# การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

โดย

นายไพศาล สิมาเลาเต่า

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2550

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

# การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด

โดย
นายไพศาล สิมาเลาเต่า

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคอมพิวเตอร์
บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร
ปีการศึกษา 2550
ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

# RECOGNITIVE APPLICATION DEVELOPMENT USING GRID

**By**

**Paisan Simalaotao**

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree**

**MASTER OF SCIENCE**

**Department of Computing**

**Graduate School**

**SILPAKORN UNIVERSITY**

**2007**

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร อนุมัติให้วิทยานิพนธ์เรื่อง " การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด " เสนอโดย นายไพศาล สิมาเลาเต่า เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

…….....

(รองศาสตราจารย์ ดร.ศิริชัย ชินะตังกูร)

คณบดีบัณฑิตวิทยาลัย

วันที่.....เดือน..... พ.ศ.....

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์

คณะกรรมการตรวจสอบวิทยานิพนธ์

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.จันทนา ผ่องเพ็ญศรี)

...../...../.....

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ )

...../...../.....

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์)

...../...../.....

46307311 : สาขาวิชาวิทยาการคอมพิวเตอร์
คำสำคัญ : การรู้จำ / เครือข่ายใยประสาท / การประมวลผลแบบกริด

ไพศาล สิมาเลาเต่า : การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด. อาจารย์ที่ปรึกษาวิทยานิพนธ์ : ผศ.ดร.ปานใจ ธารทัศนวงศ์. 125 หน้า.

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด (Recognition Application Development Using Grid: RADUG) เพื่อใช้ประกอบการวัดประสิทธิภาพการประมวลผล ซึ่งจะมีการพัฒนาโปรแกรมประยุกต์ 2 แบบ คือ แบบที่มีการประมวลผลบนเครื่องคอมพิวเตอร์เครื่องเดียวที่มีหน่วยประมวลผลเดียว และแบบที่มีการประมวลผลแบบกริด โดยเลือกใช้เทคนิคเครือข่ายใยประสาทในการรู้จำ และใช้อัลกอริทึมแบบ Backpropagation ในการเรียนรู้ ตัวอย่างที่ใช้ในการวิจัยการรู้จำคือ ภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ A-Z กำหนดให้มี ขนาด 100x100 พิกเซล และให้สีตัวอักษรและสีพื้นหลังมีความแตกต่างอย่างชัดเจน ภาพตัวอย่างที่ใช้จะผ่านการประมวลผลภาพเบื้องต้นเพื่อแปลงค่าสีเป็นสีขาว-ดำ และถูกส่งเข้าสู่ขั้นตอนการเรียนรู้ของเครือข่ายใยประสาท โดยการเรียนรู้ที่มีการประมวลผลแบบกริด จะทำการกระจายขั้นตอนการประมวลผลการเรียนรู้ไปยังหน่วยประมวลผลต่าง ๆ ที่อยู่บนกริด ทำให้เกิดการทำงานแบบกระจาย

ผลการวิจัยได้จากการเปรียบเทียบค่าเวลาที่ใช้ในการประมวลผลในขั้นตอนการเรียนรู้ของโปรแกรมประยุกต์ทั้ง 2 แบบ พบว่าการประมวลผลแบบกริดใช้เวลาในการประมวลผลน้อยกว่าการประมวลผลเพียงโปรเซสเดียว โดยถ้าแบ่งการทำงานออกเป็น 2 โปรเซส จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 1.59 ถึง 1.83 เท่า สำหรับการแบ่งออกเป็น 4 โปรเซส จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 2.27 ถึง 3.22 เท่า และถ้าแบ่งออกเป็น 8 โปรเซส จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 2.90 ถึง 6.33 เท่า ทั้งนี้ขึ้นกับตัวอักษรที่นำมาใช้เพื่อการเรียนรู้ ทั้งนี้ยังพบว่าการประมวลผลแบบกริดจะไม่เกิดประโยชน์เท่าที่ควรถ้าตัวอักษรที่นำมาใช้มีขนาดเล็กเกินไป

มหาวิทยาลัยศิลปากร ลงวันลงชื่อ

---

ภาควิชาคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ปีการศึกษา 2550
ลายมือชื่อนักศึกษา.....
ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์ .....

