้นหากต้องการค่าความแม่นยำกว่า 1 จะทำ 4 ขั้นตอน. รูป 2.15 แสดงภาพประกอบของ.

- ขั้น 1,  $a_0 = 8.5$  และ  $b_0 = 13$ , ดังนั้น  $a_1 = a_0 + (b_0 - a_0) \cdot 0.382 = 10.219$  และ  $b_1 = b_0 - (b_0 - a_0) \cdot 0.382 = 11.281$ . เมื่อคำนวณหาค่า  $g_a = g(10.219) = -0.59$  และ  $g_b = g(11.281) = -0.078$ .
- ขั้น 2,  $g_a^{(1)} < g_b^{(1)}$  ดังนั้นยับเข้าฝั่ง  $b_0$  โดยให้  $b_0^{(2)} = b_1^{(1)} = 11.281$  และ  $b_1^{(2)} = a_1^{(1)} = 10.219$ . เราต้องคำนวณค่า  $a_1^{(2)}$  ใหม่. นั่นคือ  $a_1^{(2)} = a_0^{(2)} + (b_0^{(2)} - a_0^{(2)}) \cdot \rho = 9.562$ . และคำนวณค่า  $g_a = g(9.562) = -0.696$ , ส่วนค่า  $g_b = -0.078 (= g_a^{(1)})$  ไม่ต้องคำนวณใหม่.

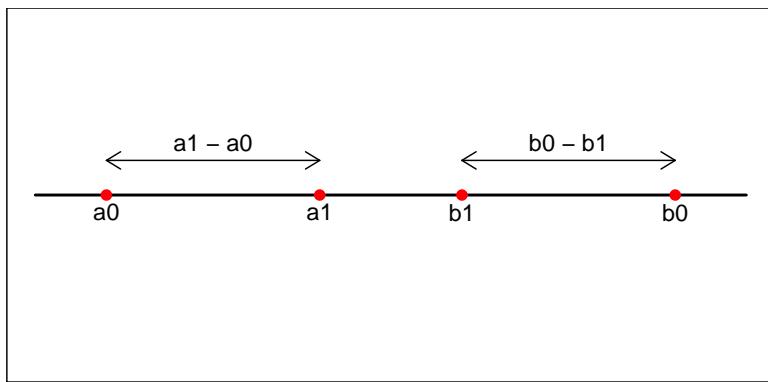
<sup>1</sup> ตัวยก  $(n)$  ระบุว่าเป็นค่าที่ได้ในขั้นตอนที่  $n$



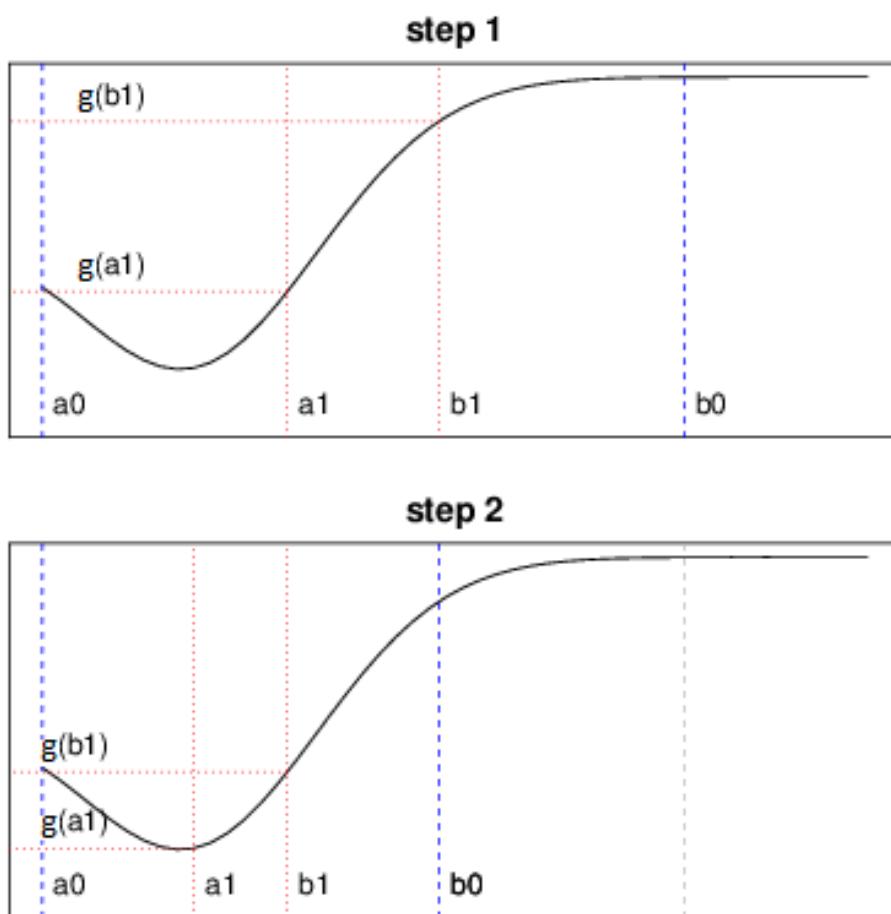
รูปที่ 2.10: ตัวอย่างแสดงฟังชันฐานนิยมเดียว

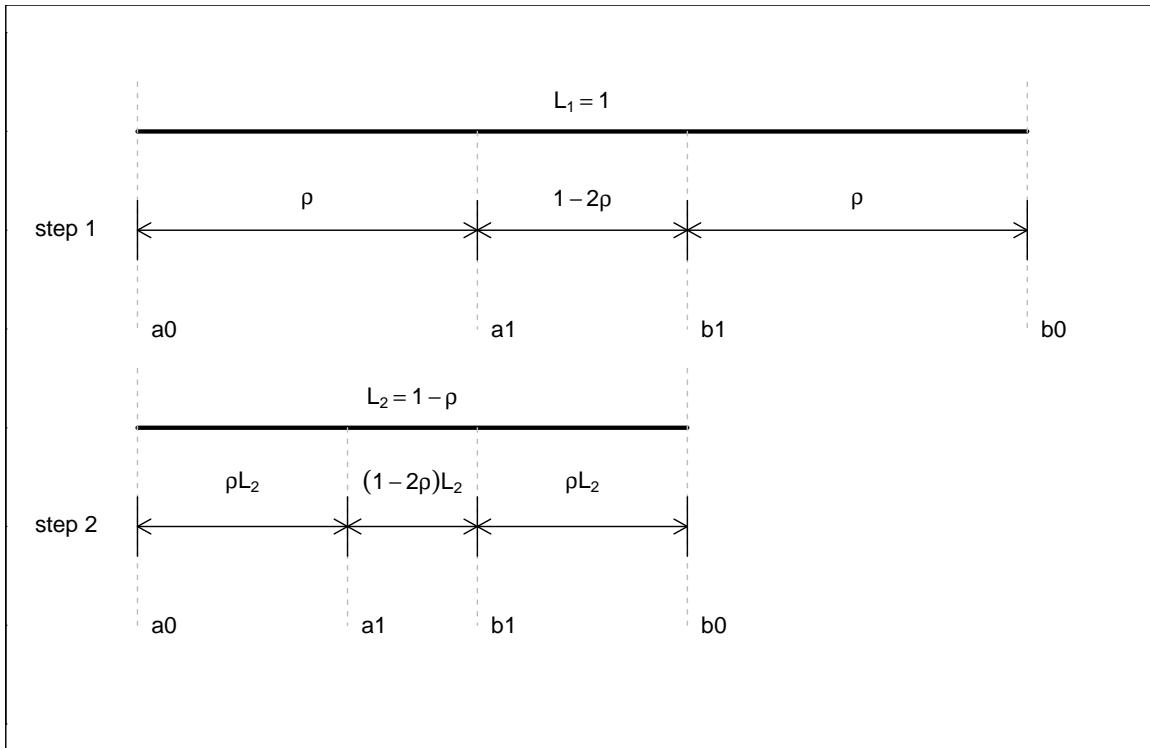


รูปที่ 2.11: ตัวอย่างแสดงแนวคิดของการขยับช่วงเข้าโดยดูจากค่าฟังชันระหว่างช่วงสองค่า. ภาพซ้ายบน  $g_a = g_1(a_1) < g_b = g_1(b_1)$  สอนัยว่าค่าทำน้อยที่สุดไม่อยู่ในช่วงย่อย  $[b_1, b_0]$  เพราะว่า ค่า  $g_1(b) \geq g_b$ , สำหรับทุกๆค่าของ  $b \in [b_1, b_0]$ . (เนื่องจาก  $g_1$  เป็นฐานนิยม, มีหลุมเดียว.) ภาพขวาบน  $g_a = g_2(a_1) < g_b = g_2(b_1)$  สอนัยว่าค่าทำน้อยที่สุดไม่อยู่ในช่วงย่อย  $[b_1, b_0]$ . หากเปรียบเทียบ ภาพซ้ายบนกับภาพขวาบน จะเห็นว่า เมื่อ  $g_a < g_b$  ค่าทำน้อยที่สุดอาจจะอยู่ช่วง  $[a_0, a_1]$  (กรณี  $g_1$ ) หรืออาจจะอยู่ช่วง  $[a_1, b_1]$  (กรณี  $g_2$ ) ก็ได้. แต่ค่าทำน้อยที่สุดไม่อยู่ในช่วงย่อย  $[b_1, b_0]$  แน่ๆ. ภาพซ้ายล่าง และ ภาพขวาล่าง แสดงสถานะการณ์ตัวอย่างเมื่อ  $g_a > g_b$  ว่าอาจเป็นกรณี  $g_3$  หรือ กรณี  $g_4$  ก็ได้ แต่ที่รู้แน่ๆคือ ไม่ว่าจะเป็นกรณี  $g_3$  หรือกรณี  $g_4$  ค่าทำน้อยที่สุดก็ไม่อยู่ในช่วงย่อย  $[a_0, a_1]$ .



รูปที่ 2.12: การแบ่งส่วนจากช่วงที่สนใจ

รูปที่ 2.13: การขยับช่วงโดยจัดอัตราส่วน  $\rho$  ให้พอดีช่วงจะช่วยประหยัดการคำนวณค่าฟังชันได้. ตัวอย่าง เช่น ค่า  $g_b$  ในขั้นตอนที่ 2 จะเท่ากับค่า  $g_a$  จากขั้นตอนที่ 1 จะช่วยประหยัดไม่ต้องทำการคำนวณ  $g(b_1)$  อีก



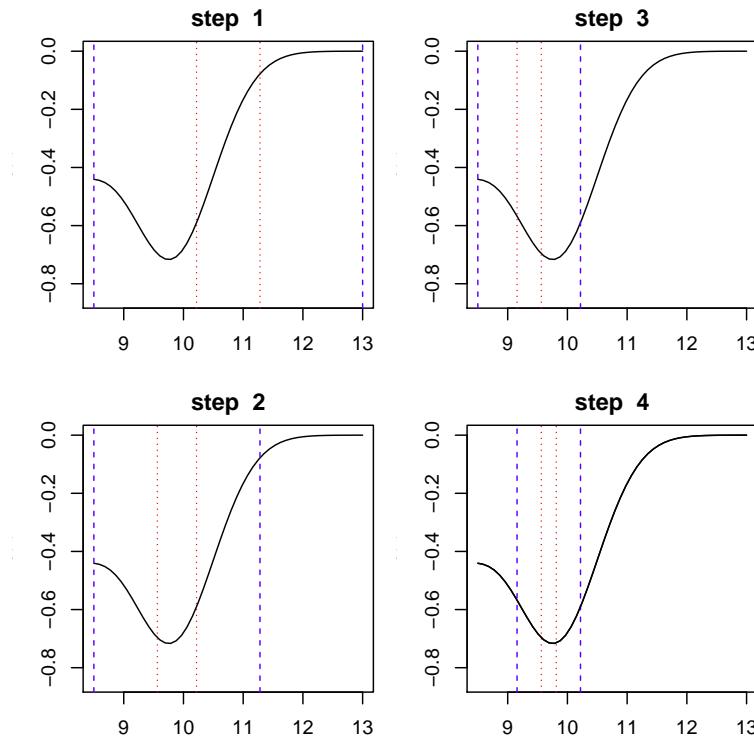
รูปที่ 2.14: สัดส่วนต่างๆ ในเทอมของ  $\rho$ . สังเกตว่า หากต้องการจัดการคำนวนให้มีประสิทธิภาพ กรณี  $a_0^{(2)} = a_0^{(1)}, a_1^{(2)}$  เป็นจุดใหม่,  $b_1^{(2)} = a_1^{(1)}$  และ  $b_0^{(2)} = b_1^{(1)}$ . หากจัดสัดส่วนแบบนี้แล้ว ช่วยให้แต่ละขั้นตอนของการคำนวนมีจุดที่ต้องคำนวนใหม่ เพียงจุดเดียว. ค่า  $\rho$  สามารถพิจารณาคำนวนได้จากความสัมพันธ์  $\rho(b_0^{(2)} - a_0^{(2)}) = b_0^{(2)} - b_1^{(2)} \Rightarrow \rho(b_1^{(1)} - a_0^{(1)}) = b_1^{(1)} - a_1^{(1)}$ .

- ขั้น 3,  $g_a^{(2)} < g_b^{(2)}$  ดังนั้นขยับเข้าฝั่ง  $b_0$  อีก ทำนองเดียวกัน จะได้ว่า  $a_0 = 8.5, a_1 = 9.157, b_1 = 9.562, b_0 = 10.219, g_a = -0.568$  และ  $g_b = -0.696$ .
- ขั้น 4,  $g_a^{(3)} > g_b^{(3)}$  ดังนั้นขยับเข้าฝั่ง  $a_0$  บ้าง เราจะได้  $a_0 = 9.157, a_1 = 9.562, b_1 = 9.813, b_0 = 10.219$  และ  $g_a = -0.696$  กับ  $g_b = -0.715$ .
- เนื่องจาก  $g_a^{(4)} > g_b^{(4)}$  (ขยับเข้าฝั่ง  $a_0$  ได้) หลังจากจบขั้นตอนที่ 4 ก็จะได้คำตอบว่า  $x^* \in [9.562, 10.219]$ .

□

## 2.4 วิธีลงเกรเดียนต์

วิธีการค้นหาแบ่งช่วงท่องคำ แม้จะทำงานได้ดี แต่เมื่อข้อจำกัดที่สำคัญคือ วิธีการค้นหาแบ่งช่วงท่องคำเป็นจะทำงานแบบการค้นหาตามเส้น (Line Search). นั่นคือ วิธีการค้นหาแบ่งช่วงท่องคำสามารถทำงานได้เฉพาะกับปัญหาที่ค่าตัวแปรตัดสินใจเป็นสเกลาร์เท่านั้น ไม่สามารถทำงานได้กับตัวแปรตัดสินใจที่มีหลายๆ มิติ  $\mathbf{x} \in \mathbb{R}^D, D > 1$ .



รูปที่ 2.15: ตัวอย่างการทำค้นหาแบบช่วงทองคำ 4 ขั้นตอน. ภาพซ้ายบนแสดงขั้น 1 เส้นแนวตั้งแสดงจุด  $a_0, a_1, b_1$ , และ  $b_0$ . ภาพซ้ายล่างแสดงขั้น 2 ช่วงขยับเข้าจากด้านขวา. ภาพขวาบนแสดงขั้น 3 ช่วงขยับเข้าจากด้านขวา. ภาพขวาล่างแสดงขั้น 4 ช่วงขยับเข้าจากด้านซ้าย. แต่ละขั้นตอนจะทำให้ช่วงแคบลงเป็น  $\approx 0.6$  เท่าของขนาดช่วงก่อนหน้า.

วิธีหนึ่งในการแก้ปัญหาการหาค่าน้อยที่สุดที่นิยม เพราะทำงานได้ดี สามารถใช้ได้กับปัญหาที่มีตัวแปรตัดสินใจที่มีหลายมิติ และเขียนโปรแกรมได้่ายมากรๆ ก็คือ วิธีลงเกรเดียนต์.

วิธีลงเกรเดียนต์ใช้กับการหาค่าดีที่สุดแบบไม่มีเงื่อนไข (Unconstrained Optimization) และอาศัยแนวคิดเดียวกับเงื่อนไขจำเป็นอันดับแรกสำหรับกรณีภายนอก. นั่นคือ วิธีลงเกรเดียนต์จะใช้ค่าเกรเดียนต์ของฟังชันเป้าหมายต่อตัวแปรตัดสินใจเป็นเครื่องชี้ทาง ในการค้นหาค่าทำน้อยที่สุด โดยพยายามจะค้นหาค่าทำน้อยที่สุด ในทิศทางที่น่าจะเจอเจอที่ค่าเกรเดียนต์เป็นศูนย์.

กล่าวง่ายๆ ทิศทางของเกรเดียนต์(ของฟังชันเป้าหมายต่อตัวแปรตัดสินใจ) ณ จุดเริ่มต้นของค่าใดก็ตาม ของตัวแปรตัดสินใจ ก็คือทิศทางที่ หากขยับ(ค่าตัวแปรตัดสินใจ)ออกจากการจุดค่า <sup>2</sup> นั้นเล็กน้อยตามทิศของเกรเดียนต์แล้ว ค่าฟังชันเป้าหมายจะเพิ่มขึ้น. ในทางกลับกัน หากขยับออกจากจุดเดิมเล็กน้อยไปตามทิศตรงข้ามกับเกรเดียนต์แล้ว ฟังชันเป้าหมาย ณ จุดที่ขยับไปจะมีค่าลดลง. แล้วถ้าขยับจุดที่พิจารณาไปเรื่อยๆ ในลักษณะนี้ ในที่สุด ก็จะสามารถขยับลงไปจนถึงจุดของค่าทำน้อยที่สุดได. ซึ่งก็คือ แนวคิดของวิธีลงเกรเดียนต์ (Gradient Descent Method) ซึ่งเขียนเป็น สมการปรับค่าของตัวแปรตัดสินใจคือ

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla g(\mathbf{x}^{(k)}) \quad (2.11)$$

<sup>2</sup> จุดค่าที่กล่าวถึงในที่นี้ เป็นจุดค่าในปริภูมิหลายมิติ. หลายคนอาจสับสนระหว่างค่าตัวเลขกับจุดค่าในปริภูมิหลายมิติ. ค่าตัวเลขหนึ่งค่า เป็นสเกลาร์ เช่น  $x = 16$ . นั่นคือ ค่าหนึ่งค่า. แต่ค่าจุดหนึ่งจุด สำหรับจุดในปริภูมิหลายมิติ เป็นเวกเตอร์ เช่น  $\mathbf{x} = [255, 150, 0]^T$  แทนสีแดงอมแสด. ค่า  $[255, 150, 0]^T$  นี้ คือค่าแทนหนึ่งจุดในปริภูมิ 3 มิติของระบบสีแดงเขียวน้ำเงิน เป็นต้น.

โดย  $\mathbf{x}^{(k)}$  แทนค่าของตัวแปรตัดสินใจในการคำนวนครั้งที่  $k$ , ค่า  $\alpha_k$  มีค่า  $> 0$  เรียกว่า ขนาดก้าว (Step Size). โดย ถ้า  $\alpha_k$  มีค่าเล็กพอก็จะรับประทานได้ว่า ค่าของฟังชันเป้าหมายของจุดที่ขยับไปใหม่จะมีค่าน้อยกว่าเดิม  $g(\mathbf{x}^{(k+1)}) < g(\mathbf{x}^{(k)})$  หรือ กล่าวอีกอย่างคือ หากค่าขนาดก้าวเล็กพอก็ วิธีลิงเกรเดียนต์รับประทานที่จะลู่เข้าหาค่าทำน้อยที่สุดท้องถิ่น.

ตัวอย่าง. จงหาค่าทำน้อยที่สุดของฟังชันเป้าหมาย  $g(x) = -e^{-(x-5)^2}$  ด้วยวิธีลิงเกรเดียนต์ และให้เริ่มต้นจาก  $x^{(0)} = 6.5$  และใช้ค่าขนาดก้าวเป็น 0.5.

- อันดับแรก คือการหาเกรเดียนต์ของฟังชันเป้าหมาย ซึ่งคือ

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

- คำนวน (สมการ 2.11) ครั้งที่  $k = 1$  ได้

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

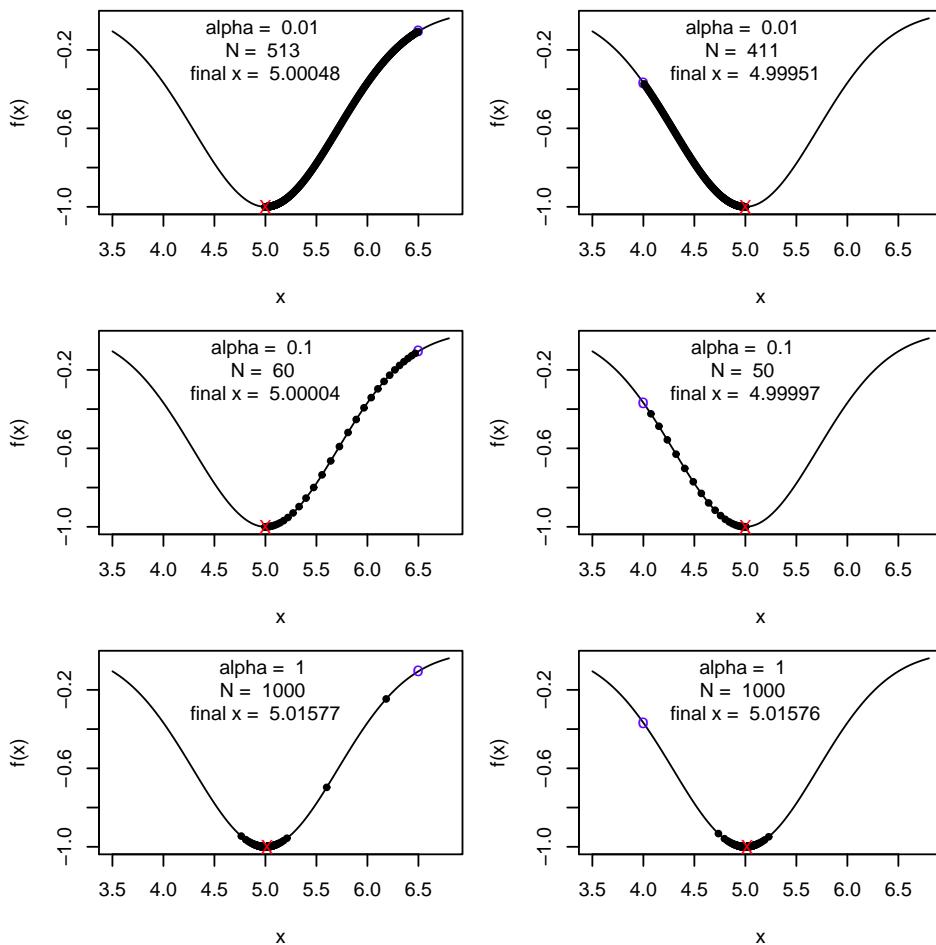
- คำนวนครั้งที่ 2 ถึง 8 จะได้

$$\begin{aligned} x^{(2)} &= x^{(1)} - (0.5) \cdot \nabla g(x^{(1)}) \\ &= 6.3419 - (0.5) \cdot \nabla g(6.3419) = 6.1202 \\ x^{(3)} &= 6.1202 - (0.5) \cdot \nabla g(6.1202) = 5.8009 \\ x^{(4)} &= 5.8009 - (0.5) \cdot \nabla g(5.8009) = 5.3792 \\ x^{(5)} &= 5.0508 \\ x^{(6)} &= 5.0001 \\ x^{(7)} &= 5.0000 \\ x^{(8)} &= 5.0000 \end{aligned}$$

- และผลลัพธ์  $x = 5$  ซึ่งคือค่าตอบของฟังชันเป้าหมายนี้. สังเกตุค่าเกรเดียนต์  $\nabla g(5) = 0$  ซึ่งเป็นไปตามเงื่อนไข จำเป็นอันดับแรกสำหรับกรณีภายนอก.

□

การเลือกใช้ค่าขนาดก้าว อาจเลือกใช้ค่าขนาดก้าวเป็นค่าคงที่ทุกๆ ครั้งของการคำนวนก็ได้ แต่หากเลือกค่าขนาดก้าวที่เล็กเกินไป ก็อาจทำให้ต้องทำการคำนวนหลายรอบกว่าที่จะได้ความแม่นยำตามที่ต้องการ. แต่หากเลือกค่าขนาดก้าวที่ใหญ่เกินไป ก็อาจทำให้ไม่ได้ความแม่นยำตามที่ต้องการ ดังแสดงในสองภาพล่างของรูป 2.16 ที่แม้จะทำการคำนวนหลายรอบมากกว่า แต่ก็ยังได้ค่าความแม่นยำแย่กว่ามาก หรือบางครั้งการใช้ขนาดก้าวที่ใหญ่เกินไปมาก นอกจากจะไม่ได้ค่าความแม่นยำที่ต้องการแล้ว ยังอาจนำไปสู่การลู่ออก (Divergence) ด้วย (ดูแบบฝึกหัดท้ายบทข้อ 3).



รูปที่ 2.16: ตัวอย่างผลจากวิธีลงเกรเดียนต์ด้วยขนาดก้าวค่าต่างๆ. ทุกภาพทางซ้ายเริ่มที่  $x^{(0)} = 6.5$ . ทุกภาพทางขวาเริ่มที่  $x^{(0)} = 4$ . จุดเริ่มต้นแสดงด้วยวงกลมสีน้ำเงิน. ค่า  $x$  แต่ละรอบการคำนวณแสดงด้วยจุดสีดำ. ค่าสุดท้ายที่ได้แสดงด้วยกาบath สีแดง.

## 2.5 วิธีลงขั้นที่สุด

จากวิธีลงเกรเดียนต์ แทนที่จะเลือกค่าขนาดก้าวคงที่ค่าใดค่าหนึ่ง วิธีลงขั้นที่สุด (Steepest Descent Method) พัฒนาวิธีลงเกรเดียนต์ โดยการเลือกใช้ค่า  $\alpha_k$  ที่จะทำให้พังขั้นเป้าหมายลดลงมากที่สุดในการขยับแต่ละครั้ง. นั่นคือ

$$\alpha_k = \arg \min_{\alpha \geq 0} g(\mathbf{x}^{(k)} - \alpha \nabla g(\mathbf{x}^{(k)})). \quad (2.12)$$

จากตัวอย่างเดิมของวิธีลงเกรเดียนต์. หากำทำน้อยที่สุดของพังขั้นเป้าหมาย  $g(x) = -e^{-(x-5)^2}$  ด้วยวิธีลงขั้นที่สุด และให้เริ่มจาก  $x^{(0)} = 6.5$

- เกรเดียนต์  $\nabla g(x) = -e^{-(x-5)^2} \cdot (-2x + 10)$ .

- วิธีลงชั้นที่สุดจะใช้การค้นหาแบ่งช่วงทองคำเพื่อหา  $\alpha \in [0, 1]$  ที่ทำให้  $h(\alpha)$  น้อยที่สุด,

$$\begin{aligned} h(\alpha) &= g(x - \alpha \nabla g(x)) \\ &= -\exp \left\{ -(x - \alpha \cdot \left( -e^{-(x-5)^2} \cdot (-2x + 10) \right) - 5)^2 \right\}. \end{aligned}$$

- การคำนวณครั้ง 1,

$$\alpha_1 = \arg \min_{\alpha \in [0, 1]} -\exp \left\{ -(6.5 - \alpha \cdot \left( -e^{-(6.5-5)^2} \cdot (-2(6.5) + 10) \right) - 5)^2 \right\}$$

โดยผลจากการค้นหาแบ่งช่วงทองคำได้  $\alpha_1 = 1$

- ตั้งนั้น  $x^{(1)} = 6.5 - (1)\nabla g(6.5) = 6.1838$ .

- การคำนวณครั้ง 2,

$$\alpha_2 = \arg \min_{\alpha \in [0, 1]} -\exp \left\{ -(6.1838 - \alpha \cdot \left( -e^{-(6.1838-5)^2} \cdot (-2(6.1838) + 10) \right) - 5)^2 \right\}$$

โดยผลจากการค้นหาแบ่งช่วงทองคำได้  $\alpha_2 = 1$

- ตั้งนั้น  $x^{(2)} = 6.1838 - (1)\nabla g(6.1838) = 5.6008$ .

- การคำนวณครั้ง 3, ผลจากการค้นหาแบ่งช่วงทองคำได้  $\alpha_3 = 0.7173$

- ตั้งนั้น  $x^{(3)} = 5.6008 - (0.7173)\nabla g(5.6008) = 4.9997$ .

- การคำนวณครั้ง 4, ผลจากการค้นหาแบ่งช่วงทองคำได้  $\alpha_4 = 0.5000$

- ตั้งนั้น  $x^{(4)} = 4.9997 - (0.5)\nabla g(4.9997) = 5.000$ .

- ซึ่งที่  $x = 5$ , ค่า  $\nabla g(5) = 0$  และผลลัพธ์เข้าค่า  $x = 5$  ซึ่งคือคำตอบ.

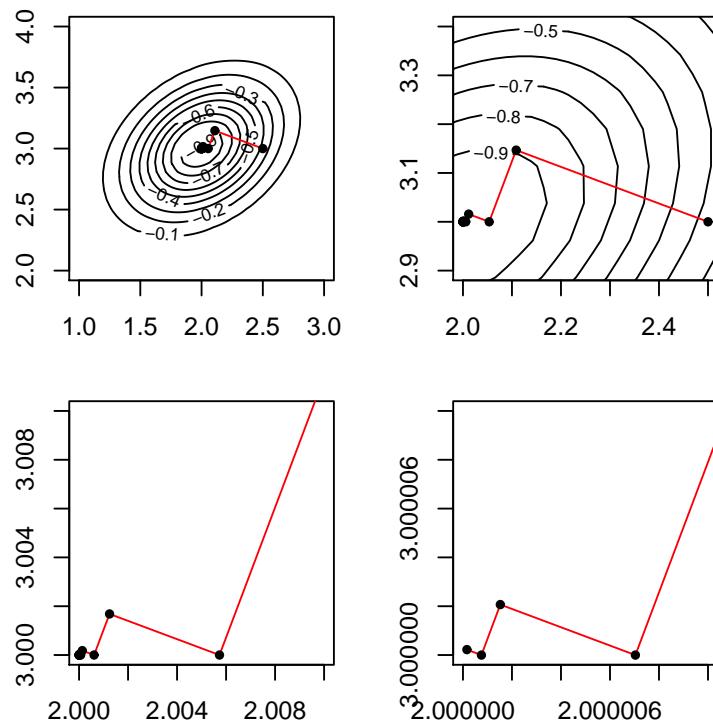
□

ธรรมชาติของวิธีลงชั้นที่สุด. รูป 2.17 แสดงผลการขยับค่าตัวแปรตัดสินใจ ในการคำนวณแต่ละครั้งของ วิธีลงชั้นที่สุด. ธรรมชาติของวิธีลงชั้นที่สุดจะให้ผลในลักษณะที่การขยับค่าตัวแปรตัดสินใจแต่ละครั้งจะมี ทิศทางตั้งฉากกับทิศทางการขยับเดิม (ดูแบบฝึกหัดท้ายบทที่ 4).

## 2.6 แบบฝึกหัด

- จะใช้การค้นหาแบ่งช่วงทองคำ เพื่อหาค่าทำน้อยที่สุดของฟังชันต่อไปนี้ โดยให้ผลลัพธ์แม่นยำขนาด ผิดพลาดไม่เกิน 0.0001

- ก.  $g(x) = (x - 5)^2$  โดยเริ่มจากช่วง  $x \in [0, 10]$ .



รูปที่ 2.17: ผลจากแต่ละการคำนวณของวิธีลิงชันที่สุด

- ๔.  $g(x) = -0.4 \exp\{-(x-8)^2\} - 0.7 \exp\{-(x-9.8)^2\}$  โดยเริ่มจากช่วง  $x \in [8.5, 13]$ .
- ๕.  $g(x) = 2.813x + 1.843x^2 - 2.187x^3 + 0.5312x^4$  โดยเริ่มจากช่วง  $x \in [-2, 2]$ .

โค้ดข้างล่างนี้สำหรับฟังชันทำการค้นหาแบ่งช่วงทองคำ โดยอาร์กูเมนต์  $f$  แทนฟังชันจุดประสงค์ อาร์กูเมนต์  $a0$  และ  $b0$  แทนขอบซ้ายและขอบขวาของช่วงที่ต้องการค้นหา อาร์กูเมนต์  $tol$  แทนค่าความแม่นยำที่ยอมรับได้.

รายการ 2.1: การค้นหาแบ่งช่วงทองคำ (Golden Section Search)

---

```

1 goldensearch <- function(f, a0, b0, tol=0.1,
2   rho=0.382, log=FALSE){
3
4   diff <- b0 - a0
5   a1 <- a0 + rho*diff
6   b1 <- b0 - rho*diff
7
8   fa0 <- f(a0)
9   fb0 <- f(b0)
10  fa1 <- f(a1)
11  fb1 <- f(b1)
12

```

```

13   logs <- matrix(c(a0,fa0,b0,fb0),4,1)
14
15   while (diff > tol) {
16     if( fa1 < fb1 ) {
17       b0 <- b1
18       fb0 <- fb1
19       b1 <- a1
20       fb1 <- fa1
21
22       diff <- b0 - a0
23       a1 <- a0 + rho*diff
24       fa1 <- f(a1)
25
26     } else { ## fa1 >= fb1
27       a0 <- a1
28       fa0 <- fa1
29       a1 <- b1
30       fa1 <- fb1
31
32       diff <- b0 - a0
33       b1 <- b0 - rho*diff
34       fb1 <- f(b1)
35
36     } ## end if
37
38     logs <- cbind(logs, matrix(c(a0,fa0,b0,fb0),4,1))
39   }## end while
40
41   rownames(logs) <- c("a0", "f(a0)", "b0", "f(b0)")
42   if(log) { return(logs) }
43   else { return((a0+b0)/2) }
44 }## goldensearch

```

---

ตัวอย่างการใช้งาน เช่น ถ้าฟังชันเป้าหมายคือ  $g(x) = \cos(x)$  และต้องการหาค่า  $x$  ในช่วง  $x \in [0, 2\pi]$  โดยให้ค่าผิดพลาดน้อยกว่า 0.001 ก็สามารถทำได้โดย

```

g <- function(x){ cos(x) }
results <- goldensearch(g, 0, 2*pi, tol=0.001)

```

ซึ่งจะได้ผลคือ 3.141534 ซึ่งตรงกับที่ความรู้เดิมคือโคչายน์ฟังชันมีค่าน้อยที่สุดที่  $\pi \approx 3.1416$  และได้ค่าผิดพลาดน้อยกว่า 0.001.

## 2. จากการโค้ดวิธีลงเกรเดียนต์ข้างล่าง จงตอบคำถามข้อ ก. ถึง ง.

รายการ 2.2: วิธีลงเกรเดียนต์ (Gradient Descent Method). อาร์กูเมนต์  $x$  แทนค่าเริ่มต้นของตัวแปรตัดสินใจ อาร์กูเมนต์  $df$  แทนเกรเดียนต์พังชั่น อาร์กูเมนต์  $alpha$  แทนค่าขนาดก้าว อาร์กูเมนต์  $tol$  แทนค่าความแม่นยำที่ยอมรับได้ อาร์กูเมนต์  $MaxN$  แทนจำนวนรอบคำนวณสูงสุด. โปรแกรมจะหยุดเมื่อได้ค่าความแม่นยำที่ยอมรับได้ หรือครบจำนวนรอบคำนวณสูงสุด ขึ้นกับว่า อะไรถึงก่อน

```

1 gd <- function(x, df, alpha=0.1, tol=1e-5,
2   MaxN=200, log=TRUE){
3
4   D <- length(x)
5   logs <- matrix(0, D, MaxN)
6
7   for(i in 1:MaxN){
8
9     new.x <- x - alpha * df(x)
10
11    logs[,i] <- new.x
12
13    if(is.infinite(new.x)) break;
14    if(is.nan(new.x)) break;
15
16    if( abs(new.x - x) < tol ){
17      break;
18    }
19    x <- new.x
20
21  }
22
23  if(log){ return(logs[,1:i]) }
24  return(new.x)
25 }
```

ตัวอย่างเช่น หากต้องการหาค่าทำน้อยที่สุดของพังชั่น  $g(x) = (x - 8)^2$ , (หนึ่ง) หาอนุพันธ์ของมา เช่น  $\nabla g(x) = 2(x - 8)$ , (สอง) เลือกจุดเริ่มต้น และค่าขนาดก้าว  $\alpha$  เช่น ให้  $x^{(0)} = 0$  และ  $\alpha = 0.3$  และรัน

```
df <- function(x){ 2 * (x-8) }
```

```
results <- gd(x=0, df, alpha=0.3)
```

ผลลัพธ์ใน `results` จะเก็บค่า  $x$  ในแต่ละการคำนวณ โดยค่าสุดท้ายคือค่าตอบที่ได้ เช่น

&gt; results

```
[1] 4.800000 6.720000 7.488000 7.795200 7.918080 7.967232 7.986893 7.994757
[9] 7.997903 7.999161 7.999664 7.999866 7.999946 7.999979 7.999991 7.999997
```

คำตอบที่ได้คือ  $7.999997$  ซึ่งผิดจากคำตอบที่ถูกต้อง  $x^* = 8$  น้อยกว่า  $0.00001$  (หรือ  $1 \times 10^{-5}$  ตามที่ระบุไว้ด้วย tol)

จะใช้วิธีลงเกรเดียนต์ ตอบคำถามข้างล่าง

- ก. หากำทำน้อยที่สุดของฟังชัน  $f(x) = -\exp\{-(x-7)^2\}$  เริ่มต้นที่ 5.5 และใช้  $\alpha = 0.3$ .
- ข. หากำทำน้อยที่สุดของฟังชัน  $h(\mathbf{x} = [x_1, x_2]^T) = (2x_1 - 9)^2 + (x_2 - 8)^2$  โดยให้เลือกจุดเริ่มต้น และค่า  $\alpha$  เอง.
- ค. ทำข้อ ก. ใหม่โดยเปลี่ยนค่าเริ่มต้น เป็น 2 และอธิบายว่าเกิดอะไรขึ้น
- ง. ทำข้อ ก. ใหม่โดยใช้ค่าเริ่มต้นที่ 5.5 และหากำ  $\alpha$  ที่ทำให้เข้าได้เร็วที่สุด.

3. จะใช้วิธีลงเกรเดียนต์เพื่อหากำทำน้อยที่สุดของฟังชันที่กำหนดดังอาร์โค้ดข้างล่าง

```
w <- c(1.827579e-14, 2.812698e+00, 1.843386e+00,
      -2.187302e+00, 5.312169e-01)
```

```
f <- function(x){
  w[1] + w[2]*x + w[3]*x^2 + w[4]*x^3 + w[5]*x^4
}
```

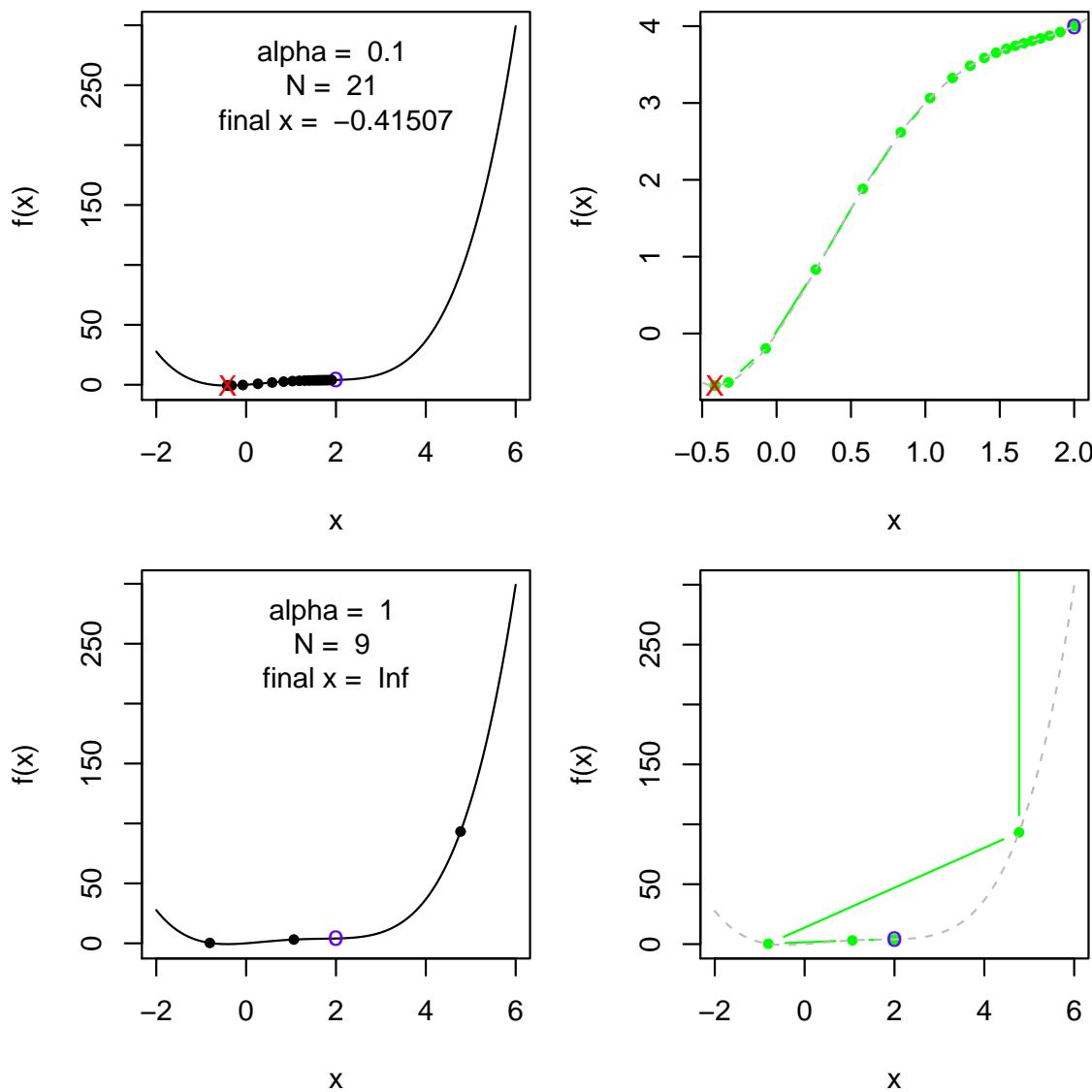
ใช้ค่าเริ่มต้นที่  $x^{(0)} = 2$  แต่ทดลองค่า  $\alpha$  เป็น 0.1 กับ 1 และได้ผลดังแสดงในรูป 2.18. ตอนที่ใช้ค่าขนาดกำก้าวเป็น 0.1 วิธีลงเกรเดียนต์ทำการคำนวณไป 21 ครั้งแล้วได้ผลลัพธ์เป็น  $-0.41507$ . ตอนที่ใช้ค่าขนาดกำก้าวเป็น 1 วิธีลงเกรเดียนต์ทำการคำนวณแค่ 9 ครั้งแล้วได้ผลลัพธ์เป็น  $\infty$ , ซึ่งไม่ถูกต้อง. พอนำค่าตัวแปรตัดสินใจที่ขยับแต่ละการคำนวณมาวดกราฟ ดังแสดงในภาพขวाल่าง (รูป 2.18) อธิบายว่าเกิดอะไรขึ้น ทำไม่จึงมีค่าของตัวแปรตัดสินใจที่อยู่ใกล้คำตอบแล้วจึงวิงออกห่างไป.

4. จากอาร์โค้ดข้างล่างทำวิธีลงชันที่สุด จงหากำทำน้อยที่สุดของฟังชัน ข้อ ก. ถึง ง.

รายการ 2.3: วิธีลงชันที่สุด (Steepest Descent Method). อาร์กูเมนต์ grad แทนเกรเดียนต์ฟังชัน อาร์กูเมนต์ f แทนฟังชันจุดประสงค์ อาร์กูเมนต์ x0 แทนค่าเริ่มต้นของตัวแปรตัดสินใจ อาร์กูเมนต์ tol แทนค่าความแม่นยำที่ยอมรับได้ อาร์กูเมนต์ MaxN แทนจำนวนรอบคำนวณสูงสุด. โปรแกรมจะหยุดคำนวณเมื่อได้ค่าความแม่นยำที่ยอมรับได้ หรือถึงจำนวนรอบคำนวณสูงสุด ขึ้นกับว่าอะไรถึงก่อน

---

1 `steepestdescent <- function(grad, f, x0, tol=1e-5,`



รูปที่ 2.18: แบบฝึกหัดข้อ 13. สองภาพบนแสดงผลจากการคำนวณแต่ละครั้งของวิธีลงเกรเดียนต์ที่ใช้  $\alpha = 0.1$ . สองภาพล่าง แสดงผลจากการใช้  $\alpha = 1$ . โดย ภาพทางซ้ายแสดงภาพรวม ค่าเริ่มต้น (แทนด้วยวงกลม) ค่าสุดท้าย (แทนด้วยกาบบาท) ค่าที่ได้จากการคำนวณแต่ละครั้ง (แทนด้วยจุดสีดำ) และผลสรุปซึ่งเขียนไว้บริเวณกลางภาพ และภาพขวาขยายให้เห็นเส้นทางการขยับของค่าตัวแปรตัดสินใจจากการคำนวณแต่ละครั้ง. ผลสรุปที่กากบาท ของสองภาพซ้าย คือ ค่าขนาดก้าว จำนวนรอบการคำนวณที่ทำ ค่าตัวแปรตัดสินใจค่าสุดท้ายที่ได้ ตามลำดับจากบนลงล่าง

```

2   MaxN=500, log=FALSE){
3
4   D <- length(x0)
5
6   logs <- matrix(c(0,0,x0),2+D,1)
7
8   diff <- tol*2
9   for (i in 1:MaxN) {
10
11     g <- function(a){ f(x0 - a*grad(x0)) }
12     alpha <- goldensearch(g, a0=0, b0=1,
13       tol=tol, log=FALSE)
14
15     gradF <- grad(x0)
16     x <- x0 - alpha*gradF
17
18     diff <- sqrt( mean((gradF)^2) )
19     x0 <- x
20     logs <- cbind(logs, c(i,alpha,x0))
21
22     if (diff < tol) break
23 }## for
24
25 if(log){ return(logs) }
26 else { return(x) }
27 }## end steepestdescent

```

---

ตัวอย่าง เช่นถ้าหากต้องการหาค่าทำน้อยที่สุดของ  $f(\mathbf{x} = [x_1, x_2]^T) = (x_1 - 3)^2 + (x_2 - 5)^2$  สามารถทำได้โดย (หนึ่ง) หาอนุพันธ์ของฟังชันเป้าหมาย(สอง) กำหนดจุดเริ่มต้น เช่น อาจเลือกจุด  $[4, 5]^T$  เป็นจุดเริ่มต้น และรันฟังชัน steepestdescent ดังข้างล่าง

```

f <- function(x){ (x[1] - 3)^2 + (x[2] - 5)^2 }
df <- function(x){ matrix(c(2*(x[1] - 3), 2*(x[2] - 5)), 2, 1) }

results <- steepestdescent(df, f, x0=matrix(c(4,5), 2, 1), log=TRUE)

```

โค้ดข้างต้น f และ df กำหนดฟังชันเป้าหมายและอนุพันธ์ตามลำดับ.

ผล results ที่ได้คือ

```
[,1]      [,2]      [,3]
```

```
[1,] 0 1.000000 2.000000
[2,] 0 0.499999 0.499999
[3,] 4 3.000002 3.000000
[4,] 5 5.000000 5.000000
```

โดยแต่ละคอลัมน์คือ การคำนวณแต่ละครั้ง แผลแรกบอกว่าเป็นการคำนวณครั้งที่เท่าไร (ค่าเริ่มต้น นับเป็นการคำนวณครั้งที่ 0), แผลที่สองบอกค่า  $\alpha$  ที่ใช้, แผลต่อๆมาเป็นค่าของตัวแปรตัดสินใจ เช่น จากผลข้างต้น `steepestdescent` ทำการคำนวณ 2 ครั้ง โดย ครั้งที่ 1 ใช้  $\alpha = 0.499999$  และได้ขยับค่าตัวแปรไปเป็น 3.000002 กับ 5.000000 และครั้งที่ 2 ใช้  $\alpha = 0.499999$  และได้ขยับค่าตัวแปรไปเป็น 3 กับ 5 ซึ่งคือคำตอบ.

- ก. จงหาค่าทำน้อยที่สุดของ  $h([x_1, x_2]^T) = -\exp(-(x_1 - 7)^2) + (x_2 - 4)^2$  โดยใช้  $[0, 0]^T$  เป็นค่าเริ่มต้น.
- ข. จงหาค่าทำน้อยที่สุดของ  $g(x) = x^2 + 4x + 15$  โดยใช้ 15 เป็นค่าเริ่มต้น.
- ค. จงหาค่าทำน้อยที่สุดของ  $f(\mathbf{x} = [x_1, x_2]^T) = -\exp(-\{z_{11}(x_1^2 - 2x_1v_1 + v_1^2) + (z_{12} + z_{21})(x_1x_2 - v_1x_2 - x_1v_2 + v_1v_2) + z_{22}(x_2^2 - 2x_2v_2 + v_2^2)\})$  โดย  $v_1 = 2, v_2 = 3, z_{11} = 4, z_{12} = z_{21} = -1.5, z_{22} = 5$ , และ ใช้ค่าเริ่มต้นเป็น  $[2.5, 3]^T$  เสร็จแล้วว่าดูรูปแบบรูป [2.17](#). [คำใบ้: ลอง `help(contour)`]
- ง. ทำข้อ ค. อีกครั้งโดยสุ่มค่าเริ่มต้น เสร็จแล้วว่าดูรูปแบบเดียวกับ ค. [คำใบ้: ลอง `help(rnorm)`]

## บทที่ 3

### พื้นฐานสำหรับการเรียนรู้ของเครื่อง

“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

“วิธีการมีเป็นล้านและมากกว่า แต่หลักการมีไม่มาก บุคคลผู้ยึดในหลักการสามารถเลือกวิธีการได้อย่างดี บุคคลผู้ลองแต่ละวิธีการ ละเลยหลักการ ย่อมแน่นอนว่าจะมีปัญหา”

—ราล์ฟ วัลโด อีเมอร์สัน

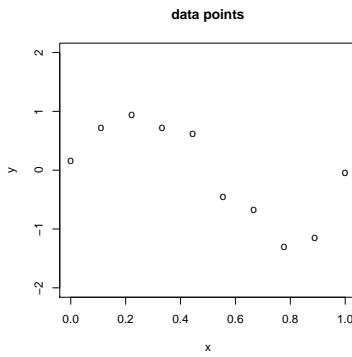
วิธีการเรียนรู้ของเครื่องมีหลากหลายมากมายแตกต่างกันไปตามลักษณะงานที่ต้องการ. หัวข้อ 3.1 อภิปรายตัวอย่างง่ายๆ ของวิธีการเรียนรู้ของเครื่อง. แม้จะเป็นตัวอย่างง่ายๆ แต่ก็สะท้อนแนวคิดและหลักการที่สำคัญของศาสตร์การเรียนรู้ของเครื่อง.

#### 3.1 ตัวอย่างการหาค่าคาดถอยมิติเดียวด้วยฟังชันพหุนาม

ตัวอย่างง่ายๆ ของการประมาณค่าของฟังชันไซ ๕ ( $x$ ) โดยที่ไม่รู้กระบวนการภายในของฟังชันไซ แต่เมื่อตัวอย่างค่าอินพุต ได้แก่  $x_1, x_2, \dots, x_N$  และตัวอย่างเอาท์พุตจากฟังชันไซ ได้แก่  $t_1, t_2, \dots, t_N$ . นั่นคือ เรารู้ว่า  $t_1$  ได้จาก  $\xi(x_1)$ ,  $t_2$  ได้จาก  $\xi(x_2)$ ,  $t_3$  ได้จาก  $\xi(x_3)$ , ...,  $t_N$  ได้จาก  $\xi(x_N)$ . ถึงแม้ว่าจะไม่รู้สมการหรือโครงสร้าง หรือกระบวนการภายในในฟังชันไซจริงๆ แต่เราสามารถประมาณฟังชันไซได้ด้วยการจำลองพฤติกรรมของฟังชันไซจากข้อมูลที่มีเหล่านี้.

รูปที่ 3.1 แสดงตัวอย่างข้อมูล 10 จุดข้อมูล ได้แก่  $(x_1, t_1), (x_2, t_2), \dots, (x_{10}, t_{10})$ . ตัวอย่างนี้ ฟังชันไซถูกสร้างมาจากการ

$$\xi(x) = \sin(2\pi x) + \varepsilon \quad (3.1)$$



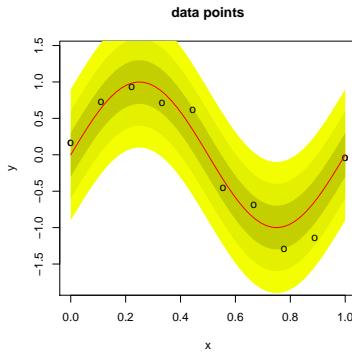
รูปที่ 3.1: จุดข้อมูลตัวอย่าง 10 จุด

โดย  $\epsilon$  เป็นค่าสุ่มที่มีการกระจายแบบเกาส์ที่ค่าเฉลี่ยเป็น 0 และ ค่าเบี่ยงเบนมาตรฐานเป็น 0.3, เอียน ย่อๆ คือ  $\epsilon \sim \mathcal{N}(\mu = 0, \sigma = 0.3)$ . เส้นทิปในรูป 3.2 แสดงกระบวนการภายนอกของฟังชันที่ต้องการ เลียนแบบ.

ในทางปฏิบัติ เราจะไม่รู้กระบวนการภายนอกของฟังชันที่ต้องการเลียนแบบนี้ เพราะหากเรารู้กระบวนการภายนอกนี้ เราถึงสามารถสร้างโมเดลหรือการจำลองสมการจากกระบวนการภายนอกนี้ได้โดยตรง. การศึกษาถึงลักษณะปกติ หรือโครงสร้าง หรือกระบวนการภายนอกนี้ ต้องอาศัยความเข้าใจเหตุผล ปฏิกรรม ปัจจัยที่เกี่ยวข้อง และอาจต้องการการทดลอง ทดสอบ วิเคราะห์ ทบทวนปรับปรุง ซึ่งเป็นกระบวนการทางวิชาการ และหลักการศึกษาวิจัยเฉพาะสำหรับงานแต่ละด้าน แต่ละแขนง แต่ละศาสตร์ ซึ่งต้องอาศัยผู้เชี่ยวชาญเฉพาะด้าน และมักใช้ทรัพยากรและเวลาในการทำมาก. เทคนิคที่อภิปราย ณ ที่นี่แสดงวิธีที่ใช้การประมาณฟังชันที่สนใจ โดยอาศัยเพียงข้อมูลตัวอย่าง ซึ่งเป็นแนวทางหนึ่งที่ให้ผลดีมากในทางปฏิบัติ. หมายเหตุ เทคนิคการประมาณฟังชันแบบนี้ ไม่ได้เป็นแนวทางที่จะแข่งขัน หรือแทนที่การศึกษาเพื่อเข้าใจถึงโครงสร้างหรือกระบวนการภายนอกนี้ ซึ่งเป็นหัวใจของศาสตร์ต่างๆ แต่เป็นเสมือนเครื่องมือที่เสริมเพิ่มเติมขึ้นมา. นอกจากนั้น ทั้งการประมาณฟังชันและการศึกษาถึงความเข้าใจอย่างถ่องแท้ถึงโครงสร้างกระบวนการภายนอกนี้ ยังอาจช่วยซึ่งกันและกัน ช่วยให้ทั้งงานการประมาณฟังชันทำได้มีประสิทธิภาพมากขึ้น และงานการศึกษาโครงสร้างกระบวนการภายนอกทำได้สะดวกเร็วมากยิ่งขึ้น.

ตัวอย่างนี้มีลักษณะคล้ายอย่างที่คล้ายกับข้อมูลจริงๆ คือ จุดข้อมูลจะช่วยบอกลักษณะปกติ (Regularity) ของกระบวนการเบื้องหลัง (เส้นสีแดง) ในขณะเดียวกัน แต่ละจุดข้อมูลจะมีสัญญาณรบกวน (Noise) ปนอยู่. สัญญาณรบกวนนี้ ( $\epsilon$  ในสมการ 3.1) อาจจะมาจากธรรมชาติของกระบวนการที่สังเกตุ ซึ่งสัญญาณรบกวน อาจมีลักษณะเชิงสุ่ม เช่น การสื่อสารสัญญาณกัมมันตรังสี, หรืออาจจะมาจากความหลากหลายของลักษณะ บางอย่างที่ไม่ได้วัด เช่น ความยาวของเม็ดข้าวของสายพันธุ์เดียวกัน อาจแปรผันหลายค่า ที่อาจจะมาจากการปริมาณน้ำ สารอาหาร แสงแดด ที่ต้นข้าวได้รับ ซึ่งเป็นข้อมูลที่ไม่ได้วัด, หรืออาจจะมาจากการธรรมชาติของกระบวนการวัดและสังเกตุเองก็เป็นได้.

เป้าหมายของตัวอย่างนี้ก็คือ สามารถทำนายค่าประมาณเอาท์พุต  $\hat{t}$  ของอินพุต  $\hat{x}$  ได้ แม้อินพุต  $\hat{x}$  จะ เป็นค่าใหม่ที่ไม่เคยเห็นมาก่อน. ซึ่งจะว่าไปแล้ว มันก็คือ การหาลักษณะปกติ  $\sin(2\pi x)$  เส้นทิปสีแดงในรูป 3.2 นั่นเอง. สำหรับเรื่องการทำนายค่าเอาท์พุตของอินพุตค่าใหม่ที่ไม่เคยเห็นมาก่อน เมื่อกล่าวถึงอินพุต



รูปที่ 3.2: จุดข้อมูลตัวอย่าง 10 จุด พร้อม เส้นทิบสีแดงแสดงค่าลักษณะปกติ  $\sin(2\pi x)$  (โครงสร้างภายในที่สร้างข้อมูลออก มาโดยปราศจากสัญญาณรบกวน) และพื้นที่สีเหลืองแสดงบริเวณที่ห่างจากค่าลักษณะปกติ  $\sin(2\pi x)$  เป็นระยะ  $\sigma$ ,  $2\sigma$ , และ  $3\sigma$  ตามลำดับความเข้มของสี

ค่าใหม่ที่ไม่เคยเห็นมาก่อน ทว่าไปนั้นหมายถึงอินพุตอะเรกีไดที่ไม่ซ้ำกับตัวอย่าง. ตัวอย่างของจุดข้อมูล 10 จุดที่แสดงในรูป 3.1 มีค่าอินพุตและเอาต์พุตดังนี้

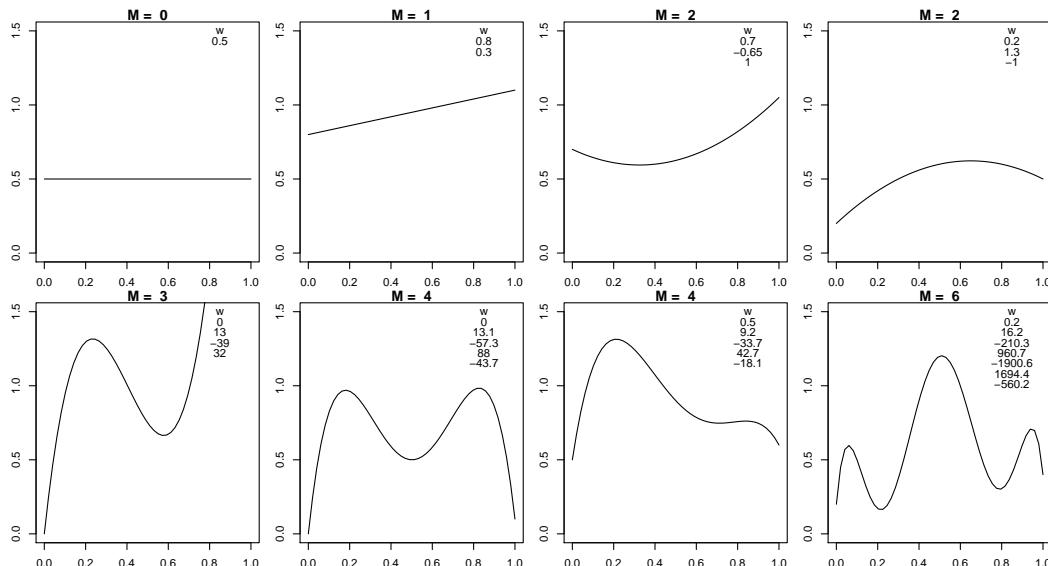
```
> x
[1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444
[6] 0.5555556 0.6666667 0.7777778 0.8888889 1.0000000
> t
[1] 0.16004540 0.72357142 0.93128763 0.71170076
[5] 0.60979515 -0.46044723 -0.68360173 -1.29926264
[9] -1.14706643 -0.04490931
```

อินพุตใหม่ๆที่ไม่เคยเห็นมาก่อน เช่น  $x = 0.1, x = 0.5, x = 0.9$ , เป็นต้น. แต่  $x = 0, x = 0.1111111, x = 0.2222222, \dots, x = 1$  ไม่ใช้อินพุตใหม่ๆ เพราะค่าเหล่านี้มีอยู่ในตัวอย่างแล้ว.

**โมเดลฟังชันพหุนาม.** ตัวอย่างนี้แสดงการประมาณค่าเอาท์พุต โดยการสร้างโมเดลขึ้นมา ซึ่งสำหรับที่นี่จะใช้ ฟังชันพหุนาม (Polynomial Function) เป็นโมเดล. ฟังชันพหุนามอธิบายความสัมพันธ์ระหว่างเอาต์พุตกับอินพุต ดังนี้

$$y(x, \mathbf{w}) = w_0 + w_1 \cdot x + w_2 \cdot x^2 + \dots + w_M \cdot x^M = \sum_{j=0}^M w_j x^j \quad (3.2)$$

โดย ค่าของฟังชัน  $y$  คือค่าประมาณเอาท์พุต, ตัวแปร  $x$  แทนค่าอินพุตที่สังสัย, ตัวแปร  $M$  เป็นลำดับ (order) ของฟังชันพหุนาม และ ตัวแปร  $w_0, w_1, w_2, \dots, w_M$  คือค่าสัมประสิทธิ์ต่างๆของพหุนาม และกรณีนี้ ก็ เป็นพารามิเตอร์ของโมเดลด้วย. เพื่อความสะดวก บางครั้งเราจะอ้างถึงด้วยพารามิเตอร์  $w_0, w_1, w_2, \dots, w_M$  หลายๆตัวนี้ ด้วยตัวแปร  $\mathbf{w}$ .



รูปที่ 3.3: รูปทรงต่างๆของฟังชันพหุนามปรับเปลี่ยนไปตามลำดับของพหุนาม และค่าของพารามิเตอร์.

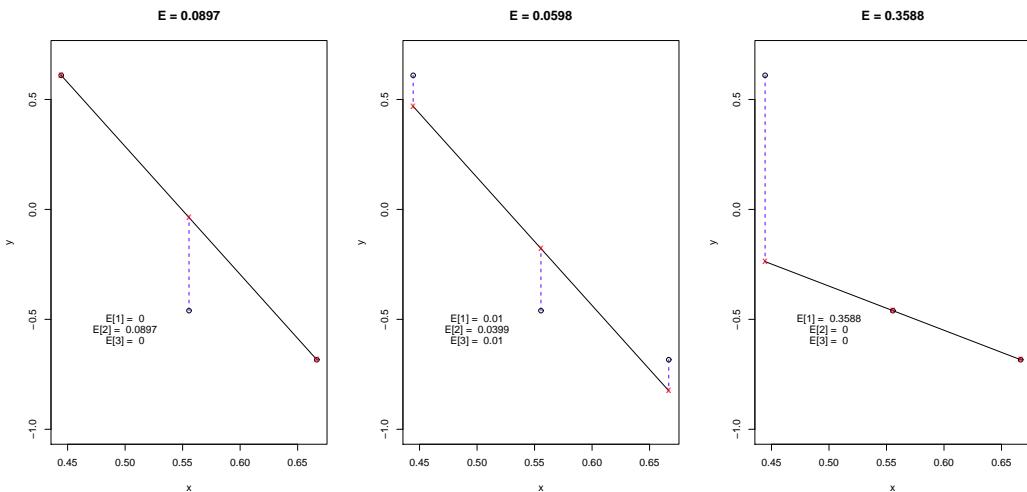
รูปกราฟของฟังชันพหุนามสามารถปรับเปลี่ยนรูปทรงได้ตามความซับซ้อนของฟังชัน (ระบุโดยลำดับพหุนาม  $M$ ) และค่าของพารามิเตอร์  $\mathbf{w}$ . รูป 3.3 แสดงตัวอย่างรูปทรงของกราฟจากฟังชันพหุนามที่ลำดับและพารามิเตอร์ค่าต่างๆ. สังเกตว่าค่าลำดับพหุนามยิ่งมาก ฟังชันพหุนามก็จะมีระดับของอิสระภาพ (Degree of Freedom) มาตรฐานไปด้วย. นั่นคือ หาก  $M = 0$  ฟังชันพหุนามสามารถแสดงเป็นเส้นตรงแนวโนนได้เท่านั้น, หาก  $M = 1$  ฟังชันพหุนามสามารถแสดงเป็นเส้นตรงที่ความชันต่างๆได้ด้วย, หาก  $M = 2$  ฟังชันพหุนามเพิ่มความสามารถที่จะโค้งอีก 1 งอ, หาก  $M = 3$  ฟังชันพหุนามเพิ่มความสามารถที่จะโค้งอีก 2 งอ เช่นนี้ เป็นต้น

**การฝึกโมเดล.** การฝึกโมเดล (Training) หรือการให้โมเดลเรียน (Learning) ก็คือการปรับค่าพารามิเตอร์  $\mathbf{w}$  เพื่อให้โมเดลสามารถฟังชันที่สนใจได้ โดยจะใช้ตัวอย่างที่มีมาช่วยปรับค่าพารามิเตอร์ เพื่อให้อาทีพุตของโมเดลใกล้กับอาทีพุตของตัวอย่างมากที่สุด สำหรับที่ค่าอินพุตเดียวกัน.

การวัดความใกล้ระหว่างอาทีพุตของโมเดลกับอาทีพุตของตัวอย่าง จะใช้ค่าผลต่างกำลังสอง (สมการ 3.3)

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 \quad (3.3)$$

โดย ตัวเลข  $\frac{1}{2}$  นี้ใส่เพื่อความสะดวก ซึ่งจะได้เห็นต่อไปในขั้นตอนการหาอนุพันธ์. ฟังชันเป้าหมาย (สมการ 3.3) นี้อาจจะเรียกว่า ฟังชันค่าผิดพลาด (Error Function) หรือฟังชันค่าใช้จ่าย (Cost Function). ถ้าค่าที่ทำนายผิดจากค่าอาทีพุตของตัวอย่างมาก ค่าผลต่างกำลังสองก็จะมากตามไปด้วย. หรือ ถ้าค่าที่ทำนายผิดจากเป้าหมายน้อย ค่าผลต่างกำลังสองก็จะน้อย. รูปที่ 3.4 แสดงค่าผลต่างกำลังสอง สำหรับผลของการทำนายแบบต่างๆ.



รูปที่ 3.4: ค่าผลต่างกำลังสองสำหรับผลทำนายจากโมเดลแบบต่างๆ ซึ่งทำนายค่าจุดตัวอย่าง 3 จุด. จุดข้อมูลตัวอย่างแสดงด้วยวงกลม. ภาคบทแทนค่าเอาร์พุตจากโมเดล (ค่าทำนาย). เส้นทึบแสดงให้เห็นค่าฟังชันของโมเดลตลอดช่วงค่าที่แสดง. เส้นประแสดงระยะห่างระหว่างค่าทำนายที่เป็นเอาร์พุตจากโมเดลและค่าตัวอย่างจากจุดข้อมูลที่อินพุตเดียวกัน. ภาพซ้าย โมเดลทายจุดข้อมูลที่ 1 และ 3 ได้อย่างแม่นยำ  $E[1]=0$  และ  $E[3]=0$  แต่ทายจุดข้อมูลที่ 2 ผิดไป โดยทายค่ามากเกินไป ซึ่งผลรวมของระยะห่างทั้งสามจะได้  $E = 0.0897$ . ภาพกลาง โมเดลทายจุดข้อมูลทั้ง 3 ผิดไปบ้าง โดยทายค่าจุดที่ 1 และ 3 ต่ำเกินไป แต่ทายจุดที่ 2 สูงเกินไป และได้ผลรวมของระยะห่างทั้งสาม  $E = 0.0598$ . ภาพขวา โมเดลทายจุดข้อมูล 1 ผิดไปมาก แต่ทายจุดที่ 2 และ 3 ได้อย่างแม่นยำ และได้ผลรวมของระยะห่างทั้งสาม  $E = 0.3588$ . การใช้ผลรวมของระยะห่างในการวัดคุณภาพของโมเดล เช่นนี้เพื่อให้สามารถสังหันความสามารถในการทำนายได้สำหรับทุกๆ จุดข้อมูล ซึ่งเชื่อว่าจะช่วยสังหันคุณภาพการเรียนแบบกระบวนการเบื้องหลังที่สนใจได้ดี

การทำนายค่าเอาร์พุตที่เป็นค่าต่อเนื่องแบบนี้ จะเรียกว่า การหาค่าถดถอย (Regression), ซึ่งจะต่างจากการจำแนกประเภท (Classification) ที่ค่าของเอาร์พุตจะจำกัดอยู่เฉพาะค่าในเซตของคำตออบที่กำหนด (การจำแนกประเภทจะอภิปรายในหัวข้อ 4.3). ค่าความใกล้ หรือ ค่าฟังชันค่าผิดพลาด  $E$  นี้ สำหรับการทำค่าเอาร์พุตตามค่าอินพุต, และตัววัดผลการทำงานคือค่าความผิดพลาดของการทำนาย ตามสมการ 3.3.

ทบทวนคำนิยามของการเรียนรู้ของเครื่องโดย ทอม มิชเซล [53], การหาค่าถดถอยนี้ จัดเป็นวิธีการเรียนรู้ของเครื่อง โดย ประสบการณ์คือการสังเกตอินพุตและเอาร์พุตของตัวอย่าง, งานคือการทำนายค่าเอาร์พุตตามค่าอินพุต, และตัววัดผลการทำงานคือค่าความผิดพลาดของการทำนาย ตามสมการ 3.3.

### 3.1.1 ฝึกโมเดลฟังชันพหุนามอันดับหนึ่ง

ตัวอย่างนี้เลือกฟังชันพหุนามอันดับ 1. นั่นคือ  $y(x, \mathbf{w}) = w_0 + w_1x$  ในการทำนายค่าฟังชันไซ โดยจะเลือกค่า  $w_0$  และ  $w_1$  จากค่าที่ให้ความผิดพลาดต่ำสุดของการทำนายข้อมูลตัวอย่าง  $N = 10$  จุดข้อมูล. ค่าความผิดพลาดต่ำสุดที่เป็นไปได้จะเกิดขึ้นเมื่อ  $\frac{\partial E}{\partial w_0} = 0$  และ  $\frac{\partial E}{\partial w_1} = 0$ . (ดูหัวข้อ 2.1 สำหรับทบทวนการทำค่าดีที่สุด) เมื่อแทนค่า  $E$  จากสมการ 3.3 จะได้

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

$$\frac{\partial \frac{1}{2} \sum_{n=1}^N \{w_0 + w_1 x_n - t_n\}^2}{\partial w_1} = 0 \quad (3.5)$$

และเมื่อทำอนุพันธ์เสร็จจะได้

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

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

หลังจากจัดรูปใหม่ โดยเรียงตามพารามิเตอร์ จะได้

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

$$w_0 \sum_{n=1}^N \{x_n\} + w_1 \sum_{n=1}^N \{x_n^2\} - \sum_{n=1}^N \{t_n \cdot x_n\} = 0 \quad (3.9)$$

ซึ่งเมื่อจัดรูปสมการ 3.8 และ 3.8 ให้อยู่ในรูปเมตริกซ์จะได้

$$\begin{bmatrix} N & \sum_{n=1}^N x_n \\ \sum_{n=1}^N x_n & \sum_{n=1}^N x_n^2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N t_n \\ \sum_{n=1}^N t_n \cdot x_n \end{bmatrix}. \quad (3.10)$$

จากค่าจุดข้อมูลในหัวข้อ 3.1, นั่นจะได้  $N = 10$ ,  $\sum_{n=1}^N x_n = 5$ ,  $\sum_{n=1}^N x_n^2 = 3.519$ ,  $\sum_{n=1}^N t_n = -0.499$ , และ  $\sum_{n=1}^N t_n x_n = -1.991$ . เมื่อแก้สมการแล้วจะได้ค่า

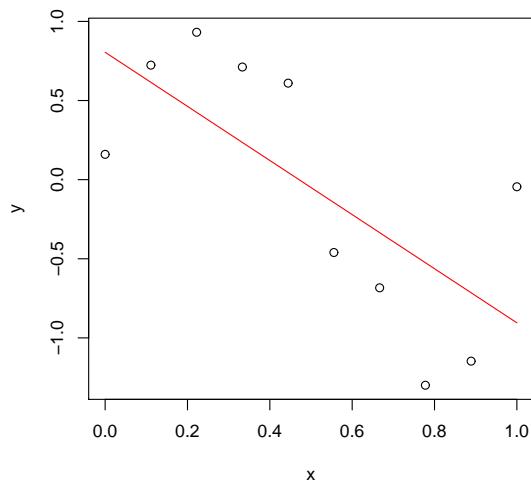
$$[w_0, w_1]^T = [0.805, -1.709]^T.$$

แนวทางที่ทำนี้เรียกว่า วิธีกำลังสองน้อยที่สุด (Least Squares Method) ซึ่งคือ การหาค่าของพารามิเตอร์ที่ทำให้ค่าทำนายผิดพลาดกำลังสองมีค่าน้อยที่สุด หรือ การหาค่าของพารามิเตอร์ที่เป็นตัวทำน้อยที่สุดของฟังชันค่าผิดพลาด. นั่นคือ การหา

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{1}{2} \sum_n \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2.$$

### 3.1.2 การใช้โมเดลฟังชันพหุนาม

หลังจากได้ค่าพารามิเตอร์ที่เหมาะสมแล้ว (เช่น  $[w_0, w_1]^T = [0.805, -1.709]^T$ ) ค่าประมาณของฟังชันก็สามารถคำนวณได้จากโมเดลที่แทนค่าพารามิเตอร์เหล่านั้น  $y = 0.805 + -1.709x$ . รูป 3.5 แสดงจุดข้อมูลและผลทำนายจากโมเดลหลังจากกระบวนการฝึกโมเดล.



รูปที่ 3.5: ผลจากโมเดลพหุนามอันดับหนึ่งที่ผิดแล้ว (เส้นสีแดง)

### 3.1.3 กิจกรรมปฏิบัติลงโปรแกรมหาค่าคาดถอยด้วยฟังชันพหุนาม

“ไม่ว่าบทกวีจะบรรยายกลืนร-schema ติของมะม่วงว่า อร่อย หอม หวาน เพียงใด  
ผู้อ่านก็ไม่อาจเข้าใจได้ดีเท่ากับได้ลิ้มลองซิมรสด้วยตนเอง  
เช่นเดียวกัน การอ่านหรือฟังทฤษฎีการเรียนรู้ของเครื่องมากเท่าใด ก็ไม่อาจช่วยให้เข้าใจได้  
เท่ากับลองประสบการณ์ด้วยตนเอง”

—ผู้เขียน

ทั้งข้อนี้ให้ตัวอย่าง เพื่อผู้อ่านสามารถนำไปทดลองลงมือปฏิบัติ เพื่อช่วยให้เข้าใจการทำงานของการสร้างโมเดลฟังชันพหุนาม และการใช้โมเดลในการทำนาย ซึ่งเป็นพื้นฐานเบื้องต้นสำหรับศาสตร์การเรียนรู้ของเครื่อง.

โค้ดข้างล่างนี้ใช้เพื่อสร้างจุดข้อมูล 10 จุด โดยแต่ละจุดข้อมูลมีค่าอินพุตตั้งแต่ 0 ถึง 1 และความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตคือ  $y = \sin(2\pi x) + \epsilon$  โดย  $\epsilon$  แทนสัญญาณรบกวน (ในโค้ดทำสัญญาณรบกวนด้วย `rnorm`)

```
N = 10;
dp.x <- seq(0, 1, len=N)
dp.t <- sin(2*pi*dp.x) + rnorm(10, mean=0, sd=0.3)
```

โค้ดข้างล่างนิยามฟังชัน `w.polyfit1` สำหรับฝึกโมเดล(หาค่าพารามิเตอร์). สังเกตเมทริกซ์ A และเวคเตอร์ b เปรียบเทียบกับสมการ 3.10.

```
w.polyfit1 <- function(xn, tn){
```

```
# xn and tn are training data.

N <- length(xn)

sumx <- sum(xn)
sumx2 <- sum(xn^2)
sumt <- sum(tn)
sumtx <- sum(tn*xn)

A <- matrix(c(N, sumx, sumx, sumx2), 2, 2, byrow=T )
b <- matrix(c(sumt, sumtx), 2, 1)

w <- solve(A,b)

return(w)
}##end w.polyfit1
```

โค้ดข้างล่างนิยามฟังชัน `y.poly1` สำหรับทำนายค่าเอาท์พุต. สังเกต  $y[i] = w[2]*x[i] + w[1]$  ทำการคำนวณโมเดลพหุนามที่ค่า  $x[i]$ .

```
y.poly1 <- function(x,w){
## w must have length 2: [w0 w1] = c(w[1], w[2])

N.x <- length(x)
y <- rep(0, len=N.x)
for(i in 1:N.x){
  y[i] = w[2]*x[i] + w[1];
}##end i

return(y)
}## end y.poly1
```

ข้อมูลตัวอย่างที่สร้างขึ้นมา (อินพุต `dp.x` และเอาท์พุต `dp.t`) นำมาใช้ฝึกโมเดลเพื่อหาค่าพารามิเตอร์  $w$  ได้ดังนี้

```
> w <- w.polyfit1(dp.x, dp.t)
```

```
> w
```

```
[,1]
```

```
[1,] 0.8050556
```

```
[2,] -1.7098887
```

หลังจากฝึกโมเดลตามขั้นตอนข้างต้นแล้ว โมเดลนี้ (สมบูรณ์แล้ว ด้วยค่าพารามิเตอร์  $\hat{\beta}$ ) สามารถนำไปประมวลค่าเอาท์พุตต่างๆ ที่อินพุต  $0.1, 0.25, 0.5, 0.75, 0.9$  ดังแสดงในตัวอย่างข้างล่างนี้

```
> y.poly1(0.1, w)
```

```
[1] 0.6340668
```

```
> y.poly1(0.25, w)
```

```
[1] 0.3775835
```

```
> apply(matrix(c(0.1, 0.25, 0.5, 0.75, 0.9)), 1, y.poly1, w)
```

```
[1] 0.6340668 0.3775835 -0.0498887 -0.4773609 -0.7338442
```

สังเกตุการใช้คำสั่ง `apply` แทนการเรียกฟังชันทีละครั้ง. ผู้อ่านสามารถเปลี่ยนกระบวนการที่ใช้สร้างจุดข้อมูล และ/หรือ เปลี่ยนค่าจำนวนข้อมูล  $N$  แล้วทดลองฝึกโมเดลและใช้โมเดลที่ฝึกประมวลค่าเอาท์พุตที่สร้างใหม่ เพื่อความเข้าใจที่ดีขึ้นได้.

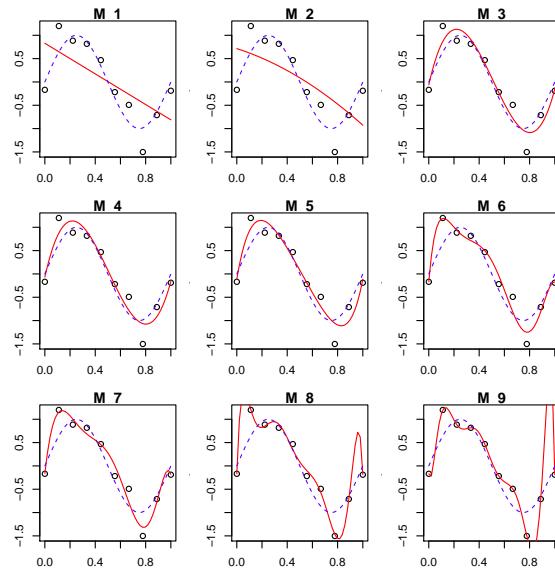
## 3.2 การเลือกโมเดล

สำหรับข้อมูลชุดหนึ่ง เราสามารถใช้โมเดลต่างๆ กันเพื่อประมวลข้อมูลชุดนั้นได้ เช่น เราสามารถใช้โมเดลฟังชันพหุนามอันดับต่างๆ เพื่อประมวลจุดข้อมูล ดังแสดงในรูป 3.6. สังเกตุพหุนามอันดับ 9 ผ่านจุดข้อมูลทุกจุด แต่รูปทรงกราฟจากโมเดลจะเปลี่ยนแปลงเร็วมาก. โดยทั่วไปแล้ว พหุนามอันดับ 9 สามารถผ่านจุดข้อมูล 10 จุดที่ใช้ฝึกได้ทุกจุด. หมายเหตุ พหุนามอันดับ 9 มีดีกรีของความเป็นอิสระเป็น 10 (10 Degrees of Freedom, การที่สามารถควบคุมด้วยสัมประสิทธิ์ 10 ค่า  $w_0, \dots, w_9$  ได้).

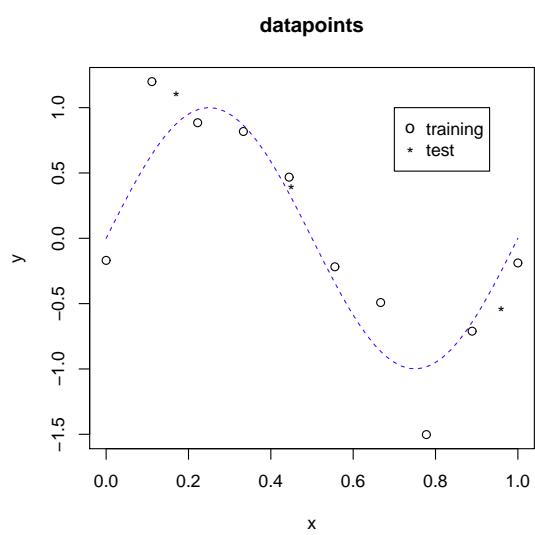
แต่หากได้ข้อมูลมาเพิ่ม (หรืออาจจะเป็นจุดข้อมูลที่กันไว้แต่แรก) ดังรูป 3.7 แล้วเมื่อประเมินค่าความผิดพลาดจากการทำนายด้วยโมเดลพหุนามที่ดีกรีต่างๆ จะได้ผลดังแสดงในรูป 3.8.

จากค่าความผิดพลาดจากการทำนายด้วยโมเดลพหุนามในรูป 3.8 แสดงให้เห็นว่า การทำนายข้อมูลชุดฝึก (สัญลักษณ์ ‘0’) ค่าความผิดพลาดลดลงเรื่อยๆ เมื่อดีกรีของพหุนามเพิ่มขึ้น (เทียบเท่ากับการใช้โมเดลที่มีความซับซ้อนสูงขึ้น). แต่การทำนายข้อมูลชุดทดสอบ (สัญลักษณ์ ‘\*’) ลดลงช่วงแรก แล้วเพิ่มขึ้นอย่างมากในช่วงหลัง.