46307311 : MAJOR : COMPUTER SCIENCE
KEY WORD : RCOGNITION, NEURAL NETWORK, GRID COMPUTING, MPI
PAISAN SIMALAOTAO : RECOGNITIVE APPLICATION DEVELOPMENT USING GRID. THESIS ADVISOR : ASST.PROF.PANJAI TANTATSANAWONG,Ph.D... 125 pp.

The objective of the current research is to develop a recognition application, for measuring computation efficiency by using grid computing. Two types of the application development including single processor computing and grid computing with applied neural network technique and using backpropagation algorithm are employed. The examples used in this recognition research are pictures of English capital alphabets (A-Z) with a size of 100 x 100 pixels. The alphabets' color and the pictures' background must totally different and the pictures are pre-processing convert to white-black color. They submitted to recognition learning step of neural network by using grid computing, which distribute the recognition learning processes to other processor units on the grids.

The results based on the average time of the two types of execution indicate less time is used for using grid compared with the single process. The average computing time with two concurrent processes has increasing rate by a factor of 1.59 to 1.83 depending on the complexity of alphabets. For the four concurrent processes has the average of increasing rate by a factor of 2.27 to 3.22. Finally, the eight concurrent processes has average increasing rate by a factor of 2.90 to 6.33. The results show that the complexity of alphabets and number of processes impacts the computing time. In addition, grid technology has not benefit when too small alphabets are used.

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

Department of Computing Graduate School, Silpakorn University Academic Year 2007
Student's signature .....
Thesis Advisor's signature .....

# กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลงได้ด้วยความกรุณาอย่างสูงของผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำปรึกษา คำแนะนำในการแก้ไขข้อบกพร่องต่าง ๆ จนงานวิจัยนี้สำเร็จลงได้ด้วยดี ผู้วิจัยขอขอบพระคุณอย่างสูงมา ณ โอกาสนี้

ผู้วิจัยขอขอบพระคุณรองศาสตราจารย์ ดร.จันทนา ผ่องเพ็ญศรี ประธานกรรมการและผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ กรรมการผู้ทรงคุณวุฒิที่ได้กรุณาให้คำแนะนำในการแก้ไขข้อบกพร่องต่าง ๆ ในงานวิจัยนี้

ผู้วิจัยขอขอบพระคุณคณาจารย์ภาควิชาคอมพิวเตอร์ มหาวิทยาลัยศิลปากรทุกท่านที่ได้ให้ความรู้พร้อมทั้งคำแนะนำตลอดมา ตลอดจนเจ้าหน้าที่ประจำภาควิชาที่ได้อำนวยความสะดวกในการติดต่อประสานงาน

ผู้วิจัยขอขอบคุณมหาวิทยาลัยราชภัฏนครปฐม และคุณแม่ที่ให้ทุนสนับสนุนการศึกษาในระดับปริญญามหาบัณฑิต

ผู้วิจัยขอขอบคุณคณาจารย์กลุ่มโปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏนครปฐม เพื่อนร่วมงานทุกคน พี่ ๆ เพื่อน ๆ และน้อง ๆ ที่เป็นกำลังใจแก่ผู้วิจัยเสมอมา

และท้ายสุดนี้ ผู้วิจัยขอขอบพระคุณคุณพ่อ คุณแม่ และครอบครัวที่ห่วงใยและเป็นกำลังใจแก่ผู้วิจัยเสมอมา