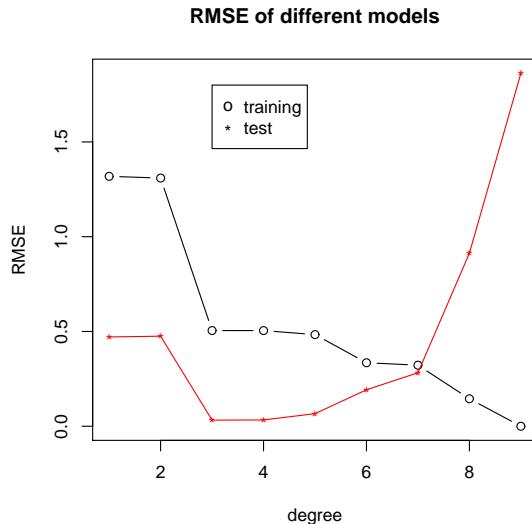
จริงๆแล้ว พหุนามดีกรีที่สูงกว่า สามารถที่จะทำตัวเสมอเป็น ดีกรีที่ต่ำกว่าได้ โดยการปรับให้ค่าสัมประสิทธิ์ที่เหลือมีกำลังสูงๆเป็นศูนย์. กล่าวอย่างง่ายคือ พหุนามดีกรีที่สูงกว่าสามารถที่จะให้ผลการทำนายที่ไม่แย่ไปกว่าพหุนามที่ดีกรีต่ำกว่าได้. ยิ่งกว่านั้น สำหรับตัวอย่างนี้ เราอาจบอกได้ว่า โมเดลที่จะทำนาย



รูปที่ 3.6: ผลจากโมเดลพหุนามอันดับต่างๆ กัน (เส้นทึบสีแดง) และค่าของฟังชันที่ใช้สร้างจุดข้อมูลเมื่อไม่มีสัญญาณรบกวน (เส้นประสีน้ำเงิน)



รูปที่ 3.7: จุดข้อมูลที่ใช้ก่อโมเดลและจุดที่ใช้ทดสอบ



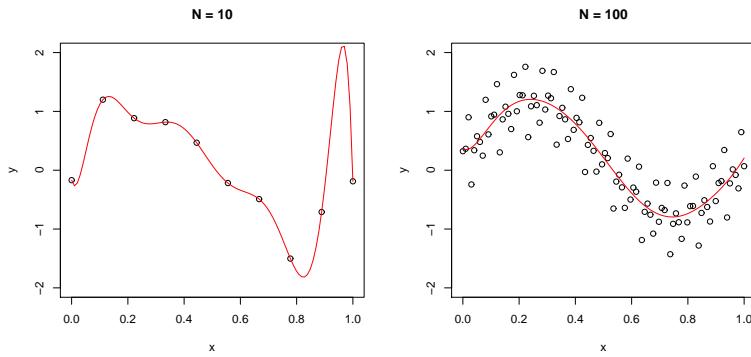
รูปที่ 3.8: ผลจากค่าผิดพลาดแบบอาร์เรอเมสของโมเดลพหุนามที่ดีกรีต่างๆ กับชุดข้อมูลที่ใช้ฝึก (แทนด้วยสัญลักษณ์ ‘o’) กับชุดข้อมูลทดสอบ (แทนด้วยสัญลักษณ์ ‘\*’)

ข้อมูลชุดนี้ได้ดีที่สุดคือ โมเดลแทนเส้นประสีน้ำเงิน ในรูป 3.6 ซึ่ง คือ  $\sin(2\pi x)$  ที่ใช้สร้างจุดข้อมูลขึ้นมา. และจากการขยายของอนุกรม泰勒 (Taylor Series Expansion) ของ  $\sin(2\pi x)$  น่าจะให้ผลที่ว่า ผลการทำนายน่าจะยิ่งดีขึ้นเมื่อดีกรีสูงขึ้น (ดูแบบฝึกหัดข้อ 10) เพื่อจะเข้าใจพฤติกรรมการฝึกโมเดล เมื่อพิจารณาค่าของสัมประสิทธิ์ที่ได้จากการฝึกโมเดล ดังแสดงในตาราง 3.1. จะเห็นว่า เมื่อดีกรีสูงขึ้น ขนาดของค่าสัมประสิทธิ์ใหญ่ขึ้นด้วย. โดยเฉพาะที่ดีกรี 9 ค่าสัมประสิทธิ์สูงปรับให้เข้ากับจุดข้อมูลอย่างมาก เห็นได้จากค่าสัมประสิทธิ์ที่มีขนาดใหญ่มากๆ (ค่าลบมากๆ หรือค่าบวกมากๆ) ทำให้พหุนามสามารถผ่านจุดข้อมูลได้ทุกจุด แต่ว่าระหว่างจุดข้อมูล ค่าของพหุนามกลับเปลี่ยนแปลงอย่างรุนแรง (ภาพท้ายสุดของรูป 3.6). สิ่งที่เกิดขึ้นก็คือ โมเดลที่ซับซ้อนสูงได้ปรับตัวให้เข้ากับสัญญาณ rob กวนของข้อมูล แทนที่จะปรับให้เข้ากับข้อมูลโดยทั่วไป ซึ่งแม้จะทำให้โมเดลสามารถพยากรณ์ข้อมูลได้อย่างแม่นยำ แต่ทำให้โมเดลสูญเสียความสามารถในการทำนายข้อมูลอื่นๆ (ที่ไม่ใช้ข้อมูลฝึกหัด)ลดลง. กรณีที่โมเดลที่ฝึกแล้วปรับตัวไปกับสัญญาณ rob กวน จะเรียกว่าเกิดโอเวอร์ฟิตติ้ง (Overfitting).

ตารางที่ 3.1: ค่าสัมประสิทธิ์ของฟังชันพหุนามหลังผ่านการฝึก

$M$	$w_0$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$
1	0.83	-1.64								
2	0.71	-0.87	-0.78							
3	-0.06	11.89	-34.39	22.41						
6	-0.17	31.54	-259	935	-1686	1446	-467			
9	-0.17	-18.6	1010	-11724	64085	-195203	349413	-365011	205751	-48301

รูป 3.9 แสดงผลการฝึกพหุนามดีกรี 9 ด้วยชุดข้อมูลที่สร้างจากฟังชันเดียว กัน แต่ขนาดข้อมูลต่างกัน. จากรูป 3.9 จะเห็นว่า ที่พหุนามดีกรีเดิม เมื่อจำนวนข้อมูลมากขึ้น ปัญหาโอเวอร์ฟิตติ้งลดลง. หรือ อาจ



รูปที่ 3.9: พหุนามดีกรี 9 ที่ฝึกกับชุดข้อมูลขนาดต่างกัน. ภาพซ้ายฝึกกับข้อมูลขนาด 10 จุดข้อมูล. ภาพขวาฝึกกับข้อมูลขนาด 100 จุดข้อมูล

จะกล่าวอีกอย่างได้ว่า เมื่อมีข้อมูลมากขึ้น เราสามารถใช้โมเดลที่ซับซ้อนมากขึ้นได้. บีชอบ[9] กล่าวถึงว่า ผู้เชี่ยวชาญบางคนถึงกับแนะนำว่า จำนวนข้อมูลไม่ควรน้อยกว่า 5 ถึง 10 เท่าของจำนวนพารามิเตอร์ของโมเดล.

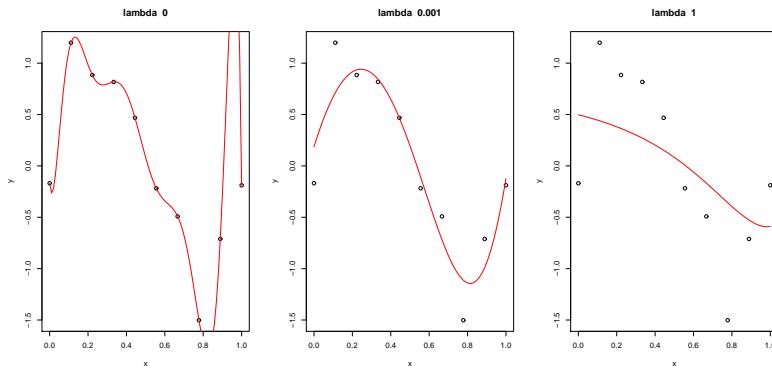
อย่างไรก็ตาม เราไม่ควรจะต้องจำกัดจำนวนพารามิเตอร์ของโมเดลตามขนาดของข้อมูลที่เรามี เพราะว่า เราควรจะสามารถเลือกจำนวนพารามิเตอร์ของโมเดลตามความซับซ้อนของปัญหาได้. นอกจากนั้น จำนวนพารามิเตอร์ของโมเดลอาจจะไม่ได้บอกระดับความซับซ้อนของโมเดลเสมอไป เช่น การทำเรกูล่าไรเซชัน (Regularization) หรือการใช้แนวทางแบบเบย์เซียน (Bayesian) ก็จะช่วยลดปัญหาโอเวอร์ฟิตติ้งได้ โดยไม่จำเป็นต้องลงจำนวนพารามิเตอร์ลง.

**เรกูล่าไรเซชัน.** เรกูล่าไรเซชัน (Regularization) เป็นวิธีหนึ่งที่นิยมใช้เพื่อช่วยลดปัญหาโอเวอร์ฟิตติ้ง. แนวทางก็คือ การใส่พินอลตี้เทอม (Penalty Term) เข้าไปในฟังชันเป้าหมาย เพื่อจะถ่วงดูลงให้ค่าสัมประสิทธิ์มีค่าใหญ่เกินไป. สมการ 3.11 แสดงฟังชันเป้าหมาย ที่ประกอบด้วยค่าผิดพลาดและพินอลตี้เทอม ซึ่งเป็นเทอมแรกและเทอมที่สองทางขวามีอตามลำดับ. นั่นคือ ฟังชันเป้าหมาย

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (3.11)$$

เมื่อ  $\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$  และ พารามิเตอร์  $\lambda$  ควบคุมสมดุลย์ระหว่างอิทธิพลของค่าผิดพลาดจากการคำนวณและอิทธิพลของพินอลตี้เทอม. จากมุมมองของการหาค่าน้อยที่สุด พารามิเตอร์  $\lambda$  อาจถูกเรียกเป็น ลากของพารามิเตอร์ (Lagrange Parameter, ดูหัวขอ 4.2 เพิ่มเติม หรือเรื่อง Constrained Optimization ของซองและเชค[18]). บีชอบ[9] ชี้ว่า บ่อยครั้งที่ พินอลตี้เทอมจะไม่รวม  $w_0$  (นั่นคือ ใช้  $\sum_{i=1}^M w_i^2$  แทน  $\|\mathbf{w}\|^2$ ). หรือ ถ้ามี  $w_0$  ก็อาจจะมีพารามิเตอร์ควบคุมอิทธิพลเฉพาะของตัวเอง.

รูป 3.10 แสดงผลจากเรกูล่าไรเซชัน จาลากของพารามิเตอร์ค่าต่างๆ. ภาพซ้ายสุด  $\lambda = 0$  เทียบเท่ากับการไม่ได้ใช้พินอลตี้เทอมเลย. โอเวอร์ฟิตติ้งเห็นได้ชัดในกรณีนี้. ภาพกลางแสดงค่าลากของพารามิเตอร์ที่เหมาะสม ค่าลากของพารามิเตอร์ที่เหมาะสมจะช่วยบังคับโมเดลที่มีความซับซ้อนสูงให้ทำตัว



รูปที่ 3.10: พหุนามดีกรี 9 กับเรกูล่าไรเซชันด้วยลากองจ์พารามิเตอร์ค่าต่างๆ. ภาพข่ายแสดงโอลเวอร์ฟิตติ้ง ( $\lambda = 0$ ). ภาพกลางแสดงโมเดลที่เหมาะสม ( $\lambda = 0.001$ ). ภาพขวาแสดงอันเดอร์ฟิตติ้ง ( $\lambda = 1$ )

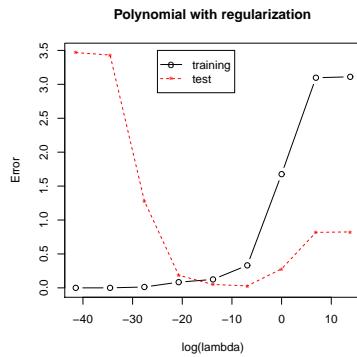
เมื่อกับโมเดลความซับซ้อนต่ำลง. ค่าประมาณจากโมเดลแสดงด้วยเส้นทึบสีแดง มีลักษณะใกล้เคียงกับ  $\sin(2\pi x)$  ที่ใช้สร้างจุดข้อมูล. แต่ถ้าหากใช้ค่าลากองจ์พารามิเตอร์มากเกินไป ก็อาจทำให้เกิดอันเดอร์ฟิตติ้ง (Underfitting) ได้ดังแสดงในภาพขวาสุด.

เทียบกับตาราง 3.1 ตาราง 3.2 แสดงให้เห็นว่า ถ้าใช้ค่า  $\lambda$  ใหญ่พอดี เรกูล่าไรเซชันช่วยควบคุมให้ค่าสัมประสิทธิ์ไม่ใหญ่เกินไปได้. แต่ถ้าใช้ค่า  $\lambda$  ใหญ่เกินไป ก็ทำให้ค่าสัมประสิทธิ์น้อยเกินไปได้ เช่นกัน. รูป 3.11 แสดงผลค่าผิดพลาดของโมเดลพหุนามดีกรี 9 กับเรกูล่าไรเซชันที่ค่าลากองจ์ต่างๆ เมื่อประเมินกับข้อมูลชุดฝึกหัดและชุดทดสอบ. สังเกตว่าค่าผิดพลาดของโมเดล เมื่อประเมินกับชุดฝึกหัด ค่าผิดพลาดของโมเดลจะน้อยลง เมื่อใช้ค่าลากองจ์พารามิเตอร์น้อยๆ (ให้ผลคล้ายกับการใช้ฟังชันพหุนามดีกรีสูงๆ). ส่วนเมื่อประเมินกับชุดทดสอบ ค่าผิดพลาดของโมเดลจะต่ำสุดที่ค่าลากองจ์ราวๆ 0.001 หรือ  $\log(\lambda) \approx -6.91$ .

ตารางที่ 3.2: ค่าสัมประสิทธิ์ของพหุนามกับเรกูล่าไรเซชันที่ลากองจ์พารามิเตอร์ค่าต่างๆ

สัมประสิทธิ์	$\lambda = 0$	$\lambda = 10^{-5}$	$\lambda = 1$
$w_0$	-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

สำหรับการประเมินโมเดล สิ่งที่สำคัญคือคุณสมบัติที่โมเดลสามารถทำงานอย่างข้อมูลที่ไม่เคยเห็นมาก่อนได้



รูปที่ 3.11: เรกว่าไเรซ์ชั่นด้วยลากองจ์พารามิเตอร์ค่าต่างๆ ประเมินด้วยข้อมูลชุดฝึกหัด กับ ชุดทดสอบ

ดี หรือเรียกว่า คุณสมบัติความทั่วไป (Generalization). เพื่อเลือกความซับซ้อนของโมเดล เช่น การเลือก ดีกรีของพหุนาม หรือการเลือกค่าลากองจ์ของเรกว่าไเรซ์ชั่น เราทำได้โดย การวัดคุณสมบัติความทั่วไปของ โมเดลที่ความซับซ้อนต่างๆ กัน. วิธีง่ายๆ และตรงไปตรงมาที่สุด ก็คือ การแบ่งข้อมูลออกเป็น 2 ชุด ได้แก่ ชุด ที่ใช้ฝึกโมเดล (Training Set) ที่ใช้หาค่าพารามิเตอร์  $\mathbf{W}$  และชุดตรวจสอบ (Validation Set หรือ Hold-Out Set) ที่ใช้เลือกความซับซ้อนของโมเดล เช่น  $M$  หรือ  $\lambda$ .

นอกจากนี้ เมื่อเลือกโมเดลได้แล้ว เพื่อทดสอบความสามารถของโมเดล เราควรจะมีข้อมูลที่แยกมา อีกชุดเพื่อทดสอบ เรียกว่า ชุดทดสอบ (Test Set). ที่ต้องมีชุดทดสอบนี้อีก เพื่อกันปัญหาที่เราอาจจะเลือก โมเดลที่เกิดโอเวอร์ฟิตติ้งกับชุดตรวจสอบ. ถ้าเราเลือกโมเดลได้ดี ค่าผิดพลาดที่ประเมินกับข้อมูลชุดทดสอบ ไม่ควรห่างมากจากค่าผิดพลาดที่ประเมินกับข้อมูลชุดตรวจสอบ.

วิธีการเลือกโมเดลโดยการแบ่งบางส่วนของข้อมูลมาเป็นชุดตรวจสอบเดิ้นนั้นเหมาะสมกับกรณีที่มีข้อมูล จำนวนมาก. แต่หากข้อมูลมีขนาดจำกัด ผู้ทำโมเดลควรจะทำอย่างไร เมื่อการฝึกโมเดลให้ดีต้องการข้อมูล จำนวนมาก และการทำวิเคราะห์ที่ดีหรือการทดสอบที่ดีก็ต้องการข้อมูลจำนวนมากเช่นกัน. การแบ่งส่วน ข้อมูลที่ขนาดเล็กอยู่แล้ว ยังจะทำให้แต่ละส่วนมีขนาดเหล็กลงไปอีก. วิธีหนึ่งที่ออกแบบมาเพื่อช่วยลดปัญหา นี้ คือ วิธีครอสвалиเดชั่น (Cross-Validation). แนวคิดคือ การฝึกโมเดลและการทำวิเคราะห์หลายครั้ง แล้วเอาผลมาเฉลี่ยกัน เพื่อหาโมเดลที่มีคุณสมบัติความทั่วไปดีที่สุด โดยที่จะแบ่งข้อมูลออกเป็น  $S$  ส่วน แต่ละครั้งจะเลือกส่วนหนึ่งมาเป็นชุดตรวจสอบ และใช้ส่วนที่เหลือ ( $S - 1$  ส่วน) สำหรับฝึกโมเดล. สำหรับ  $S$  ส่วน จะเรียกว่า วิธีครอสвалиเดชั่น  $S$  พับ ( $S$ -Fold Cross-Validation). วิธีครอสвалиเดชั่น  $S$  พับทำการ ฝึกและวิเคราะห์ที่  $S$  ครั้ง ที่แต่ละครั้งจะใช้ส่วนที่ทำวิเคราะห์แต่ละต่างกัน. เมื่อทำงานครบทุกส่วนแล้ว จึง นำค่าผิดพลาดที่ประเมินจากแต่ละครั้ง รวม  $S$  ค่ามาหาค่าเฉลี่ย เป็นค่าความผิดพลาดครอสвалиเดชั่นของ โมเดล (Cross-Validation Error). ค่าความผิดพลาดครอสвалиเดชั่นนี้สามารถใช้เปรียบเทียบกับโมเดลอื่น (หรือโมเดลเดียวกันแต่ความซับซ้อนอื่น) เพื่อหาโมเดล(หรือความซับซ้อน)ที่ดีที่สุด.

รูป 3.12 แสดงแผนภาพการแบ่งข้อมูลสำหรับวิธีครอสвалиเดชั่น 5 พับ ( $S = 5$ ) และการจัดสรรงroup ข้อมูล สำหรับการฝึกและการทำวิเคราะห์ที่ดีในแต่ละครั้ง. การฝึกและทำวิเคราะห์แต่ละครั้งจะเรียกเป็นวิเคราะห์ นั้น (Validation Run). ในภาพแสดง 5 วิเคราะห์ที่รันแรก (Run 1) ฝึกโมเดลด้วยข้อมูล 4 ส่วนแรก



รูปที่ 3.12: วิธีกรอสวัลิเดชั่น 5 พับ. ข้อมูลทั้งหมดจะถูกแบ่งออกเป็น 5 ส่วน และวิธีกรอสวัลิเดชั่นจะทำทั้งหมด 5 ครั้ง โดยแผนภาพแสดงในเห็นว่า การทำครั้งแรกใช้ข้อมูล 4 ส่วนแรกสำหรับการฝึก และส่วนสุดท้ายสำหรับวัลิเดชั่น. ส่วนที่ใช้สำหรับวัลิเดชั่นจะแสดงเป็นสีเข้ม. ครั้งที่สอง สาม สี่ และห้าก็ทำเช่นเดิม เพียงแต่เปลี่ยนส่วนที่มาทำวัลิเดชั่น.

และโมเดลที่ฝึกแล้วไปทำวัลิเดชั่นกับส่วนหลังสุด (แรงงานสีเข้มในรูป). รันที่สองฝึกโมเดลด้วยข้อมูลส่วนอื่นยกเว้นส่วนที่ 4 (แรงงาน) และทำวัลิเดชั่นกับส่วนที่ 4 ที่กันออกไว้. ทำเช่นนี้จนครบ 5 รัน และนำเอาผลที่ได้มาเฉลี่ย.

ด้วยวิธีนี้ แต่ละรันจะฝึกโมเดลด้วยข้อมูลขนาด  $S - 1$  ของที่มีอยู่ และผลค่าผิดพลาดจากกรอสวัลิเดชั่นก็เป็นค่าเฉลี่ยของค่าผิดพลาดที่ได้จากการทุกส่วนของข้อมูล. วิธีกรอสวัลิเดชั่นนี้ทำให้เสมอว่ามีข้อมูลมากขึ้น ทำการฝึกและการทำวัลิเดชั่น.

เมื่อวิธีกรอสวัลิเดชั่นจะช่วยลดปัญหาของขนาดข้อมูลที่จำกัด และเป็นวิธีที่ใช้ข้อมูลได้อย่างคุ้มค่า แต่ข้อเสียของวิธีกรอสวัลิเดชั่นคือ การที่ต้องทำการรันทั้งหมด  $S$  ครั้ง โดยเฉพาะ หากถ้าการรันแต่ละครั้งใช้เวลา many. กล่าวอีกนัยหนึ่ง วิธีกรอสวัลิเดชั่นใช้การคำนวนที่เพิ่มขึ้น เพื่อแก้ปัญหาข้อมูลขนาดเล็ก. ดังนั้น หากถ้าการรันแต่ละครั้งใช้การคำนวนมากอยู่แล้ว แนวทางของวิธีกรอสวัลิเดชั่นอาจจะไม่เหมาะสม เช่น การฝึกโมเดลอาจมีการคำนวนสูง หรือการทำกรอสวัลิเดชั่นกับโมเดลที่มีพารามิเตอร์ที่ควบคุมความซับซ้อนหลายตัว อथิ การทำกราฟิกเรขาคณิตด้วยลากองจ์พารามิเตอร์หลายตัว อาจทำปริมาณการคำนวนเพิ่มขึ้นมาก.

**เกณฑ์สารสนเทศ.** นอกจากแนวทางของวิธีกรอสวัลิเดชั่นที่ใช้ผลทดสอบกับข้อมูลชุดวัลิเดชั่นเป็นดัชนีบ่งชี้แล้ว อีกแนวคิดหนึ่งก็คือการหาสูตรคำนวนที่ใช้ค่าประเมินผลกับข้อมูลชุดฝึกหัด รวมความซับซ้อนของโมเดลเข้าไปด้วย โดยไม่ต้องทำวัลิเดชั่น. แนวคิดนี้ ก็คือแนวทางของเกณฑ์สารสนเทศ (Information Criteria). เกณฑ์สารสนเทศ เป็นวิธีที่พยายามจะลดนำหนักผลประเมินที่ดีเกินไปกับข้อมูลชุดฝึกหัด(ที่อาจเกิดจากโควตาฟิตติ้ง) ด้วยการใช้ความซับซ้อนของโมเดลเข้าไปด้วย เช่น เกณฑ์สารสนเทศอาไกอิเกะ (Akaike Information Criteria, คำย่อ AIC).

เกณฑ์สารสนเทศอาไกอิเกะ คำนวนได้จาก

$$AIC = \ln p(\mathcal{D} | \mathbf{w}_{ML}) - M. \quad (3.12)$$

โดย  $p(\mathcal{D}|\mathbf{w}_{ML})$  คือลักษณะที่มีของค่าควรจะเป็นที่จัดแล้วพอดีที่สุด (Best-Fit Log Likelihood) หรือกล่าวง่ายๆคือค่าลักษณะที่มีของความน่าจะเป็นที่ไม่เดลของจะให้ค่าเหมือนกับข้อมูลฝึกหัดเมื่อใช้ค่าพารามิเตอร์ที่ดีที่สุด ( $\mathbf{w}_{ML}$ ) และ  $M$  คือจำนวนพารามิเตอร์ของโมเดล.

ค่าเกณฑ์สารสนเทศอาไอกิเกะ AIC นี้ยิ่งมาก หมายถึงโมเดลยิ่งดียิ่งเท่าไร ข้อดีของการใช้สูตรนี้ในการเลือกโมเดลคือเราแค่ทำการฝึกโมเดลอย่างเดียว ไม่ต้องทำวิเคราะห์ ดังนั้นจึงไม่ต้องแบ่งข้อมูลไว้สำหรับวิเคราะห์และยังตัดขั้นตอนการทำวิเคราะห์ชั้นลงได้ด้วย. อายุ่งไรก็ตาม บีชอบ[9] ได้ชี้ว่า ในทางปฏิบัติ สูตรลักษณะแบบนี้มักจะลำเอียงไปเลือกโมเดลที่ซับซ้อนน้อยเกินไป.

### 3.3 ความน่าจะเป็น

ปัจจัยหลักเรื่องหนึ่งสำหรับวิชาการเรียนรู้ของเครื่องและการรู้จำรูปแบบ ก็คือความไม่แน่นอน (Uncertainty). ความไม่แน่นอนอาจมาจากหลายสาเหตุ เช่น ความไม่เที่ยงของเครื่องมือ หรือวิธีการวัด หรือวิธีการเก็บข้อมูล, สัญญาณรบกวน, ขนาดของข้อมูลที่จำกัด, หรือแม้แต่รรรมชาติความหลากหลายและความแปรผันของข้อมูลเอง. ทฤษฎีความน่าจะเป็น (Probability Theory) เป็นแนวทางหนึ่งที่ให้กรอบวิธีการสำหรับการวัดและการจัดการกับความไม่แน่นอน และยังเป็นพื้นฐานที่สำคัญสำหรับการเรียนรู้ของเครื่อง.

#### 3.3.1 เซต

รูปแบบ (Pattern) ที่เราสนใจ เช่น ลักษณะของรูปซ้าง, สัญญาณเสียงของคำว่า “คันหา” เพื่อระบบสั่งงานด้วยเสียง, ลักษณะสำคัญของเอกสาร ที่บอกว่าเป็นเอกสารเกี่ยวกับข่าวกีฬา, ลักษณะสำคัญของดอกไม้ที่ใช้ระบุได้ว่าเป็นดอกไม้พันธุ์หรัญญิก เป็นต้น. ทฤษฎีความน่าจะเป็นจะมองรูปแบบเหล่านี้เป็นเหตุการณ์ เช่น เหตุการณ์ที่รูปที่สนใจเป็นภาพของซ้าง, เหตุการณ์ที่สัญญาณเสียงที่ได้มาเป็นเสียงของคำว่า “คันหา” เป็นต้น.

เซตแทนกลุ่มของรูปแบบหรือเหตุการณ์ที่เราสนใจ เช่น เซตของรูปซ้างแบบต่างๆ, เซตของเสียงคำว่า “คันหา” ที่น้ำเสียง สำเนียง ต่างๆ, เซตของอักษรในภาษาไทย, เป็นต้น. ตาราง 3.3<sup>1</sup> แทนสัญญาณลักษณะ และคำศัพท์ที่เกี่ยวข้องกับเซตและความน่าจะเป็นที่ใช้บ่อยๆ.

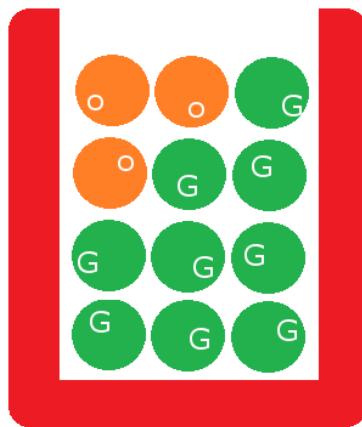
#### 3.3.2 ความน่าจะเป็น

กล่าวง่ายๆแล้ว ความน่าจะเป็น (Probability) ก็คือโอกาสที่เหตุการณ์ที่สนใจจะเกิดขึ้น. นั่นคือ หากสมมติว่าเราทำการทดลองซ้ำๆเป็นจำนวน  $N$  ครั้ง โดยให้สภาพแวดล้อมเหมือนเดิมมากเท่าที่จะเป็นไปได้. กำหนดให้  $A$  เป็นเหตุการณ์ที่เราสนใจ โดย  $A$  อาจจะเกิดขึ้นหรือไม่เกิดในแต่ละการทำซ้ำก็ได้. สิ่งที่เราจะพบคือ เมื่อจำนวนทำซ้ำ  $N$  ใหญ่มากและใหญ่ขึ้นๆเป็นลำดับ อัตราส่วนของจำนวนครั้งที่จะเกิด  $A$  ในแต่ละการทำซ้ำ จะเข้าสู่ค่าหนึ่ง ซึ่งค่านั้นคือความน่าจะเป็นของ  $A$ .

<sup>1</sup> ดัดแปลงจาก ตาราง 1.1 ของกริมเมต์กับสเตรชาเกอร์[30].

ตารางที่ 3.3: ภาษาเฉพาะที่ใช้ในเรื่องเซตกับเรื่องความน่าจะเป็น

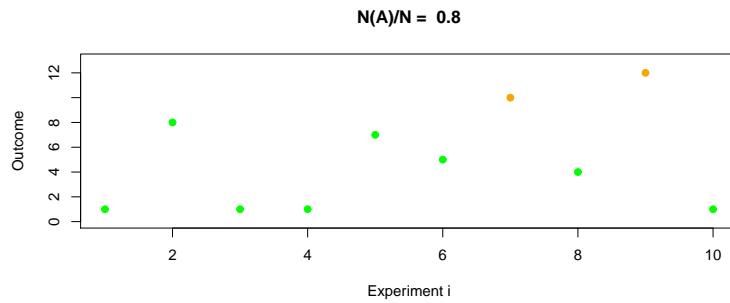
สัญลักษณ์ทั่วไป	ภาษาเฉพาะในเรื่องเซต	ภาษาเฉพาะในเรื่องความน่าจะเป็น
$\Omega$	กลุ่มของตัวตุ	ปริภูมิตัวอย่าง (Sample Space)
$\omega$	สมาชิกของ $\Omega$	เหตุการณ์พื้นฐาน หรือรูปแบบ
$A$	เซตย่อย (Subset) ของ $\Omega$	เหตุการณ์ที่มีรูปแบบใน $A$
$A^c$	ส่วนเติมเต็ม (Complement) ของ $A$	เหตุการณ์ที่ไม่มีรูปแบบใน $A$
$A \cap B$	อินเตอร์เซกชัน (Intersection)	เหตุการณ์ที่มีรูปแบบทั้งใน $A$ และใน $B$
$A \cup B$	ยูเนียชน (Union)	เหตุการณ์ที่มีรูปแบบใน $A$ หรือใน $B$ หรือในทั้งคู่
$A \setminus B$	ผลต่าง (Difference)	เหตุการณ์ที่มีรูปแบบใน $A$ แต่ไม่มีรูปแบบใน $B$
$\emptyset$	เซตว่าง (Empty Set)	เหตุการณ์ที่เป็นไปไม่ได้



รูปที่ 3.13: กล่องใส่ลูกบอล ซึ่งมีลูกบอลอยู่ภายใน 12 ลูก เป็นลูกบอลสีส้มสามลูกและที่เหลือเป็นสีเขียว

ขยายความคือ หากกำหนดให้  $N(A)$  แทนจำนวนครั้งที่จะเกิดเหตุการณ์  $A$  ในการทำซ้ำทั้งหมด  $N$  ครั้ง อัตราส่วน  $\frac{N(A)}{N}$  จะค่อยๆ ลุ่มเข้าสู่ค่าๆ หนึ่ง เมื่อ  $N$  เพิ่มขึ้น. ค่าๆ นั้นของอัตราส่วนจะเรียกว่า ความน่าจะเป็นที่เหตุการณ์  $A$  จะเกิดขึ้น ในแต่ละการทำซ้ำ โดยค่าความน่าจะเป็นนี้ แทนด้วยสัญลักษณ์  $\mathbb{P}(A)$ .

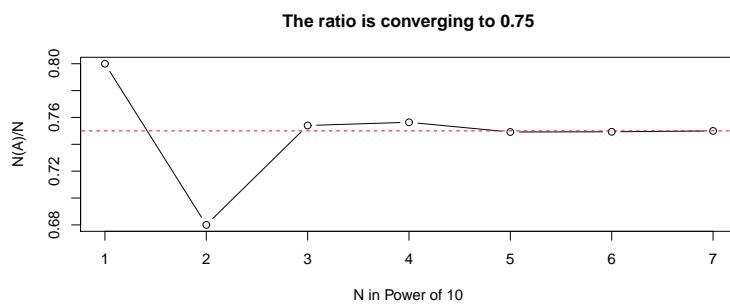
หาก  $A$  เป็นเหตุการณ์ที่เป็นไปไม่ได้,  $A = \emptyset$ , ดังนั้น  $N(\emptyset) = 0$  และ  $\mathbb{P}(\emptyset) = 0$ . ในทางกลับกัน หาก  $A$  พุดถึงทุกๆ เหตุการณ์ที่เป็นไปได้  $A = \Omega$ , ดังนั้น  $\mathbb{P}(\Omega) = 1$ . ค่าของความน่าจะเป็น จะอยู่ระหว่าง  $[0, 1]$ . ตัวอย่างเช่น สมมติมีกล่องใส่ลูกบอลสีต่างๆ ดังแสดงในรูป 3.13 หากเราสุ่มหยิบลูกบอล ออกมาจากกล่อง 1 ลูก, ให้  $A$  เป็นเหตุการณ์ที่เราหยิบได้ลูกบอลสีเขียว. สมมติเราทำการทดลอง(สุ่มหยิบ)ซ้ำ  $N = 10$  เราได้ผลดังแสดงในรูป 3.14 ซึ่งบอกได้ว่า อัตราส่วนที่หยิบได้ลูกบอลสีเขียว เป็น  $\frac{N(A)}{N} = \frac{8}{10} = 0.8$ . หากเราเพิ่มจำนวนการทำซ้ำ  $N$  จาก 10 เป็น 100, 1000, 10000, ... เราจะเริ่มเห็นว่าอัตราส่วน  $\frac{N(A)}{N}$  ลุ่มเข้าสู่ค่าๆ หนึ่ง, ดังแสดงในตาราง 3.4. เมื่อนำค่าต่างๆ ไปพล็อตกราฟ จะได้ดังรูป 3.15 ซึ่งจะเห็นว่าค่าที่อัตราส่วน  $\frac{N(A)}{N}$  ลุ่มเข้าหาคือ 0.75. นั่นคือ ความน่าจะเป็นของการสุ่มหยิบได้ลูกเขียว,  $\mathbb{P}(A) = 0.75$ . มองจากอีกมุมหนึ่ง ในกล่องมีลูกบอล 12 และเป็นลูกสีเขียวอยู่ 9 หากสุ่มหยิบด้วยความยุติธรรมแล้ว โอกาสที่จะหยิบได้ลูกเขียว ก็จะเป็น  $\frac{9}{12} = 0.75$  ซึ่งค่าที่คำนวนนี้ก็สอดคล้องกับค่าความน่าจะเป็นที่ได้การทดลองข้างต้น.



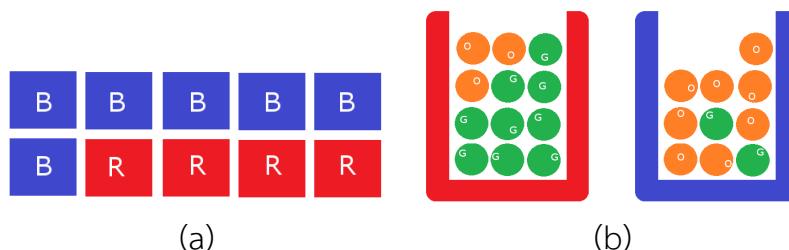
รูปที่ 3.14: ผลจากการทดลองสุ่มหยิบลูกบอล 10 ครั้ง จากกล่องลูกบอลที่แสดงในรูป 3.13. ลูกที่ 1–9 สีเขียว ลูกที่ 10–12 สีส้ม. จากการสุ่มทำ 10 ครั้ง มีครั้งที่ 7 และ 9 ที่หยิบได้ลูกบอลสีส้ม. ดังนั้น อัตราส่วนจำนวนครั้งที่หยิบได้ลูกบอลสีเขียว คือ 0.8 (ระบุที่ด้านบนของภาพ)

ตารางที่ 3.4: อัตราส่วนของการสุ่มได้ลูกบอลสีเขียว เมื่อจำนวนการทำซ้ำเพิ่มขึ้น

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



รูปที่ 3.15: อัตราส่วน  $\frac{N(A)}{N}$  ลู่เข้าหา  $\mathbb{P}(A) = 0.75$  เมื่อ  $N$  เพิ่มขึ้น (แสดงด้วยเส้นประสีแดง)



รูปที่ 3.16: ตัวอย่างกล่องสีบรรจุลูกบอลสี สำหรับอภิปรายพื้นฐานเรื่องความน่าจะเป็นแบบมีเงื่อนไข. ภาพ (a) แสดงสัดส่วนของกล่องสีฟ้ากับกล่องสีแดง (มีกล่องสีแดงอยู่ 4 กล่อง ที่เหลือเป็นสีฟ้า). ภาพ (b) แสดงสัดส่วนของลูกบอลสีภายในกล่องสองกล่อง โดย กล่องข้างสีแดงมีลูกบอลสีส้มอยู่ 3 ลูก ที่เหลือสีเขียว และกล่องขวาสีฟ้ามีลูกบอลสีเขียวอยู่ 2 ลูก ที่เหลือสีส้ม

ความน่าจะเป็นมีเงื่อนไข (Conditional Probability). จากตัวอย่างของรูป 3.13 ลองดูอีกตัวอย่างที่คราวนี้มีกล่อง 2 แบบ กล่องสีแดง และ กล่องสีน้ำเงิน ดังรูป 3.16. สมมติว่ากล่องที่จะได้ก็ถูกสุ่มมา และโอกาสที่จะสุ่มได้กล่องแดงเป็น  $\frac{4}{10}$  หรือความน่าจะเป็นที่จะได้กล่องสีแดง  $\mathbb{P}(C = 'r') = 0.4$ , โดย  $C = 'r'$  แทนเหตุการณ์ที่จะได้กล่องสีแดง. ในทำนองเดียวกัน ความน่าจะเป็นที่จะได้กล่องสีฟ้า  $\mathbb{P}(C = 'b') = 0.6$ .

หากเรารู้แล้วว่าเป็นกล่องสีแดง เมื่อสุ่มหยิบลูกบอลมา เราจะรู้ว่าโอกาสที่จะหยิบได้ลูกบอลสีเขียว คือ  $\frac{9}{12} = 0.75$  หรือกล่าวได้ว่า ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีเขียวเมื่อหยิบจากกล่องสีแดง  $\mathbb{P}(B = 'g' | C = 'r') = 0.75$  โดย  $B = 'g'$  แทนเหตุการณ์ที่จะหยิบได้ลูกบอลเป็นสีเขียว. ทำนองเดียวกัน ก็จะได้ความน่าจะเป็นมีเงื่อนไขอี่นๆ (Conditional Probabilities) ดังนี้

- $\mathbb{P}(B = 'o' | C = 'r') = 0.25$ ,
- $\mathbb{P}(B = 'g' | C = 'b') = 0.20$ ,
- $\mathbb{P}(B = 'o' | C = 'b') = 0.80$ .

สังเกตว่า ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีเขียวเมื่อรู้ว่าเป็นกล่องสีแดง  $\mathbb{P}(B = 'g' | C = 'r')$  ไม่เหมือนกับความน่าจะเป็นที่จะหยิบลูกบอลสีเขียวและได้กล่องสีแดง  $\mathbb{P}(B = 'g', C = 'r')$ . สำหรับความน่าจะเป็นที่จะหยิบลูกบอลสีเขียวเมื่อรู้ว่าเป็นกล่องสีแดง เราไม่ต้องสนใจเลยว่าโอกาสที่จะได้กล่องสีแดง เป็นอย่างไร. ในขณะที่ ความน่าจะเป็นที่จะหยิบลูกบอลสีเขียวและได้กล่องสีแดงจะประกอบด้วยโอกาสที่จะได้กล่องสีแดง  $\mathbb{P}(C = 'r')$  และโอกาสที่จะหยิบได้ลูกบอลสีเขียวจากกล่องนั้น  $\mathbb{P}(B = 'g' | C = 'r')$  ซึ่งเขียนเป็นสมการได้

$$\begin{aligned}\mathbb{P}(B = 'g', C = 'r') &= \mathbb{P}(C = 'r') \cdot \mathbb{P}(B = 'g' | C = 'r') \\ &= (0.4) \cdot (0.75) = 0.3.\end{aligned}\tag{3.13}$$

ในทำนองเดียวกันก็จะได้ว่า

- ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีส้มและได้กล่องสีแดง  
(หรือกล่าวอีกอย่างคือ ความน่าจะเป็นที่จะได้กล่องสีแดงและหยิบได้ลูกบอลสีส้ม)

$$\begin{aligned}\mathbb{P}(B = 'o', C = 'r') &= \mathbb{P}(C = 'r', B = 'o') \\ &= (0.4) \cdot (0.25) = 0.1,\end{aligned}$$

- ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีเขียวและได้กล่องสีฟ้า,  $\mathbb{P}(B = 'g', C = 'b') = (0.6) \cdot (0.2) = 0.12$ ,
- ความน่าจะเป็นที่จะหยิบได้ลูกบอลสีส้มและได้กล่องสีฟ้า,  $\mathbb{P}(B = 'o', C = 'b') = (0.6) \cdot (0.8) = 0.48$ .

ตาราง 3.5 สรุปค่าความน่าจะเป็นเหล่านี้. สังเกตุประเด็นใหญ่ 3 ประเด็น ดังนี้ ประเด็นที่ 1 ผลรวมของความน่าจะเป็นของทุกๆเหตุการณ์เป็น 1. นั่นคือ

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

ธรรมชาตินี้เป็นคุณสมบติพื้นฐานของความน่าจะเป็น.

ประเด็นที่ 2 ความน่าจะเป็นของเหตุการณ์  $X$  เท่ากับผลรวมของความน่าจะเป็นของเหตุการณ์  $X$  และ  $Y$  สำหรับทุกๆความเป็นไปได้ของ  $Y$ ,

$$\begin{aligned}\mathbb{P}(C = 'r') &= \mathbb{P}(C = 'r', B = 'g') + \mathbb{P}(C = 'r', B = 'o') \\ &= 0.3 + 0.1 = 0.4\end{aligned}\tag{3.14}$$

$$\begin{aligned}\mathbb{P}(C = 'b') &= \mathbb{P}(C = 'b', B = 'g') + \mathbb{P}(C = 'b', B = 'o') \\ &= 0.12 + 0.48 = 0.6.\end{aligned}\tag{3.15}$$

ข้อสังเกตุนี้บอกรูปแบบที่เรียกว่า กฎการบวก (Sum Rule)

$$p(X) = \sum_Y p(X, Y)\tag{3.16}$$

เมื่อ  $p(X)$  แทนความน่าจะเป็นของเหตุการณ์  $X$  และ  $p(X, Y)$  แทนความน่าจะเป็นที่จะมีทั้งเหตุการณ์  $X$  และเหตุการณ์  $Y$ .

กฎของการบวกนอกจากจะใช้หาค่าความน่าจะเป็นของการได้กล่องสีแดงหรือค่าความน่าจะเป็นของการได้กล่องสีฟ้าแล้ว ยังสามารถใช้หาความน่าจะเป็นของการได้ลูกบอลสีเขียวได้ โดยไม่สนใจกล่อง. สมการ 3.17 แสดงการใช้กฎของการบวกในการหาความน่าจะเป็นของการได้ลูกบอลสีเขียว. ในทำนองเดียวกัน ความน่าจะเป็นของการได้ลูกบอลสีส้ม ก็หาได้ดังแสดงในสมการ 3.18,

$$\begin{aligned}\mathbb{P}(B = 'g') &= \mathbb{P}(C = 'r', B = 'g') + \mathbb{P}(C = 'b', B = 'g') \\ &= 0.3 + 0.12 = 0.42\end{aligned}\tag{3.17}$$

$$\begin{aligned}\mathbb{P}(B = 'o') &= \mathbb{P}(C = 'r', B = 'o') + \mathbb{P}(C = 'b', B = 'o') \\ &= 0.1 + 0.48 = 0.58.\end{aligned}\tag{3.18}$$

ตาราง 3.5 สรุปค่าความน่าจะเป็นของตัวอย่างลูกบอลสีกับกล่อง.

ประเด็นที่ 3 เมื่อพิจารณาความน่าจะเป็นมีเงื่อนไข เช่น ความน่าจะเป็นที่หยอดได้ลูกบอลสีเขียวเมื่อรู้ว่ากล่องสีแดง,  $\mathbb{P}(B = 'g' | C = 'r')$ , ความหมายคือพิจารณาเฉพาะเวลาที่ได้กล่องเป็นสีแดงว่ามีโอกาสได้

ตารางที่ 3.5: สรุปค่าความน่าจะเป็นของตัวอย่างลูกบอลสีกับกล่อง

กล่อง, $C$	ลูกบอล, $B$	
	เขียว, ‘g’	ส้ม, ‘o’
แดง, ‘r’	0.3	0.1
ฟ้า, ‘g’	0.12	0.48

ลูกบอลสีเขียวเท่าไร หรือเขียนได้เป็น

$$\begin{aligned} \mathbb{P}(B = 'g' | C = 'r') &= \lim_{N \rightarrow \infty} \frac{N(B = 'g', C = 'r')}{N(C = 'r')} \\ &= \lim_{N \rightarrow \infty} \frac{N(B = 'g', C = 'r')}{N} \cdot \frac{N}{N(C = 'r')} \\ &= \mathbb{P}(B = 'g', C = 'r') \cdot \frac{1}{\mathbb{P}(C = 'r')}. \end{aligned} \quad (3.19)$$

สมการ 3.19 สอดคล้องกับ สมการ 3.13 ที่อธิบายไปก่อนหน้า. ความจริงข้อนี้สรุปอภิมาเรียกว่า กฎของการคูณ (Product Rule), นั่นคือ

$$p(X, Y) = p(Y|X) \cdot p(X). \quad (3.20)$$

เพื่อความสะดวก ความน่าจะเป็นของ  $X$  และ  $Y$  หรือ  $p(X, Y)$  อาจจะถูกเรียกว่า ความน่าจะเป็นร่วม (Joint Probability). เทอม  $p(Y|X)$  เป็นความน่าจะเป็นมีเงื่อนไข (Conditional Probability) และ  $p(X)$  บางครั้งจะเรียกว่า ความน่าจะเป็นตามขอบ (Marginal Probability).

กฎของการบวกสามารถใช้คำนวณย้อยกลับได้ว่า ถ้าลูกบอลที่ได้สีเขียว มีโอกาสมากเท่าไรที่มันจะถูกหยิบมาจากกล่องสีแดง หรือ ความน่าจะเป็นของการได้กล่องสีแดงเมื่อรู้ว่าหยิบได้ลูกบอลสีเขียว

$$\mathbb{P}(C = 'r' | B = 'g') = \frac{\mathbb{P}(C = 'r', B = 'g')}{\mathbb{P}(B = 'g')} \quad (3.21)$$

และจากการบวก จะได้ว่า

$$\mathbb{P}(B = 'g') = \sum_{c \in \{'r', 'b'\}} \mathbb{P}(C = c, B = 'g')$$

ดังนั้น

$$\begin{aligned} \mathbb{P}(C = 'r' | B = 'g') &= \frac{\mathbb{P}(C = 'r', B = 'g')}{\sum_c \mathbb{P}(C = c, B = 'g')} \\ &= \frac{0.3}{0.3 + 0.12} = \frac{0.3}{0.42} \approx 0.71. \end{aligned} \quad (3.22)$$

นอกจากนั้น สมการ 3.21 ช่วยเพิ่มแนวทางที่สามารถคำนวณไปในแนวอื่นอีกด้วย เช่น

$$\mathbb{P}(C = 'r' | B = 'g') = \frac{\mathbb{P}(B = 'g' | C = 'r') \cdot \mathbb{P}(C = 'r')}{\mathbb{P}(B = 'g')} \quad (3.23)$$

ซึ่งเรารู้ว่า  $\mathbb{P}(B = 'g' | C = 'r') = 0.75$ ,  $\mathbb{P}(C = 'r') = 0.4$  และ  $\mathbb{P}(B = 'g') = 0.42$  (จากสมการ 3.17), ดังนั้นก็จะรู้ว่า  $\mathbb{P}(C = 'r' | B = 'g') = (0.75 \cdot 0.4) / 0.42 \approx 0.71$ , ซึ่งก็สอดคล้องกับผลก่อนหน้าที่แสดงในสมการ 3.22.

### 3.3.3 ตัวอย่างปัญหาความน่าจะเป็นมีเงื่อนไข

ตัวอย่างหนึ่งที่แสดงผลของการใช้งานความน่าจะเป็นแบบมีเงื่อนไขได้เป็นอย่างดี คือตัวอย่างปัญหามอนตี้霍ล (Monty Hall Problem). สถานะการณ์คือ สมมติว่าคุณบ่าวผุดได้ไปเล่นเกมส์โชว์ ที่คุณบ่าวผุดจะต้องเลือกเปิดประตูหนึ่งในสามประตู มีประตูหนึ่งที่ซ่อนหีบสมบัติรัตนมณีไว้ อีกสองประตูซ่อนหีบของไร้ หลังจากคุณบ่าวผุดเลือกประตูไปแล้ว แทนที่พิธีกรจะเปิดประตูนั้นออกทันที พิธีกรกลับเดินไปเปิดอีกประตูให้ดูว่ามีหีบของอยู่หลังประตูนั้น และเสนอโอกาสให้คุณบ่าวผุดจะเปลี่ยนใจไปเลือกประตูที่เหลืออยู่ซึ่งเป็นประตูคุณบ่าวผุดไม่ได้เลือกแต่แรกและก็ยังไม่ถูกเปิด คุณบ่าวผุดควรจะเลือกยืนยันประตูเก่า หรือควรจะเลือกเปลี่ยนไปประตูใหม่?

ปัญหานี้ ในมุมมองของความน่าจะเป็นมีเงื่อนไขเท่ากับการหาค่าความน่าจะเป็นที่ประตูใหม่จะมีหีบสมบัติรัตนมณีอยู่ เมื่อรู้ว่าแล้วว่า ประตูหนึ่งถูกเลือกไปแล้วและอีกประตูหนึ่งถูกเปิดไปแล้ว นั่นคือ การหา  $\mathbb{P}(A = 3 | C = 1, H = 2)$ , โดย ให้  $A = 3$  แทนการที่ประตูที่สามจะมีหีบสมบัติรัตนมณีอยู่  $C = 1$  แทนคุณบ่าวผุดเลือกประตูที่หนึ่ง  $H = 2$  แทนพิธีกรเปิดประตูที่สองไปแล้ว หากรู้ว่าพิธีกรเปิดประตูที่สองแล้ววิเคราะห์ดูจะพบว่า

- ถ้าหีบสมบัติรัตนมณีอยู่ประตูที่คุณบ่าวผุดเลือก พิธีกรมีโอกาสเลือกหนึ่งในสองประตูที่เหลือ ดังนั้นในกรณีนี้ โอกาสที่พิธีกรจะเปิดประตูที่สอง คือ  $\mathbb{P}(H = 2 | C = 1, A = 1) = 0.5$ .
- พิธีกรจะไม่เปิดประตูที่มีหีบสมบัติรัตนมณีอยู่ ดังนั้นในกรณีที่หีบสมบัติรัตนมณีอยู่หลังประตูที่สอง โอกาสที่พิธีกรจะเปิดประตูที่สองคือ  $\mathbb{P}(H = 2 | C = 1, A = 2) = 0$ .
- ถ้าหีบสมบัติรัตนมณีไม่อยู่ประตูที่คุณบ่าวผุดเลือก พิธีกรต้องเปิดประตูเดียวที่เหลืออยู่ ดังนั้น โอกาสคือ  $\mathbb{P}(H = 2 | C = 1, A = 3) = 1$ .

จากกฎของการคูณ จะได้ว่า

$$\mathbb{P}(A = 3 | C = 1, H = 2) = \frac{\mathbb{P}(H = 2 | C = 1, A = 3) \cdot \mathbb{P}(A = 3, C = 1)}{\mathbb{P}(H = 2, C = 1)}$$

และเมื่อนับ จะได้ว่า  $\mathbb{P}(A = 3, C = 1) = \frac{1}{9}$  เพราะว่า มีโอกาสเป็น  $(A = 1, C = 1)$ ,  $(A = 1, C = 2)$ ,  $(A = 1, C = 3)$ ,  $(A = 2, C = 1)$ , ...,  $(A = 3, C = 3)$  แต่ละอันเท่าๆกัน (มี 9 แบบ เพราะฉะนั้น ก็ 1 ใน 9) และจากกฎการบวก จะได้ว่า

$$\begin{aligned}\mathbb{P}(H = 2, C = 1) &= \mathbb{P}(H = 2, C = 1, A = 1) + \mathbb{P}(H = 2, C = 1, A = 2) \\ &\quad + \mathbb{P}(H = 2, C = 1, A = 3) \\ &= \mathbb{P}(H = 2 | C = 1, A = 1) \cdot \mathbb{P}(C = 1, A = 1) \\ &\quad + \mathbb{P}(H = 2 | C = 1, A = 2) \cdot \mathbb{P}(C = 1, A = 2) \\ &\quad + \mathbb{P}(H = 2 | C = 1, A = 3) \cdot \mathbb{P}(C = 1, A = 3) \\ &= (0.5) \cdot \frac{1}{9} + (0) \cdot \frac{1}{9} + 1 \cdot \frac{1}{9} = \frac{1.5}{9}.\end{aligned}$$

ดังนั้นก็จะได้

$$\mathbb{P}(A = 3 | C = 1, H = 2) = \frac{\frac{1}{9}}{\frac{1.5}{9}} = \frac{2}{3}.$$

ผลที่ได้บวกว่า เมื่อพิธีกรเสนอให้คุณบัวผุดเปลี่ยนประตู ถ้าคุณบัวผุดเปลี่ยน คุณบัวผุดมีโอกาสได้หีบสมบัติ รัตนมณีเป็น  $\frac{2}{3} \approx 66.7\%$ .

เมื่อดูผลการคำนวณแล้ว หลายๆคนอาจไม่เชื่อและอาจคิดว่ามันเป็นแค่ผลจากทฤษฎี ผู้เขียนจึงใช้โค้ด ต่อไปนี้จำลองสถานะการณ์ให้เห็น และผู้เขียนสนับสนุนให้ผู้อ่านที่สนใจทดลองด้วยตนเอง. โค้ดต่อไปนี้ จำลองสถานะการณ์ทั้งหมด  $N = 100$  ครั้ง

```

1 N = 100
2 doors = c(1,2,3)
3
4 recs = matrix(0, 3, N)
5 for(i in 1:N){
6   ## which door has jewels
7   jewels = sample(doors,1)
8
9   ## which door has been chosen
10  chosen = sample(doors,1)
11
12  ## which door has the host opened
13  open = sample(rep(doors[-c(jewels, chosen)],2), 1)
14
15  recs[,i] = c(jewels, chosen, open)
16 }
17
```

```

18 ## Count how many times Rising Lotus will get the jewels without switching
19 sum(recs[2,] == recs[1,])
20
21 ## Count how many times Rising Lotus will get the jewels with switching
22 sum(recs[2,] != recs[1,])

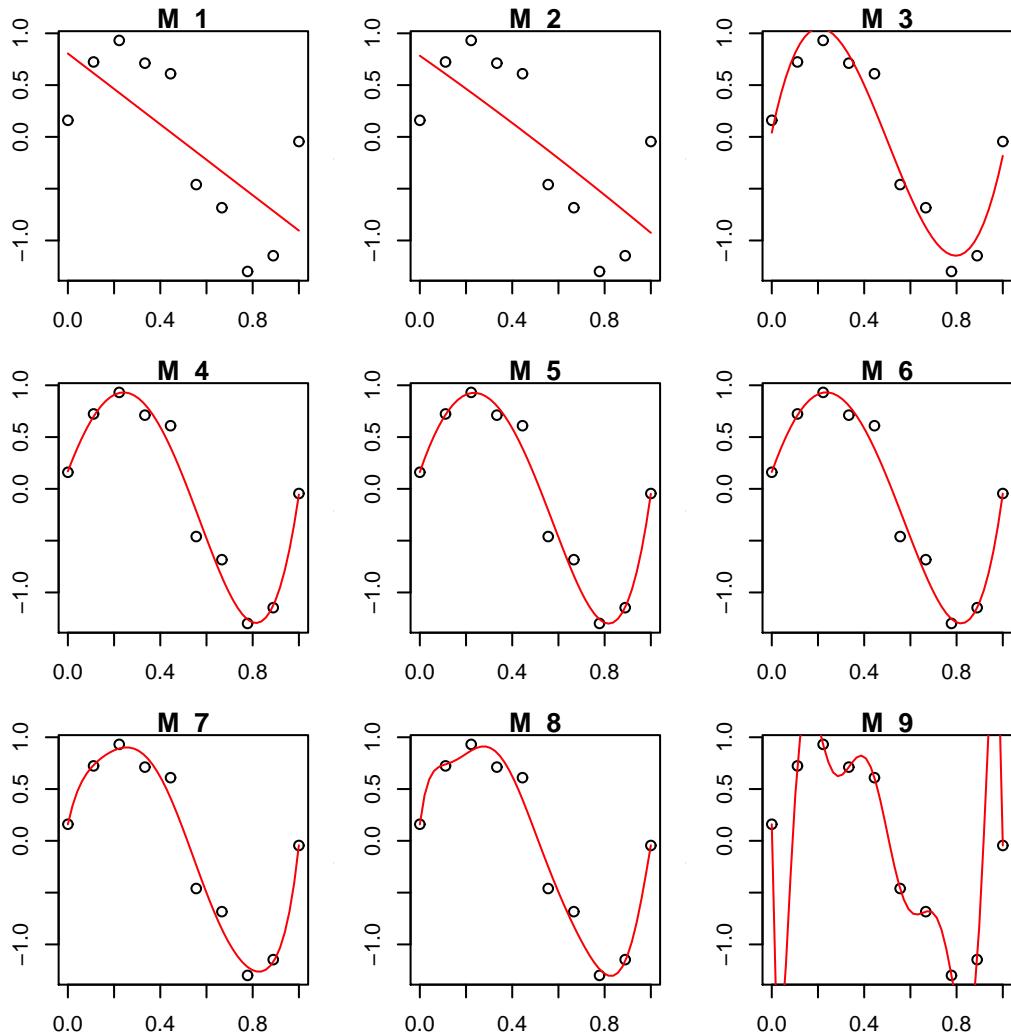
```

---

สังเกตว่า การจำลองประตูที่ซ่อนหีบสมบัติรัตนมณี `jewels = sample(doors,1)` และ ประตูที่คุณบัวผุดเลือก `chosen = sample(doors,1)` จะสุ่มตรงๆจากประตูที่มี. แต่ประตูที่พิธิกรเลือกเปิดให้ดู พิธิกรจะเลือกเปิดได้เฉพาะประตูที่ไม่มีหีบสมบัติหรือไม่ถูกเลือก ในโค้ดใช้ ตัวนี้ `-c(jewels, chosen)` เพื่อตัดประตูที่มีหีบสมบัติและประตูที่ถูกเลือกออกจากรายการประตูที่พิธิกรจะเลือกเปิดได้. เมื่อนับจำนวนครั้งที่ประตูที่คุณบัวผุดเลือก (บันทึกไว้ใน `recs[2,]`) ที่ตรงกับประตูที่มีหีบสมบัติ (บันทึกไว้ใน `recs[1,]`) และเปรียบเทียบกับจำนวนครั้งที่ประตูที่คุณบัวผุดเลือกไม่ตรงกับประตูที่มีหีบสมบัติ แปลว่า ถ้าคุณบัวผุดเปลี่ยน คุณบัวผุดจะได้สมบัติรัตนมณี. เมื่อทดลองด้วยตนเองแล้ว ผู้อ่านจะเห็นว่า ผลที่ได้จะสอดคล้องกับการคำนวณ  $\mathbb{P}(A = 3|C = 1, H = 2) = \frac{2}{3}$  และผลจะยิ่งเห็นชัดขึ้น เมื่อลองให้จำนวนครั้ง  $N$  เพิ่มขึ้น.

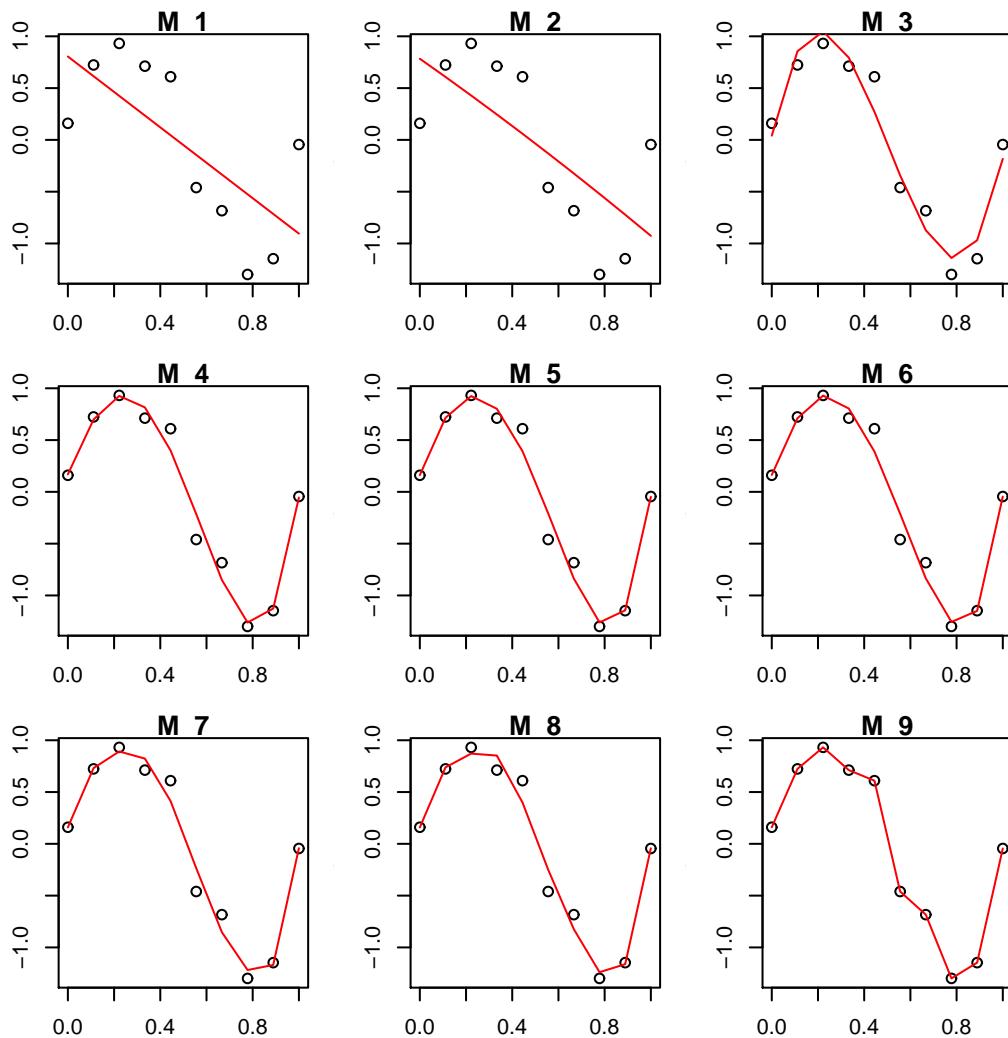
## 3.4 แบบฝึกหัด

1. จากตัวอย่างการพัฒนาที่มาของสมการฝึกฟังชั่นพหุนามอันดับหนึ่ง (ตั้งแต่สมการ 3.4, 3.5 ไปจนได้ สมการ 3.10) จงหาสมการฝึกฟังชั่นพหุนามอันดับสองและอันดับสาม (เขียนเป็นเมตริกซ์แบบเดียวกับสมการ 3.10)
2. จากตัวอย่างในกิจกรรมโปรแกรมหาค่าลดถอย จงเขียนฟังชั่นเพื่อคำนวณหาค่าพารามิเตอร์และค่าเออท์พุต สำหรับฟังชั่นพหุนามอันดับสองและอันดับสาม (ดูตัวอย่าง `w.polyfit1` และ `y.poly1`)
3. สังเกตุสมการฝึกโนเมลฟังชั่นพหุนามอันดับ 2 และ 3 ที่ได้จากข้อ 1, จงหาสมการฝึกฟังชั่นพหุนามอันดับใดๆ  $M$  โดยที่  $M$  อาจเป็นค่าใดๆ ตั้งแต่ 1 ขึ้นไป.
4. จากแบบฝึกหัดข้อ 3 จงเขียนฟังชั่นเพื่อคำนวณหาค่าพารามิเตอร์และค่าเออท์พุต สำหรับฟังชั่นพหุนามอันดับ  $M$  โดย  $M$  อาจเป็นค่าใดๆ ตั้งแต่ 1 ขึ้นไป. ตัวอย่างเช่น อาจเรียกฟังชั่นด้วย `w <- w.polyfitM(dp.x, dp.t, M=8), forecast.y <- y.poly(x, w)` เป็นต้น.
5. จากโปรแกรมที่เป็นคำตอบของแบบฝึกหัดข้อ 4 จงทดลองฝึกฟังชั่นพหุนามอันดับ 1, 2, 3, 4, 5, 6, 7, 8, และ 9 แล้วนำผลมาวัดภาพดังรูป 3.17.
6. จากแบบฝึกหัดข้อ 5 ถ้าหากทำข้อ 5 และไม่ได้ผลดังรูป 3.17 แต่กลับได้ผลดังรูป 3.18. จงอภิปรายว่า เกิดอะไรขึ้น และอภิปรายข้อดีข้อเสียเปรียบเทียบระหว่างการวัดภาพในแบบรูป 3.17 และรูป 3.18.



รูปที่ 3.17: ภาพประกอบแบบฝึกหัดข้อ 5. ผลจากโมเดลพหุนามอันดับ 1 ถึง 9 ที่ฝึกแล้ว (เส้นสีแดง)

7. จากแบบฝึกหัดข้อ 5 จงศึกษาผลที่ได้และอภิปรายว่า ถ้าจะเลือกต้องเลือกโมเดลระหว่างฟังชันพหุนามอันดับที่ 1 ถึง 9 ผู้ใช้ควรจะเลือกฟังชันพหุนามอันดับใด เพราะอะไร.
8. จงวิเคราะห์โปรแกรมในแบบฝึกหัดข้อ 4 และทดลองฝึกโมเดลด้วยอันดับที่สูงกว่า 9 ว่าเป็นอย่างไร ถ้ารันไม่ได้ เป็นเพราะอะไร อภิปราย. [คำใบ้: ถ้าโปรแกรมที่เขียนใช้ `solve` หรือวิธีการแก้สมการตริงๆ เพื่อหาค่า  $w$  ดังโดยตัวอย่างอันดับหนึ่งในหัวข้อ 3.1 แล้ว จะไม่สามารถแก้สมการได้ เมื่อจำนวนจุดข้อมูลน้อยกว่าจำนวนสมการ.]
9. ภายในฟังชัน `w.polyfit1` หรือ `w.polyfitM` ให้ทดลองเปลี่ยนจากการแก้สมการด้วย `solve` เป็นการใช้วิธีลงเกรเดียนต์ (ดูหัวข้อ 2.4) และทดลองฝึกโมเดลด้วยอันดับต่างๆ ว่าเป็นอย่างไร เปรียบเทียบกับผลที่ได้กับค่าตอบจากแบบฝึกหัดข้อ 7-8.



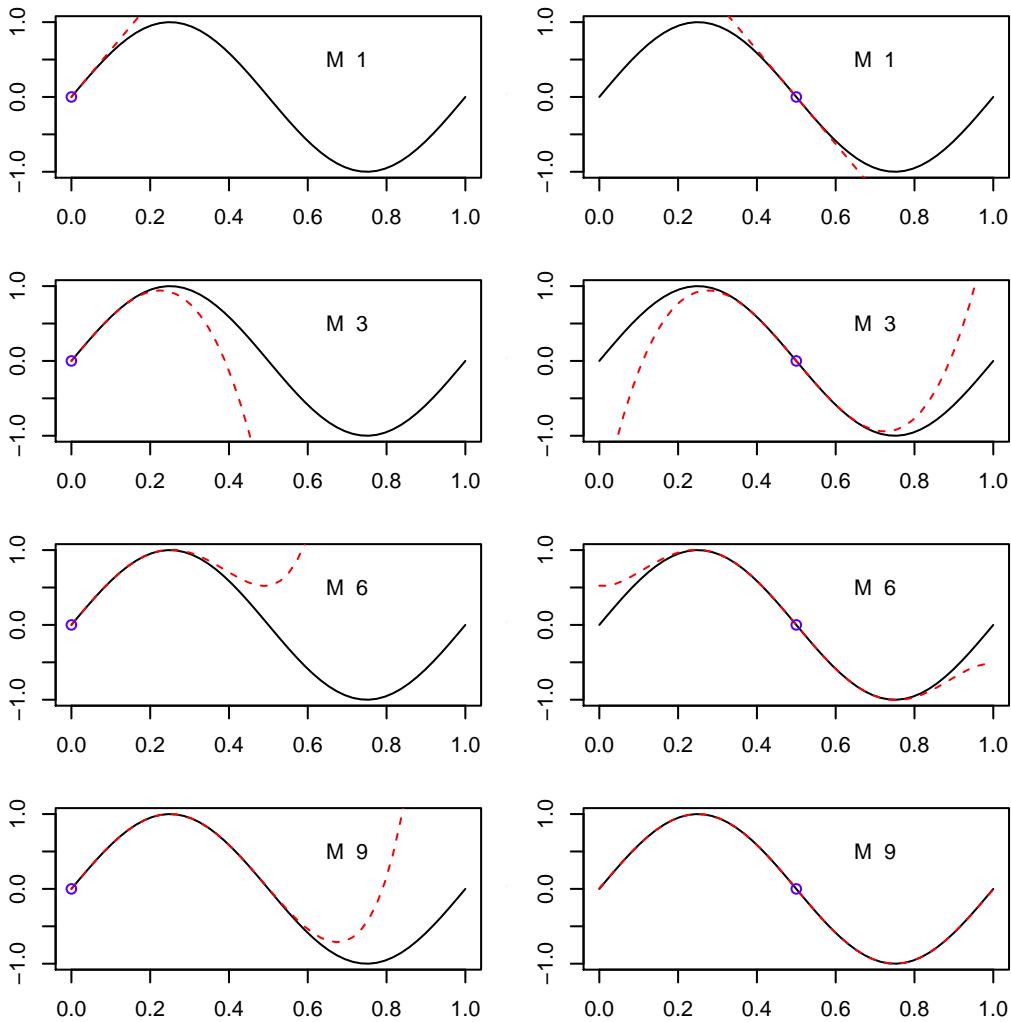
รูปที่ 3.18: ภาพประกอบแบบฝึกหัดข้อ 6. ผลจากโมเดลพหุนามอันดับ 1 ถึง 9 ที่ฝึกแล้ว (เส้นสีแดง)

10. การขยายของอนุกรม泰勒์เลอร์ (Taylor Series Expansion) กล่าวว่า ถ้า  $f(x)$  สามารถหาอนุพันธ์ได้ทุกๆ อันดับไม่จำกัด (indefinitely differentiable) ที่  $x_0$  และ

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \dots \quad (3.24)$$

จากการขยายของอนุกรม泰勒์เลอร์ จะกระจายพังชั่น  $\sin(2\pi x)$ .

11. จงนำสมการที่ได้จากข้อ 10 ไปเขียนโปรแกรมและวาดกราฟ โดยให้อนุกรมกระจายไปถึงตัวที่ 1 ถึง 9 (วาด 9 ภาพ) และ (ก) ให้  $x_0 = 0$  กับ (ข) ให้  $x_0 = 0.5$ , ดังตัวอย่างในรูป 3.19. อภิปรายสิ่งที่ได้.



รูปที่ 3.19: ตัวอย่างภาพของการกระจายเทย์เลอร์. เส้นทึบสีดำแสดงฟังชัน  $\sin(2\pi x)$ . เส้นประสีแดงแสดงการประมาณจากการกระจายเทย์เลอร์. ภาพต่างๆทางซ้ายแสดงการกระจายเทย์เลอร์ที่ใช้  $x_0 = 0$ . ภาพต่างๆทางขวาแสดงการกระจายเทย์เลอร์ที่ใช้  $x_0 = 0.5$ . ดีกรีของ การกระจายเทย์เลอร์ระบุอยู่ในแต่ละภาพ

12. จากตัวอย่างการพัฒนาจนได้สมการ 3.10 จะแสดงการหาสมการ 3.25,

$$\begin{bmatrix} N & \sum_{n=1}^N x_n \\ \sum_{n=1}^N x_n & (\sum_{n=1}^N x_n^2) + \lambda \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N t_n \\ \sum_{n=1}^N t_n \cdot x_n \end{bmatrix} \quad (3.25)$$

เมื่อใช้กรุลาเรเช่นกับพหุนามดีกรีหนึ่ง,

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{(w_0 + w_1 x_n) - t_n\}^2 + \frac{\lambda}{2} (w_1^2). \quad (3.26)$$

สังเกตุ ไม่มีการทำกรุลาเรเช่นกับ  $w_0$ .

13. ในทำนองเดียวกับข้อ 12 สำหรับพหุนามดีกรีเดียว  $M$  กับ เรศุลาเรขาชั้น จะแสดงการหาสมการ 3.27,

$$\begin{bmatrix} N & \sum_{n=1}^N x_n & \sum_{n=1}^N x_n^2 & \cdots & \sum_{n=1}^N x_n^M \\ \sum_{n=1}^N x_n & \sum_{n=1}^N x_n^2 + \lambda & \sum_{n=1}^N x_n^3 & \cdots & \sum_{n=1}^N x_n^{M+1} \\ \sum_{n=1}^N x_n^2 & \sum_{n=1}^N x_n^3 & \sum_{n=1}^N x_n^4 + \lambda & \cdots & \sum_{n=1}^N x_n^{M+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{n=1}^N x_n^M & \sum_{n=1}^N x_n^{M+1} & \sum_{n=1}^N x_n^{M+2} & \cdots & \sum_{n=1}^N x_n^{2M} + \lambda \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N t_n \\ \sum_{n=1}^N t_n \cdot x_n \\ \sum_{n=1}^N t_n \cdot x_n^2 \\ \vdots \\ \sum_{n=1}^N t_n \cdot x_n^M \end{bmatrix}. \quad (3.27)$$

14. ตามหัวข้อความน่าจะเป็น (§3.3.2) จงลองรันโปรแกรมข้างล่าง เพื่อจำลองการทดลองสำหรับ  $N = 10$ ,

---

```

1 N = 10
2 picked = matrix(0, 1, N)
3 for(i in 1:N){
4   picked[i] = sample(12,1) ## (* doing the experiment: *)
5   ## (* picking a ball from a box of 12 balls. *)
6 }
```

---

ซึ่งตัวแปร `picked` จะเก็บผลที่ได้ไว้. นำผลที่ได้มาแสดงตั้งรูป 3.14.

15. จากแบบฝึกหัดข้อ 14, เพิ่มค่า  $N$  เป็นค่าต่างๆ ดังแสดงในตาราง 3.4 และเก็บค่าอัตราส่วน  $N(A)/N$  แล้วนำมา作กราฟ ดังแสดงในรูป 3.15.

# บทที่ 4

## โมเดลเชิงเส้น

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

—Sir Isaac Newton

“ถ้าผมมองเห็นได้远ไป ก็มันก็ได้มาด้วยการยืนบนไหล่ของยักษ์”

—เชอร์ ไอแซค นิวตัน

บท 3 อภิปรายการใช้ฟังชันพหุนามในการทำนายค่าเอาร์พุต. การสร้างโมเดล เพื่อทำนายค่าของเอาร์พุตโดยที่เอาร์พุตเป็นค่าต่อเนื่อง จะเรียกว่า การหาค่าคาดถอย (Regression). โมเดลพหุนามเป็นหนึ่งในตัวอย่างของกลุ่มโมเดลที่นิยมใช้ในการหาค่าคาดถอย เรียกว่า กลุ่มโมเดลหาค่าคาดถอยเชิงเส้น (Linear Regression Model) หรือสั้นๆว่า โมเดลเชิงเส้น. คุณสมบัติที่สำคัญของโมเดลเชิงเส้น ก็คือ ค่าของเอาร์พุตของโมเดลจะมีความสัมพันธ์เชิงเส้นกับค่าพารามิเตอร์ของโมเดล. เช่นในกรณีของฟังชันพหุนาม ค่าเอาร์พุต  $y$  มีความสัมพันธ์เชิงเส้นกับ  $\mathbf{w}$  ใน  $y = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M$  เมื่อ  $x$  คืออินพุต. สังเกตว่า โมเดลเชิงเส้นมีเอาร์พุตที่มีความสัมพันธ์เชิงเส้นกับพารามิเตอร์ แต่ไม่จำเป็นว่าเอาร์พุตต้องมีความสัมพันธ์เชิงเส้นกับอินพุต. และ แทนที่ต่างๆที่คุณอยู่กับพารามิเตอร์แต่ละตัว จะเรียกว่า เบซิสฟังชัน (Basis Functions) เช่นในกรณีฟังชันพหุนาม ก็คือ  $1, x, x^2, \dots, x^M$ . นั่นคือ กำหนดให้ เบซิสฟังชัน  $\phi_m(x) = x^m$  เมื่อ  $m = 0, 1, \dots, M$ .

### 4.1 การหาค่าคาดถอยเชิงเส้น

ในกรณีที่อินพุตเป็นตัวแปรเดียว (Scalar Variable,  $x \in \mathbb{R}$ ) ฟังชันพหุนามเป็นโมเดลเชิงเส้นที่สามารถใช้ทำการหาค่าคาดถอยได้. แต่หากต้องการทำนายค่าเอาร์พุตจากอินพุต โดยที่อินพุตเป็นตัวแปรหลายมิติ (Multi-Dimensional Variable,  $\mathbf{x} \in \mathbb{R}^D, D > 1$ ) เช่น ตัวอย่างการทำนายปริมาณการ์บอนไดออกไซด์ในอากาศ ในฤดูกาล เก็บเกี่ยวอ้อย<sup>1</sup> จาก อุณหภูมิ ความเร็วลม ความชื้น และขนาดพื้นที่ของไร่ อ้อยที่มีการเพาะ

<sup>1</sup> ถึงแม้การเป็นการสร้างมูลภาวะอย่างมาก รวมถึงความเสียหายอื่น เช่น สายไฟฟ้าในบริเวณใกล้เคียง หักนิวิสัย ความเสี่ยงที่ไม่สามารถควบคุมໄฟได้ การเผาอ้อยเพื่อกำจัดภัยเป็นปัญหาด้านความปลอดภัย สิ่งแวดล้อม และสร้างปัญหาสุขภาพกับประเทศไทยอยู่.

ก่อนเก็บเกี่ยวในบริเวณใกล้เคียง. กรณีนี้ อินพุตมี 4 มิติ. นั่นคือ  $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$  สำหรับ อุณหภูมิ ความเร็วลม ความชื้น และขนาดพื้นที่ ตามลำดับ.

สำหรับกรณีที่อินพุตเป็นตัวแปร  $D$  มิติ โมเดลเชิงเส้นแบบที่ง่ายที่สุด ก็คือโมเดลการหาค่าออดถอยเชิงเส้น (Linear Regression),

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad (4.1)$$

เมื่อ  $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$ .

แทนที่จะใช้อินพุตโดยตรง เบซิสฟังชันสามารถนำมาใช้เพื่อเพิ่มความสามารถของโมเดลได้ ดังสมการที่ เป็นการรวมเชิงเส้นของเบซิสฟังชัน,

$$y(\mathbf{x}, \mathbf{y}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \quad (4.2)$$

เมื่อ  $\phi_j(\mathbf{x})$  คือ เบซิสฟังชัน. สมการนี้ใช้ค่าดัชนี  $j$  วิ่งไปถึงค่าสูงสุดที่  $M - 1$  เพื่อความสะดวกที่จะได้มี จำนวนพารามิเตอร์ทั้งหมดเป็น  $M$  ตัว. พารามิเตอร์  $w_0$  คือ ค่าอฟเซต (Offset) หรือ บางครั้งเรียกว่า ค่าไบอัส (Bias).

เพื่อความสะดวก นิยาม  $\phi_0(\mathbf{x}) = 1$  ซึ่งทำให้

$$y(\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (4.3)$$

เมื่อ  $\mathbf{w} = [w_0, \dots, w_{M-1}]^T$  และ  $\boldsymbol{\phi}(\mathbf{x})$  แทน  $[\phi_0(\mathbf{x}), \dots, \phi_{M-1}(\mathbf{x})]^T$ .

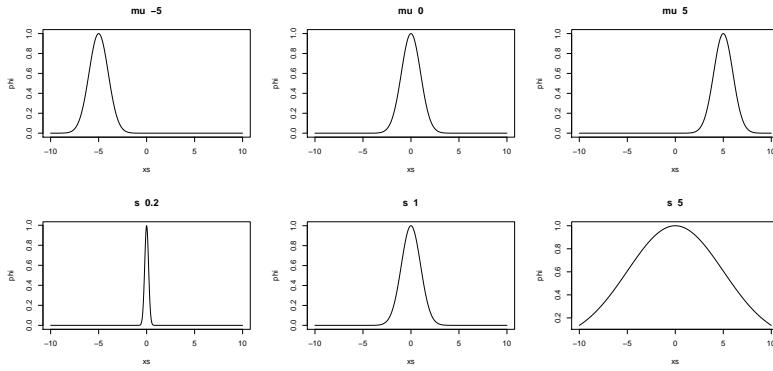
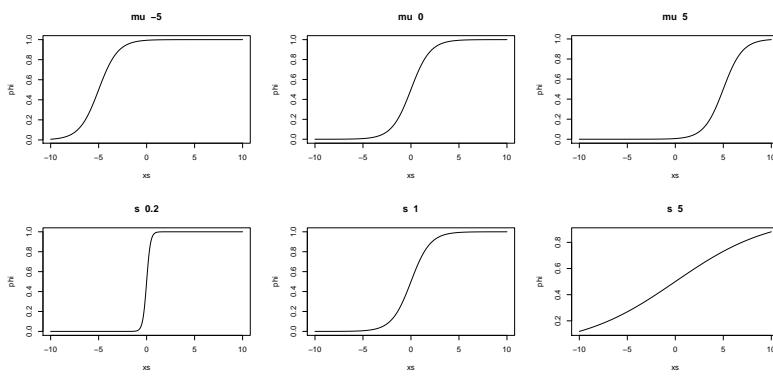
เราอาจมองเบซิสฟังชัน  $\boldsymbol{\phi}(\mathbf{x})$  ว่าเป็น ลักษณะสำคัญ (Features) ของอินพุตได้. ถ้าหากเลือกเบซิสฟังชันเป็นฟังชันไม่เป็นเชิงเส้น (Non-Linear Function) ของอินพุต เราอาจจะมีโมเดลที่เอาร์พุตมีความ สัมพันธ์ไม่เป็นเชิงเส้นกับอินพุตได้ ในขณะที่เอาร์พุตมีความสัมพันธ์เชิงเส้นกับพารามิเตอร์อยู่. การที่ เอาร์พุตมีความสัมพันธ์เชิงเส้นกับพารามิเตอร์ จะทำให้การวิเคราะห์และการฝึกโมเดลทำได้ง่าย ดังที่จะ ได้เห็นต่อไป (หัวข้อ 4.1.1)

ตามที่กล่าวไปข้างต้น ฟังชันพหุนามเป็นกรณีที่  $\phi_j(x) = x^j$ . การใช้เบซิสฟังชันแบบนี้ จะทำให้โมเดล มีคุณสมบัติเป็นฟังชันทั่วถึง (Global Function) กับอินพุต. นั่นคือ ถ้าอินพุตค่าซึ่งหนึ่งเปลี่ยนแปลง ก็จะ มีผลกระทบเบซิสฟังชันทุกๆตัว.

หากไม่ต้องการคุณสมบัติฟังชันทั่วถึง ก็อาจเลือกใช้เบซิสฟังชันที่มีลักษณะเป็นฟังชันห้องถิน เช่น การ ใช้เกลล์เชียนเบซิสฟังชัน (Gaussian Basis Function). เกลล์เชียนเบซิสฟังชันนิยามได้ดังนี้

$$\phi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2s^2}\right) \quad (4.4)$$

เมื่อ  $\mu_j$  เป็นโมเดลพารามิเตอร์ ซึ่ง  $\mu_j$  ทำหน้าที่เสมือนตำแหน่งในบริภูมิอินพุตที่ทำให้อินพุตมีผลต่อเบซิสฟังชันมากที่สุด. นั่นคือ ยิ่ง  $x$  มีค่าใกล้  $\mu_j$  เท่าไร  $\phi_j$  ก็จะตอบสนองได้ดีเท่านั้น และ  $s$  เป็นโมเดล

รูปที่ 4.1: เกส์เชียนเบชิสฟังชันที่ค่าสเกล  $s$  และตัวแหน่ง  $\mu_j$  ต่างๆรูปที่ 4.2: ซิกมอยด์เบชิสฟังชันที่ค่าสเกล  $s$  และตัวแหน่ง  $\mu_j$  ต่างๆ

พารามิเตอร์ทำหน้าที่เหมือนสเกลปรับลดขยายผลการตอบสนอง  $\phi_j$ . รูป 4.1 แสดงการตอบสนองของเบชิสฟังชันต่ออินพุต ที่  $\mu_j$  และ  $s$  ต่างๆ.

หรือ ผู้ใช้อาจเลือกซิกมอยด์เบชิสฟังชัน (Sigmoid Basis Function),

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \quad (4.5)$$

เมื่อ  $\sigma(a)$  คือซิกมอยด์ฟังชัน (Sigmoid Function) หรืออีกชื่อคือ โลจิสติกฟังชัน (Logistic Function),

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \quad (4.6)$$

รูป 4.2 แสดงการตอบสนองของซิกมอยด์เบชิสฟังชันต่ออินพุต ที่  $\mu_j$  และ  $s$  ต่างๆ. สังเกตว่า  $\mu_j$  มีผลต่อตัวแหน่งของรูปทรง. ส่วน  $s$  ความคุณสเกลหรือการยืดหดในแนวอนของรูปทรง (แต่ยังเป็นทรงตัว 'S' เมื่อเดิม).

### 4.1.1 วิธีจัดแล้วพอดีที่สุด

หากมองว่า ความสัมพันธ์ระหว่างเออต์พุต  $t$  กับอินพุต  $\mathbf{x}$  คือผลรวมจากฟังชันเชิงกำหนดกับสัญญาณรบกวน,

$$t = y(\mathbf{x}, \mathbf{w}) + \varepsilon \quad (4.7)$$

เมื่อ  $y(\mathbf{x}, \mathbf{w})$  เป็นฟังชันเชิงกำหนด (Deterministic Model) มีพารามิเตอร์  $\mathbf{w}$ , กรณีนี้เราจะใช้โมเดลเชิงเส้น (สมการ 4.3). และ  $\varepsilon$  เป็นสัญญาณรบกวนแบบเกาส์เซียนที่มีค่าเฉลี่ยเป็น 0. ดังนั้นสามารถเขียนความน่าจะเป็นแบบมีเงื่อนไข  $p(t|\mathbf{x})$  ได้ว่า

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (4.8)$$

โดยตัวแปร  $\mathbf{w}$  และ  $\beta$  เป็นพารามิเตอร์ และ  $\mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$  แทนการกระจายแบบเกาส์เซียน ซึ่งนั่นคือ

$$\mathcal{N}(\mathbf{z}|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{z}-\mu)^2\right\}$$

สำหรับ  $\mathbf{z} \in \mathbb{R}^D$ .

ถ้ามีข้อมูล  $\mathcal{D}$  ซึ่งประกอบด้วยอินพุต  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  และเออต์พุต  $\mathbf{T} = \{t_1, t_2, \dots, t_N\}$  ตามลำดับ. โดยมีสมมติฐานว่า แต่ละจุดข้อมูลเป็น i.i.d. (independent and identically distributed) แบบในสมการ 4.8, ฟังก์ชันควรจะเป็นของข้อมูลชุดนี้ คือ

$$p(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}) \quad (4.9)$$

$$= \prod_{n=1}^N \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \quad (4.10)$$

ตอนนี้สามารถใช้วิธีจัดแล้วพอดีที่สุด (Maximum Likelihood) เพื่อหาค่าพารามิเตอร์  $\mathbf{w}$  และ  $\beta$  ที่ทำให้ได้ฟังก์ชันควรจะเป็นมีค่ามากที่สุด.