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

**สารบัญ**


หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญตาราง ..... ญ

สารบัญรูป ..... ฏ

สารบัญแผนภูมิ ..... ฑ

บทที่

1 บทนำ ..... 1

ความเป็นมาและความสำคัญของปัญหา ..... 1

วัตถุประสงค์การวิจัย ..... 3

ขอบเขตการวิจัย ..... 3

ขั้นตอนการวิจัย ..... 4

ประโยชน์ที่คาดว่าจะได้รับ ..... 4

2 วรรณกรรมที่เกี่ยวข้อง ..... 6

วรรณกรรมเกี่ยวกับเทคนิคเครือข่ายใยประสาทในการรู้จำตัวอักษร ..... 6

วรรณกรรมเกี่ยวกับเทคโนโลยีกริดและการพัฒนาระบบแบบกระจาย ..... 8

3 ทฤษฎีที่เกี่ยวข้อง ..... 10

การรู้จำตัวอักษรด้วยเทคนิคเครือข่ายใยประสาท ..... 10

โครงสร้างของเพอร์เซปตรอน (Perceptron) ..... 11

เพอร์เซปตรอนแบบหลายชั้น (Multilayer perceptron) ..... 12

คุณสมบัติของเครือข่ายใยประสาท ..... 12

Back-propagation algorithm ..... 14

เทคโนโลยีกริด ..... 17

ความสามารถของ Computing Grid ..... 17

รูปแบบการใช้งานระบบ Computing Grid ..... 18

การเขียนโปรแกรมแบบขนานโดยการส่งผ่านข้อความด้วย MPI ..... 20

ลักษณะทั่วไปของ MPI ..... 20

โพรซิเดอร์และอาร์กิวเมนต์ ..... 21


มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

บทที่ หน้า
ตัวแปรอะเรย์และค่าคงที่ ... 22
ประเภทของตัวแปร ... 22
ฟังก์ชั่นในการส่งผ่านข้อความด้วย MPI ... 23
4 วิธีดำเนินการวิจัย ... 26
การเตรียมตัวอย่างสำหรับการรู้จำ ... 26
เครื่องมือและอุปกรณ์ ... 26
ซอร์ฟแวร์ที่ใช้ในการวิจัย ... 27
ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ ... 27
ขั้นตอนการนำเข้ารูปภาพสำหรับใช้ในการเรียนรู้ ... 28
ขั้นตอนการประมวลผลภาพเบื้องต้น ... 28
ขั้นตอนการเรียนรู้และการรู้จำตัวอักษรภาษาอังกฤษ ... 36
ขั้นตอนการวัดเวลาในการประมวลผลการเรียนรู้ ... 38
ขั้นตอนการทดสอบการรู้จำ ... 39
ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำโดยใช้การประมวลผลกริด ... 41
ขั้นตอนการนำเข้ารูปภาพสำหรับใช้ในการเรียนรู้ ... 43
ขั้นตอนการประมวลผลภาพเบื้องต้น ... 43
ขั้นตอนการเรียนรู้และการรู้จำตัวอักษรภาษาอังกฤษ ... 44
ขั้นตอนการวัดเวลาในการประมวลผลการเรียนรู้ ... 56
การประเมินผล ... 56
5 ผลการดำเนินงานวิจัย ... 58
ตัวอย่างภาพตัวอักษรที่ใช้ในการวิจัย ... 58
ผลการวัดประสิทธิภาพการประมวลผล ... 58
6 อภิปรายผล สรุป และข้อเสนอแนะ ... 61
อภิปรายผลการวิจัย ... 61
สรุปผลการวิจัย ... 63
ข้อเสนอแนะเกี่ยวกับงานวิจัย ... 64
แนวทางการพัฒนางานวิจัยในอนาคต ... 64
บรรณานุกรม ... 65
ภาคผนวก ... 67

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

บทที่ หน้า
ภาคผนวก ก การวัดประสิทธิภาพการประมวลผลการเรียนรู้ของตัวอักษร..... 68
ภาคผนวก ข คู่มือการใช้งานโปรแกรม. ..... 121
ประวัติผู้วิจัย ..... 125