ในทางปฏิบัติค่าความน่าจะเป็นของข้อมูลแต่ละจุดมีค่าน้อย และเมื่อทำการคูณความน่าจะเป็นของทุกๆ จุดเข้าด้วยกันอาจทำให้เกิดปัญหาเชิงเลข ที่ไม่สามารถแทนค่าฟังก์ชันควรจะเป็นค่าน้อยໄได้ วิธีแก้ปัญหานี้คือใช้ลอกการที่มีเข้ามาช่วย โดยคูณสมบัติของลอกการที่มีจะช่วยแก้ปัญหาการคำนวณเชิงเลขในทางปฏิบัติได้ และไม่ได้ทำให้จุดประสงค์ที่ต้องการเปลี่ยนไป ดังนั้นเมื่อใส่ลอกการที่มีเข้าไปกับสมการ 4.9 จะได้

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta) &= \sum_{n=1}^N \ln \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_{\mathcal{D}}(\mathbf{w}) \end{aligned} \quad (4.11)$$

เมื่อ

$$E_{\mathcal{D}}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2. \quad (4.12)$$

แล้วการเดียนต์ของลอกการวิที่มีฟังก์ชันควรจะเป็น เมื่อเทียบกับ  $\mathbf{w}$  ก็สามารถทำได้ว่า

$$\nabla \ln p(\mathbf{T}|\mathbf{w}, \beta) = \beta \cdot \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\} \boldsymbol{\phi}(\mathbf{x}_n)^T. \quad (4.13)$$

จากเงื่อนไขจำเป็นอันดับแรก (ดูหัวข้อ 2.2) กำหนดให้  $\nabla \ln p(\mathbf{T}|\mathbf{w}, \beta) = 0$  และจัดรูปใหม่ จะได้,

$$\mathbf{0} = \sum_{n=1}^N t_n \boldsymbol{\phi}(\mathbf{x}_n)^T - \mathbf{w}^T \cdot \sum_{n=1}^N \boldsymbol{\phi}(\mathbf{x}_n) \boldsymbol{\phi}(\mathbf{x}_n)^T. \quad (4.14)$$

ซึ่งคือได้

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{T} \quad (4.15)$$

$$\Phi = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \phi_0(\mathbf{x}_3) & \phi_1(\mathbf{x}_3) & \phi_2(\mathbf{x}_3) & \dots & \phi_{M-1}(\mathbf{x}_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}. \quad (4.16)$$

ลักษณะเดียวกัน เมื่อหาค่าของ  $\beta$  ที่ทำให้สมการ 4.11 มีค่ามากที่สุด ก็จะได้

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \{t_n - \mathbf{w}_{ML}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2. \quad (4.17)$$

สังเกตุ สมการ 4.17 คือค่าแปรปรวนของเอาต์พุต  $t_n$  เทียบกับค่าเฉลี่ยจากโมเดลเชิงเส้น.

## 4.1.2 การเรียนรู้โดยลำดับ

การหาค่าพารามิเตอร์ด้วยสมการ 4.15 และ 4.17 จะใช้ข้อมูล  $\mathcal{D}$  ทั้งหมดในการฝึกทีเดียว. ลักษณะการฝึกแบบใช้ข้อมูลทั้งหมดที่เดียวแบบนี้จะเรียกว่า เป็นการฝึกแบบกลุ่ม หรือการเรียนรู้แบบกลุ่ม (Batch Learning).

แต่ถ้าข้อมูลมีขนาดใหญ่มากๆ การเรียนรู้แบบกลุ่มนี้อาจจะมีปัญหากับการทำการคำนวณได้ เพราะเมตตริกซ์  $\Phi$  (สมการ 4.16) จะมีขนาดใหญ่มาก. นอกจากนั้น หากมีข้อมูลใหม่เพิ่มขึ้นมา เราจะต้องทำการรวมกับข้อมูลเก่า และฝึกกับข้อมูลทั้งหมดที่เดียว. วิธีที่มีประสิทธิภาพกว่า ในกรณีที่มีข้อมูลขนาดใหญ่มากๆ

หรือกรณีที่มีข้อมูลมาเพิ่ม คือ การใช้การฝึกโดยลำดับ หรือการเรียนรู้โดยลำดับ (Sequential Learning หรือ Online Learning).

การเรียนรู้โดยลำดับจะใช้จุดข้อมูลที่ละเอียดในการปรับค่าพารามิเตอร์. การเรียนรู้โดยลำดับยังเหมาะสมกับแอ�플ิเคชันตามเวลาจริง (Real-Time Applications) ที่ข้อมูลใหม่จะเข้ามาเรื่อยๆ และระหว่างนั้นก็สามารถใช้ค่าพารามิเตอร์ล่าสุดในการทำงานได้ โดยไม่ต้องรอให้ได้ข้อมูลมาครบ.

วิธีลงเกรเดียนต์ (หัวข้อ 2.1) สามารถนำมาใช้เพื่อช่วยในกระบวนการเรียนรู้โดยลำดับ ดังนี้

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha \cdot \nabla \{-\ln p(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta)\} \quad (4.18)$$

เมื่อ  $k$  คือตัวนับครั้งที่คำนวณ,  $\alpha$  คือค่าขนาดก้าว (หรือสำหรับการเรียนรู้ของเครื่อง ค่านี้มักถูกเรียกว่า อัตราการเรียนรู้, Learning Rate),  $\mathbf{X}$  และ  $\mathbf{T}$  คือข้อมูลที่จะใช้ในการคำนวณครั้งที่  $k$ .

สังเกตุ วิธีลงเกรเดียนต์ออกแบบมาสำหรับปัญหาการหาตัวทำน้อยที่สุด แต่วิธีจัดแล้วพอดีที่สุดเป็นปัญหาการหาตัวทำมากที่สุด ฉะนั้นฟังชันเป้าหมายจึงใช้  $-\ln p(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta)$  โดยเครื่องหมายลบใช้เพื่อแปลงปัญหาการหาตัวทำมากที่สุดมาเป็นปัญหาการหาตัวทำน้อยที่สุด (ดูหัวข้อ 2.1 เพิ่มเติม)

สำหรับโมเดลเชิงเส้น (ดูสมการ 4.13 เปรียบเทียบ), จะได้สมการปรับค่าพารามิเตอร์ว่า

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \alpha \cdot \left( t_n - \mathbf{w}^{(k)T} \boldsymbol{\phi}(\mathbf{x}_n) \right) \boldsymbol{\phi}(\mathbf{x}_n) \quad (4.19)$$

โดย เช่นเดียวกับวิธีลงเกรเดียนต์ ค่าอัตราการเรียนรู้จะต้องเลือกให้เหมาะสม อาทิ ไม่ใหญ่เกินไป เพื่อให้อัลกอริทึมลู่เข้า หรือ ไม่เล็กเกินไป เพื่อจะได้มีต้องทำการคำนวณหลายรอบมากเกินไป.

## 4.2 เรกูลาไรเซชัน

บทที่ 3 อภิปรายเรื่องเรกูลาไรเซชัน (Regularization) ได้บ้าง. ซึ่งเรกูลาไรเซชัน คือการควบคุมค่าของพารามิเตอร์ไม่ให้ใหญ่เกินไป เพื่อลดปัญหาโอเวอร์ฟิตติ้ง (Overfitting) โดยใช้กลไกของพื้นอตีที่เพิ่มเข้าไปในฟังชันเป้าหมาย ได้แก่ การนิยาม ฟังชันค่าผิดพลาดรวม (Total Error Function) เป็น

$$E_{total}(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

โดย  $\lambda$  คือส่วนประกอบเรกูลาไรเซชัน ซึ่งทำหน้าที่เสมอเป็นค่าน้ำหนักของการลงโทษที่ใช้ค่าของพารามิเตอร์  $\mathbf{w}$  ใหญ่เกินไป. เทอมหลัง  $E_W(\mathbf{w})$  คือพื้นอตี. เทอมหน้า  $E_D(\mathbf{w})$  คือค่าผิดพลาดจากการประมาณค่าของข้อมูล.

รูปแบบที่ง่ายที่สุดของของเรกูลาไรเซชัน คือการใช้ผลรวมของค่าพารามิเตอร์กำลังสอง,

$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}. \quad (4.20)$$

เช่นหากฟังชันค่าผิดพลาด คือ  $E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2$ , ฟังชันค่าผิดพลาดรวม ก็จะเป็น

$$E_{total}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}. \quad (4.21)$$

เรกูแลไลเซน์ในรูปแบบนี้ อาจจะถูกเรียกว่า การเลื่อนน้ำหนัก (Weight Decay) เพราะว่า กลไกนี้จะทำให้ค่าพารามิเตอร์หรือค่าน้ำหนักลดลงเข้าหาศูนย์ นอกจากจะมีผลประโยชน์จากคุณภาพการทำนายที่ยืนยันด้วยข้อมูลมาถ่วงดูลไว้. ค่าผิดพลาดของสมการ 4.21 จะเป็นฟังชันกำลังสอง (Quadratic Function) ของ  $\mathbf{w}$  ซึ่งทำให้ง่ายต่อการทำนายค่าพารามิเตอร์ที่เหมาะสม.

หลังจากแก้สมการที่อนุพันธ์ของค่าผิดพลาดรวมเป็นศูนย์เมื่อเทียบกับ  $\mathbf{w}$  จะได้ว่า

$$\mathbf{w} = (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{t}. \quad (4.22)$$

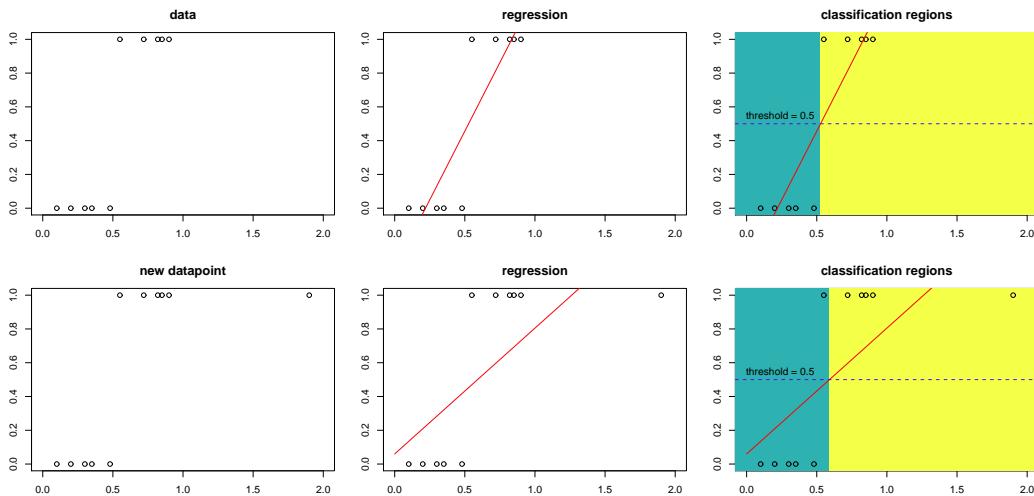
โดย  $\mathbf{I}$  คือเมตริกซ์อัตลักษณ์ (Identity Matrix).

### 4.3 การจำแนกประเภทด้วยโลจิสติกถดถอย

หัวข้อ 4.1 ยกประยุกต์โมเดลเชิงเส้นสำหรับการทำค่าถดถอย ซึ่งคือการคำนایค่าเอาร์พุทที่เป็นค่าต่อเนื่อง. หัวข้อนี้ยกประยุกต์การใช้โมเดลเชิงเส้นสำหรับงานการทำแนกกลุ่ม. ตัวอย่างเช่น งานการทำนายผู้ป่วยโรคเบาหวานจากความยาวของรอบเอวของตัวผู้ทดสอบ. สมมติว่า เรายังมีข้อมูลที่เป็นขนาดความยาวรอบเอวที่ทำนอร์มอลайซ์มาแล้ว<sup>2</sup> แทนด้วยตัวแปร  $x$  และผลการตรวจที่บอกว่าเป็นเบาหวานหรือไม่ โดยกำหนดให้ 0 แทนผลการตรวจว่า ไม่เป็นเบาหวาน หรือ ผลเป็นลบ (Negative) และกำหนดให้ 1 แทนผลการตรวจว่า เป็นเบาหวาน หรือ ผลเป็นบวก (Positive). เมื่อนำข้อมูลมาวัดกราฟโดยใช้ค่านอร์มอลายซ์ของรอบเอวเป็นแกนนอนและผลการตรวจเป็นแกนตั้ง ได้กราฟ ดังแสดงในภาพข้างบนในรูป 4.3.

เมื่อเราทำการหาค่าถดถอยด้วยโมเดลเส้นตรง  $\hat{y} = w_0 + w_1 x$  จะได้ค่าของโมเดลตั้งเส้นสีแดงที่แสดงในภาพกลางบน. ซึ่งอาจจะมองว่าค่า  $\hat{y}$  คำนایค่าของผลตรวจ ที่ค่าอินพุต  $x$ . ซึ่งก็อาจจะกำหนดว่า ถ้าค่า  $\hat{y}$  ใกล้ค่ากลุ่มใหญ่มากกว่า ก็จะให้จำแนกเป็นกลุ่มนั้น อาทิ หากค่า  $\hat{y}$  มากกว่าค่าชิดแบ่ง (Threshold) เช่น 0.5 ก็จะคำนایว่า ผลการตรวจเป็นบวก. แต่หากถ้า  $\hat{y} < 0.5$  จะคำนایว่าผลเป็นลบ. ภาพขวาบน แสดงค่าชิดแบ่ง (ที่ 0.5) เป็นเส้นประ และเมื่อ  $\hat{y}$  (เส้นทึบสีแดง) มีค่ามากกว่าค่าชิดแบ่ง เราจะคำนัยผลการตรวจเป็นบวก (แสดงด้วยพื้นที่เรางานสีเหลืองอ่อน). ทำนองคล้ายกัน เมื่อ  $\hat{y}$  มีค่าน้อยกว่าค่าชิดแบ่ง เราจะคำนัยผลการตรวจเป็นลบ (พื้นที่เรางานสีฟ้า). จะเห็นว่า จุดข้อมูลที่ผลตรวจจริงเป็นบวก (ค่าแกนตั้งเป็น 1) ทุกจุดอยู่ภายใต้เส้นพื้นที่ซึ่งถูกคำนัยว่าเป็นบวก และจุดข้อมูลที่ผลตรวจจริงเป็นลบ (ค่าแกนตั้งเป็น 0) ทุกจุดอยู่ภายใต้เส้นพื้นที่ซึ่งถูกคำนัยว่าเป็นลบ. ดังนี้ คือโมเดลเชิงเส้นสามารถทำนายได้ถูกต้องทั้งหมดในกรณีนี้.

<sup>2</sup>การนำอัตราร์มอลายซ์ (Normalization) คือ การปรับอินพุตให้อยู่ในช่วงหนามะสม, ดูหัวข้อ 6 เพิ่มเติม.



รูปที่ 4.3: ตัวอย่างแสดงปัญหาการใช้การหาค่าถดถอยมาทำการจำแนกประเภท

ตอนนี้ ถ้าเกิดว่าเราได้ข้อมูลมาเพิ่ม อีกหนึ่งจุดข้อมูล ดังแสดงในภาพช้าย่อลง (จุดที่เพิ่มมาอยู่ที่  $x = 1.9$  และผลตรวจเป็นบวก, จุดข้อมูลอยู่ที่มุมขวาบนของภาพ). หากดูจากภาพชัยย่อลง ข้อมูลที่เพิ่มเข้ามา ใหม่ก็ยังลือเข้าไปในเขตของการทำนายบวก ผลการทำนายก็ไม่น่าจะเปลี่ยนแปลงอะไร.

แต่ถ้าหากเราใช้การหาค่าถดถอย แล้วฝึกโมเดลด้วยข้อมูลทั้งหมด รวมจุดข้อมูลใหม่ที่ได้มา เราจะได้ค่าของโมเดล ดังสันสีแดงในภาพกลางย่อลง (เปรียบเทียบกับภาพกลางบน ซึ่งแสดงตอนที่ยังไม่มีจุดข้อมูลใหม่). หากใช้ค่าชี้ดีบบ์เบิร์ดเดิม ในการจำแนกประเภท จะทำให้มีจุดข้อมูลเดิมที่เคยทำนายได้ถูกต้อง แต่หลังจากได้ข้อมูลใหม่และใช้ข้อมูลใหม่นี้ประกอบในการหาค่าพารามิเตอร์ที่ถูกต้อง จุดข้อมูลนี้กลับไปอยู่ภายนอกพื้นที่สีฟ้า ซึ่งเป็นเขตที่ทำนายว่าผลการตรวจเป็นลบ ซึ่งผิด.

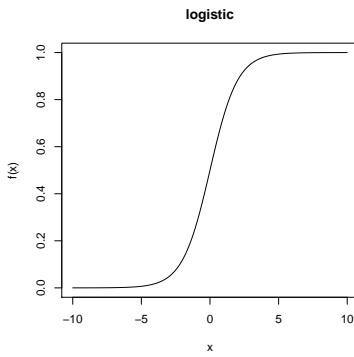
การที่มีข้อมูลเพิ่มขึ้น และตำแหน่งนั่นในปริภูมิข้อมูลของจุดข้อมูลที่เพิ่มขึ้นก็ไม่น่าจะทำให้ผลการทำงานของโมเดลแย่ลง แต่กลับทำให้ผลการทำนายแย่ลงเช่นนี้ บ่งบอกว่า การใช้โมเดลสำหรับการหาค่าถดถอยมาใช้กับงานจำแนกกลุ่ม โดยเพียงใช้กลไกของค่าชี้ดีบบ์เบิร์ดเข้ามาช่วยอาจไม่เพียงพอ.

นอกจากปัญหาข้างต้นแล้ว ผลจากการหาค่าถดถอยยังอาจมีค่ามากกว่า 1 มากๆ หรือน้อยกว่า 0 มากๆ ซึ่งในการปฏิบัติจริง จะทำให้จัดการได้ยากมาก. วิธีแก้ไขคือ แทนที่จะใช้การหาค่าถดถอยกับกลไกของชี้ดีบบ์เบิร์ด ดังตัวอย่างข้างต้น เราจะใช้ตัวช่วยซึ่งได้แก่ โลจิสติกฟังชัน (Logistic Function หรือ อีกชื่อคือ sigmoid function, Sigmoid Function),

$$h(a) = \frac{1}{1 + \exp(-a)}. \quad (4.23)$$

สำหรับงานการจำแนกประเภทระหว่างสองกลุ่ม (Binary Classification) เราสามารถทำนายค่ากลุ่ม  $y \in \{0, 1\}$  จาก

$$y = h(f(\mathbf{x}, \mathbf{w})) \quad (4.24)$$



รูปที่ 4.4: โลจิสติกฟังชัน

เมื่อ  $h(\cdot)$  คือโลจิสติกฟังชัน และ  $f(x, \mathbf{w})$  คือฟังชันหาค่าถดถอย. ฟังชันหาค่าถดถอยที่ง่ายที่สุดอันหนึ่งคือ  $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \cdot \mathbf{x}$ . วิธีนี้เรียกว่า โลจิสติกถดถอย (Logistic Regression). รูป 4.4 แสดงความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตของโลจิสติกฟังชัน. สังเกตุ ค่าของเอาต์พุตจะอยู่ระหว่าง 0 กับ 1 โดยถ้าอินพุตมีค่ามากๆ ค่าเอาต์พุตจะใกล้กับ 1. ในขณะที่ถ้าอินพุตมีค่าน้อยๆ ค่าเอาต์พุตจะใกล้กับ 0.

### 4.3.1 การฝึกโมเดลแบ่งกลุ่ม

วิธีโลจิสติกถดถอยสามารถใช้ทำนายกลุ่มของข้อมูลได้แล้ว เพียงแต่ก่อนจะใช้ โมเดลโลจิสติกถดถอยก็ต้องการค่าพารามิเตอร์ที่เหมาะสมเข่นกัน. การฝึกโมเดลโลจิสติกถดถอยก็สามารถทำได้แบบเดียวกับการฝึกโมเดลการหาค่าถดถอย นั่นคือ หา  $\mathbf{w}^* = \arg \min_{\mathbf{w}} E = \frac{1}{2} \sum_n \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2$ , โดย  $t_n \in \{0, 1\}$  คือ กลุ่มของจุดข้อมูลที่  $n$ . ฟังชันเป้าหมาย  $E = \frac{1}{2} \sum_n \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2$  สามารถเรียกว่า ฟังชันค่าใช้จ่าย (Cost Function)

$$\text{Cost}(y(\mathbf{x}, \mathbf{w}), t) = \frac{1}{2} \{y(\mathbf{x}, \mathbf{w}) - t\}^2. \quad (4.25)$$

ดังนั้น การฝึกโมเดลสามารถเขียนในรูปของฟังชันค่าใช้จ่ายได้ ดังนี้

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_n \text{Cost}(y(\mathbf{x}_n, \mathbf{w}), t_n). \quad (4.26)$$

สำหรับปัญหาการแบ่งกลุ่มสองกลุ่ม เนื่องจากค่า  $t_n$  มีค่าเป็นไปได้แค่ 2 ค่า ลักษณะพิเศษนี้สามารถนำมาใช้ประโยชน์เพื่อทำให้การฝึกโมเดลมีประสิทธิภาพมากขึ้นได้ โดยกำหนดให้

$$\text{Cost}(y, t) = \begin{cases} -\log(y) & \text{if } t = 1, \\ -\log(1 - y) & \text{if } t = 0 \end{cases} \quad (4.27)$$

โดย เพื่อความกระหึ่ด บางครั้งจะเขียน  $y$  แทน  $y(\mathbf{x}, \mathbf{w})$ .

สังเกตุ สมการ 4.27, ค่าของฟังชันค่าใช้จ่ายจะเป็น 0 เมื่อ  $t = 1$  และ  $y = 1$  หรือ  $t = 0$  และ  $y = 0$ . กล่าวง่ายๆ คือ  $\text{Cost}(y, t) \rightarrow 0$  เมื่อโมเดลจำแนกประเภทได้ถูกต้อง. ค่าของฟังชันจุดประสงค์จะเป็น  $\infty$  เมื่อ  $t = 1$  แต่  $y = 0$  หรือ เมื่อ  $t = 0$  แต่  $y = 1$ . กล่าวง่ายๆ คือ  $\text{Cost}(y, t) \rightarrow \infty$  เมื่อโมเดลจำแนกประเภทผิด.

จากสมการ 4.27, ฟังชันค่าใช้จ่ายสามารถเขียนในรูปที่กระซับขึ้นได้ ดังนี้

$$\text{Cost}(y, t) = -t \cdot \log(y) - (1-t) \cdot \log(1-y). \quad (4.28)$$

การฝึกโมเดล ซึ่งคือการแก้สมการ 4.26 ต้องการเกรเดินต์ของฟังชันค่าใช้จ่ายเทียบกับ  $\mathbf{w}$ . เพื่อความกระหัดรัด กำหนดให้  $J \equiv \text{Cost}$ ,

$$\nabla_{\mathbf{w}} J = \left[ \frac{\partial J}{\partial w_0} \quad \frac{\partial J}{\partial w_1} \quad \frac{\partial J}{\partial w_2} \quad \dots \quad \frac{\partial J}{\partial w_M} \right]^T \quad (4.29)$$

และ เมื่อแทนสมการ 4.28 หาอนุพันธ์ ทำพิชณิตและจัดรูป จนสุดท้ายจะได้ว่า

$$\frac{\partial J}{\partial w_m} = \{y - t_n\} \cdot \phi_m(\mathbf{x}_n) \quad (4.30)$$

เมื่อ  $y$  คือ ค่าที่ทำนายจากโมเดลโลจิสติกโดย  $y(\mathbf{x}_n, \mathbf{w}) = h(\mathbf{w}^T \cdot \boldsymbol{\phi}(\mathbf{x}))$ , ฟังชัน  $h(z)$  คือโลจิสติกฟังชัน (สมการ 4.23, และ  $\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x})]^T$ .

หากเลือกใช้  $\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{x}_n$ , จะได้ว่า

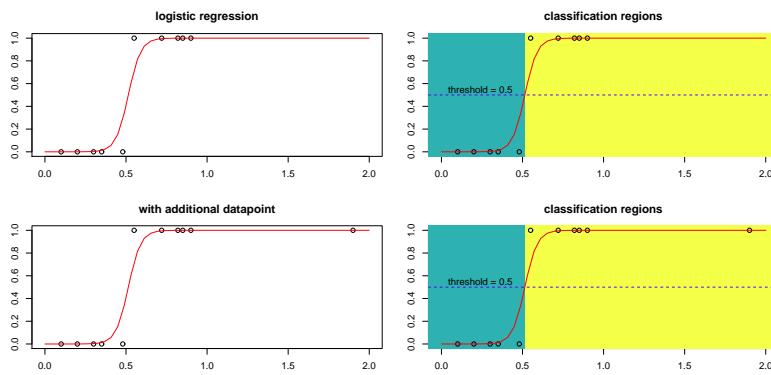
$$\frac{\partial J}{\partial w_m} = \{y(\mathbf{x}_n, \mathbf{w}) - t_n\} \cdot x_m^{(n)} \quad (4.31)$$

เมื่อ  $x_m^{(n)}$  คือ ค่าอินพุตมิติที่  $m$  ของจุดข้อมูลที่  $n$  (กล่าวอีกอย่างคือ ค่าของฐานข้อมูลที่ เรคอร์ด  $n$  พิล์ดที่  $m$ ). ด้วยเกรเดินต์ (สมการ 4.31) วิธีลงเกรเดินต์ก็สามารถนำมาใช้ในการฝึกโมเดลได้.

### 4.3.2 ตัวอย่างการจำแนกประเภท

กลับมาที่ตัวอย่างการจำแนกประเภท สำหรับปัญหาการทำนาย ผลตรวจน้ำหนาดรอบเอว. เมื่อใช้โมเดลโลจิสติกโดย จะได้ผลดังแสดงในรูป 4.5 ซึ่งจุดข้อมูลใหม่ไม่ได้ทำให้ผลการแบ่งกลุ่มเดิมที่ดีแล้วเปลี่ยนไป (เปรียบเทียบกับรูป 4.3 ที่ใช้โมเดลการหาค่าลดตอน). นอกจากนั้น เอาร์พูตจากโมเดลโลจิสติกโดยยังมีค่าอยู่ระหว่าง 0 กับ 1 ซึ่งสอดคล้องกับลักษณะงานมากกว่า และยังช่วยให้สามารถตีความในเชิงความน่าจะเป็นได้มากขึ้น.

กล่าวคือ เราอาจตีความได้ว่า เอาร์พูตจากโมเดล  $y$  คือค่าทำนายความน่าจะเป็นที่จุดข้อมูลจะอยู่ในกลุ่ม  $t = 1$  และเนื่องจากปัญหาการแบ่งกลุ่มนี้เป็นการแบ่งระหว่าง 2 กลุ่ม ความน่าจะเป็นที่จุดข้อมูลน่าจะอยู่ในกลุ่มที่  $t = 0$  จะทำนายว่าประมาณ  $1 - y$ . ดังนั้นหากความน่าจะเป็น  $y$  มีค่ามากกว่า 0.5 ก็ควรจะทายว่า จุดข้อมูลอยู่ในกลุ่ม  $t = 1$ . ทำนองเดียวกัน ถ้า  $y < 0.5$  (ความน่าจะเป็น  $t = 0$  คือ  $1 - y > 0.5$ ) ก็หมายความว่า จุดข้อมูลอยู่ในกลุ่ม  $t = 0$ .



รูปที่ 4.5: ตัวอย่างการจำแนกประเภทด้วยวิธีโลจิสติกด泊อย เปรียบเทียบกับรูป 4.3.

ตัวอย่างงานจำแนกกลุ่มข้อมูลชุดไอริส. ตัวอย่างปัญหาการจำแนกผู้ป่วยโรคเบาหวานจากขนาดรอบเอว แสดงปัญหาแบบที่อินพุตมีหนึ่งมิติ. เพื่อให้เห็นภาระงานการจำแนกกลุ่มทั่วๆไปได้ดีขึ้น พิจารณาตัวอย่างปัญหาการจำแนกสเปชีส์ดอกไม้สกุลไอริสของข้อมูลชุดไอริส (Iris Dataset<sup>3</sup>)

อนุกรมวิธาน (Taxonomy) คือวิธีการแบ่งกลุ่มของสิ่งมีชีวิตตามลักษณะร่วม. เช่น โดเมน แบ่งตามลักษณะของเซลล์ ได้แก่ โพรัคิวอตที่เซลล์ไม่มีนิวเคลียส, อาร์คีย์ที่เซลล์ไม่มีนิวเคลียส แต่มีเยื่อหุ้มเซลล์พิเศษที่ทำให้มันทดสอบสภาพแวดล้อมที่รุนแรงได้, และยุแคริวอตที่เซลล์มีนิวเคลียส. หมายเหตุ ไรวัสนันไม่ครบเป็นเซลล์ในตัวเอง และไม่จัดอยู่ใน 3 โดเมนนี้ แม้ความมีชีวิตของไรวัสเอง ก็ยังท้าทายนิยามของคำว่า “มีชีวิต”.

การจำแนกประเภทนี้ทำเป็นลักษณะของลำดับชั้น คือ โดเมน (Domain), อาณาจักร (Kingdom), ไฟลัม (Phylum) หรือ หมวด (Division) สำหรับอาณาจักรพีช, ชั้น (Class), อันดับ (Order), วงศ์ (Family), สกุล (Genus), และ สปีชีส์ (Species) รวมถึงอาจมีหมวดหมู่แยกย่อยไปอีก เช่น ชนิดย่อย (subspecies). มนุษย์จะถูกจัดอยู่ใน โดเมนยุแคริวอต, อาณาจักรสัตว์, ไฟลัมคอร์ดาตา (Chordata), ชั้น mammals (Mammalia), อันดับபְּרִימֶטָס (Primates), วงศ์ hominidae (Hominidae), สกุล homo (Homo), และ สปีชีส์ homo sapiens (Homo sapiens). ขณะที่ลิงชิมแบนซี ถูกจัดอยู่ โดเมน, อาณาจักร, ไฟลัม, ชั้น, อันดับ, และ วงศ์ เดียวกับมนุษย์ แต่ ใช้ สกุลแพน (Pan) และ สปีชีส์แพน โทรโกลไดเตส (Pan troglodytes).

ข้อมูลชุดไอริสเก็บความยาวและความกว้างของกลีบดอกและกลีบเลี้ยงของดอกไม้สกุลไอริส 3 สปีชีส์ที่มี ไอริส เชโตชา (Iris setosa), ไอริส แวร์ซิคอลเออร์ (Iris versicolor), และ ไอริส เวอร์จิニเกา (Iris virginica). หากต้องการจำแนกสปีชีส์ของดอกไม้สกุลไอริส จากความยาวและความกว้างของกลีบดอกและกลีบเลี้ยง เนื่องจากกลุ่มสปีชีส์มี 3 กลุ่ม งานลักษณะนี้จะจัดเป็น งานจำแนกกลุ่มแบบหลายกลุ่ม ซึ่งจะอธิบายรายต่อไปในหัวข้อ 4.3.3. แต่เพื่อแนะนำแนวคิดของงานจำแนกกลุ่มเบื้องต้น ณ ที่นี่ จะอธิบายการจำแนกกลุ่มระหว่างดอกที่เป็นสปีชีส์ไอริส เชโตชา กับกลุ่มอื่นที่ไม่ใช่ไอริส เชโตชา.

ถ้านำค่าความยาวของกลีบเลี้ยงและกลีบดอกจากชุดข้อมูลไอริสไปหาดจุดลงบนระนาบ โดยให้แกนนอน (x-axis) แทนความยาวของกลีบเลี้ยง (sepal length) และแกนตั้ง (y-axis) แทนความยาวกลีบดอก (petal length) จะได้ภาพดังแสดงในภาพซ้ายบนในรูป 4.6. รูป 4.6 ภาพซ้ายบน จุดข้อมูลจากสปีชีส์ไอ

<sup>3</sup>ชุดข้อมูลไอริสจาก Edgar Anderson (1935) ซึ่งเป็นข้อมูลที่มาพร้อมกับอาร์โบรเจค และสามารถเรียกใช้ได้ด้วยคำสั่ง `iris`, ดูคำอธิบายเพิ่มเติม `help(iris)`.

ริส เชโตชา (Iris setosa) เรียกเป็น กลุ่ม 1 แทนด้วยสัญลักษณ์ ‘+’. ส่วนจุดข้อมูลจากสปีชีส์อื่น ทั้งไอริส แวร์ซิคอลเออร์ (Iris versicolor) และไอริส เวอร์จินิกา (Iris virginica) เรียกเป็น กลุ่ม 0 แทนด้วยสัญลักษณ์ ‘x’ ดังระบุในภาพ.

หลังจากนำข้อมูลไปฝึกโมเดลโลจิสติกโดย เราจะได้ค่าพารามิเตอร์  $\mathbf{W}$  ที่เหมาะสม และเมื่อนำโมเดลโลจิสติกโดยที่ฝึกเสร็จแล้วไปทำนายผล ได้ผลดังแสดงในภาพบนขวา (รูป 4.6). ผลการจำแนกกลุ่มด้วยโมเดลที่ฝึกมา กลุ่ม 1 (ไอริส เชโตชา) แสดงด้วยสัญลักษณ์สีเหลือง กลุ่ม 0 (ไอริส แวร์ซิคอลเออร์หรือไอริส เวอร์จินิกา) แสดงด้วยสัญลักษณ์วงกลม. สังเกต โมเดลสามารถจำแนกกลุ่มได้ถูกต้องทั้งหมด.

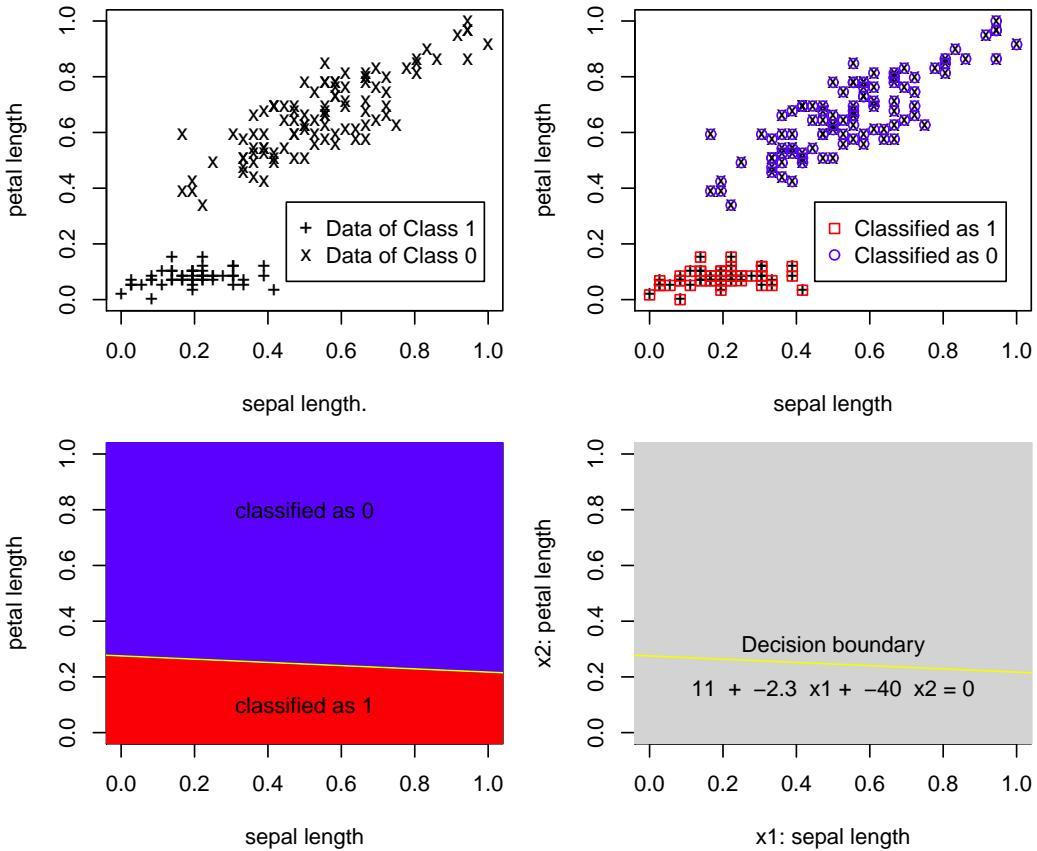
คุณสมบัติของโมเดลที่ได้ จะแบ่งปริภูมิอินพุต (Input Space) ออกเป็นพื้นที่หรือย่านที่จะทายว่าเป็นกลุ่ม 1 และย่านที่จะทายว่าเป็นกลุ่ม 0 ดังแสดงในภาพล่างซ้ายนี้ (รูป 4.6). เนื่องจากເອົາຕົວພຸດຂອງงานการจำแนกกลุ่มเป็นค่าไม่ต่อเนื่อง คุณสมบัติของโมเดลที่ใช้ในการแบ่งกลุ่มสามารถแสดงเป็นเขตบริเวณในปริภูมิอินพุตได้อย่างชัดเจน.

ย่านที่ทายกลุ่ม 1 คือปริภูมิอินพุตบริเวณที่ทำให้ค่าເອົາຕົວພຸດຂອງໂລຈິສຕິກດຄອຍ  $y > 0.5$  เมื่อ  $y = h(\mathbf{w}^T \mathbf{x})$ . ซึ่งที่  $h(\mathbf{w}^T \mathbf{x}) > 0.5$  ก็คือค่า  $\mathbf{w}^T \mathbf{x} > 0$  (ดูรูป 4.4 ประกอบ). ทำนองเดียวกัน ย่านที่ทายกลุ่ม 0 คือ  $y < 0.5$  หรือ  $\mathbf{w}^T \mathbf{x} < 0$ . ดังนั้น การใช้โมเดลในการจำแนกกลุ่ม จึงเสมือนเป็นการหาเส้นแบ่งเขตแคน ที่แบ่งระหว่างบริเวณของปริภูมิอินพุตที่จะทายว่าเป็นกลุ่ม 1 กับบริเวณของกลุ่ม 0 ซึ่งกรณีนี้ เส้นแบ่งนั้นอยู่ที่  $\mathbf{w}^T \mathbf{x} = 0$  ดังแสดงในภาพขวาล่างของรูป 4.6. เส้นแบ่งนี้จะเรียกว่า เส้นแบ่งตัดสินใจ (Decision Boundary).

สำหรับโลจิสติกโดย ที่ใช้  $\mathbf{w}^T \mathbf{x}$  ซึ่งเป็นฟังชันเชิงเส้น จะทำให้ความสามารถในการจำแนกประเภท จำกัดอยู่เฉพาะกับข้อมูลที่สามารถแบ่งกลุ่มได้ด้วยเส้นแบ่งตัดสินใจที่เป็นเส้นตรง (หรือ ระนาบสำหรับปริภูมิตอนุพัธหลายมิติ) เท่านั้น และคุณภาพการแบ่งกลุ่มจะลดลง เมื่อใช้กับข้อมูลที่มีลักษณะดังแสดงในรูป 4.7. ซึ่งปัญหานี้อาจแก้ได้โดยใช้เบซิสฟังชันที่มีลักษณะไม่เป็นเชิงเส้น (non-linear) เช่น การใช้ดีกรีที่สูงขึ้น ได้แก่แทนที่จะใช้  $y = h(w_0 + w_1x_1 + w_2x_2)$  อาจจะใช้  $y = h(w_0 + w_1x_1 + w_2x_1^2 + w_3x_2 + w_4x_2^2 + w_5x_1x_2)$  หรือ ดีกรีที่สูงขึ้นตามจำเป็น. แต่วิธีนี้นอกจากการจะไม่สะดวกแล้ว ยังทำได้ยากในทางปฏิบัติกับข้อมูลที่มีมิติสูงๆ ซึ่งสำหรับข้อมูลมิติสูงๆแล้ว การตรวจสอบดูความสัมพันธ์ระหว่างอินพุตจะทำได้ยาก. บทที่ 5 และ ?? อภิปรายถึงโมเดลที่มีประสิทธิภาพในการสร้างเส้นแบ่งตัดสินใจที่มีลักษณะไม่เป็นเชิงเส้น (Non-Linear Decision Boundary).

### 4.3.3 การจำแนกประเภทแบบหลายกลุ่ม

ตัวอย่างที่อภิปรายข้างต้นเป็นการจำแนกประเภทแบบ 2 กลุ่ม. ถ้าหากต้องการจำแนกประเภทแบบหลายกลุ่ม (Multiclass Classification) ก็สามารถทำได้โดยที่คือ แทนที่จะใช้ເອົາຕົວ 1 มิติ ( $t \in \{0, 1\}$ ) เราจะใช้ເອົາຕົວ  $K$  มิติ โดยให้  $K$  เท่ากับจำนวนกลุ่ม และจะสร้างโมเดลที่ทำให้สำหรับแต่ละอินพุต จะมีເອົາຕົວແຄມิตรีเดียวเท่านั้นที่มีค่าเป็น 1 ที่เหลือจะมีค่าเป็น 0 ซึ่งการจัดการลักษณะนี้จะเรียกว่า การเข้ารหัสหนึ่งไปเค (1-of-K Coding),  $\mathbf{t} \in \{0, 1\}^K$  และ  $\sum_{k=1}^K t_k = 1$ .



รูปที่ 4.6: เส้นแบ่งตัดสินใจ. ภาพข้างบน แสดงจุดข้อมูลจากสปีชีส์ไอริส เชโตชา (Iris setosa) เรียกเป็น กลุ่ม 1 แทนด้วยสัญลักษณ์ ‘+'. ส่วนจุดข้อมูลจากสปีชีส์อื่น ทั้งไอริส แวร์ซิคอลเออร์ (Iris versicolor) และไอริส เวอร์จินิกา (Iris virginica) เรียกเป็น กลุ่ม 0 แทนด้วยสัญลักษณ์ ‘x'. ภาพข้างบน แสดงผลการทำนายกลุ่มของแต่ละจุดข้อมูล สัญลักษณ์สีเหลี่ยม แทนการทำนาย เป็นกลุ่ม 1 (ไอริส เชโตชา) สัญลักษณ์วงกลม แทนการทำนายเป็นกลุ่ม 0 (ไอริส แวร์ซิคอลเออร์หรือไอริส เวอร์จินิกา). ภาพข้างล่าง แสดงบริเวณที่คุณสมบัติของโมเดลแบ่งบริภูมิอินพุต ออกเป็นบริเวณที่ทำนายว่าเป็นกลุ่ม 1 และบริเวณที่ทำนายว่าเป็นกลุ่ม 0. ภาพข้างล่าง แสดงเส้นแบ่งตัดสินใจระหว่างสองบริเวณ.

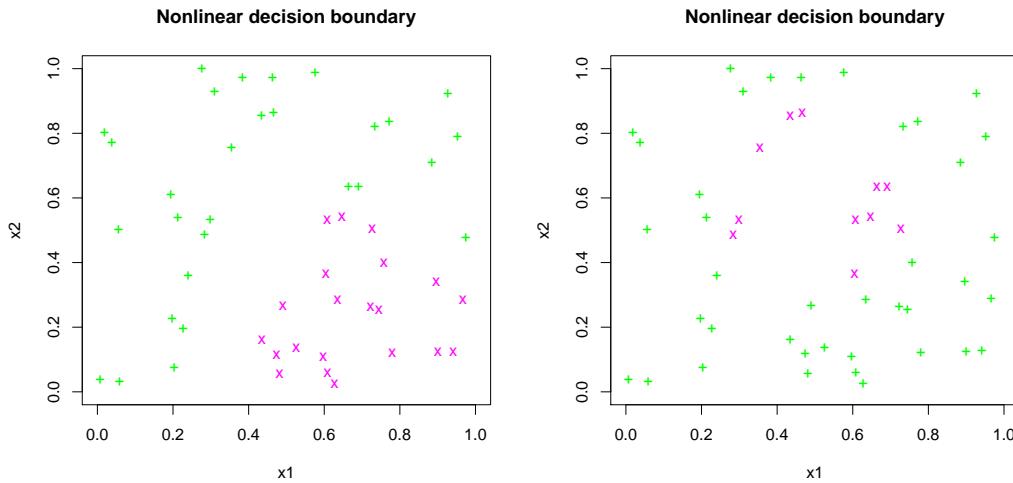
ฟังชันค่าใช้จ่ายอาจจะถูกนิยามในลักษณะเดียวกับสมการ 4.28,

$$\text{Cost}(\mathbf{y}, \mathbf{t}) = - \sum_{k=1}^K \{ t_k \ln y_k + (1 - t_k) \ln(1 - y_k) \}. \quad (4.32)$$

สมการ 4.32 อาจตีความได้ว่า คือ ลบลอกรากที่มีของค่าความควรจะเป็น โดยความควรจะเป็นของกลุ่มที่ อินพุต  $\mathbf{x}$  และโมเดล  $\mathbf{y}(\mathbf{x}, \mathbf{w})$  คือ

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^K y_k(\mathbf{x}, \mathbf{w})^{t_k} \cdot (1 - y_k(\mathbf{x}, \mathbf{w}))^{1-t_k}. \quad (4.33)$$

ตัวอย่าง ชุดข้อมูลไอริสที่มีจุดข้อมูลของความยาวและความกว้างของกลีบดอก (petal) และกลีบเลี้ยง (sepal) ดอกไม้ในสกุลไอริส อยู่ 3 สปีชีส์ ไอริส เชโตชา (Iris setosa), ไอริส แวร์ซิคอลเออร์ (Iris versicolor), และไอริส เวอร์จินิกา (Iris virginica). ภาพตัวอย่างของดอกไม้ในสกุลไอริส แสดงในรูป 4.8.



รูปที่ 4.7: ตัวอย่างแสดงชุดข้อมูลที่ต้องการเส้นแบ่งตัดสินใจที่มีลักษณะไม่เป็นเส้นตรง. สัญลักษณ์ ‘+’ แทนจุดข้อมูลของกลุ่ม 1 สัญลักษณ์ ‘x’ แทนจุดข้อมูลของกลุ่ม 0. ภาพซ้าย กรณีที่ต้องการเส้นแบ่งตัดสินใจในลักษณะเส้นตรง. ภาพขวา กรณีที่ต้องการเส้นแบ่งตัดสินใจในลักษณะเส้นงậpกาม

ข้อมูลชุดนี้มี 150 จุดข้อมูล แต่ละจุดข้อมูลมีค่าคุณลักษณะ (attributes) อยู่ 4 ค่า ได้แก่ ความยาวกลีบเลี้ยง (Sepal Length), ความกว้างกลีบเลี้ยง (Sepal Width), ความยาวกลีบดอก (Petal Length) และความกว้าง (Petal Width). และมีเฉลยເອາະພຸດຫີ່ອຈາກບອກສປີ່ສ. ข้อมูลชุดนี้มีฉลากอยู่ 3 กลุ่ม *setosa*, *versicolor*, และ *virginica*. คุณลักษณะทั้ง 4 จะใช้เป็นอินพุต ( $\mathbf{x} \in \mathbb{R}^4$ ) และฉลากບອກສປີ່ສเป็นເອາະພຸດ โดยให้ເອາະພຸດເປັນ  $t = [1 \ 0 \ 0]^T$  แทนจุดข้อมูลที่ເປັນ *setosa*,  $t = [0 \ 1 \ 0]^T$  แทนจุดข้อมูลที่ເປັນ *versicolor*, และ  $t = [0 \ 0 \ 1]^T$  แทนจุดข้อมูลที่ເປັນ *virginica*.

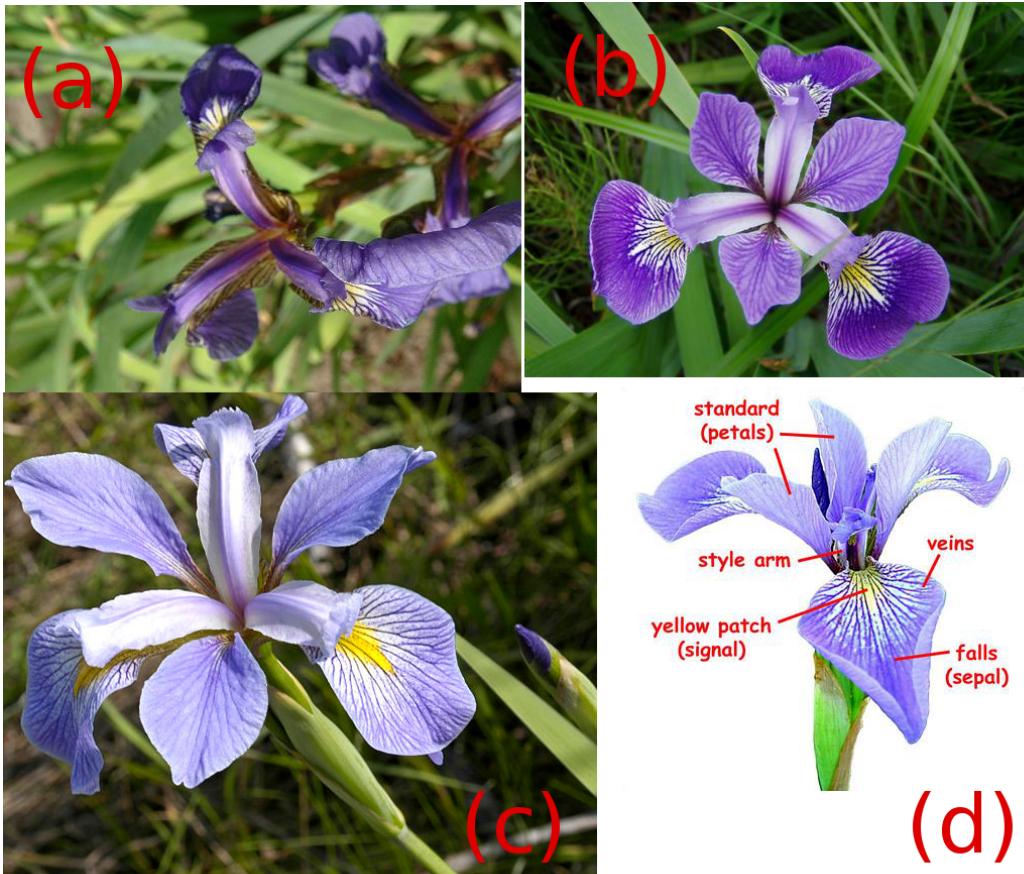
BREAK HERE

รูป 4.9 แสดงช่วงของค่าอินพุตที่มีติต่างๆ ของจุดข้อมูลทั้ง 3 กลุ่ม โดย กลุ่ม 1 แทน *setosa*, กลุ่ม 2 แทน *versicolor*, และกลุ่ม 3 แทน *virginica*. สังเกตว่าการแบ่งข้อมูลชุดนี้ จะไม่ยากนัก เนื่องจากแม้ว่าอินพุตจะเป็น 4 มิติ แต่ค่า Petal.Length หรือ Petal.Width ของจุดข้อมูลจากกลุ่มต่างๆ แยกตัวกันดีพอสมควร. โดยเฉพาะการแยกกลุ่ม 1 นั้นสามารถใช้ค่าของ Petal.Length หรือ Petal.Width ก็สามารถจำแนกจุดข้อมูลกลุ่ม 1 ออกจาก 2 กลุ่มที่เหลือได้อย่างถูกต้องแล้ว. รูป 4.10 แสดงจุดข้อมูล (ที่อยู่ในปริภูมิ 4 มิติ) เมื่อนำมาดลงบนระนาบ 2 มิติด้วยอินพุตคู่ต่างๆ. เห็นได้ชัดว่าเราสามารถใช้ระนาบเส้นตรงแยกกลุ่ม โดยเฉพาะกลุ่ม 1 ออกมาได้. แต่อย่างไรก็ตาม ตัวอย่างนี้จะใช้อินพุตทั้ง 4 มิติในการสร้างโมเดลเพื่อจำแนกประเภท.

เริ่มด้วยการฝึกโมเดล เพื่อหาค่า  $\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{n=1}^N \text{Cost}(\mathbf{y}(\mathbf{x}_n, \mathbf{w}), \mathbf{t}_n)$ . (ดูแบบฝึกหัดท้ายบทข้อ 6) สมมติค่าพารามิเตอร์ที่ได้  $\mathbf{w}^*$  คือ

>  $\mathbf{w}$

[,1] [,2] [,3]



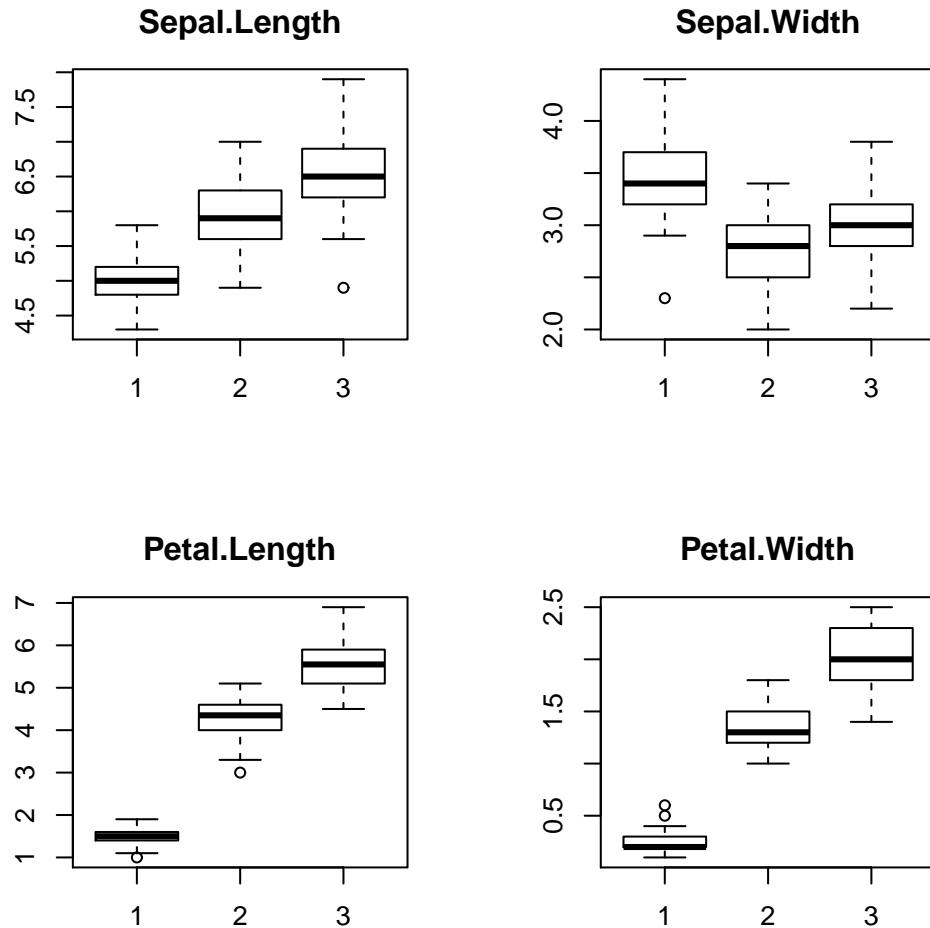
รูปที่ 4.8: ตัวอย่างของดอกไม้ในสกุลไอริส. ภาพ a ตัวอย่างดอกไอริส เชโตร์ชา. ภาพ b ตัวอย่างดอกไอริส แวร์ชีคอเลอร์. ภาพ c ตัวอย่างดอกไอริส เวอร์จินิกา. ภาพ d แสดงส่วนประกอบของดอก รวมถึงกลีบดอกและกลีบเลี้ยง. ภาพ a-c จาก Wikimedia Commons: Radomil Binek (29 May 2005, [http://en.wikipedia.org/wiki/File:Kosaciec\\_szczecinkowaty\\_Iris\\_setosa.jpg](http://en.wikipedia.org/wiki/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg)), Danielle Langlois (July 2005, [http://en.wikipedia.org/wiki/File:Iris\\_versicolor\\_3.jpg](http://en.wikipedia.org/wiki/File:Iris_versicolor_3.jpg)), Frank Mayfield (28 May 2007, [http://en.wikipedia.org/wiki/File:Iris\\_virginica.jpg](http://en.wikipedia.org/wiki/File:Iris_virginica.jpg)). ภาพ d จาก <http://www.fs.fed.us/wildflowers/beauty/iris/flowers.shtml>, วันที่ตึงข้อมูล 25 ม.ค. พ.ศ.2557)

```
[1,] 1.154165 1.0913158 -0.4060659
[2,] 0.829558 0.4565969 -2.1633640
[3,] 3.432528 -1.6508103 -1.5221858
[4,] -5.667041 0.5354391 2.9062448
[5,] -1.735660 -1.3257176 2.4796421
```

สำหรับ 3 กลุ่ม (ตามคอลัมน์ สำหรับกลุ่ม 1 กลุ่ม 2 และกลุ่ม 3 ตามลำดับ) และ 5 คุณลักษณะ ได้แก่  $[1, x_1, x_2, x_3, x_4]^T$  (ตามแล้ว สำหรับไบอัส (bias), Sepal.Length, Sepal.Width, Petal.Length, และ Petal.Width ตามลำดับ).

หลังจากได้ค่าพารามิเตอร์  $\mathbf{w}^*$  แล้ว นำค่าที่ได้มาใช้ร่วมกับโมเดลเพื่อทำนายกลุ่ม เช่น การจำแนกประเภทด้วยโลจิสติกถดถอยกับลักษณะเชิงเส้น

$$\mathbf{y} = [h(\mathbf{w}_1^T \mathbf{x}), h(\mathbf{w}_2^T \mathbf{x}), h(\mathbf{w}_3^T \mathbf{x})]^T$$



รูปที่ 4.9: แสดงช่วงของค่าอินพุตมิติต่างๆ โดยกลุ่ม 1 แทน setosa, กลุ่ม 2 แทน versicolor, และกลุ่ม 3 แทน virginica.

โดย จำกตัวอย่างข้างต้น,

$$\mathbf{w}_1 = [1.154165, 0.829558, 3.432528, -5.667041, -1.735660]^T$$

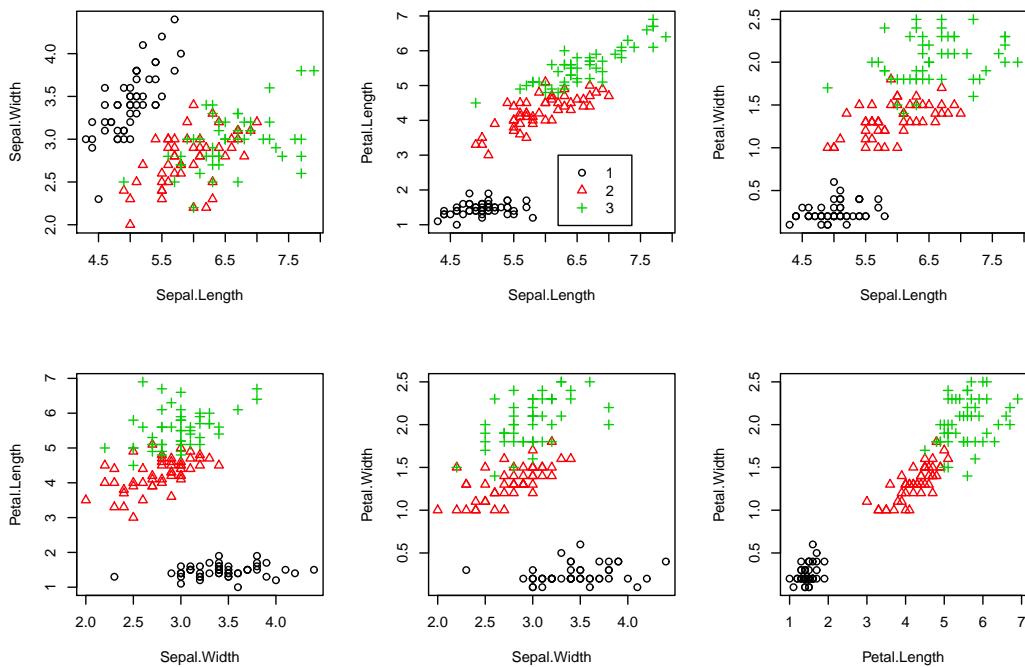
เป็นต้น. ตั้งนั้นเมื่อนำมาคำนวณกับค่า  $\mathbf{x}$  เช่น ถ้าจุดข้อมูลมีค่า

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.1	3.5	1.4	0.2

จะได้  $\mathbf{x} = [1, 5.1, 3.5, 1.4, 0.2]$ , และคำนวณได้ว่า  $y_1 = h(\mathbf{w}_1^T \mathbf{x}) = h(9.117769) = 0.9998903$  เช่นเดียวกัน จะได้ว่า  $y_2 = 0.1331482$  และ  $y_3 = 5.019370 \times 10^{-6}$  หรือกล่าวง่ายๆ คือ

$$\mathbf{y} = [0.9998903, 0.1331482, 5.019370 \times 10^{-6}]^T.$$

เมื่อเปรียบเทียบค่าทั้งสามแล้วพบว่า  $y_1$  มีค่ามากที่สุด ตั้งนั้นจะหมายว่า จุดข้อมูลจุดนี้จะอยู่ กลุ่มที่ 1. รูป 4.11 แสดงตัวอย่างผลจากการทำนายกลุ่มด้วยโมเดล. จะเห็นว่าส่วนใหญ่โมเดลทำนายได้อย่างถูก



รูปที่ 4.10: จุดข้อมูลเมื่อวัดลงบนระนาบ 2 มิติ ด้วยค่าอินพุตคู่ต่างๆ

ต้อง ซึ่งเมื่อคิดเป็นตัวเลขรวม คือ ทายถูก  $96.7\%$ <sup>4</sup>. ตาราง 4.1 แจกแจงผลการจำแนกประเภทแต่ละกลุ่ม ด้วยเมตริกซ์สับสน (Confusion Matrix). จะเห็นว่า โมเดลสามารถจำแนกจุดข้อมูลจากกลุ่ม 1 ได้ถูกต้อง สมบูรณ์ ในขณะที่ยังสับสนกับข้อมูลจากกลุ่ม 2 และ 3 บ้าง. นั่นคือ มีจุดข้อมูลจริงอยู่กลุ่มที่ 2 แล้วทายผิด ว่าเป็นกลุ่ม 3 อยู่ที่  $8\%$ . การแจกแจงเช่นนี้ช่วยให้เข้าใจความเสี่ยงของการใช้โมเดลได้ดีขึ้น โดยเฉพาะ การนำไปใช้งานกับข้อมูลที่ละเอียดอ่อน เช่น แอ��哀ดิเคชันด้านการแพทย์ เป็นต้น.

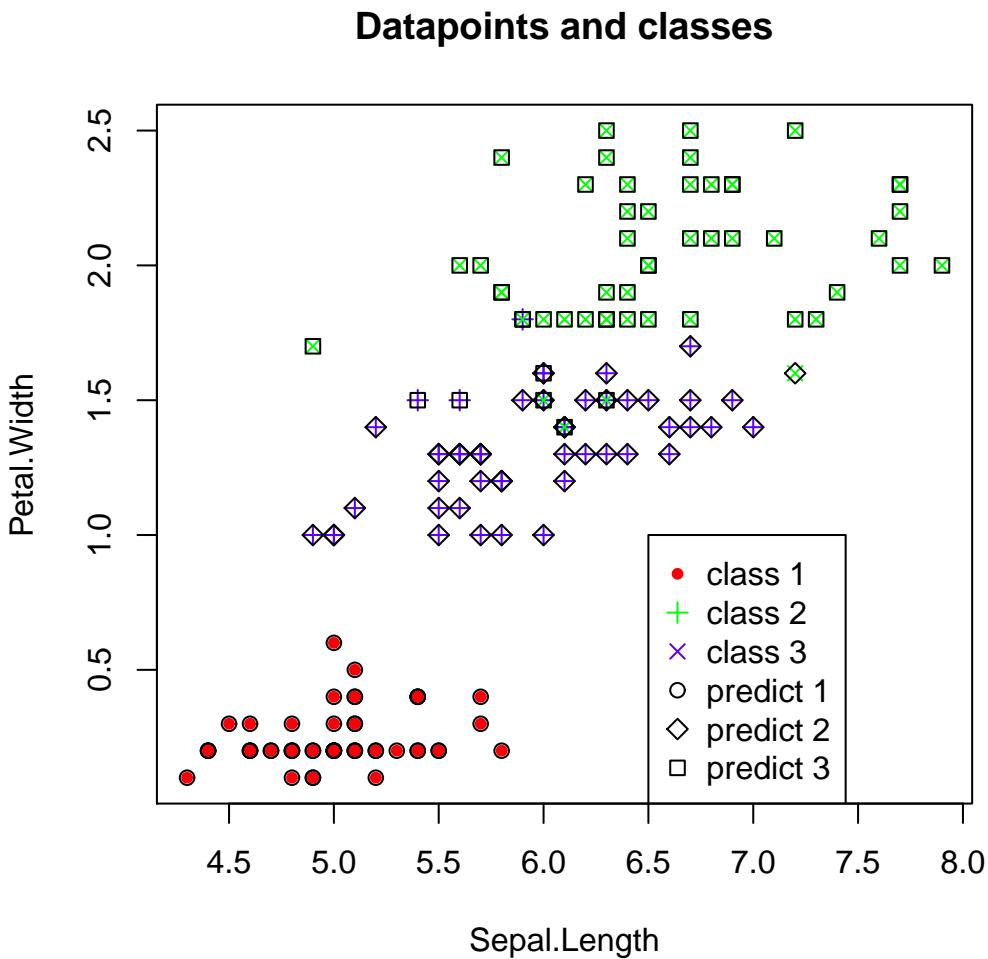
ตารางที่ 4.1: เมตริกซ์สับสน. ผลการจำแนกประเภทชุดข้อมูลไหร่สำคัญแต่ละกลุ่ม

กลุ่มจริง	ทายเป็นกลุ่ม 1	ทายเป็นกลุ่ม 2	ทายเป็นกลุ่ม 3
1	100%	0%	0%
2	0%	92%	8%
3	0%	2 %	98%

## 4.4 แบบฝึกหัด

1. จงเขียนโปรแกรมเพื่อนำสมการ 4.15 และ 4.17 ไปปฏิบัติ และสาธิตการใช้งาน.

<sup>4</sup>การประเมินผล โมเดลที่ถูกต้องจะต้องทำการทดสอบกับข้อมูลที่ไม่ได้ถูกใช้ในการฝึกโมเดล (ดูหัวข้อ 3.2) แต่ผลที่แสดงในหัวข้อนี้ มีจุดประสงค์เพื่อประกอบคำอธิบายเรื่องวิธีการจำแนกประเภท และไม่ได้ทดสอบกับข้อมูลอีกชุด.



รูปที่ 4.11: แสดงจุดข้อมูลของชุดข้อมูลไอิริสบนระนาบ 2 มิติ โดย แกนนอนแทน Sepal.Length และแกนตั้งแทน Petal.Width. กลุ่มจริง 1 ถึง 3 (class 1-3) และกลุ่มที่ทำนาย (predict 1-3) ใช้สัญลักษณ์ตามระบุในภาพ. หมายเหตุ การทำนายใช้ข้อมูลครบทั้ง 4 มิติ แต่เพื่อความสะดวกจึงเลือกแสดงบนระนาบ 2 มิติ

2. จงเขียนโปรแกรมเพื่อนำสมการ 4.19 ไปปฏิบัติ สาธิตการใช้งาน และเปรียบเทียบกับแบบฝึกหัดข้อ 1.
3. จงแสดงในเห็นว่าค่า  $\mathbf{w}$  ในสมการ 4.22 จะทำให้ค่าผิดพลาดรวม (สมการ 4.21) มีค่าน้อยที่สุด.
4. จากตัวอย่างปัญหาการจำแนกกลุ่มของคนเป็นโรคเบาหวานจากขนาดรอบเอว โดยมีข้อมูล ดังแสดงในตาราง 4.2, N แทนผลลบ และ P แทนผลบวก. จงทดลองใช้โมเดลโลจิสติกถดถอยในการจำแนกกลุ่ม และทดลองทำอีกรังสึมีข้อมูลเพิ่ม คือ  $(x, y) = (1.90, P)$ . เปรียบเทียบผลที่ได้กับรูป 4.5.
5. จงหาอนุพันธ์ของฟังชันต่อไปนี้
  - $f(x) = 1 / \{1 + \exp(-x)\}$  และแสดงให้เห็นว่า  $f'(x) = \{1 - f(x)\} \cdot f(x)$ .

ตารางที่ 4.2: ข้อมูลปัญหาขนาดรอบเวลาที่บันทึกไว้ในตารางนี้ แสดงค่าของตัวแปรต่างๆ ที่ได้รับการตรวจสอบเบ้าหวาน

x	0.10	0.21	0.31	0.35	0.48	0.55	0.85	0.72	0.82	0.90
y	N	N	N	N	N	P	P	P	P	P

- $f(x, y) = x \cdot \log(y)$  เทียบกับ  $y$ .
- แสดงว่ากราฟเดียนต์ของสมการ 4.28 เทียบกับ  $w_m$  คือ สมการ 4.30.

คำใบ้

- $\frac{d \log(x)}{dx} = \frac{1}{x}$ .
- ถ้า  $y$  เป็นฟังชันของ  $x$  และ  $f$  เป็นฟังชันของ  $y$ , และจากกฎลูกโซ่ (Chain Rule) ได้ว่า  $\frac{df}{dx} = \frac{df}{dy} \cdot \frac{dy}{dx}$ .

6. สำหรับฟังชันลักษณะใดๆ  $\phi_m(\mathbf{x})$ , เช่น  $\phi_m(\mathbf{x}) = x_m$  สำหรับลักษณะเชิงเส้นดิรีหิ่ง หรือ  $\phi_m(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x}-\mu_m)^2}{\sigma_m^2}\right)$  สำหรับลักษณะแบบเกาส์เชี่ยน, จงหาอนุพันธ์,

$$\frac{\partial C}{\partial w_{mk}} = \frac{\partial \sum_{n=1}^N C_n}{\partial w_{mk}}$$

เมื่อ  $C_n \equiv \text{Cost}(\mathbf{y}(\mathbf{x}_n, \mathbf{w}), \mathbf{t}_n)$  นิยามดังสมการ 4.32 และ  $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_K]^T$  โดย  $y_k = h(\mathbf{w}_k^T \phi) = 1/(1 + \exp(\mathbf{w}_k^T \phi))$ ;  $\phi = [\phi_0(\mathbf{x}) \ \phi_1(\mathbf{x}) \ \cdots \ \phi_M(\mathbf{x})]^T$ ;  $\mathbf{w}_k = [w_{0k} \ w_{1k} \ \cdots \ w_{Mk}]^T$  และ จงแสดงให้เห็นว่า

$$\begin{aligned} \frac{\partial C}{\partial w_{km}} &= - \sum_{n=1}^N \left\{ \frac{t_{nk}}{y_{nk}} - \frac{(1-t_{nk})}{(1-y_{nk})} \right\} \cdot \frac{\partial y_{nk}}{\partial w_{km}} \\ &= - \sum_{n=1}^N \{t_{nk}(1-y_{nk}) - (1-t_{nk})y_{nk}\} \cdot \phi_m(\mathbf{x}_n) \\ &= - \sum_{n=1}^N \{y_{nk} - t_{nk}\} \cdot \phi_m(\mathbf{x}_n) \end{aligned} \quad (4.34)$$

เมื่อ

$$\frac{\partial y_{nk}}{\partial w_{km}} = \{1 - h(\mathbf{w}_k^T \phi)\} \cdot h(\mathbf{w}_k^T \phi) \cdot \phi_m(\mathbf{x}_n).$$

7. จงเขียนโปรแกรมเพื่อสร้างโมเดลจำแนกประเภทชุดข้อมูลไอรีส โดยใช้โมเดลโลจิสติกดถอยกับ โมเดล เชิงเส้นดีกรีหนึ่ง  $\phi_m(\mathbf{x}) = x_m$  และทดสอบโมเดลที่ได้กับชุดข้อมูลไอรีส เปรียบเทียบกับผลที่แสดงในหัวข้อ 4.3.3.

8. จงแบ่งข้อมูลเป็นชุดฝึกหัดและชุดทดสอบก่อน แล้วจงใช้วิธีโลจิสติกดถอยกับโมเดลเชิงเส้นดีกรีหนึ่ง ในการจำแนกประเภท พิรุณประเมินประสิทธิภาพของโมเดล.

9. จงใช้โลจิสติกดถอยกับโมเดลเชิงเส้นดีกรีหนึ่ง ในการจำแนกประเภทของชุดข้อมูลไวน์ โดยประเมินผลการทำงานด้วยวิธีกรอสวาร์ลีเดชัน (ดูหัวข้อ 3.2)

# บทที่ 5

## โครงข่ายประสาทเทียม

“Alone we can do so little. Together we can do so much.”

— Helen Keller

“ลำพังเราทำได้น้อยนิด รวมกันเราทำได้มากน่าย”

— เฮเลน เคลเลอร์

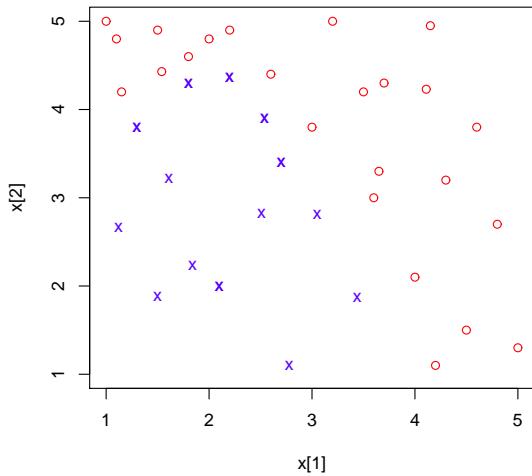
บทที่ 4 อภิปรายโมเดลเชิงเส้น สำหรับการหาค่าคาดถอยและการจำแนกประเภท. โมเดลเชิงเส้นกับเบซิสฟังชันแบบพหุนาม ถึงแม้จะมีคุณสมบัติดีๆ หลายอย่าง เช่น ค่าพารามิเตอร์ของโมเดลเหล่านี้สามารถหาได้ง่าย แต่การนำไปใช้งานจริงกลับมีข้อจำกัดมาก โดยเฉพาะอย่างยิ่งในเรื่องของคำสาปของมิติ.

คำสาปของมิติ (Curse of Dimensionality) กล่าวถึง ปัญหาของคุณสมบัติการขยาย (scalability) ที่เมื่อตัวแปรมีมิติสูงขึ้น ปริมาณการคำนวณที่ต้องการจะมีเพิ่มขึ้นอย่างมากตามหาศalarm ขึ้นความยุ่งยากอื่นๆ. ตัวอย่างเช่น การแบ่งกลุ่ม หากเลือกใช้ดีกรีหนึ่ง เช่น  $\phi(\mathbf{x} = [x_1, x_2]^T) = [x_1, x_2]^T$  เส้นแบ่งตัดสินใจ(decision boundary) จะเป็นแค่เส้นตรงธรรมดา ซึ่งหาก ข้อมูลมีความสัมพันธ์ระหว่างมิติ ดังแสดงในรูป 5.1 หากต้องการเส้นแบ่งตัดสินใจที่ดีหยุ่นกว่าเส้นตรง เราอาจจะใช้เบซิสฟังชันดีกรีที่สูงขึ้น เช่น  $\phi(\mathbf{x} = [x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1 \cdot x_2, x_2^2]^T$  หรือดีกรีสาม  $\phi(\mathbf{x} = [x_1, x_2]^T) = [x_1, x_2, x_1^2, x_1 \cdot x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3]^T$ . สังเกต ถ้าข้อมูลที่มิติมากขึ้น เช่น  $\phi([x_1, x_2, x_3]^T) = [x_1, x_2, x_3, x_1^2, x_1 x_2, x_1 x_3, x_2^2, x_2 x_3, x_3^2, x_1^3, x_1 x_2^2, x_1 x_2 x_3, x_1^2 x_2, \dots, x_3^3]^T$  แล้วจินตนาการถึงการทำดีกรีที่สูงขึ้นไปอีก หรือ เมื่อข้อมูลมีมิติที่มากขึ้นอีก ความซับซ้อนของสมการจะเพิ่มขึ้นไปอย่างมากมากจากจะจัดการได้.

แนวทางหนึ่งที่จะช่วยบรรเทาปัญหาคำสาปของมิติได้บ้าง ก็คือการเลือกใช้เบซิสฟังชันที่ปรับตัวเองได้. กล่าวคือ เบซิสฟังชันก็มีพารามิเตอร์ที่ปรับตัวเองได้. โมเดลที่เป็นที่รู้จักดีในแนวคิดนี้คือโครงข่ายประสาทเทียม ซึ่งที่ชนิดที่นิยมก็คือ เพอร์เซปตรอนหลายชั้น.

### 5.1 เพอร์เซปตรอนหลายชั้น

แนวคิดของเพอร์เซปตรอนหลายชั้น (Multi-Layer Perceptron ตัวย่อ MLP) เริ่มมาจาก การพยายามสร้างระบบที่เลียนแบบการทำงานของเซลล์ประสาทของสิ่งมีชีวิต (รูป 5.2 แสดงโครงสร้างเซลล์ประสาททั่วๆไป).



รูปที่ 5.1: ตัวอย่างข้อมูลที่เส้นแบ่งตัดสินใจเกินดีกรีเชิงเส้น จุดข้อมูลจากกลุ่ม 1 แสดงด้วยวงกลม จุดข้อมูลจากกลุ่มที่ 2 แสดงด้วยกาบท

ปี 1957 แฟรงค์ โรเซนแบล็ท (Frank Rosenblatt) ได้สาธิตการทำงานของเซลล์ประสาทจำลอง ด้วยเครื่องคอมพิวเตอร์ และ โรเซนแบล็ทเรียกเซลล์ประสาทจำลองนั้นว่า เพอร์เซปตรอน (perceptron). แนวคิดของเพอร์เซปตรอน คือการใช้หน่วยคำนวณง่ายๆ หลายๆ หน่วย ต่อกันเป็นโครงข่าย และผลรวมของมันสามารถทำการคำนวณที่ซับซ้อนได้.

รูป 5.3 แสดงโครงสร้างของเพอร์เซปตรอน. การคำนวณของเพอร์เซปตรอน ก็คือจะนำเอาอินพุตแต่ละตัวไปคูณกับค่าน้ำหนักของอินพุตนั้นๆ และนำค่าผลคูณทั้งหมดมาบวกกัน แล้วหากผลบวกมีค่ามากพอ นั่นคือเกินค่าระดับกระตุน (threshold) เพอร์เซปตรอนจะอยู่ในสถานะถูกกระตุน (ให้อาต์พุตเป็น 1) แต่หากผลบวกมีค่าต่ำกว่าระดับกระตุน เพอร์เซปตรอนจะอยู่ในสถานะไม่ถูกกระตุน (ให้อาต์พุตเป็น 0). ดังนั้นอาต์พุตของเพอร์เซปตรอน สามารถเขียนเป็นสมการได้ว่า

$$y = \begin{cases} 0 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D < \tau \\ 1 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D \geq \tau \end{cases}$$

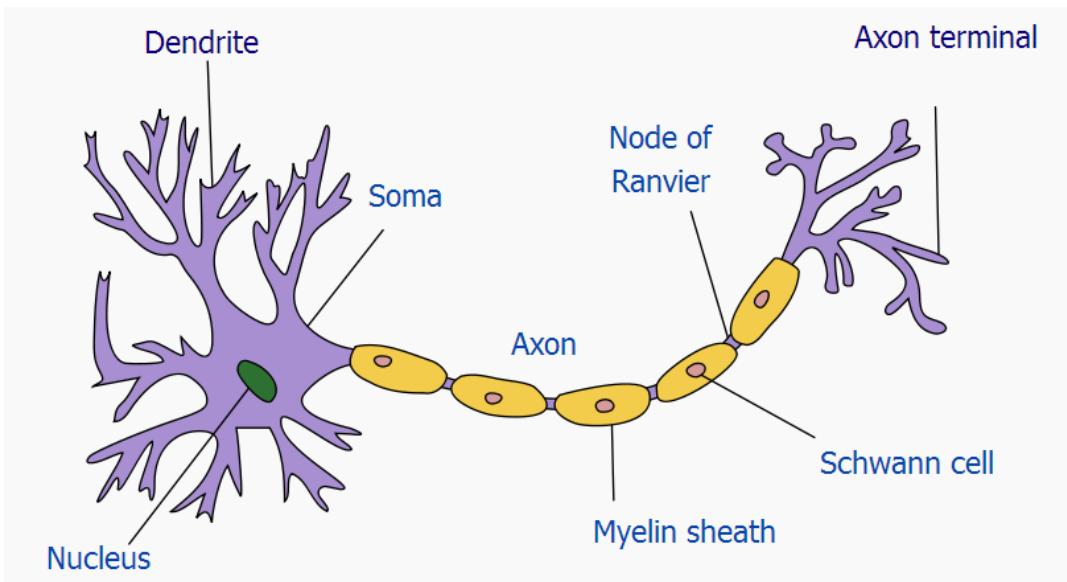
เมื่อ  $w_1, \dots, w_D$  เป็นน้ำหนักของอินพุต  $x_1, \dots, x_D$  ตามลำดับ,  $\tau$  คือ ค่าระดับกระตุน, และ ให้ 1 แทนค่าแสดงสถานะถูกกระตุน และ 0 แทน สถานะไม่ถูกกระตุน.

เพื่อความสะดวก เรานิยาม ไบอัส (bias) เป็น  $b = -\tau$  และ เราจะได้

$$y = \begin{cases} 0 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b < 0 \\ 1 & \text{เมื่อ } w_1x_1 + \dots + w_Dx_D + b \geq 0 \end{cases}$$

ซึ่งทำให้เราเขียนได้เป็น

$$y = f \left( \sum_i w_i x_i + b \right) \quad (5.1)$$



รูปที่ 5.2: รูปแสดงโครงสร้างของเซลล์ประสาททั่วๆไป เดนไดร็ต (dendrite) ทำหน้าที่ส่งผ่านอินพุตของเซลล์ และ ออกซอน (axon) ทำหน้าที่ส่งผ่านกับเอาต์พุตของเซลล์ (ภาพจาก [http://en.wikipedia.org/wiki/File:Neuron\\_Hand-tuned.svg](http://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg))

เมื่อ เรานิยามฟังชัน  $f(\cdot)$  เป็นฟังชันจำกัดแข็ง (hard limit function),

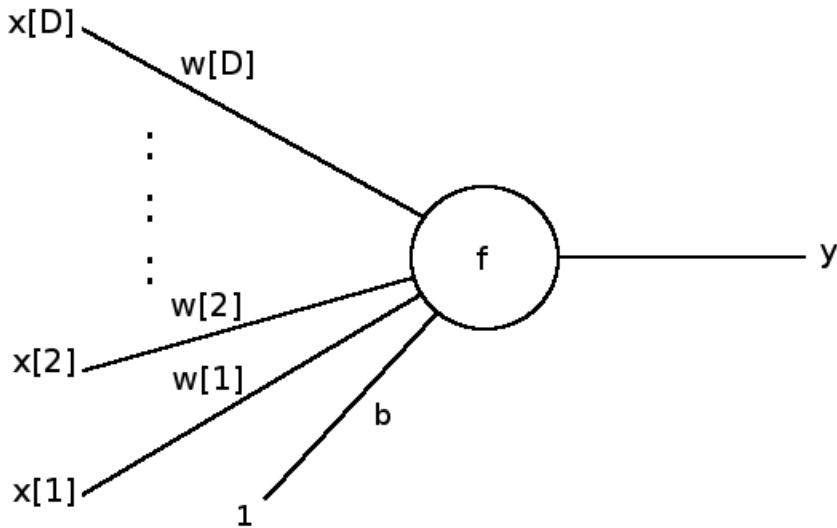
$$f(a) = \begin{cases} 0 & \text{เมื่อ } a < 0, \\ 1 & \text{เมื่อ } a \geq 0. \end{cases} \quad (5.2)$$

เนื่องจากฟังชันนี้จะให้ค่าสถานะการกระตุ้นของเพอร์เซปตอรอน ฟังชันนี้จึงมักถูกเรียกว่า ฟังชันการกระตุ้น (activation function).

สมการ 5.1 จะเข้ากับ แผนภาพในรูป 5.3 โดยอินพุต 1,  $x[1], \dots, x[D]$  จะคูณกับหนึ่งตัว  $b$ ,  $w[1], \dots, w[D]$  ตามลำดับ และ ผลคูณจะถูกนำมารวมกัน ก่อนที่จะผ่านไปเข้าฟังชันการกระตุ้น  $f$  ที่จะให้ค่าเอาต์พุต  $y$  ออกมานะ.

ในตอนนั้น งานของโรเซนแบล็ททำให้วิเคราะห์โดยเฉพาะอย่างยิ่งการปัญญาประดิษฐ์ตื่นเต้นมาก ที่เราจะสามารถสร้างเครื่องจักรที่สามารถเลี่ยบแบบการทำงานของสมองมนุษย์ได้ เกิดการคาดการณ์ถึงศักยภาพ ความสามารถต่างๆ ที่เครื่องคอมพิวเตอร์จะสามารถทำได้. แต่ความฝันและความหวังก็ล่มสลายไป หลังจาก มาร์вин มินสกี้ (Marvin Minsky) และ เซมวอร์ ปาเปิต (Seymour Papert) ได้ร่วมกันเขียนหนังสือเพอร์เซปตอรอนส์[52] ที่วิเคราะห์โครงสร้างและการทำงานของเพอร์เซปตอรอน. ประเด็นสำคัญของหนังสือ คือ มินสกี้และปาเปิตกล่าวว่าเพอร์เซปตอรอนนั้นสามารถทำได้แต่งานง่ายๆ เช่นหากเป็นงานการจำแนกประเภท ก็สามารถทำงานได้กับปัญหาที่สามารถแบ่งได้ด้วยเส้นแบ่งตัดสินใจเชิงเส้นเท่านั้น ไม่สามารถทำงานที่ซับซ้อนกว่านั้นได้. พ้อมทั้งยังยกตัวอย่าง การทำงานของตระรกะ เอ็กซ์-ออร์ (XOR) หรือ exclusive OR ที่เพอร์เซปตอรอนไม่สามารถเลี่ยบแบบได้. ตาราง 5.1 แสดงพฤติกรรมของตระรกะเอ็กซ์-ออร์.