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**สารบัญตาราง**


| ตารางที่ | | หน้า |
|---|---|---|
| 1 | ตารางเปรียบเทียบประเภทตัวแปรที่ใช้ใน MPI และภาษาซี……..... | 22 |
| 2 | รายละเอียดทรัพยากรที่ใช้ในการประมวลผลของ Tubkaew Grid Cluster ..... | 42 |
| 3 | สรุปผลการวัดประสิทธิภาพการประมวลผลการเรียนรู้..... | 59 |
| 4 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร A..... | 69 |
| 5 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร B..... | 71 |
| 6 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร C..... | 73 |
| 7 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร D..... | 75 |
| 8 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร E..... | 77 |
| 9 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร F..... | 79 |
| 10 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร G..... | 81 |
| 11 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร H..... | 83 |
| 12 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร I..... | 85 |
| 13 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร J..... | 87 |
| 14 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร K..... | 89 |
| 15 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร L..... | 91 |
| 16 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร M..... | 93 |
| 17 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร N..... | 95 |
| 18 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร O..... | 97 |
| 19 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร P..... | 99 |
| 20 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Q..... | 101 |
| 21 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร R..... | 103 |
| 22 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร S..... | 105 |
| 23 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร T..... | 107 |
| 24 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร U..... | 109 |
| 25 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร V..... | 111 |
| 26 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร W..... | 113 |
| 27 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร X..... | 115 |
| 28 | เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Y..... | 117 |


มหาวิทยาลัยศิลปากร ลิขสิทธิ์

ตารางที่ หน้า
29 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Z ..... 119

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**สารบัญรูป**


| รูปที่ | | หน้า |
|---|---|---|
| 1 | เพอร์เซปตรอน (Perceptron) | 11 |
| 2 | แสดงสถาปัตยกรรมของ Feed-forward multilayer perceptron | 13 |
| 3 | แสดงรูปแบบ Backpropagation Neural Network | 15 |
| 4 | Computing Grid | 19 |
| 5 | Grid Architecture for Computational Economy | 20 |
| 6 | ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ | 27 |
| 7 | ขั้นตอนการอ่านค่าจากไฟล์รูปภาพตัวอักษรภาษาอังกฤษ | 28 |
| 8 | ขั้นตอนการอ่านค่าสีในแต่ละพิกเซลเพื่อทำการรวมค่าสี | 29 |
| 9 | ขั้นตอนการปรับค่าสีเพื่อเพิ่มความแตกต่างระหว่างตัวอักษรกับพื้นหลัง | 30 |
| 10 | ขั้นตอนการแทนค่าตัวอักษร | 31 |
| 11 | แสดงขั้นตอนการหาค่าขอบเขตบนของตัวอักษร | 32 |
| 12 | แสดงขั้นตอนการหาค่าขอบเขตล่างของตัวอักษร | 33 |
| 13 | แสดงขั้นตอนการหาค่าขอบเขตซ้ายของตัวอักษร | 34 |
| 14 | แสดงขั้นตอนการหาค่าขอบเขตขวาของตัวอักษร | 35 |
| 15 | แสดงขั้นตอนการจำกัดขอบเขตของตัวอักษร | 36 |
| 16 | การอ่านค่าข้อมูลในชั้นนำเข้าข้อมูล | 37 |
| 17 | แสดงชั้นของ Multilayer Feed Forward Network | 38 |
| 18 | แสดงขั้นตอนการฝึกสอนตามแนวคิด Backpropagation Algorithm | 38 |
| 19 | ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ | 40 |
| 20 | โครงสร้างการทำงานของหน่วยประมวลผลหลายตัว (Multiprocessor) | 41 |
| 21 | โครงสร้างการเชื่อมต่อของ Front node และ Compute Node | 42 |
| 22 | แสดงการแบ่งส่วนการทำงานของการรู้จำที่ใช้การประมวลผลแบบกระจาย | 45 |
| 23 | แสดงลักษณะการประมวลผลในแต่ละโหนดของเครือข่ายใยประสาท | 45 |
| 24 | ตัวอย่างการส่งค่าระหว่างโหนดในชั้นข้อมูลเข้าไปยังชั้นซ่อนกรณีภาพตัวอักษร A | 47 |
| 25 | การกระจายตัวแปรชุดข้อมูลนำเข้าด้วย MPI_Bcast | 48 |
| 26 | การแบ่งข้อมูลจากตัวแปรชุดข้อมูลค่าถ่วงน้ำหนัก 2 มิติ เป็นตัวแปรชุด 1 มิติ | 49 |
| 27 | การกระจายตัวแปรชุดข้อมูลค่าน้ำหนักด้วย MPI_Scatter | 50 |
| 28 | การคำนวณของแต่ละโปรเซสที่กระจายกันประมวลผลตาม Rank ต่างๆ | 51 |


มหาวิทยาลัยศิลปากร ลิขสิทธิ์

| รูปที่ | | หน้า |
|---|---|---|
| 29 | การรับค่าตัวแปรชุดที่ส่งมาจาก Rank ต่างๆด้วย MPI_Gather ..... | 52 |
| 30 | การกระจายตัวแปรชุดข้อมูลเข้าด้วย MPI_Bcast..... | 54 |
| 31 | การกระจายตัวแปรชุดข้อมูลค่าผิดพลาดด้วย MPI_Scatter ..... | 55 |
| 32 | การประมวลผลเพื่อปรับค่าถ่วงน้ำหนัก..... | 56 |
| 33 | เข้าสู่โปรแกรมจะพบเมนูสำหรับเข้าใช้งานโปรแกรมประยุกต์การรู้จำ ..... | 122 |
| 34 | เลือกไฟล์ภาพเข้าสู่ระบบเพื่อเริ่มต้นการประมวลผลภาพเบื้องต้น ..... | 122 |
| 35 | เริ่มทำการประมวลผลการเรียนรู้ ..... | 123 |
| 36 | เมื่อประมวลผลเสร็จจะแสดงเวลาที่ใช้และค่าผิดพลาดที่ยอมรับได้..... | 123 |
| 37 | แสดงผลเมื่อทำการทดสอบการรู้จำ..... | 124 |


มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**สารบัญแผนภูมิ**


| แผนภูมิที่ | | หน้า |
|---|---|---|
| 1 | แสดงเวลาในการเรียนรู้ตัวอักษร A | 70 |
| 2 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร A | 70 |
| 3 | แสดงเวลาในการเรียนรู้ตัวอักษร B | 72 |
| 4 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร B | 72 |
| 5 | แสดงเวลาในการเรียนรู้ตัวอักษร C | 74 |
| 6 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร C | 74 |
| 7 | แสดงเวลาในการเรียนรู้ตัวอักษร D | 76 |
| 8 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร D | 76 |
| 9 | แสดงเวลาในการเรียนรู้ตัวอักษร E | 78 |
| 10 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร E | 78 |
| 11 | แสดงเวลาในการเรียนรู้ตัวอักษร F | 80 |
| 12 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร F | 80 |
| 13 | แสดงเวลาในการเรียนรู้ตัวอักษร G | 82 |
| 14 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร G | 82 |
| 15 | แสดงเวลาในการเรียนรู้ตัวอักษร H | 84 |
| 16 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร H | 84 |
| 17 | แสดงเวลาในการเรียนรู้ตัวอักษร I | 86 |
| 18 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร I | 86 |
| 19 | แสดงเวลาในการเรียนรู้ตัวอักษร J | 88 |
| 20 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร J | 88 |
| 21 | แสดงเวลาในการเรียนรู้ตัวอักษร K | 90 |
| 22 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร K | 90 |
| 23 | แสดงเวลาในการเรียนรู้ตัวอักษร L | 92 |
| 24 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร L | 92 |
| 25 | แสดงเวลาในการเรียนรู้ตัวอักษร M | 94 |
| 26 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร M | 94 |
| 27 | แสดงเวลาในการเรียนรู้ตัวอักษร N | 96 |
| 28 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร N | 96 |


มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

| แผนภูมิที่ | | หน้า |
|---|---|---|
| 29 | แสดงเวลาในการเรียนรู้ตัวอักษร O | 98 |
| 30 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร O | 98 |
| 31 | แสดงเวลาในการเรียนรู้ตัวอักษร P | 100 |
| 32 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร P | 100 |
| 33 | แสดงเวลาในการเรียนรู้ตัวอักษร Q | 102 |
| 34 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Q | 102 |
| 35 | แสดงเวลาในการเรียนรู้ตัวอักษร R | 104 |
| 36 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร R | 104 |
| 37 | แสดงเวลาในการเรียนรู้ตัวอักษร S | 106 |
| 38 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร S | 106 |
| 39 | แสดงเวลาในการเรียนรู้ตัวอักษร T | 108 |
| 40 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร T | 108 |
| 41 | แสดงเวลาในการเรียนรู้ตัวอักษร U | 110 |
| 42 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร U | 110 |
| 43 | แสดงเวลาในการเรียนรู้ตัวอักษร V | 112 |
| 44 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร V | 112 |
| 45 | แสดงเวลาในการเรียนรู้ตัวอักษร W | 114 |
| 46 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร W | 114 |
| 47 | แสดงเวลาในการเรียนรู้ตัวอักษร X | 116 |
| 48 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร X | 116 |
| 49 | แสดงเวลาในการเรียนรู้ตัวอักษร Y | 118 |
| 50 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Y | 118 |
| 51 | แสดงเวลาในการเรียนรู้ตัวอักษร Z | 120 |
| 52 | แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Z | 120 |


มหาวิทยาลัยศิลปากร ลิขสิทธิ์

# บทที่ 1
# บทนำ

ทรัพยากรที่มีความสำคัญอย่างหนึ่งของระบบคอมพิวเตอร์ คือ หน่วยประมวลผลกลาง (Central Processing Unit) ซึ่งทำหน้าที่ในการคำนวณ เปรียบเทียบ ประมวลผลข้อมูลหรือคำสั่งต่างๆ โดยทั่วไปคอมพิวเตอร์จะใช้หน่วยประมวลผลกลางในการประมวลผลเพียงครั้งละ 1 งานเท่านั้น และการใช้งานของผู้ใช้ทั่วไปจะไม่สามารถใช้งานหน่วยประมวลผลกลางได้อย่างเต็มประสิทธิภาพ โดยจะใช้งานพื้นฐานเป็นหลัก อาทิ งานพิมพ์เอกสาร ดูหนัง ฟังเพลง หรือแม้แต่เล่นเกม ซึ่งความสามารถในการประมวลผลจะถูกใช้เพียง10–40% ของหน่วยประมวลผลกลางเท่านั้น ในขณะที่หลาย ๆ คนหรือหลาย ๆ หน่วยงานจำเป็นต้องใช้ความสามารถในการประมวลผลสูงกว่าเพื่อทำงานเฉพาะด้านหรือวิเคราะห์ข้อมูลปริมาณมหาศาล ในปัจจุบันหลาย ๆ หน่วยงานจึงมีแนวคิดในการใช้หน่วยประมวลผลกลางแบบที่ใช้งานทั่วไปหลาย ๆ ตัวมาช่วยกันประมวลผลงานใดงานหนึ่ง เพื่อลดเวลาการประมวลผลประกอบกับแนวโน้มของหน่วยประมวลผลกลางมีราคาที่ถูกลงเรื่อย ๆ จึงมีการนำความสามารถนี้มาใช้สร้างซุปเปอร์คอมพิวเตอร์ที่มีราคาไม่สูงมากนัก งานวิจัยนี้ มีแนวคิดในการนำลักษณะการทำงานดังกล่าวมาประยุกต์กับความรู้ด้านปัญญาประดิษฐ์ ซึ่งเป็นการศึกษาถึงพฤติกรรมการเรียนรู้ และการคิดของมนุษย์ โดยเลือกใช้แนวคิดทางด้านการเรียนรู้และการรู้จำด้วยเทคนิคเครือข่ายใยประสาท (Neural Networks) เพื่อให้สามารถนำมาประยุกต์ใช้กับการพัฒนาระบบรู้จำและทำการแยกสิ่งต่าง ๆ ได้ด้วยความเร็วที่เพิ่มขึ้น