ผลงานหนังสือเพอร์เซปตอรอนส์ นอกจากจะทำให้เพอร์เซปตอรอนเสื่อมความสนใจแล้ว ยังทำให้เทคนิค



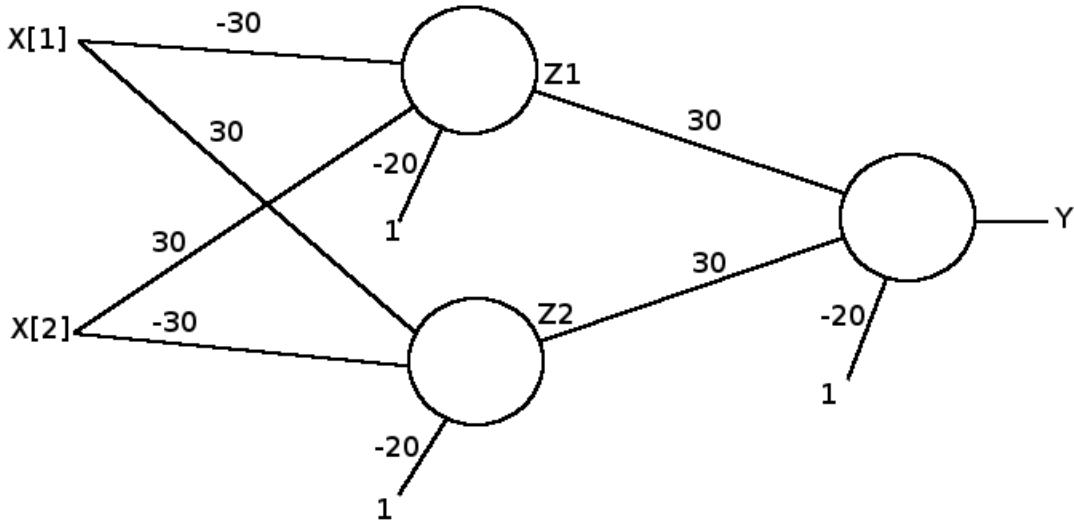
รูปที่ 5.3: แผนผังแสดงโครงสร้างของเพอร์เซปตรอน ซึ่ง เอาร์พุต  $y = f(b + w[1] \cdot x[1] + \dots + w[D] \cdot x[D])$  โดย พังช์  $f$  จะเป็นพังช์จำกัดแข็ง (hard limit function) สมการ 5.2

ตารางที่ 5.1: ตระกconte็กซ์-ออร์ (XOR หรือ exclusive OR)

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

ทางด้านโครงข่ายประสาทเทียมทั้งหมด รวมไปถึงสาขาวิชาปัญญาประดิษฐ์ เสียความนิยมและเสื่อมความสนใจไปในช่วงหลายปีต่อจากนั้น จนเรียกว่า ช่วงเวลาหนึ่งเป็น หน้าหนาวของปัญญาประดิษฐ์ (AI Winter). โครงข่ายประสาทเทียมเสียความนิยมไป จนกระทั่งหลายปีให้หลัง งานของเวอร์โนส[79] และโดยเฉพาะอย่างยิ่งงานของกลุ่มของรูเมลาร์ต อินตัน และวิลเลียม[68] ที่ออกแบบให้เห็นถึงประสิทธิผลของโครงข่ายประสาทเทียม และนำเสนอวิธีการหาค่าพารามิเตอร์ที่มีประสิทธิภาพ ซึ่งงานเหล่านี้ได้ช่วยฟื้นฟูความนิยมของโครงข่ายประสาทเทียมกลับมาใหม่.

เมื่อจะกล่าวไปแล้ว สิ่งที่มินสกี้กับปาเปิตกาว่า เพอร์เซปตรอนทำงานได้แต่งานง่ายๆ ก็ไม่ได้ดีดีขนาดนัก. เพียงแต่ว่า มินสกี้กับปาเปิตสรุปความเห็น จากการวิเคราะห์การทำงานของเพอร์เซปตรอน ชั้นเดียว (one-layer perceptron). การทำงานของโครงข่ายสมองมนุษย์ไม่ได้เป็นชั้นเดียว ในลักษณะเดียวกันโครงข่ายประสาทเทียมที่มีประสิทธิผล จะต้องมีโครงสร้างมากกว่าหนึ่งชั้น. นั่นก็คือ ที่มาของพัฒนาการต่อมา ได้แก่ เพอร์เซปตรอนหลายชั้น (Multi-Layer Perceptron คำย่อ MLP) ซึ่งซื้อได้เน้นย้ำว่า มีการใช้โครงข่ายประสาทเทียมหลายชั้น.



รูปที่ 5.4: โครงข่ายเพอร์เซปตรอนที่ทำงานเลี้ยงแบบตระกูลเอ็กซ์-ออร์ได.

รูป 5.4 แสดงเพอร์เซปตรอนสองชั้น (2-layer perceptron) ที่สามารถเลียนแบบการทำงานของตระกูลเอ็กซ์-ออร์ได้. เอาต์พุตของเพอร์เซปตรอนชั้นแรก ซึ่งคือ  $z_1 = f(-30x_1 + 30x_2 - 20)$  และ  $z_2 = f(30x_1 - 30x_2 - 20)$  ทำหน้าที่เป็นอินพุตของเพอร์เซปตรอนชั้นที่สอง และเอาต์พุตของเพอร์เซปตรอนชั้นที่สอง (และกรณีนี้ก็เป็นเอาต์พุตของโครงสร้างทั้งหมด) คือ  $y = f(30z_1 + 30z_2 - 20)$ . และตาราง 5.2 แจกแจงการทำงาน โดย  $a_1^{(1)}, a_2^{(1)}, a^{(2)}$  เป็นผลรวมของน้ำหนักคูณอินพุตของเพอร์เซปตรอนชั้นที่หนึ่งตัวบัน ตัวล่าง และชั้นที่สอง ตามลำดับ. สังเกตว่า เราสามารถเปลี่ยนค่าน้ำหนักของโครงข่ายประสาทเทียมไปใช้ค่าอื่นได้ โดยที่การทำงานยังคงเดิมได้ เช่น เราอาจใช้ค่า  $20, -20, -10$  แทน  $30, -30, -20$  ในรูปได้ โดยพฤติกรรมของโครงข่ายยังคงให้ผลความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตคงเดิม. ซึ่ง นี่คือลักษณะอย่างหนึ่งของโครงข่ายประสาทเทียม ที่ ค่าน้ำหนักที่ดีที่สุดของโครงข่ายประสาทเทียมมีได้หลายชุด.

หากมองจากทฤษฎีการหาค่าดีที่สุด ปัญหาการหาค่าน้ำหนักที่ดีที่สุดของโครงข่ายประสาทเทียม จะเป็นปัญหาในลักษณะที่เรียกว่า ปัญหาที่ไม่เป็นค่อนเวกซ์ (non-convex problem) ซึ่งการหาค่าน้ำหนักที่ดีที่สุดในปัญหาแบบนี้จะทำได้ยาก. ในทางปฏิบัติ การฝึกโครงข่ายประสาทเทียมก็คือการหาค่าน้ำหนักที่ดี แต่ไม่อาจรับประกันได้เลยว่าเราจะได้ค่าน้ำหนักที่ดีที่สุด ซึ่งเรื่องนี้เป็นลักษณะอย่างหนึ่งที่ผู้ใช้โครงข่ายประสาทเทียมมักจะห้อนอกมาว่า โครงข่ายประสาทเทียมนั้นฝิกยาก

ปัจจุบันโครงข่ายประสาทเทียมเป็นโมเดลได้ถูกนำไปใช้อย่างกว้างขวาง ในงานหลาย ๆ ลักษณะ เช่น การหาค่าลดตอน การจำแนกประเภท การประมาณฟังชัน และในแอพพลิเคชันต่างๆ รวมถึงงานการประมวลผลภาพและการประมวลผลเสียงพูด. นอกจากนี้ มีการศึกษาโครงข่ายประสาทเทียมในทางทฤษฎี และพิสูจน์ว่าโครงข่ายประสาทเทียมเป็นตัวประมาณค่าสากล (universal approximator)[23, 36] ซึ่ง ความหมายคือ โครงข่ายประสาทเทียมสามารถแทนฟังชันใดๆ ก็ได้ ที่ความละเอียดตามที่ต้องการ หาก

ตารางที่ 5.2: การทำงานในแต่หน่วยของเพอร์เซปตอรอน 2 ชั้นในรูป 5.4

$x_1$	$x_2$	$a_1^{(1)}$	$z_1$	$a_2^{(1)}$	$z_2$	$a^{(2)}$	$y$
0	0	$-30(0) + 30(0) - 20 = -20$	$f(-20) = 0$	$30(0) - 30(0) - 20 = -20$	$f(-20) = 0$	$30(0) + 30(0) - 20 = -20$	$f(-20) = 0$
0	1	$-30(0) + 30(1) - 20 = 10$	$f(10) = 1$	$30(0) - 30(1) - 20 = -50$	$f(-50) = 0$	$30(1) + 30(0) - 20 = 10$	$f(10) = 1$
1	0	$-30(1) + 30(0) - 20 = -50$	$f(-50) = 0$	$30(1) - 30(0) - 20 = 10$	$f(10) = 1$	$30(1) + 30(0) - 20 = 10$	$f(10) = 1$
1	1	$-30(1) + 30(1) - 20 = -20$	$f(-20) = 0$	$30(1) - 30(1) - 20 = -20$	$f(-20) = 0$	$30(0) + 30(0) - 20 = -20$	$f(-20) = 0$

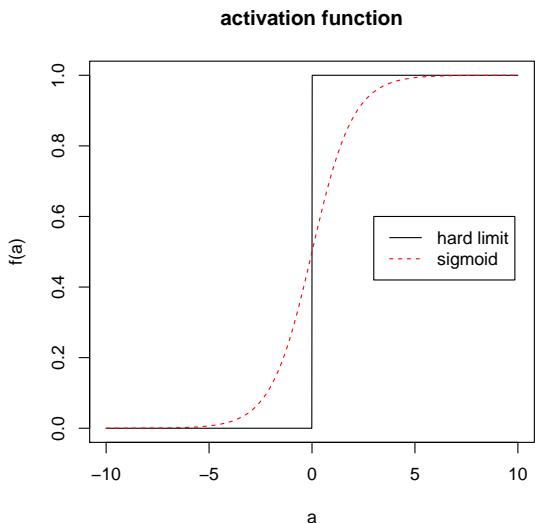
มีจำนวนหน่วยคำนวณมากเพียงพอ. ตามทฤษฎีแล้ว แค่โครงข่ายประสาทเทียมแบบสองชั้นก็เป็นตัวประมาณค่าสากลได้แล้ว.

กลับมาที่เรื่องการหาค่าน้ำหนักที่เหมาะสมของโครงข่ายประสาทเทียม อย่างที่เห็นจากตัวอย่าง การที่จะทำให้โครงข่ายประสาทเทียมทำงานได้ตามต้องการได้นั้น นอกจากโครงข่ายจะต้องมีโครงสร้างที่รองรับได้แล้ว (มีจำนวนหน่วยคำนวณเพียงพอ) ค่าของน้ำหนักต่างๆ จะต้องมีค่าที่เหมาะสมด้วย. การหาค่าน้ำหนักของโครงข่ายประสาทเทียม เราสามารถทำได้ในลักษณะเดียวกับ การที่เราหาค่าพารามิเตอร์ของฟังชันพุนามในหัวข้อ 3.1 นั้นคือการใช้วิธีของการหาค่าดีที่สุด. รายละเอียดของกระบวนการหาค่าน้ำหนัก หรือมักนิยมเรียกว่าการฝึกโครงข่ายประสาทเทียม จะถูกในหัวข้อ 5.3.

ก่อนจะศึกษากระบวนการหาค่าน้ำหนัก มีประเด็นที่น่าสนใจที่ควรกล่าวถึงก่อน นั่นคือการหาอนุพันธ์ เป็นเครื่องมือสำคัญอย่างหนึ่ง สำหรับวิธีของการหาค่าดีที่สุด. เพอร์เซปตอรอนใช้ฟังชันกระตุ้นเป็นฟังชันจำกัดแข็ง แต่เนื่องจาก ฟังชันจำกัดแข็ง(สมการ 5.2) เป็นฟังชันที่มีค่าไม่ต่อเนื่อง (ที่  $a = 0$ ) ทำให้ เราไม่สามารถหาค่าอนุพันธ์ของฟังชันจำกัดแข็งได้ ส่งผลให้การหาค่าน้ำหนักที่เหมาะสมของเพอร์เซปตอรอนทำได้ยาก.

ฟังชันจำกัดแข็ง แม้จะเลียนแบบการทำงานของเซลล์ประสาท แต่ลักษณะทางคณิตศาสตร์ของมันเป็นอุปสรรคที่สำคัญ. การสร้างโมเดลทางคณิตศาสตร์เพื่อประโยชน์ในการทำความเข้าใจเซลล์ประสาททางชีวภาพ จะจดอยู่ในขอบข่ายของประสาทวิทยาเชิงคำนวณ (computational neuroscience) ซึ่งเป็นสาขาเฉพาะ และอยู่นอกเหนือจากขอบเขตของหนังสือเล่มนี้. แม้กระนั้น ฟังชันจำกัดแข็งเองก็ไม่ได้อธิบายการทำงานของเซลล์ประสาททางชีวภาพได้อย่างเที่ยงตรงซึ่งที่เดียว นอกจากนั้น สิ่งที่ต้องการจริงๆ ในมุมมองทางวิศวกรรม ก็คือเครื่องมือที่ใช้งานได้ เช่น โมเดลที่มีความสามารถในการทำงานที่ดี เป็นต้น.

การที่จะฝึกโครงข่ายประสาทเทียมที่ใช้ฟังชันจำกัดแข็งจะทำได้ยากมาก หรือไม่สามารถทำได้อย่างมีประสิทธิภาพ เมื่อปัญหาอยู่ที่ฟังชันจำกัดแข็ง วิธีแก้อย่างหนึ่งก็คือ การผ่อนปรนฟังชันจำกัดแข็งลง โดยแทนที่จะใช้ฟังชันจำกัดแข็ง ฟังชันที่ประมาณฟังชันจำกัดแข็งจึงถูกนำมาใช้แทน. ฟังชันประมาณนั้นคือ ฟังชันซิกมอยด์ (sigmoid function หรือบางครั้งเรียก logistic function หรือ logistic sigmoid function). ฟังชันซิกมอยด์ เป็นฟังชันค่าต่อเนื่อง ดังนั้นจึงสามารถหาอนุพันธ์ตลอดช่วงค่าของอินพุต. เราอาจจะมองว่า ฟังชันซิกมอยด์เป็นการประมาณฟังชันจำกัดแข็ง ด้วยฟังชันที่ต่อเนื่องที่สามารถหาค่าอนุพันธ์ได้. รูป 5.5 เปรียบเทียบค่าของฟังชันจากฟังชันจำกัดแข็งกับฟังชันซิกมอยด์



รูปที่ 5.5: ภาพเปรียบเทียบฟังชันจำกัดแข็งกับฟังชันซิกมอยด์

การใช้ฟังชันซิกมอยด์เป็นพัฒนาการของโครงข่ายประสาทเทียมของยุคต้นๆ. เมื่อความเข้าใจพื้นฐานทางคณิตศาสตร์ดีขึ้น แม้แต่การใช้ซิกมอยด์เองก็ถูกผ่อนปรนลงไป. โดยเฉพาะอย่างยิ่ง งานยุคหลังของการศึกษาโครงข่ายประสาทเทียมแบบลีกที่พบว่า ปัจจัยสำคัญอย่างหนึ่งที่ช่วยแก้ปัญหาการล้มเหลวของโครงข่ายแบบลีก คือ การเปลี่ยนฟังชันกราฟตุ้นไปใช้ฟังชันเชิงเส้นควบคุม (rectified linear) ที่มีช่วงพลวัตรตีกว่าซิกมอยด์ ซึ่งช่วยลดปัญหาเกรเดียนต์เลื่อนหาย (vanishing-gradient issue) ลงได้.

อย่างไรก็ตามแม้ ชื่อของเพอร์เซปตรอนจะเขื่อมโยงกับฟังชันจำกัดแข็ง แต่ในทางปฏิบัติแล้ว ชื่อเพอร์เซปตรอนหลายชั้น (MLP) ก็มักจะอ้างถึงโครงข่ายประสาทเทียมที่ใช้ฟังชันซิกมอยด์เป็นฟังชันกราฟตุ้น<sup>1</sup>

การอภิรายข้างต้นถูกถึงพัฒนาการของโครงข่ายประสาทเทียม จากแรงบันดาลใจที่ได้จากการเลียนแบบเซลล์ประสาททางชีวภาพ มาเป็นโมเดลแบบเพอร์เซปตรอน จนพัฒนาต่อมาเป็นเพอร์เซปตรอนหลายชั้น. ในทางปฏิบัติ การใช้งานโครงข่ายประสาทเทียมไม่จำเป็นต้องเข้าใจ หรือไปผูกภาพกับโครงข่ายประสาททางชีวภาพ และการทำงานของโครงข่ายประสาทเทียมก็สามารถอธิบายได้ เช่นเดียวกับโมเดลทางคณิตศาสตร์อื่นๆ. หัวข้อ 5.2 อธิบายโครงข่ายประสาทเทียม จากมุ่งมองของโมเดลทางคณิตศาสตร์ ที่ลักษณะเฉพาะของโครงข่ายประสาทเทียม สามารถเข้าไปช่วยแก้ปัญหาการสร้างโมเดลที่ยืดหยุ่นได้ ดังที่เกริ่นไว้ตอนต้น.

เกร็ดความรู้สมองมนุษย์ (เรียบเรียงจาก [60] [33] และ [80]) โดยเฉลี่ยแล้ว สมองมนุษย์มีขนาดประมาณ 1.13 ถึง 1.26 ลิตร และหนักประมาณ 1.3 กก. ใช้ออกซิเจนประมาณ 20 เปอร์เซ็นของปริมาณหัวหมดที่ร่างกายรับเข้าไป และใช้กำลังงานประมาณ 25 วัตต์[29].

<sup>1</sup> ฟังชันไฮเปอร์โบลิกแทนเงนต์ (hyperbolic tangent function)  $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$  ก็เป็นอีกฟังชันที่นิยมนำมาใช้เป็นฟังชันกราฟตุ้นของโครงข่ายประสาทเทียม. แต่เพื่อความกระชับ เราจะพูดถึงซิกมอยด์ฟังชันเพียงอย่างเดียว.

สมองเชื่อมต่อกับส่วนอื่นๆ ของร่างกายผ่านไขสันหลัง และระบบประสาทนอกส่วนกลาง (Peripheral Nervous System ค่าย่อ PNS) ไขสันหลังทำหน้าที่หลักๆ คือเชื่อมต่อสัญญาณควบคุมจากสมองไปยังส่วนต่างๆ ของร่างกาย และส่งผ่านสัญญาณรับรู้จากส่วนต่างๆ ของร่างกายกลับไปยังสมอง และไขสันหลังเองก็มีระบบประสาทของตัวเองที่ช่วยทำงาน เช่นการควบคุมการทำงานของกล้ามเนื้อ รวมถึงการทำงานของระบบประสาทในส่วนต่างๆ ของร่างกาย เป็นต้น. ระบบประสาทนอกส่วนกลางมีหน้าที่หลักคือเชื่อมต่อสัญญาณจากสมองและไขสันหลังไปสู่วัยรุ่นต่างๆ.

การทำงานของสมองมีลักษณะคล้ายคณิตกรรมการของกลุ่มผู้เชี่ยวชาญจำนวนมาก นั่นคือ ส่วนต่างๆ ของสมองทำงานร่วมกัน แต่ร่วมกันแต่ละส่วนของสมองมีหน้าที่เฉพาะด้าน. เราอาจมองได้ว่า ส่วนของสมองมีสามส่วนใหญ่ๆ คือ สมองส่วนบน (forebrain), สมองส่วนกลาง (midbrain), และ สมองส่วนล่าง (hindbrain).

สมองส่วนล่างนับรวมส่วนบนของไขสันหลัง ก้านสมอง (brain stem) และ เชเรเบลัม (cerebellum). สมองส่วนล่างจะควบคุมการทำงานที่เป็นพื้นฐานของการดำเนินชีพ เช่น การหายใจ และการเต้นของหัวใจ. เชเรเบลัมช่วยประสานงานเรื่องการเคลื่อนไหวและการเรียนรู้ของการเคลื่อนไหวที่เกิดจากการฝึกทำซ้ำ เช่น การเล่นเปียโนหรือการตีกลองแทนนิส จะอาศัยการทำงานของเชเรเบลัมช่วย.

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

เนื่องจากระบบประมวลผลภาพในสมองมีทั้งที่สมองส่วนกลางและบริเวณเปลือกสมองส่วนการเห็นในสมองส่วนบน ทฤษฎีวิวัฒนาการเชื่อว่า การประมวลภาพที่สมองส่วนกลางเป็นวิวัฒนาการในช่วงก่อน (สัตว์หลายชนิด เช่น กบ ใช้การประมวลภาพที่สมองส่วนกลางเป็นหลัก) และเปลือกสมองส่วนการเห็นเป็นวิวัฒนาการในช่วงต่อมา. ผู้เชี่ยวชาญด้านประสาทวิทยาเดวิด ลินเดน (ผู้เขียนหนังสือ Accidental Mind[49]) ได้อธิบายเพิ่มเติมในการสนทนาส่วนตัวว่า หากการประมวลผลภาพของสมองส่วนกลางเสียหาย แต่ส่วนอื่นๆ ในระบบการมองเห็นยังดีอยู่ รวมถึงเปลือกสมองส่วนการเห็นก็ยังดีอยู่ ผู้ป่วยจะรับรู้ถึงการมองเห็นได้แต่พบว่าผู้ป่วยจะมีการตอบสนองการประสานงานระหว่างมือและตา (hand-eye coordination) ที่ช้าลงอย่างชัดเจน.

สมองส่วนบนเป็นส่วนที่ใหญ่ที่สุดในสมองส่วน. สมองส่วนบนประกอบด้วยเชเรบัม (cerebrum) และส่วนสมองใน (the inner brain). หมายเหตุ เชเรบัม (ของสมองส่วนบน) มาจากภาษาลาติน แปลตรงตัวว่า สมอง ขณะที่ เชเรเบลัม (ของสมองส่วนล่าง) มาจากภาษาลาติน ซึ่งแปลตรงตัวว่า สมองน้อย. เชเรบัมคือภาพของสมองที่คนทั่วไปจะนึกถึงเมื่อกล่าวถึงสมอง. เชเรบัมทำหน้าที่หลักในการรับรู้ ความจำ การวางแผน การคิด การจินตนาการ รวมถึงศีลธรรม นิสัย และบุคลิกภาพ. เมื่อมองจากด้านบน เชเรบัมดูเหมือนจะแบ่งได้เป็นซีกซ้ายและซีกขวา โดยมีดูเหมือนมีร่องแบ่งสมองสองซีกนี้ออกจากกัน. สมองทั้งสองซีกเชื่อมต่อกันผ่านเส้นใยประสาทรียกว่า คอร์ปัส คาโลซัม (corpus callosum). สมองทั้งสองซีกนี้ทำงานร่วมกัน แต่สมองซีกซ้ายจะควบคุมการทำงานของร่างกายซีกขวา และสมองซีกขวาจะควบคุมการทำงานของร่างกายซีกซ้าย โดยสมองซีกซ้ายจะเด่นด้านการทำงานเกี่ยวกับภาษา การวิเคราะห์รายละเอียด และทักษะเชิงรูปธรรม ในขณะที่สมองซีกขวาจะเด่นด้านการอ่านภาพรวม และทักษะเชิงนามธรรม. การทำงานไขว้ระหว่างซีกสมองกับร่างกายนั้น แม้จะยังไม่มีคำอธิบายว่าเหตุใดกลไกของร่างกายจึงเป็นเช่นนั้น แต่ข้อเท็จจริงคือสัญญาณจากสมองซีกหนึ่งจะไขว้ไปบังคับร่างกายอีกซีกหนึ่ง ดังนั้น หากสมองซีกหนึ่งเสียหาย ร่างกายอีกซีกหนึ่งจะได้รับผลกระทบ เช่น ผู้ป่วยโรคหลอดเลือดสมอง เมื่อเกิดสมองซีกขวาเสียหาย จะส่งผลให้ผู้ป่วยเป็นอัมพาตในซีกซ้ายของร่างกาย.

การศึกษาที่น่าสนใจเกี่ยวกับสมองซีกซ้ายและขวา หล่ายกรณีได้มาจาก การศึกษาผู้ป่วยโรคลมชักรุนแรง ที่แพทย์ต้องตัดคอร์ปัส คาโลซัมเพื่อลดความรุนแรงของการล้มซึ่งไม่ให้แพร่ขยายข้ามซีกสมองได้. เช่นนี้ในตัวอย่างที่บรรยายโดยวีโนกราด

อพ[78] คือ การศึกษาที่นำผู้ป่วยที่ผ่านการตัดการเชื่อมต่อระหว่างสมองซึ่งกันและขวากัน มาใส่ค้อนแทกเลนส์พิเศษ เพื่อแยกการมองเห็นระหว่างตาซ้ายและตาขวาออกจากกัน. ตาซ้ายและร่างการศึกษาเชื่อมโยงกับสมองซึ่งกัน ตาขวาและร่างการศึกษาเชื่อมโยงกับสมองซึ่งกัน. เมื่อให้ตาขวาบุพลาของเท้าของໄก่ และให้ตาซ้ายบุพลาของบ้านที่ถูกหิมะท่วม พร้อมสั่งให้ผู้ทดลองซึ่งเลือกภาพที่เกี่ยวข้องด้วยมือซ้ายและขวา ผู้ทดลองซึ่งมือขวาไปที่ภาพตัวแม่ໄก่ และซึ่งมือซ้ายไปที่ภาพล้วง ผู้ทดลองอธิบายถึงเท้าໄก่ได้ แต่ไม่สามารถอธิบายภาพของบ้านที่ถูกหิมะท่วมได้ และเมื่อให้ผู้ทดลองอธิบายเหตุผลที่ซึ่งเลือกภาพแม่ໄก่ และ พลั่ว สมองส่วนซ้าย ซึ่งไม่ได้รับรู้ภาพของบ้านที่ถูกหิมะท่วม ก็พยายามอธิบายไปว่า เท้าໄก่เกี่ยวข้องกับแม่ໄก่ และพลั่วเกี่ยวข้องคือเป็นเครื่องมือตักมูลไก่. กรณีนี้ ผู้เชี่ยวชาญอธิบายว่า สมองซึ่งกันและขวากันมีความสามารถทางภาษา แต่ไม่ได้รับภาพที่สมองซึ่งกันและขวากัน ไม่ได้รับรู้ถึงภาพบ้านที่หิมะท่วม แต่สมองซึ่งกันและขวากันมีความสามารถทางภาษา จึงไม่สามารถอธิบายออกมาเป็นคำพูดได.

เซเรบรัมแต่ละชีกยังสามารถแบ่งเป็นส่วนต่างๆได้อีก ซึ่งแต่ละส่วนของเซเรบรัมมักจะเรียกว่ากลีบ(lobe). เซเรบรัมมีกลีบหลักๆ เช่น กลีบหน้า (frontal lobe), กลีบข้าง (parietal lobe), กลีบท้ายทอย (occipital lobe), กลีบขมับ (temporal lobe) เป็นต้น. สมองกลีบหน้าจะอยู่บริเวณหลังหน้าพากของเรา และทำหน้าที่เกี่ยวกับ การวางแผน การจินตนาการถึงอนาคต การใช้เหตุผล การควบคุมด้วย บุคคลิกภาพ และ ศีลธรรม. ลักษณะที่สำคัญที่สุดของกลีบหน้าคือ กลีบขมับซึ่งเป็นบริเวณที่ทำหน้าที่เกี่ยวกับการควบคุมการเคลื่อนไหว. ในกลีบหน้าของสมองซึ่งมีบริเวณใบරาก้า (Broca's area) ซึ่งเป็นส่วนที่ทำหน้าที่เกี่ยวกับการใช้ภาษา.

สมองกลีบข้างซึ่งอยู่ถัดจากกลีบหน้าเข้ามา (บริเวณใต้กล่างกระหม่อม) ทำหน้าที่เกี่ยวกับส กลิ่น สัมผัส รวมถึงการรับรู้ การเคลื่อนไหวของร่างกาย ความสามารถในการอ่านหนังสือและการคิดคำนวณตัวเลข ก็เกี่ยวข้องกับสมองกลีบข้าง. สมองกลีบห้ายทอยอยู่ถัดจากกลีบข้างไปทางหน้าหลัง (บริเวณห้ายทอย) ทำหน้าที่หลักเกี่ยวกับการมองเห็น. เปลือกสมองส่วนการเห็น ซึ่งเป็นส่วนประมวลผลการมองเห็นหลักก็อยู่ในบริเวณกลีบท้ายทอย. สมองกลีบขมับจะอยู่ใต้กลีบหน้าและกลีบข้าง ซึ่งเมื่อเทียบ กับภายนอกแล้วจะอยู่บริเวณมีบ. สมองกลีบขมับทำหน้าที่หลักเกี่ยวกับการประมวลผลเสียงต่างๆ และมีหน้าที่ช่วยในการรวม ความจำและความรับรู้ต่างๆทั้งภาพ เสียง กลิ่น และ สัมผัส เข้าด้วยกัน.

ที่ผิวนอกของเชrebรัมจะเป็นชั้นของเนื้อยื่อที่หนาประมาณ 2 ถึง 4 มิลลิเมตร ซึ่งเรียกว่า เชrebอลคอร์เท็กซ์ (cerebral cortex). การประมวลผลของสมองส่วนใหญ่เข้ากันว่าเกิดขึ้นภายในเนื้อยื่อส่วนนี้ เนื้อยื่อส่วนนี้จะมีสีเท็มกว่าเนื้อยื่อส่วนด้านใน และมักถูกอ้างถึงในชื่อของเนื้อเทา (gray matter) เปรียบเทียบกับเนื้อขาว (white matter) ซึ่งอยู่ภายในเนื้อเทาจะประกอบด้วยเซลล์ประสาท หลอดเลือดผอย และ เซลล์เกลีย. เซลล์ประสาทในเนื้อเทาจะมีไขมันที่เป็นชนวนน้อยกว่าเซลล์ประสาทในเนื้อขาว จึงทำให้สีของเนื้อยื่อโดยรวมดูเข้มกว่า. รายละเอียดของเซลล์ประสาทจะอธิบายในเกร็ดความรู้เซลล์ประสาท. เนื่องจากเชrebอลคอร์เท็กซ์เป็นผิวของสมอง รอยหยักของสมองจะช่วยเพิ่มพื้นที่ผิวและปริมาณของเนื้อเทาซึ่งสัมพันธ์กับปริมาณของข้อมูลที่สมองสามารถประมวลผลได้.

ส่วนสมองในเป็นอีกบริเวณในสมองส่วนบน. ส่วนสมองในนี้จะเข้ามือต่อไขสันหลังเข้ากับเซเรบรัม. ส่วนสมองในทำหน้าที่เกี่ยวกับอารมณ์ มีส่วนในการเปลี่ยนแปลงการรับรู้และการตอบสนองไปตามสถานะของอารมณ์ในขณะนั้นๆ มีส่วนช่วยเรื่องการเคลื่อนไหวต่างๆที่เราทำโดยไม่ต้องคิดถึงการเคลื่อนไหวเหล่านั้น และมีส่วนสำคัญในกระบวนการสร้างความจำ. ส่วนประกอบต่างๆของส่วนสมองในนี้จะมีเป็นคู่ๆทางซ้ายและขวา โดยมีส่วนประกอบที่สำคัญ เช่น ไฮปोราลามัส (hypothalamus) รามาลัส (thalamus) นาซอลังก์เกลีย (basal ganglia) อะมิกดาลา (amygdala) hippocampus เป็นต้น. ไฮปอราลามัสเป็นเสมือนศูนย์กลางการจัดการอารมณ์. รามาสช่วยจัดการข้อมูลที่ผ่านไปมาระหว่างเซเรบรัมและไขสันหลัง. นาซอลังก์เกลียช่วยการเริ่มและประสานงานการเคลื่อนไหวต่างๆ. โรคพาร์กินสันซึ่งผู้ป่วยจะมีอาการที่เด่นชัดคือมีปัญหาการเคลื่อนไหว เช่น อาการสั่น เดินหรือเคลื่อนไหวได้ช้า เป็นโรคที่เกี่ยวพันกับเซลล์ประสาทที่เชื่อมต่อกับนาซอลังก์เกลียนี้. อะมิกดาลาทำหน้าที่เกี่ยวกับอารมณ์ ความกลัว ความก้าวหน้า และ ความจำที่เกี่ยวข้องกับอารมณ์ความรู้สึก. มีงานศึกษาที่พบความเกี่ยวข้องระหว่างขนาดของอะมิกดาลาของบุคคลกับความสัมพันธ์ทางสังคมของบุคคลนั้น. hippocampus ทำหน้าที่จัดส่งความจำใหม่ไปเก็บในตำแหน่งที่เหมาะสมในเซเรบรัม และค้นหาความจำที่ต้องการจากเซเรบรัม. ผู้ป่วยที่สูญเสียอิบิโภคเมป์ไปจะ

สูญเสียความสามารถในการสร้างความทรงจำใหม่.

## 5.2 โครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า

จากโมเดลเชิงเส้น (บทที่ 4), เราสามารถเขียนเอาต์พุต ในรูปของผลจากฟังชันกระตุ้นของผลรวมของค่าน้ำหนักคูณค่าเบซิสฟังชัน ได้ว่า

$$y(\mathbf{x}, \mathbf{w}) = f \left( \sum_{j=1}^M w_j \phi_j(\mathbf{x}) \right) \quad (5.3)$$

เมื่อ  $f(\cdot)$  เป็นฟังชันกระตุ้น ซึ่งในกรณีของการจำแนกประเภทจะใช้เป็นฟังชัน sigmoid  $f(a) = 1 / \{1 + \exp(-a)\}$  หรือว่า ในกรณีการหาค่าคาดถอยจะใช้เป็นฟังชันเอกลักษณ์  $f(a) = a$ . ส่วน  $\phi_j(\mathbf{x})$  เป็นเบซิสฟังชัน. จากเหตุผลที่เกริ่นตอนต้นของบท เราต้องการเบซิสฟังชันที่สามารถปรับตัวเองได้ นอกจากนั้น เบซิสฟังชันต้องไม่เป็นฟังชันเชิงเส้นกับ  $\mathbf{x}$  (ดูแบบฝึกหัดข้อ 5).

แนวคิดของโครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า (Feed-Forward Network) สามารถนำมาช่วยสถานะการณ์นี้ได้. โครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า อาจมองได้ว่าเป็นการรวมโมเดลย่อยๆ แต่ละตัวเข้าด้วยกัน. โมเดลย่อยแต่ละตัวทำการคำนวณง่ายๆ และส่งผลการคำนวณต่อไปให้โมเดลย่อยตัวอื่น ที่ก็ทำการคำนวณง่ายๆ เช่นกัน และส่งต่อผลการคำนวณต่อไป เป็นลำดับชั้น จนกระทั่งได้คำตอบที่ต้องการ. การส่งต่อผลการคำนวณจะเป็นการส่งต่อไปในทางเดียว โดยไม่มีการส่งค่าบันกลับ. เราจึงเรียกโมเดลแบบนี้ว่า โครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า<sup>2</sup>

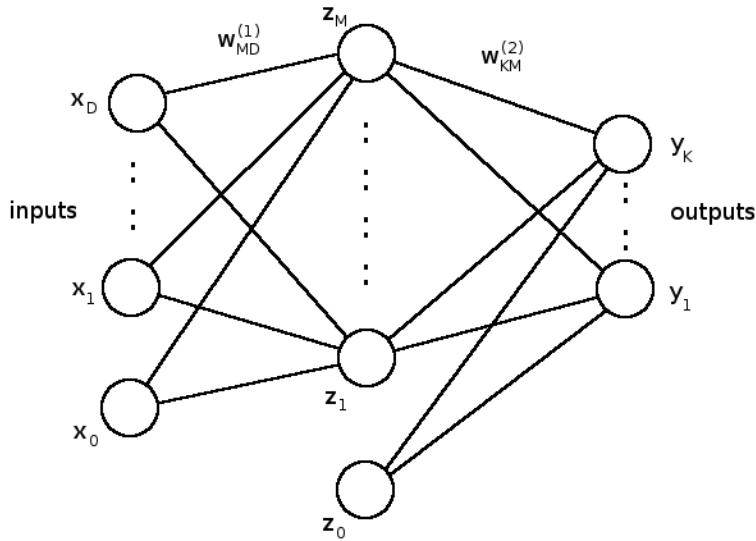
การคำนวณง่ายๆ ที่แต่ละโมเดลย่อยทำก็แค่ (1) ทำการบวกผลคูณของน้ำหนักกับค่าอินพุตของโมเดล (สมการ 5.5) และ (2) นำผลบวกที่ได้ ซึ่งเรียกว่า การกระตุ้น (activation) ไปผ่านฟังชันการกระตุ้น (สมการ 5.4). ดังนั้น เอาต์พุตของโมเดลย่อย  $j$  ในชั้น  $l$  สามารถเขียนในรูปคณิตศาสตร์ได้ว่า

$$z_j^{(l)} = h(a_j) \quad (5.4)$$

$$a_j = \sum_{i=1}^M w_{ji} z_i^{(l-1)} \quad (5.5)$$

เมื่อ  $h(\cdot)$  เป็นฟังชันกระตุ้น,  $w_{ji}$  เป็นค่าน้ำหนักของโมเดลย่อย  $j$  จากอินพุตที่  $i$ ,  $z_j^{(l)}$  เป็นเอาต์พุตของโมเดลย่อยนี้, และ  $z_i^{(l-1)}$  เป็นอินพุตที่  $i$  ของโมเดลย่อยนี้. สังเกตุ ค่าอินพุตของโมเดลย่อย  $z_i^{(l-1)}$  ทำหน้าที่เป็นเหมือนกับเบซิสฟังชันในสมการ 5.3 และ อินพุตของโมเดลย่อยตัวหนึ่งก็ได้มาจากการเอาต์พุตของโมเดลย่อยตัวอื่นๆ ที่อยู่ชั้นก่อนหน้า. การจัดโครงสร้างลักษณะนี้ ก็ทำให้เราได้โมเดลที่มีเบซิสฟังชัน

<sup>2</sup> แม้ว่าชื่อเพอร์เซปตรอนแบบหลายชั้นไม่ได้เจาะจงว่า โครงข่ายจะต้องมีลักษณะผ่านผลการคำนวณไปทางเดียว แต่ในทางปฏิบัติแล้ว เพอร์เซปตรอนแบบหลายชั้นและโครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า มักหมายถึงโครงข่ายแบบเดียวกัน.



รูปที่ 5.6: โครงสร้างของโครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า. ภาพดัดแปลงจาก [9]

ที่สามารถปรับตัวเองได้. สำหรับโครงข่ายประสาทเทียม เราแม้จะเรียกโมเดลย่ออย่างนี้ว่า หน่วยประสาท (neuron) หรือ หน่วยย่อย (unit).

รูป 5.6 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้า ที่มีโครงสร้าง 2 ชั้น<sup>3</sup> โดยที่อินพุต  $\mathbf{x}$  มี  $D$  มิติ นั่นคือ  $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_D]^T$ . สังเกตว่า  $x_0 = 1$  เสมอ. ค่า  $x_0 = 1$  นี้กำหนดขึ้นเพื่อความสะดวก ทำให้เราสามารถเขียนผลบวกของอินพุตคูณหนักในรูปการคูณของเวคเตอร์ได้ เช่น เราสามารถเขียน  $a_1 = \mathbf{w}_1^T \mathbf{x}$  แทนการเขียน  $a_1 = w_{10} + \sum_{d=1}^D w_{1d} x_d$ . ค่าคงที่  $w_{10}, w_{20}, \dots$  จะเรียกว่า ไบอส (bias).

ชั้นอินพุต (input layer) คือกลุ่มของค่าอินพุตของโครงข่ายและไบอส เช่น  $x_1, \dots, x_D$  และ  $x_0$  ในรูป 5.6. ชั้นซ่อน (hidden layer) คือกลุ่มของหน่วยย่อย ที่รับอินพุตมาจากการหน่วยย่อยในชั้นก่อนหน้า และส่งเอาต์พุตต่อไปให้หน่วยย่อยในชั้นถัดไป เช่น หน่วยย่อยที่มีเอาต์พุตเป็น  $z_1, \dots, z_M$  (และ เพื่อความสะดวก รวมถึง  $z_0$  ซึ่งเป็นไบอสในชั้นซ่อนด้วย) ในรูป. ชั้นเอาต์พุต (output layer) คือกลุ่มของหน่วยย่อยที่ส่งเอาต์พุตออกไปเป็นเอาต์พุตของโครงข่าย เช่น หน่วยย่อยที่มีเอาต์พุตเป็น  $y_1, \dots, y_K$  ในรูป. หน่วยย่อยที่อยู่ในชั้นซ่อน จะเรียกว่า หน่วยซ่อน (hidden unit). ในรูป 5.6 แสดงชั้นซ่อนแค่ 1 ชั้น ซึ่งหากต้องการ เราสามารถสร้างโครงข่ายให้มีชั้นซ่อนมากกว่านี้ได้ ซึ่งการใช้ชั้นซ่อนจำนวนมากๆ จะเรียกว่า เป็นโครงข่ายแบบลึก. ชั้นซ่อนในรูป มีจำนวนหน่วยย่อย  $M$  หน่วย ดังนั้นจะมีค่าที่คำนวณออกมา คือ  $z_1, \dots, z_M$  และ เช่นเดียวกัน ค่า  $z_0$  ถูกกำหนดขึ้นมาเพื่อความสะดวก และ  $z_0 = 1$  เสมอ.

ในรูป เอาต์พุตของโครงข่าย  $\mathbf{y}$  มี  $K$  มิติ. นั่นคือ  $\mathbf{y} = [y_1, \dots, y_K]^T$ . ค่าของเอาต์พุตของหน่วยย่อย

<sup>3</sup> การนับจำนวนชั้นของโครงสร้างประสาทเทียมอาจแตกต่างกันไป เช่น บางคนอาจจะนับโครงสร้างในรูป 5.6 เป็น 3 ชั้น หรือ บางคนอาจจะนับเฉพาะแค่ชั้นซ่อน ซึ่งก็จะนับอีกมาเป็น 1 ชั้น. หนังสือเล่มนี้ยึดแนวทางตาม [9] ที่โครงสร้างในรูป 5.6 เป็นโครงสร้างแบบ 2 ชั้น ตาม ค่าพารามิเตอร์ที่มีอยู่ 2 ชุด คือ  $\{w_{md}^{(1)}\}$  กับ  $\{w_{km}^{(2)}\}$ .

แต่ละตัว (ทั้ง  $z_1, \dots, z_M$  และ  $y_1, \dots, y_K$ ) คำนวณจาก สมการ 5.4 และ 5.5, เช่น

$$\begin{aligned} \text{ชั้นที่ } 2 \quad y_k &= h_2(a_k^{(2)}) = h_2\left(\sum_{m=0}^M w_{km}^{(2)} z_m\right) \\ \text{และ ชั้นที่ } 1 \quad z_m &= h_1(a_m^{(1)}) = h_1\left(\sum_{d=0}^D w_{md}^{(1)} x_d\right) \end{aligned}$$

เมื่อ  $h_1(\cdot)$ ,  $h_2(\cdot)$  เป็นฟังชันกระตุ้นของชั้นที่ 1 และ 2 ตามลำดับ,  $a_m^{(1)}$ ,  $m = 1, \dots, M$  และ  $a_k^{(2)}$ ,  $k = 1, \dots, K$  เป็น การกระตุ้นของหน่วยอยู่ต่างๆ ของชั้นที่ 1 และ 2 ตามลำดับ. จากสมการข้างต้น ความสัมพันธ์ระหว่าง เอาต์พุตของโครงข่าย  $y_k$  กับ อินพุต  $x_d$  คือ

$$y_k = h_2\left(w_{k0}^{(2)} + \sum_{m=1}^M w_{km}^{(2)} \cdot h_1\left(w_{m0}^{(1)} + \sum_{d=1}^D w_{md}^{(1)} x_d\right)\right). \quad (5.6)$$

เมื่อเปรียบเทียบสมการ 5.6 กับ 5.3 จะเห็นว่า  $z_m$  ทำหน้าที่เหมือนกับค่าเบซิสฟังชันในสมการ 5.3. เพื่อ ความสะดวก เราสามารถเขียนสมการ 5.4 และ 5.5 ได้ในรูปเมตริกซ์ได้,

$$\mathbf{z}^{(l)} = \mathbf{h}(\mathbf{a}^{(l)}) \quad (5.7)$$

$$\mathbf{a}^{(l)} = \mathbf{W}^{(l)} \cdot \dot{\mathbf{z}}^{(l-1)} \quad (5.8)$$

$$\dot{\mathbf{z}}^{(l-1)} = [1, z_1^{(l-1)}, z_2^{(l-1)}, \dots, z_P^{(l-1)}]^T \quad (5.9)$$

เมื่อ การกระตุ้นของหน่วยต่างๆ ในชั้น  $l$  คือ  $\mathbf{a}^{(l)} = [a_1^{(l)}, a_2^{(l)}, \dots, a_Q^{(l)}]^T$ , เมื่อชั้น  $l$  มี  $Q$  หน่วย. ฟังชัน กระตุ้น  $\mathbf{h}(\cdot)$  ทำงานกับเมตริกซ์แบบตัวต่อตัว นั่นคือ  $\mathbf{h}([a_1^{(l)}, \dots, a_Q^{(l)}]^T) = [h(a_1^{(l)}), \dots, h(a_Q^{(l)})]^T$ . น้ำหนักของชั้น  $l$  คือ  $\mathbf{W}^{(l)}$  เป็น เมตริกซ์ขนาด  $Q \times (1+P)$  นั่นคือ  $\mathbf{W}^{(l)} = [w_{qp}^{(l)}]_{q=1, \dots, Q; p=0, \dots, P}$  และ  $\mathbf{z}^{(0)} = \mathbf{x}$ . หมายเหตุ เราสามารถกำหนดฟังชันการกระตุ้น  $\mathbf{h}(\cdot)$  แตกต่างไปสำหรับแต่ละชั้นได้ โดย ใช้ตัวแปร  $\mathbf{h}^{(l)}(\cdot)$  แทน เพื่อระบุว่าเป็นฟังชันของชั้น  $l$ .

ในทำนองเดียวกันสำหรับโครงข่าย 2 ชั้น สมการ 5.6 สามารถแยกแจงได้เป็น

$$\dot{\mathbf{x}} = [1, \mathbf{x}] \quad (5.10)$$

$$\mathbf{z} = \mathbf{h}(\mathbf{W}^{(1)} \cdot \dot{\mathbf{x}}) \quad (5.11)$$

$$\dot{\mathbf{z}} = [1, \mathbf{z}] \quad (5.12)$$

$$\mathbf{y} = \boldsymbol{\sigma}(\mathbf{W}^{(2)} \cdot \dot{\mathbf{z}}) \quad (5.13)$$

เมื่อ  $\boldsymbol{\sigma}(\cdot)$  คือ ฟังชันกระตุ้นชั้นเอาต์พุต ที่ทำงานกับเมตริกซ์แบบตัวต่อตัว. รายการ 6.1 แสดงโปรแกรมใน ภาษาอาร์โตรเจคสำหรับคำนวณค่าเอาต์พุตของโครงข่าย โดยอาศัยความสามารถเชิงเมตริกซ์ของอาร์โตร เจค.

การเขียนโปรแกรมในรูปเมตริกซ์ช่วยให้โปรแกรมกระชับและอ่านง่ายขึ้น. นอกจากนั้น สำหรับระบบการคำนวณที่สนับสนุนการคำนวณเมตริกซ์ เช่น อาร์เพรเจค, แมทแลป, และ ไฟรอน-นัมไฟ. โปรแกรมที่เขียนโดยใช้คุณสมบัติการคำนวณเมตริกซ์ จะให้ผลการรันที่มีประสิทธิภาพมากกว่าโปรแกรมที่ทำงานเดียวกันแต่เขียนโดยใช้การวนลูป เนื่องจากระบบเหล่านี้ภายในได้มีการปรับแต่งเพื่อการคำนวณเมตริกซ์อย่างมีประสิทธิภาพมาแล้ว

### 5.2.1 พังชั้นกระตุน

หัวข้อ 5.1 อภิปรายว่า โครงข่ายประสาทเทียมใช้พังชั้นซิกมอยด์เป็นพังชั้นกระตุน. โดยทั่วไป เราสามารถกล่าวได้ว่า พังชั้นซิกมอยด์เป็นพังชั้นกระตุนที่เหมาะสมและใช้งานได้กับโครงข่ายประสาทเทียม ยกเว้นพังชั้นกระตุนของหน่วยย่อยในชั้นเออต์พุต ที่สำหรับงานบางประเภท พังชั้นกระตุนชนิดอื่นจะมีความเหมาะสมกับงานมากกว่า. เพื่อจำแนกความต่าง ให้  $\sigma(\cdot)$  แทนพังชั้นกระตุนของหน่วยย่อยในชั้นเออต์พุต. เราสามารถเขียนสมการ 5.6 สำหรับโครงข่ายจ่ายไปข้างหน้า 2 ชั้น ได้เป็น

$$y_k = \sigma(a_k^{(2)}) \quad (5.14)$$

$$a_k^{(2)} = w_{k0}^{(2)} + \sum_{j=1}^M w_{kj}^{(2)} \cdot z_j^{(1)} \quad (5.15)$$

$$z_j^{(1)} = h(a_j^{(1)}) \quad (5.16)$$

$$a_j^{(1)} = w_{j0}^{(1)} + \sum_{i=1}^D w_{ji}^{(1)} \cdot z_i^{(0)} \quad (5.17)$$

$$z_i^{(0)} = x_i \quad (5.18)$$

โดย  $h(\cdot)$  เป็นพังชั้นซิกมอยด์ นั่นคือ  $h(a) = 1/\{1 + \exp(a)\}$  และพังชั้นกระตุนชั้นเออต์พุต  $\sigma(\cdot)$  จะขึ้นกับ ชนิดของงานที่นำโครงข่ายประสาทเทียมไปใช้ (ตาราง 5.3).

สังเกตุสมการ 5.14 และ 5.15 คือ เออต์พุตและการกระตุนของโครงข่ายชั้นที่สอง. ในลักษณะเดียวกัน สมการ 5.16 และ 5.17 คือ เออต์พุตและการกระตุนของโครงข่ายชั้นที่หนึ่ง. สมการ 5.18 คือ อินพุตของโครงข่าย หรือ อาจจะมองเป็นเสมือนกับ เออต์พุตชั้นที่ศูนย์ก็ได้. ลักษณะเช่นนี้ เป็นตั้งรูปแบบสมการ 5.4 และ 5.5 ที่กล่าวไป.

เราพิจารณาการใช้งานโครงข่ายประสาทเทียม กับ งาน 3 ชนิดหลักๆ คือ การหาค่าคงด้อย, การจำแนกประเภทสองกลุ่ม, และ การจำแนกประเภทหลายกลุ่ม. การใช้พังชั้นเอกลักษณ์ (identity function:  $y_k = a_k^{(2)}$ ) จะทำงานได้กับงานการหาค่าคงด้อย นั่นคือ เออต์พุตของโมเดลสามารถมีค่าได้ทุกค่าของจำนวนจริง. การใช้พังชั้นเอกลักษณ์จะไม่จำกัดค่าของเออต์พุตอยู่แค่ในช่วง 0 ถึง 1 แบบที่เกิดขึ้นกับการใช้พังชั้นซิกมอยด์. งานการจำแนกประเภทสองกลุ่ม เป็นงานชนิดที่พังชั้นซิกมอยด์เหมาะสมที่จะใช้เป็นพังชั้นกระตุน. สำหรับการจำแนกประเภทสองกลุ่ม พังชั้นซิกมอยด์จะทำงานได้กว่าพังชั้นเอกลักษณ์ ดังเหตุผลที่อธิบายในหัวข้อ 4.3. นอกจากนั้น การใช้พังชั้นซิกมอยด์ยังทำให้เราสามารถตีความค่าเออต์พุต  $y_k$  ในเชิง

ตารางที่ 5.3: พังชั้นกระตุ้นสำหรับหน่วยอยู่ในชั้นเออตพุตของโครงข่ายประสาทเทียม (ด้านบนของการกระตุ้นถูกละไว้เพื่อความกระชับของสมการ)

การหาค่าทดแทน (regression) ใช้ฟังชั้นเอกลักษณ์

$$y_k = a_k.$$

การจำแนกประเภทสองกลุ่ม (biclass classification) ใช้ฟังชั้นซิกโนยด์

$$y_k = \frac{1}{1+\exp(-a_k)}.$$

การจำแนกประเภทหลายกลุ่ม (multiclass classification) ใช้ฟังชั้นซอฟต์แม็กซ์

$$y_k = \frac{\exp(a_k)}{\sum_q \exp(a_q)}.$$

ความน่าจะเป็นได้. เช่น ค่าเออตพุตอาจตีความเป็นค่าทำนายความน่าจะเป็นที่จะจัดเป็นกลุ่ม 1 กรณีที่แบ่งระหว่างกลุ่ม 0 กับ กลุ่ม 1. ถ้าค่า  $y_k$  ใกล้กับ 1 ก็หมายความว่า มีความน่าจะเป็นของกลุ่ม 1 สูง การทำยกลุ่ม 1 จึงสมเหตุสมผล. . แต่ถ้าค่า  $y_k$  ใกล้กับ 0 ก็หมายความว่า มีความน่าจะเป็นของกลุ่ม 0 ซึ่งคือ  $1 - y_k$  มีค่าสูง ดังนั้นหากทำยกลุ่ม 0 ก็จะมีโอกาสสูงมากกว่า. การจำแนกประเภทแบบหลายกลุ่ม เราสามารถใช้รหัสหนึ่งไปเค (1-of-K coding) และใช้ฟังชั้นกระตุ้นเป็นฟังชั้นซอฟต์แม็กซ์ (softmax function) เช่นเดียวกับที่ยกในหัวข้อ 4.3.3. ฟังชั้นซอฟต์แม็กซ์  $y_k = \frac{\exp(a_k^{(2)})}{\sum_{q=1}^K \exp(a_q^{(2)})}$  ก็สามารถมองเป็นการตีความเออตพุต  $y_k$  ในเชิงความน่าจะเป็นได้ เช่นกัน. นั่นคือ  $y_k$  แทนค่าทำนายความน่าจะเป็นที่จะเป็นกลุ่ม  $k$ . สังเกตว่า  $0 \leq y_k \leq 1$  และ  $\sum_k y_k = 1$  ซึ่งไม่ได้ละเอียดคุณสมบติของความน่าจะเป็น. ตาราง 5.3 สรุปฟังชั้นกระตุ้นของชั้นเออตพุต สำหรับงานหลักๆ ของการประยุกต์ใช้โครงข่ายประสาทเทียม.

**เกร็ดความรู้เซลล์ประสาท** (เรียบเรียงจาก [33] และ [80]) สมองมนุษย์ประกอบเซลล์ชนิดต่างๆ มากมาย เช่น เส้นเลือด เซลล์เกลีย เซลล์ประสาท. เส้นเลือดทำหน้าที่รับส่งอากาศ น้ำ อาหาร. เซลล์เกลีย(neuroglia) ทำหน้าที่สนับสนุนต่างๆ รวมถึงการรักษาภาวะร่างดูด (homeostasis) เพื่อให้ภายในสมองมีสภาพที่เหมาะสม เช่น การควบคุมระดับความเข้มข้นของโซเดียมและแคลเซียมในอุ่น เป็นต้น. เซลล์ประสาททำหน้าที่หลักของสมอง ได้แก่ การควบคุมระบบการทำงานต่างๆ ในร่างกายให้เป็นปกติ รวมไปถึง การให้ความสามารถในการจำ การเรียนรู้ การคิด และการทำความเข้าใจ.

สมองมนุษย์มีเซลล์ประสาทอยู่ประมาณ แสนล้านเซลล์ เซลล์ประสาทเองก็มีอยู่หลายประเภท แต่โครงสร้างพื้นฐานมีลักษณะคล้ายๆ กัน นั่นคือเซลล์ประสาทแต่ละเซลล์มีใบประสาทเพื่อรับสัญญาณเข้าสู่เซลล์เรียกว่า денดrite(dendrite). สัญญาณต่างๆ ที่เข้าสู่เซลล์จะถูกนำมารวมกันที่นิวเคลียส และผลรวมของสัญญาณที่รับเข้ามา ทั้งสัญญาณกระตุ้นและสัญญาณยับยั้ง จะเป็นตัวตัดสินว่าเซลล์ประสาทนั้นจะอยู่ในสถานะถูกกระตุ้นหรือไม่. ถ้าเซลล์ประสาทอยู่ในสถานะถูกกระตุ้น มันจะส่งสัญญาณออกไปให้กับเซลล์ประสาทอื่นๆ ที่รับสัญญาณจากมัน ผ่านไปประสาทนำออกสัญญาณเรียกว่า axon. จุด

ต่อระหว่างแอกซอนของเซลล์ประสาทตัวหนึ่งกับเดนไดร็ตของเซลล์ประสาಥีกเซลล์หนึ่ง เป็นจุดประسانประสาทที่เรียกว่า ไซแนปส์ (synapse). แนวคิดพื้นฐานนี้เองที่ โรเซนแบล็อก นำไปสร้างโมเดลเพอร์เซปตรอน (ดูรูป 5.2 และ 5.7 ประกอบ) เมื่อเปรียบเทียบเพอร์เซปตรอนกับเซลล์ประสาท ผลลัพธ์ของอินพุตกับค่าน้ำหนักของเพอร์เซปตรอน ( $x_n \times [D]$  กับ  $w[D]$  ในรูป 5.3) เทียบได้กับ ความแรงของสัญญาณประสาทสัญญาณหนึ่งที่รับเข้ามาผ่านไซแนปส์แล้วกำลังเดินทางเข้าสู่นิวเคลียสของเซลล์ประสาท เพื่อไปรวมกับความแรงของสัญญาณประสาทสัญญาณอื่นๆที่รับเข้ามาผ่านไซแนปส์จุดอื่นๆ. ความแรงของสัญญาณประสาทสัญญาณหนึ่งที่รับเข้ามาผ่านไซแนปส์ จะขึ้นอยู่กับ สัญญาณที่ส่งมา (เปรียบเทียบกับ  $x[D]$ ) และ ความแข็งแรงในการเชื่อมต่อสัญญาณของไซแนปส์ (เปรียบเทียบกับ  $w[D]$ ).

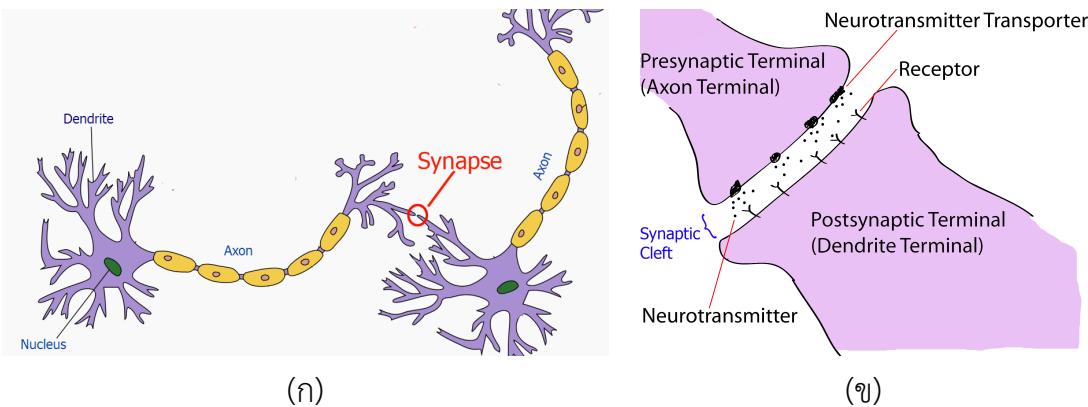
โดยเฉลี่ยแล้ว เซลล์ประสาทแต่ละเซลล์จะมีไซแนปส์ประมาณห้าพันจุด ซึ่งนั่นคือเมื่อรวมแล้ว ในสมองมนุษย์หนึ่งคนจะมีการเชื่อมต่อประสาทอยู่ราวๆห้าร้อยล้านล้านไซแนปส์. การรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทมีหลายกลไก เช่น กลไกทางเคมี (ผ่านสารสื่อประสาท) กลไกทางไฟฟ้า และ กลไกเชิงภูมิคุ้มกัน. แต่กลไกหลักของการส่งสัญญาณประสาทคือ กลไกทางเคมี ซึ่งคือการรับส่งสัญญาณประสาทระหว่างเซลล์ประสาทโดยดำเนินการผ่านสารสื่อประสาท (neurotransmitter). เซลล์ประสาทที่ส่งสัญญาณจะปล่อยสารสื่อประสาทออกมาผ่านโปรตีนที่ทำหน้าที่ส่งสารสื่อประสาท เรียกว่า ทรานสปอร์เตอร์ (neurotransmitter transporter) และเซลล์ประสาทที่รับสัญญาณจะรับสารสื่อประสาทเหล่านั้นเข้าไปผ่านโปรตีนที่ทำหน้าที่รับสารสื่อประสาท เรียกว่า รีเซปเตอร์ (receptor).

เมื่อรีเซปเตอร์รับสารสื่อประสาทเข้ามา นั่นคือ โครงสร้างของสารสื่อประสาทจับกับโครงสร้างของรีเซปเตอร์ แล้วทำให้ กลไกของรีเซปเตอร์เปิดทำงาน โมเลกุลที่จับกับรีเซปเตอร์ จะเรียกว่า ลิกเอนด์ (ligand). กลไกของการจับตัวระหว่างรีเซปเตอร์ กับลิกเอนด์นี้ จะเป็นกลไกในลักษณะแม่คุณแจ-ลูกคุณแจ. นั่นคือ โครงสร้างของรีเซปเตอร์แต่ละชนิดจะจับตัวได้เฉพาะกับลิกเอนด์ที่มีโครงสร้างที่เข้ากันได้เท่านั้น เช่น สารสื่อประสาทอาเซติทิลโคลีน (acetylcholine) ซึ่งเป็นสารสื่อประสาทที่เซลล์ประสาทใช้ต่อกระตุ้นเซลล์กล้ามเนื้อ จะจับกับรีเซปเตอร์สำหรับอาเซติทิลโคลีนได้เท่านั้น และ รีเซปเตอร์สำหรับสารสื่อประสาทตัวอื่น ก็ไม่อาจจับกับอาเซติทิลโคลีนได้เช่นกัน. การเข้าใจกลไกการทำงานลักษณะนี้ ช่วยให้เกสัชศาสตร์สามารถออกแบบตัวยาที่เฉพาะเจาะจงกับสารสื่อประสาทเฉพาะตัวได้ เช่น ยาต้านอาการเครียด ฟลูโอดีติน (Fluoxetine) ที่เฉพาะเจาะจงกับสารสื่อประสาทเซอโรโทนิน(Serotonin).

หมายเหตุ เซลล์ประสาทแต่ละชนิดจะมีลักษณะเฉพาะตัวต่างกันและจะทำงานกับสารสื่อประสาทเฉพาะชนิด เช่น เซลล์ประสาทเซอโรโทนิน (serotonin neurons) ที่อยู่บริเวณดอร์ซอฟราฟีนเคลลีส (dorsal raphe nucleus) ของก้านสมอง จะทำงานกับสารสื่อประสาทเซอโรโทนิน[48], เซลล์ประสาทซีอี1พีรามิดอล (CA1 pyramidal neurons) ที่อยู่บริเวณซีอี1 (CA1) ของชีปไปแคมป์ส จะทำงานกับสารสื่อประสาทกลูตาเมท(Glutamate)[57], เซลล์ประสาทมิดเบรนโดยพามิโนจิก (midbrain dopaminergic neurons) ที่อยู่หลายابริเวณรวมถึง พื้นที่เวนทรอලเทกเมนทอล (ventral tegmental area) ในสมองส่วนกลาง จะทำงานกับสารสื่อประสาทโดยพามีน(Dopamine)[57]. เซลล์ประสาทบางชนิดทำงานกับสารสื่อประสาทมากกว่าหนึ่งชนิด เช่น เซลล์ประสาทนามกลาง (medium spiny neurons) ที่อยู่บริเวณบากอลแกงเกลีย ส่งสัญญาณออกผ่านกาบา (GABA) แต่สามารถรับสัญญาณผ่านสารสื่อประสาทหลายชนิดรวมถึงกลูตาเมทและโดยพามีน.

## 5.3 การฝึกโครงข่าย

เมื่อเรามีโครงข่ายประสาทเทียมแล้ว สิ่งสำคัญที่จะทำให้โครงข่ายสามารถทำงานตามที่เราต้องการได้ก็คือ การใช้ค่าพารามิเตอร์ที่เหมาะสม. สำหรับโครงข่ายประสาทเทียม, เราจะเรียกการหาค่าพารามิเตอร์หรือ ค่าน้ำหนักที่เหมาะสมว่า การฝึกโครงข่าย (network training). เราจะฝึกโครงข่าย ด้วยข้อมูลตัวอย่าง ซึ่งประกอบด้วย อินพุตและเอาต์พุตตัวอย่าง  $N$  จุดข้อมูล  $\{\mathbf{x}_n\}$ ,  $\{\mathbf{t}_n\}$ ,  $n = 1, \dots, N$  ตามลำดับ. สิ่ง



รูปที่ 5.7: ภาพแสดงเซลล์ประสาทซึ่งมีต่อสัญญาณกันผ่านไซแนปซ์ (synapse) โดยสัญญาณที่สื่อสารกันนั้นทำโดยผ่านกลไกของสารสื่อประสาท(neurotransmitter) (ก) เซลล์ทางชั้ยมีส่งสัญญาณผ่านแอกซอน เชื่อมเข้าสู่อีกเซลล์ที่ไซแนปซ์ เพื่อผ่านเดนไดรต์ไปสู่นิวเคลียสของเซลล์ทางชั้ยมีอ (ภาพดัดแปลงจาก [http://en.wikipedia.org/wiki/File:Neuron\\_Hand-tuned.svg](http://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg)) (ข) ภาพขยายส่วนของไซแนปซ์ ซึ่งส่วนปลายของแอกซอน (presynaptic terminal) จะส่งสารสื่อประสาทอกมา และส่วนปลายของเดนไดรต์ (postsynaptic terminal) จะรับสารสื่อประสาทเข้าไป

ที่เราต้องการคือ นำนายค่าเออต์พุต  $\mathbf{y}$  จากอินพุตตัวอย่าง  $\mathbf{x}_n$  ให้ค่าใกล้เคียงกับ เออต์พุตตัวอย่าง  $\mathbf{t}_n$  มากที่สุด หรือ พุดอีกอย่างคือ ถ้าเราทำการหาค่าตัดตอน การฟีกโครงข่าย ก็คือ การหาค่าน้ำหนัก  $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}\}$  (เมื่อ  $L$  คือจำนวนชั้นของโครงข่าย) ที่ทำให้พึงชั้นความผิดพลาดมีค่าน้อยที่สุด. นั่นคือ  $\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$  เมื่อ

$$E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \boldsymbol{\theta}) - \mathbf{t}_n\|^2 \quad (5.19)$$

ซึ่งเมื่อแทนพึงชั้นของโมเดลที่ใช้เข้าไป เราจะสามารถใช้วิธีการหาค่าน้ำหนัก  $\boldsymbol{\theta}$  ได้. ตัวอย่างเช่น หากเราใช้โครงข่ายจ่ายไปข้างหน้า ขนาด 2 ชั้น, โมเดลก็คือ

$$y_k = w_{k0}^{(2)} + \sum_{m=1}^M w_{km}^{(2)} \cdot h \left( w_{m0}^{(1)} + \sum_{d=1}^D w_{md}^{(1)} x_d \right).$$

เพื่อความสะดวก เราจะพิจารณา ค่าผิดพลาดที่จุดข้อมูลที่  $n$  เพียงจุดเดียวก่อน, เราจะได้

$$E_n = \frac{1}{2} \sum_{k=1}^K \{y_k - t_{k,n}\}^2 \quad (5.20)$$

$$= \frac{1}{2} \sum_{k=1}^K \left\{ w_{k0}^{(2)} + \sum_{m=1}^M w_{km}^{(2)} \cdot h \left( w_{m0}^{(1)} + \sum_{d=1}^D w_{md}^{(1)} x_{d,n} \right) - t_{k,n} \right\}^2 \quad (5.21)$$

เมื่อ  $x_{d,n}$  และ  $t_{k,n}$  คือ อินพุตมิติที่  $d$  และ เออต์พุตมิติที่  $k$  ของจุดข้อมูลที่  $n$  ตามลำดับ.

ตั้งนั้นค่าอนุพันธ์ของค่าผิดพลาด เมื่อเทียบกับค่าน้ำหนักชั้นที่ 2 คือ

$$\frac{\partial E_n}{\partial w_{j0}^{(2)}} = y_{j,n} - t_{j,n} \quad (5.22)$$

$$\frac{\partial E_n}{\partial w_{ji}^{(2)}} = (y_{j,n} - t_{j,n}) \cdot z_i^{(1)}, \quad \text{เมื่อ } i = 1, \dots, M \quad (5.23)$$

และ  $z_i^{(1)} = h\left(w_{i0}^{(1)} + \sum_{d=1}^D w_{id}^{(1)} x_{d,n}\right)$ .

ในลักษณะเดียวกัน ค่าอนุพันธ์ของค่าผิดพลาดเมื่อเทียบกับค่าน้ำหนักชั้นที่ 1 คือ

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \sum_{k=1}^K (y_{k,n} - t_{k,n}) \cdot \frac{\partial y_{k,n}}{\partial w_{ji}^{(1)}} \quad (5.24)$$

$$= \sum_{k=1}^K (y_{k,n} - t_{k,n}) \cdot w_{kj}^{(2)} \cdot \frac{\partial h\left(w_{j0}^{(1)} + \sum_{i=1}^D w_{ji}^{(1)} x_{i,n}\right)}{\partial w_{ji}^{(1)}} \quad (5.25)$$

$$= h'(a_j^{(1)}) \frac{\partial a_j^{(1)}}{\partial w_{ji}^{(1)}} \cdot \sum_{k=1}^K (y_{k,n} - t_{k,n}) \cdot w_{kj}^{(2)} \quad (5.26)$$

เมื่อ  $a_j^{(1)} = w_{j0}^{(1)} + \sum_{i=1}^D w_{ji}^{(1)} x_{i,n}$  และ  $h'(a)$  คือ อนุพันธ์ของ  $h(a)$ . สำหรับ  $h(a) = \frac{1}{1+\exp(-a)}$ , อนุพันธ์  $h'(a) = h(a) \cdot (1 - h(a))$ .

ซึ่งสุดท้าย เราจะได้

$$\frac{\partial E_n}{\partial w_{j0}^{(1)}} = h'(a_j^{(1)}) \cdot \sum_{k=1}^K (y_{k,n} - t_{k,n}) \cdot w_{kj}^{(2)} \quad (5.27)$$

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = h'(a_j^{(1)}) \cdot x_{i,n} \cdot \sum_{k=1}^K (y_{k,n} - t_{k,n}) \cdot w_{kj}^{(2)} \quad \text{เมื่อ } i = 1, \dots, D. \quad (5.28)$$

จากสมการ 5.22, 5.23, 5.27, and 5.28, ค่าเราสามารถใช้ความรู้จากศาสตร์การหาค่าตีที่สุด เช่น วิธีลงชั้นที่สุด (รายการ 2.3) ในการปรับปรุงค่าน้ำหนักได้,

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \alpha \cdot \frac{\partial E_n}{\partial w_{ji}^{(l)}} \quad (5.29)$$

โดย ค่าของน้ำหนักใหม่ (เทอมทางซ้ายมือ) จะคำนวณจากค่าน้ำหนักเดิม (เทอมแรกทางขวา) ลบค่าอนุพันธ์ของเป้าหมาย ที่คูณด้วย  $\alpha$  ที่ทำหน้าที่เป็นชั้นก้าว(step size) หรือ เรียกว่า อัตราการเรียนรู้ (learning rate) เพื่อรักษาให้อัลกอริทึมลู่เข้า, ดังที่ได้ถูกในหัวข้อ 2.1.

ตัวอย่างและฟังชันความผิดพลาด  $E(\theta)$  ที่เราถูกกันในหัวข้อนี้ หมายความว่า ค่าที่ได้จากการคำนวณของตัวอย่างที่ได้ออกมาต้องใกล้เคียงกับค่าที่เราตั้งไว้ ดังที่ได้อธิบายในบท 4 ว่า สำหรับงานการจำแนกประเภท 2 กลุ่ม การใช้ฟังชันความผิดพลาด (error function) หรือ อาจเรียกว่า ฟังชันเป้าหมาย objective function หรือ ฟังชันค่าใช้จ่าย cost function)

$$E_n = -t_n \log(y) - (1 - t_n) \log(1 - y) \quad (5.30)$$

จะช่วยให้โมเดลทำงานได้ดีกว่า. สำหรับงานการจำแนกประเภทแบบหลายกลุ่ม เราจะใช้การใช้ฟังชันความผิดพลาด

$$E_n = -\sum_k t_{kn} \log(y_k). \quad (5.31)$$

**การฝึกแบบออนไลน์กับแบบอффไลน์** การฝึกโครงข่ายโดยปรับปรุงค่าน้ำหนัก โดยทำการคำนวณจากจุดข้อมูลที่ลงทะเบียน (เช่น สมการ 5.32) จะเรียกว่า เป็นการฝึกแบบออนไลน์ (online training) หรือ การฝึกแบบส่วนเพิ่ม (incremental mode). นอกจากการฝึกแบบออนไลน์แล้ว เราสามารถที่จะรวมผลค่าอนุพันธ์สำหรับทุกๆ จุดข้อมูล และปรับปรุงค่าน้ำหนักที่ได้รับได้,

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \alpha \cdot \sum_{n=1}^N \frac{\partial E_n}{\partial w_{ji}^{(l)}} \quad (5.32)$$

เมื่อ  $N$  คือ จำนวนจุดข้อมูล. เราจะเรียกการฝึกแบบนี้ว่า การฝึกแบบอффไลน์ (offline training) หรือ การฝึกแบบกลุ่ม (batch mode).

ในทางปฏิบัติ ลำดับของข้อมูลจะถูกสลับในแต่ละรอบฝึกทั้งในการฝึกแบบอฟฟไลน์และออนไลน์ เพื่อคุณภาพของการฝึกที่ดี (ลดความเสี่ยงที่น้ำหนักจะไปติดกับค่าที่ไม่ตั้งแต่แรกและจะปรับเปลี่ยนยากในภายหลัง).

การเลือกรห่วงของการฝึกแบบออนไลน์กับแบบอฟฟไลน์ เอย์กิน[34]ได้อธิบายถึงข้อดีข้อเสียดังนี้. การฝึกแบบอฟฟไลน์ จะให้การประเมินค่าเกรดเฉลี่ยที่แม่นยำ ดังนั้นจึงรับประทานว่า เมื่อทำงานกับวิธีลงเกรดเฉลี่ยแล้วจะได้น้ำหนักที่เป็นค่าตัวทำน้อยที่สุด(ห้องถัน). นอกจากนั้น การฝึกแบบอฟฟไลน์ยังเป็นการคำนวณแบบขนาด (และสามารถนำศาสตร์และเทคโนโลยีการประมวลผลแบบขนาดมาช่วยได้) และ เมื่อมองจากมุมมองทางสถิติศาสตร์ การฝึกแบบอฟฟไลน์ก็เป็นรูปแบบการอนุมานทางสถิติ(statistical inference)อย่างหนึ่ง ดังนั้นการฝึกแบบอฟฟไลน์จะเหมาะสมกับงานการหาค่าถดถอย. แต่ข้อเสียของการฝึกแบบอฟฟไลน์ ก็คือความต้องการใช้หน่วยความจำบริมาณมาก (เพราะว่าใช้ข้อมูลทั้งหมด ในการคำนวณแต่ละยก).

ข้อดีของการฝึกแบบออนไลน์เมื่อเทียบกับการสลับลำดับของข้อมูลแต่ละรอบฝึกจะช่วยลดความเสี่ยงในการเข้าไปติดอยู่ในค่าตัวทำต่ำสุดห้องถันได้ และเมื่อเทียบกับการฝึกแบบอฟฟไลน์ การฝึกแบบออนไลน์จะต้องการใช้หน่วยความจำบริมาณน้อยกว่า. นอกจากนั้นในทางปฏิบัติแล้วพบว่า การฝึกแบบออนไลน์สามารถใช้ประโยชน์จากข้อมูลที่ชำชองกันได้มากกว่าการฝึกแบบอฟฟไลน์ และการฝึกแบบออนไลน์ยังช่วยให้สามารถติดตามการเปลี่ยนแปลงเล็กๆ ในข้อมูลได้ด้วย โดยเฉพาะอย่างยิ่งเมื่อกระบวนการที่อยู่เบื้องหลังข้อมูล (กระบวนการที่สร้างข้อมูล) เป็นกระบวนการที่ไม่คงที่ (nonstationary). แม้ว่า การฝึกแบบออนไลน์จะทำงานได้ช้ากว่า แต่การฝึกแบบออนไลน์มีการใช้อย่างแพร่หลายโดยเฉพาะกับงานการจำแนกรูปแบบ (pattern-classification task) เพราะว่า การฝึกแบบออนไลน์นั้นนำไปสร้างหรือเขียนโปรแกรมได้ง่ายกว่า และ สามารถขยายขึ้นไปทำงานกับข้อมูลขนาดใหญ่และใช้กับปัญหาการจำแนกรูปแบบยากๆได้.

**รอบฝึก (epoch)** เมื่อจากวิธีการฝึกจะเป็นการคำนวนซ้ำหลายครั้ง หรือรอบ. เราจะเรียก การปรับปรุงค่า  $\theta$  ที่หนักครบทุกตัวในแต่ละยกว่าเป็น 1 รอบฝึก (epoch). โดยทั่วไป แล้วเราต้องทำการฝึกหลายรอบ ก่อนที่ค่า  $\theta$  ที่หนักจะถูกลำเลี้ยดค่าทำต่ำสุด. จำนวนรอบฝึกที่ต้องทำขึ้นกับหลายปัจจัย เช่น ความซับซ้อนของปัญหา, ขนาดและความซับซ้อนของโครงข่าย, อัตราการเรียนรู้, ค่าเริ่มต้น และ วิธีการที่ใช้ฝึก. ดูหัวข้อ 7.4 เพิ่มเติม สำหรับคำแนะนำวิธีการเลือกจำนวนรอบฝึก.

## 5.4 การแพร่กระจายย้อนกลับ

เราได้เห็นภาพรวมของการฝึกโครงข่ายแล้ว ซึ่งหัวใจของการฝึกโครงข่ายก็คือการคำนวนค่าอนุพันธ์ของเป้าหมาย. สังเกตว่า รูปแบบของอนุพันธ์ที่ได้นั้น ขึ้นอยู่กับชั้นของค่าหนัก เนพาเจาจะจงกับโครงข่าย 2 ชั้นเท่านั้น และ มีการคำนวนที่คล้ายๆ กันสำหรับอนุพันธ์แต่ละตัว. หัวข้อนี้อธิบายวิธีที่มีประสิทธิภาพมากขึ้นในการคำนวนค่าอนุพันธ์ของเป้าหมาย  $\frac{\partial E}{\partial w_{ji}}$  สำหรับแต่ละ  $w_{ji}$ . หมายเหตุ ในกรณีที่บริบทชัดเจน ด้วย  $\cdot_n$  (ระบุจุดข้อมูล) และ ด้วย  $\cdot_1$  (ระบุชั้นของโครงข่าย) อาจจะถูก略ไว้ เพื่อความกระชับ และเรียบง่ายของนิพจน์ทางคณิตศาสตร์.

หากสังเกตว่า พังชั้นเป้าหมาย  $E_n$  จะขึ้นกับค่าหนัก  $w_{ji}$  โดยผ่านตัวแปรการกระตุ้น  $a_j$  เท่านั้น, ดังนั้น จากโครงสร้างของโครงข่ายและกฎเชิงโซ่อิเล็กทรอนิกส์ (chain rule), เราจะได้

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ji}} \quad (5.33)$$

และเพื่อความสะดวก เรากำหนดให้

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j}. \quad (5.34)$$

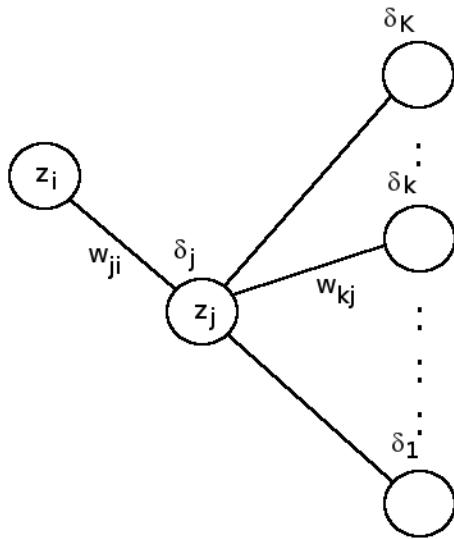
จาก  $a_j = \sum_i w_{ji} z_i$  เมื่อเรากำหนด  $z_0 = 1$ , เราจะได้

$$\frac{\partial a_j}{\partial w_{ji}} = z_i. \quad (5.35)$$

เมื่อแทนสมการ 5.34 และ 5.35 ลงในสมการ 5.33, ได้

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i. \quad (5.36)$$

สมการ 5.36 บอกว่า เราสามารถคำนวนค่าอนุพันธ์ของพังชั้นเป้าหมายได้จาก การคูณค่า  $\delta_j$  ของผู้ถูกต้องของค่า  $w_{ji}$  กับค่า  $z_i$  ของผู้ตั้นทางของค่า  $w_{ji}$ , ดูรูป 5.8 ประกอบ. ซึ่งหากเราถูกละ  $\delta_j$ , ค่าอนุพันธ์ของพังชั้นเป้าหมายก็สามารถคำนวนได้อย่างง่ายดาย.



รูปที่ 5.8: แสดงตัวแปร ในการทำการแพร่กระจายย้อนกลับ (backpropagation).

พิจารณา  $\delta_k$  ของหน่วยเอาต์พุต, เราจะได้

$$\delta_k = y_k - t_k. \quad (5.37)$$

ส่วน  $\delta_j$  ของหน่วยซ่อน, เมื่อใช้กฎลูกโซ่ เราจะได้

$$\delta_j = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \cdot \frac{\partial a_k}{\partial a_j} \quad (5.38)$$

เมื่อ  $a_k$  แทนค่าการกระตุ้นของหน่วยอยู่ต่างๆ ที่หน่วยซ่อนของ  $a_j$  เชื่อมต่อผ่านไปสู่เอาต์พุต. รูป 5.8 แสดงหน่วยอยู่และ การเชื่อมต่อ. สังเกตุ หน่วยอยู่ที่  $j$  (วงกลมกลางรูป) ต่อผ่านหน่วยอยู่ 1 ถึง  $K$  (วงกลมต่างๆ ทางขวา) ไปสู่เอาต์พุต.

เมื่อ แทนสมการ 5.37, 5.4, และ 5.5 ลงในสมการ 5.38 จะได้

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k. \quad (5.39)$$

สมการ 5.39 คือ สมการการแพร่กระจายย้อนกลับ (backpropagation), ซึ่ง บอกว่า ค่าของ  $\delta$  ของแต่ละหน่วยอยู่ หาได้จากการแพร่กระจายย้อนกลับของ  $\delta$  ต่างๆ จากหน่วยอยู่ที่อยู่ลีกลงไป (ใกล้เอาต์พุต). อัลกอริทึมแพร่กระจายย้อนกลับ (error backpropagation หรือ เรียกย่อๆ backpropagation) สรุปขั้นตอนการคำนวณต่างๆ ดังนี้

- 1. ทำการคำนวณไปข้างหน้า (forward propagation), สมการ 5.5 และ 5.4;
- 2. คำนวณค่า  $\delta_k$  สำหรับทุกๆ หน่วยเอาต์พุต, สมการ 5.37;

- 3. ทำแพร่กระจายย้อนกลับค่า  $\delta$  ต่างๆ เพื่อคำนวณ  $\delta_j$ ; ทุกหน่วยอยู่, สมการ 5.39;
- 4. คำนวณค่าอนุพันธ์เป้าหมาย, สมการ 5.36.

อัลกอริทึมแพร่กระจายย้อนกลับ เป็นวิธีการคำนวณค่าอนุพันธ์ของฟังชันจุดประสีกสำหรับโครงข่ายประชาทเทียมที่มีประสิทธิภาพ และครอบคลุมการคำนวณสำหรับโครงข่ายประชาทเทียมที่มีจำนวนชั้นหลายชั้นได้. ค่าอนุพันธ์ของฟังชันเป้าหมายนี้สามารถนำไปใช้ประกอบกับวิธีการหาค่าตัวทำดีที่สุดเพื่อหาค่าน้ำหนักที่เหมาะสมสำหรับโครงข่ายประชาทเทียมได้. บทที่ 6 อภิปรายตัวอย่างการประยุกต์ใช้ทฤษฎีของโครงข่ายประชาทเทียมที่ถูกในบทนี้ รวมถึงนำเสนอตัวอย่างโปรแกรมภาษาอาาร์โปรเจกท์ที่นำคณิตศาสตร์ที่พัฒนาขึ้นมาใช้ไปสร้างเป็นโปรแกรม.

#### เก็ตความรู้จิตและการเรียนรู้ (เรียบเรียงจาก [80], [2], และ [62])

“ทุกสิ่งเริ่มที่จิต นำโดยจิต และสร้างโดยจิต” ธรรมบท[2]

จิตคือสภาวะเชิงการรับรู้และเชิงสติปัญญา ซึ่งรวมถึง สติรู้ตัว การรับรู้สัมผัส ความรู้สึก อารมณ์ การคิด การตัดสินใจ การจำ การรับประสีก การเรียนรู้ และการตอบสนองต่อสภาวะแวดล้อม. ส่วนประกอบของจิตนั้น อาจจัดออกได้เป็น 4 หมวด. หมวดหนึ่ง วิญญาณ(vijnana) ซึ่งคือสติรู้ตัว(consciousness). หมวดสอง สัญญา (samjana) ซึ่งคือการรับรู้(perception) ผ่านสัมผัสทางการมองเห็น สัมผัสทางการได้ยิน สัมผัสทางการได้กลิ่น สัมผัสการรับรส สัมผัสทางกาย และสัมผัสที่มาจากการจิตเอง. จิตอาจมีสัมผัส เช่น กัน ดังตัวอย่างของการแขนงหลวง (phantom limbs) ที่เกิดในผู้ที่เสียแขนขาไป แต่ภายนอกยังรู้สึก คันหรือเจ็บที่แขนหรือขาที่ไม่มีอยู่. แม้ว่าสาเหตุของอาการนี้ยังไม่มีสาเหตุที่ทางการแพทย์ยอมรับร่วมกันอย่างกว้างขวาง แต่ นายแพทย์วิลัยานุร รามาฉันทรัน (Vilayanur S. Ramachandran) อธิบายว่า สัมผัสที่รู้สึกนั้นมาจากการส่วนที่เคยทำงาน กับแขนหรือขาส่วนนั้นขาดสัญญาณที่เคยได้รับ และอาจส่งสัญญาณออกมากทั้งๆที่ไม่ได้รับสัญญาณรับรู้จริงๆ. จากทฤษฎีนั้น นายแพทย์รามาฉันทรันเสนอวิธีการบำบัดอาการแขนงหลวง โดยออกแบบกระบวนการให้ผู้มีอาการได้ฝึกประสาทรับรู้ใหม่ ซึ่งพบว่าได้ผลดีมาก. หากมองจากมุมมองทางวิศวกรรม สัญญาณที่ขาดหายไปจากแขนขาที่เสียไปนั้น อาจให้ผลในลักษณะคล้าย การทิ่งจรวดไฟฟ้ารับอินพุตมาจากข้อปลายที่ปล่อยลอยอยู่ ซึ่งข้อปลายที่ปล่อยลอยอาจรับสัญญาณรบกวน(noise)เข้ามาแทนได้ การฝึกประสาทรับรู้ใหม่ ก็อาจคล้ายการต่อข้อปลายนั้นลงกราวด์(ground)เพื่อปิดสัญญาณรบกวนนั้นลง. หมวดสาม เวทนา (vedana) ซึ่งคือความรู้สึก(feeling) อารมณ์ ความชอบ ความไม่ชอบ ความวางแผน และ หมวดสี่ ลังชา (sankhara) ซึ่งคือการคิดและกระบวนการเชิงสติปัญญาอีก (mental activity) ได้แก่ การตัดสินใจ การตอบสนองต่อสภาวะแวดล้อม การจำ การรับประสีก และการเรียนรู้.

นักประสาทวิทยาด้านสติปัญญาชั้นนำ รีเบกก้า แซ็ก[72] เสื่อว่า รูปแบบของกิจกรรมทางไฟฟ้าของเซลล์ประสาทเกี่ยวข้องโดยตรงกับจิตของเรา เพียงแต่ว่าการวิทยาศาสตร์ยังไม่รู้อะไรมากเกี่ยวกับความสัมพันธ์ระหว่างรูปแบบของกิจกรรมทางไฟฟ้า และการทำงานของจิต.

จากมุมมองของกิจกรรมทางไฟฟ้า การเรียนรู้ก็เหมือนการปรับเปลี่ยนวงจรหรือเปลี่ยนการเชื่อมต่อภายในวงจร ซึ่งส่งผลให้เกิดการปรับเปลี่ยนพฤติกรรมของกิจกรรมทางไฟฟ้า. สภาพพลศาสติกของระบบประสาท (synaptic plasticity) คือความสามารถของระบบประสาทที่สามารถเพิ่มหรือลดความแข็งแรงของการเชื่อมต่อสัญญาณประสาทระหว่างเซลล์ได้. สภาพพลศาสติกของระบบประสาทนี้เชื่อว่า เป็นคุณสมบัติที่อยู่เบื้องหลังความสามารถในการจำและการเรียนรู้ของสมอง. กลไกนี้ เปรียบเทียบได้กับการปรับค่าน้ำหนักหรือการฝึกโครงข่ายประชาทเทียม แต่ประเด็นหนึ่งที่ต่างกันก็คือ การเปลี่ยนค่าน้ำหนักของโครงข่ายประชาทเทียมจะทำเฉพาะในขั้นตอนการฝึกโครงข่ายประชาทเทียม และค่าน้ำหนักที่ดีแล้วจะถูกตรึงให้คงค่าเหล่านั้น

ไวรังที่ขณะใช้งาน. แต่ระบบประสาท(ทางชีวภาพ)จะเปลี่ยนแปลงตัวเองตลอดเวลา เปลี่ยนขณะเรียนรู้ เปลี่ยนขณะคิด เปลี่ยนขณะทำกิจกรรมต่างๆ เปลี่ยนขณะทำงาน เปลี่ยนขณะไม่ได้ทำงาน เปลี่ยนขณะเล่น เปลี่ยนขณะทำสิ่งที่มีประโยชน์ เปลี่ยนขณะพักผ่อน เปลี่ยนขณะนอนหลับ และที่สำคัญเปลี่ยนแม้มั่ต่อขณะทำสิ่งที่เป็นโทษ เช่น เมื่อสิ่งที่เราคาดหวังไม่ได้ดึงใจ แล้วเราไม่ชอบใจ ถ้าเราเลือกที่จะกรร สมองจะเรียนรู้การตอบสนองนั้น และ เมื่อเราทำแบบนั้นบ่อยๆ เราเก็บกากลายเป็นคนที่กรรจ่ายหรือกล่าวอย่างหัดเงินก็คือ เราฝึกสมองของเราให้ก่งที่จะอยู่ในสภาพภาวะอารมณ์กรรนั้นเอง.

กลไกเบื้องหลังการส่งสัญญาณของระบบประสาท กล่าวโดยคร่าวๆ ก็คือ เมื่อสัญญาณจากนิวเคลียสของเซลล์ประสาทดินทางถึงปลายแอกซอนซึ่งเป็นปลายสำหรับส่งสัญญาณออก สัญญาณซึ่งถ่ายทอดมาในรูปความต่างศักดิ์จะทำให้ช่องไอออนที่ควบคุมด้วยแรงดันไฟฟ้า(voltage-gated ion channel) ของปลายแอกซอนเปิดออก ทำให้แคลเซียมไอออน (calcium ion สัญลักษณ์  $Ca^{2+}$ ) ซึ่งอยู่ในของเหลวรอบเซลล์ ไหลเข้าสู่ปลายแอกซอน. แคลเซียมไอออนที่เข้าสู่ปลายแอกซอนจะทำปฏิกิริยากับโปรตีนและเอนไซม์ภายในแอกซอน ซึ่งส่งผลให้เกิดการปล่อยสารสื่อประสาทออกมา. สารสื่อประสาทที่ออกมานี้จะมีบางโมเลกุลที่ได้จับกับรีเซปเตอร์ที่ปลายเดนไดร์ต ซึ่งปลายเดนไดร์ตคือปลายประสาทนิ่นนำสัญญาณเข้าสู่เซลล์ประสาทตัวที่จะรับสัญญาณ. เมื่อรีเซปเตอร์จับกับสารสื่อประสาทแล้ว จะทำให้รีเซปเตอร์เปลี่ยนโครงสร้างซึ่งจะเปิดช่องให้ไอออนบวกไหลเข้าสู่เดนไดร์ต เมื่อไอออนบวกไหลเข้าสู่เดนไดร์ตจะทำให้ความต่างศักดิ์ของเดนไดร์ตจุดนั้นเปลี่ยนไป ซึ่งความต่างศักดินี้เองเป็นสัญญาณที่ถ่ายทอดต่อไปยังนิวเคลียสของเซลล์ประสาท.

สารสื่อประสาทเป็นกลไกหลักในการช่วยส่งสัญญาณประสาทข้ามเซลล์ประสาท แต่ตัวสารสื่อประสาทเองไม่ได้ถูกส่งเข้าไปในเดนไดร์ตของเซลล์ประสาทด้วยรับ. มันทำหน้าที่เสนอช่วยเปิดประตูให้ไอออนบวกได้เข้าไปในปลายเดนไดร์ตดังที่ได้อธิบายไปข้างต้น. หลังจากสารสื่อประสาทจับกับรีเซปเตอร์ได้สักพัก มันจะหลุดออกมานะ. สารสื่อประสาททั้งที่พึงหลุดมาจากการจับกับรีเซปเตอร์และที่ยังไม่ได้จับกับรีเซปเตอร์เลยจะถูกกำจัดออกไปโดยกลไกหลายชนิด เช่น การทำลายทิ้ง หรือ การที่ปลายแอกซอนดึงสารสื่อประสาทเหล่านี้กลับเข้าไปภายในเพื่อนำกลับไปใช้ใหม่.

การเรียนรู้หรือการปรับความแข็งแรงของการเชื่อมต่อสัญญาณระหว่างเซลล์ประสาท เกี่ยวข้องกับการปรับความสามารถในการรับส่งสัญญาณประสาทระหว่างเซลล์. เมื่อกล่าวถึงสัญญาณประสาทโดยละเอียดขึ้น สัญญาณประสาทจะถูกส่งในรูปแบบ ขณะที่สัญญาณประสาทส่งผ่านเส้นทางจากนิวเคลียสของเซลล์ประสาทตัวหนึ่งไปสู่นิวเคลียสของเซลล์ประสาทอีกด้วย หนึ่ง มีการเปลี่ยนรูปแบบอย่างหลายครั้ง. เมื่อเซลล์ประสาಥอยู่ในสถานะถูกระดุ้น นิวเคลียสของเซลล์จะส่งสัญญาโนอกมานในรูปความถี่ของพัลส์. นั่นคือ นิวเคลียสของเซลล์ประสาทจะส่งสัญญาโนในลักษณะพัลส์(pulse) ซึ่งมักเรียกว่าศักยะงาน (action potential) เช่น ค่าความต่างศักย์ภายในเซลล์ประสาทจะมีค่าประมาณ  $-70$  มิลลิโวลต์เมื่อเทียบกับจุดภายนอกเซลล์ แต่เมื่อมีศักยะงานเกิดขึ้น ค่าความต่างศักย์นี้เพิ่มขึ้นอย่างรวดเร็วจากค่าพักตัวที่ประมาณ  $-70$  มิลลิโวลต์ ไปสูงสุดที่ประมาณ  $40$  มิลลิโวลต์ หลังจากนั้นจะลดค่าลงอย่างรวดเร็วมาที่ราบ  $-90$  มิลลิโวลต์ และกลับมาจบที่ค่าพักตัวที่ประมาณ  $-70$  มิลลิโวลต์ โดยที่ศักยะงานแต่ละลูกจะนานประมาณ  $4$  มิลลิวินาที. ความถี่ที่เรือจำนานศักยะงานต่อวินาทีจะขึ้นกับความแรงของสถานะการกระตุ้นของเซลล์ เช่น เซลล์โอลแฟกตอร์ีคอร์ทิกซ์ไฟฟ้าเซลล์ประสาท โดยใช้ตุ้มน้ำหนักดึงกล้ามเนื้อของกบและวัดสัญญาโนไฟฟ้าจากเนื้อเยื่อประสาทของมัน และพบความสัมพันธ์ที่ชัดเจน ระหว่างค่าน้ำหนักที่ยืดกล้ามเนื้อออก (ตัวแทนของปริมาณความแรงของการกระตุ้น) กับความถี่ของศักยะงานที่เกิดขึ้น. ศักยะงานนี้คือสัญญาโนที่ถูกส่งออกจากนิวเคลียสผ่านไปที่ปลายแอกซอน สัญญาโนที่ส่งผ่านออกจากการปล่อยแอกซอนของเซลล์ตัวส่งเข้าไปสู่ปลายเดนไดร์ตของเซลล์ตัวรับจะส่งผ่านกลไกของสารสื่อประสาท และปลายเดนไดร์ตรับสัญญาโนประสาทเข้ามาในรูประดับความต่างศักย์ระหว่างภายในปลายเดนไดร์ตและภายนอก ซึ่งรูปแบบที่เปลี่ยนไปของสัญญาโนประสาทจากนิวเคลียสของตัวส่งไปจนถึงปลายแอกซอน(ในรูปศักยะงาน) ผ่านไปแบบ (ในรูปสารสื่อประสาท) และรับเข้าสู่ปลายเดนไดร์ตไปจึงส่งเข้าไปสู่นิวเคลียสของตัวรับ (ในรูประดับความแรงของความต่างศักย์) เกี่ยวข้องสัมพันธ์กับทฤษฎีที่อธิบายกระบวนการการเรียนรู้เซลล์ประสาท กล่าวถึง กลไกหลัก 2 กลไก. นั่น

คือ การเพิ่มความแข็งแรงเชิงประสาทระยะยาว (long-term synaptic potentiation คำย่อ LTP) และ การลดถอยความแข็งแรงเชิงประสาทระยะยาว (long-term synaptic depression คำย่อ LTD). จากรายงานศึกษาการเขื่อมต่อของเซลล์ในสิบไปแคมปัสโดยเฉพาะไซแนปส์ที่เขื่อมต่อระหว่างเซลล์ประสาทในบริเวณซีเอช3 (CA3) ที่ส่งแอกซอน (ซึ่งเรียกว่า แซฟเฟอร์คอลเลตเตอร์อล Schaffer collaterals) ไปเขื่อมต่อกับเซลล์ประสาทในบริเวณซีเอช1 สรุปว่า เมื่อมีการกระตุนด้วยสัญญาณประสาทความถี่สูงผ่านไซแนปส์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดรตร์ของไซแนปส์นั้นจะมีค่าเพิ่มขึ้นมาก และค่านั้นจะคงอยู่เป็นเวลานาน (หลายนาทีหรือหลายวัน หลังจากการกระตุนนั้น) โดยค่าความต่างศักย์ที่ปลายเดนไดรตร์ที่เขื่อมต่อไซแนปส์นั่นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การเพิ่มความแข็งแรงเชิงประสาทระยะยาว. และในทางกลับกัน เมื่อมีการกระตุนด้วยสัญญาณประสาทความถี่ต่ำผ่านไซแนปส์ ค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดรตร์ของไซแนปส์นั้นจะมีค่าลดลงมาก และค่านั้นก็คงอยู่เป็นเวลานาน โดยค่าความต่างศักย์ที่ปลายเดนไดรตร์ที่เขื่อมต่อไซแนปส์นั่นจะไม่ได้รับผลกระทบ. สิ่งนี้เรียกว่า การลดถอยความแข็งแรงเชิงประสาทระยะยาว. ประเด็นสำคัญ คือ (1) มีการเปลี่ยนแปลงความแข็งแรงของไซแนปส์ระยะยาวเกิดขึ้น (2) ผลการเปลี่ยนแปลงความแข็งแรงขึ้นกับความถี่ (3) ผลการเปลี่ยนแปลงเกิดขึ้นเฉพาะตัวของไซแนปส์.

กลไกเบื้องหลังนั้นอธิบายว่า ไซแนปส์ของแซฟเฟอร์คอลเลตเตอร์อล ทำงานผ่านสารสื่อประสาทกลูตาเมท และปลายประสาทของเซลล์ตัวรับที่ซีเอช1มีรีเซปเตอร์ที่ทำงานกับกลูตาเมทอยู่ 2 ชนิด คือ แอมປารีเซปเตอร์ (AMPA receptor) และ เօน เออมดีเอรีเซปเตอร์ (NMDA receptor). เมื่อปลายประสาทของเซลล์ที่ซีเอช3จะส่งสัญญาณกระตุนส่งผ่านไซแนปส์ มันจะปล่อยสารสื่อประสาทกลูตาเมทออกมา. เมื่อกลูตาเมทจับกับแอมປารีเซปเตอร์ แอมປารีเซปเตอร์จะเปิดออก และด้วยคุณสมบัติของแอมປารีเซปเตอร์ โซเดียมไอออนจะไหลเข้าสู่ปลายเดนไดรตร์ตัวรับที่ซีเอช1 แคลเซียมไอออนบางส่วนก็อาจไหลเข้าได้บ้างแต่ไม่มาก. แอมປารีเซปเตอร์ที่เปิดรับโซเดียมไอออนแล้วชักพักก็จะปิดลง. เมื่อกลูตาเมทจับกับเօนเออมดีเอรีเซปเตอร์จะเปิดออกเช่นกัน แต่เօนเออมดีเอรีเซปเตอร์จะมีแมกนีเซียมไอออนปิดช่องอยู่ ทำให้ไอออนยังไม่สามารถไหลผ่านได้. หลังจากโซเดียมไอออนผ่านแอมປารีเซปเตอร์เข้าสู่เซลล์ซีเอช1 สักพักปริมาณโซเดียมไอออนจะถูกสูบออกโดยกลไกของโซเดียม-โพแทสเซียมปั๊ม (sodium-potassium pump) ซึ่งเป็นเอนไซมน์ที่ทำงานรักษาระดับของเซลล์.

**การเพิ่มความแข็งแรงเชิงประสาทระยะยาว** หากการกระตุนมีความถี่สูงพอที่โซเดียมไอออนที่ไหลเข้ามาจะสะสมได้ (ความถี่สูงพอที่จะสะสมโซเดียมไอออน ที่เหลือจากการสูบออกของโซเดียม-โพแทสเซียมปั๊ม) ปริมาณโซเดียมไอออนที่เพิ่มขึ้นมาก จะทำให้เกิดไฟฟ้าสถิตย์ที่มากพอจะผลักแมgnีเซียมไอออนที่ปิดเอนเออมดีเอรีเซปเตอร์ออกໄไปได้. เมื่อแมgnีเซียมหลุดออกໄไป แคลเซียมไอออนก็สามารถผ่านเข้ามาทางเอนเออมดีเอรีเซปเตอร์ได้. แคลเซียมไอออนที่ไหลเข้าปริมาณมากจะช่วยเพิ่มค่าความต่างศักย์ที่ได้รับที่ปลายเดนไดรตร์ขึ้นไป. นอกจากนั้นแคลเซียมไอออนปริมาณมากจะจับกับโปรตีนคีเนส (Protein kinases) ซึ่งส่งผลให้เกิดการประกอบแอมປารีเซปเตอร์และติดแอมປารีเซปเตอร์เข้าไปทำงานที่ปลายเชื่อมประสาทเพิ่มขึ้น. ผลการเพิ่มแอมປารีเซปเตอร์จากการกระวนการนี้จะคงอยู่เพียงแค่เวลาสั้นๆไม่กี่ชั่วโมงเท่านั้น แต่หากมีแคลเซียมไอออนไหลเข้ามาในปริมาณมาก เป็นระยะเวลานานพอ (มีการกระตุนด้วยสัญญาณประสาทความถี่สูงเป็นระยะเวลานาน) จะส่งผลต่อเนื่องไปจนทำให้เกิดการเพิ่มปัจจัยการลอกรหัสดีเอ็นเอ (transcription factor) ซึ่งส่งผลต่อการแสดงออกของยีน (gene expression) และทำให้เกิดการสร้างโปรตีนที่ทำให้เกิดทั้งแอมປารีเซปเตอร์ใหม่เพิ่มขึ้น และ โกรทแฟคเตอร์ (growth factor) ที่จะไปทำให้เกิดการสร้างไซแนปส์ใหม่เพิ่มขึ้น ซึ่งผลจากการกระวนการนี้จะยานานและคงทนมาก.

**การลดถอยความแข็งแรงเชิงประสาทระยะยาว** สำหรับการกระตุนที่มีความถี่ต่ำ จะส่งผลให้ปริมาณของแคลเซียมไอออนอยู่ในระดับต่ำ. ปริมาณของแคลเซียมไอออนที่อยู่ในระดับต่ำจะไปกระตุนการทำงานของเอนไซม์ฟอสฟेटเตส (phosphatase) ซึ่งส่งผลไปลดจำนวนแอมປารีเซปเตอร์ที่ทำงานได้ลง.

การเพิ่มความแข็งแรงเชิงประสาทระยะยาว และ การลดถอยความแข็งแรงเชิงประสาทระยะยาว ต่างก็มีปัจจัยมาจากปริมาณแคลเซียมไอออนในปลายไซแนปส์ตัวรับ โดย ปริมาณแคลเซียมไอออนในระดับสูงจะทำให้เกิดการเพิ่มความแข็งแรงเชิงประสาทระยะยาว และ ปริมาณแคลเซียมไอออนในระดับต่ำจะทำให้เกิดการลดถอยความแข็งแรงเชิงประสาทระยะยาว. ระดับ

ที่เป็นจีดแบ่งระหว่างการเพิ่มและการลดถอยนี้ เนื่องจากจะเปลี่ยนแปลงตามสภาพของเซลล์ในลักษณะที่ช่วยรักษาสมดุลย์ (ทฤษฎีบีซีเอ็ม BCM theory[10]) ได้แก่ การเปลี่ยนแปลงในลักษณะการป้อนกลับเชิงลบ (negative feedback) เช่น เมื่อใช้แบบสัญญาณในการเพิ่มความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้จะสูงขึ้นเพื่อช่วยลดความเสี่ยงของการเพิ่มมากขึ้น และ เมื่อใช้แบบสัญญาณในการลดถอยความแข็งแรงเชิงประสาทระยะยาว ระดับขีดแบ่งนี้จะลดลงเพื่อช่วยลดความเสี่ยงของการลดถอยจนสูญเสียการเชื่อมต่อ.

จากสภาพพลาสติกของระบบประสาทและทฤษฎีบีซีเอ็ม เราอาจกล่าวได้ว่า การเรียนรู้ที่มีประสิทธิภาพคือ การเรียนรู้ต่อเนื่อง ลับกับการหยุดพักผ่อน เพื่อให้เกิดการกระตุ้นประสาทต่อเนื่องยาวนานเพียงพอ ที่จะทำให้เกิดการสร้างการเชื่อมต่อใหม่ และ หยุดพักเพื่อให้ระดับขีดแบ่งปรับตัวลงมา ทำให้การสร้างการเชื่อมต่อใหม่ทำได้ง่ายขึ้น เพราะ การเรียนรู้ต่อเนื่องเป็นเวลา กินไป จะเพิ่มระดับขีดแบ่งซึ่งจะทำให้การเข้าสู่ภาวะการเพิ่มความแข็งแรงเชิงประสาทระยะยาวทำได้ยากขึ้น.

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

Marcus Tullius Cicero

“ทำสิ่งโดยอย่ามากเกินไป ให้ความพอดีเป็นตัวชี้ทาง”

มาร์คัส ทูลลิอุส ซิเซอร์

## 5.5 แบบฝึกหัด

1. จงแสดงให้เห็นว่าอนุพันธ์ของสมการ 5.21 คือสมการ 5.22, 5.23, 5.27, และ 5.28. และจงแสดงให้เห็นว่าอนุพันธ์ของ  $h(a) = \frac{1}{1+\exp(-a)}$  คือ  $h'(a) = h(a) \cdot (1 - h(a))$ .

2. จงแสดงให้เห็นว่า  $\frac{\partial E_n}{\partial a_k} = y_k - t_k$  เมื่อพัฒนาระตุนของโครงข่ายชั้นเอาร์พุตและพัฒนาเป็นดังนี้

- (ก) สำหรับการหาค่าถดถอย เอาร์พุตเป็นพัฒนาเอกลักษณ์ (identity function) ซึ่งคือ  $y_k = a_k$  และพัฒนาค่าใช้จ่าย (cost function) คือ  $\text{Cost}_n = \frac{1}{2} \sum_k (y_k - t_{kn})^2$ .
- (ข) สำหรับการจำแนกประเภทแบบสองกลุ่ม เอาร์พุตเป็นพัฒนาซิกโนแอด (sigmoid function) ซึ่งคือ  $y_1 = \frac{1}{1+\exp(-a_1)}$  และพัฒนาค่าใช้จ่ายคือ  $\text{Cost}_n = -t_n \log(y_1) - (1-t_n) \log(1-y_1)$ .
- (ค) สำหรับการจำแนกประเภทแบบหลายกลุ่ม เอาร์พุตเป็นพัฒนาซอฟต์แมกซ์ (softmax function) ซึ่งคือ  $y_k = \frac{\exp(a_k)}{\sum_{q=1}^K \exp(a_q)}$  และพัฒนาค่าใช้จ่ายคือ  $\text{Cost}_n = -\sum_k t_{kn} \log(y_k)$ .

3. จงแสดงว่าสมการ 5.39 ได้มาจากสมการ 5.38.

4. โครงข่ายประสาทเทียมก็สามารถทำเรกุลารีเซชันได้เช่นเดียวกับโมเดลเชิงเส้น (หัวข้อ 4.2). จากพัฒนาเป้าหมายสำหรับการหาค่าถดถอย (สมการ 5.19), เทอมสำหรับเรกุลารีเซชันสามารถเพิ่มเข้าไปได้

เป็น

$$\mathbf{Cost}_n = E_n + \frac{\lambda_1}{2} \sum_{i \neq 0} (w_{ji}^{(1)})^2 + \frac{\lambda_2}{2} \sum_{j \neq 0} (w_{kj}^{(2)})^2 \quad (5.40)$$

สังเกตว่าเรกูลาริเซชั่นเพอมไม่รวมค่าไบอัส ( $w_{j0}^{(1)}, w_{k0}^{(2)}$ ). จะแสดงให้เห็นว่า

$$\frac{\partial \mathbf{Cost}_n}{\partial w_{ji}^{(1)}} = \frac{\partial E_n}{\partial w_{ji}^{(1)}} + \lambda_1 w_{ji}^{(1)} \quad (5.41)$$

$$\frac{\partial \mathbf{Cost}_n}{\partial w_{kj}^{(2)}} = \frac{\partial E_n}{\partial w_{kj}^{(2)}} + \lambda_2 w_{kj}^{(2)}. \quad (5.42)$$

5. จากสมการ 5.3 จงอภิปรายความเหมือนและความต่าง ในเชิงความสามารถในการประมาณความสัมพันธ์ข้อมูล ระหว่างโมเดลที่มีเบซิสฟังชันที่เป็นฟังชันเชิงเส้นกับอินพุต และโมเดลที่มีเบซิสฟังชันเป็นอินพุตตรงๆ (นั่นคือ  $\phi_j(\mathbf{x}) = x_j$ ) พิรุณยกตัวอย่างให้เห็นภาพ และจงอภิปรายถึงเหตุผลของการเลือกเบซิสฟังชันที่ต้องไม่เป็นฟังชันเชิงเส้นกับอินพุต ดังกล่าวในตอนต้นของหัวข้อ 5.2

6. จงหาข้อมูลและเปรียบเทียบโครงข่ายประสาทเทียมกับวิธีชั้บเพอร์เวคเตอร์แมชชีน(support vector machine)[20] และวิธีป่าสุ่ม(random forest)[14] ในเชิงฟังชันจุดประสงค์ และการคำนวณเพื่อทำนายกลุ่ม และกระบวนการในการคำนวณค่าน้ำหนัก พิรุณอภิปรายข้อดีข้อเสียของทั้งสามวิธี

7. จงหาข้อมูลการพัฒนาและผู้เชี่ยวชาญที่มีส่วนสำคัญในการพัฒนาของโครงข่ายประสาทเทียม วิธีชับเพอร์เวคเตอร์แมชชีน และวิธีป่าสุ่ม. และจงอภิปรายสิ่งที่ได้เรียนรู้จากการศึกษาประวัติการพัฒนาของวิธีการต่างๆ รวมถึงวิธีการทำงานของผู้เชี่ยวชาญที่มีส่วนสำคัญในการพัฒนาของวิธีการเหล่านี้

8. จงหาข้อมูลของวิธีการเรียนรู้ของเครื่องแบบต่างๆ เช่น โครงข่ายแบบแผนที่จัดโครงสร้างตัวเอง (self-organizing map)[45], วิธีการกรองเชิงความร่วมมือ (Collaborative Filtering)[24], วิธีเคเมินส์ (K-Means)[9], วิธีการประมาณความหนาแน่นเชิงแก่น (Kernel Density Estimation)[9], วิธีวิเคราะห์ส่วนประกอบหลัก (Principle Component Analysis)[9], และ ชาช่าอัลกอริทึม (SARSA algorithm)[74] และจงอภิปรายภาระกิจของวิธีต่างๆเหล่านี้ จากมุมมองของศาสตร์การหารากค่าดีที่สุด (บทที่ 2) พิรุณระบุฟังชันจุดประสงค์ และตัวแปรตัดสินใจที่วิธีเหล่านี้กำหนดด้วย



## บทที่ 6

### การประยุกต์ใช้โครงข่ายประสาทเทียม

“In theory there is no difference between theory and practice.

In practice there is.”

—Yogi Berra

“ในทางทฤษฎีแล้วไม่มีความแตกต่างระหว่างทฤษฎีกับการปฏิบัติ  
แต่ในทางปฏิบัติแล้วต่าง”

—โยギ เบอร์ร่า

บท 5 อภิปรายการทำงานของโครงข่ายประสาทเทียม ว่าสามารถทำงานได้อย่างไรในมุมมองเชิงทฤษฎี.  
อย่างไรก็ตาม การนำโครงข่ายประสาทเทียมไปใช้งานในทางปฏิบัติ ยังมีรายละเอียดอีกมาก ที่มีผลให้  
โครงข่ายประสาทเทียมสามารถทำงานได้อย่างมีประสิทธิภาพเพิ่มขึ้น และบ่อยครั้งที่สำหรับการเรียนรู้ของ  
เครื่องแล้วไม่ใช่เรื่องแปลกเลย ที่จะพบกรณีที่ความแตกต่าง ระหว่างความสามารถในการทำงานได้อย่างมี  
ประสิทธิภาพกับไม่มีประสิทธิภาพ เทียบเท่าได้กับการทำงานได้หรือไม่ได้เลยทีเดียว.

ปัจจัยสำคัญเรื่องแรก คือการกำหนดค่าเริ่มต้นของค่าน้ำหนัก (weight initialization). สังเกตว่า  
โครงข่ายประสาทเทียมมีโครงสร้างที่มีความสมมาตรและมีรูปแบบซ้ำซ้อนกัน很多อยู่ (แต่ละหน่วยประสาท  
เทียมมีสูตรการคำนวนเหมือนกัน) ดังนั้นถ้าหากกำหนดค่าเริ่มต้นของค่าน้ำหนัก เป็น 0 ทั้งหมด ค่าน้ำหนัก  
ต่างๆแม้จะเป็นค่าของหน่วยประสาทคนละหน่วย แต่ก็จะถูกปรับปรุงไปเป็นค่าใหม่อนๆกัน. สุดท้ายแล้ว  
ไม่ว่าจะฝึกอย่างไร หรือไม่เคลมมีความลึกเท่าไร หรือมีจำนวนหน่วยซ่อนมากเท่าไร ผลที่ได้ก็คือโครงข่าย  
ประสาทเทียมที่ค่าน้ำหนักเหมือนๆกันหลายๆค่า ซึ่งผลก็คือโครงข่ายประสาทเทียมจะไม่สามารถทำงานได้  
เต็มศักยภาพของมัน. กล่าวว่าอย่างๆ ในทางปฏิบัติแล้วการกำหนดค่าน้ำหนักเริ่มต้นเป็นคุณย์ทุกค่า จะทำให้  
การฝึกโครงข่ายประสาทเทียมล้มเหลว. วิธีการกำหนดค่าเริ่มต้นของค่าน้ำหนักที่ง่ายที่สุด และสามารถ  
ทำงานได้อย่างมีประสิทธิภาพพอสมควร คือการสุ่มค่าเริ่มต้น ดังที่เราจะได้เห็นในตัวอย่าง (หัวข้อ 6.1). ผู้  
อ่านที่สนใจศาสตร์และศิลป์ของการกำหนดค่าน้ำหนักเริ่มต้น สามารถศึกษาวิธีเหงียน-วิดโกร์ (Nguyen-  
Widrow Weight Initialization) ซึ่งเป็นวิธีการกำหนดค่าเริ่มต้นของค่าน้ำหนักที่มีประสิทธิภาพมากขึ้นได้  
จากบทความของเหงียนและวิดโกร์[59].

นอกจากนั้นยังอีกประเด็นอื่นๆอีก ที่มีผลต่อประสิทธิภาพของการใช้งานโครงข่ายประสาทเทียม เช่น การทำการประมวลผลก่อนและหลัง (pre- and post-processing). หัวข้อ 6 กล่าวถึงการทำ normalization ให้กับข้อมูลที่มีค่าอยู่ในช่วงที่ต่างกัน ไม่สามารถใช้ในโครงข่ายประสาทเทียมได้ ดังนั้น จึงเป็นการทำการทำ normalization ให้กับข้อมูลที่มีค่าอยู่ในช่วงที่ต่างกัน เช่น 0.001 ถึง 0.002 ในขณะที่มีค่าอยู่ในช่วง 2000 ถึง 4000 เมื่อนำค่าของอินพุตเข้าไปฝึกโครงข่ายประสาทเทียมโดยตรง การฝึกอาจต้องใช้เวลานานมากที่น้ำหนักจะสามารถปรับตัวเองมา เพื่อรักษาสมดุลของสเกลที่ต่างกันมากๆขนาดนี้ของอินพุตได้ ไม่อย่างนั้น อินพุตมิตรที่สองที่มีขนาดใหญ่กว่ามากอาจจะมีอิทธิพลกับเอาร์พุตมากจนอินพุตมิตรที่หนึ่งไม่มีความหมาย และส่งผลให้โมเดลไม่สามารถทำงานได้อย่างมีประสิทธิภาพ ( เพราะเมื่อสร้างโมเดลจากอินพุตมิตรที่สองเท่านั้น )

การทำ normalization สามารถทำได้หลายวิธี วิธีที่นิยมแบบที่หนึ่ง คือการปรับค่าให้ค่าอยู่ในช่วงที่กำหนด เช่น  $[-1, 1]$  หรือ  $[0, 1]$ . หากช่วงที่ต้องการคือ  $[y_{\min}, y_{\max}]$  ค่าที่ทำการ normalization คือ  $y$  สามารถคำนวณจาก

$$y = (y_{\max} - y_{\min}) \cdot \frac{x - x_{\min}}{x_{\max} - x_{\min}} + y_{\min}$$

เมื่อ  $x$  คือค่าเดิม และ  $x_{\min}$  กับ  $x_{\max}$  คือค่าน้อยที่สุดกับค่ามากที่สุดของ  $x$ . ตัวอย่างโค้ดสำหรับการทำ normalization แบบนี้แสดงในรายการ 7.12.

วิธีการทำ normalization ที่นิยมแบบที่สอง คือการปรับค่าเพื่อให้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานเป็น 0 กับ 1 ตามลำดับ. ดังนั้น ค่าที่ทำการ normalization คือ  $y$  สามารถคำนวณได้จาก

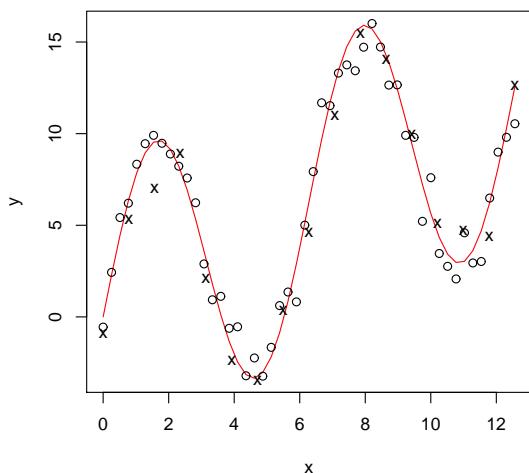
$$y = \frac{x - \bar{x}}{\sigma_x}$$

เมื่อ  $\bar{x}$  และ  $\sigma_x$  คือค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของ  $x$  (ค่าเดิมก่อนทำการ normalization). ตัวอย่างโค้ดสำหรับการทำ normalization เพื่อปรับค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานแสดงในรายการ 6.3.

หมายเหตุ การทำการ normalization ทำเพื่อปรับสเกลของมิติต่างๆให้สมดุลกัน ดังนั้นการทำ normalization จะทำแต่ละมิติ เช่น หากมีสองมิติ  $\mathbf{x} = [x_1, x_2]^T$  และหากต้องการทำ normalization ให้อยู่ในช่วง  $[0, 1]$  ทั้งคู่ ก็ทำได้โดยแยกทำการ normalization สำหรับแต่ละมิติซึ่งจะได้

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} (x_1 - x_{1\min}) / (x_{1\max} - x_{1\min}) \\ (x_2 - x_{2\min}) / (x_{2\max} - x_{2\min}) \end{bmatrix}$$

เมื่อ  $y_1$  และ  $y_2$  คือค่าที่ทำการ normalization และ  $x_{1\min}$  และ  $x_{1\max}$  คือค่าน้อยที่สุดกับค่ามากที่สุดของมิติที่หนึ่ง ( $x_1$ ) และ  $x_{2\min}$  กับ  $x_{2\max}$  คือค่าน้อยที่สุดกับค่ามากที่สุดของมิติที่สอง ( $x_2$ ).



รูปที่ 6.1: แสดงข้อมูลตัวอย่าง จุดข้อมูลสำหรับการฝึก (วงกลม) และ การทดสอบ (กากระบท); เส้นทึบแสดงฟังชันที่ใช้สร้างข้อมูลแต่ไม่มีสัญญาณรบกวน.

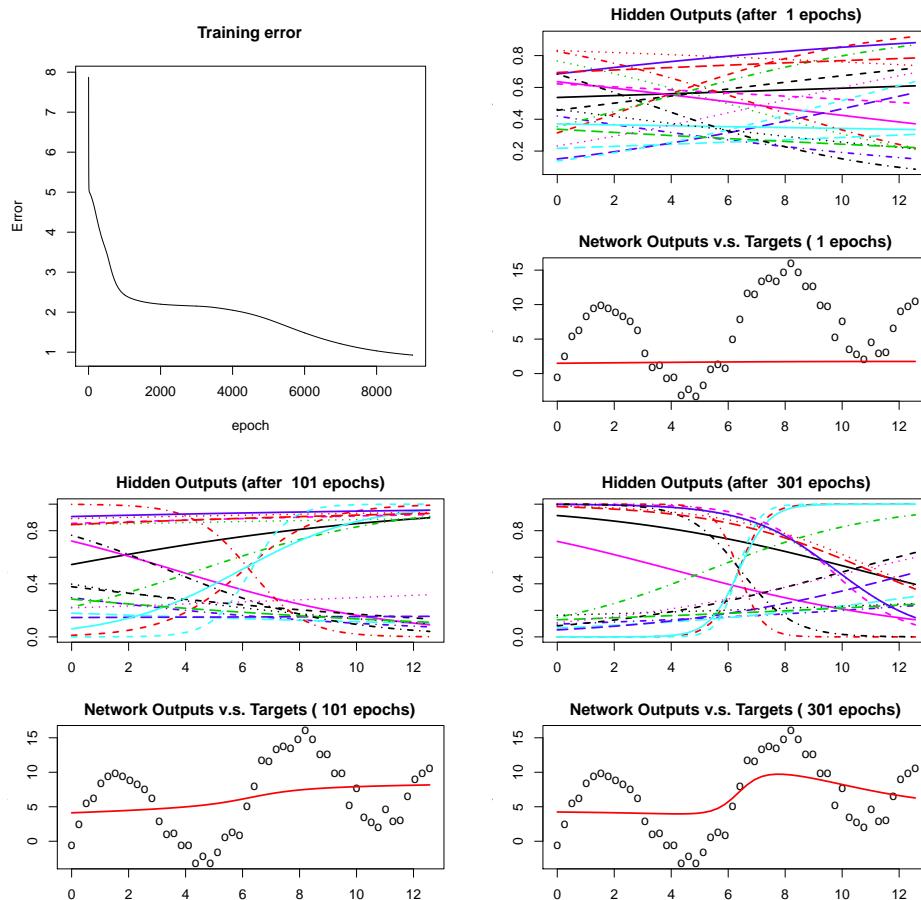
## 6.1 ตัวอย่างง่ายๆ

ตัวอย่างง่ายๆของการหาค่าคาดถอยของข้อมูลที่ทั้งอินพุตและเอาต์พุตมีขนาดหนึ่งมิติ. ข้อมูลดูนี้สร้างจากความสัมพันธ์  $y = x + 8 \cdot \sin(x) + \epsilon$  โดย  $\epsilon$  คือค่าสัญญาณรบกวน ที่สุ่มมาจากการกระจายแบบปกติ  $\epsilon \sim \mathcal{N}(0, 1)$ . ตัวอย่างนี้แสดงการสร้างข้อมูลขึ้นมา 50 จุดข้อมูล (หรือเรียกว่ากอ yogurt ระหว่างเปลี่ยน) สำหรับฝึกโครงข่าย และใช้ 17 จุดข้อมูลสำหรับทดสอบ (ประมาณ 33%) ดังแสดงใน รูป 6.1.

ตัวอย่างนี้เลือกใช้โครงข่ายประสาทเทียมสองชั้น โดยเลือกใช้ 20 หน่วยชั้อน และใช้ฟังชันกระตุ้นของเอาต์พุตเป็นฟังชันเอกลักษณ์ที่เหมาะสมสำหรับการหาค่าคาดถอย. การทำนอร์มอลайเซชันก็เลือกใช้วิธีที่ปรับค่าอินพุต  $x$  เพื่อให้ค่าเฉลี่ยและเบี่ยงเบนมาตรฐานเป็น 0 และ 1 ตามลำดับ. สำหรับการฝึกโครงข่าย ตัวอย่างนี้กำหนดค่าเริ่มต้นน้ำหนักแบบสุ่มค่า โดยค่าสุ่มจากการแจกแจงเอกรูป (uniform distribution) จากช่วงค่า  $[-0.5, 0.5]$ . ตัวอย่างนี้ใช้วิธีลงเกรเดียนต์ในการฝึก และเลือกใช้ค่าอัตราการเรียนเป็น 0.01 และ 0.0003 สำหรับน้ำหนักชุดชั่วонและน้ำหนักชุดเอาต์พุตตามลำดับ โดยทำการฝึกโครงข่ายทั้งหมด 9001 รอบ.

รูป 6.2 แสดงค่าผิดพลาด (Root Mean Squared Error, RMSE) ที่รอบฝึก (epoch) ต่อๆ (ภาพบนซ้าย). ค่าผิดพลาดที่แสดงนี้เป็นค่าผิดพลาดของข้อมูลชุดฝึกหัด. ด้วยคุณสมบัติของวิธีลงเกรเดียนต์ ถ้าเราใช้ค่าอัตราการเรียนเล็กพอกว่าวิธีลงเกรเดียนต์รับประกันว่า ค่าผิดพลาดนี้จะลดลงในแต่ละรอบ (หรืออย่างน้อยก็จะไม่เพิ่มขึ้น). หลังจากการฝึกผ่านไป 1 รอบฝึก (ภาพบนขวา) สังเกตว่าค่าเอาต์พุตของหน่วยชั่วонต่างๆยังมีลักษณะสุ่มอยู่มาก. ค่าเอาต์พุตของหน่วยชั่วอนจะขึ้นกับค่าอินพุต  $x$  และ ค่าน้ำหนักชั้นที่หนึ่ง  $\mathbf{w}^{(1)}$  เท่านั้น. หลังจาก 1 รอบฝึก ค่าเอาต์พุตของโครงข่ายยังมีค่าคงที่ (ใกล้ๆ 0) ตลอดช่วงค่าอินพุต  $x$  (ภาพขวาที่ 2 จากบน) เพราะเรากำหนดค่าเริ่มต้นของน้ำหนักให้มีขนาดน้อยๆ. ค่าน้ำหนักชั้นที่

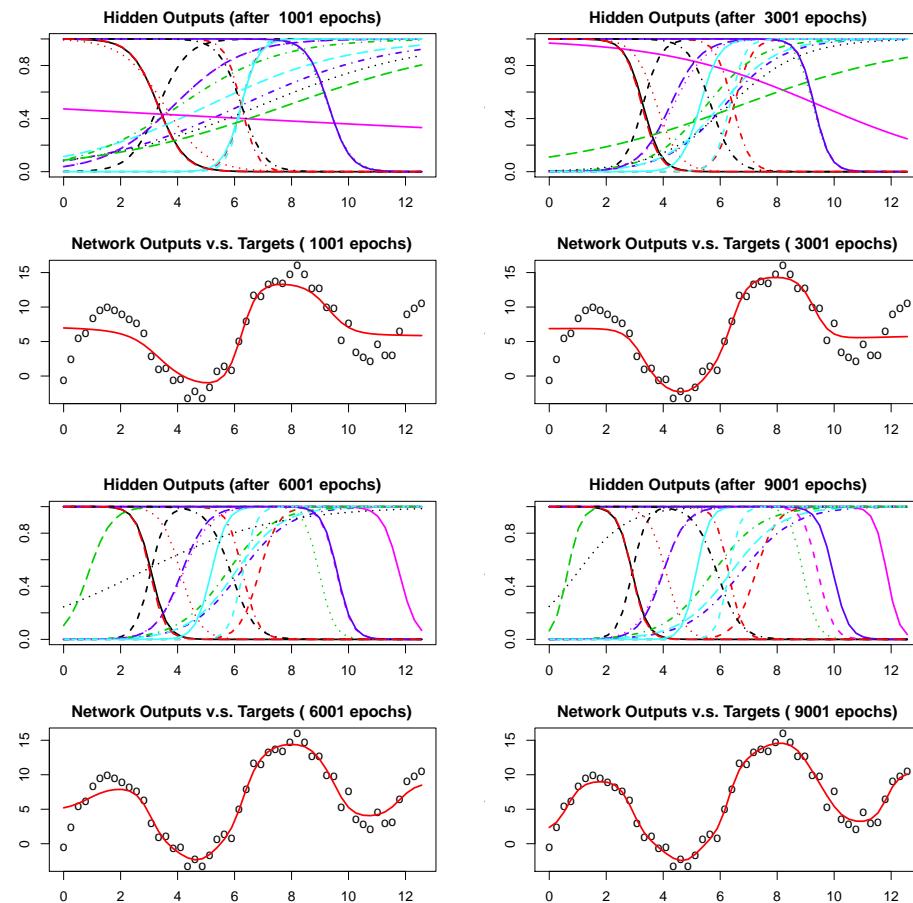
รูปที่ 6.2: การฝึกโครงข่าย: ภาพย่อของชั้นปิดพลาดที่รอบฝึกต่างๆ ภาพย่อที่เหลือแสดงเอาต์พุตของหน่วยซ่อนต่างๆ (ชื่อภาพย่อ Hidden Outputs) และเอาต์พุตของโครงข่ายเปรียบกับจุดข้อมูลฝึก (ชื่อภาพย่อ Network Outputs โดยเส้นทึบแสดงเอาต์พุตของโครงข่าย และวงกลมแทนจุดข้อมูลที่ใช้ฝึก) หลังการฝึกไป 1, 101, และ 301 รอบ ตามระบบในแต่ละภาพย่อ



สองบอกส่วนผสมของเอาต์พุตของหน่วยซ่อนต่างๆ เพื่อผสมออกมาเป็นเอาต์พุตของโครงข่าย. หลังจาก 1 รอบฝึก เอาต์พุตต่างๆของหน่วยซ่อน มีค่าต่ำต่ำตลอดช่วงอนพุต (มีค่าอยู่ประมาณ 0.1 ถึง 0.9) แล้ว ค่าของน้ำหนักชั้นที่สองก็ต่ำ (เพราะเริ่มต้นด้วยค่าสุ่มจากช่วงค่าน้อยๆ และเพิ่งฝึกไปเพียงแค่ 1 รอบ ด้วยค่าอัตราการเรียนที่น้อย) ตั้งนั้นค่าเอาต์พุตของโครงข่ายจะเป็นอย่างที่เห็น ซึ่งไม่ได้ใกล้เคียงกับจุดข้อมูลฝึกที่เป็นเป้าหมายเลย.

หลังทำการฝึกไป 101 รอบ เอาต์พุตของชั้นย่อยกระจายตัวแตกต่างกันมากขึ้น และเอาต์พุตของโครงข่ายก็เริ่มปรับตัวให้พอดังเกตุเห็นความโถงได้บ้าง. หลังทำการฝึกไป 301, 1001, 3001 รอบ เอาต์พุตของชั้นย่อยกระจายตัวแตกต่างกันค่อนข้างชัดเจน และเอาต์พุตของโครงข่ายก็เห็นเป็นรูปร่างตามจุดข้อมูลฝึกชัดเจนขึ้น. แต่สังเกตุว่าที่ 3001 รอบฝึก บริเวณปลายทางของเอาต์พุตโครงข่ายยังไม่ได้ปรับตัวโถงอีก รับกับจุดข้อมูลฝึก (ประมาณบริเวณค่าแกนนอน  $x > 11$ ) และเมื่อสังเกตุเอาต์พุตของชั้นซ่อนจะเห็นว่า ยังไม่มีเอาต์พุตของชั้นย่อยหน่วยใดที่มีช่วงโถงอยู่บริเวณนั้น. เปรียบเทียบเอาต์พุตหลังการฝึก 3001 รอบ กับเอาต์พุตหลังการฝึก 6001 รอบ จะเห็นว่า เอาต์พุตของโครงข่ายหลังการฝึก 6001 รอบจะมีการโถงอีก รับกับข้อมูลฝึกในช่วงปลายทาง และความยืดหยุ่นในการปรับตัวโถงอนนั้น ก็มาจากเอาต์พุตของชั้นซ่อน

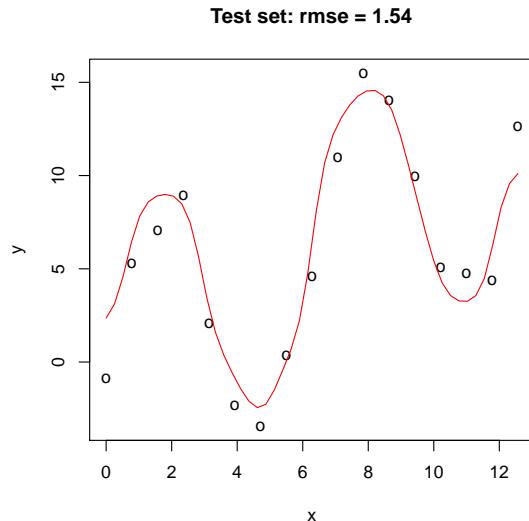
รูปที่ 6.3: การฝึกโครงข่าย (ต่อ): ภาพย่อแสดงเอาต์พุตของหน่วยซ่อนต่างๆ (ชื่อภาพย่อ Hidden Outputs) และภาพย่อแสดงเอาต์พุตของโครงข่ายเบรีบเทียบกับจุดข้อมูลฝึก (ชื่อภาพย่อ Network Outputs โดย สันทิบแสดงเอาต์พุตของโครงข่าย และวงกลมแทนจุดข้อมูลที่ใช้ฝึก) หลังการฝึกไป 1001, 3001, 6001 และ 9001 ตามระบุในแต่ละภาพย่อ



หน่วยหนึ่งที่ปรับตัวมาจากการเปลี่ยนพลวัตรขัยบماอยู่ช่วงตั้งก่อตัว (เอาต์พุตของชั้นย่อหน่วยหนึ่ง ที่แสดงด้วยสันทิบสีบานเย็น—เอาต์พุตของหน่วยซ่อนสืบเด่นขาวสุดในภาพ—ได้ขัยบช่วงที่เปลี่ยนค่าแนวตั้งจาก 1 มาเป็น 0 มาอยู่ที่บริเวณค่าแกนนอน  $x \approx 11$ ).

หลังจากฝึกไป 9001 รอบ ค่าน้ำหนักของโครงข่ายก็ถูกเข้า และเอาต์พุตของโครงข่ายก็ปรับตัวรับกับค่าของจุดข้อมูลฝึกได้ดี (ภาพขาวล่างสุด รูป 6.3) รูป 6.4 แสดงผลการทดสอบของโครงข่ายที่ฝึกแล้วนี้ กับข้อมูลชุดทดสอบ.

สังเกตุระหว่างการฝึก (รูป 6.2 และ 6.3 หลังฝึกไป 101, 301, 1001, 3001, และ 6001) เอาต์พุตของหน่วยซ่อนจะค่อยๆปรับตัว เพื่อทำให้เอาต์พุตของโครงข่ายใกล้เคียงกับข้อมูลทดสอบมากขึ้น. นอกจากนี้ รูป 6.2 และ 6.3 ยังเสริมมุ่งที่มองว่า โครงข่ายประสาทเทียมสามารถมีความแม่นยำและรวดเร็วได้. โดยบางภาพ เช่น ภาพของเอาต์พุตของหน่วยซ่อนและโครงข่ายหลัง 6001 รอบ เราสามารถเห็นความสัมพันธ์ระหว่างเอาต์พุตของหน่วยซ่อนและเอาต์พุตของโครงข่าย โดยเฉพาะที่อินพุตมีค่ามากกว่า 11. หรือ ความสัมพันธ์ระหว่างเอาต์พุตของหน่วยซ่อนและของโครงข่ายหลัง 9001 รอบฝึก โดยเฉพาะที่อินพุตมีค่าน้อยกว่า 2.



รูปที่ 6.4: ผลการหาค่าถดถอยด้วยโครงข่ายประสาทเทียม. กราฟเส้นทึบแสดงผลทำนายของโครงข่าย และวงกลมแสดงจุดข้อมูลทดสอบ. ระยะห่างในแนวตั้วระหว่างวงกลมและเส้นทึบ คือค่าความผิดพลาด ซึ่งจากตัวอย่างนี้ได้ค่าความผิดพลาด RMSE เป็น 1.54

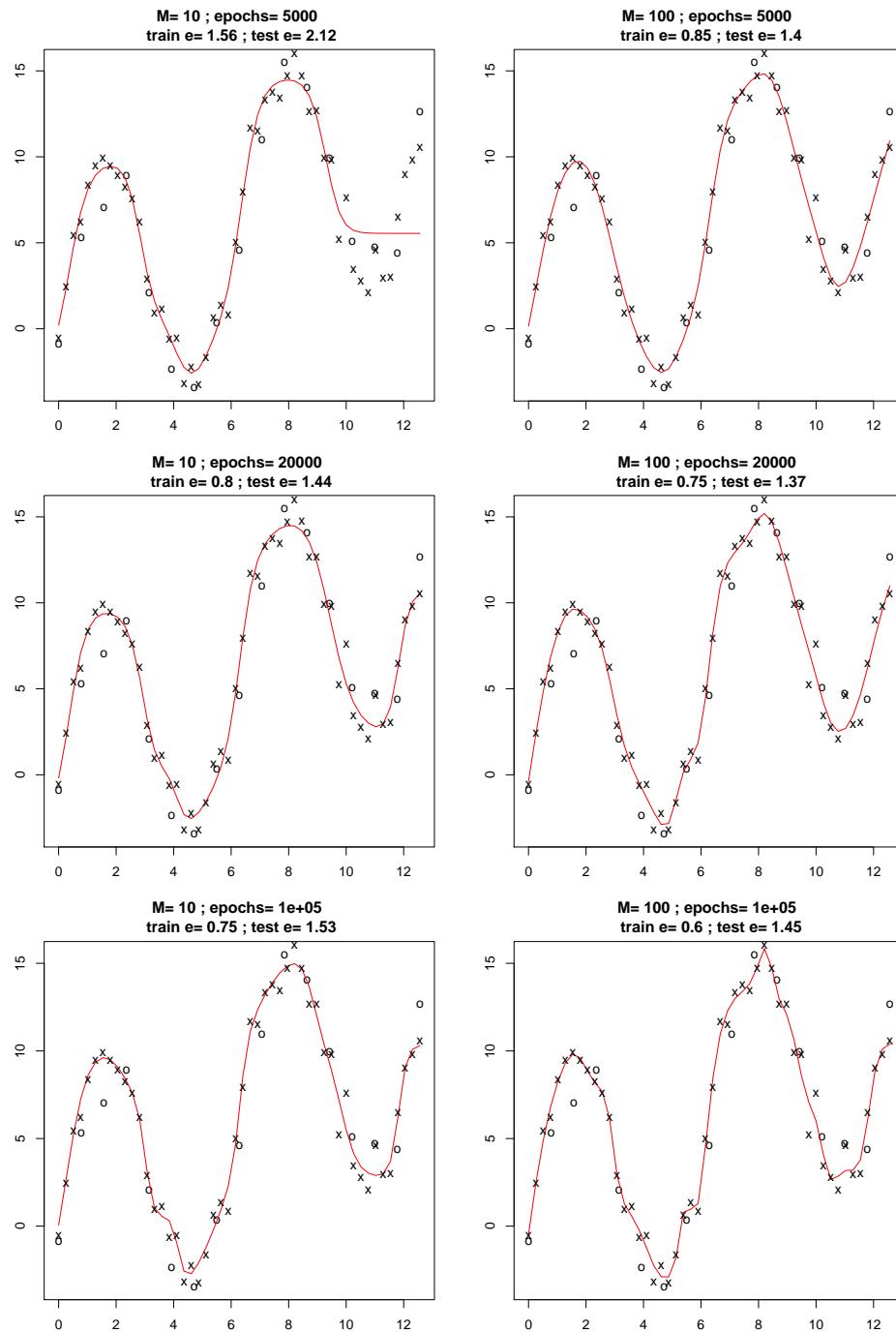
## การหยุดก่อนกำหนด

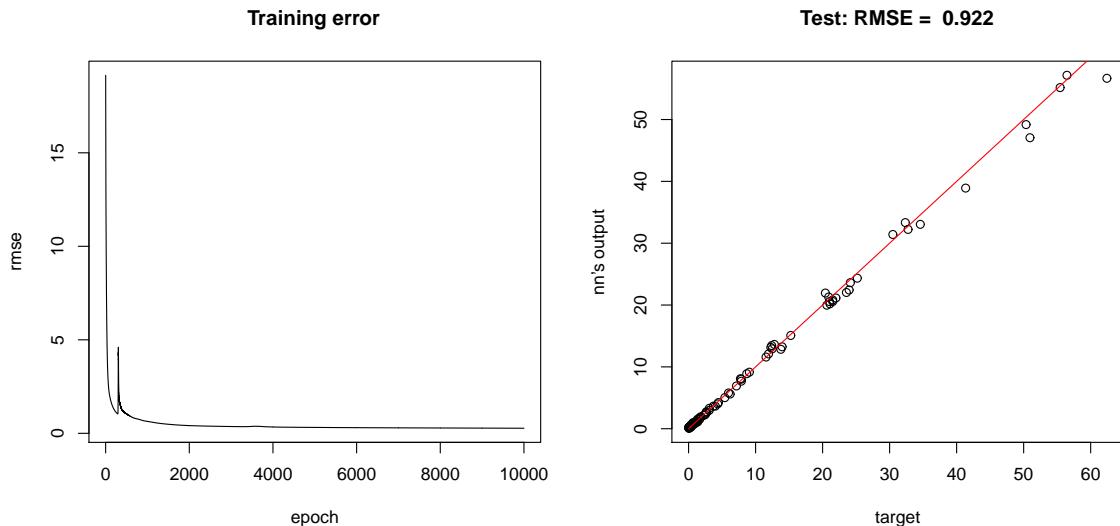
เช่นเดียวกับโมเดลทำนายอื่นๆ การใช้งานโครงข่ายประสาทเทียมก็ต้องคำนึงถึง เรื่องโอเวอร์ฟิตติ้ง (overfitting) หรือคุณสมบัติความทั่วไป (generalization) ด้วย. การเลือกใช้โมเดลที่มีความยืดหยุ่นสูงๆ เช่นโครงข่ายประสาทเทียมสองชั้นที่มีจำนวนหน่วยซ่อนมากๆ ก็อาจจะเสี่ยงที่จะสูญเสียคุณสมบัติความทั่วไปได้. นั่นคือ แทนที่จะได้โมเดลที่สามารถประมาณระบบที่สนใจ หรือทำนายปริมาณที่สนใจได้ดี แต่ผลอาจจะเป็นการที่ได้โมเดลที่ปรับตัวไปเข้ากับสัญญาณรบกวนของข้อมูลชุดฝึกหัดแทน. ผู้อ่านสามารถทบทวนเรื่องโอเวอร์ฟิตติ้งหรือการเสียคุณสมบัติความทั่วไปได้จากหัวข้อ 3.1 และ 3.2.

รูป 6.5 แสดงผลจากการใช้โครงข่ายขนาดหน่วยซ่อน 10 กับ 100 และฝึก 5,000, 20,000, และ 100,000 รอบ. สังเกตว่า สำหรับโครงข่ายขนาดหน่วยซ่อนเป็นสิบ ( $M = 10$ ) การฝึกสองหมื่นรอบทำให้ได้คุณภาพของโมเดลดีขึ้น (ค่าผิดพลาดของชุดทดสอบลดลงจาก 2.12 ที่ห้าพันรอบฝึก เหลือเพียง 1.44 ที่สองหมื่นรอบฝึก). แต่การฝึกต่อไปถึงหนึ่งแสนรอบฝึก แม้จะทำให้ค่าผิดพลาดของชุดฝึกต่ำลง (จาก 0.8 ที่สองหมื่นรอบฝึก เป็น 0.75 ที่หนึ่งแสนรอบฝึก) แต่ค่าผิดพลาดของชุดทดสอบกลับเพิ่มขึ้น (จาก 1.44 ที่สองหมื่นรอบฝึก เป็น 1.53 ที่หนึ่งแสนรอบฝึก). เช่นเดียวกัน โครงข่ายขนาดหน่วยซ่อนหนึ่งร้อย ( $M = 100$ ) ที่หนึ่งแสนรอบฝึก ที่ผลลัพธ์แสดงการที่โมเดลพยายามปรับตัวไปประมาณสัญญาณรบกวนของชุดฝึกอย่างเห็นได้ชัด. สิ่งที่สำคัญ คือ ค่าผิดพลาดของชุดฝึกที่ลดลง ในขณะที่ค่าผิดพลาดของชุดทดสอบเพิ่มขึ้น. นี่คือการที่โมเดลปรับตัวไปประมาณสัญญาณรบกวนที่เห็นในชุดฝึก แทนที่จะประมาณระบบที่สนใจ. พุดอีกอย่างคือ โมเดลเสียคุณสมบัติความทั่วไป หรือเกิดโอเวอร์ฟิตติ้งขึ้นนั่นเอง.

เรื่องนี้ นอกจากทำให้ได้โมเดลทำนายที่คุณภาพต่ำแล้ว ยังทำให้เสียเวลาในการฝึกโครงข่ายเพิ่มขึ้น

รูปที่ 6.5: ผลจากโครงข่ายขนาดหน่วยซ่อนสิบหน่วย ( $M = 10$ ) กับหนึ่งร้อยหน่วย ( $M = 100$ ) ที่การฝึกผ่าน 5000, 20000, และ 100000 รอบ โดย แกนนอนแสดงค่าอินพุต แกนตั้งแสดงค่าเอาต์พุต เส้นทึบสีแดงแสดงแสดงเอาต์พุตของโครงข่าย การบทແທນจุดข้อมูลฝึก และวงกลมแทนจุดข้อมูลทดสอบ





รูปที่ 6.6: การฝึกและทดสอบการหาค่าถดถอยของข้อมูลชุดเรือยอชท์ (yacht dataset) ด้วยโครงข่ายประสาทเทียม.

ด้วย. วิธีหนึ่งที่นิยมใช้กับการฝึกโครงข่ายประสาทเทียมเพื่อลดปัญหานี้คือการทำหยุดก่อนกำหนด (Early Stopping).

การทำหยุดก่อนกำหนดนั้นตรงไปตรงมาตามธรรมชาติของคุณสมบัติความทั่วไป. สัญญาณปังซึ่ว่าโมเดลเริ่มเกิดโอเวอร์ฟิตติ้งคือการที่โมเดลทำนายค่าข้อมูลชุดฝึกได้ดีขึ้น แต่ทำนายข้อมูลอื่น (จากระบบเดียวกัน)ได้แย่ลง. ดังนั้น นอกจากชุดทดสอบ (ที่แยกเก็บไว้ทดสอบตอนสุดท้ายแล้ว) วิธีหยุดก่อนกำหนดจะแบ่งข้อมูลออกอีกส่วนเป็น ชุดвалиเดชัน (Validation Set) ซึ่งเป็นเสมือนชุดข้อมูลทดสอบระหว่างการทำโมเดล. และระหว่างการฝึกโครงข่าย ก็ค่อยตรวจสอบค่าผิดพลาดของการทำนายกับชุดข้อมูลвалиเดชันนี้ ถ้าค่าผิดพลาดของชุดвалиเดชันมีค่าเพิ่มขึ้น ก็เป็นสัญญาณบอกว่าอาจเกิดโอเวอร์ฟิตติ้งขึ้น. สัญญาณนี้สามารถใช้เป็นเครื่องมือช่วยในการตัดสินใจที่จะหยุดการฝึกได้.

## 6.2 ตัวอย่างการหาค่าถดถอย

หัวข้อนี้แสดงตัวอย่างการประมาณค่าแรงต้านที่เหลือค้าง (residuary resistance) ของเรือยอชต์ขณะแล่นอยู่ ซึ่งก็เป็นปัญหาการหาค่าถดถอย. ข้อมูลชุดนี้ได้มาจากการฐานข้อมูลของคลังข้อมูลยูซีไอ[6] ที่ <http://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics>.

ข้อมูลชุดนี้มี 308 ระเบียน นั่นคือ มี 308 จุดข้อมูล และแต่ละจุดข้อมูลจะมี 7 เอกข้อมูล (7 มิติ). เอกข้อมูลที่ 1-6 ซึ่งได้แก่ ตำแหน่งตามแนวยาวเรือของศูนย์กลางการลอยตัว (longitudinal position of the center of buoyancy เป็นลักษณะที่ช่วยอธิบายการกระจายน้ำหนักของเรือตามแนวยาวว่าอยู่หน้าลำ กลางลำ หรือท้ายลำอย่างไร), ค่าสัมประสิทธิ์ปริซึม (prismatic coefficient เป็นลักษณะที่ช่วยอธิบายรูปทรงของห้องเรือ โดยวัดจากอัตราส่วนของปริมาตรที่อยู่ใต้น้ำของห้องเรือ เปรียบเทียบกับปริมาตรของรูปทรงปริซึมที่มีความยาวเท่ากัน และพื้นที่หน้าตัดเท่ากับ พื้นที่หน้าตัดที่กว้างที่สุดของห้องเรือ), อัตราส่วน

ความยาวเรือกับการกระจัด (length-displacement ratio เป็นลักษณะที่ช่วยบ่งชี้ถึงความหนักของเรือ เทียบกับความยาว เรือที่หนักจะมีค่านี้มาก เรือที่เบาจะมีค่านี้น้อย), อัตราส่วนความกว้างเรือกับระดับjm้น้ำ (beam-draught ratio เป็นลักษณะทรงท้องเรือที่วัดจาก อัตราส่วนความกว้างเรือกับความกว้างของส่วนที่กว้างที่สุดของเรือในแนวระดับน้ำ), อัตราส่วนความยาวกับความกว้างเรือ (length-beam ratio), และตัวเลขฟรูดู (Froude number เป็นค่าที่บ่งบอกความต้านทานของการทิ่มตัวเคลื่อนที่ในน้ำ). เขตข้อมูลทั้งหมดนี้บรรยายลักษณะรูปร่างและการกระจายน้ำหนักของห้องเรือ และเขตข้อมูลนี้จะใช้เป็นอินพุตของโมเดล.

เขตข้อมูลที่ 7 (Residuary resistance per unit weight of displacement) เป็นค่าความต้านทานเหลือค้าง ซึ่งเป็นแรงต้านสำคัญที่เกิดกับเรือ และเกี่ยวพันกับลักษณะต่างๆของเรือ ที่วัดเป็นคุณสมบัติและแสดงในเขตข้อมูลที่ 1-6. การประมาณค่าความต้านทานเหลือค้างนี้ได้ดีจะช่วยอย่างมากในการประเมินสมบัติของเรือ รวมถึงช่วยเป็นข้อมูลประกอบสำหรับการออกแบบเรือด้วย เช่น การเลือกรูปทรงและขนาดของห้องเรือ การกำหนดน้ำหนักบรรทุกของเรือ การเลือกขนาดของเครื่องยนต์ที่เหมาะสม เป็นต้น. (ดูคำอธิบายเพิ่มเติมจากคลังข้อมูลยูซีโอ และอรทิโกสาและคณะ[61])

จากข้อมูล 308 จุด เราแบ่งข้อมูล 215 จุด (70%) เป็นข้อมูลชุดฝึก และใช้ที่เหลือสำหรับการทดสอบ. เราใช้โครงข่ายขนาด 10 หน่วยอยู่ในการทำการหาค่าลดตอนนี้. รูป 6.6 แสดงค่าผิดพลาดระหว่างการฝึกและผลการทดสอบ. ผลการทดสอบในภาพขวามือแสดงให้เห็นว่าโครงข่ายประสาทเทียมที่ฝึกแล้วทำงานได้ดีพอสมควร ดังเห็นได้จากค่าความผิดพลาด (RMSE) ของชุดทดสอบมีค่าต่ำ และค่าเออร์พุตจากโครงข่ายใกล้กับค่าจริงมาก (ผลการคาดค่าเออร์พุตจากโครงข่ายกับค่าจริงเรียงตัวเกือบทั้งส่วน  $y = x$ ).

สังเกตว่า ค่าผิดพลาดระหว่างการฝึก (ภาพขามือ) บางครั้งกลับมีค่าเพิ่มขึ้นได้. ที่เป็นเช่นนี้ก็ เพราะว่า ค่าอัตราการเรียนรู้ที่เลือกใช้อาจมีค่าสูงเกินไปเล็กน้อย ( $\text{rho}_h=0.001$ ,  $\text{rho}_o=0.0003$ , ดูหัวข้อ 6.5.2). แต่เนื่องจากการฝึกสามารถดำเนินต่อไปได้ (ดูจากค่าผิดพลาดที่หลังจากนั้นก็ลดลงอย่างต่อเนื่อง) ค่าอัตราการเรียนรู้ชุดนี้ก็ถือว่าใช้ได้ (ค่าอัตราการเรียนรู้ไม่มากเกินไปจนทำให้รีลงชันที่สุดล้มเหลว)

ตัวอย่างนี้เลือกใช้ค่าอัตราการเรียนรู้ให้เป็นค่าคงที่ตลอดการฝึก. การเลือกค่าน้อยอาจช่วยให้ค่าผิดพลาดลดลงต่อเนื่องอย่างดี (ค่าพารามิเตอร์ลู่เข้าค่าทำงานอยู่ที่สุดท้องถิ่น) แต่ก็อาจต้องเพิ่มจำนวนรอบฝึกเพื่อชดเชย (ส่งผลโดยอ้อมทำให้ฝึกเสร็จช้าลง). การเลือกใช้ค่าอัตราการเรียนรู้ค่าใหม่ขึ้นทำให้ต้องการรอบฝึกน้อยลง (ส่งผลโดยอ้อมให้ฝึกเสร็จเร็วขึ้น) แต่หากใช้ค่าอัตราการเรียนรู้ค่าใหม่เกินไปจะทำให้ค่าพารามิเตอร์ไม่ลู่เข้า และทำให้การฝึกไม่มีเสถียรภาพและ อาจจะทำให้ผลการฝึกล้มเหลวได้.

ในการประยุกต์ใช้งานโครงข่ายประสาทเทียม ผู้ใช้ต้องตรวจสอบกระบวนการฝึก เช่น ตรวจสอบดูค่าผิดพลาดต่อรอบฝึก ดังภาพข่ายในรูป 6.6 เพื่อตรวจสอบว่าการฝึกเป็นไปด้วยดี. หากค่าผิดพลาดเพิ่มขึ้นเรื่อยๆ หรือมีการเปลี่ยนแปลงค่ารุนแรงมาก นั่นอาจเป็นสัญญาณว่าอัตราการเรียนรู้มีค่ามากเกินไป. แต่หากแนวโน้มของค่าผิดพลาดยังลดลงอยู่เมื่อครบจำนวนรอบฝึกแล้ว นั่นอาจบ่งบอกถึงการครองลงเพิ่มจำนวนรอบฝึกดู. ค่าผิดพลาดต่อรอบฝึกของภาพข่ายในรูป 6.6 มีลักษณะราบตอนปลายๆ (ค่าผิดพลาดไม่ค่อยลดลงอีก) แสดงถึงว่าจำนวนรอบฝึกน่าจะเพียงพอแล้ว.

เรื่องค่าอัตราการเรียนรู้นั้น ไม่จำเป็นที่ต้องเลือกใช้อัตราการเรียนรู้ที่มีค่าคงที่ตลอดการฝึก. การฝึกโครงข่ายอาจทำโดยเลือกใช้อัตราการเรียนรู้ที่มีค่าสูงๆในตอนต้นๆ และปรับค่าลดลงเมื่อฝึกไปสักพักแล้ว

ก็ได้ นอกจากนั้น การปรับค่าอัตราการเรียนรู้ตามผลการทำงานของโมเดลก็สามารถทำได้ วิธีที่เลือกมาฝึกโมเดลก็จะเกี่ยวพันโดยตรงกับค่าอัตราการเรียนรู้ เพราะค่าอัตราการเรียนรู้สำหรับการฝึกโครงข่ายประสาทเทียมก็คือค่าขนาดก้าว (step size) ของ วิธีลิงเกรเดียนต์ที่เลือกใช้นั่นเอง.

ดังที่อภิรายไปแล้วว่า การฝึกโครงข่ายก็คือการหาค่า'n้ำหนัก'ที่ทำให้ฟังชั่นเป้าหมายมีค่าน้อยที่สุด. ดังนั้น วิธีการหาค่าน้อยที่สุดที่มีประสิทธิภาพต่างๆ เช่น บีเอฟจีเอส (BFGS) หรือ คอนจูเกตเกรเดียนต์ (Conjugate Gradient) ก็สามารถนำมาใช้ในการฝึกโครงข่ายได้. นอกจากนั้น ยังมีวิธีที่ออกแบบมาเฉพาะสำหรับฝึกโครงข่ายและมีประสิทธิภาพมากกว่าวิธีลิงเกรเดียนต์อย่างวิธี เช่น วิธีเลเวนเบร็กมาร์คอร์ต (Levenberg Marquardt[32]), เรซิลิエンต์แบกพรอพาเกชัน (Resilient Backpropagation, คำย่อ RPROP[65]), และ สเกลคอนจูเกตเกรเดียนต์ (Scaled Conjugate Gradient, คำย่อ SCG[54]) เป็นต้น. หัวข้อ 7.2 และ 7.3.1 แสดงตัวอย่างการใช้วิธีบีเอฟจีเอสและสเกลคอนจูเกตเกรเดียนต์ในการฝึกโครงข่าย.

### 6.3 ตัวอย่างการจำแนกประเภท

หัวข้อนี้แสดงตัวอย่างการจำแนกประเภท ด้วยข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ (Mammographic Mass) จากคลังข้อมูลชุดที่ [25] ให้ที่ <http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass>. ข้อมูลชุดนี้ เอลเตอร์และคณะ[25] ใช้ศึกษาการทำนายผลการตรวจภาพเอ็กซเรย์เต้านมของร้องรอยมวลเนื้อ (mammographic mass lesion) ว่าเป็นเนื้อดี (benign) หรือเนื้อร้าย (malignant) จากค่าไบแรตส์ (BI-RADS) และอายุของผู้ป่วย. วิธีตรวจภาพเอ็กซเรย์เต้านม (Mammography) เป็นวิธีที่มีประสิทธิผลมากที่สุดในการตรวจมะเร็งทรวงอก [25]. ข้อมูลชุดนี้ประกอบด้วย ค่าการประเมินไบแรตส์ (BI-RADS assessment ที่มีค่าในช่วง  $\{1, 2, 3, \dots, 5\}$ ), อายุของผู้ป่วย (เลขจำนวนเต็มของอายุ หน่วยเป็นปี), รูปทรงของมวลเนื้อ (mass shape ซึ่งถูกแทนด้วย 1 สำหรับทรงกลม round, 2 สำหรับทรงรี oval, 3 สำหรับทรงกลีบยอด lobular, 4 สำหรับทรงที่ผิดแปลง irregular), ลักษณะขอบของมวลเนื้อ (mass margin ซึ่งถูกแทนด้วย 1 สำหรับเขตรอบขั้ดเจน circumscribed, 2 สำหรับขอบเขตเป็นกลีบยอดๆ microlobulated, 3 สำหรับขอบเขตคลุมเครื่อง obscured, 4 สำหรับขอบเขตยากจะระบุ ill-defined, 5 สำหรับขอบเขตเป็นลักษณะ命名หรือปุ่ม spiculated), ความหนาแน่นของมวลเนื้อ (mass density ซึ่งถูกแทนด้วย 1 สำหรับความหนาแน่นสูง high, 2 สำหรับความหนาแน่นกลางๆ iso, 3 สำหรับความหนาแน่นต่ำ low, 4 สำหรับมวลเนื้อมีไขมันอยู่ fat-containing), และความร้ายแรง (severity ซึ่งมีสองค่า 0 สำหรับเนื้อดี benign หรือ 1 สำหรับเนื้อร้าย malignant). ค่าความร้ายแรงคือค่าที่ต้องการทำนาย (ว่าเป็นเนื้อดีหรือเนื้อร้าย). การสามารถทำนายค่าความร้ายแรงได้อย่างแม่นยำจะช่วยให้แพทย์สามารถตัดสินใจได้ดีขึ้นว่า ควรจะทำการตัดเนื้อจากบริเวณที่สงสัยออกตรวจเพื่อยืนยันผลหรือไม่.

กระบวนการรักษามะเร็งเป็นกระบวนการที่มีผลกระทบมาก เช่น นอกเหนือไปจากความเสี่ยงอื่นๆ ผู้รับการรักษาจะเจ็บปวดทรมานจากกระบวนการรักษาเองด้วย. ดังนั้นการวินิจฉัยผู้เข้ารับกระบวนการรักษามะเร็งจึงมีความสำคัญมาก ที่จะระบุผู้ที่

ต้องการรับการรักษาอย่างถูกต้อง และผลกระทบจากการระบุผิดพลาดมีผลเสียหายมากทั้งสองทางไม่ว่า ระบุผิดว่าผู้ตรวจเป็นมะเร็งโดยที่ไม่ได้เป็น (false positive) หรือระบุผิดว่าผู้ตรวจไม่ได้เป็นโดยที่ผู้ตรวจเป็น (false negative).

การตรวจที่ผลการตรวจน่าเชื่อถือมากที่สุดคือการตัดเนื้อออกตรวจ (biopsy). แต่การทำการตัดเนื้อออกตรวจ ผู้ตรวจต้องเข้ารับการผ่าตัด. ซึ่งจุดประสงค์ของการทำวินิจฉัยเบื้องต้นด้วยภาพเอ็กซเรย์เต้านมของมวลเนื้อ ก็เพื่อลดการทำการทำการตัดเนื้อออกตรวจที่ไม่จำเป็นลงไป. หากการวินิจฉัยเบื้องต้นด้วยภาพเอ็กซเรย์เต้านมของมวลเนื้อสามารถให้ผลที่เชื่อถือได้ ผู้ตรวจที่ผลเบื้องต้นออกเป็นเนื้อดี ก็จะได้เพียงแต่สังเกตุอาการ โดยไม่จำเป็นต้องเข้าทำการตัดเนื้อออกตรวจ.

ข้อมูลชุดนี้มี 961 ระเบียน (961 จุดข้อมูล), เฉลย หรือ ผลการตรวจจริง (ground truth) มี 516 ระเบียนที่ผลเป็นเนื้อดี และ 445 ระเบียนที่ผลเป็นเนื้อร้าย.

ข้อมูลชุดนี้มีค่าบางค่าของเขตข้อมูลที่ไม่ครบ (missing attribute values) ได้แก่ ค่าการประเมินไปแทนส์ขาดไป 2 ค่า, อายุขาดไป 5 ค่า, รูปทรงของมวลเนื้อขาดไป 31 ค่า, ลักษณะขอบของมวลเนื้อขาดไป 48 ค่า, ความหนาแน่นของมวลเนื้อขาดไป 76 ค่า, ความร้ายแรงมีค่ารอบทุกรอบเปลี่ยน.

**การจัดการกับค่าที่ขาดไปของบางเขตข้อมูล (missing data)** มีหลายวิธีที่นิยมใช้จัดการกับกรณีที่ข้อมูลมีบางระเบียนที่มีข้อมูลไม่ครบทุกเขตข้อมูล (missing data) วิธีง่ายๆ เช่น (1) การตัดจุดข้อมูลที่มีค่าไม่ครบทั้งไปเลย หรือ (2) การแทนค่าข้อมูลที่ขาดหายไปด้วยค่าเฉลี่ย (สำหรับเขตข้อมูลค่าต่อเนื่อง continuous-value field) หรือแทนด้วยค่าที่พบรอยที่สุดในเขตข้อมูลนั้น (สำหรับเขตข้อมูลที่เป็นลักษณะฉลากของหมวดหมู่ categorical field) หรือ (3) การแทนค่าที่หายไปด้วยทุกค่าที่เป็นไปได้ นั่นคือจุดข้อมูลที่ค่าเขตข้อมูลหายไปจะถูกแทนด้วยจุดข้อมูลใหม่หลายๆ จุด โดยที่จุดใหม่แต่ละจุดจะมีค่าเหมือนจุดเดิม ยกเว้นค่าที่หายไปแทนด้วยค่าที่เป็นไปได้ค่าหนึ่ง ดังนั้นจำนวนจุดข้อมูลใหม่ที่เพิ่มมาแทนนี้จะเท่ากับจำนวนค่าที่เป็นไปได้ของค่าเขตข้อมูลที่หายไป. การทดลองของกรีซีมาลา-บุสเสและญี่ปุ่น[31]พบว่า การแทนค่าที่หายไปด้วยค่าที่พบรอยที่สุดนั้น แม้เป็นวิธีที่ทำได้ง่าย แต่ไม่ใช่วิธีให้ผลที่ดีเลย. ตัวอย่างวิธีที่นำไปใช้ 6.5.3 แสดงการใช้วิธีแทนค่าที่หายไปด้วยทุกค่าของเขตข้อมูล ซึ่งเป็นวิธีที่กรีซีมาลา-บุสเสและญี่ปุ่นนำสำหรับผลการทำงาน แต่เป็นวิธีที่ค่อนข้างยุ่งยากในการทำ. ดูกรีซีมาลา-บุสเสและญี่ปุ่น[31]เพิ่มเติมสำหรับวิธีต่างๆ ในการจัดการข้อมูลที่ขาดหายไป และผลการทดลองเบรียบเทียบวิธีการต่างๆ.

สังเกตุโครงข่ายประสาทเทียมแบบจ่ายไปข้างหน้าสองชั้นมีสิ่งที่ผู้ใช้ต้องกำหนดอยู่หลายอย่าง เช่น จำนวนหน่วยซ่อน, วิธีการฝึก ที่แม้จะเลือกใช้วิธีลงเกรเดียนต์แล้วก็ยังต้องเลือก ค่าอัตราการเรียนรู้ และจำนวนรอบฝึก เป็นต้น. สิ่งต่างๆ ที่ต้องเลือกเหล่านี้ก็สามารถมองเป็นพารามิเตอร์ของโมเดลได้เช่นกัน และเพื่อกันการสับสนกับค่าน้ำหนัก (ซึ่งก็เป็น พารามิเตอร์ของโมเดล) เราจะเรียกพารามิเตอร์ที่เป็นเหมือนตัวกำหนดลักษณะใหญ่ของโมเดลว่า ไฮเปอร์พารามิเตอร์ (hyperparameters) เพื่อให้ต่างจากพารามิเตอร์ ซึ่งใช้อ้างถึงค่าน้ำหนัก. โดยปกติ กระบวนการคือเลือกค่าของไฮเปอร์พารามิเตอร์ก่อน และวิจัยนำโมเดลไปฝึกเพื่อหาค่าที่ดีของพารามิเตอร์ของโมเดล (ซึ่งสำหรับโครงข่ายประสาทเทียมพารามิเตอร์ก็คือค่าน้ำหนัก).

เช่นเดียวกับวิธีการเลือกโมเดลที่ยกในหัวข้อ 3.2, วิธีครอสвалиเดชั่น (cross-validation) สามารถนำมาช่วยในการเลือกค่าไฮเปอร์พารามิเตอร์เหล่านี้ได้. วิธีครอสвалиเดชั่นแบบสิบพับเป็นที่นิยมใช้มาก. บาง

งานวิจัย เช่น บทความของแมคแแคฟฟรีย์[51] แนะนำว่า โดยทั่วไปการใช้วิธีครอสвалиเดชั้นสิบพับเป็นตัวเลือกที่ดี. แต่จุดประสงค์ของตัวอย่างนี้คือ เพื่อแสดงให้เห็นการนำโครงข่ายประสาทเทียมไปประยุกต์ใช้ และเพื่อลดเวลาในการเตรียมตัวอย่าง หัวข้อนี้แสดงตัวอย่างที่ใช้วิธีครอสвалиเดชั้นแบบห้าพับ (5-fold cross-validation). วิธีแบบสิบพับก็สามารถทำได้ในลักษณะเดียวกัน.

ทบทวนอีกรัง จุดประสงค์ของวิธีครอสвалиเดชั้นคือเพื่อช่วยในการเลือกโมเดล หรือในกรณีนี้คือเลือกค่าของไฮเปอร์พารามิเตอร์ที่จะใช้. ค่าไฮเปอร์พารามิเตอร์ที่สนใจในที่นี้คือ จำนวนหน่วยซ่อน ( $M$ : 5, 10, 30) ที่ฝึกด้วยวิธีลงเกรเดียนต์กับอัตราการเรียนรู้ ( $\{\rho_1, \rho_2\}$ :  $\{0.01, 0.001\}$ ,  $\{0.01, 0.01\}$ ,  $\{0.001, 0.0\}$ ) และทำการฝึก 5000 รอบฝึกสำหรับทุกชุดของค่าไฮเปอร์พารามิเตอร์. ดังนั้น นั่นคือ เท่ากับว่ามีไฮเปอร์พารามิเตอร์ที่สนใจอยู่ 9 ชุด แต่ละชุดจะทำการฝึกและทดสอบ 5 ครั้ง (เนื่องจากทำวิธีครอสвалиเดชั้นแบบห้าพับ) ประสิทธิภาพของไฮเปอร์พารามิเตอร์แต่ละชุด จะได้จากการเฉลี่ยค่าความแม่นยำ<sup>1</sup> ของทั้ง 5 ครั้ง.