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

## 1. ความเป็นมาและความสำคัญของปัญหา

การทำงานของผู้ใช้งานทั่วไปส่วนใหญ่จะเป็นการใช้งานขั้นพื้นฐาน ทำให้การใช้งานหน่วยประมวลผลกลางไม่เต็มประสิทธิภาพ กลุ่มนักวิจัยหรือหน่วยงานต่าง ๆ จึงมีแนวคิดในการนำเอาความสามารถในการประมวลผลส่วนที่ยังไม่ถูกใช้งานมาใช้ เพื่อนำไปใช้ประมวลผลข้อมูลจำนวนมหาศาลทางด้านวิทยาศาสตร์ ทำให้ได้ซูเปอร์คอมพิวเตอร์ที่มีราคาไม่สูงมากนักมาใช้งาน โดยแนวความคิดดังกล่าวถือเป็นแนวคิดพื้นฐานของ "การประมวลผลแบบกริด" (Grid Computing) ซึ่งเป็นวิวัฒนาการของการประมวลแบบกระจาย (Distributed System) โดยมีจุดมุ่งหมายที่การ

แบ่งปันทรัพยากรในการทำงานและการประมวลผลในระดับใหญ่ (Large Scale Resource Sharing) ทรัพยากรส่วนใหญ่ คือ เครื่องคลัสเตอร์ (Cluster) หรือกลุ่มของเครื่องคอมพิวเตอร์ โดยเน้นเรื่องการบริหารและแบ่งปันทรัพยากรต่าง ๆ ภายในกลุ่มของเครื่องคอมพิวเตอร์ที่กระจายอยู่ตามสถานที่หรือหน่วยงานที่มีการเชื่อมต่อกันผ่านเครือข่ายให้ใช้งานร่วมกันได้ ซึ่งเป็นแนวคิดส่วนหนึ่งเพื่อให้สามารถรองรับการประมวลผลที่ต้องการประสิทธิภาพในการคำนวณสูง (High Performance Computing: HPC)

ทั้งนี้แนวโน้มในการพัฒนาระบบงานต่าง ๆ ให้มีความชาญฉลาด สามารถเรียนรู้ จดจำ เข้าใจและใช้ข้อมูลต่าง ๆ ในการแก้ไขปัญหาหรือแยกแยะสิ่งต่าง ๆ ได้อย่างมีประสิทธิภาพ เริ่มเข้ามามีบทบาทมากขึ้น การศึกษาเพื่อพัฒนาความชาญฉลาดให้กับระบบงาน เกิดขึ้นจากการศึกษาและจำลองพฤติกรรมการเรียนรู้ ตลอดจนการคิดของมนุษย์ซึ่งเป็นแนวคิดของ “ปัญญาประดิษฐ์ (Artificial Intelligence : AI)” ซึ่งแนวคิดตลอดจนเทคนิคต่าง ๆ ที่เกี่ยวข้อง เป็นการทำงานที่มีความยุ่งยากซับซ้อน และจำเป็นต้องมีการประมวลผลหลายครั้ง เพื่อให้ได้ผลลัพธ์ที่มีความเป็นไปได้มากที่สุด อีกทั้งยังต้องทำการเปรียบเทียบ เพื่อคัดเลือกให้ได้ผลลัพธ์ที่ดีที่สุดของแต่ละแนวคิดทำให้การนำแนวความคิดและเทคนิคต่าง ๆ มาประยุกต์ใช้กับงานบางอย่างเกิดความล่าช้าในการประมวลผลขึ้น แนวคิดที่สำคัญและได้รับความนิยมอย่างหนึ่งของปัญญาประดิษฐ์ คือ แนวคิดในเรื่องของ “การรู้จำ (Cognitive Sciences)” ซึ่งเป็นแนวคิดในการพัฒนาระบบคอมพิวเตอร์ให้สามารถเรียนรู้ จดจำ และพัฒนาการทำงานของตนเองได้ แนวคิดทางด้านการรู้จำ สามารถใช้เทคนิคในการทำงานได้มากมายหลายวิธี ซึ่งวิธีที่มีประสิทธิภาพในการทำงานวิธีหนึ่งคือ เทคนิคทางด้าน “เครือข่ายใยประสาท (Neural Networks)” ที่จำลองการทำงานของสมองมนุษย์ในการเรียนรู้และจดจำข้อมูลต่างๆ โดยสร้างเป็นเครือข่ายสำหรับการประมวลผลที่มีความสัมพันธ์กับค่าน้ำหนัก (Numeric Weight) และสามารถปรับปรุงค่าน้ำหนักของ เครือข่าย (Network Weight) หลังจากใส่รูปแบบข้อมูลสำหรับการเรียนรู้ให้แก่เครือข่ายในแต่ละครั้ง การเลือกใช้อัลกอริทึมแบบ back-propagation ในการฝึกแบบ Feed-Forward Neural Networks ค่าที่ได้รับจากเครือข่าย (Output) จะถูกนำไปเปรียบเทียบกับผลลัพธ์ที่คาดหวัง (Target) เพื่อทำการคำนวณหาค่าความผิดพลาด และส่งกลับเข้าสู่เครือข่ายเพื่อใช้ปรับปรุงค่าน้ำหนักต่อไปจาก แนวคิดดังกล่าวทำให้เครือข่ายนี้สามารถเรียนรู้ความสัมพันธ์ระหว่างข้อมูลที่ป้อนเข้ามากับรูปแบบที่กำหนด แต่ขั้นตอนการทำงานดังกล่าวทำให้สูญเสียเวลาในการประมวลผลมาก