ตาราง 6.1 แสดงผลจากการทำครอสвалиเดชั้นห้าพับ. คอลัมน์  $M$ , คอลัมน์  $\rho_1$ , และ คอลัมน์  $\rho_2$  แสดงชุดของไฮเปอร์พารามิเตอร์ที่สนใจ. คอลัมน์ พับ 1 ถึง คอลัมน์ พับ 5 แสดงค่าความแม่นยำของแต่ละพับ. คอลัมน์ เฉลี่ย แทนค่าเฉลี่ยของค่าความแม่นยำของทั้ง 5 พับ. ค่าความแม่นยำ คืออัตราการทำนายถูก,  $r = \frac{\text{Count}(\mathbf{y}=\mathbf{t})}{N}$  เมื่อ  $N$  คือ จำนวนจุดข้อมูลที่ทดสอบ และ  $\text{Count}(\mathbf{y} = \mathbf{t})$  แทนจำนวนที่ค่าที่ทำนายตรงกับผลจริง นั่นคือ

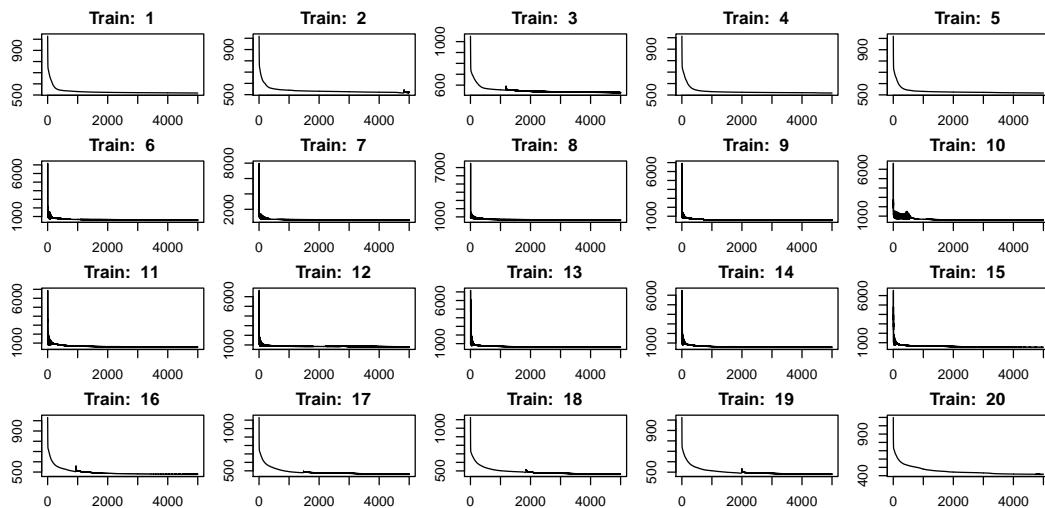
$$\begin{aligned}\text{Count}(\mathbf{y} = \mathbf{t}) &= \sum_{i=1}^N \delta(y_i, t_i), \\ \delta(y_i, t_i) &= \begin{cases} 0 & \text{เมื่อ } y_i \neq t_i, \\ 1 & \text{เมื่อ } y_i = t_i, \end{cases}\end{aligned}$$

โดย  $y_i$  และ  $t_i$  คือค่าที่ทำนายและผลจริงของจุดข้อมูลที่  $i$  ตามลำดับ. ดังนั้น ค่า  $r$  จะอยู่ระหว่าง 0 กับ 1.

ตารางที่ 6.1: ผลจากการทำครอสвалиเดชั้นห้าพับของค่าไฮเปอร์พารามิเตอร์ชุดต่างๆ

$M$	$\rho_1$	$\rho_2$	ค่าความแม่นยำ					
			พับ 1	พับ 2	พับ 3	พับ 4	พับ 5	เฉลี่ย
5	0.01	0.001	0.8297	0.8401	0.8428	0.8537	0.8320	0.8397
5	0.01	0.01	0.8054	0.8238	0.8293	0.8428	0.8211	0.8245
5	0.001	0.01	0.8459	0.7886	0.8645	0.8537	0.8266	0.8359
10	0.01	0.001	0.8541	0.8266	0.8618	0.8509	0.8103	0.8407
10	0.01	0.01	0.8054	0.8455	0.8780	0.8103	0.8374	0.8353
10	0.001	0.01	0.8568	0.8130	0.8591	0.8537	0.8347	0.8434
30	0.01	0.001	0.8162	0.8293	0.8753	0.8320	0.7913	0.8288
30	0.01	0.01	0.8297	0.8428	0.8780	0.8455	0.7940	0.8380
30	0.001	0.01	0.8595	0.8428	0.8509	0.8591	0.8130	0.8451

<sup>1</sup>บ่อยครั้ง ที่ผลมักจะแสดงออกมาในรูปอัตราการทำนาย แต่ในที่นี้ใช้ค่าความแม่นยำ หรืออาจเรียกว่าอัตราการทำนายถูก. ทั้งนี้ทั้งนั้น ค่าความแม่นยำและอัตราการทำนายผิดสัมพันธ์กัน โดย อัตราการทำนายผิด =  $1 - \text{ค่าความแม่นยำ}$ .



รูปที่ 6.7: ตัวอย่างแสดงค่าฟังชันเบ้าหมายระหว่างการฝึกของการฝึกและประเมินผลครั้งที่ 1 ถึง 20 (จาก 45 ครั้ง).

การทำวิเคราะห์เชิงคือเพื่อเลือกโมเดล เช่นเลือกชุดของไฮเปอร์พารามิเตอร์ที่ดีที่สุดสำหรับคุณสมบัติความทั่วไป (generalization). จากผลที่ได้มาพบว่าชุดค่า  $M = 30, \rho_1 = 0.001, \text{ และ } \rho_2 = 0.01$  ให้ผลดีที่สุดที่ค่าความแม่นยำเฉลี่ย 0.8451. โดยอันดับสองคือชุดค่า  $M = 10, \rho_1 = 0.001, \rho_2 = 0.01$  ที่ค่าความแม่นยำเฉลี่ย 0.8434. ดังนั้น ตัวอย่างนี้จะเลือกใช้โครงข่ายขนาดจำนวนหน่วยซ่อน 30 หน่วย และใช้อัตราการเรียนรู้เป็น  $\rho_1 = 0.001$  และ  $\rho_2 = 0.01$ .

หมายเหตุ ในทางปฏิบัติ อาจเลือกใช้ชุด 10, 0.001, 0.01 แทนก็ได้ เพราะค่าเฉลี่ยต่างกันไม่มาก และการใช้หน่วยอยู่ 10 หน่วย จะทำให้การคำนวนทำได้เร็วกว่า 30 หน่วย. แต่อย่างไรก็ตาม ผลของโครงสร้างเดิมจะใช้ค่าเฉลี่ยเป็นหลัก ไม่ใช้ค่าที่ดีที่สุด. ค่าที่ดีที่สุดในตารางคือ 0.8780 จาก พับที่ 3 ของ  $M = 10, \rho_1 = 0.010, \text{ และ } \rho_2 = 0.010$ . ค่าที่ดีที่สุดจากแต่ละพับอาจเกิดจากการที่พับนั้นๆ เลือกข้อมูลชุดที่ง่ายไปทำการทดสอบก็ได้. แต่จุดประสงค์คือการเลือกโมเดลที่ใช้กับข้อมูลได้ดีโดยทั่วไป (คุณสมบัติความทั่วไป). นั่นคือด้วยทั่วไป ไม่ใช่ดีมากๆ ในบางครั้ง แต่กลับแย่มากบางครั้ง (คุณสมบัติความทั่วไป คือความคงเส้นคงวาของคุณภาพการทำงาน). ดังนั้นในการเลือกโมเดลด้วยวิธีโครงสร้างเดิม จึงใช้ค่าเฉลี่ย ซึ่งเป็นค่าตัวแทนของสถานะการณ์ทั่วไป.

หนึ่งในสิ่งสำคัญที่ควรตรวจสอบก่อนสรุปตามผลที่ได้คือ ตรวจสอบว่ากระบวนการฝึกเป็นได้ด้วยดี เช่น ตรวจสอบค่าฟังชันเบ้าหมายต่อรอบฝึก ดังแสดงในรูป 6.7 ว่าค่าฟังชันเบ้าหมายมีค่าลดลงจนเกือบคงที่. หากค่าฟังชันเบ้าหมายมีค่าลดลงอยู่ แต่ยังไม่ถึงจุดคงที่ ผู้สร้างโมเดลอាជะพิจารณาเพิ่มรอบการฝึกขึ้น. หรือหากค่าฟังชันเบ้าหมายกลับเพิ่มขึ้นอย่างมาก ค่าอัตราการเรียนรู้ที่เลือกใช้อาจมากเกินไป.

จากไฮเปอร์พารามิเตอร์ทั้ง 9 ชุดที่สนใจ และการทำโครงสร้างเดิมห้าพับสำหรับแต่ละชุด ทำให้มีการฝึกและประเมินผลทั้งหมดรวม 45 ครั้ง. ผู้สร้างโมเดลควรจะตรวจสอบการฝึกทั้ง 45 ครั้งนี้. รูป 6.7 แสดงตัวอย่างของค่าฟังชันเบ้าหมายระหว่างการฝึก ของการฝึกและประเมินผลครั้งที่ 1 ถึง 20. เส้นกราฟของค่าฟังชันเบ้าหมายลดลงจนประมาณคงที่ในทุกๆ รอบปัจจุบัน แสดงให้เห็นว่าการฝึกทั้งหมดเป็นไปด้วยดี. สังเกตุว่า การฝึกครั้งที่ 10 ช่วงต้นมีการเปลี่ยนแปลงค่าฟังชันเบ้าหมายต่อรอบฝึกค่อนข้างรุนแรง. แต่ภายหลังจา

กราว่า 1,000 รอบฝึก ค่าฟังชันเป้าหมายก็ลดลงด้วยดี จนเกือบคงที่ในรอบการฝึกทั้งๆ.

หลังจากได้ข้อมูลไปแล้ว การสร้างโมเดลจะใช้ข้อมูลทั้งหมดทุกพับในการฝึก โมเดล และการประเมินประสิทธิภาพของโมเดลสุดท้ายที่ได้ ก็ทำด้วยข้อมูลที่แยกไว้ต่างหากอีกชุด (ไม่อยู่ในพับใดๆ). ซึ่งสุดท้ายแล้ว ตัวอย่างนี้ได้โมเดลที่มีค่าความแม่นยำเป็น 0.842 (หรือ อัตราทายผิด 0.158). ตาราง 6.2 แจกแจงผลด้วยเมตริกซ์ความสับสน (confusion matrix) ที่แสดง จำนวนที่ทายว่าเป็นกลุ่ม 1 และผลจริงก็เป็นกลุ่ม 1 (จำนวนบวกจริง, true positive), จำนวนที่ทายว่าเป็นกลุ่ม 1 แต่ผลจริงเป็นกลุ่ม 0 (จำนวนบวกเท็จ, false positive), จำนวนที่ทายว่าเป็นกลุ่ม 0 แต่ผลจริงเป็นกลุ่ม 1 (จำนวนลบเท็จ, false negative), และ จำนวนที่ทายว่าเป็นกลุ่ม 0 และผลจริงก็เป็นกลุ่ม 0 (จำนวนลบจริง, true negative). นอกจากนั้น ค่าความเที่ยงตรง (precision) และค่าการเรียกกลับ (recall) ก็แสดงด้านซ้ายและด้านล่างของตารางตามลำดับ.

ตารางที่ 6.2: ผลประเมินของโครงข่ายที่ได้แสดงในรูปเมตริกซ์ความสับสน, ค่าความเที่ยงตรง, และค่าการเรียกกลับ

		ผลจริง		
		1	0	
ผลทั้งหมด	1	true positive 162	false positive 17	Precision = 0.905
	0	false negative 56	true negative 227	Recall = 0.743

ถ้าจะอธิบายถึงค่าความเที่ยงตรงและค่าการเรียกกลับ, ก็ควรอภิปรายถึงความสมดุลของการกระจายของกลุ่มข้อมูลก่อน. ข้อมูลชุดนี้มีการกระจายข้อมูลพอๆ กัน ของข้อมูลกลุ่ม 1 (ผลเป็นเนื้อร้าย malignant) และข้อมูลกลุ่ม 0 (ผลเป็นเนื้อดี benign). กล่าวคือ มีจำนวนระเบียนใกล้เคียงกัน นั่นคือ 445 ระเบียน (กลุ่ม 1) และ 516 ระเบียน (กลุ่ม 0). ลักษณะการกระจายเช่นนี้ทำให้ค่าความแม่นยำสูงท่อนกับคุณภาพการทำนายจริงของโมเดลได้ดี. แต่หากการกระจายไม่สมดุลอย่างมาก เช่น สมมติอัตราส่วนคนเป็นมะเร็ง ตับอ่อนต่อประชากรมีค่าน้อยกว่า 1% เพียงแต่โมเดลทำนายผลเป็นไม่เป็นมะเร็งสำหรับทุกๆ ตัวอย่างที่เข้ามาทดสอบ โมเดลนั้นก็มีโอกาสถูกถูกถึง 99% (ความแม่นยำ เป็น 0.99) นั่นคือ เท่ากับทายว่าไม่มีใครเป็นมะเร็งตับอ่อนเลย ซึ่งแม้จะมีโอกาสถูกถูกสูงมากๆ แต่มันไม่ได้มีประโยชน์ต่อการช่วยระบุกลุ่มเสี่ยงเลย.

ดังนั้นแทนที่จะใช้แค่ค่าความแม่นยำ, ค่าความเที่ยงตรง (สมการ 6.1) และค่าการเรียกคืน (สมการ 6.3) จะช่วยสูงท่อนคุณภาพการทำนายสำหรับข้อมูลที่มีการกระจายข้อมูลไม่สมดุลได้มากกว่า.

สำหรับงานจำแนกประเภทระหว่างสองกลุ่ม นั่นคือกลุ่มบวกและกลุ่มลบ. ค่าความเที่ยงตรงสามารถ

คำนวณได้จาก

$$\text{Precision} = \frac{\text{True Positive}}{\text{Predicted Positive}} \quad (6.1)$$

$$= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (6.2)$$

เมื่อ True Positive คือจำนวนจุดข้อมูลที่ทำนายเป็นกลุ่มบวกและผลเฉลยเป็นกลุ่มบวก (บวกจริง) และ False Positive คือจำนวนจุดข้อมูลที่ทำนายเป็นกลุ่มบวกแต่ผลเฉลยเป็นกลุ่มลบ (บวกเท็จ)

ในทำนองเดียวกัน ค่าการเรียกกลับสามารถคำนวณได้จาก

$$\text{Recall} = \frac{\text{True Positive}}{\text{Actual Positive}} \quad (6.3)$$

$$= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (6.4)$$

เมื่อ False Negative คือจำนวนจุดข้อมูลที่ทำนายเป็นกลุ่มลบแต่ผลเฉลยเป็นกลุ่มบวก (ลบเท็จ)

หมายเหตุ เมื่อจะใช้ค่าความเที่ยงตรงและค่าการเรียกกลับในการวัดผล จะกำหนดให้ กลุ่ม 1 (กลุ่มบวก) เป็นกลุ่มที่มีสัดส่วนน้อย (เช่น กลุ่มของมะเร็งตับอ่อน) และกลุ่ม 0 (กลุ่มลบ) เป็นกลุ่มใหญ่ (เช่น กลุ่มที่ไม่ได้เป็น).

ค่าความเที่ยงตรงและค่าการเรียกกลับจะช่วยให้ສهท้อนความสามารถของโมเดลได้ดีขึ้น โดยเฉพาะในกรณีที่ข้อมูลมีการกระจายระหว่างกลุ่มไม่สมดุล. เตต่อกำร มีผลแสดงเป็นตัวเลข 2 ค่านั้นอาจทำให้ลำบากในการเปรียบเทียบผล เช่น ผลของโมเดลหนึ่งอาจจะได้ค่าความเที่ยงตรงสูงแต่ค่าการเรียกกลับไม่มาก แต่ อีกโมเดลหนึ่งที่มีค่าความเที่ยงตรงต่ำแต่ค่าการเรียกกลับสูงมาก. ดังนั้น เพื่อความสะดวก ค่าคะแนนเอฟ ( $F$  Score หรือ บางครั้งเรียก  $F_1$  Score) ซึ่งเป็นค่าเฉลี่ยเชิงเรขาคณิตจึงมักใช้ในการสรุปค่าความเที่ยงตรงและค่าการเรียกกลับเป็นตัวเลขตัวเดียว. ค่าคะแนนเอฟสามารถคำนวณได้จาก,

$$F = 2 \cdot \frac{P \cdot R}{P + R} \quad (6.5)$$

เมื่อ  $F$  แทนค่าคะแนนเอฟ,  $P$  และ  $R$  แทนค่าค่าความเที่ยงตรงและค่าการเรียกกลับตามลำดับ. จากผลในตาราง 6.2 โมเดลในตัวอย่างจะมีค่าคะแนนเอฟเป็น  $2 \cdot \frac{0.905 \cdot 0.743}{0.905 + 0.743} = 0.816$ .

## 6.4 ตัวอย่างการจำแนกประเกทแบบหลายกลุ่ม

หัวข้อนี้อภิปรายการจำแนกประเกทแบบหลายกลุ่ม โดยใช้ตัวอย่าง ข้อมูลชุดรูปของลายมือเขียนตัวเลข. ข้อมูลชุดรูปของลายมือเขียนตัวเลข (Handwritten Digit Images) ได้มาจากการสแกนหน้าจคอมพิวเตอร์ สำนักงานไปรษณีย์สหรัฐอเมริกา (U.S. Postal Service). (ดู [46] เพิ่มเติมสำหรับรายละเอียด) ข้อมูลแต่ละภาพผ่านการจัดแนว (alignment), ซ่อมภาพเอียง (deslantation), และจัดขนาด (scaling). แต่ละ

ระเบียน(จุดข้อมูล) มี 256 มิติ (เก็บค่า 256 ค่า) สำหรับค่าระดับสีเทาของ 256 พิกเซลของแต่ละภาพ ลายมือเขียน ที่แต่ละภาพเป็นภาพของตัวเลขขนาด  $16 \times 16$  พิกเซล. ข้อมูลสามารถดาวน์โหลดได้จาก <http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/zip.info.txt>, [..zip.train.gz](#), และ [..zip.test.gz](#) สำหรับคำอธิบาย, ข้อมูลชุดฝึกหัด และข้อมูลชุดทดสอบ ตามลำดับ.

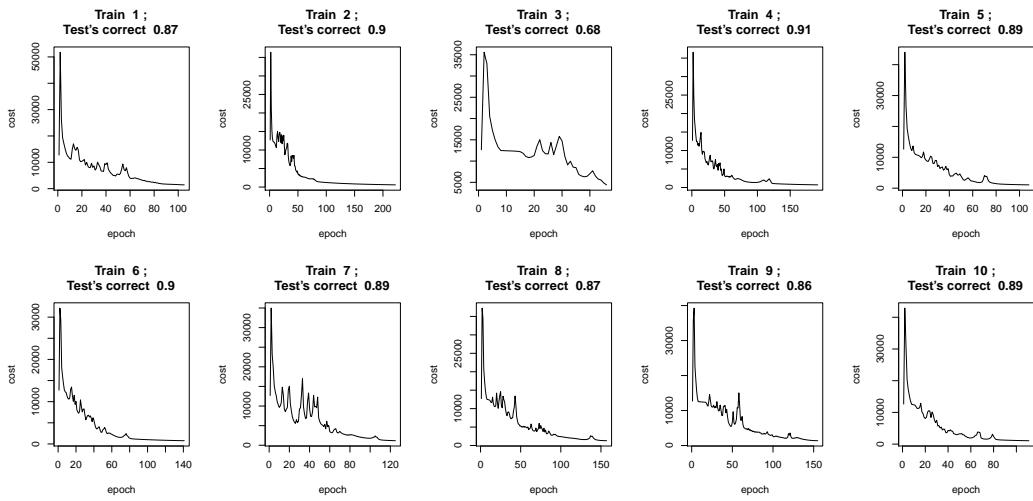
คำว่า “มิติ” มีความหมายเปลี่ยนแปลงตามบริบท. มิติของงานด้านภาพอาจทำให้ผู้อ่านสับสนกับมิติของจุดข้อมูล. ภาพสองมิติจะจัดเรียงตามแนวโนนและแนวตั้ง เช่น ภาพขนาด  $40 \times 30$  จะมีขนาด 40 พิกเซลตามแนวโนน และ 30 พิกเซลตามแนวตั้ง. แต่เมื่อนำภาพหนึ่งมาแปลงเป็นหนึ่งจุดข้อมูล ค่าของพิกเซลแต่ละค่าคือแต่ละมิติของจุดข้อมูล. นั่นคือ ภาพสองมิติขนาด  $40 \times 30$  หากแปลงเป็นจุดข้อมูล จะได้จุดข้อมูลที่มีขนาดมิติเป็น 1,200 มิติ.

ตัวอย่างแสดงการใช้โครงข่ายประสาทเทียมสองชั้น โดยที่ชั้นเอ้าต์พุตใช้ฟังก์ชันกระตุนเป็นฟังก์ชันซอฟต์แมกซ์ (softmax function). ประเด็นหนึ่งที่สำคัญคือการศึกษาความสัมพันธ์ระหว่างจำนวนหน่วยช่องกับผลการทำงาน. รูป 6.8 แสดงความก้าวหน้าของการฝึกทั้ง 10 ครั้ง ของโครงข่ายประสาทเทียมขนาด 40 หน่วยช่อง. ทุกครั้ง โครงข่ายประสาทเทียมถูกฝึกด้วยวิธีลิงเกรเดียนต์ และใช้ค่าอัตราการเรียนรู้คงที่ตลอด. นั่นคือ  $\rho_h = 0.0002$  และ  $\rho_o = 0.002$ , รอบฝึกสูงสุดคือ 500 รอบ, ทำการหยุดก่อนกำหนด เมื่อค่าผิดพลาดของชุดข้อมูลถูกลดเหลือเพิ่มขึ้นจากค่าที่ดีที่สุดมากกว่า 1 เท่าตัว. ค่าเริ่มต้นของค่าน้ำหนักทุกค่าสุ่มมาจากช่วง  $[-0.1, 0.1]$  ตามการแจกแจงเอกรูป (uniform distribution).

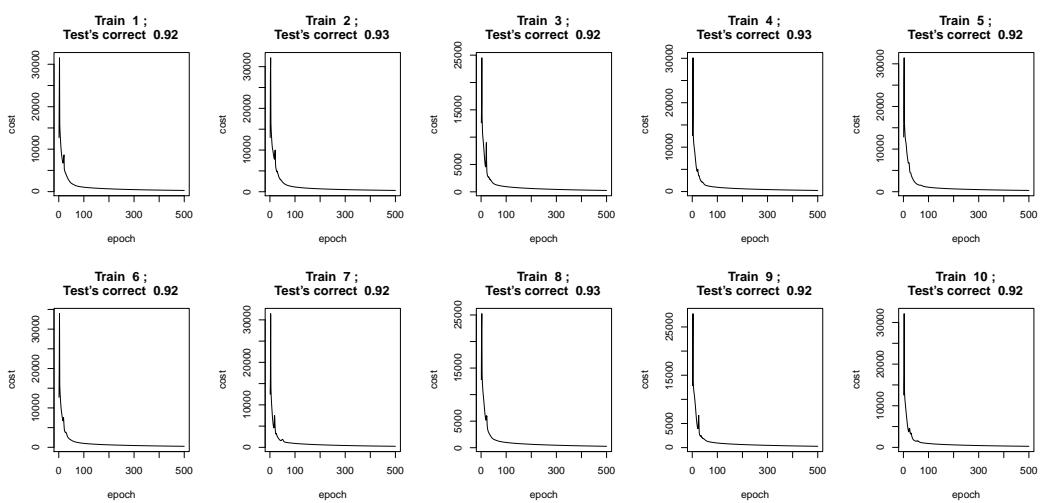
การที่ทำการฝึก 10 ครั้ง ก็เพื่อหารครั้งที่ดีที่สุด จากการสุ่มค่าน้ำหนักเริ่มต้น. สังเกตุ มีการฝึกหลายครั้ง ที่ยังไม่ถึงการถูกเข้า เช่น การฝึกที่ 3, 8 และ 9. การฝึกเหล่านี้แม้เพิ่มจำนวนรอบฝึกก็จะไม่ช่วยอะไร เพราะการฝึกทั้ง 10 ครั้งนี้ ทุกครั้งหยุดก่อน 500 รอบฝึก ซึ่งเป็นจำนวนรอบฝึกสูงสุดที่ตั้งไว้. ที่เป็นเช่นนี้ได้ ก็เพราะมีทำการหยุดก่อนกำหนด (Early Stopping ดูหัวขอ 6.1). ข้อสังนิษฐานเบื้องต้นคือ ค่าอัตราการเรียนรู้ที่ใช้อาจใช้ค่าสูงเกินไป ทำให้มีการฝึกเข้าใกล้ค่าที่ดีที่สุด แล้วค่าน้ำหนักถูกปรับให้เล็กๆ น้อยๆ แต่อาจทำให้ผลลัพธ์เดือนัยลง ซึ่งส่งผลให้เกิดการหยุดก่อนกำหนดขึ้น. การไม่ทำการหยุดก่อนกำหนด ก็อาจเป็นวิธีหนึ่ง แต่อาจนำปัญหาอื่นมาให้ เช่น อาจทำให้เสียเวลาฝึกนานขึ้น แต่กลับได้ไม่เดลที่โอเวอร์ฟิต กับข้อมูลฝึกหัด แต่ไม่สามารถทำนายได้ดีกับข้อมูลที่ต้องการใช้งานจริง (เสียคุณสมบัติความทั่วไป). สิ่งที่ควรทำสำหรับกรณีนี้ คือ ทดลองลดค่าอัตราการเรียนรู้ลง ได้แก่ ลองใช้  $\rho_h = 0.0001$  และ  $\rho_o = 0.001$ . ผลการฝึกด้วยค่าอัตราการเรียนรู้ที่น้อยลงแสดงดังรูป 6.9.

รูป 6.9 แสดงการถูกเข้าในทุกๆ การฝึก และทุกการฝึกสามารถทำจนครบรอบฝึกสูงสุด. ข้อสังนิษฐานเบื้องต้นน่าจะถูกต้อง. นั่นคือ ค่าอัตราการเรียนรู้ของ การฝึกครั้งแรกมากเกินไป (รูป 6.8). สังเกตุว่า นอกจากทุกๆ การฝึกถึงการถูกเข้า แล้วผลการทดสอบยังดีขึ้นด้วย. นั่นคือ ค่าความแม่นยำ (Test's correct ในรูป) ซึ่งคือ อัตราการทายถูกเมื่อทดสอบกับช้อมูลชุดทดสอบ มีค่าเพิ่มจากช่วงค่า 0.68 ถึง 0.91 ในครั้งแรก (รูป 6.8) เป็น 0.92 ถึง 0.93 (รูป 6.9).

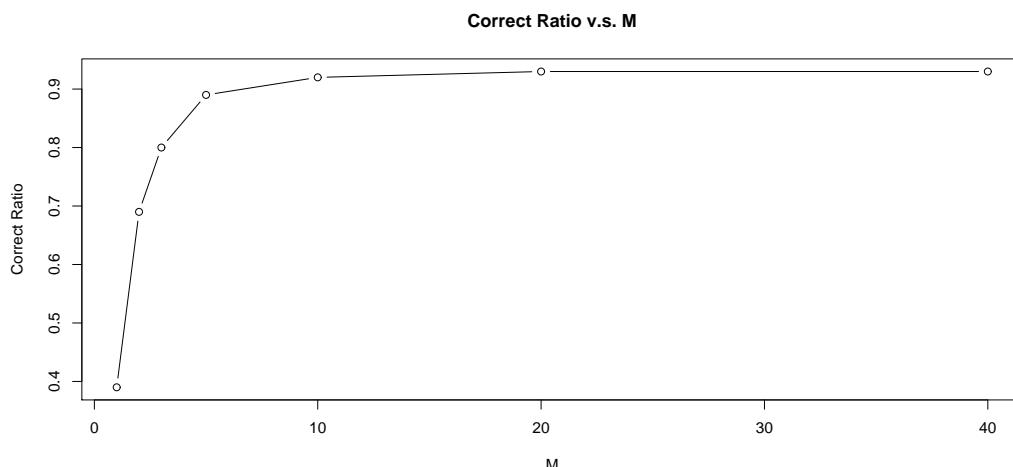
ในลักษณะเดียวกัน การทดลองฝึกโครงข่ายประสาทเทียมที่มีจำนวนหน่วยช่องต่างๆ ให้ผลดังแสดงในรูป 6.10. เช่นเดียวกับแนวปฏิบัติที่ดีในการฝึกโครงข่ายประสาทเทียมทั่วๆ ไป ต้องมีการตรวจสอบดูว่าการ



รูปที่ 6.8: ผลการฝึกโครงข่ายประสาทเทียมขนาดเล็กบีบหน่วยซ่อน 10 ครั้ง. สังเกตุการที่การฝึกหยุดก่อนจำนวนรอบการฝึกที่กำหนด (ซึ่งคือ 500 รอบฝึก) ก็เพราะการทำหยุดก่อนกำหนด และ อาจเป็นเพราะการเลือกใช้ค่าอัตราการเรียนรู้ที่สูงเกินไป.



รูปที่ 6.9: ผลการฝึกโครงข่ายประสาทเทียมขนาดเล็กบีบหน่วยซ่อน 10 ครั้ง ที่ใช้ค่าอัตราการเรียนรู้น้อยลง จากการฝึกที่แสดงในรูป 6.8.



รูปที่ 6.10: ผลการฝึกโครงข่ายประสาทเทียมขนาดหน่วยซ่อนต่างๆ.

ฝึกเป็นไปอย่างสมบูรณ์ เพราะการจะสรุปผลได้อย่างถูกต้องว่าจำนวนหน่วยซ่อนเท่าใดดีกว่า ต้องอยู่บนพื้นฐานว่าการฝึกทำได้อย่างสมบูรณ์สำหรับจำนวนหน่วยซ่อนนั้นๆแล้ว.

สังเกตว่า หากจำนวนหน่วยซ่อนน้อยเกินไป การเพิ่มจำนวนหน่วยซ่อนสามารถช่วยเพิ่มความถูกต้องในการจำแนกได้ แต่พอจำนวนหน่วยซ่อนเพียงพอแล้ว (20 หน่วยซ่อน) การเพิ่มจำนวนหน่วยซ่อน (เป็น 40 หน่วยซ่อน) ไม่ได้ช่วย เพิ่มความแม่นยำขึ้นเลย. การใช้หน่วยซ่อนมากเกินความจำเป็นไม่สามารถเพิ่มความแม่นยำขึ้นได้ แต่การคำนวณจะต้องทำการคำนวณค่า  $\mathbf{W}^{(1)}$  และ  $\mathbf{W}^{(2)}$  เป็นเมตริกซ์ขนาด  $[M \times (D + 1)]$  และ  $[K \times (M + 1)]$  ตั้งนั้น ที่ 20 หน่วยซ่อน, ค่าน้ำหนัก เป็นเมตริกซ์ขนาด  $[20 \times 257]$  และ  $[10 \times 21]$  ตามลำดับ และ ที่ 40 หน่วยซ่อน, ค่าน้ำหนัก เป็นเมตริกซ์ขนาด  $[40 \times 257]$  และ  $[10 \times 41]$  ตามลำดับ. ดังนั้นตัวอย่างนี้จึงเลือกใช้โครงข่ายประสาทเทียมที่มีจำนวน 20 หน่วยซ่อน และเมื่อนำผลทดสอบมาแสดงในเมตริกซ์แห่งความสับสน (confusion matrix) จะได้ตาราง 6.3.

ข้อมูลชุดทดสอบปีขนาด 2007 ระเบียน และโครงข่ายไทยผิดทั้งหมด 148 ระเบียน คิดเป็น 7.37%. ตัวอย่างของระเบียนที่ไทยผิดแสดงในรูป 6.11.

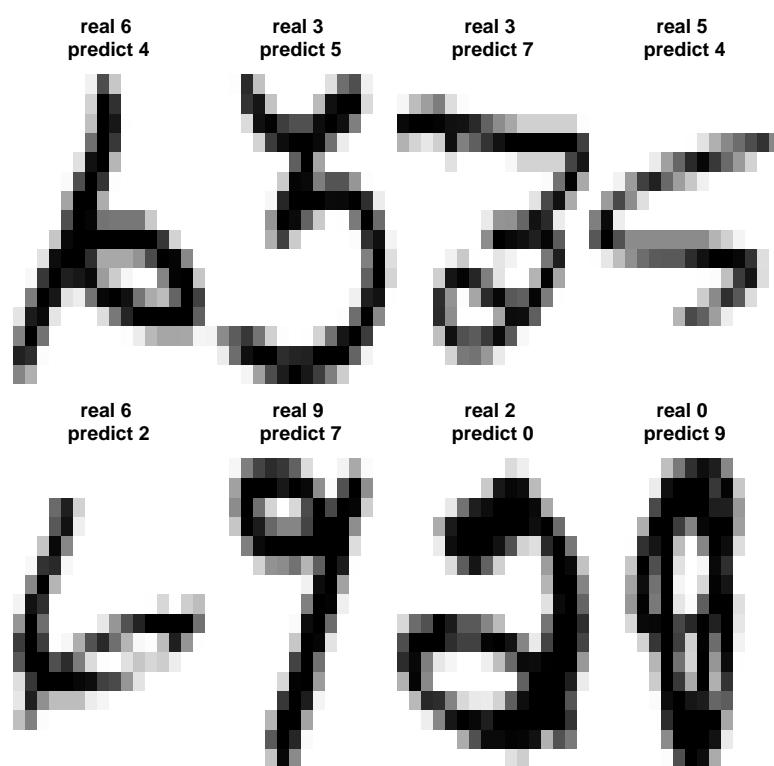
## 6.5 อาร์โค้ด

อาร์โค้ดในหัวข้อนี้ดัดแปลงมาจากอาร์โค้ดของสื่อประกอบการสอนของแอนเดอร์สัน<sup>2</sup>

<sup>2</sup>Charles Anderson, สื่อประกอบการเรียน วิชา CS545 Machine Learning, Fall 2006, Department of Computer Science, Colorado State University, Fort Collins, CO, USA.

ตารางที่ 6.3: ผลประเมินของโครงข่ายที่ได้ในรูปแบบตริกซ์แจงความสับสน

ผลทายกลุ่ม	กลุ่มจริง									
	0	1	2	3	4	5	6	7	8	9
0	348	0	3	2	2	3	0	0	5	0
1	0	254	0	0	2	0	0	0	0	2
2	2	0	178	3	3	0	2	1	2	1
3	2	0	3	146	0	7	0	1	4	0
4	2	2	4	1	180	3	3	7	0	3
5	1	0	1	9	3	143	4	0	4	0
6	2	2	1	0	2	0	161	0	1	0
7	0	2	2	2	1	1	0	134	0	3
8	1	1	6	2	2	1	0	1	149	2
9	1	3	0	1	5	2	0	3	1	166



รูปที่ 6.11: ตัวอย่างของรูปที่ทายผิด. กลุ่มที่ถูกต้อง (real) และ ผลการทาย (predict) เขียนกำกับไว้หนีอแต่ละรูป.

### 6.5.1 โค้ดสำหรับตัวอย่างง่ายๆ

รายการ 6.1 แสดงอาร์โค้ดสำหรับซิกมอยด์ฟังชัน (sigmoid), อนุพันธ์ของซิกมอยด์ฟังชัน (dsigmoid), และฟังชันคำนวณค่าของโครงข่าย (nnOutput). โดยฟังชัน nnOutput รับค่าพารามิเตอร์ของโครงข่าย (ตัวแปร net), อินพุต (ตัวแปร X), และชนิดปัญหาที่โครงข่ายทำงาน (ตัวแปร nntype). ตัวแปร net เป็นตัวแปรชนิด list ที่ต้องประกอบด้วย net\$W1 และ net\$W2 ที่เป็นเมตริกซ์ ขนาด  $M \times (D + 1)$  และ  $K \times (M + 1)$  สำหรับค่าน้ำหนักชั้นที่ 1 และ 2 ตามลำดับ, เมื่อ  $D$  คือ จำนวนมิติของอินพุต X;  $M$  คือ จำนวนหน่วยช่อง; และ  $K$  คือ จำนวนมิติของเอ้าต์พุตโครงข่าย. ฟังชัน nnOutput ทำการคำนวณสมการ 5.10, 5.11, 5.12, และ 5.13. ตาราง 5.3 สรุปฟังชันกระตุ้นของชั้นเอ้าต์พุตที่ขึ้นกับชนิดปัญหา. อาร์กูเมนต์ nntype ใช้ระบุชนิดปัญหา และมีค่าดีฟอลต์เป็นชนิดปัญหาการหากาถดถอย.

รายการ 6.1: ซิกมอยด์ฟังชัน (sigmoid), อนุพันธ์ของซิกมอยด์ฟังชัน (dsigmoid), และฟังชันคำนวณค่าของโครงข่าย (nnOutput)

---

```

1 sigmoid <- function (a){
2   return( 1 / (1 + exp(-a)) )
3 }##end sigmoid
4
5 dsigmoid <- function (z) {
6   return( (1 - z)*z )
7 }##end dsigmoid
8
9 nnOutput <- function (net, X, nntype='regression') {
10   X <- as.matrix(X)
11
12   ## Forward Pass
13   dotX <- rbind(1, X)
14   Z <- sigmoid( net$W1 %*% dotX )
15   dotZ <- rbind(1,Z)
16   N <- ncol(X)
17   K <- nrow(net$W2)
18   A <- net$W2 %*% dotZ;
19   if(nntype == 'regression'){
20     Y <- A
21   }##end if
22   if(nntype == 'biclass'){
23     Y <- 1/(1+exp(-A))
24   }##end if
25   if(nntype == 'multiclass'){
26     Y <- exp(A)/matrix(colSums(exp(A)), K, N, byrow=TRUE)
27 }##end if

```

```

28
29   return(Y)
30 }
```

สังเกตุ  $h'(a) = h(a) \cdot (1 - h(a))$  แต่ dsigmoid ใช้  $(1 - z)*z$  เพราะว่า dsigmoid รับอาร์กูเมนต์เป็น ตัวแปร  $z$  ซึ่งคือ  $h(a)$ . การทำเช่นนี้จะช่วยลดการคำนวนซ้ำซ้อนลงได้.

รายการ 6.2 แสดงโค้ดของฟังชันฝึกโครงข่าย nnTrain สำหรับโครงข่ายสองชั้น. ฟังชัน nnTrain รับอาร์กูเมนต์เป็น อินพุตของข้อมูลชุดฝึก X, เอ้าต์พุตของข้อมูลชุดฝึก T, จำนวนหน่วยชั้อน nHidden, อัตราเรียนรู้สำหรับน้ำหนักชั้นชั้อนและชั้นเอ้าต์พุต rhoh และ rhoo ตามลำดับ. นอกจากอาร์กูเมนต์ข้างต้น nnTrain ยังอนุญาติให้เลือกช่วงค่าของน้ำหนักตอนเริ่มต้นผ่านตัวแปร wmax, เลือกจำนวนรอบฝึกผ่านตัวแปร nEpoch, เลือกที่จะทำการวาดกราฟแสดงค่าผิดพลาดขณะฝึกหรือไม่ผ่านตัวแปร graph, และให้เลือกที่จะใส่โครงข่าย net เข้ามาเพื่อฝึกต่อได้. เพื่อให้แน่ใจว่าโค้ดของการแพร่กระจายย้อนกลับทำได้อย่างถูกต้อง, ควรเปรียบเทียบผลกับการคำนวนค่าเกรดเดียนต์แบบเชิงเลขด้วย (ดูแบบฝึกหัดข้อ 11)

รายการ 6.2: ฟังชันฝึกโครงข่าย (nnTrain)

```

1 nnTrain <- function (X,T,nHiddens,rhoh,rho0,
2   wmax=0.1,nEpochs=2000,graph=TRUE,net=NULL) {
3
4   D <- nrow(X)
5   N <- ncol(X)
6   K <- nrow(T)
7   M <- nHiddens
8
9   if (is.null(net)) {
10     ## Initialize weights
11     W1 <- matrix(runif(M*(1+D),-wmax,wmax),M,1+D)
12     W2 <- matrix(runif(K*(1+M),-wmax,wmax),K,1+M)
13
14     ## history of error
15     errors <- matrix(0,1,nEpochs)
16     firstEpoch <- 1 # this will be used in indexing of errors
17
18   } else {
19     W1 <- net$W1
20     W2 <- net$W2
21     errors <- matrix(c(net$errors, rep(0,nEpochs)),nrow=1)
22     firstEpoch <- length(net$errors)+1
23
24   }#if
25 }
```

```

26   dotX <- rbind(1,X)
27
28   for (epoch in 1:nEpochs){
29
30     ## (1) Forward propagation
31     ## Calculate hidden unit outputs, Z, which is M x N
32     Z <- sigmoid(W1 %*% dotX)
33
34     ## Calculate output unit outputs, Y, which is K x N.
35     dotZ <- rbind(1,Z)
36     Y <- W2 %*% dotZ
37
38     ## (2) Evaluate output delta
39     DELTA2 <- Y - T
40
41     ## (3) Backpropagate errors
42     S <- t(W2[,-1,drop=FALSE]) %*% DELTA2
43     DELTA1 <- dsigmoid(Z)*S
44
45     ## (4) Evaluate derivatives
46     dE2 <- DELTA2 %*% t(dotZ)
47     dE1 <- DELTA1 %*% t(dotX)
48
49     ## Update weights with Gradient Descent
50     W2 <- W2 - rhoo * dE2
51     W1 <- W1 - rhoh * dE1
52
53     errors[epoch+firstEpoch-1] <- sqrt(mean(DELTA2^2)) # RMSE
54
55     ne <- epoch + firstEpoch - 1
56
57     if (graph && ne > 9 && (ne %% round(ne/10) == 0)) {
58       plot(errors[1:(firstEpoch+epoch-1)],xlab="Epoch",ylab="RMSE",type="l",←
59             main="RMSE")
60     }
61   }#for
62
63   list(W1=W1, W2=W2, Z=Z, Y=Y, errors=errors)
64 }#nnTrain

```

---

สังเกตุภายใต้โค้ดว่า ถ้าไม่มีการใส่ `net` เป็นอาร์กูเมนต์ (เท่ากับ `net` เป็น `NULL`), โปรแกรมก็จะเริ่มสุ่มกำหนดค่าเริ่มต้นให้กับน้ำหนักชั้นที่ 1 และ 2 คือ  $W_1$  และ  $W_2$  ตามลำดับ. ตัวแปร `errors` มีไว้เพื่อเก็บประวัติค่าผิดพลาดระหว่างการฝึก.

หมายเหตุ โค้ดในรายการ 6.2 ใช้ค่าที่เก็บใน `errors` เป็น  $\text{sqrt}(\text{mean}(\text{DELTA2}^2))$ , ซึ่งคือ ค่า  $\sqrt{\frac{1}{N} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\|^2}$  (สมการ 5.19). เรียกว่า ค่าอาาร์เอมเอส (`rms`). ค่านี้ไม่ใช่ค่าฟังชันจุดประสงค์ ซึ่งคือ  $\frac{1}{2} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\|^2$  (สมการ 5.19). แต่ค่าอาาร์เอมเอสนี้จะเป็นสัดส่วนตามฟังชันจุดประสงค์ จึงสามารถใช้ตรวจสอบความก้าวหน้าของการฝึกได.

นอกจากนี้ โค้ดในรายการ 6.2 ทำการคำนวนไปข้างหน้า (forward pass) ด้วยฟังชัน `rate_tnn` สำหรับปัญหาการหาค่าผลตอบย. สำหรับปัญหานิดอื่น ฟังชัน `rate_tnn` ชั้นเอาร์พุตจะต้องถูกเปลี่ยนให้เหมาะสมตามที่อภิปรายในหัวข้อ 5.2.1.

เมื่อทำการเปรียบเทียบสมการ 5.37, 5.39 และ 5.36 กับโค้ด `nnTrain` จะเห็นว่า ฟังชัน `nnTrain` ทำการคำนวนการแพร์กระยะจ่อนกลับ และใช้วิธีลงเกรเดียนต์ในการฝึกโครงข่าย. สุดท้าย `nnTrain` ส่งผลลัพธ์ที่ได้ออกมาผ่านตัวแปรชนิด `list` (ที่มีส่วนสำคัญที่สุดคือ ค่าน้ำหนัก  $W_1$  กับ  $W_2$ ).

ฟังชันที่สำคัญอีกอย่างคือส่วนที่ใช้ทำนอร์มอลайเซชัน (ฟังชัน `normalize`) ซึ่งแสดงในรายการ 6.3. ฟังชันที่แสดงนี้ทำนอร์มอลайเซชันแบบปรับค่าค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน (หัวข้อ 6).

สังเกตุ `stdevs[stdevs==0] <- 1` ทำเพื่อป้องกันปัญหาเชิงคำนวน ที่เมื่ออินพุตบางมิติมีค่าเดียวตลอด (ทำให้ค่าเบี่ยงเบนมาตรฐานเป็น 0 และจะทำให้เกิดการหารด้วย 0 เกิดขึ้น). วิธีนี้เป็นทางแก้ชั่วคราว. วิธีที่ดีกว่าคือ ถ้าเกิดสถานะกรณ์ที่มีค่าอินพุตบางมิติมีค่าเดียวตลอด, การฝึกโครงข่ายอาจจะพิจารณาตัดมิตินั้นออกไปเลย เพราะการที่อินพุตมิตินั้นไม่เคยเปลี่ยนแปลงเลย อินพุตมิตินั้นจะไม่ผลช่วยให้ทำนายเอาร์พุตได้ดีขึ้นเลย แต่การมีมิติมากขึ้นทำให้โครงข่ายต้องทำการคำนวนมากขึ้น.

สังเกตุเพิ่มเติมอีกว่า ฟังชัน `normalize` อนุญาตให้เลือกได้ว่าจะให้ฟังชันทำการคำนวนค่าเฉลี่ย (`means`) กับ ค่าเบี่ยงเบนมาตรฐาน (`stdevs`) เอง หรือ อนุญาตให้ผู้ใช้สามารถระบุค่าเฉลี่ยกับค่าเบี่ยงเบนมาตรฐานเข้าเป็นอาร์กูเมนต์ได้. รวมทั้ง ผู้ใช้สามารถเลือกให้ฟังชัน นอกจากทำนอร์มอลайเซชันของค่าที่ต้องการแล้ว ยังให้มันบอกค่า `means` และ `stdevs` ที่ใช้ทำนอร์มอลайเซชันออกมาได้ด้วย. ผู้ใช้ต้องการค่า `means` และ `stdevs` ที่ใช้ในการทำนอร์มอลайเซชันนี้ เพราะ หากมีการทำนอร์มอลайเซชันกับอินพุตของข้อมูลฝึกหัด ด้วย `means` และ `stdevs` ค่าเท่าใด ผู้ใช้ก็ควรจะใช้ค่าเท่านั้นในการทำนอร์มอลайเซชันกับอินพุตของข้อมูลทดสอบหรือข้อมูลที่จะใช้งานด้วย.

### รายการ 6.3: ฟังชันทำนอร์มอลайเซชัน (`normalize`)

---

```

1 normalize <- function(X, means=apply(X, 1, mean), stdevs=apply(X, 1, sd),
2                               returnParams=FALSE) {
3   stdevs[stdevs==0] <- 1
4
5   D <- nrow(X)
6   N <- ncol(X)
7   X <- (X - matrix(rep(means, N), D, N)) /

```

```

8     matrix(rep(stdevs,N),D,N)
9 if (returnParms)
10   list(data=X,means=means,stdevs=stdevs)
11 else
12   X
13 }

```

---

ตัวอย่างง่ายๆ (หัวข้อ 6.1) เตรียมข้อมูลจากโค้ดที่แสดงในรายการ 6.4. สังเกตว่าเมื่อทำนอร์มอลайเซชัน ทำนอร์มอลายเซชันของอินพุตทั้งชุดฝึกและชุดทดสอบ และการทำนอร์มอลายเซชันของชุดทดสอบก็ใช้พารามิเตอร์ของนอร์มอลายเซชันค่าเดียวกับชุดฝึก (ดูตัวแปร r\$means และ r\$stdevs).

รายการ 6.4: เตรียมข้อมูลสำหรับตัวอย่างง่ายๆ

```

1 f <- function (x) { x + 8 * sin(x) + rnorm(length(x)) }
2
3 N <- 50
4 train.X <- matrix(seq(0,4*pi,len=N),1,N)
5 train.T <- f(train.X)
6 r <- normalize(train.X,returnParms=TRUE)
7 train.Xn <- r$data
8
9 test.X <- matrix(seq(0,4*pi,len=round(N/3)),nrow=1)
10 test.T <- f(test.X)
11 test.Xn <- normalize(test.X,r$means,r$stdevs)

```

---

ฟังชัน nnTrain สร้างและฝึกโครงข่ายประสาทเทียม เช่น

```

net <- nnTrain(train.Xn,train.T, nHiddens=20,
                rho_h=0.01, rho_o=0.0003, wmax=0.5, nEpochs=9001)

```

ซึ่งจะสร้างโครงข่ายประสาทเทียมสองชั้นขนาด 20 หน่วยช่อน และกำหนดค่าเริ่มต้นของค่าน้ำหนักโดยสุ่มจากช่วง  $[-0.5, 0.5]$  ทำการฝึก 9001 รอบกับข้อมูลที่มีอินพุตเป็น train.Xn และเอาต์พุตเป็น train.T ฝึกด้วยค่าอัตราการเรียนรู้  $\rho_h = 0.01$  และ  $\rho_o = 0.0003$  สำหรับชั้นช่อนและชั้นเอาต์พุตตามลำดับ.

เมื่อฝึกโครงข่ายประสาทเทียมเสร็จ โครงข่ายที่ฝึกแล้วถูกทดสอบโดยการนำไปทำนายผลอินพุตของข้อมูลชุดทดสอบ ดังนี้

```
test.y <- nnOutput(net,test.Xn)
```

และค่าที่ทำนายของชุดทดสอบถูกนำมาเปรียบเทียบกับเอาต์พุตจริง (เฉลยของข้อมูลชุดทดสอบ),

```
test.rmse <- sqrt(mean((test.y - test.T)^2))
```

ซึ่งตัวอย่างนี้ได้ค่าผิดพลาดเป็น 1.54 ดังแสดงในรูป 6.4. โค้ดข้างต้นไม่จำเป็นต้องระบุ `nntype='regression'` เพราะการหาค่าถดถอยเป็นค่าดีฟอลต์สำหรับฟังชัน `nnOutput` อยู่แล้ว.

สำหรับการทำการหดก่อนกำหนด การเพิ่มการทำการหดก่อนกำหนดก็เพียงแค่ตัดแปลงโค้ดในรายการ 6.2 โดยเพิ่มอาร์กูเมนต์เข้าไปตอนนิยามฟังชัน เช่น

```
nnTrain <- function (X,T,nHiddens,rhoh,rho0,wmax=0.1,nEpochs,
graph=TRUE,net=NULL, earlystopping=FALSE, early.tol=0.1,
val.X=NULL, val.T=NULL)
```

โดย 2 บรรทัดท้ายเพิ่มทางเลือก `earlystopping` เพื่อเป็นตัวแปรกำหนดว่าจะทำหดก่อนกำหนดหรือไม่ (ค่าดีฟอลต์คือไม่ทำ). นอกจากอินเตอร์เฟซของฟังชัน ภายในตัวฟังชัน (function body, ก่อนบรรทัดที่ 61 ของรายการ 6.2) ก็เพิ่มโค้ด ดังนี้

```
if (earlystopping) {
  val.Y <- nnOutput(list(W1=W1, W2=W2), val.X)
  val.E <- sum( (val.Y - val.T)^2 )

  if (val.E < best.val.E) {
    best.net <- list(W1=W1, W2=W2, Z=Z, Y=Y, errors=errors)
    best.val.E <- val.E
  }

} else if( val.E > (1 + early.tol) * best.val.E ) break;

}##end if
```

สังเกตว่า โค้ดข้างต้นยอมให้ค่าผิดพลาดของชุดвалиเดชั้นเพิ่มขึ้นได้บ้าง ตัวอย่างเช่น `early.tol=0.1` คือ ยอมให้เพิ่มขึ้นได้ไม่เกิน 10%. หรือ หากต้องการ อาจใช้จำนวนครั้งที่ยอมให้ค่าผิดพลาดของชุดвалиเดชั้นเพิ่มขึ้น เป็นเงื่อนไขของการหดการฝึกก็ได้ หรือ แม้แต่อาจใช้เงื่อนไขหลายๆอย่างผสมกันได้ ดังเช่น การทำงานของชุดเครื่องมือโครงข่ายประสาทเทียมของโปรแกรมแมทแลบ (Matlab's neural network toolbox) ซึ่งเป็นชุดซอฟต์แวร์สำหรับโครงข่ายประสาทเทียมของบริษัทแมธเวิร์ค (MathWorks) ที่มีการใช้เงื่อนไขผสม เพื่ออนุญาติให้ผู้ใช้เลือกการทำหดก่อนกำหนดให้เหมาะสมกับปัญหาที่ทำได้.

## 6.5.2 โค้ดสำหรับตัวอย่างการหาค่าถดถอย (ข้อมูลชุดเรือยอชต์)

ข้อมูล `yacht_hydrodynamics.data` อยู่ในรูปแบบแฟ้มข้อความ (textfile) ที่แต่ละระเบียบแสดงเป็นบรรทัด. ข้อมูลนี้สามารถถูกนำเข้าได้ดังนี้

```
y.dat <- read.csv("yacht_hydrodynamics.data", sep = " ")
```

ซึ่งอาร์กูเมนต์ `sep = " "` ระบุว่าแต่ละเขตข้อมูลใน `yacht_hydrodynamics.data` จะคั่นด้วยช่องว่าง (สังเกตุเนื้อหาในไฟล์ `yacht_hydrodynamics.data`<sup>3</sup>).

เมื่อนำเข้าข้อมูลเรียบร้อย ข้อมูลส่วนหนึ่งควรจะถูกกันไว้เพื่อการตรวจสอบประสิทธิผลของการทำนายโดยตัวอย่างเลือกใช้สัดส่วน 70 : 30 สำหรับข้อมูลสำหรับนำไปใช้สร้างโมเดลและทดสอบตามลำดับ. โดยที่ต้องมีข้อความสำคัญคือการแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกหัด,

```
N <- nrow(y.dat)
N.train <- round(0.7 * N)
train.ids <- sample(N, N.train)

train.X <- t(y.dat[train.ids,1:6])
train.T <- t(y.dat[train.ids,7])
```

ตัวอย่างนี้เลือกข้อมูลออกมา 215 จุด (ราวๆ 70% ของทั้งหมด) มาเป็นชุดฝึกหัด. สังเกตุ การแบ่งข้อมูลออกมานี้จะสุ่มออกมา. ไม่ใช่การแยกโดยเรียงตามลำดับ. ฟังชัน `sample(N, N.train)` จะให้ค่าสุ่มออกมานะ `N.train` ค่า โดยแต่ละค่าจะสุ่มจาก  $\{1, 2, 3, \dots, N\}$  โดยไม่มีการเลือกซ้ำ. นอกจากนั้น การสุ่มนี้เป็นการสุ่มเลือกจุดข้อมูล ดังนั้นค่าที่สุ่มคือค่าของดัชนีของจุดข้อมูล ไม่ใช่สุ่มค่าอินพุตหรือเอาต์พุตโดยตรง. ตัวแปร `train.ids` คือ ดัชนีที่เลือกมาสำหรับการฝึกโครงข่ายประสาทเทียม (การสร้างโมเดล). อินพุตและเอาต์พุตของชุดฝึกหัด คือ `train.X` และ `train.T` ตามลำดับ.

ข้อมูลส่วนที่เหลือจะใช้เพื่อทดสอบโมเดลที่ได้ เรียกว่าชุดทดสอบ

```
test.X <- t(y.dat[-train.ids,1:6])
test.T <- t(y.dat[-train.ids,7])
```

ข้อมูลในชุดทดสอบเป็นจุดข้อมูลที่ไม่อยู่ในชุดฝึกหัด. ในอาร์เพรเจค การเลือกดัชนีเป็นลบคือการเลือกที่จะไม่เอาข้อมูลของดัชนีเหล่านั้น. แทนที่จะใช้ดัชนีเป็นลบ อาจฟังชันเพื่อหาสมาชิกที่ต่างกันของสองเซตได้นั่นคือ `setdiff(1:N, train.ids)` ก็จะให้ดัชนีที่ไม่อยู่ในชุดฝึกหัดออกจากได้เช่นกัน.

สำหรับข้อมูลชุดนี้ ช่วงค่าของแต่ละมิติต่างกันมาก, ซึ่งสามารถตรวจสอบได้ง่ายๆ จาก `summary(t(train.X))`. หมายเหตุ ผลของการรัน `summary(t(train.X))` ไม่ได้แสดง ณ ที่นี่.

การที่ช่วงค่าของแต่ละมิติต่างกันมาก จะทำให้การฝึกโครงข่ายให้ได้ผลดีทำได้ยาก (ดังอภิปรายในหัวข้อ 6) ดังนั้น จึงควรที่จะนำร่วมกันไว้ เช่น,

<sup>3</sup>บางครั้ง อาจ ต้อง ทำการ จัดการ ไฟล์ ข้อมูล ก่อน ที่ จะ สามารถ นำ ข้อมูล เข้า มา ได้ อย่าง ถูก ต้อง เช่น ไฟล์ ข้อมูล `yacht_hydrodynamics.data` ที่ใช้ช่องว่างสองช่องติดกัน แทนที่จะใช้แค่ช่องเดียว. เพื่อให้ฟังชัน `read.csv` โหลดข้อมูลได้ถูกต้อง (ผลลัพธ์มีขนาด  $307 \times 7$ ) อาจทำการจัดการแก้ไขให้ช่องว่างสองช่องที่ติดกัน โดยแก้ให้เป็นช่องเดียวกัน ด้วยฟังชัน `Replace` ซึ่งมีในโปรแกรมจัดการไฟล์ข้อมูลทั่วไป.

```
r <- normalize(train.X, returnParms=TRUE)
train.Xn <- r$data
test.Xn <- normalize(test.X, means=r$means, stdevs=r$stdevs)
```

สังเกตุการทำอ้อมอย่างเช่นสำหรับอินพุตของชุดฝึกหัด. อาร์กูเมนต์ `returnParms=TRUE` จะระบุให้ฟังชัน `normalize` ให้ค่าพารามิเตอร์ที่ใช้ทำอ้อมอย่างเช่น (ได้แก่ `r$means` และ `r$stdevs`) ออกมากด้วย (นอกจากค่าที่ทำอ้อมอย่างเช่นอกมา `r$data`). ในการทำอ้อมอย่างเช่น ควรทำอ้อมอย่างเช่นกับอินพุตทุกชุดข้อมูล (ไม่ว่าอินพุตของชุดฝึกหัดหรือชุดทดสอบหรือการใช้งานจริง) และทำด้วยค่าพารามิเตอร์เดียวกัน. สังเกตว่าอินพุตมี 6 มิติ ค่าพารามิเตอร์ `r$means` และ `r$stdevs` ก็มี 6 ชุด สำหรับแต่ละมิติ. หมายเหตุ ค่าของตัวแปร `r$means` และ `r$stdevs` ที่เก็บค่าพารามิเตอร์ของการอ้อมอย่างเช่นไม่ได้แสดง ณ ที่นี่.

ตัวอย่างนี้เลือกใช้โครงข่ายสองชั้นขนาด 10 หน่วยช่อง ใช้อัตราเรียนรู้เป็น  $\rho_h = 0.001$  และ  $\rho_o = 0.0003$  และ ฝึก 10000 รอบ, ดังโค้ด

```
net <- nnTrain(train.Xn, train.T, nHiddens=10, rhoH=0.001,
rhoO=0.0003, wmax=0.5, nEpochs=10000)
```

ซึ่งเมื่อรันเสร็จ ผลลัพธ์ซึ่งคือค่าน้ำหนักที่ได้จะอยู่ที่ตัวแปร `net$W1` และ `net$W2`.

โครงข่ายที่ผ่านการฝึกดีแล้วสามารถนำไปใช้งานได้ และควรจะทดสอบผลการทำงาน ซึ่งสามารถทดสอบด้วยข้อมูลชุดทดสอบ ดังนี้

```
test.y <- nnOutput(net, test.Xn)
```

ผลการทำนายเก็บไว้ในตัวแปร `test.y`. ค่าความผิดพลาดสามารถตรวจดูได้จาก

```
test.rmse <- sqrt(mean((test.y - test.T)^2))
```

ภาพชี้อย่างรูป 6.6 แสดงตัวอย่างการนำเสนอผล. เนื่องจากข้อมูลชุดนี้อินพุตมีหลายมิติ การแสดงความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตทำได้ยาก. ตัวอย่างนี้แค่ต้องการประเมินผลว่า เอาต์พุตจากโครงข่ายแตกต่างจากค่าจริงเท่าใด ดังนั้นจึงอาจใช้แพล็อกแสดงความสัมพันธ์ระหว่างเอาต์พุตจากโครงข่ายและค่าจริง

```
plot(test.T, test.y)
```

ซึ่งภาพด้านขวาของรูป 6.6 เพียงเพิ่มเส้นตรง  $y = x$  เข้าไปเพื่อความสะดวกในการอ่าน จุดที่หัวเส้น (`test.y = test.T`) แสดงถึงการทำนายได้ถูกต้องแม่นยำ จุดที่หัวเส้นมากเท่าไรแสดงถึงค่าผิดพลาดที่มีมากเท่านั้น.

### 6.5.3 โค้ดสำหรับตัวอย่างการจำแนกประเภท (ข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ)

ข้อมูลชุดภาพเอ็กซเรย์เต้านมของมวลเนื้อ (Mammographic Mass Dataset) ที่ดาวน์โหลดมา สามารถนำเข้าตัวแปรได้โดยคำสั่ง

```
mammo <- read.csv("mammographic_masses.data",
sep = ",", header = FALSE)
```

เมื่อนำข้อมูลเข้ามาอยู่ในตัวแปร `mammo` เสร็จแล้ว ก็ควรตรวจสอบขนาดของข้อมูลในตัวแปร (`dim(mammo)` ว่าได้  $961 \times 6$ ) และค่าที่โหลดเข้ามาว่าถูกต้อง.

ค่าที่โหลดเข้ามาจะมีค่า ‘?’ อยู่ ซึ่ง ‘?’ หมายถึง ข้อมูล ณ ตำแหน่งนั้นไม่มี (missing data). เพื่อความสะดวกสำหรับการจัดการด้วยอาร์เพรูเจค<sup>4</sup>, ตัวอย่างนี้ทำการเปลี่ยน ‘?’ ให้เป็น NA ก่อนที่จะแปลงทั้งตัวแปรให้เป็นชนิด `numeric`,

```
mam1 <- mammo
mam1[mam1 == '?'] <- NA
mam2 <- apply(mam1, 2, as.numeric)
rownames(mam2) <- rownames(mam1)
```

คำสั่ง `mam1[mam1 == '?'] <- NA` จะเลือกทุกๆ ค่าที่เป็น ‘?’ และแทนด้วย NA. ส่วน `mam2 <- apply(mam1, 2, as.numeric)` จะช่วยทำให้ตัวแปร `mam2` เป็นเมตริกซ์. และ `rownames(mam2) <- rownames(mam1)` แค่ทำเพื่อความสะดวกที่จะนำไปใช้ (เป็นตัวเลข).

#### 6.5.3.1 การจัดการกับเขตข้อมูลขาดหาย

บางเขตข้อมูลที่ไม่มีค่า (missing data)ควรต้องถูกจัดการก่อนนำข้อมูลเข้าไปฝึกสร้างโมเดลเพื่อทำนายผล. ดังที่ได้กล่าวไปในหัวข้อ 6.3, วิธีหนึ่งในการจัดการกับค่าบางเขตที่ขาดข้อมูลคือ การตัดทุกๆ ระเบียนที่มีค่าบางเขตที่ขาดข้อมูลออกไปเลย โดยสำหรับอาร์เพรูเจคทำได้ง่ายๆ ดังนี้

```
mam3 <- na.omit(mam2)
```

ตัวแปร `mam3` จะเป็นเมตริกซ์ที่ไม่มีค่าบางเขตที่ขาดไป (ขนาดจะเหลือ  $830 \times 6$ ).

แต่ตัวอย่างนี้เลือกใช้วิธีแทนทุกค่าที่เป็นไปได้ (assigning all possible values of the attribute). รายการ 6.5 แสดงโค้ดการจัดการกับค่าบางเขตข้อมูลขาดไปด้วยวิธีแทนทุกค่าที่เป็นไปได้. สำหรับแต่ละ

<sup>4</sup>การมี ‘?’ ซึ่งเป็นชนิด `text` จะบังคับให้อาร์เพรูเจคกำหนดตัวแปร `mammo` เป็นตัวแปรชนิด `dataframe`. ตัวแปรชนิด `dataframe` มีข้อดีคือสามารถรับชนิดข้อมูลได้หลากหลายทั้ง `text` ทั้ง `numeric` แต่ข้อเสียคือไม่สามารถทำงานกับตัวปฏิบัติการของเมตริกซ์ได้. การเปลี่ยน ‘?’ ให้เป็น NA ทำได้ไม่ยาก และ NA สามารถเปลี่ยนเป็นชนิด `numeric` ได้.

เขตข้อมูล (1 ถึง 6 เขตข้อมูล) ถ้าในเขตข้อมูลนั้นมี ข้อมูลขาด (มี NA, ตรวจได้โดยคำสั่ง `is.na`),

(1) ให้ระบุตัวชี้ของระเบียนที่เขตข้อมูลนั้นขาดไป

ทำโดย `missing.ids <- which(is.na(mam2[,i]))`

(2) ให้หาทุกค่าที่เป็นไปได้ของเขตข้อมูลนั้น ไม่รวมค่า NA

ทำโดย `unique.vals <- setdiff(unique(mam2[,i]), NA)`

(3) แต่ละระเบียนที่มีเขตข้อมูลขาด, ให้เพิ่มระเบียนที่แทนค่าที่ขาดด้วยค่าที่เป็นไปได้อื่นเข้าไป (ภายใน `for` loop ของตัวแปร `j`),

(4) หลังจากเพิ่มระเบียนใหม่เข้าไปครับแล้ว ลบระเบียนเก่าที่เขตข้อมูลขาดออกไป.

สังเกตุ การลบระเบียนเก่าที่เขตข้อมูลขาดออกไปจะทำทีหลัง (บรรทัด 21, รายการ 6.5). หลังจากซ่อมข้อมูลเสร็จ ควรตรวจสอบข้อมูลที่ซ่อมใหม่ เพื่อให้แน่ใจว่าการซ่อมทำได้อย่างถูกต้อง ข้อมูลที่ซ่อมแล้วจะอยู่ในตัวแปร `mam2` ซึ่งตอนนี้มีขนาด  $2308 \times 6$ .

รายการ 6.5: ตัวอย่างการจัดการกับค่าบางเขตข้อมูลขาดไป ด้วยวิธีแทนทุกค่าที่เป็นไปได้

---

```

1 for(i in 1:6){ ## go through each field
2
3   if( sum(is.na(mam2[,i])) > 0 ){
4
5     ## Identify missing records
6     missing.ids <- which(is.na(mam2[,i]))
7     N.miss <- length(missing.ids)
8
9     ## Find possible attribute values
10    unique.vals <- setdiff(unique(mam2[,i]), NA)
11    N.vals <- length(unique.vals)
12
13    for(j in 1:N.miss){
14      new.rec <- matrix(mam2[missing.ids[j],], N.vals, 6, byrow=T)
15      for(k in 1:N.vals){
16        new.rec[k,i] <- unique.vals[k]
17      }
18      mam2 <- rbind(mam2, new.rec)
19    }##end for j
20
21    mam2 <- mam2[-missing.ids, ]
22  }##end if
23 }##end for i

```

---

### 6.5.3.2 การจัดเตรียมข้อมูล

หลังจากจัดการกับเขตข้อมูลขาดหายเรียบร้อย การจัดเตรียมข้อมูลก็สามารถทำได้ในลักษณะเดิม. นั่นคือ การแบ่งข้อมูลเป็นชุดฝึกกับชุดทดสอบ และทำนอร์มอลайเซ่น, ดังแสดงในรายการ 6.6. ตัวอย่างนี้แบ่งข้อมูลราวๆ 80% ของจุดข้อมูลสำหรับการฝึก และที่เหลือ(ราวๆ 20%) สำหรับการทดสอบทั้งหมด. สังเกตุ (1) การแบ่งข้อมูลจะใช้การสุ่ม, ไม่ใช่เรียงลำดับ และ (2) การทำนอร์มอลайเซ่น จะใช้ค่าพารามิเตอร์ชุดเดียวกัน (ดูเปรียบเทียบ การกำหนดค่า `set1.Xn` เทียบกับ `set2.Xn`). ค่านอร์มอลайเซ่นพารามิเตอร์ `r$means` และ `r$stdevs` จะต้องเก็บไว้ เพราะ ทุกครั้งที่ใช้โครงข่ายประเทียม ต้องใช้ค่าเหล่านี้ในการทำนอร์มอลайเซ่น.

รายการ 6.6: ตัวอย่างการจัดเตรียมข้อมูล Mammographic Mass

---

```

1 N ← nrow(mam2)
2 ids ← sample(N)
3
4 N1 ← round(N*0.8)    ## number of datapoints for X-val/Train
5 N2 ← N - N1           ## number of datapoints for final testing
6
7 set1.X ← t(mam2[ids[1:N1],1:5])
8 set1.T ← t(mam2[ids[1:N1],6])
9 r ← normalize(set1.X, returnParms=TRUE)
10 set1.Xn ← r$data
11
12 set2.X ← t(mam2[ids[-1:-N1],1:5])
13 set2.T ← t(mam2[ids[-1:-N1],6])
14 set2.Xn ← normalize(set2.X, means=r$means, stdevs=r$stdevs)
```

---

### 6.5.3.3 การแบ่งข้อมูลเพื่อทำครอสвалиเดชั่น

ตัวอย่างนี้เลือกทำครอสвалиเดชั่นแบบ 5 พับ (5-fold cross validation). การทำครอสвалиเดชั่นทำเพื่อเลือกโมเดล ซึ่งในที่นี้คือค่าไชเบอร์พารามิเตอร์ ( $M$ ,  $\rho_1$ , และ  $\rho_2$ ). ดังนั้นส่วนของข้อมูลสำหรับฝึกหัด (`set1.Xn` กับ `set1.T`) จะแบ่งมาเพื่อทำครอสвалиเดชั่น.

หมายเหตุ ตัวอย่างนี้แยกข้อมูลส่วนทดสอบสุดท้ายออกอย่างชัดเจน หากมีข้อมูลเพียงพอ ควรแยกส่วนทดสอบสุดท้ายออกจากกระบวนการสร้างโมเดล ดังเช่นที่แสดงในตัวอย่างนี้. อย่างไรก็ตาม บิชอป[9]อภิปรายว่า การทำครอสвалиเดชั่นนั้น เมื่อเลือกความซับซ้อน หรือค่าไชเบอร์พารามิเตอร์ของโมเดลได้แล้ว ก็สามารถนำข้อมูลทั้งหมดไปใช้ในกระบวนการฝึกโมเดลได้เลย และค่าซึ่งวัดคุณภาพการทำนายของโมเดล ก็ได้จากการทำครอสвалиเดชั่นทุกพับ.

รายการ 6.7 แสดงโดยการแบ่งพับสำหรับครอสвалиเดชั่น. เนื่องจากจำนวนระเบียนของข้อมูลส่วนฝึกหัดนี้มี 1846 ระเบียน จึงแบ่งให้มี 1 พับที่มี 370 ระเบียน ที่เหลืออีก 4 พับมี 369 ระเบียน. มันไม่

สำคัญว่าพับที่มี 370 ระเบียนเป็นพับไหน เพราะการแบ่งพับทำด้วยการสุ่ม และสุดท้ายผลของครอสวอลิเดชันได้จากการนำผลของทุกพับมาเฉลี่ยกัน. สังเกตุการแบ่งพับนี้ไม่ได้แบ่งที่ข้อมูลจริง เพียงใช้การจัดแบ่งด้วยที่จะใช้อ้างถึงข้อมูลไว้กับพับต่างๆเท่านั้น. หมายเหตุ ตัวอย่างในรายการ 6.7 ทำการสุ่มด้วย `id1s <- sample(N1)` ซึ่ง ณ ที่นี่ไม่มีความจำเป็นต้องทำ เพราะข้อมูลใน `set1.Xn` และ `set1.T` นั้นถูกสุ่มมาแล้ว แต่การสุ่มอีกทีนึงไม่ได้เสียหายอะไร.

รายการ 6.7: ตัวอย่างโค้ดการแบ่งพับสำหรับครอสวอลิเดชัน

---

```

1 fold.Ns <- c(370, 369, 369, 369, 369)
2 fold.ids <- list()
3
4 id1s <- sample(N1)
5
6 id.marks <- c(0, cumsum(fold.Ns))
7 for(i in 1:5){
8   fold.ids[[i]] <- id1s[ (id.marks[i]+1):id.marks[i+1] ]
9 }
```

---

#### 6.5.3.4 โค้ดของการแบ่งกลุ่มเปรียบเทียบกับโค้ดการหาค่าถดถอย

ตัวอย่างนี้ใช้โครงข่ายประสาทเทียมกับงานการแบ่งกลุ่ม, โค้ดที่ใช้จะคล้ายกันมากกับการหาค่าถดถอย ต่างกันแค่ฟังชันกระตุ้นของชั้นเออต์พุต ดังที่ยกในบท 5.

โค้ดโครงข่ายประสาทเทียมสำหรับการหาค่าถดถอยในรายการ 6.2 สามารถถูกปรับให้เหมาะสมกับการแบ่งกลุ่ม (ดูตาราง 5.3 ประกอบ) โดยการแก้ฟังชันกระตุ้นของชั้นเออต์พุตให้เป็น sigmoid ได้แก่ การแก้บรรทัดที่ 36 ของ `nnTrain` ในรายการ 6.2 จาก `Y <- W2 %*% dotZ` เป็น

```

A <- W2 %*% dotZ
Y <- 1/(1 + exp(-A))
```

เพียงเท่านี้ โค้ดที่ได้ก็ทำโครงข่ายประสาทเทียมสำหรับงานแบ่งกลุ่มได้แล้ว. หมายเหตุ ฟังชัน `nnOutput` (รายการ 6.1) เตรียมเออต์พุตสำหรับงานจำแนกกลุ่มไว้แล้ว โดยเพียงเลือกใช้ `nntype = 'biclass'`.

เพื่อความสะดวก ตัวอย่างนี้จะใช้ฟังชันจำกัดแข็ง `hardlimit` (รายการ 6.8) สำหรับช่วยปรับเออต์พุตจากโมเดลให้อยู่ใน  $\{0, 1\}$ . ฟังชันจำกัดแข็ง `hardlimit` จะให้ค่า 1 หากค่าที่ได้มากกว่าค่าระดับตัดแบ่งเขต `cutoff`, ไม่อย่างนั้น จะให้เป็น 0. ค่าดีฟอลต์ของ `cutoff` ตั้งไว้ที่ 0.5.

รายการ 6.8: ฟังชันจำกัดแข็ง `hardlimit` เพื่อช่วยจัดการเออต์พุตสุดท้ายสำหรับการแบ่งกลุ่ม

---

```

1 hardlimit <- function(x, cutoff=0.5){
2   y <- apply(x > cutoff, c(1,2), as.numeric)
3 }
```

---

นอกจากนั้น หากต้องการ อาจเปลี่ยนการคำนวณค่าผิดพลาดให้เข้ากับงานแบ่งกลุ่ม (สมการ 5.30 ตามที่กล่าวในหัวข้อ 5.3) ได้แก่ การแก้บรรทัดที่ 53 ของ nnTrain ในรายการ 6.2 เป็น

```
errors[epoch+firstEpoch-1] <- sum( -(T*log(Y) + (1-T)*log(1-Y)) )
```

หมายเหตุ การเปลี่ยนการคำนวณ errors นี้ ไม่มีผลต่อผลการฝึกของโครงข่ายประสาทเทียม เพียงแต่จะช่วยให้ค่าผิดพลาดที่แสดงตรงกับทฤษฎีเท่านั้น. ที่การเปลี่ยนการคำนวณ errors นี้ ไม่มีผลต่อผลการฝึกก็ เพราะว่าค่า errors (บรรทัดที่ 53 รายการ 6.2) ใช้แค่เพื่อแสดงความก้าวหน้าของผลการฝึกเท่านั้น. ส่วนการฝึกซึ่งใช้วิธีลงเกรเดียนต์นั้น ใช้แต่ค่าเกรเดียนต์  $\frac{\partial E_n}{\partial a_k}$  ซึ่งยังคงเป็น  $y_k - t_k$  และทำงานกับสมการ 5.30 โดยอัตโนมัติ ดูแบบฝึกหัดบทที่ 5 ข้อ 2 ประกอบ.

### 6.5.3.5 การทำครอสвалиเดชั่นเพื่อเลือกไฮเปอร์พารามิเตอร์

รายการ 6.9 แสดงโค้ดการทำครอสвалиเดชั่น เพื่อเลือกค่าไฮเปอร์พารามิเตอร์ที่เหมาะสม. สังเกตุในทุกๆ ชุดของค่าไฮเปอร์พารามิเตอร์ มีการฝึกและทดสอบ 5 ครั้งสำหรับвалиเดชั่นห้าพับ (บรรทัด 24 ถึง 47). แต่ละครั้งของการทำвалиเดชั่น มี 1 พับที่ถูกเก็บไว้สำหรับการวัดผล (บรรทัด 28, 29, 31-33). โค้ดบรรทัด 28 และ 29 จะเลือกข้อมูลทุกรอบเป็น ยกเว้นระเบียนของดัชนีที่อยู่ใน fold.ids[[i]] ซึ่งเป็นตัวแปรเก็บดัชนีต่างๆ ของพับที่ถูกเลือกไว้สำหรับทดสอบวาลิเดชั่น. ในอาร์โปรดัก การใช้ค่าดัชนีเป็นลบหมายถึง การไม่เอาข้อมูลของดัชนีนั้น. และข้อมูลของดัชนีที่ไม่นำไปฝึกโดยจะนำไปใช้ทำвалиเดชั่น ดังโค้ดบรรทัด 31-33. ผลการทำвалиเดชั่นจะถูกบันทึกในตัวแปร records, โดยตัวแปร records จะเก็บทั้งค่าไฮเปอร์พารามิเตอร์ต่างๆ ที่ต้องการตรวจสอบ ได้แก่ m, rho1, rho2, Nepoch รวมถึง ดัชนีวัลิเดชั่น i และ ผลการทดสอบวาลิเดชั่น (ตัวแปร accuracy).

ตัวอย่างโค้ดในรายการ 6.9 นี้อาจใช้เวลา\_ranถึงเกือบครึ่งชั่วโมง ขึ้นกับคอมพิวเตอร์ที่ใช้รัน. การเพิ่มโค้ดเพื่อรายงานสถานะการทำงาน (เช่น บรรทัด 26 และ 44) จะช่วยลดความกังวลได้บ้าง เวลาที่โค้ดที่ใช้เวลานานในการรัน.

รายการ 6.9: ตัวอย่างโค้ดการทำครอสвалиเดชั่นห้าพับ

---

```

1 Ms <- c(5, 10, 30)
2
3 ## triplet of rhoh, rho0, and # epochs
4 rho.Ns <- list(c(0.01, 0.001, 5000),
5                 c(0.01, 0.01, 5000),
6                 c(0.001, 0.01, 5000))
7
8 nets <- list()
9
10 records <- matrix(0, 45, 6) ## for m, rhoh, rho0, Nepochs, fold, error
11
```

```

12 i.rec <- 1
13
14 cat('Start at \n')
15 print(Sys.time())
16
17 for(m in Ms){
18   for(j in 1:3){
19
20     rho1=rho.Ns[[j]][1]
21     rho2=rho.Ns[[j]][2]
22     Nepoch=rho.Ns[[j]][3]
23
24     for( i in 1:5 ){
25
26       cat('** Running :', i.rec, ' of 45\n')
27
28       train.Xn <- set1.Xn[,-fold.ids[[i]]]
29       train.T <- set1.T[,-fold.ids[[i]],drop=F]
30
31       xval.Xn <- set1.Xn[,fold.ids[[i]]]
32       xval.T <- set1.T[,fold.ids[[i]]],drop=F]
33       xval.N <- ncol(xval.T)
34
35       net <- nnTrain(train.Xn, train.T,
36         nHiddens=m, rhoh=rho1, rho0=rho2, wmax=0.5,
37         nEpochs=Nepoch, plottitle=paste('Cost at run ', i.rec))
38
39       nets[[i.rec]] <- net
40       xval.y <- hardlimit(nnOutput(net, xval.Xn, nntype='biclass'))
41       accuracy <- sum(xval.y == xval.T)/xval.N
42
43       records[i.rec,] <- c(m, rho1, rho2, Nepoch, i, accuracy)
44       cat('Done:', i.rec, ' correct = ', round(accuracy,2), '\n')
45
46       i.rec <- i.rec + 1
47     }##end for i
48   }##end j
49 }##end for m

```

หมายเหตุ ค่าเริ่มต้นของน้ำหนักก็มีผลต่อโมเดล และในการฝึกโครงข่ายประสาทเทียมค่าเริ่มต้นจะถูกสุมเข็น. ดังนั้น เพื่อเพิ่มความมั่นใจว่า ผลที่ได้จากการทำครอสวอลิดेचันแต่ละครั้งไม่ได้บังเอิญมาจากการค่า

เริ่มต้นที่สู่มไปเจอค่าดีหรือไม่ดีเป็นพิเศษ ผู้ฝึกอาจทำครอสวอลิดेचั่นหลายครั้งแล้วเลือกผลที่ดีที่สุดออกมา. นั่นคือ ผู้ฝึกอาจแก้ไขเพิ่มลูปที่ครอบลูป `for( i in 1:5 )` เพื่อทำครอสวอลิดเดชั่นช้าๆ หลายครั้ง โดยที่ให้มีการกำหนดค่าเริ่มต้นใหม่ สำหรับทุกๆ ครั้งของการซ้ำทำครอสวอลิดเดชั่น(ทุกพับของครอสวอลิดเดชั่นให้ค่าเริ่มต้นเดียวกัน). และใช้ผลที่ดีที่สุดของทุกช้าเป็นตัวแทนของครอสวอลิดเดชั่นนั้นๆ<sup>5</sup>. การทำซ้ำแล้วเลือกตัวแทนที่คาดว่ามาจากกรณีที่ดีที่สุดก็จะช่วยให้น่าเชื่อถือมากขึ้นว่า ผลที่ได้สรุปมาจากการข่ายประสาทเทียมที่ปรับได้ดีที่สุดแล้ว. หมายเหตุ การเลือกตัวแทนของกลุ่มว่าจะเลือกจากค่าเฉลี่ย (average case), ค่าที่ดีที่สุด (best case), หรือค่าที่แย่ที่สุด (worst case) นั้นขึ้นกับคำถามที่ต้องการตอบ เช่น กรณีนี้คือหากต้องการหาค่าไฮเปอร์พารามิเตอร์ที่ทำงานได้ดีที่สุด โดยสมมติฐานคือ เมื่อใช้ค่าไฮเปอร์พารามิเตอร์ที่กำหนดแล้ว ค่าน้ำหนักสามารถฝึกได้ค่าดีที่สุด ดังนี้แล้วตัวแทนกลุ่มนึงควรเลือกใช้ค่าที่ดีที่สุด. แต่กรณีของการเลือกตัวแทนจาก 5 พับของครอสวอลิดเดชั่น ต้องการผลจากทุกพับ ไม่ใช่เฉพาะพับที่ดีที่สุด ซึ่งผลที่ดีอาจมาจากความบังเอิญที่ได้ชุดทดสอบที่ยาก แต่ต้องการผลจากทุกพับที่มีทั้งดีทั้งแย่ทั้งกลางๆ จึงใช้ค่าเฉลี่ยมาเป็นตัวแทนของผลจาก 5 พับ.

**การตรวจตอบผลและการเลือกโมเดล** สิ่งสำคัญคือ ผู้ฝึกโครงข่ายควรตรวจสอบว่าการฝึกเป็นไปด้วยความเรียบร้อย เช่น ตรวจดูค่าผิดพลาดเทียบกับรอบการฝึก ในแต่ละการฝึก. โค้ดข้างล่างนี้แสดงกราฟกราฟค่าผิดพลาดเทียบกับรอบการฝึก ของการฝึกทั้ง 45 ครั้งออกแบบ

```
par(mfrow=c(5,9))
par(mar=c(2,2,2,1))
for(i in 1:45){
  plot(1:5000, nets[[i]]$errors, type='l',
    xlab='epochs', ylab='cost', main=paste('Train: ', i))
}
```

รูป 6.7 แสดงตัวอย่างครั้งที่ 1-20. หากการฝึกดูเรียบร้อยดี ผู้ฝึกก็สามารถนำค่าจากครอสวอลิดเดชั่นมาใช้เลือกค่าไฮเปอร์พารามิเตอร์ได้ โดยดูผลได้จากค่าที่บันทึกไว้ในตัวแปร `records` และสามารถนำไปวิเคราะห์ได้ดังตาราง 6.1.

เมื่อได้โมเดลแล้ว (กรณีนี้ คือได้ค่าไฮเปอร์พารามิเตอร์ ได้แก่ จำนวนหน่วยช่อง 30, อัตราเรียนรู้ชั้นช่อง 0.001, อัตราเรียนรู้ชั้นเอต้าพูต 0.01), การฝึกสามารถดำเนินต่อ โดยการนำโมเดลที่ได้มาหาค่าน้ำหนักที่ดีที่สุดจากข้อมูลทั้งหมด ดังนี้

<sup>5</sup> ที่นี่ใช้คำว่าครอสวอลิดเดชั่น เพื่อหมายถึงการทำวอลิดเดชั่นทุกพับ. แต่ละพับของครอสวอลิดเดชั่น ก็คือการทำวอลิดเดชั่น ซึ่งประกอบด้วยการทำครอสวอลิดเดชั่น 5 ครั้ง แล้วทำการทำซ้ำครอสวอลิดเดชั่น ห้าพับคือการทำวอลิดเดชั่น 5 ครั้งช้า เช่น หากทำครอสวอลิดเดชั่นห้าพับเท่ากับการทำวอลิดเดชั่น 5 ครั้ง และการทำซ้ำครอสวอลิดเดชั่นห้าพับคือการทำวอลิดเดชั่น 5 ครั้งช้า เช่น หากทำครอสวอลิดเดชั่นห้าพับช้า 40 ครั้ง ก็เท่ากับต้องการทำวอลิดเดชั่น  $5 \times 40 = 200$  ครั้ง เพียงแต่ทุก 5 ครั้งสรุปเป็นครอสวอลิดเดชั่น 1 ช้า. ครอสวอลิดเดชั่นแต่ละช้าจะมีค่าผลทดสอบที่ได้จากค่าเฉลี่ยของ 5 ครั้ง. ดังนั้นสำหรับ 40 ช้าจะมีค่าผลทดสอบ 40 ค่า และจะเลือกตัวแทนเป็นค่าที่ดีที่สุดจาก 40 ค่านี้.

```

fnets <- list()

for(i in 1:10){

  net <- nnTrain(set1.Xn, set1.T, nHiddens=30,
    rhoh=0.001, rho=0.01, wmax=0.5,
    nEpochs=5000, plottitle=paste('Cost at run ', i))
  fnets[[i]] <- net
}## end for i

```

สังเกตว่า แทนที่จะฝึกครั้งเดียว โคดตัวอย่างนี้ฝึก 10 ครั้ง และเก็บโมเดลที่ฝึกแต่ละครั้งไว้ในตัวแปร `fnets[[i]]` เมื่อ `i` เป็นตัวแปรสำหรับตัวนี้ของครั้งที่ฝึก. ตัวอย่างนี้ทำการฝึกทั้งหมด 10 ครั้ง และจะเลือกโมเดลสุดท้ายเพื่อไปใช้งานจากโมเดลที่ดีที่สุดของ 10 ครั้งนี้. การฝึกหลายครั้งและเลือกใช้โมเดลจากครั้งที่ให้ผลดีที่สุด ช่วยลดโอกาสที่การฝึกจะไปติดที่ค่าน้ำหนักແย่ๆ เนื่องจากสุ่มเริ่มต้นได้ค่าน้ำหนักແย่ๆ ลง. ผลของการฝึกแต่ละครั้งสามารถตรวจสอบได้ดังนี้

```

par(mfrow=c(5,2))
par(mar=c(2,2,2,1))

for(i in 1:10){

  set2.y <- hardlimit(nnOutput(fnets[[i]], set2.Xn, nntype='regression'))
  accuracy <- sum(set2.y == set2.T)/N2

  plot(1:5000, fnets[[i]]$errors, type='l',
    xlab='epochs', ylab='cost',
    main=paste('net ', i, ', correct ', round(accuracy,3)))
}

```

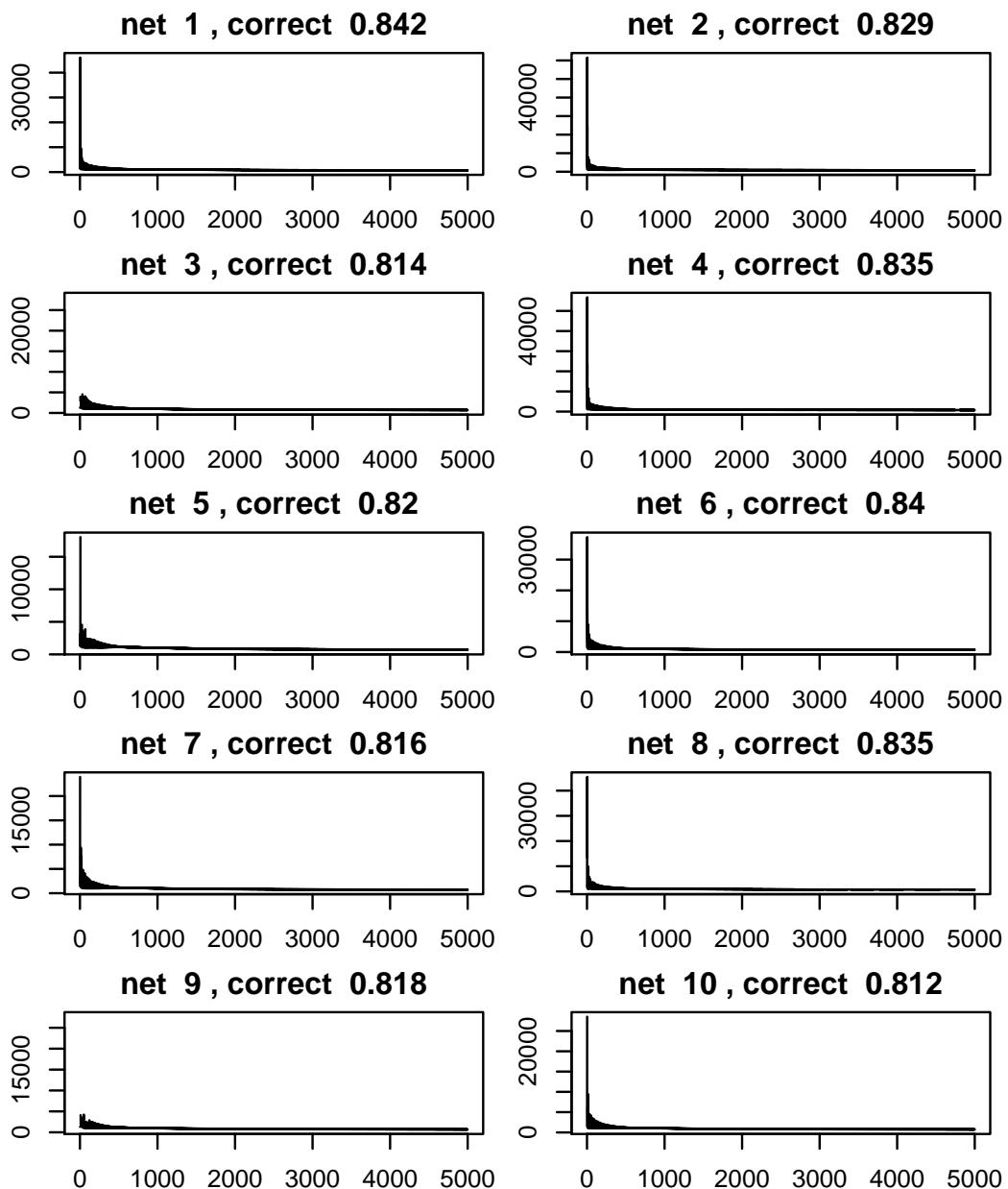
ซึ่งผลแสดงดังรูป [6.12](#).

จากรูป จะเห็นว่า (1) การฝึกทั้ง 10 ครั้งเป็นไปได้ด้วยดีทั้งหมด (ค่าผิดพลาดลู่เข้าสู่ค่าต่ำสุดเมื่อจบการฝึก) และ (2) ผลทดสอบกับข้อมูลชุดทดสอบสุดท้าย (ตัวแปร `set2.Xn` และ `set2.T`) ได้โมเดลที่ดีที่สุดคือ `fnets[[1]]` (ทำนายถูกต้องด้วยความแม่นยำ 0.842).

#### 6.5.3.6 ความเที่ยงตรง การเรียกกลับ และคะแนนเอฟ

จากโมเดลที่ได้ ค่าความเที่ยงตรง ค่าการเรียกกลับ และค่าคะแนนเอฟ สามารถคำนวณได้ดังนี้

```
set2.y <- hardlimit(nnOutput(fnets[[1]], set2.Xn, nntype='biclass'))
```



รูปที่ 6.12: ตัวอย่างโครงข่ายประสาทเทียมขนาด 30 หน่วยย่อย ที่อัตราการเรียนรู้  $\rho_1 = 0.001$  และ  $\rho_2 = 0.01$  และทำการฝึก 10 ครั้ง.

```

true.pos <- sum( (set2.y == 1) & (set2.T == 1) )
true.neg <- sum( (set2.y == 0) & (set2.T == 0) )
false.pos <- sum( (set2.y == 1) & (set2.T == 0) )
false.neg <- sum( (set2.y == 0) & (set2.T == 1) )

r.precis <- true.pos/(true.pos + false.pos)
r.recall <- true.pos/(true.pos + false.neg)
F.score <- 2*r.precis*r.recall/(r.precis + r.recall)

```

โดย accuracy, true.pos, true.neg, false.pos, false.neg, r.precis, r.recall, และ F.score คือ ค่าความแม่นยำ, ค่าบวกจริง, ค่าลบจริง, ค่าบวกเท็จ, ค่าลบเท็จ, ค่าความเที่ยงตรง, ค่าการเรียกกลับ, และค่าคะแนนเอฟ ตามลำดับ. ดูตาราง 6.2 ประกอบ.

#### 6.5.4 โค้ดสำหรับตัวอย่างการจำแนกกลุ่มแบบหลายกลุ่ม (ข้อมูลชุดรูปของลายมือ เขียนตัวเลข)

โค้ดสำหรับตัวอย่างข้อมูลชุดรูปของลายมือเขียนตัวเลขกีคล้ายๆ กับตัวอย่างก่อนๆ. สิ่งที่ต่างกันคือ (1) ตัวอย่างนี้เป็นงานการจำแนกกลุ่มแบบหลายกลุ่ม และ (2) มีการจัดการกับลักษณะเฉพาะของข้อมูล.

##### 6.5.4.1 โค้ดสำหรับการจำแนกกลุ่มแบบหลายกลุ่ม

สำหรับงานการจำแนกกลุ่มแบบหลายกลุ่มจะใช้เอาร์พุตแบบรหัสหนึ่งไปเค (1-of-K Coding) และเพื่อความสะดวก ตัวอย่างนี้สร้างฟังชั่นเฉพาะ encode.OK และ decode.OK สำหรับการแปลงเอาร์พุตที่บอกค่ากลุ่มไปเป็นรหัสหนึ่งไปเค และแปลงกลับ ตามลำดับ. รายการ 6.10 และ 6.11 แสดงโค้ดของ encode.OK และ decode.OK ตามลำดับ. ฟังชั่น encode.OK แปลงจากเอาร์พุตค่าเดียว เช่น กลุ่ม ‘2’ เป็น ค่ารหัสหนึ่งไปเค ( $K$  มิติ) เช่น [01000]. ส่วนฟังชั่น decode.OK แปลงจากเอาร์พุตในรหัสหนึ่งไปเค เช่น [01000] กลับเป็นค่าเดียว เช่น ‘2’.

รายการ 6.10: โค้ดฟังชั่น encode.OK

---

```

1 encode.OK <- function(T, classes=sort(unique(as.character(T)))){
2   K <- length(classes)
3   N <- length(T)
4
5   T.K <- (matrix(T,K,N, byrow=TRUE) ==
6             matrix(classes,K,N,byrow=FALSE))*1
7

```

```

8   rownames(T.K) <- classes
9   return(T.K)
10 }

```

#### รายการ 6.11: โค้ดฟังชัน decode.OK

```

1 decode.OK <- function(Y.K, classes=rownames(Y.K)){
2   if (is.null(classes)) {
3     classes <- as.character(seq(1, nrow(Y.K)))
4   }# end if
5
6   Y.class <- classes[apply(Y.K, 2, which.max)]
7   return(matrix(Y.class, nrow=1))
8 }

```

นอกจากนั้น โค้ดสำหรับโครงข่ายประสาทเทียมในรายการ 6.2 ต้องปรับเปลี่ยนให้เหมาะสมกับงานการจำแนกกลุ่มแบบหลายกลุ่มด้วย (ดูตาราง 5.3 ประกอบ) โดยการปรับแก้ฟังชันกระตุ้นของชั้นเออต์พุตให้เป็นซอฟต์แมกซ์ฟังชัน. นั่นคือ การแก้ไข บรรทัดที่ 36 ของ nnTrain ในรายการ 6.2 จาก  $Y \leftarrow W2 \%*%$  dotZ เป็น

```

A <- W2 %*% dotZ
Y <- exp(A)/matrix(colSums(exp(A)), K, N, byrow=TRUE)

```

เช่นเดียวกับตัวอย่างการจำแนกกลุ่มแบบสองกลุ่ม การคำนวณค่าผิดพลาดอาจปรับแก้ให้เข้ากับงานจำแนกกลุ่มแบบหลายกลุ่ม (สมการ 5.31 ตามที่ถูกในหัวข้อ 5.3) ได้แก่ การแก้ไขบรรทัดที่ 53 ของ nnTrain ในรายการ 6.2 เป็น

```
errors[epoch+firstEpoch-1] <- -sum( T * log(Y) )
```

#### 6.5.4.2 โค้ดสำหรับการฝึกและทดสอบ

การนำข้อมูลเข้าตัวแปรของอาร์เรย์ projectile และแปลงตัวแปรนี้ให้เป็นเมตริกซ์ สามารถทำได้ดังนี้

```

train.zip <- read.table('zip.train')
numzip <- apply(train.zip,c(1,2),as.numeric)

```

สังเกตุ ตัวแปร numzip จะเป็นเมตริกซ์ขนาด  $7291 \times 257$ . นั่นคือ ข้อมูลชุดนี้มี 7,291 จุดข้อมูล และแต่ละจุดข้อมูลมี 257 มิติ (1 มิติบอกว่าภาพเป็นภาพของเลขใด และ 256 มิติสำหรับค่าของแต่ละพิกเซลของภาพ). หากสำรวจดูค่าของตัวแปร numzip โดยคำสั่ง summary(numzip) จะเห็นว่าค่าของพิกเซล ถูกกำหนดอร์มอไลเซชันมาแล้ว (ค่าอยู่ในช่วง  $[-1, 1]$ ). คอลัมน์แรกแทนเฉลย และ 256 คอลัมน์ต่อมาเป็นค่าความเข้มของพิกเซลที่ถูกกำหนดอร์มอไลเซชันมาแล้ว.

ภาพของแต่ละจุดข้อมูลสามารถนำมาเรียงกลับเป็นภาพ เพื่อตรวจดูด้วยตาได้โดยใช้คำสั่ง `image` เช่น

```
image(seq(1,16), seq(1,16), t(apply(matrix(numzip[19,-1],16,16),1,rev)))
```

สำหรับคุณรูปของจุดข้อมูลที่ 19. สังเกตว่า มีการตัดมิติที่ 1 ออก (`numzip[19,-1]`) เพราะมิติที่ 1 ของตัวแปรเป็นค่าเฉลยที่บวกกัน เช่นหากเรียก `numzip[19,1]` ก็ได้ค่าเฉลยของมาว่าภาพในจุดข้อมูลที่ 19 นี้เป็นภาพของเลขได. ดังนั้น การแยกข้อมูลนี้ออกเป็นอินพุต `X` และเอาต์พุต `T` จึงทำดังนี้

```
X <- t(numzip[, -1]) # X is D x N
```

```
T <- t(numzip[, 1]) # T is 1 x N
```

ข้อมูลควรจัดสรรอากเป็นชุดฝึกและชุดทดสอบ ซึ่งสามารถทำได้ดังนี้

```
N <- ncol(X)
```

```
id.rand <- sample(N)
```

```
marker <- round(0.75*N)
```

```
train.X <- X[, id.rand[1:marker]]
```

```
train.T <- T[, id.rand[1:marker]]
```

```
train.T.K <- encode.OK(train.T)
```

```
validate.X <- X[, id.rand[-1:-marker]]
```

```
validate.T <- T[, id.rand[-1:-marker]]
```

```
validate.T.K <- encode.OK(validate.T)
```

ตัวอย่างนี้แยกratio 75% ของข้อมูลสำหรับการฝึก และที่เหลือสำหรับการทำวิเคราะห์ ข้อมูลชุดนี้มีชุดข้อมูลสำหรับทดสอบแยกไว้ต่างหากอยู่แล้ว. ชุดข้อมูลสำหรับทดสอบ ก็สามารถโหลดมาได้ในลักษณะเดียวกัน

```
test.zip <- read.table('zip.test')
```

```
test.num <- apply(test.zip, c(1,2), as.numeric)
```

```
test.X <- t(test.zip[, -1]) # X is D x N
```

```
test.T <- t(test.zip[, 1]) # T is 1 x N
```

โครงข่ายประสาทเทียมสามารถถูกฝึกได้เช่นเดียวกับตัวอย่างที่ผ่านมา และ เพื่อความสะดวก ตัวอย่างนี้ใช้การเขียนเป็นลูป เพื่อทำการฝึกหลายครั้ง สำหรับการสุมค่าเริ่มต้นของค่าน้ำหนัก ดังนี้

```

records <- matrix(0, 1, 10)
nets <- vector('list', 10)

for(i in 1:10){

  nets[[i]] <- nnTrain(train.X,train.T.K,
    nHiddens=40, rhoh=0.0002, rho=0.002, wmax=0.1,
    nEpochs=500, net=NULL,
    earlystopping=TRUE, early.tol=1,
    val.X=validate.X, val.T=validate.T.K)

  ## Test Network
  test.y <- nnOutput(nets[[i]],test.X, nntype='multiclass')
  Accuracy <- sum(
    decode.OK(test.y, c('0','1','2','3','4','5','6','7','8','9'))
    ==test.T)/N.test
  records[i] = Accuracy
}

```

โค้ดข้างต้น แสดงตัวอย่างการฝึก 10 ครั้ง เก็บโครงข่ายประสาทเทียมที่ฝึกแล้วทุกๆครั้ง (ซึ่งเก็บในตัวแปร `nets`). โครงข่ายประสาทเทียมที่ฝึกได้ดีที่สุดเลือกได้จากค่าความแม่นยำที่บันทึกไว้ในตัวแปร `records`. แต่ก่อนจะเลือก ผู้ใช้ต้องตรวจสอบดูว่าผลการฝึกเป็นไปด้วยดีหรือไม่ เช่น ตรวจสอบดูว่าค่าผิดพลาดระหว่างการฝึกกลุ่มข้าเรียบร้อยแล้ว ซึ่งอาจได้จากการล้อตผลการฝึกทั้ง 10 ครั้ง

โค้ดวัดผลการฝึกทั้ง 10 ครั้ง ทำได้ดังนี้

```

p=par(mfrow=c(2,5))
for(i in 1:10){
  N = max(which(nets[[i]]$errors > 0))
  plot(1:N, nets[[i]]$errors[1:N], xlab='epoch', ylab='cost',
    type='l', main= paste('Train ', i,
    '\nTest\'s correct ', round(records[i],2)))
}
par(p)

```

รูป 6.8 และ 6.9 แสดงตัวอย่างการตรวจสอบการฝึก. หากการฝึกมีปัญหา ก็ควรปรับปรุงแก้ไขให้เรียบร้อยก่อน เช่น หากค่าอัตราการเรียนรู้มากเกินไป ก็ปรับลดลงมา หรือหากผลการฝึกดูเหมือนต้องการรอบฝึกเพิ่ม ก็ควรปรับแก้ไขเพิ่มรอบฝึกขึ้นตามความเหมาะสมกับสถานะการณ์.

เมื่อผลการฝึกดูเรียบร้อยดี ผู้เตรียมโครงข่ายประสาทเทียมก็จะสามารถเลือกโครงข่ายประสาทเทียมจากครั้งที่ได้ผลดีที่สุด เช่น เมื่อการฝึกครั้งที่ 2 ให้ผลดีที่สุด ตัวอย่างนี้ก็เลือกโครงข่ายที่เก็บในตัวแปร nets[[2]] มาใช้งาน. ตัวอย่างโค้ดข้างล่างเก็บโครงข่ายประสาทเทียมนี้ไว้ใช้งานต่อไปในไฟล์ chosenNet.RData ดังนี้

```
net <- nets[[2]]
save(net, file='chosenNet.RData')
```

เมื่อต้องการนำโครงข่ายประสาทเทียมมาใช้งาน ก็เพียงแต่เรียก

```
load('chosenNet.RData')
```

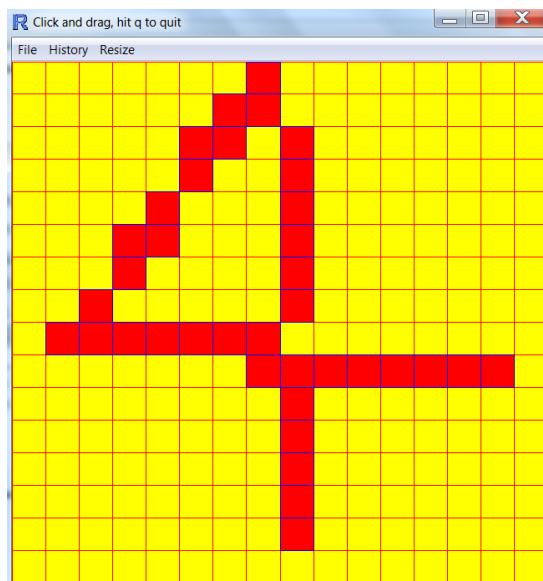
จากนั้นก็สามารถนำตัวแปรที่บันทึกไว้ (ตัวแปร net) ไปใช้งานได้เลย เช่น สามารถเรียกใช้ nnOutput(net, oinputที่ลงลับ, nntype='multiclass') เพื่อทำนายกลุ่มจากค่าอินพุตที่ลงลับ. หัวข้อ 6.5.4.3 แสดงตัวอย่างแอพพลิเคชันง่ายๆที่ใช้โครงข่ายประสาทเทียมที่บันทึกเก็บไว้นี้.

### 6.5.4.3 โค้ดสำหรับทดลองโปรแกรมใช้งานโครงข่ายประสาทเทียม

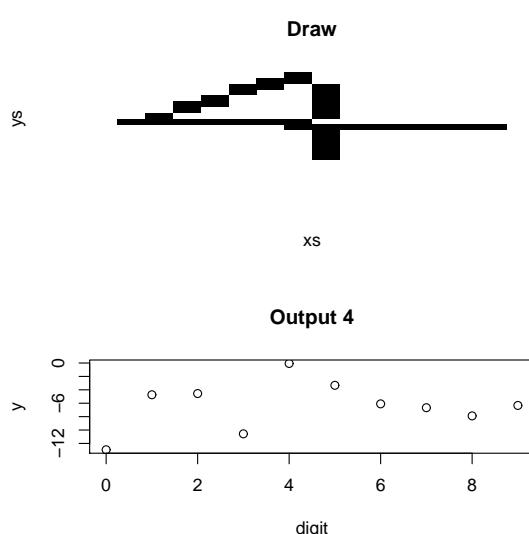
โค้ดในรายการ 6.12 เมื่อรันโดยการเรียก source('freeDraw.r') โปรแกรมจะเปิดหน้าต่างรับอินพุตขึ้นมา (รูป 6.13). ที่หน้าต่างรับอินพุต ผู้ใช้สามารถเขียนตัวเลขที่ต้องการลงไปได้โดยใช้เม้าส์. การเขียนตัวเลขเริ่มโดยการคลิกข่าวครั้งแรก เพื่อของการเริ่มเขียน แล้วจึงลากเมาส์เพื่อเขียนเลข (ไม่จำเป็นกดปุ่มใดๆ) และเมื่อต้องการหยุดเขียนให้คลิกเมาส์อีกครั้ง. ผู้ใช้สามารถเริ่มเขียนและหยุดเขียนได้ตลอด โดยคลิกขวาเพื่อสลับระหว่างการเขียนและหยุดเขียน. จนเมื่อเขียนเสร็จและต้องการให้โปรแกรมอ่านว่าเป็นเลขใด ให้กดคีย์บนคีย์บอร์ด (คีย์ใดก็ได้ เช่น 'q', 'w', 'e', ...). รูป 6.13 แสดงตัวอย่างหน้าต่างรับอินพุตที่วัดเลข 4 ลงไปแล้ว. เมื่อกดคีย์ โปรแกรมจะอ่านออกมาว่าเป็นภาพที่วัดเป็นภาพของเลขใด. ตัวอย่างผลแสดงดังรูป 6.14.

โค้ดโปรแกรม freeDraw (รายการ 6.12) เรียกใช้ library(grid) สำหรับทำกราฟฟิกส์ และการโหลดโครงข่ายประสาทเทียมที่ฝึกไว้แล้วทำในบรรทัดที่ 10. โปรแกรม freeDraw เรียกใช้ พังชั้นต่างๆที่ได้อภิรายไปแล้ว เช่น nnOutput, sigmoid, decode.OK. ตัวโปรแกรมจะเริ่มด้วยการเปิดหน้าต่างรับอินพุต (บรรทัด 90-104) แล้วลงทะเบียนพังชั้นสำหรับจัดการอีเวนต์ (บรรทัด 203-206) โดยลงทะเบียนพังชั้นต่างๆที่สำหรับจัดการอีเวนต์ ได้แก่

- พังชั้น dragmousedown สำหรับการคลิกเมาส์ ซึ่งจะสลับระหว่างการเริ่มวัดและการหยุดวัด.  
หากเริ่มวัด พังชั้น dragmousedown จะวัดตำแหน่งของเมาส์บนหน้าต่างรับอินพุต และลงทะเบียน



รูปที่ 6.13: ตัวอย่างหน้าต่างรับอินพุตที่ผู้ใช้วาดเลข 4 ลงไป.



รูปที่ 6.14: ตัวอย่างผลจากโปรแกรม freeDraw. ภาพบนแสดงภาพที่ผู้ใช้วาด. ภาพล่างแสดงค่าซอร์ฟต์แมกน์ของหั้งสิบกั่มและโครงข่ายประสาทเทียมจะเลือกทายเป็นกลุ่มที่มีค่าซอร์ฟต์แมกน์สูงที่สุด ซึ่งในภาพนี้คือกลุ่มของตัวเลข ‘4’.

ฟังชัน dragmousemove สำหรับอีเวนต์ที่มีการเลื่อนเมาส์ (บรรทัด 171). แต่หากหยุดดาวด ก็จะถอนการลงทะเบียนรับอีเวนต์การเลื่อนเมาส์ออก (บรรทัด 177). ฟังชัน dragmousedown ใช้ตัวแปร write.on เก็บสถานะของหน้าต่างว่าอยู่ในสถานะ “วาด” หรือ “หยุดดาวด”

ตัวแปรที่ใช้เก็บค่าของภาพที่วาด (ค่าของพิกเซลขนาด  $16 \times 16$ ) คือ ตัวแปร zs ประกาศในบรรทัด 55 ด้วยค่าเริ่มต้นเป็น 0 สำหรับทุกพิกเซล. สังเกตุการกำหนดค่าสำหรับตัวแปรที่เป็นตัวแปรส่วนกลาง (global variable) ที่ใช้ตัวปฏิบัติการ <<- แทน <- หรือ =. ค่าใน zs จะถูกเซต ด้วยฟังชัน setZ. บรรทัด 111-112 ของฟังชัน setZ ทำการแปลงจากตำแหน่งของเมาส์ มาเป็นตำแหน่งของพิกเซล ก่อนที่จะเซตค่าของ zs ที่ตรงกับตำแหน่งพิกเซลนั้น (บรรทัด 114). ฟังชัน setZ จะถูกเรียก เมื่อฟังชัน dragmousedown เริ่มวาด (บรรทัด 173) หรือ drawmousemove ทำงาน (บรรทัด 186). ฟังชัน drawZ จะวาดภาพในหน้าต่างรับอินพุตใหม่ให้ตอบสนองการเปลี่ยนแปลง.

- ฟังชัน keydown สำหรับจัดการการกดคีย์ ซึ่งจะเรียกฟังชัน draw.off (บรรทัด 61-78) มาทำงาน. ฟังชัน draw.off จะนำภาพที่ผู้ใช้วาดมาจัดเรียง และแปลงค่าให้อยู่ในรูปแบบอินพุตของโครงข่ายประสาทเทียม (บรรทัด 67) ก่อนจะเรียกใช้โครงข่ายประสาทเทียมเพื่อจำแนกว่าภาพที่วาดเป็นภาพของเลขใด (บรรทัด 70) และแสดงผลออกมา (บรรทัด 72-74).

สำหรับรายละเอียด การโปรแกรมอาร์โปรเจคแบบอีเวนต์drif Fen (event-driven) ผู้อ่านสามารถศึกษาเพิ่มเติมได้จาก help(getGraphicsEvent) และการจัดการกราฟฟิกส์ในโปรแกรม freeDraw ก็สามารถศึกษาเพิ่มเติมได้จาก help(grid.polygon) และ help(grid.rect).

รายการ 6.12: โค้ดโปรแกรม freeDraw (ให้ตั้งชื่อไฟล์เป็น freeDraw.r)

```

1 ## Created Aug 3rd, 2013
2 ## Modified from help(getGraphicsEvent)
3
4 library(grid)
5
6 #####
7 ## Load trained net
8 #####
9
10 load('chosenNet.RData')
11
12 #####
13 ## functions
14 #####
15
16 gray2 = seq(0,1, len=2)
17
18 disp.img <- function(ImageData,...){

```

```

19   xs <- seq(1,16)
20   ys <- xs
21   z <- matrix(ImageData, 16, 16)
22   zr <- apply(z,1,rev)
23   image(xs,ys,t(zr), ...)
24 }
25
26 nnOutput <- function (net,X) {
27
28   K <- nrow(net$W2)
29   N <- ncol(X)
30
31   dotX <- rbind(1,X)
32   Z <- sigmoid( net$W1 %*% dotX )
33   dotZ <- rbind(1,Z)
34   A <- net$W2 %*% dotZ
35   Y <- exp(A)/matrix(colSums(exp(A)), K, N, byrow=TRUE)
36
37   return(Y)
38 }
39
40 sigmoid <- function (a) 1 / (1 + exp(-a))
41
42 decode.OK <- function(Y.K, classes=rownames(Y.K)){
43   if (is.null(classes)) {
44     classes <- as.character(seq(1,nrow(Y.K)))
45   }# end if
46
47   Y.class <- classes[apply(Y.K, 2, which.max)]
48   return(matrix(Y.class,nrow=1))
49 }
50
51 #####
52 ## Global variable
53 #####
54
55 zs <- matrix(0, 16, 16)
56
57 #####
58 ## draw.off
59 #####

```

```
60
61 draw.off <- function(){
62     x11()
63     p=par(mfrow=c(2,1))
64
65     zs.zip <- matrix(0, 256, 1)
66
67     zs.zip <- 2*c(t(apply(zs,1, rev)))-1
68     disp.img(zs.zip, axes=F, col=gray2, main='Draw')
69
70     draw.y <- nnOutput(net, matrix(zs.zip, 256, 1))
71
72     plot(0:9, log(draw.y),
73           main=paste('Output', decode.OK(draw.y)),
74           xlab='digit', ylab='y')
75
76     par(p)
77
78 }##end draw.off
79
80
81 #####
82 ## draw.on
83 #####
84
85 draw.on <- function() {
86
87     xs <- (1:16 - 0.5)/16
88     ys <- (1:16 - 0.5)/16
89
90     #####
91     ## Draw template
92     #####
93
94     grid.polygon(c(0,1,1,0), c(0,0,1,1),
95     gp = gpar(col='blue', fill='yellow'))
96
97     x.grid <- rep(xs, each=16)
98     y.grid <- rep(ys, times=16)
99
100    grid.rect(x = x.grid,
```

```

101             y = y.grid,
102             width = 1/16,
103             height = 1/16,
104             gp = gpar(col="red"))
105
106 #####
107 ## Define functions and handlers
108 #####
109
110 setZ <- function(x, y){
111     ix <- which.min((xs - x)^2)
112     iy <- which.min((ys - y)^2)
113
114     zs[ix, iy] <- 1
115
116     return(list(i=ix, j=iy))
117 }##end setZ
118
119 drawZ.all <- function(z){
120
121     for(i in 1:16){
122         for(j in 1:16){
123             if(zs[i,j] == 1){
124                 grid.rect(x = xs[i], y = ys[j],
125                             width = 1/16,    height = 1/16,
126                             gp = gpar(col='blue', fill='red'))
127             }else{
128                 grid.rect(x = xs[i], y = ys[j],
129                             width = 1/16,    height = 1/16,
130                             gp = gpar(col='blue', fill='yellow'))
131             }
132         }
133     }##end for ij
134
135 }##end drawZ.all
136
137 drawZ <- function(ix, iy){
138
139     if(zs[ix,iy] == 1){
140         grid.rect(x = xs[ix], y = ys[iy],
141                     width = 1/16,    height = 1/16,

```

```
142                     gp = gpar(col='blue', fill='red'))
143             }else{
144                 grid.rect(x = xs[ix], y = ys[iy],
145                             width = 1/16, height = 1/16,
146                             gp = gpar(col='blue', fill='yellow'))
147             }
148
149     }#end drawZ
150
151 ##########
152 ## Handlers
153 #########
154
155 startx <- NULL
156 starty <- NULL
157 usr <- NULL
158     write.on <- FALSE
159
160 devset <- function()
161     if (dev.cur() != eventEnv$which) dev.set(eventEnv$which)
162
163 dragmousedown <- function(buttons, x, y) {
164     startx <- x
165     starty <- y
166     devset()
167     usr <- par("usr")
168     write.on <- !(write.on)
169
170     if(write.on){
171         eventEnv$onMouseMove <- dragmousemove
172
173             loc.Z <- setZ(x, y)
174             drawZ(loc.Z$i, loc.Z$j)
175
176     } else {
177         eventEnv$onMouseMove <- NULL
178     }##end if(write.on)
179
180     NULL
181 }
182 }
```

```

183     dragmousemove <- function(buttons, x, y) {
184         devset()
185
186         loc.Z <- setZ(x, y)
187         drawZ(loc.Z$i, loc.Z$j)
188         NULL
189     }
190
191     mouseup <- function(buttons, x, y) {
192         eventEnv$onMouseMove <- NULL
193     }
194
195     keydown <- function(key) {
196
197         draw.off()
198
199         eventEnv$onMouseMove <- NULL
200         NULL
201     }
202
203     setGraphicsEventHandlers(prompt="Click and drag, hit q to quit",
204                             onMouseDown = dragmousedown,
205                             onKeybd = keydown)
206     eventEnv <- getGraphicsEventEnv()
207
208 }## end draw.on()
209
210 #####
211 ## main
212 #####
213
214
215 savepar <- par(ask=FALSE)
216
217 draw.on()
218 # This currently only works on the Windows
219 # and X11(type = "Xlib") screen devices...
220
221 getGraphicsEvent()
222
223 par(savepar)

```

---

โปรแกรม freeDraw นี้เป็นโปรแกรมสั้นๆ เพื่อแสดงให้เห็นการนำโมเดลที่ฝึกแล้วไปใช้งาน. หลังจากได้โมเดลแล้ว ผู้พัฒนาโปรแกรมไม่จำเป็นต้องทำการฝึกอีก ดังตัวอย่างในโปรแกรม freeDraw นี้. แต่หากผู้พัฒนาโปรแกรมต้องการให้แอพพลิเคชันสามารถปรับปรุงตัวเองได้ตลอด เช่น หากผู้ใช้เขียนออกไปแล้ว โปรแกรมอ่านเป็นตัวเลขผิด ผู้ใช้อาจจะกดลบแล้ววัดใหม่ หรือผู้ใช้อาจจะเลือกพิมพ์เข้าไปแทน ซึ่งพฤติกรรมเหล่านี้สามารถตรวจสอบได้และ ก็สามารถเพิ่มการเรียนรู้เข้าไป เพื่อทำให้โปรแกรมทำงานได้ดีขึ้น ได้ หรือแม้แต่การเรียงกลุ่มที่มีค่าของฟ์แมกซ์สูงสุด และให้ผู้ใช้เลือกกลุ่มที่ถูกต้อง ก็จะสามารถได้รับผลลัพธ์จากผู้ใช้ตลอดการใช้งาน โดยที่ไม่รบกวนผู้ใช้มากเกินไป.

## 6.6 แบบฝึกหัด

1. จงเลือกชุดข้อมูลมาสำหรับการหาค่าตัดตอน 1 ชุด สำหรับการแบ่งกลุ่ม 1 ชุด, เปรียบเทียบผลการใช้โครงข่ายประสาทเทียม ในแบบมุ่งของการใช้จำนวนหน่วยช้อนต่างๆ ค่าอัตราการเรียนรู้ต่างๆ และจำนวนรอบฝึกต่างๆ. อภิปรายความสัมพันธ์ของจำนวนหน่วยช้อน, ค่าอัตราการเรียนรู้ของชั้นช้อนและชั้นเอาต์พุต และจำนวนรอบฝึกในเรื่องผลการทำงาน.
2. จงเลือกชุดข้อมูลมาสำหรับการหาค่าตัดตอน 2 ชุด สำหรับการแบ่งกลุ่ม 2 ชุด, เปรียบเทียบการสุ่มค่าเริ่มต้นให้กับโครงข่ายประสาทเทียมแบบต่างๆ เช่นสุ่มจากการกระจายรูปเดียว (uniform distribution) ที่ช่วงค่าต่างๆ (ค่า  $w_{max}$  ในรายการ 6.1) หรือสุ่มจากการกระจายปกติ (normal distribution) ที่ใช้ค่าเบี่ยงเบนมาตรฐานต่างๆ. อภิปรายผลที่ได้.
3. จงเลือกชุดข้อมูลมาสำหรับการหาค่าตัดตอน 2 ชุด สำหรับการแบ่งกลุ่ม 2 ชุด, เปรียบเทียบการใช้โครงข่ายประสาทเทียมที่มีการทำอิร์มอไลเซชันกับข้อมูล กับการใช้โครงข่ายประสาทเทียมที่ไม่มีการทำอิร์มอไลเซชัน. อภิปรายผลที่ได้.
4. จากแบบฝึกหัดข้อ 4 บท 5 จงตัดแปลงโค้ดของ `nnTrain` (รายการ 6.2) เพื่อให้รองรับการทำReLU ไตรเชชัน. คำใบ้ `dE2` และ `dE1` ในบรรทัด 46 และ 47 (รายการ 6.2) คือ  $\frac{\partial E_n}{\partial w_{ji}^{(2)}}$  และ  $\frac{\partial E_n}{\partial w_{ji}^{(1)}}$ , ดูสมการ 5.42 และ 5.41 ประกอบ.
5. จากตัวอย่างในหัวข้อ 6.1 จงทดลองทำใหม่ด้วยโครงข่ายประสาทเทียมกับการฝึกที่ใช้ReLUไตรเชชัน ให้ทดลองค่า  $\lambda_1$  และ  $\lambda_2$  หลายๆค่า เช่น  $0, 0.000001, 0.001, 1, 10$  เป็นต้น. จงอภิปรายผลโดยเฉพาะความสัมพันธ์ของค่า  $\lambda_1$  และ  $\lambda_2$  กับผลการทำนายที่ได้.

6. จงเลือกชุดข้อมูลมา 3 ชุดสำหรับการหาค่าถดถอย และใช้โครงข่ายประสาทเทียมในการทำโมเดลและการประเมินผล. จงเปรียบเทียบและอภิปรายผลระหว่างการใช้การหยุดก่อนกำหนด (Early Stopping), การทำเรกูลาไรเซชัน (Regularization) และการที่ไม่ใช้ทั้งสองอย่าง.
7. จงเลือกชุดข้อมูลมา 3 ชุดสำหรับการจำแนกกลุ่ม และใช้โครงข่ายประสาทเทียมในการทำโมเดลและการประเมินผล. จงเปรียบเทียบและอภิปรายผลการใช้การหยุดก่อนกำหนด, การทำเรกูลาไรเซชัน และการที่ไม่ใช้ทั้งสองอย่าง.
8. จงเลือกชุดข้อมูล 3 ชุดที่แต่ละชุดมีเขตข้อมูลบางเขตขาดหาย และจงเปรียบเทียบ อภิปรายผล จากวิธีจัดการกับเขตข้อมูลบางเขตขาดหาย วิธีต่างๆ ได้แก่ วิธีการตัดระเบียนที่มีเขตข้อมูลขาดหาย วิธีการแทนค่าที่ขาดหายด้วยทุกค่าที่เป็นไปได้ วิธีการแทนค่าที่ขาดหายไปด้วยค่าเฉลี่ยหรือค่าที่พบบ่อยที่สุด.
9. จงศึกษาผลของการที่มีข้อมูลขาดหาย โดยเลือกชุดข้อมูลที่ไม่มีเขตข้อมูลขาดหาย (no missing data) มา 1 ชุด และทดลองสร้างชุดข้อมูลใหม่ 5 ชุด โดยที่แต่ละชุดสร้างจากข้อมูลต้นฉบับ แต่สุ่มละค่าบางค่าของข้อมูลออกไป 0.1%, 1%, 10%, 20%, และ 40%. ให้ใช้โครงข่ายประสาทเทียมทำโมเดลกับข้อมูลทั้ง 6 ชุดนี้ (รวมต้นฉบับด้วย) ประเมินผล และเปรียบเทียบ. ให้เลือกวิธีใช้วิธีจัดการกับข้อมูลขาดหาย พร้อมอธิบายเหตุผล. จงอภิปรายความสัมพันธ์ระหว่างปริมาณข้อมูลขาดหายและผลการทำนาย. (ดูตัวอย่างจาก [37])
10. จงออกแบบการทดลอง เพื่อเปรียบเทียบผลของการใช้วาลีเดชัน (แบ่งสองกลุ่มทำทีเดียว) กับผลของการใช้ครอสวาลีเดชัน (แบ่งเป็นหลายพับและทำเท่าจำนวนพับครึ่ง) โดยเฉพาะปัจจัยที่ขนาดข้อมูลต่างๆ. คำแนะนำ ให้ใช้ข้อมูลอย่างน้อย 3 ชุด และเลือกให้ข้อมูลมีความยากง่ายต่างกัน. และจากข้อมูล 3 ชุดหลัก ให้สร้างชุดข้อมูลใหม่ที่ขนาดเล็กลง จากการสุ่มจากชุดข้อมูลเดิม เพื่อใช้ในการศึกษาผลที่ปริมาณข้อมูลต่างกัน.
11. จากวิธีคำนวณค่าเกรเดียนต์เชิงเลข

$$\nabla_{\theta} J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \vdots \\ \frac{\partial J}{\partial \theta_M} \end{bmatrix}, \text{ และ } \frac{\partial J}{\partial \theta_i} \approx \frac{J(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i + \varepsilon \\ \theta_{i+1} \\ \vdots \end{bmatrix}) - J(\begin{bmatrix} \vdots \\ \theta_{i-1} \\ \theta_i - \varepsilon \\ \theta_{i+1} \\ \vdots \end{bmatrix})}{2\varepsilon} \quad (6.6)$$

เมื่อ  $J(\theta)$  คือค่าฟังชันจุดประสงค์ สำหรับปัญหา 3 แบบ ได้แก่ การหาค่าถดถอย การจำแนกกลุ่มแบบสองกลุ่ม และการจำแนกกลุ่มแบบหลายกลุ่ม, จงทดสอบค่าเกรเดียนต์ที่คำนวณจากวิธีแพร่กระจายเบรียบเทียบกับค่าเกรเดียนต์ที่คำนวณจากการเชิงเลข (ดูโค้ดในรายการ 6.13) โดยให้ใช้ค่า  $\epsilon$  เป็น 1, 0.01 และ 0.0001 ตามลำดับ. จงสรุปและอภิปรายผล.

คำใบ้ ดูตัวอย่างโค้ดวิธีตรวจสอบจากรายการ 6.14. หากทุกอย่างถูกต้อง ค่าเกรเดียนต์ที่คำนวณจากการวิเคราะห์ `fn.grad` ควรจะใกล้เคียงมากๆ กับค่าเกรเดียนต์ ที่ประมาณจากการคำนวณเชิงเลข `num.vals`.

**รายการ 6.13:** ฟังชัน `numericalGrad`. โค้ดตัวอย่างการคำนวณค่าเกรเดียนต์จากฟังชันจุดประสงค์ด้วยวิธีเชิงเลข การคำนวณด้วยวิธีเชิงเลขทำเพื่อใช้ตรวจสอบความถูกต้องของโค้ดการแพร่กระจายย้อนกลับ

---

```

1 numericalGrad <- function(cost.fn, w, epsilon=1e-4){
2   ## cost.fn(w) returns a single value of cost function.
3   ## w: a vector of parameters
4
5   Nw = length(w)
6   numgrad = matrix(0, Nw, 1);
7
8   for( i in 1:Nw){
9     theta1 = w;    theta2 = w;
10    theta1[i] = w[i] + epsilon;
11    theta2[i] = w[i] - epsilon;
12
13    numgrad[i] = ( cost.fn(theta1) - cost.fn(theta2) );
14  }##end for
15
16 return(numgrad/(2*epsilon))
17 }##end numericalGrad(..)

```

---

**รายการ 6.14:** ตัวอย่างการเปรียบเทียบค่าเกรเดียนต์กับค่าประมาณเกรเดียนต์ด้วยวิธีเชิงเลข

---

```

1 quadratic.fn <- function(x){
2   val = x[1]^2 + 3*x[1]*x[2]
3   grad = matrix(0, 2, 1)
4   grad[1] = 2*x[1] + 3*x[2]
5   grad[2] = 3*x[1]
6   return(list(cost=val, grad=grad))
7 }##end quadratic.fn
8
9 x = matrix(c(4, 10), 2, 1)
10 quad.vals <- quadratic.fn(x)
11

```

---

```
12 fn.cost ← quad.vals$cost  
13 fn.grad ← quad.vals$grad  
14  
15 num.vals ← numericalGrad(function(a){quadratic.fn(a)$cost}, x, epsilon=0.01)
```

---

12. จงเลือกชุดข้อมูล 1 ชุด และวัดค่าความสัมพันธ์ของค่าเออาร์พุตของหน่วยซ่อนต่างๆ สรุปผล และอภิปรายความสัมพันธ์ของค่าเออาร์พุตของหน่วยซ่อนต่างๆ กับการทำงานของโครงข่ายประสาทเทียม.

## บทที่ 7

# การฝึกที่มีประสิทธิภาพและคำแนะนำเพิ่มเติม

“Failure is the key to success; each mistake teaches us something.” — Morihei Ueshiba

“ความล้มเหลวคือกุญแจสู่ความสำเร็จ แต่ละความผิดพลาดสอนเราบางอย่าง” — โมริเอนะ อุเอชิบะ

บทที่ 5 อธิบายหลักพื้นฐานของโครงข่ายประสาทเทียม. บทที่ 6 แสดงตัวอย่างการประยุกต์ใช้โครงข่ายประสาทเทียม. บทนี้จะถกถึงการฝึกที่มีประสิทธิภาพมากขึ้น และคำแนะนำสำหรับการประยุกต์ใช้โครงข่ายประสาทเทียม.

โครงข่ายประสาทเทียมที่ฝึกเสร็จแล้วจะสามารถทำการคำนวณได้อย่างรวดเร็ว. แต่หากผู้อ่านได้ทดลองทำตามตัวอย่างในบทที่ 6 คงพอรู้สึกว่า การเตรียมและการฝึกโครงข่ายประสาทเทียมจะใช้เวลาพอสมควร. เวลาที่ใช้ในการฝึกก็ขึ้นอยู่กับขนาดข้อมูลฝึก ขนาดของโครงข่าย และพารามิเตอร์ของการฝึก. ข้อมูลฝึกที่มีหลายมิติ ก็เท่ากับปัจจัยหนึ่งที่ทำให้ตัวแปรค่าน้ำหนักมีขนาดใหญ่ขึ้น (สมการ 5.11, ค่าน้ำหนักของโครงข่ายซึ่งที่หนึ่ง  $\mathbf{W}^{(1)} \in \mathbb{R}^{M \times (1+D)}$  โดย  $D$  แทนจำนวนมิติของอินพุต) ข้อมูลมีจำนวนจุดข้อมูลมาก ก็ทำให้แต่ละรอบฝึกต้องคำนวณกับข้อมูลมากขึ้น โครงข่ายประสาทเทียมที่มีขนาดใหญ่ก็มีจำนวนค่าน้ำหนักที่ต้องคำนวณมากขึ้น จำนวนรอบฝึกมากขึ้นก็จะใช้เวลาฝึกนานขึ้น.

อย่างที่ถกกันไป การฝึกโครงข่ายประสาทเทียมจริงๆแล้วก็คือการทำห้ามที่สุด โดยตัวทำน้อยที่สุดที่ต้องการหา ก็คือค่าน้ำหนัก และฟังชันจุดประสงค์ที่ต้องการให้มีค่าน้อยที่สุด ก็คือค่าผิดพลาดของการทำนาย. วิธีที่สาขิตในบทที่ 6 คือวิธีลงเกรเดียนต์. วิธีลงเกรเดียนต์ทำงานได้ดี มีเสถียรภาพสูง เนื่องจากโปรแกรมได้ง่าย. แต่ศาสตร์และคิลป์ของวิชาการหาค่าดีที่สุดมีอีกหลายวิธีที่สามารถทำงานได้เร็วกว่าวิธีลงเกรเดียนต์ เช่น วิธีลงเกรเดียนต์กับอัตราการเรียนรู้ที่ปรับตัวได้ (Gradient Descent with Adaptive Learning Rule), วิธีลงเกรเดียนต์กับโมเมนตัม (Gradient Descent with Momentum, คำย่อ GDM), วิธีบีอีพีจีเอส (BFGS), วิธีคอนจูเกตเกรเดียนต์ (Conjugate Gradient, คำย่อ CG), วิธีสเกลคอนจูเกตเกรเดียนต์ (Scaled Conjugate Gradient, คำย่อ SCG), วิธีเลเวนเบิร์ก-มาრค华ร์ด (Levenberg-Marquardt, คำ

ย่อ LM) เป็นต้น. จากหลากหลายวิธีที่สามารถนำมาใช้ได้ บทที่จะแสดงตัวอย่างการใช้ 3 วิธี ได้แก่ วิธีลงเกรเดียนต์กับโมเมนตัม วิธีบีอฟจีเอส และวิธีสเกลคอนจูเกตเกรเดียนต์.

ตัวอย่างต่อไปนี้มีจุดประสงค์หลัก เพื่อสาธิตการนำศาสตร์และคิลป์ของวิชาการหาค่าดีที่สุดมาใช้ช่วยการฝึกโครงข่ายประสาทเทียม ไม่ได้มีจุดประสงค์เพื่ออธิบายทฤษฎีเบื้องหลังวิธีเหล่านี้.

## 7.1 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีลงเกรเดียนต์กับโมเมนตัม

index Gradient Descent with Momentum

วิธีหาค่าน้อยที่สุด เช่น วิธีลงเกรเดียนต์ สามารถใช้ฝึกโครงข่ายประสาทเทียมได้ดี. หากแต่ส่วนใหญ่แล้ว วิธีลงเกรเดียนต์จะใช้เวลานานมาก. การทำงานช้านี้ส่วนหนึ่งอาจมาจากการเลือกค่าอัตราการเรียนรู้ที่ไม่เหมาะสม. วิธีลงชันที่สุดช่วยแก้ปัญหาการเลือกค่าอัตราการเรียนรู้ได้ โดยการใช้วิธีการค้นหาตามแนวเส้น (line search) เช่น วิธีค้นหาแบบช่วงของค่า (หัวข้อ 2.3). แต่แม้จะใช้วิธีลงชันที่สุดแล้ว การฝึกก็ยังอาจทำได้ช้าอยู่เนื่องมาจากการลู่เข้าในลักษณะส่ายไปมา ดังถูกในหัวข้อ 2.5.

ตัวอย่างเช่น การหาค่า  $[x_1, x_2]^T = \arg \min_{x_1, x_2} f(x_1, x_2)$  โดย

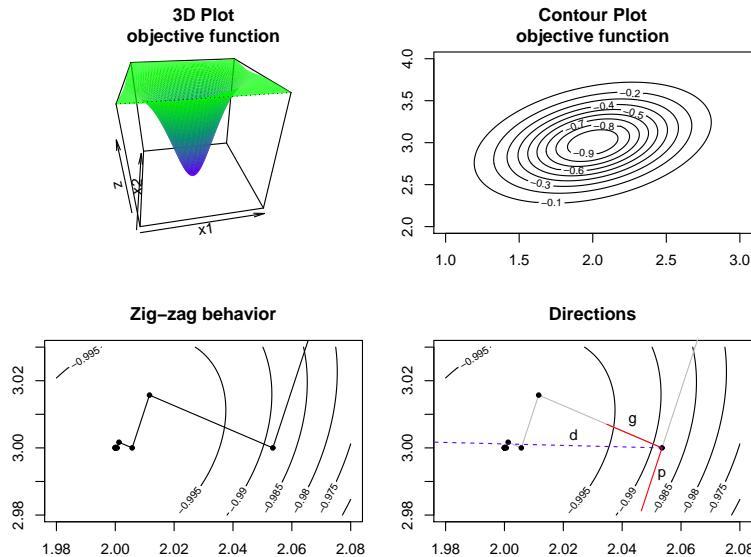
$$f(x_1, x_2) = -\exp(-\{4x_1^2 - 7x_1 - 3x_1x_2 - 24x_2 + 5x_2^2 + 43\}). \quad (7.1)$$

รูป 7.1 ภาพซ้ายบนแสดงค่าฟังชันจุดประสงค์  $f(x_1, x_2)$  เมื่อพล็อตสามมิติ. ภาพขวาบนแสดงเมื่อทำพล็อตแบบค่อนหัวร์. ภาพล่างซ้ายแสดงเส้นทางของค่า  $[x_1, x_2]^T$  ที่รอบคำนวนต่างๆ เมื่อคำนวนด้วยวิธีลงชันที่สุด. สังเกตุเส้นทางของค่าตัวแปรมีลักษณะส่ายไปมา (พฤติกรรมซิกแซก) ก่อนที่จะเข้าสู่ค่าตอบที่ถูกต้อง,  $[x_1^*, x_2^*]^T = [2, 3]^T$ . ภาพล่างขวาแสดงค่าตัวแปรที่รอบคำนวนหนึ่ง แล้วพิจารณาทิศทางการลงเกรเดียนต์ (เส้น  $\mathbf{g}$  ในภาพ) และโมเมนตัมที่วิ่งมา (เส้น  $\mathbf{p}$  ในภาพ) จะเห็นว่าทิศทางของโมเมนตัม  $\mathbf{p}$  กับทิศทางการลงเกรเดียนต์  $\mathbf{g}$  สามารถรวมกันเพื่อเป็นทิศทางที่จะใช้ปรับค่าตัวแปรได้ (เส้นประ  $\mathbf{d}$  ในภาพ). สังเกตุ เส้นประ  $\mathbf{d}$  ในภาพมีทิศทางที่เข้าหากำตอบที่ถูกต้องมากกว่าทิศทางลงเกรเดียนต์  $\mathbf{g}$ . สมการ 7.2 แสดงการปรับค่าตัวแปรตามแนวคิดนี้

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} + \alpha \mathbf{p}^{(i)} \quad (7.2)$$

$$\mathbf{p}^{(i+1)} = -\nabla_{\mathbf{w}} E^{(i+1)} + \beta \mathbf{p}^{(i)} \quad (7.3)$$

เมื่อ  $\mathbf{w}^{(i+1)}$  แทนค่าตัวแปรตัวที่รอบคำนวน  $i+1$ ,  $\alpha$  แทนขนาดก้าว และ  $\mathbf{p}^{(i)}$  แทนโมเมนตัมของรอบการคำนวนที่  $i$ . หลังจากปรับค่าตัวแปร  $\mathbf{w}^{(i+1)}$  เราจะได้โมเมนตัมของรอบคำนวนที่  $i+1$  ดังแสดงในสมการ 7.3, เมื่อ  $\nabla_{\tilde{\mathbf{w}}} E^{(i+1)}$  แทนเกรเดียนต์ของตัวแปรที่รอบคำนวน  $i+1$  และ  $\beta$  แทนอัตราโมเมนตัม



รูปที่ 7.1: พื้นที่นี้จุดประสงค์ (ภาพบนซ้ายแสดงพื้นที่นี้จุดประสงค์ในการวัดสามมิติ ภาพบนขวาพื้นที่นี้จุดประสงค์ในการวัดแบบคอนทัวร์) เส้นทางการลู่เข้าสู่ค่าตัวทำน้อยที่สุดด้วยวิธีลงชันที่สุดแสดงพฤษติกรรมซิกแซก (ภาพล่างซ้าย) และทิศทางลงเกรเดียนต์เมื่อรวมกับโมเมนต์ ณ รอบคำนวนหนึ่ง ซึ่งแสดงให้เห็นว่า หากใช้ทิศทางลงเกรเดียนต์รวมกับทิศทางโมเมนต์เป็นทิศทางในการปรับค่า จะช่วยลดปัญหาพฤษติกรรมซิกแซกได้.

(momentum rate). สังเกตุสมการ 7.3 เมื่อเปรียบเทียบกับภาพล่างซ้ายในรูป 7.1 นั้นคือ

$\mathbf{p}^{(i+1)}$  เปรียบกับทิศทาง  $\mathbf{d}$  ในภาพ,

$-\nabla_{\mathbf{w}} E^{(i+1)}$  เปรียบกับทิศทาง  $\mathbf{g}$  ในภาพ,

และ  $\mathbf{p}^{(i)}$  เปรียบกับทิศทาง  $\mathbf{p}$  ในภาพ

โดยมี  $\beta$  ที่สามารถใช้เพื่อปรับอัตราส่วนสมรรถห่วงทิศทาง  $\mathbf{g}$  และ  $\mathbf{p}$  เพื่อให้ได้ทิศทาง  $\mathbf{d}$  ที่เหมาะสมได.

รายการ 7.1 แสดงโค้ดวิธีลงเกรเดียนต์กับโมเมนต์. รูป 7.2 แสดงเส้นทางการปรับค่าตัวแปรของวิธีลงเกรเดียนต์กับโมเมนต์ เปรียบเทียบกับวิธีลงชันที่สุด และวิธีลงเกรเดียนต์. การกำหนดค่าพารามิเตอร์  $\alpha$  และ  $\beta$  อาจทำโดยการลองผิดลองถูก[66]. อย่างไรก็ตาม หากค่าที่ต้องการไม่ลู่เข้า แนะนำให้ทดลองลดค่าของพารามิเตอร์ทั้งสองลง. การให้ค่า  $\beta = 0$  จะทำให้วิธีลงเกรเดียนต์กับโมเมนต์ลดรูปไปเป็นวิธีลงเกรเดียนต์ธรรมด้า. ส่วนค่า  $\alpha$  คือขนาดก้าว ซึ่งความสัมพันธ์ของขนาดก้าวและการลู่เข้าได้อภิปรายในหัวข้อ 2.4.

รายการ 7.1: ตัวอย่างโค้ดวิธีลงเกรเดียนต์กับโมเมนต์

---

```

1 ##########
2 ## Gradient Descent with Momentum
3 #########
4
5 gdm <- function(grad, f, x0, alpha=0.01, beta=0.2, tol=0.00001, MaxN=1000, log=<-
  FALSE){
```

```

6 ## Example
7 ##   f  <- function(x){(x[1] - 4)^4 + (x[2] - 3)^2 + 4*(x[3] + 5)^4}
8 ##   df <- function(x){ matrix(c( 4*(x[1] - 4)^3, 2*(x[2] - 3), 16*(x[3] + 5)^3 )<-
9 ##                           ,3,1)}
9 ##   gdm(df, f, alpha=0.008, beta=0.2, x0=matrix(c(4,2,-1),3,1), Log=TRUE, tol=1e<-
10 ##        -5)

11 D <- length(x0)
12
13 p <- matrix(0, D, 1)
14 logs <- matrix(c(0,p,x0),1+2*D,1)
15
16 diff <- tol^2
17 for (i in 1:MaxN) {
18
19   gradF <- grad(x0)
20   p <- -gradF + beta*p
21   x <- x0 + alpha*p
22
23   x0 <- x
24   logs <- cbind(logs, c(i,p,x))
25   diff <- sqrt( mean((gradF)^2) )
26   if (diff < tol) break
27 }## for
28
29 if(log){ return(logs) }
30 else { return(x) }
31 }## end gdm

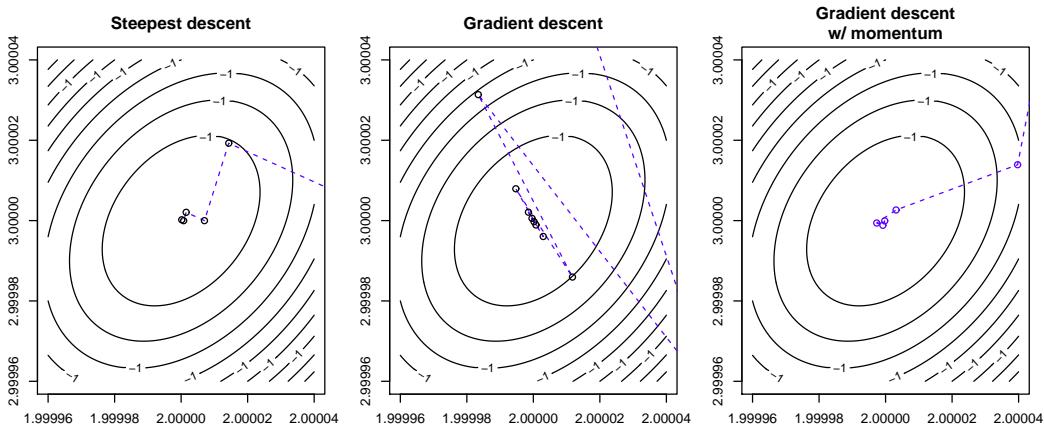
```

---

## 7.2 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีบีเอฟจีเอส

วิธีบีเอฟจีเอส (BFGS Method) ซึ่งชื่อวิธี มาจากอักชระแรกของ Broyden, Fletcher, Goldfarb, และ Shanno ที่เป็นชื่อเหล่าผู้พัฒนาอัลกอริทึมนี้ ที่พัฒนาออกมากพร้อมๆ กันอย่างเป็นอิสระในปี ค.ศ. 1970. วิธีบีเอฟจีเอสพัฒนามาจากวิธีนิวตัน (Newton Method). นอกจากจะใช้เกรเดียนต์ที่เป็นอนุพันธ์อันดับหนึ่ง แล้ว วิธีนิวตันยังใช้ไฮเซียน (Hessian) ที่เป็นอนุพันธ์อันดับสองด้วย. ผู้อ่านที่สนใจรายละเอียดทฤษฎีเบื้องหลังของวิธีบีเอฟจีเอสสามารถศึกษาเพิ่มเติมได้จากทำการหาค่าดีที่สุดเบื้องต้น[18].

วิธีบีเอฟจีเอสนิยมใช้มากในงานทั่วไปของการหาค่าดีที่สุด และในอาร์โพรเจกต์มีฟังชันสำหรับให้ใช้ได้แก่ ฟังชัน `optim` และเลือกวิธีเป็น `BFGS`. ฟังชัน `optim` เป็นฟังชันที่นำไปสำหรับทำการหาค่าดีที่สุด



รูปที่ 7.2: พฤติกรรมลู่เข้าของวิธีลิงชันที่สุด วิธีลิงเกรเดียนต์ และวิธีลิงเกรเดียนต์กับโมเมนตัม.

และมีวิธีให้เลือกใช้หลายวิธี ออาทิ เนลเดอร์-เมด (Nelder–Mead[56]) บีเอฟจีเอส และคอนจูเกตเกรเดียนต์ (CG[26]) รวมถึงวิธีการหาค่าดีที่สุดเชิงข้อจำกัดกล่อง (Box-Constrained Optimization[15]) และวิธีการอบอุ่นจำลอง (Simulated Annealing[8, 44]).

รายการ 7.2 แสดงโค้ดการใช้งานฟังชัน `optim` สำหรับตัวอย่าง่ายๆ. ฟังชัน `ob.fn` เป็นฟังชันจุดประสงค์  $f(\mathbf{x}) = (x_1 - 8)^2 + (x_2 - 4)^2$  ซึ่งเห็นได้ว่าค่าตัวทำน้อยที่สุดคือ  $\mathbf{x} = [8, 4]^T$ . เกรเดียนต์ของฟังชันจุดประสงค์  $[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}]^T$  กำหนดใน `gr.fn`. ตัวอย่างนี้ทดลองให้ค่าเริ่มต้นเป็น  $[0, 0]$  (บรรทัด 6). บรรทัด 7 เรียกใช้ฟังชัน `optim` ด้วยวิธี BFGS. และคำตอบแสดงในตัวแปร `r$par`. เช่นเดียวกับวิธีลิงเกรเดียนต์ ผู้ใช้ควรตอบสองก่อนว่าอัลกอริทึมทำงานได้สมบูรณ์ก่อนจะนำคำตอบไปใช้งาน เช่น ผู้ใช้ควรตรวจสอบค่า `r$convergence` ว่าเป็น 0 หรือไม่. ค่า `r$convergence` เป็น 0 หมายถึงอัลกอริทึมลู่เข้า.

ผลของ `optim` จะตอบออกมารูปแบบเป็นชนิดตัวแปรชนิด `list` ที่ประกอบด้วย

ตัวแปรอยู่ <code>\$par</code>	ซึ่งคือคำตอบที่ต้องการ (ตัวทำน้อยสุด),
ตัวแปรอยู่ <code>\$value</code>	(ค่าฟังชันจุดประสงค์ที่น้อยที่สุด),
ตัวแปรอยู่ <code>\$counts</code>	บอกจำนวนครั้งการเรียกใช้
	<code>function</code> สำหรับฟังชันจุดประสงค์ (ได้แก่ <code>ob.fn</code> )
	และ <code>gradient</code> สำหรับเกรเดียนต์ (ได้แก่ <code>gr.fn</code> ),
ตัวแปรอยู่ <code>\$convergence</code>	บอกผลการทำงานของอัลกอริทึม
	ค่า <code>\$convergence</code> เป็น 0 หมายถึงอัลกอริทึมลู่เข้า,
และตัวแปรอยู่ <code>\$message</code>	สำหรับหมายเหตุเพิ่มเติมจากอัลกอริทึม.

รายละเอียดการใช้งานของ `optim` สามารถศึกษาเพิ่มเติมได้จาก `help(optim)`.

รายการ 7.2: ตัวอย่างการใช้ฟังชัน `optim`

---

```

1 ob.fn <- function(x) {
2   (x[1] - 8)^2 + (x[2] - 4)^2
3 }
```

```

4 gr.fn <- function(x){ c(2*(x[1]-8), 2*(x[2]-4)) }
5
6 x0 = c(0,0)
7 r <- optim(x0, ob.fn, gr.fn, method='BFGS')

```

---

วิธีบีเอฟจีเอสสามารถนำไปแทนวิธีลงเกรเดียนต์สำหรับการฝึกโครงข่ายประสาทเทียม ได้โดยการตัดแปลงโค้ดสำหรับการฝึกโครงข่าย ดังแสดงในรายการ 7.3. เมื่อเปรียบเทียบกับโค้ดในรายการ 6.2 จะเห็นว่าโค้ดในในรายการ 7.3 ถูกจัดเป็นสัดเป็นส่วนมากกว่า โดยแยกโค้ดสำหรับฟังชั่นจุดประสงค์ (ob.fn) และเกรเดียนต์ (gr.fn) ออกไปอย่างชัดเจน. สังเกตุว่า ตัวอย่างนี้ยังคงใช้วิธีการกระจายย้อนกลับ (Backpropagation) อよ. วิธีการกระจายย้อนกลับเป็นวิธีที่ใช้ทางเดียนต์ และวิธีบีเอฟจีเอสก็ยังต้องการค่าเกรเดียนต์อよ.

รายการ 7.3: ฟังชั่นฝึกโครงข่ายประสาทเทียมโดยใช้ฟังชั่น optim สำหรับการหาค่าถดถอย

---

```

1 nnTrain.optim <- function(X, T, nHiddens, net=NULL, wmax=0.1, ...){
2
3   D <- nrow(X)
4   N <- ncol(X)
5   K <- nrow(T)
6   M <- nHiddens
7
8   #####
9   ## initialize weights
10 #####
11 if (is.null(net)) {
12   ## Initialize weights
13   W1 <- matrix(runif(M*(1+D),-wmax,wmax),M,1+D)
14   W2 <- matrix(runif(K*(1+M),-wmax,wmax),K,1+M)
15   net <- list(W1=W1, W2=W2)
16 } else {
17   W1 <- net$W1
18   W2 <- net$W2
19 }#if
20
21 #####
22 ## Define objective function
23 #####
24 ob.fn <- function(ws){
25
26   cost = 0.5 * sum( (nnOutput(pack.w(ws), X) - T)^2 )
27
28   return(cost)
29 }##end ob.fn

```

```

30
31 ##### Define gradient function #####
32 ## Define gradient function
33 #####
34 gr.fn <- function(ws){
35
36     net = pack.w(ws)
37     W1 = net$W1
38     W2 = net$W2
39
40     dotX <- rbind(1,X)
41
42     ## (1) Forward propagation
43     Z <- sigmoid(W1 %*% dotX)
44     dotZ <- rbind(1,Z)
45     A <- W2 %*% dotZ
46     Y <- A      ## regression output
47
48     ## (2) Evaluate output delta
49     DELTA2 <- Y - T
50
51     ## (3) Backpropagate errors
52     S <- t(W2[,-1,drop=FALSE]) %*% DELTA2
53     DELTA1 <- dsigmoid(Z)*S
54
55     ## (4) Evaluate derivatives
56     dE2 <- DELTA2 %*% t(dotZ)
57     dE1 <- DELTA1 %*% t(dotX)
58
59     dws <- c(dE1, dE2)
60
61     return(dws)
62 }##end gr.fn
63
64 #####
65 ## arg min ob.fn
66 #####
67
68 ws <- unpack.w(net)
69 res <- optim(ws, ob.fn, gr.fn, ...)
70

```

```
71 return(c(res, list(net=pack.w(res$par, D, M, K))))
72 }##end nnTrain.optim
```

อีกจุดที่สำคัญคือ ค่าน้ำหนักของโครงข่ายประสาทเทียมแบ่งเป็นชุดตามชั้นของโครงข่าย. บทนี้ใช้ตัวอย่างของโครงข่ายประสาทเทียมสองชั้น ซึ่งมีค่าน้ำหนัก 2 ชุด ได้แก่ ตัวแปร  $W_1$  (ขนาด  $M \times (1+D)$ ) และ ตัวแปร  $W_2$  (ขนาด  $K \times (1+M)$ ). แต่พึงชี้น  $optim$  รับค่าตัวแปรชุดเดียว (และตัวแปรนี้ต้องอยู่ในรูปเวคเตอร์ด้วย). ดังนั้นตัวแปร  $W_1$  และ  $W_2$  จะถูกจัดรูปแบบไว้ในตัวแปร  $ws$  (ที่เป็นเวคเตอร์ มีความยาว  $M \times (1+D) + K \times (1+M)$ ).

โค้ดในบรรทัด 68 ทำงานจัดรูปแบบตัวแปรออกเป็น  $ws$  ด้วยฟังชัน  $unpack.w$ . และหลังจากรันฟังชัน  $optim$  เสร็จ โค้ดในบรรทัด 71 ก็จัดรูปแบบตัวแปรกลับไปเป็น  $W_1$  และ  $W_2$  อีกรershing ด้วยฟังชัน  $pack.w$ . สังเกตุ ที่บรรทัด 59 ค่าเกรเดียนต์ 2 ชุด ได้แก่ ตัวแปร  $dE1$  และ ตัวแปร  $dE2$  สำหรับ  $\nabla_{w^{(1)}} E$  และ  $\nabla_{w^{(2)}} E$  ก็ต้องถูกจัดรูปแบบออกเป็นตัวแปร  $dws$  เพื่อสัมพันธ์กับตัวแปร  $ws$ . โค้ดของฟังชัน  $unpack.w$  และ  $pack.w$  แสดงในรายการ 7.4.

#### รายการ 7.4: ฟังชัน $pack.w$ และ $unpack.w$

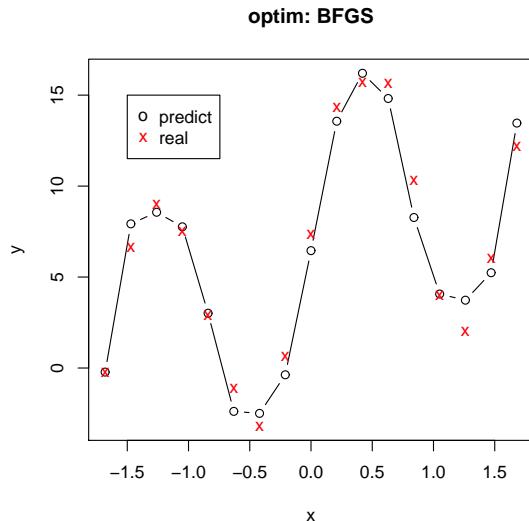
```
1 pack.w <- function(ws, D, M, K){
2   ## W1: M x (1+D)
3   ## W2: K x (1+M)
4
5   W1 <- matrix(ws[1:(M*(1+D))], M, 1+D)
6   W2 <- matrix(ws[-1:-(M*(1+D))], K, 1+M)
7
8   list(W1=W1, W2=W2)
9 }## end pack.w
10
11 unpack.w <- function(net){
12   ws <- c(net$W1, net$W2)
13 }## end unpack.w
```

โค้ด  $train.optim$  (รายการ 7.3) เป็นโค้ดสำหรับงานการหาค่าลดตอนย. หัวข้อนี้แสดงตัวอย่างง่ายๆ (หัวข้อ 6.1) โดยสร้างข้อมูลขึ้นมาด้วยโค้ดในรายการ 6.4. เมื่อได้ข้อมูลสำหรับฝึกหัดและทดสอบมาแล้ว โครงข่ายประสาทเทียมก็สามารถฝึกด้วยวิธีบีเอฟจีเอส โดยรันโค้ดดังนี้

```
res <- nnTrain.optim(train.Xn, train.T, nHiddens=20,
method = "BFGS", hessian = TRUE, control=list(maxit=3000))
```

โดยการระบุ  $hessian = TRUE$  เป็นพารามิเตอร์สำหรับวิธีบีเอฟจีเอสที่สามารถใส่ไว้ได้<sup>1</sup> และตัวอย่างนี้ใช้ฟังชันทำงานได้กับการตั้งค่านี้.

<sup>1</sup> คำอธิบายจาก  $help(optim)$  ไม่ได้ระบุรายละเอียดและคำอธิบายที่เพียงพอสำหรับ  $hessian = TRUE$ . แต่จากการทดลองใช้ฟังชันทำงานได้กับการตั้งค่านี้.



รูปที่ 7.3: ผลการทำนายจากโครงข่ายประสาทเทียมที่ฝึกด้วยวิธีบีเอฟจีอสเปรียบเทียบกับค่าจริง.

`control=list(maxit=3000)` เพื่อกำหนดให้วิธีบีเอฟจีอสสามารถติดต่อได้สูงสุดถึงแค่ 3,000 รอบ.

เข่นเดียวกับหัวข้อ 6.1 ผลที่ได้สามารถนำมาตรวจสอบได้จากการคำนวณ RMSE<sup>2</sup> ซึ่งคือ  $SE = \text{sum}((y - \text{test.T})^2)$  หรือค่าผิดพลาดของ rmse =  $\sqrt{\text{mean}((y - \text{test.T})^2)}$  รูป 7.3 แสดงค่าที่ทำนายจากโครงข่ายประสาทเทียมที่ฝึกแล้วด้วยวิธีบีเอฟจีอส ซึ่งคาดเดาดังนี้

```

y <- nnOutput(res$net, test.Xn)

pmn <- min( c(y, test.T) )
pmx <- max( c(y, test.T) )
plot(test.Xn, y, ylim=c(pmn,pmx), type='b', main='optim: BFGS',
      xlab='x', ylab='y')
points(test.Xn, test.T, pch='x', col='red')
legend(-1.5,15, c('predict', 'real'),
      pch=c('o','x'), col=c('black', 'red'))

```

## 7.3 การฝึกที่มีประสิทธิภาพมากขึ้นด้วยวิธีอสซีจี

มาร์ติน โมลเลอร์ สร้างวิธีสเกลต์คอนจูเกตเกรเดียนต์[54] หรือวิธีอสซีจี (Scaled Conjugate Gradient, ชื่อย่อ SCG) สำหรับการฝึกโครงข่ายประสาทเทียมที่มีประสิทธิภาพ และช่วยลดความอ่อนไหวจากพารามิเตอร์ของอัลกอริทึมที่ผูกเชื่อมโยงระบุลง. วิธีอสซีจีพัฒนามาจากวิธีคอนจูเกตเกรเดียนต์ (Conjugate Gradient).

<sup>2</sup>ค่าฟังชันจุดประสงค์จะเป็น  $0.5 * SE$

ณ เวลาที่โมลเลอร์พัฒนาวิธีอสซีจี วิธีนี้ในตระกูลวิธีคอนจูเกตเกรเดียนต์ที่นิยมนำมาใช้ในการฝึกโครงข่ายประสาทเทียม ก็คือวิธีคอนจูเกตเกรเดียนต์กับการค้นหาตามเลี้น (Conjugate Gradient with Line Search, คำย่อ CGL) ซึ่งทิศทางของการปรับค่าน้ำหนักจะใช้ทิศทางของคอนจูเกตเกรเดียนต์ และใช้วิธีการค้นหาตามเลี้นเพื่อหาค่าขนาดก้าวที่เหมาะสม. เมื่อเปรียบเทียบกับทิศทางของเกรเดียนต์ (ที่วิธีลงเกรเดียนต์ใช้) ทิศทางของคอนจูเกตเกรเดียนต์จะช่วยเพิ่มประสิทธิภาพในการปรับค่าน้ำหนัก ช่วยให้ปรับเข้าสู่ค่าตัวที่ทำน้อยที่สุดได้เร็วขึ้น ด้วยการลดพัฒกรรมซิกแซก (ซึ่งคือการขยับค่าตัวแบบตัดสินใจ ในลักษณะส่ายไปมาในการเข้าหาค่าที่ต้องการของวิธีลงเกรเดียนต์ (ดูรูป 2.17 ประกอบ). ในขณะที่การใช้วิธีการค้นหาตามเลี้น<sup>3</sup> ช่วยหาค่าขนาดก้าวที่เหมาะสม สามารถช่วยลดจำนวนรอบฝึกลงได้ แต่ก็กลับเพิ่มการคำนวณในแต่ละรอบฝึกขึ้นมาก จากการคำนวณที่ใช้ในวิธีการค้นหาตามเลี้นเอง.

กล่าวโดยย่อคือ จากพื้นฐานของวิธีคอนจูเกตเกรเดียนต์กับการค้นหาตามเลี้น โมลเลอร์ใช้การปรับขนาด (scale) ช่วยในการคำนวณค่าของขนาดก้าว แทนการใช้วิธีการค้นหาตามเลี้น โดย อาศัยคุณสมบัติทางคณิตศาสตร์มาช่วยทำการประมาณ เพื่อไม่ต้องคำนวณค่าอนุพันธ์อันดับสอง(ไฮเซ็น)โดยตรง.<sup>4</sup> ผลคือ ได้วิธีอสซีจี ซึ่งมีประสิทธิภาพมากกว่าวิธีคอนจูเกตเกรเดียนต์กับการค้นหาตามเลี้น และวิธีอสซีจีเป็นหนึ่งในวิธีที่มีประสิทธิภาพมากที่สุดในการฝึกโครงข่ายประสาทเทียม โดยเฉพาะ สำหรับโครงข่ายประสาทเทียมขนาดใหญ่ (เช่น มีจำนวนค่าน้ำหนักเป็นพันๆค่า)[77]. ทฤษฎีอธิบายการทำงานของวิธีคอนจูเกตเกรเดียนต์ สามารถศึกษาเพิ่มเติมได้จากทำรายการหาค่าตัวที่สุดเบื้องต้นของช่องและเขต[18]. รายละเอียดการพัฒนาของวิธีอสซีจี ผู้อ่านที่สนใจสามารถศึกษาเพิ่มเติมได้จากบทความของโมลเลอร์เอง [54].

### 7.3.1 ตัวอย่างการใช้วิธีอสซีจีแทนวิธีลงเกรเดียนต์ในการฝึก

เนื่องจากอาร์โพรเจคไม่มีวิธีอสซีจีมาให้ ตัวอย่างนี้แสดงโค้ดวิธีอสซีจี จากอัลกอริทึมในบทความของโมลเลอร์[54] ดังที่ได้ยกมาแสดงในตาราง 7.1. โมลเลอร์ใช้สัญญาณลักษณ์ ~ เพื่อ แทนตัวแปรที่เป็นเวคเตอร์: ตัวแปรที่สำคัญ ได้แก่  $\tilde{p}$  แทนค่าน้ำหนักที่ต้องการหา เพื่อให้ฟังชันจุดประสงค์  $E(\tilde{p})$  มีค่าน้อยที่สุด,  $\tilde{r}$  แทนทิศทางลงเกรเดียนต์,  $\tilde{p}$  แทนทิศทางที่จะปรับค่าน้ำหนัก,  $\alpha$  แทนค่าขนาดก้าว,  $\Delta$  ใช้เพื่อทดสอบว่าค่าน้ำหนักที่ปรับช่วยลดค่าของฟังชันจุดประสงค์ได้หรือไม่, ตัวห้อยระบุรอบฝึก, และตัวแปร  $k$  เป็นตัวชี้ของรอบฝึก.

ขั้นตอนแรกของอัลกอริทึมอสซีจี คือ การกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ของอัลกอริทึม รวมถึง คำนวณทิศทางลงเกรเดียนต์  $\tilde{r}_1$  และทิศทางสำหรับการปรับค่า  $\tilde{p}_1$ . ขั้นตอนที่ 2-4 ทำการปรับสเกล เพื่อ ให้ไฮเซ็น  $E''(\tilde{p}_k)$  มีคุณสมบัติที่ต้องการ นั่นคือ เมตริกซ์ไฮเซ็นเป็นบวกแน่นอน (Positive Definite). คุณสมบัติบวกแน่นอนของไฮเซ็น เป็นสิ่งสำคัญที่จะทำให้ทิศทางของคอนจูเกตเกรเดียนต์ (Conjugate

<sup>3</sup> วิธีการค้นหาตามเลี้น คือ วิธีการหาค่าตัวที่ทำน้อยที่สุดสำหรับปัญหาตัวแปรมิติเดียว. หัวข้อ 2.3 อภิปรายวิธีการค้นหาแบบช่วงทองคำ ซึ่งเป็นวิธีการค้นหาตามเลี้นวิธีหนึ่ง.

<sup>4</sup> ค่าไฮเซ็นเป็นกุญแจที่สำคัญที่ใช้ในการเลือกค่าขนาดก้าวที่เหมาะสม เมื่อไม่ใช้วิธีการค้นหาตามเลี้น. แต่การคำนวณค่าไฮเซ็นโดยตรงจะใช้การคำนวณมาก. ดังนั้นหากสามารถประมาณค่าไฮเซ็นได้ดี โดยใช้การคำนวณไม่มาก ก็เท่ากับผ่านอุปสรรคสำคัญในการหาขนาดก้าวที่เหมาะสมไปได้.

ตารางที่ 7.1: อัลกอริทึมของวิธีเอสซีจี[54] โดย  $\tilde{w}$  แทนเวคเตอร์ของค่าน้ำหนัก,  $E(\tilde{w})$  แทนค่าผิดพลาดของการทำนาย (ฟังชันจุดประสงค์), และ  $E'(\tilde{w})$  แทนเกรเดียนต์.

1. Choose weight vector  $\tilde{w}_1$  and scalars  $0 < \sigma \leq 10^{-4}$ ,  $0 < \lambda_1 \leq 10^{-6}$ ,  $\bar{\lambda}_1 = 0$ .  
Set  $\tilde{p}_1 = \tilde{r}_1 = -E'(\tilde{w}_1)$ ,  $k = 1$  and success = true.
2. If success = true, then calculate second order information:  

$$\sigma_k = \sigma / |\tilde{p}_k|,$$

$$\tilde{s}_k = (E'(\tilde{w}_k + \sigma_k \tilde{p}_k) - E'(\tilde{w}_k)) / \sigma_k,$$

$$\delta_k = \tilde{p}_k^T \tilde{s}_k.$$
3. Scale  $\delta_k$ :  $\delta_k = \delta_k + (\lambda_k - \bar{\lambda}_k) |\tilde{p}_k|^2$ .
4. If  $\delta_k \leq 0$  then make the Hessian matrix positive definite:  

$$\bar{\lambda}_k = 2(\lambda_k - \delta_k / |\tilde{p}_k|^2),$$

$$\delta_k = -\delta_k + \lambda_k |\tilde{p}_k|^2,$$

$$\lambda_k = \bar{\lambda}_k.$$
5. Calculate step size:  

$$\mu = \tilde{p}_k^T \tilde{r}_k,$$

$$\alpha_k = \mu_k / \delta_k.$$
6. Calculate the comparison parameter:  

$$\Delta_k = 2\delta_k [E(\tilde{w}_k) - E(\tilde{w}_k + \alpha_k \tilde{p}_k)] / \mu_k^2.$$
7. If  $\Delta_k \geq 0$  then a successful reduction in error can be made:  

$$\tilde{w}_{k+1} = \tilde{w}_k + \alpha_k \tilde{p}_k,$$

$$\tilde{r}_{k+1} = -E'(\tilde{w}_{k+1}),$$

$$\bar{\lambda}_k = 0, \text{ success} = \text{true}.$$
If  $k \bmod N = 0$  then restart algorithm:  

$$\tilde{p}_{k+1} = \tilde{r}_{k+1}$$
else:  

$$\beta_k = (|\tilde{r}_{k+1}|^2 - \tilde{r}_{k+1}^T \tilde{r}_k) / \mu_k,$$

$$\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k \tilde{p}_k.$$
If  $\Delta_k \geq 0.75$ , then reduce the scale parameter:  

$$\lambda_k = \frac{1}{4} \lambda_k.$$
else:  

$$\bar{\lambda}_k = \lambda_k,$$
success = false.
8. If  $\Delta_k < 0.25$ , then increase the scale parameter:  

$$\lambda_k = \lambda_k + (\delta_k(1 - \Delta_k) / |\tilde{p}_k|^2).$$
9. If the steepest descent direction  $\tilde{r}_k \neq \tilde{0}$ , then set  $k = k + 1$  and go to 2  
else terminate and return  $\tilde{w}_{k+1}$  as the desired minimum.

Gradient) ทำงานได้ผล. คุณสมบัติของนอนนี้ตรวจสอบได้จากตัวแปร  $\delta_k > 0$ , ซึ่ง  $\delta_k \approx \tilde{p}_k^T E''(\tilde{w}_k) \tilde{p}_k$ . ขั้นตอนที่ 4 จะปรับแก้ค่าสเกล ในกรณีที่คุณสมบัติของเขียนยังไม่เป็นบวกแน่นอน. ค่าพารามิเตอร์  $\lambda_k$  มีผลต่อขนาดก้าว  $\alpha_k$  โดย  $\lambda_k$  มีค่าสูงมาก ค่าขนาดก้าว  $\alpha_k$  จะยิ่งมีค่าน้อย. ขั้นตอนที่ 5 คำนวณค่าขนาดก้าว  $\alpha_k$ . ขั้นตอนที่ 6-7 คำนวณค่า  $\Delta_k$  เพื่อตรวจสอบว่า ค่าน้ำหนักที่จะปรับใหม่  $\tilde{w}_k + \alpha_k \tilde{p}_k$  จะช่วยลดค่าฟังชันจุดประสงค์ได้หรือไม่. ถ้าค่าใหม่ดี (ลดค่าฟังชันจุดประสงค์ได้) ก็ปรับค่าน้ำหนัก  $\tilde{w}_{k+1}$  เป็นค่าใหม่. ในขั้นตอนที่ 7 มีการตรวจสอบ  $k \bmod N$  เพื่อที่จะปรับตั้งให้อัลกอริทึมกลับเริ่มต้นเป็นทิศทางลงเกรเดียโนต์ในทุกๆ  $N$  รอบฝึก. ถ้าไม่อย่างนั้น ก็ใช้ทิศทางคอนจูเกตเกรเดียนต์. ขั้นตอนที่ 8 เป็นการปรับสเกลตามผลการทำงานของอัลกอริทึม. ขั้นตอนที่ 9 ตรวจสอบเพื่อการทำซ้ำหรือการจบการทำซ้ำ.

**หมายเหตุ** งานช่วงแรกของโมลเลอร์ (Preprint 13 Nov 1990) ใช้สูตรในการลด/เพิ่มสเกล ที่ไม่ซับซ้อนเท่าอัลกอริทึมนั้นนำเสนอในบทความปี ค.ศ. 1993 [54] ดังนี้

- ในขั้นตอนที่ 7,  
if  $\Delta_k \geq 0.75$ , then reduce the scale parameter:  $\lambda_k = \frac{1}{2}\lambda_k$
- ขั้นตอนที่ 8:  
if  $\Delta_k < 0.25$ , then increase the scale parameter:  $\lambda_k = 4\lambda_k$

และโค้ดของเน็ตแลป (Netlab[55]) เวอร์ชัน 3.2 ก็ทำอัลกอริทึมเอสซีจี ตามวิธีปรับสเกลง่ายๆนี้ และมีการจำกัดค่าสูงสุดและต่ำสุดของ  $\lambda_k$  ไว้ด้วยเพื่อรักษาเสถียรภาพ ได้แก่  $10^{-15} \leq \lambda_k \leq 10^{100}$ .

### 7.3.1.1 โค้ดสำหรับวิธีเอสซีจี

รายการ 7.5 แสดงอาร์โค้ดสำหรับอัลกอริทึมเอสซีจี. ฟังชัน `scg` รับอาร์กูเมนต์ดังนี้

- อาร์กูเมนต์ `df` สำหรับ ฟังชันเกรเดียนต์ของฟังชันจุดประสงค์<sup>5</sup>
- อาร์กูเมนต์ `f` สำหรับ ฟังชันจุดประสงค์
- อาร์กูเมนต์ `w1` สำหรับ ค่าเริ่มต้นสำหรับค่าน้ำหนักของโครงข่าย
- อาร์กูเมนต์ `prob.info` สำหรับ รับค่าเฉพาะสำหรับปัญหา  
อาทิ จำนวนมิติของอินพุตโครงข่าย `D`, จำนวนหน่วยช่อง `M`, จำนวนเอ้าต์พุตของโครงข่าย `K`, อินพุตที่ใช้ปิกโครงข่าย `X`, เอาต์พุตที่ใช้ปิกโครงข่าย `T`, และชนิดของงานที่โครงข่ายทำอยู่ `nntype` (ค่าที่รับคือ “regression”, “biclass”, หรือ “multiclass” สำหรับปัญหา การหาค่าถดถอย, การจำแนกกลุ่ม, หรือ การจำแนกกลุ่มแบบหลายกลุ่ม ตามลำดับ)

<sup>5</sup> อาร์โปรเจคสามารถรับฟังชันเป็นอาร์กูเมนต์ได้.

- อาร์กูเมนต์ `term.fn` สำหรับ ฟังชันตรวจสอบการฝึก
- อาร์กูเมนต์ `term.info` ใช้ประกอบกับฟังชันตรวจสอบการฝึก
- อาร์กูเมนต์ `MaxN` สำหรับ กำหนดจำนวนรอบฝึก หากไม่จบการฝึกก่อนกำหนด เช่น การทำการหยุดก่อนกำหนด (ทำโดยใช้กลไกของ `term.fn`)
- อาร์กูเมนต์ `sigma` และ `lambda1` เป็น พารามิเตอร์  $\sigma$  และ  $\lambda_1$  ตามลำดับ ซึ่งมีผลเลือร์แน่นกว่า โดยทั่วไป เพียงกำหนดให้  $0 < \sigma \leq 10^{-4}$  และ  $0 < \lambda_1 \leq 10^{-6}$  ก็พอ
- อาร์กูเมนต์ `log` เป็นตระรากที่ระบุว่าจะให้คำตอบของมาเป็นเฉพาะค่าน้ำหนัก (`log` เป็น `FALSE`) หรือจะให้รายละเอียดอื่นๆของมาด้วย (`log` เป็น `TRUE`)
- อาร์กูเมนต์ `doplot` เป็นตระรากเพื่อระบุให้วัดกราฟแสดงค่าผิดพลาดต่อรอบฝึกของมาด้วยหรือไม่

รายการ 7.5: โค้ดวิธีเอสซีจี

---

```

1 scg <- function(df, f, w1, prob.info=NULL, term.fn=nnTermination,
2                     term.info=NULL, MaxN=1000, sigma=5.0e-5,
3                     lambda1=5.0e-7, log=FALSE, doplot=TRUE){
4
5 ##browser()
6
7 term.results <- NULL
8 w1 <- as.matrix(w1)
9
10 ## Step 1. ##
11
12 wk <- w1
13 lambdak <- lambda1
14 lambdabar <- 0
15 rk <- -df(wk, prob.info)
16 pk <- rk
17 success <- TRUE
18
19 NR <- nrow(wk) ## size of wk: NR x 1
20
21 logs <- rbind(0, success, 0, 0, lambdabar, lambdak, 0, 0, 0, 0, 0)
22 rownames(logs) <- c("k", "success", "sigmak", "deltak", "lambdabar", "lambdak", <-
23   "muk", "alphak", "DELTA", "Ew", "Ewplus")

```

```

23
24 wtrace <- rbind(0, wk)
25
26 ## Step 2. ##
27
28 for(k in 1:MaxN){
29
30   pkSq <- t(pk) %*% pk
31   norm.pk <- sqrt(pkSq)
32
33   if(success){
34     sigmak <- as.numeric(sigma/norm.pk)
35     dEplus <- df(wk + sigmak*pk, prob.info)
36     dE <- df(wk, prob.info)
37     sk <- (dEplus - dE)/sigmak
38     deltak <- as.numeric(t(pk) %*% sk)
39   }## if(success)
40
41 ## Step 3. ##
42
43   deltak <- deltak + (lambda - lambdaBar) * pkSq
44
45 ## Step 4. ##
46
47   if(deltak <= 0){
48     lambdaBar <- 2*(lambda - deltak/pkSq)
49     deltak <- -deltak + lambda*pkSq
50     lambda <- lambdaBar
51   }## if(deltak <= 0)
52
53 ## Step 5. ##
54
55   muk <- t(pk) %*% rk
56   alphak <- as.numeric(muk/deltak)
57
58 ## Step 6. ##
59
60   wnew <- wk + alphak*pk
61   Eplus <- f(wnew, prob.info)
62   E <- f(wk, prob.info)
63

```

```

64 if(is.infinite(Eplus)){
65   cat('\nEplus is Inf; alphak =', alphak, '; muk =', muk, '\n')
66   browser()
67   ## My suggestion (Aug 23rd, 2014):
68   ## 1. Check muk: it must be > 0 (conjugate direction shold be within 90 ←
69   ##       degree from -grad)
70   ## 2. deltak should be > 0, because we corrected it in step 4.
71   ## 3. Then, alphak should be > 0.
72   ## 4. If alphak is too large, reduce alphak.
73   ##     For example, downsize alphak by magnitude of 10, alphak = alphak/10.
74   ##     If SCG works correctly, alphak should be scaled automatically:
75   ##     Lambdak will be increased to reduce alphak.
76   ##
77   ## Quick Fix: reduce alphak and restart at step 6
78   ##               or increase Lambdak and restart at step 3
79 }##end if
80
81
82 ## Step 7. ##
83
84 if(DELTA >= 0){
85   rnew <- -df(wnew, prob.info)    ## gradient descent direction
86   lambdabar <- 0
87   success <- TRUE
88
89   if(k %% NR == 0){
90     pnew <- rnew
91   }else{
92     beta <- as.numeric( (t(rnew) %*% rnew - t(rnew) %*% rk) /muk )
93     pnew <- rnew + beta*pk
94   }## end if(k %% NR == 0)
95
96   if(DELTA >=0.75){
97     lambdak <- lambdak/4
98   }## end if(DELTA >=0.75)
99
100  wk <- wnew
101  rk <- rnew
102  pk <- pnew
103

```

```

104     }else{## DELTA < 0
105         lambdabar ← lambdak
106         success ← FALSE
107     }## end if(DELTA >= 0)
108
109     ## Step 8. ##
110
111     if(DELTA < 0.25){
112         lambdak ← lambdak + deltak*(1 - DELTA)/pkSq
113     }## end if(DELTA < 0.25)
114
115     ## Step 9. ##
116
117     logs ← cbind(logs, rbind(k, success, sigmak, deltak, lambdabar, lambdak, muk←
118                             , alphak, DELTA, E, Eplus))
119     wtrace ← cbind(wtrace, rbind(k, wk))
120     term.results ← term.fn(wk, term.results, term.info, doplot=doplot, E.logs←
121                             logs[10,-1])
122
123     if (term.results$boolean){
124         ## Terminate
125         if(log){ return( list(logs=logs, wtrace=wtrace, w=wk, term.results=term.←
126                     results) ) } else { return(wk) }
127     }## end if
128
129 }## end for(k in 1:MaxN)
130
131 print("Reach maximum iterations")
132 if(log){ return( list(logs=logs, wtrace=wtrace, w=wk, term.results=term.results←
133                         , note="reach max iterations") ) }
134
135 return(wk)
136 }## end scg

```

---

โค้ดในรายการ 7.5 ระบุขั้นตอนตามอัลกอริทึมในตาราง 7.1 ในคอมเมนต์ เช่น ## Step 1. ## บอกจุดเริ่มของโค้ด ที่ทำงานตามขั้นตอนที่ 1 ในตาราง 7.1. ชื่อตัวแปรในโค้ดก็ใช้ตามชื่อตัวแปรในตาราง 7.1 เช่น ตัวแปร wk สำหรับตัวแปร  $\tilde{p}_k$ . ขั้นตอนที่ 2 ในโค้ด ค่า  $|\tilde{p}_k|^2$  ถูกคำนวณและเก็บไว้ใน ตัวแปร pkSq. ตัวแปร dE และตัวแปร dEplus เก็บค่า  $E'(\tilde{p}_k)$  และ  $E'(\tilde{p}_k + \sigma_k \tilde{p}_k)$  ตามลำดับ. โค้ดขั้นตอนที่ 3-5 ทำตามอัลกอริทึมในตาราง 7.1 อย่างตรงไปตรงมา. ขั้นตอนที่ 6 มีโค้ดที่เพิ่มพิเศษขึ้นมา เพื่อแก้ปัญหาในทางปฏิบัติ. นั่นคือ การตรวจสอบว่า Eplus มีค่าเป็นอนันต์หรือไม่. หาก Eplus มีค่าไม่เป็นอนันต์ โปรแกรม

ก็จะดำเนินอัลกอริทึ่มต่อไป แต่ถ้า  $Eplus$  มีค่าเป็นอนันต์ โปรแกรมจะหยุดทำงานด้วยคำสั่ง `browser()` เพื่อให้เราสามารถตรวจสอบสาเหตุ และตัดสินใจจัดการต่อไปได้.

ปัญหาที่พบ จนต้องเพิ่มโค้ดเพื่อตรวจสอบนี้เข้าไป คือ การที่ค่าของ  $\text{alphak}$  ใหญ่เกินไป. ในทางทฤษฎีแล้ว หากค่า  $\alpha_k$  ใหญ่เกินไป ก็จะทำให้ค่า  $E(\tilde{w}_k + \alpha_k \tilde{p}_k) > E(\tilde{w}_k)$ . กรณีนี้ จะถูกตรวจสอบด้วยค่า  $\Delta_k < 0$ , และอัลกอริทึ่มก็จะปรับสเกล เพื่อแก้ไขค่าของ  $\alpha_k$  ต่อไปได้เอง โดยเราไม่ต้องเข้าไปยุ่ง. นั่นคือทฤษฎี แต่ในทางปฏิบัติ หาก  $\alpha_k$  ใหญ่เกินไปมากๆ จนทำให้ค่า  $E(\tilde{w}_k + \alpha_k \tilde{p}_k)$  ซึ่งแทนด้วยตัวแปร  $Eplus$  ใหญ่กว่า  $1.797693 \times 10^{308}$  (สำหรับอาร์โปรดัก เวอร์ชัน 32 บิต)<sup>6</sup> แล้วอาร์โปรดักจะถือว่า ค่าอนันต์เป็นค่าอนันต์ (`Inf`), ซึ่งส่งผลต่อมาทำให้โปรแกรมไม่สามารถรันต่อไปได้.

คำแนะนำ หากเกิดกรณีนี้ขึ้น<sup>7</sup> ให้ลองสำรวจดูก่อนว่าเกิดอะไรขึ้น. และหากพบว่าค่าของ  $\text{alphak}$  มีขนาดใหญ่เกินไป ก็ให้ลองปรับขนาดของ  $\text{alphak}$  ให้เล็กลง เสร็จแล้ว ทำการคำนวณโค้ดข้างล่างนี้ใหม่

```
wnew <- wk + alphak*pk
Eplus <- f(wnew, prob.info)
```

แล้วลองตรวจสอบค่า  $Eplus$  ว่าน้อยกว่า  $E$  หรือยัง. ถ้า  $Eplus$  ยังมากเกินไป ก็ให้ลองลดขนาดของ  $\text{alphak}$  ลงอีก. เพื่อความสะดวก แนะนำว่าควรลดลงทีละ 10 เท่า จนกว่าค่า  $Eplus$  จะน้อยกว่า  $E$  แล้วจึงค่อยรันต่อ. แทนที่จะหยุดโปรแกรมโดย `browser()` และให้ผู้ใช้เข้ามาจัดการ เราสามารถเขียนโค้ดที่ลดค่า  $\text{alphak}$  ตามคำแนะนำได้เลย. แต่เนื่องจาก กรณีนี้เกิดขึ้นไม่บ่อยมาก ตัวอย่างนี้จึงเลือกวิธีหยุดโปรแกรม ไว้เพื่อตรวจสอบและศึกษาเหตุการณ์ที่เกิดขึ้น.

ขั้นตอนที่ 7 และ 8 โปรแกรมทำตามอัลกอริทึ่มอย่างตรงไปตรงมา. ส่วนขั้นตอนที่ 9 เราใช้ `term_fn` เพื่อช่วยในการตรวจสอบเงื่อนไขจบการฝึก. โค้ดของฟังชันนี้ที่ใช้ประกอบ รวมถึงตัวอย่างการใช้ฟังชัน `scg` เพื่อฝึกโครงข่ายประสาทเทียม แสดงในหัวข้อ [7.3.1.2](#).

### 7.3.1.2 โค้ดฟังชันประกอบการใช้วิธีอเลฟชีจีสำหรับฝึกโครงข่ายประสาทเทียม

รายการ [7.6](#) แสดงโค้ดสำหรับฟังชันจุดประสงค์ `mse.w` ซึ่งทำการคำนวณค่าฟังชันจุดประสงค์ (แทนด้วยตัวแปร `mse`) ดังที่ได้อธิบายในหัวข้อ [5.3](#). การคำนวณค่าตัวแปร `mse` สำหรับการหาค่าถดถอยและการแจกแจงกลุ่มแบบหลายกลุ่ม จะตรงไปตรงมา (สมการ [5.20](#) และ [5.31](#)). แต่การแจกแจงกลุ่มแบบสองกลุ่ม ซึ่งใช้ฟังชันจุดประสงค์  $\sum_n \text{Cost}_n$  เมื่อ

$$\text{Cost}_n = -t_n \log(y_n) - (1 - t_n) \log(1 - y_n)$$

สามารถทำได้โดย

```
costf <- -sum(T * log(Y) + (1-T) * log(1-Y))
```

<sup>6</sup>คำสั่ง `.Machine$double.xmax` สามารถใช้ เพื่อตรวจสอบค่ามากที่สุดที่อาร์โปรดักรับได้.

<sup>7</sup>กรณีนี้อาจจะเกิดหรือไม่เกิดก็ได้. ส่วนใหญ่แล้วจะไม่เกิด

แต่เนื่องจาก ในทางปฏิบัติ เมื่อโครงข่ายประสาทเทียมทำงานได้ดีมาก เช่น หาก  $t_n = 1$  และ ค่า  $y_n$  ก็ได้เท่ากับ 1. นั่นคือทำนายได้ถูกต้องสมบูรณ์แบบ และค่า  $\text{Cost}_n$  จะเป็น 0. แต่สิ่งที่เกิดขึ้นคือ  $\text{Cost}_n = -(1)\log(1) - (1-1)\log(1-1) = 0 - 0 \cdot \log(0)$ . เทอม  $0 \cdot \log(0)$  จะทำให้เกิดสถานะการณ์  $0 \cdot (-\infty)$  ซึ่งอาร์โปรเจคจะให้ผลเป็น NaN. ค่า NaN จะส่งผลต่อมาทำให้โปรแกรมไม่สามารถรันต่อไปได้. ดังนั้นเราจึงต้องเพิ่มการตรวจสอบ และจัดการกับเหตุการณ์นี้ ดังแสดงในบรรทัดที่ 31-34. หมายเหตุ ปัญหาเหล่านี้เป็นประเด็นการคำนวณเชิงเลขในทางปฏิบัติ ซึ่งจากคณิตศาสตร์แล้วอักษอริทึมการทำงานได้ไม่มีปัญหา แต่โปรแกรมที่เขียนขึ้นหากไม่ได้คำนึงถึงประเด็นการคำนวณเชิงเลขในทางปฏิบัติแล้ว เมื่อนำไปทำงานจะมีปัญหาอย่างมาก.

---

#### รายการ 7.6: โค้ดพังชั่นจุดประสงค์เพื่อใช้กับวิธีเอสซีจี

---

```

1 mse.w <- function(ws, prob.info){
2   ## prob.info=list(D, M, K, X, T, ...
3   ## nntype ='regression'/'biclass'/'multiclass')
4   ## Note: biclass, K must be 1.
5
6   X <- prob.info$X
7   T <- prob.info$T
8   N <- ncol(X)
9   K <- prob.info$K
10
11  nntype <- 'regression' ## default
12  if(!is.null(prob.info$nntype)){
13    nntype <- prob.info$nntype
14  }
15
16  netW <- pack.w(ws, prob.info$D, prob.info$M, prob.info$K);
17  W1 <- netW$W1;
18  W2 <- netW$W2;
19
20  dotX <- rbind(1,X);
21  Z <- sigmoid(W1 %*% dotX);
22  dotZ <- rbind(1,Z);
23  A <- W2 %*% dotZ
24
25  if(nntype == 'regression'){
26    Y <- A;
27    costf <- 0.5*sum((Y - T)^2)
28  }##end if
29  if(nntype == 'biclass'){
30    Y <- 1/(1+exp(-A))

```

```

31   costn <- T * log(Y) + (1-T) * log(1-Y)
32   perfect.ids <- which(is.nan(costn))
33   costn[perfect.ids] <- 0    ## a perfect result has 0 cost.
34   costf <- -sum(costn)
35 }##end if
36 if(nntype == 'multiclass'){
37   Y <- exp(A)/matrix(colSums(exp(A)), K, N, byrow=TRUE)
38   costf <- -sum( T * log(Y) )
39 }##end if
40
41 return(costf)
42 }## end mse.w

```

---

รายการ 7.7 แสดงโค้ดพัฒนากรเดียนต์ dE.w เพื่อใช้กับวิธีเอสซีจี. ค่ากรเดียนต์ถูกคำนวณโดยใช้วิธีการแพร่กระจายย้อนกลับ (Backpropagation) ดังที่ได้อธิบายในหัวข้อ 5.4.

#### รายการ 7.7: โค้ดพัฒนากรเดียนต์เพื่อใช้กับวิธีเอสซีจี

---

```

1 ## identical to dEwCode.r (but, rename to match mseWCode02.r)
2 ## Aug 23rd, 2014.
3
4 dE.w <- function(ws, prob.info){
5   ## prob.info=list(D, M, K, X, T, nntype)
6
7   X <- prob.info$X
8   T <- prob.info$T
9   N <- ncol(X)
10  K <- prob.info$K
11
12  nntype <- 'regression'
13  if(!is.null(prob.info$nntype)){
14    nntype <- prob.info$nntype
15  }
16
17  netW <- pack.w(ws, prob.info$D, prob.info$M, prob.info$K);
18  W1 <- netW$W1;
19  W2 <- netW$W2;
20
21  ## Forward pass
22  dotX <- rbind(1,X);
23  Z <- sigmoid(W1 %*% dotX);
24  dotZ <- rbind(1,Z);

```

```

25   A <- w2 %*% dotZ
26
27   if(nntype == 'regression'){
28     Y <- A
29     ## costf <- 0.5*sum((Y - T)^2)
30   }##end if
31   if(nntype == 'biclass'){
32     Y <- 1/(1+exp(-A))
33     ## costn <- T * log(Y) + (1-T) * log(1-Y)
34     ## perfect.ids <- which(is.nan(costn))
35     ## costn[perfect.ids] <- 0    ## a perfect result has 0 cost.
36     ## costf <- -sum(costn)
37   }##end if
38   if(nntype == 'multiclass'){
39     Y <- exp(A)/matrix(colSums(exp(A)), K, N, byrow=TRUE)
40     ## costf <- -sum( T * log(Y) )
41   }##end if
42
43   ## Backward pass
44
45   DELTA2 <- ( Y - T )
46   S <- t(w2[,-1,drop=FALSE]) %*% DELTA2      # M x N
47   DELTA1 <- dsigmoid(Z)*S                      # M x N
48   dE2 <- DELTA2 %*% t(dotZ)                   # K x (1+M)
49   dE1 <- DELTA1 %*% t(dotX)                   # M x (1+D)
50
51   return(c(dE1, dE2))
52 }## end dE.w

```

---

รายการ 7.8 แสดงโค้ดฟังชันตรวจสอบการฝึก nnTermination. การแยกการตรวจสอบการฝึกออกมาเป็นฟังชันต่างหาก ช่วยเพิ่มความยืดหยุ่นในการทำเงื่อนไขจบการฝึก แต่ก็จะทำให้โปรแกรมมีความซับซ้อนขึ้น. ฟังชัน nnTermination สามารถทำเงื่อนไขจบการฝึกได้ 3 แบบ คือ

- (1) จบเมื่อ  $\text{perf} \leq \text{tol}$  นั่นคือ ฟังชันจุดประสงค์มีค่าน้อยกว่าหรือเท่ากับค่าที่ยอมรับได้ (พารามิเตอร์ tol)
- (2) จบเมื่อ  $\text{grad.mse} \leq \text{min.grad}$  นั่นคือ เกรเดียนต์มีค่าน้อยกว่าหรือเท่ากับค่าเกรเดียนต์ที่ยอมรับได้ ( $\text{nnTermination}$  กำหนด ดีฟอลต์ ของค่าเกรเดียนต์ที่ยอมรับได้  $\text{min.grad}$  คือ  $10^{-12}$ )

- และ (3) หากทำวा�ลิเดชั่น การฝึกสามารถจบได้เมื่อ `val.fail >= max.val.fail` นั่นคือ ค่าผิดพลาดของชุดวัลิเดชั่นที่ฝึกอยู่มีค่าแย่กว่าครั้งที่ดีที่สุด เกินจำนวนครั้งที่กำหนด.

รายการ 7.8: โค้ดฟังชันตรวจสอบการฝึกเพื่อใช้กับวิธีอสซีจี

---

```

1 nnTermination <- function(xk, term.data=NULL, term.info=NULL,
2                               tol=0, min.grad=1e-12, max.val.fail=8,
3                               doplot=FALSE, E.logs=NULL, nntype='regression'){
4
5   if(is.null(term.data)){
6     term.count <- 0
7     val.fail <- 0
8     val.old.perf <- Inf
9     val.best.perf <- Inf
10    val.best.net <- NULL
11    val.best.term.count <- 0
12  } else {
13    term.count <- term.data$term.count
14    val.fail <- term.data$val.fail
15    val.old.perf <- term.data$val.old.perf
16    val.best.perf <- term.data$val.best.perf
17    val.best.net <- term.data$val.best.net
18    val.best.term.count <- term.data$val.best.term.count
19  }#end if
20
21  term.count <- term.count + 1
22
23  grad.mse <- mean( dE.w(xk, prob.info=term.info$prob.info)^2 )
24  perf <- mse.w(xk, prob.info=term.info$prob.info)
25
26  term.boolean <- FALSE
27
28  if(perf <= tol){ term.boolean <- TRUE }
29  if(grad.mse <= min.grad){ term.boolean <- TRUE }
30
31  if( !is.null(term.info$val.X) && !is.null(term.info$val.T) ){
32    val.X <- term.info$val.X
33    val.T <- term.info$val.T
34    D <- term.info$prob.info$D
35    M <- term.info$prob.info$M
36    K <- term.info$prob.info$K
37

```

```

38     net <- pack.w(xk, D, M, K)
39     val.y <- nnOutput(net, val.X, nntype=nntype)
40
41     if(nntype == 'regression'){
42         val.mse <- mean( (val.y - val.T)^2 )
43     }
44     if(nntype == 'biclass'){
45         val.cost <- val.T * log(val.y) + (1-val.T) * log(1-val.y)
46         perfect.ids <- which(is.nan(val.cost))
47         val.cost[perfect.ids] <- 0 ## perfect results has 0 costs.
48         val.mse <- -sum(val.cost)
49     }
50     if(nntype == 'multiclass'){
51         val.mse <- -sum( val.T * log(val.y) )
52     }
53
54     if(val.mse > val.old.perf){ val.fail <- val.fail + 1 }
55     if(val.fail >= max.val.fail){ term.boolean <- TRUE }
56
57     if(val.mse < val.best.perf){
58         val.best.perf <- val.mse
59         val.best.net <- net
60         val.best.term.count <- term.count
61     }#end if
62     val.old.perf <- val.mse
63
64 }#end if
65
66 if(doplot){
67     k <- length(E.logs)
68     if(k > 9 && (k %% round(k/10) == 0))
69         plot(1:length(E.logs), E.logs, xlab='epoch', ylab='E', type='l')
70 }
71
72 return(list(boolean=term.boolean, term.count=term.count,
73             val.fail=val.fail, val.old.perf=val.old.perf,
74             val.best.perf=val.best.perf, val.best.net=val.best.net,
75             val.best.term.count=val.best.term.count))
76 }# end function nnTermination

```

เพื่อช่วยให้การฝึกโครงข่ายประสาทเทียมสามารถทำได้มีประสิทธิภาพมากขึ้น เห็นได้โดยร่วมกันและวิดีโอ

เสนอวิธีเหจី-น-วิดโดร์ (Nguyen-Widrow Weight Initialization[59]) สำหรับ การกำหนดค่าเริ่มต้นของค่าน้ำหนักที่มีประสิทธิภาพมากขึ้น. รายการ 7.9 แสดงโค้ดกำหนดค่าเริ่มต้นสำหรับโครงข่ายประสาทเทียมด้วยวิธีเหจី-น-วิดโดร์.

รายการ 7.9: โค้ดกำหนดค่าเริ่มต้นสำหรับโครงข่ายประสาทเทียมด้วยวิธีเหจី-น-วิดโดร์

---

```

1 init.weights.nugyenwidrow <- function(D, M, K, activeregion=c(-4, 4)){
2
3   Wmax <- 0.7*M^(1/D);
4   Wh.rand <- matrix(runif(M*D,-1,1),M,D)
5   norm.factor <- sqrt(matrix(rowSums(Wh.rand^2),M,1)) %*% matrix(1, 1, D)
6   Wh <- Wmax * Wh.rand/norm.factor
7   bh <- Wmax*seq(-1,1,len=M)*sign(runif(M, -1, 1))
8
9   a.scale <- (activeregion[2] - activeregion[1])/2;
10  a.offset <- (activeregion[2] + activeregion[1])/2;
11
12  Wh <- Wh * a.scale
13  bh <- bh * a.scale + a.offset
14
15  W1 <- cbind(bh,Wh)
16  W2 <- matrix(runif(K*(1+M),-1,1),K,1+M)
17
18  list(W1=W1, W2=W2)
19 }## end init.weights

```

---

โค้ดทั้งหมดที่อภิปรายไปมีเพียงพอที่จะใช้ฝึกโครงข่ายประสาทเทียมด้วยวิธีเอสซีจีได้แล้ว แต่เพื่อความสะดวก โค้ดเหล่านี้จะถูกห่อไว้ในฟังชัน nnTrain.scg เพื่อให้สามารถเรียกใช้ได้สะดวก ดังแสดงในรายการ 7.10.

รายการ 7.10: โค้ดฟังชัน nnTrain.scg ซึ่งห่อรายละเอียดการฝึกโครงข่ายประสาทเทียมด้วยวิธีเอสซีจีเอาไว้เพื่อความสะดวกในการใช้งาน

---

```

1 nnTrain.scg <- function(X, T, nHiddens, net=NULL, nEpochs=500, nntype='regression'<-
2   ','
3   val.X=NULL, val.T=NULL, tol=0, min.grad=1e-12, max.val.<-
4   fail=8,
5   internal.norm=F, ...){
6   ##
7   ## val.X=NULL or val.T=NULL infers no early stopping.
8   ##
9   cat('\n* train ANN with scg on ', nntype, '\n')

```

```

9 ##########
10 ## compose termination criteria
11 #########
12
13 wrap.term <- function(xk, term.data=NULL, term.info=NULL,
14   doplot=FALSE, E.logs=NULL){
15
16   nnTermination(xk, term.data, term.info,
17     tol=tol, min.grad=min.grad, max.val.fail=max.val.fail,
18     doplot, E.logs, nntype)
19
20 }##end scg.termfn
21
22
23 #########
24 ## initialize weights
25 #########
26
27 D <- nrow(X)
28 N <- ncol(X)
29 K <- nrow(T)
30 M <- nHiddens
31
32 if (is.null(net)) {
33   ## Initialize weights
34   net <- init.weights.nugyenwidrow(D, M, K, activeregion=c(-4, 4))
35
36 }#if
37
38 ws0 <- unpack.w(net)
39
40
41 prob.info <- list(D=D, M=M, K=K, X=X, T=T,
42   nntype=nntype)
43
44 term.info <- list(prob.info=prob.info, val.X=val.X, val.T=val.T)
45
46 wsk.log <- scg(dE.w, mse.w, ws0, prob.info=prob.info,
47   term.fn=wrap.term, term.info=term.info,
48   MaxN=nEpochs, log=TRUE)
49

```

```

50   net <- pack.w(wsk.log$w, D, M, K)
51
52   if(!is.null(val.X) & !is.null(val.T)) {
53     net <- wsk.log$term.results$val.best.net;
54   }
55
56 return(c(wsk.log, list(net=net)))
57 }##end nnTrain.scg

```

---

ตัวอย่างการฝึกโครงข่ายประสาทเทียมด้วยวิธีเอสซีจี โค้ดในรายการ 7.11 แสดงรายการการฝึกโครงข่ายประสาทเทียมด้วยวิธีเอสซีจี. หมายเหตุ การทำนอร์มอลайเซชันในตัวอย่างนี้ใช้ฟังชัน `normalize` ในรายการ 7.12 ซึ่งเป็นแบบปรับค่ามากที่สุดน้อยที่สุด.

รายการ 7.11: โค้ดทดสอบการฝึกโครงข่ายประสาทเทียมด้วยวิธี SCG

```

1 f <- function (x) x + 8 * sin(x) + rnorm(length(x))
2
3 N <- 50
4 train.X <- matrix(seq(0,4*pi,len=N),1,N)
5 train.T <- f(train.X)
6 r <- normalize(train.X)
7 train.Xn <- r$norm
8
9 test.X <- matrix(seq(0,4*pi,len=round(N/3)),nrow=1)
10 test.T <- f(test.X)
11 test.Xn <- normalize(test.X,xmin=r$min,xmax=r$max)$norm
12
13 res <- nnTrain.scg(train.Xn,train.T, nHiddens=20, nEpoch=500)
14
15 y <- nnOutput(res$net, test.Xn)
16
17 pmn <- min( c(y, test.T) )
18 pmx <- max( c(y, test.T) )
19 plot(test.Xn, y, ylim=c(pmn,pmx), type='b', main='optim: BFGS',
20 xlab='x', ylab='y')
21 points(test.Xn, test.T, pch='x', col='red')
22 legend(-1.5,15, c('predict', 'real'),
23 pch=c('o','x'), col=c('black', 'red'))

```

---

รายการ 7.12: โค้ดการทำนอร์มอลайเซชันแบบปรับค่ามากที่สุดน้อยที่สุด (ดูหัวข้อ 6)

```

1 normalize <- function(X,
2   xmin=matrix(apply(X, 1, min), nrow(X), 1),
3   xmax=matrix(apply(X, 1, max), nrow(X), 1),
4   norm.min=matrix(-1, nrow(X), 1),
5   norm.max=matrix(1, nrow(X), 1) ){
6   ## X : D x N, xmin : D x 1, xmax : D x 1
7
8   D <- nrow(X);
9   N <- ncol(X);
10
11  Mmin <- matrix(xmin, D, N, byrow=FALSE);
12  Mmax <- matrix(xmax, D, N, byrow=FALSE);
13  normMmin <- matrix(norm.min, D, N, byrow=FALSE);
14  normMmax <- matrix(norm.max, D, N, byrow=FALSE);
15
16  xnorm <- (X - Mmin) * (normMmax - normMmin)/(Mmax - Mmin) + normMmin;
17
18  return(list(norm=xnorm, min=xmin, max=xmax, norm.min=norm.min, norm.max=norm.max))
19 }#end normalize

```

---

โค้ดฟังชันอื่นที่เกี่ยวข้อง nnOutput, sigmoid, dsigmoid, hard.limit, encode.OK, decode.OK, pack.w, และ unpack.w ได้อธิบายไปก่อนหน้า (รายการ 6.1, 6.8, 6.10, 6.11, และ 7.4).

ตัวอย่างการใช้ฟังชัน nnTrain.scg กับการหยุดก่อนกำหนดแสดงด้วยปัญหาการจำแนกกลุ่ม ข้อมูล ชุดรูปของลายมือเขียนตัวเลข (บทที่ 6) ดังรายการ 7.13. จากขั้นตอนการนำเข้าและเตรียมข้อมูล ในหัวข้อ 6.5.4, ตัวแปร train.X, train.T.K, validate.X, validate.T.K, test.X, และ test.T แทนอินพุต (แสดงด้วยสัญกรณ์ตาม X) และเอาต์พุต (แสดงด้วยสัญกรณ์ตาม T หรือ T.K) สำหรับชุดอย่าง สำหรับฝึก (แสดงด้วยสัญกรณ์นำ train), ชุดอย่างสำหรับвалиเดชัน (แสดงด้วยสัญกรณ์นำ validate), และ ชุดอย่างสำหรับทดสอบ (แสดงด้วยสัญกรณ์นำ test). ตัวแปร train.T.K และ validate.T.K แสดงกลุ่มในรหัสหนึ่งไปเค. ตัวแปร test.T แสดงกลุ่มด้วยอักษร ‘0’, ‘1’, ‘2’, ..., หรือ ‘9’.

รายการ 7.13: ตัวอย่างโค้ดฝึกและทดสอบโครงข่ายประสาทเทียมด้วยวิธีเอสซีจีสำหรับปัญหาข้อมูลชุดรูปของลายมือเขียนตัวเลข

---

```

1 result <- nnTrain.scg(train.X,train.T.K, nHiddens=40,
2                         nEpoch=3000, nntype='multiclass',
3                         val.X=validate.X, val.T=validate.T.K)
4
5 test.y.k <- nnOutput(result$net, test.X, nntype='multiclass')
6
7 test.y <- decode.OK(test.y.k, c('0','1','2','3','4','5','6','7','8','9'))

```

---

```
8 correct.perc <- sum(test.y == test.T)/N.test * 100
```

---

หมายเหตุ ทั้งนี้ทั้งนั้น ผู้เชี่ยวชาญทางศาสตร์การเรียนรู้ของเครื่อง เข่น แอนดรูว์ อิง[58] แนะนำว่า หากผู้ใช้ไม่ได้ชำนาญด้านการวิเคราะห์เชิงเลข (Numerical Analysis) และ การเขียนโค้ดอัลกอริทึมการหาค่า น้อยที่สุดที่ซับซ้อนเข่นี้ ไม่ควรที่จะทำเอง แต่แนะนำให้ใช้โปรแกรมสำเร็จที่มีการพัฒนาและตรวจสอบมาอย่างดีแล้ว.

## 7.4 คำแนะนำเพิ่มเติม

ตัวอย่างที่ผ่านมาอภิปรายวิธีการประยุกต์ใช้โครงข่ายประสาทเทียม รวมถึงการประเมินผลไปพอสมควร. หากประเมินผลแล้ว ผู้ฝึกโครงข่ายไม่พอใจผลการทำนายที่ได้ และหากอยากรับปรุงให้ดีขึ้น สิ่งที่อาจจะช่วยได้ คือ

- เพิ่มจำนวนข้อมูลมาใช้สำหรับการฝึก
- เลือกแคมมิติที่สำคัญบางมิตินามาใช้เป็นอินพุต
- เพิ่มลักษณะสำคัญใหม่เข้าไปในอินพุต
- เพิ่มความซับซ้อนของโมเดลขึ้น
- ลดความซับซ้อนของโมเดลลง

แต่ผู้ฝึกโมเดลควรจะลองอะไรก่อน การสุมลองแต่ละอย่างอาจใช้เวลามาก และยังอาจเพิ่มงบประมาณด้วย เช่น การออกแบบข้อมูลมาเพิ่ม (เพิ่มจำนวนจุดข้อมูล) หรือการเพิ่mlักษณะสำคัญชนิดใหม่เข้าไป (เพิ่มมิติใหม่สำหรับอินพุต). แอนดรูว์ อิง[58] แนะนำว่าผู้ฝึกโมเดลควรจะทำการทดลองง่ายๆ และใช้เล้นโค้จเรียนรู้ (Learning Curve) เป็นตัวชี้แนะ. ก่อนจะถกเรื่องการใช้เล้นโค้จเรียนรู้ มีประเด็นเรื่องไบอัส (bias) กับความแปรปัน (variance) ที่ควรจะทำความเข้าใจก่อน.

### 7.4.1 ไบอัสกับความแปรปัน

พิจารณารูป 3.10 และ 3.11 ที่ใช้พื้นที่พื้นที่ที่สามารถทำนายข้อมูล (หัวข้อ 3.1). ภาพซ้ายของรูป 3.10 แสดงกรณีที่เกิดโอลเวอร์ฟิตติ้ง ภาพกลางแสดงกรณีที่ต้องการ และภาพขวาแสดงกรณีที่เกิดอันเดอร์ฟิตติ้ง. รูป 3.11 แสดงค่าผิดพลาดจากชุดฝึกหัดและชุดทดสอบ ที่ความซับซ้อนของโมเดลต่างๆ. ความซับซ้อนของโมเดลในรูปวัดด้วย เรกุล่าเรชชันพารามิเตอร์ ซึ่งที่นั้นคือ ค่าลากของ  $\lambda$ . ถ้า  $\lambda$  มีค่ามากๆ ความซับซ้อนจะน้อย และถ้า  $\lambda$  มีค่าน้อยๆ ความซับซ้อนจะมาก. สังเกตุ ในรูป 3.11 ที่ความซับซ้อนของโมเดลที่เหมาะสม ค่าผิดพลาดของชุดทดสอบมีค่าต่ำที่สุด.

ช่วงที่ความซับซ้อนของโมเดลน้อยเกินไป ( $\lambda$  มีค่ามากเกินไป) ซึ่งคือ กรณีอันเดอร์พิทติ้ง ให้สังเกตุว่า ค่าผิดพลาดของชุดฝึกหัดจะสูง. โมเดลไม่ซับซ้อนพอจะลดค่าผิดพลาด แม้แต่ค่าผิดพลาดของข้อมูลชุดฝึกหัดลงได้. กรณีที่โมเดลทำงานได้ไม่ดีเนื่องจากความซับซ้อนน้อยเกินไปนี้ จะเรียกว่า กรณีที่มีใบอัสสูง (High Bias) ซึ่งกรณีนี้สื่อถึงอันเดอร์พิทติ้ง.

ในอีกทางหนึ่ง ช่วงที่ความซับซ้อนของโมเดลมากเกินไป ( $\lambda$  มีค่าน้อยๆ นั่นคือ  $\log(\lambda)$  เป็นลบมากๆ) กรณีนี้เป็นโอเวอร์พิทติ้ง. สังเกตุว่า ค่าผิดพลาดของชุดฝึกหัดจะต่ำ แต่ค่าผิดพลาดของชุดทดสอบจะสูง. ความต่างระหว่างค่าผิดพลาดของชุดฝึกหัดกับชุดทดสอบมีค่ามาก. โมเดลซับซ้อนมากๆ จะสามารถลดค่าผิดพลาดของข้อมูลชุดฝึกหัดลงได้ แต่หากมากเกินไป จะไปลดค่าผิดพลาดในข้อมูลชุดฝึกหัดที่เกิดจากสัญญาณรบกวนด้วย โมเดลจะทำนายกับข้อมูลชุดทดสอบได้ไม่ดี. กรณีนี้จะเรียกว่า เป็นกรณีที่มีความแปรผันสูง (High Variance) ซึ่งกรณีนี้สื่อถึงโอเวอร์พิทติ้ง. กล่าวอีกอย่างหนึ่ง ใบอัสสูงหมายถึงโมเดลของเราไม่ยึดหยุ่นมากพอ ส่วนความแปรผันสูงหมายถึงโมเดลของเรา yi ยึดหยุ่นมากเกินไป. โมเดลที่ดีที่สุดที่ทำได้ คือ โมเดลที่ลดให้ทั้งใบอัสและความแปรผันต่ำ. แต่ไม่ว่าโมเดลไหนก็ตาม เราทำดีที่สุดได้แค่ดีที่สุด จะผ่านข้อจำกัดของธรรมชาติไปไม่ได้. ทฤษฎีที่กล่าวถึงข้อจำกัดของการทำโมเดลนี้คือ ทวิ逼ของใบอัสกับความแปรผัน[27].

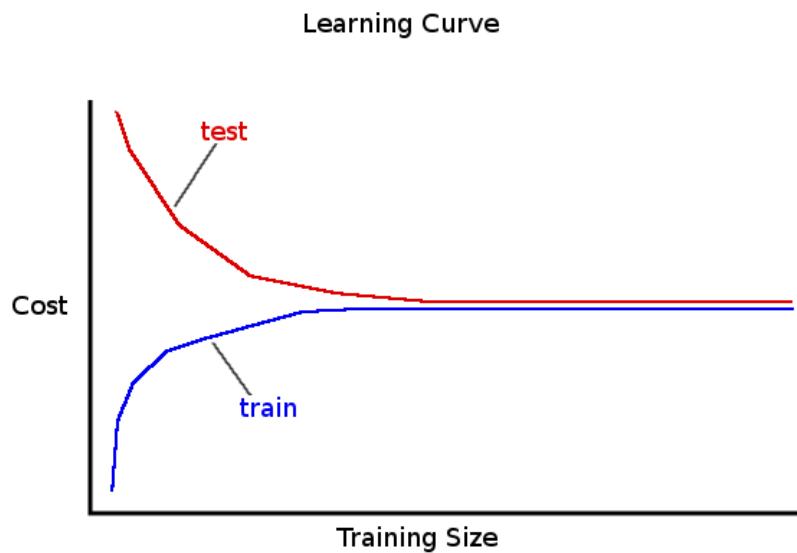
ทวิ逼ของใบอัสกับความแปรผัน (Bias/Variance Dilemma). เจมันและคณะ[27] ศึกษาความสามารถและขีดจำกัดของการทำโมเดล และสรุปว่า เราสามารถทำโมเดลให้ดีที่สุดได้โดยลดทั้งค่าใบอัสและความแปรผันให้ต่ำ. แต่ค่าผิดพลาดก็มีขีดจำกัดหนึ่ง (จົ້ນກັບธรรมชาติของปัญหา) ที่เราไม่สามารถลดลงไปให้ต่ำกว่านั้นได. การที่เรายາຍมาจะลดค่าผิดพลาดให้ต่ำกว่านั้นโดยการลดส่วนหนึ่ง ก็จะไปเพิ่มอีกส่วน เช่น หากพยาຍາມลดใบอัສมากเกินไป ก็จะทำให้ส่วนที่เกิดจากความแปรผันเพิ่ม และก็เช่นเดียวกันทางกลับกัน ซึ่งความจริงนี้เรียกว่า ทวิ逼ของใบอัสกับความแปรผัน.

#### 7.4.2 เส้นโค้งเรียนรู้

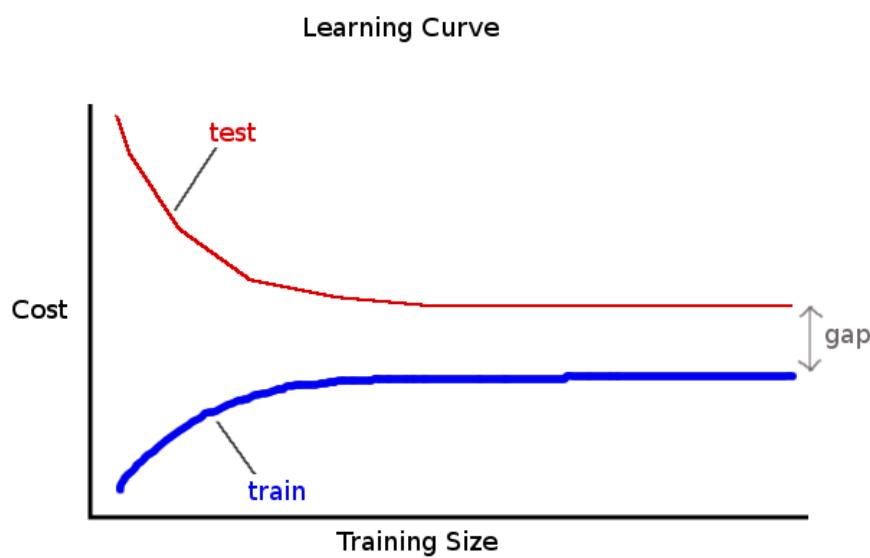
เส้นโค้งเรียนรู้ (Learning Curve) เป็นเครื่องมือช่วยแสดงความสัมพันธ์ระหว่างความพยายามที่ใช้ไปในการฝึกโมเดลกับผลการทำงานของโมเดล. เส้นโค้งเรียนรู้สามารถใช้เพื่อช่วยให้เงื่อน件ว่า โมเดลที่กำลังทำอยู่มีความเสียงที่จะมีใบอัสสูงหรือเสียงที่จะมีความแปรผันสูง. เส้นโค้งเรียนรู้ สร้างโดย การตรวจสอบผลการฝึกโมเดลที่จำนวนจุดข้อมูลฝึกต่างๆ และคำนวณค่าผิดพลาดของชุดฝึกและค่าผิดพลาดของชุดทดสอบมาด้วย ดังแสดงในรูป 7.4.

การวัดค่าผิดพลาดของชุดฝึกก็วัดค่าผิดพลาดเฉพาะจากจุดข้อมูลที่ใช้ฝึก เช่น หากฝึกโมเดลด้วย 1 จุดข้อมูล ก็หาค่าผิดพลาดของการทำนายค่า 1 จุดข้อมูลนี้. ดังนั้น เมื่อจำนวนจุดข้อมูลน้อยๆ ค่าผิดพลาดของชุดฝึกก็จะน้อยด้วย. เมื่อเพิ่มจำนวนจุดข้อมูลฝึกขึ้น ค่าผิดพลาดของชุดฝึกก็จะเพิ่มขึ้นและลุ่เข้าสู่ค่าหนึ่ง. ในขณะเดียวกัน เมื่อจำนวนจุดข้อมูลฝึกเพิ่มขึ้น ค่าผิดพลาดของชุดทดสอบจะลดลงจนลุ่เข้าสู่ค่าหนึ่ง.

ในกรณีที่ โมเดลไม่มีความเสียงจากใบอัสสูง ค่าผิดพลาดจากชุดฝึกและค่าผิดพลาดจากชุดทดสอบจะลุ่เข้าหากันที่ใกล้เคียงกัน ดังแสดงในรูป 7.4. แต่หากโมเดลไม่มีความเสียงจากความแปรผันสูง เมื่อโมเดลมีความแปรผันสูง นั่นหมายความว่า โมเดลมีความยึดหยุ่นมากเกินไป มากจนพอที่จะปรับตัวเข้ากับสัญญาณ



รูปที่ 7.4: ภาพคร่าวๆ ของเล็บโค้งเรียนรู้ในกรณีที่โมเดลมีไบอสสูง (High Bias). เมื่อจำนวนของจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดของชุดฝึกและค่าผิดพลาดของชุดทดสอบมีค่าใกล้เคียงกัน.



รูปที่ 7.5: ภาพคร่าวๆ ของเล็บโค้งเรียนรู้ในกรณีที่โมเดลมีความแปรผันสูง (High Variance). เมื่อจำนวนของจุดข้อมูลฝึกมากขึ้น ค่าผิดพลาดของชุดฝึกและค่าผิดพลาดของชุดทดสอบมีค่าต่างกัน

กระบวนการที่เห็นในข้อมูลฝึกได้. ผลคือโมเดลสามารถลดค่าผิดพลาดของชุดฝึกได้มากๆ แม้จำนวนจุดข้อมูลฝึกจะเพิ่มขึ้นมาก. แต่การที่โมเดลปรับตัวเข้ากับลักษณะของชุดฝึกไม่ได้ช่วยให้คุณภาพการทำงานของชุดทดสอบดีขึ้นด้วย (ทำร้ายยังอาจทำให้แย่ลงอีก) ดังนั้นกรณีที่เสี่ยงจากความแปรผันสูง สังเกตุได้จาก ค่าผิดพลาดของชุดทดสอบจะถูกลากเข้าสู่ค่าที่สูงกว่าค่าที่ค่าผิดพลาดของชุดฝึก ถูกเข้าอย่างเห็นได้ชัด ดังแสดงในรูป 7.5.

ทั้งรูป 7.4 และ 7.5 เป็นภาพคร่าวๆ เพื่อให้เห็นภาพรวม. ในความเป็นจริง เล็บโค้งเรียนรู้ที่วัด

อาจจะมีสัญญาณรบกวนมาก หรือแม้แต่สเกลของการวัดกราฟอาจทำให้ต้องใช้ความระมัดระวังในการอ่านเส้นโค้งเรียนรู้. หัวข้อ 7.4.2.1 แสดงตัวอย่างของเส้นโค้งเรียนรู้ จากโจทย์การหาค่าคงถอยมิติเดียว.

#### 7.4.2.1 เส้นโค้งเรียนรู้สำหรับตัวอย่างการหาค่าคงถอยมิติเดียว

จากตัวอย่างฯ (หัวข้อ 6.1 และ 6.5.1) หากทดลองวางแผนเส้นโค้งเรียนรู้ของโมเดลโครงข่ายประสาทเทียมที่ใช้แค่ 1 หน่วยซ่อน จะได้ผลดังรูป 7.6.

ภาพของเอาร์พุตหลังจากฝึกด้วยจำนวนจุดข้อมูลต่างๆ ในรูป 7.6 แสดงให้เห็นว่า การใช้โครงข่ายประสาทเทียม 1 หน่วยซ่อนนั้น ทำให้โมเดลไม่มีความยืดหยุ่นเพียงพอ. กรณีนี้คือใบอัสสูง และเส้นโค้งเรียนรู้ในภาพซ้ายบนสุดก็แสดงการลู่เข้าของค่าผิดพลาดจากทั้งชุดฝึกและชุดทดสอบ. สังเกตุว่า ค่าผิดพลาดทั้งสองลู่เข้าสู่ค่าประมาณ 20. และ เพื่อตรวจสอบดูว่าค่าทั้งสองลู่เข้าใกล้กันมากขนาดไหน ภาพขวาบนสุดแสดงอัตราส่วนระหว่างค่าผิดพลาดชุดทดสอบและชุดฝึกหัด. ที่จำนวนข้อมูลฝึกมากๆ (ปลายเส้นโค้งเรียนรู้) อัตราส่วนนี้ลู่เข้าหาค่าประมาณ 1.1. อัตราส่วนที่มีค่าใกล้ 1 นี้บ่งบอกว่าค่าผิดพลาดทั้งสองมีค่าใกล้กันมาก.

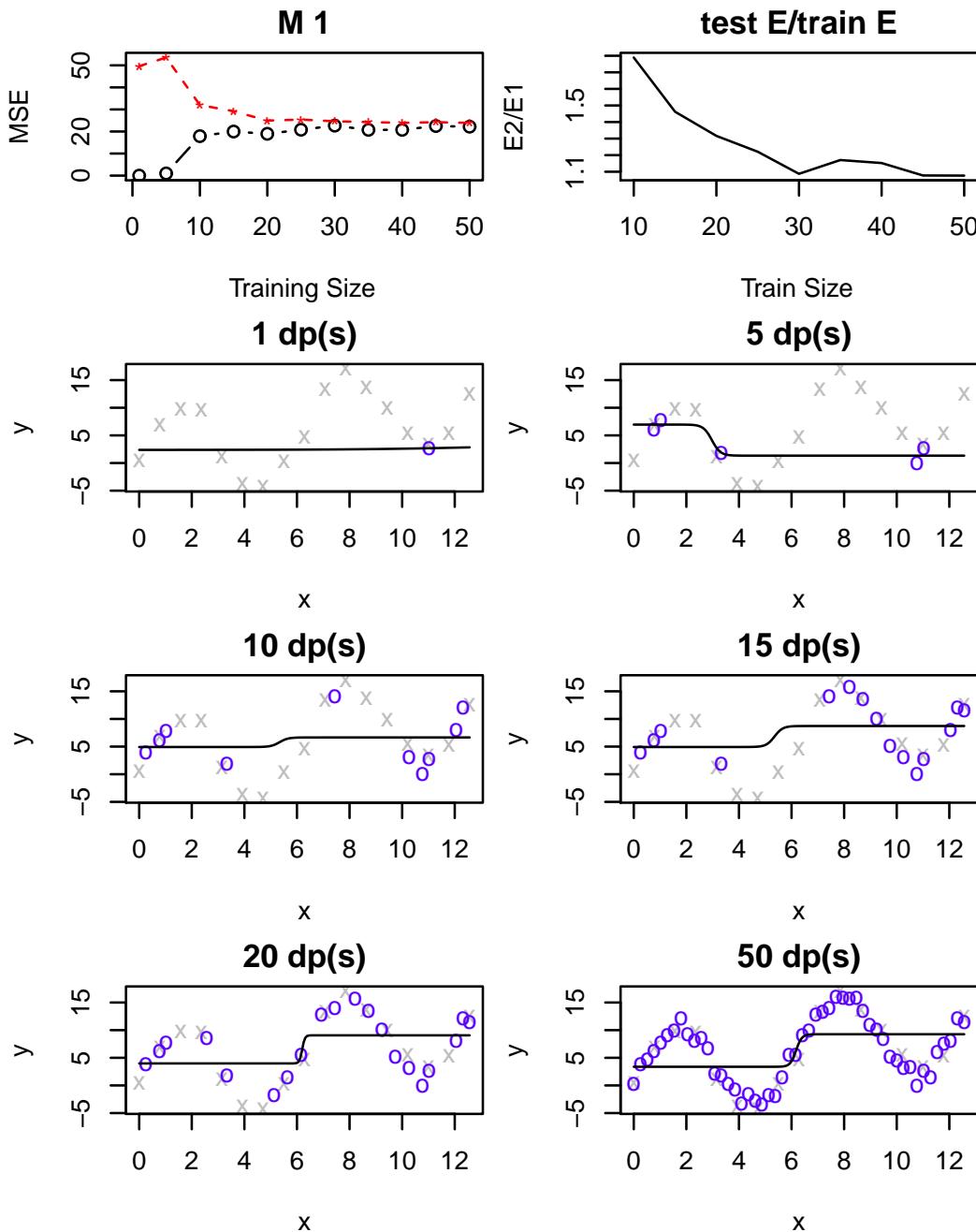
เปรียบเทียบผลข้างต้นกับกรณีที่ไม่ได้มีความแปรผันสูง ความแปรผันสูงเกี่ยวพันกับการที่ไม่ได้มีความยืดหยุ่นมากเกินไป. จากตัวอย่างเดียวกัน เมื่อใช้โครงข่ายประสาทเทียมที่มี 2000 หน่วยซ่อน เส้นโค้งเรียนรู้ที่ได้แสดงดังรูป 7.7.

ภาพของเอาร์พุตหลังจากการฝึกด้วยจำนวนจุดข้อมูลต่างๆ ที่แสดงในรูป 7.7 ยืนยันว่า การใช้โครงข่ายประสาทเทียมสองชั้นขนาด 2000 หน่วยซ่อนนั้น ทำให้โมเดลมีความยืดหยุ่นมากเกินไป มากจนโมเดลไปรับเข้ากับสัญญาณรบกวนในข้อมูลฝึกได้. กรณีนี้แสดงถึงความแปรผันสูงอย่างชัดเจน.

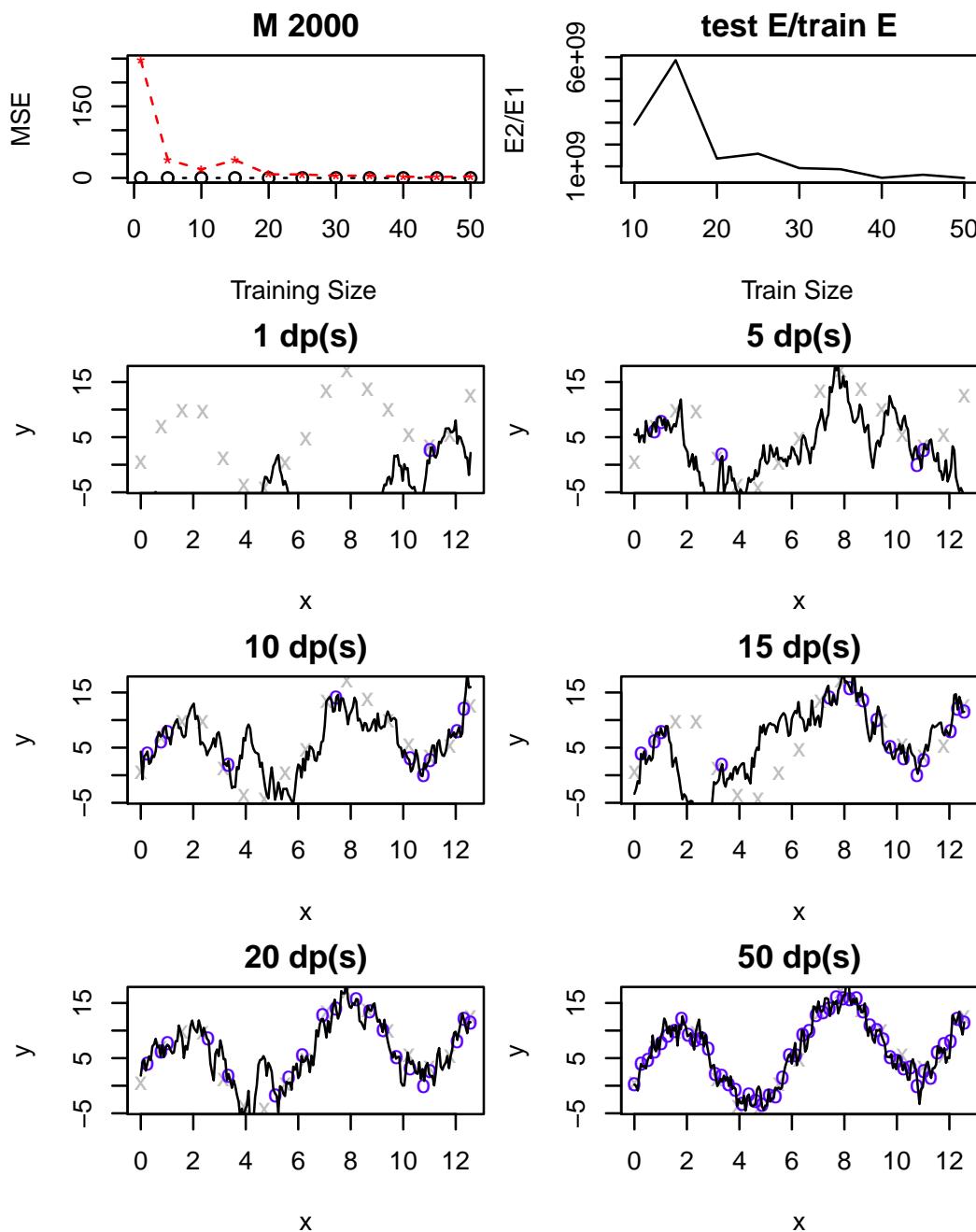
ภาพซ้ายบนสุดของรูป 7.7 แสดงเส้นโค้งเรียนรู้ ที่ค่าผิดพลาดทั้งสองลู่เข้าแล้ว. แต่เนื่องจากสเกลของภาพ จึงทำให้ยากต่อการอ่านค่าผิดพลาดที่ลู่เข้าทั้งสองนั้นใกล้เคียงหรือห่างกันมากขนาดไหน. อัตราส่วนค่าผิดพลาดที่แสดงในภาพขวาบนสุดช่วยบอกความต่างระหว่างค่าผิดพลาดทั้งสอง. สังเกตุค่าอัตราส่วนนี้ที่ปลายของกราฟ ซึ่งมีค่ามาก (ราวๆ  $10^9$  เท่า). นั่นคือ ค่าผิดพลาดทั้งสองต่างกันมาก บ่งบอกถึงกรณีความแปรผันสูงอย่างชัดเจน.

ตัวอย่างในหัวข้อ 6.1 ใช้โครงข่ายประสาทเทียม 20 หน่วยซ่อน. หากลองนำเส้นโค้งเรียนรู้มาวดดังแสดงในรูป 7.8 จะเห็นว่า ที่ปลายเส้นโค้งเรียนรู้ ค่าผิดพลาดของชุดฝึกค่อนข้างห่างจากค่าผิดพลาดของชุดทดสอบ (ค่าอัตราส่วนความผิดพลาดประมาณกว่า 5 เท่า ภาพที่ 3 จากซ้าย) ดังนั้นโครงข่ายประสาทเทียมขนาด 20 หน่วยซ่อนกับปัญหาตัวอย่างในหัวข้อ 6.1 จะเข้ากรณีของความแปรผันสูง. และภาพขวาสุดยังแสดงให้เห็นอย่างชัดเจนว่า เอาร์พุตจากโครงข่ายได้ปรับตัวไปเข้ากับสัญญาณรบกวนของข้อมูลฝึกบ้างแล้ว.

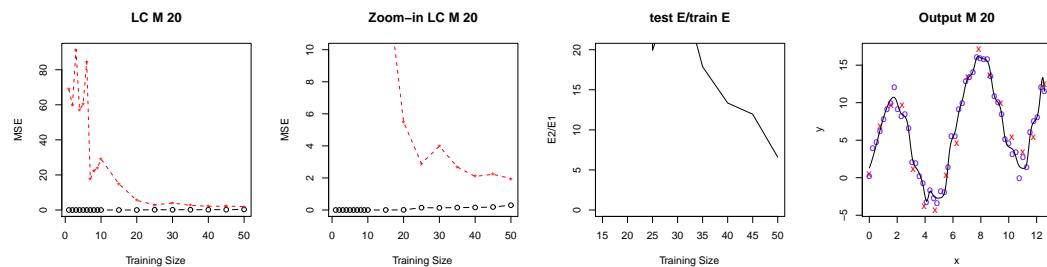
ตัวอย่างข้างต้นแสดงปัญหาการหาค่าคงถอย ที่อินพุตมีมิติเดียว. อินพุตมีมิติเดียวช่วยให้เรามองเห็นกรณีทั่วไปอัสสูงและความแปรผันสูงได้ชัดเจนขึ้น. ปัญหาที่อาจเจอในทางปฏิบัติ อาจไม่ได้มีรูปมาตรฐานที่ช่วยให้สามารถวัดเอาร์พุตของโมเดลมาดูเทียบกับจุดข้อมูลฝึกหัดและจุดข้อมูลทดสอบได้ง่ายเช่นนี้. ดังนั้นจึงใช้เส้นโค้งเรียนรู้และความห่างที่ค่าผิดพลาดของชุดฝึกและชุดทดสอบ เพื่อบ่งชี้กรณีใบอัสสูง หรือ



รูปที่ 7.6: เส้นโค้งเรียนรู้ในกรณีใบอัลตรา. เส้นโค้งเรียนรู้ (ภาพข่ายบนสุด) เส้นกราฟที่ใช้สัญลักษณ์ ‘o’ แทนค่าผิดพลาดของชุดฝึก เส้นกราฟที่ใช้สัญลักษณ์ ‘\*’ แทนค่าผิดพลาดของชุดทดสอบ. ที่จำนวนจุดข้อมูลมากๆ (ปลายของเส้นโค้งเรียนรู้) ค่าผิดพลาดของชุดฝึกและค่าผิดพลาดของชุดทดสอบมีค่าใกล้เคียงกัน. อัตราส่วน  $\frac{E_{\text{test}}}{E_{\text{train}}}$  (ภาพขวบบนสุด) มีค่าราวๆ 1.1 ที่ปลายของเส้นโค้งเรียนรู้. ภาพอื่นๆแสดงอาต์พุตที่ได้จากโมเดล (เส้นกราฟ) เมื่อฝึกด้วย 1, 5, 10, 15, 20, และ 50 จุดข้อมูล ตามระบุที่ชื่อภาพ. จุดข้อมูลที่ใช้ฝึกแสดงด้วยสัญลักษณ์ ‘o’ และจุดข้อมูลที่ใช้ทดสอบแสดงด้วยสัญลักษณ์ ‘x’



รูปที่ 7.7: เส้นโค้งเรียนรู้ในกรณีความแปรผันสูง. เส้นโค้งเรียนรู้ (ภาพช้ายบนสุด) เส้นกราฟที่ใช้สัญลักษณ์ ‘○’ แทนค่าผิดพลาดของข้อมูลต่อไปนี้ (ภาพช้ายบนสุด) เส้นกราฟที่ใช้สัญลักษณ์ ‘\*’ แทนค่าผิดพลาดของข้อมูลสอบ. ที่จำนวนจุดข้อมูลมากๆ (ปลายของเส้นโค้งเรียนรู้) ค่าผิดพลาดของข้อมูลต่อไปนี้จะลดลงมาก. กราฟในภาพช้ายบนสุดอาจมองเห็นความต่างไม่ชัดเจน แต่อัตราส่วน  $\frac{E_{\text{test}}}{E_{\text{train}}}$  (ภาพช้ายบนสุด) โดยเฉพาะที่ปลายของเส้นโค้งเรียนรู้ ช่วยเน้นถึงความแตกต่างนี้ (สังเกตุสเกลของค่าอัตราส่วนที่แกนตั้ง. ค่า  $1e+09$  คือ  $10^9$ ) ภาพอื่นๆแสดงเอาต์พุตที่ได้จากโมเดล (เส้นกราฟ) เมื่อฝึกด้วย 1, 5, 10, 15, 20, และ 50 จุดข้อมูล ตามระบุที่ชื่อภาพ จุดข้อมูลที่ใช้ฝึกแสดงด้วยสัญลักษณ์ ‘○’ และจุดข้อมูลใช้ทดสอบแสดงด้วยสัญลักษณ์ ‘x’



รูปที่ 7.8: เส้นโค้งเรียนรู้. ภาพข่ายสุดแสดงเส้นโค้งเรียนรู้ เส้นทีบสัญญาลักษณ์ ‘0’ แทนค่าผิดพลาดของชุดฝึก เส้นประสัญญาลักษณ์ ‘\*’ แทนค่าผิดพลาดของชุดทดสอบ. ภาพที่สองจากข่ายแสดงเส้นโค้งเรียนรู้ด้วยสเกลขยาย เพื่อตรวจสอบความทั่งของค่าผิดพลาดสองชุดได้ชัดเจนขึ้น. ภาพที่สามจากข่ายแสดงค่าอัตราส่วนความผิดพลาด. ภาพข่าวสุดแสดงเอาต์พุตของโครงข่าย หลังจากฝึกด้วยจำนวนจุดข้อมูลฝึกทั้งหมด เส้นทีบแทนค่าเอาต์พุตจากโครงข่ายประสาทเทียม สัญญาลักษณ์ ‘0’ แทนจุดข้อมูลฝึก สัญญาลักษณ์ ‘x’ แทนจุดข้อมูลทดสอบ.

ความแปรผันสูง สำหรับการวิเคราะห์การทำโมเดล. หัวข้อ 7.4.3 อภิปรายทางเลือกที่แนะนำ เพื่อปรับปรุงคุณภาพโมเดลตามกรณีไปอัตโนมัติ หรือความแปรผันสูง.

### 7.4.3 ทางเลือกที่แนะนำ

หลังจากพอร์ตแล้วว่า สถานะการณ์ของโมเดลที่ใช้อยู่เข้ากรณีไปอัตโนมัติ หรือความแปรผันสูง ก็สามารถพิจารณาทางเลือกได้ ดังนี้

- เก็บข้อมูลมาเพิ่มสำหรับการฝึก. การเพิ่มจำนวนข้อมูลในการฝึก จะช่วยในกรณีความแปรผันสูง
- เลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต. การเลือกเฉพาะบางมิติของข้อมูลมาเป็นอินพุต (ลดมิติของอินพุตลง) ก็จะช่วยในกรณีความแปรผันสูง
- เพิ่มลักษณะที่สำคัญใหม่เข้าไปในอินพุต. การเพิ่มลักษณะที่สำคัญใหม่เข้าไปในอินพุต (เพิ่มมิติของอินพุตขึ้น) โดยทั่วไปแล้ว จะช่วยในกรณีไปอัตโนมัติ
- เพิ่มความซับซ้อนของโมเดลขึ้น. การเพิ่มความซับซ้อนของโมเดล เช่น การเพิ่มจำนวนหน่วยซ่อน จะช่วยในกรณีไปอัตโนมัติ
- ลดความซับซ้อนของโมเดลลง. การลดความซับซ้อนของโมเดล เช่น การลดจำนวนหน่วยซ่อน การใช้ReLU หรือ Leaky ReLU หรือ การทำการหยุดก่อนกำหนด จะช่วยในกรณีความแปรผันสูง.

ตาราง 7.2 สรุปทางเลือกที่แนะนำสำหรับกรณีไปอัตโนมัติ และกรณีความแปรผันสูง. สุดท้ายหากลองวิธีทั่วๆไปดังนี้แล้ว ผลยังไม่ดีขึ้น หรือดีขึ้นแล้วแต่อยากให้ดีขึ้นอีก ผู้ทำโมเดลอาจจะลองวิเคราะห์ตรวจสอบผลที่อัลกอริทึมทำผิด ดูว่าผิดลักษณะไหน อย่างไร. เมื่อว่า ผู้ทำโมเดลอาจจะพบคุณสมบัติเฉพาะบางอย่างที่ผลมักจะผิด เช่น หากเป็นการจำแนกตัวเลขจากภาพ (ตัวอย่างในหัวข้อ 6.5.4) อัลกอริทึมอาจจำแนก

เลข 2 เป็นเลข 4 บอยๆ ซึ่งเมื่อผู้ทำไม่เดลดูภาพของเลขที่จำแนกผิด ก็อาจพบว่า มีรูปแบบการเขียนเลข 2 แบบหนึ่ง ที่มักจะจำแนกผิด. หากผู้ทำไม่เดลพรูปแบบเฉพาะนั้น ผู้ทำไม่เดลก็อาจเพิ่มการจัดการเฉพาะสำหรับรูปแบบที่สันสนบอยๆ นั้นได้ เป็นต้น.

ตารางที่ 7.2: สรุปทางเลือกที่แนะนำในการนี้ไปอัสสูรและความแปรผันสูง.

ทางเลือก	กรณี	
	ใบอัสสูร	ความแปรผันสูง
เพิ่มรอบฟีก	แนะนำ	
เพิ่มจำนวนหน่วยย่อย	แนะนำ	
ทำเรกูล่าไรเซ็น		แนะนำ
ทำการหยุดก่อนกำหนด		แนะนำ
ลดมิติของอินพุตลง		แนะนำ
เพิ่มมิติของอินพุตขึ้น		แนะนำ
เพิ่มจำนวนจุดข้อมูลฟีก	แนะนำ	แนะนำ

## 7.5 แบบฝึกหัด

- จงหาฟังชันค่าอนุพันธ์ของสมการ 7.1 และ นำฟังชันค่าอนุพันธ์ที่ได้มาเขียนโปรแกรม และวัดรูปเลียบแบบรูป 7.2 ใช้ค่าเริ่มต้น  $[x_1, x_2]^{(0)} = [2.5, 3]$  ค่าขนาดกว้าง (ทั้งวิธีลิงเกรเดียนต์ และ วิธีลงเกรเดียนต์กับโมเมนตัม) 0.125 และค่าอัตราโมเมนตัม 0.1. เปรียบเทียบค่าตอบสุดท้ายที่ได้ จำนวนรอบฟีกที่ใช้ และเวลาที่ใช้ ของทั้งสามวิธี.
- จากข้อ 1 จงทดลองวิธีลิงเกรเดียนต์กับโมเมนตัม ที่ค่าขนาดกว้างและอัตราโมเมนตัมต่างๆ. สรุปผล และอภิราย.
- จงนำวิธีลิงเกรเดียนต์กับโมเมนตัมไปใช้กับฟีกโครงข่ายประสาทเทียม. ค่าวิบ.  $w$  ในสมการ 7.2 คือ ค่าน้ำหนัก. ทดสอบการทำงานกับปัญหาการหาค่าลดถอย ปัญหาการจำแนกกลุ่มแบบสองกลุ่ม และ ปัญหาการจำแนกกลุ่มแบบหลายกลุ่ม. เปรียบเทียบผลการฟีกกับการฟีกด้วยวิธีลิงเกรเดียนต์ สรุปผล และ อภิราย.
- จงออกแบบการทดลอง และทดลองผลการทำงานของโครงข่ายประสาทเทียม เมื่อกำหนดค่าเริ่มต้น ด้วยการสุ่มจากการแจกแจงแบบเอกรูป (uniform distribution) เปรียบเทียบกับการกำหนดค่าเริ่มต้น ด้วยวิธีเหจី-វិដូរុវ. สรุปและอภิรายผล. หมายเหตุ ให้ทดลองฟีกด้วยวิธีลิงเกรเดียนต์ វិនិបីអេដីស និងវិនិបីអេសមិតិ និងทดลองประยុកត์กับปัญหาหลากหลายแบบ เช่น การหาค่าลดถอย การจำแนกกลุ่ม และ การจำแนกกลุ่มหลายกลุ่ม.

5. จงออกแบบการทดลอง และทดลองผลการทำงานของโครงข่ายประสาทเทียม เมื่อใช้วิธีฝึกต่างๆ ได้แก่ วิธีลงเกรเดียนต์, วิธีบีเอฟเจ้อส และวิธีเอสซีจี กับปัญหาหลากหลายแบบ เช่น การหาค่าคงถอย การจำแนกกลุ่ม และการจำแนกกลุ่มหลายกลุ่ม. สรุปและอภิปรายผล.
6. จากตัวอย่างในหัวข้อ 6.1 จงสร้างเส้นโค้งเรียนรู้ของโครงข่ายประสาทเทียมขนาด 20 หน่วยย่อย เพื่อเปรียบเทียบกับรูป 7.8. อภิปรายผล. หมายเหตุ รูป 7.8 ได้จากการใช้ โครงข่ายประสาทเทียมขนาด 20 หน่วยช่อง ที่ฝึกด้วยวิธีเอสซีจี แบบไม่ทำการหยุดก่อนกำหนด และรอบการฝึกสูงสุดคือ 500 รอบ และจะจบก่อนถึงรอบฝึกสูงสุดได้ เมื่อค่าพังชั่นจุดประสงค์เปลี่ยนแปลงน้อยกว่า  $10^{-8}$  หรือ ค่าเฉลี่ยของเกรเดียนต์ยกกำลังสองน้อยกว่า  $10^{-12}$ .
7. จากแบบฝึกหัดข้อ 6 จงทดลองปรับปรุงคุณภาพการทำงานของโครงข่ายประสาทเทียม โดยใช้ผลจากเส้นโค้งเรียนรู้ประกอบ. สรุปสิ่งที่ได้ทดลองทำ อภิปรายสิ่งที่ได้เรียนรู้ ชี้แจงเหตุผลที่รองรับ และวิเคราะห์ผลจากการทดลองหลังการปรับปรุง.
8. จงใช้เส้นโค้งเรียนรู้เพื่อวิเคราะห์การทำงานของโครงข่ายกับตัวอย่างการหาค่าคงถอย ในหัวข้อ 6.5.2. จงทดลองทำการปรับปรุง สรุปและอภิปรายการทดลองและผล.
9. จงใช้เส้นโค้งเรียนรู้เพื่อวิเคราะห์การทำงานของโครงข่ายกับตัวอย่างการจำแนกกลุ่ม ในหัวข้อ 6.3. จงทดลองทำการปรับปรุง สรุปและอภิปรายการทดลองและผล.
10. จงใช้เส้นโค้งเรียนรู้เพื่อวิเคราะห์การทำงานของโครงข่ายกับตัวอย่างการจำแนกกลุ่มแบบหลายกลุ่ม ในหัวข้อ 6.5.4. จงทดลองทำการปรับปรุง สรุป และอภิปรายการทดลองและผล.



## បរវាណ្យកម្ម

- [1] Abu-Mostafa, Y. S. How to teach computers to learn on their own. *Scientific American* (June 2012).
- [2] Access to Insight, W. Tipitaka: The pali canon. <http://www.accesstoinsight.org/tipitaka/index.html>.
- [3] Adrian, E. D., and Zotterman, Y. The impulses produced by sensory nerve endings. *Journal of Physiology* 61 (1926), 151–171.
- [4] 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.
- [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] Bache, K., and Lichman, M. UCI machine learning repository, 2013.
- [7] 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.
- [8] Belisle, C. J. P. Convergence theorems for a class of simulated annealing algorithms on rd. *Journal of Applied Probability* 29 (1992), 885–895.
- [9] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, New York, USA, 2006.

- [10] Blais, B. S., and Cooper, L. Bcm theory. [http://www.scholarpedia.org/article/BCM\\_theory#Original\\_BCM\\_.28Bienenstock\\_et\\_al.\\_1982.29](http://www.scholarpedia.org/article/BCM_theory#Original_BCM_.28Bienenstock_et_al._1982.29), 2008. Scholarpedia 3(3): 1570.
- [11] Blanzieri, E., and Bryl, A. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review* 29, 1 (2008), 63–92.
- [12] Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [13] Bock, R., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jirina, M., Klaschka, J., Kotrc, E., Savicky, P., Towers, S., Vaicius, A., and W., W. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A* 516 (2004), 511–528.
- [14] Breiman, L. Random forests. *Machine Learning* 45, 1 (2001), 5–32.
- [15] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing* 16 (1995), 1190–1208.
- [16] 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).
- [17] Chanloha, P., Chinrungueng, 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.
- [18] Chong, E. K. P., and Zak, S. *An Introduction to Optimization*, 2nd ed. Wiley-Interscience, 2001.
- [19] Coates, A., Abbeel, P., and Ng, A. Y. Apprenticeship learning for helicopter control. *Communication of the ACM* (2009).
- [20] Cortes, C., and Vapnik, V. Support-vector networks. *Machine Learning* 20, 3 (Sep 1995), 273–297.
- [21] Costa-Jussa, M. R., and Farrus, M. Statistical machine translation enhancements through linguistic levels: A survey. *ACM Computing Surveys* 46, 3 (2014).

- [22] 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).
- [23] Cybenko, G. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems* 2, 4 (1989), 303–314.
- [24] Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction* 4, 2 (2010), 81–173.
- [25] 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.
- [26] Fletcher, R., and Reeves, C. M. Function minimization by conjugate gradients. *Computer Journal* 7 (1964), 148–154.
- [27] Geman, S., Bienenstock, E., and Doursat, R. Neural networks and the bias/variance dilemma. *Neural Computation* 4 (1992), 1–58.
- [28] 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.
- [29] Godwin, D., and Cham, J. Your brain by the numbers. *Scientific American* (November 2012).
- [30] Grimmett, G., and Stirzaker, D. *Probability and Random Processes*, 3rd ed. ed. Oxford University Press, 2001.
- [31] 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.
- [32] Hagan, M., and Menhaj, M. Training feed-forward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks* 5, 6 (1994), 989–993.

- [33] Hanson, R., and Mendius, R. *สมองแห่งพุทธะ*, 1 ed. อัมรินทร์ธรรมะ, 378 ถ.เจียพฤกษ์ ตัลิ่งชัน กรุงเทพฯ 10170, พ.ศ. 2557. แปลโดย ณัชร สยามวาลา จาก Hanson and Mendius, Buddha's Brain: The Practical Neuroscience of Happiness, Love, and Wisdom, New Harbinger Publications, 2009.
- [34] Haykin, S. O. *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2009.
- [35] Holst, A., and Jonasson, A. Classification of movement patterns in skiing. *Frontiers in Artificial Intelligence and Applications* 257 (2013), 115–124.
- [36] Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4, 2 (1991), 251–257.
- [37] Juhola, M., and Laurikkala, J. Missing values: how many can they be to preserve classification reliability? *Artificial Intelligence Review* 40, 3 (2013), 231–245.
- [38] Katanyukul, T. Ruminative reinforcement learning. *Journal of Computers* (2014).
- [39] Katanyukul, T., and Chong, E. K. P. Intelligent inventory control via ruminative reinforcement learning. *Journal of Applied Mathematics* (2014).
- [40] 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.
- [41] 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.
- [42] Katanyukul, T., and Ponsawat, J. Customer analysis via video analytics. In *submitted to MICAI 2016* (2016), Dummy, Ed., Dummy, DummySpringer.
- [43] 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.
- [44] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science* 220 (1983), 671–680.

- [45] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 1 (1982), 59–69.
- [46] Le Cun, 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.
- [47] 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.
- [48] 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.
- [49] Linden, D. J. *The Accidental Mind: How Brain Evolution Has Given Us Love, Memory, Dreams, and God*. Belknap Press, 2008.
- [50] Magazzeni, D., Py, F., Fox, M., Long, D., and Rajan, K. Policy learning for autonomous feature tracking. *Autonomous Robots* 37, 1 (2014), 47–69.
- [51] McCaffrey, J. Understanding and using k-fold cross-validation for neural networks. *Visual Studio Magazine* (Oct 2013).
- [52] Minsky, M., and Papert, S. *Perceptrons: an introduction to computational geometry*. The MIT Press, 1969.
- [53] Mitchell, T. M. *Machine Learning*. McGraw-Hill, 1997.
- [54] Moller, M. F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6 (1993), 525–533.
- [55] Nabney, I. T. Netlab version 3.2. <http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/downloads/>.
- [56] Nelder, J. A., and Mead, R. A simplex algorithm for function minimization. *Computer Journal* 7 (1965), 308–313.
- [57] NeuroBank. <http://neuronbank.org>.

- [58] Ng, A. Machine learning class. Coursera.org, 2013.
- [59] 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.
- [60] of Neurological Disorders, N. I., and Stroke. Brain basic: Know your brain. [http://www.ninds.nih.gov/disorders/brain\\_basics/know\\_your\\_brain.htm](http://www.ninds.nih.gov/disorders/brain_basics/know_your_brain.htm), October 2012. NIH Publication No. 01 3440a.
- [61] 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).
- [62] Poo, M. M. ibioseminar: Learning and memory: From synapse to perception. <http://www.ibiology.org>, 2010.
- [63] 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.
- [64] 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.
- [65] Riedmiller, M., and Braun, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proceedings of the IEEE International Conference on Neural Networks* (1993).
- [66] Rojas, R. *Neural Networks, a systematic introduction*. Springer-Verlag, Berlin, New York, 1996.
- [67] 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.
- [68] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.

- [69] Russell, S. J., and Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.
- [70] Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development* 3 (1959), 211–229.
- [71] 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.
- [72] Saxe, R. The brain v.s. the mind. [theagenda.tvo.org](http://theagenda.tvo.org), July 2012.
- [73] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavuucuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature* 529 (2016), 484–489.
- [74] Sutton, R. S., and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [75] 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.
- [76] Tesauro, G. J. Td-gamma, a self-teaching backgammon program, achieves master-level play. *Neural Computation* 6, 2 (1994), 215–219.
- [77] The MathWorks, I. Neural network toolbox user's guide, 2010.
- [78] 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.
- [79] Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [80] Wikipedia. Wikipedia the free encyclopedia. <https://en.wikipedia.org>.

- [81] 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).
- [82] Zhu, W., Miao, J., Hu, J., and Qing, L. Vehicle detection in driving simulation using extreme learning machine. *Neurocomputing* 128 (2014), 160–165.