จากเหตุผลดังกล่าว ผู้วิจัยจึงมีแนวคิดในการนำเทคนิคเครือข่ายใยประสาทมาประยุกต์ให้สามารถทำงานบนพื้นฐานของการประมวลผลแบบกริด เพื่อพัฒนาระบบงานที่มีประสิทธิภาพ

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

ในการทำงานสูงขึ้น มีการประมวลผลที่รวดเร็ว สามารถลดเวลาในการประมวลผลและเวลาในการเรียนรู้ลงได้ ตลอดจนมีการใช้ทรัพยากรต่าง ๆ ร่วมกันได้อย่างเหมาะสม

**2. วัตถุประสงค์การวิจัย**

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมประยุกต์ที่มีการเรียนรู้ของเครือข่ายใยประสาทโดยการประมวลผลแบบกริด เพื่อลดเวลาประมวลผลในขั้นตอนการเรียนรู้ โดยใช้ลักษณะของการกระจายงานไปประมวลผลที่เครื่องต่าง ๆ ในระบบกริด ซึ่งสามารถแยกจุดมุ่งหมายหลักได้ดังนี้

2.1 เพื่อศึกษาการทำงานของเทคนิคเครือข่ายใยประสาท (Neural Network) และการประมวลผลแบบกริด (Grid Computing)

2.2 เพื่อพัฒนาโปรแกรมประยุกต์การรู้จำโดยใช้เทคนิคเครือข่ายใยประสาทที่สามารถประมวลผลแบบกริดได้

2.3 ระบบเกิดการเรียนรู้ตัวอักษรภาษาอังกฤษ ที่มีการประมวลผลการเรียนรู้ทั้งแบบปกติ และแบบที่มีประมวลผลแบบกริด

2.4 เปรียบเทียบประสิทธิภาพการประมวลผลการเรียนรู้ระหว่างการประมวลผลแบบปกติ กับแบบที่ทำงานร่วมกับการประมวลผลแบบกริด

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ลิขสิทธิ์

**3. ขอบเขตการวิจัย**

การพัฒนาโปรแกรมประยุกต์การรู้จำโดยใช้การประมวลผลแบบกริด สามารถแบ่งขั้นตอนการศึกษาและพัฒนาเพื่อการวิจัยได้เป็น 3 ขั้นตอน คือ ขั้นตอนที่ 1 ศึกษาแนวคิดและเทคนิคของเครือข่ายใยประสาทและพัฒนาโปรแกรมประยุกต์การรู้จำ ที่มีการประมวลผลตามปกติ ขั้นตอนที่ 2 คือ ศึกษาแนวคิดเกี่ยวกับการประมวลผลแบบกริด โดยการกระจายงานด้วย MPI และขั้นตอนที่ 3 คือ การประยุกต์เพื่อพัฒนาโปรแกรมประยุกต์การรู้จำโดยใช้การประมวลผลแบบกริด โดยมีรายละเอียดดังนี้

3.1 ศึกษาแนวคิดและเทคนิคของเครือข่ายใยประสาท โดยมุ่งเน้นการใช้แนวคิดของ Backpropagation ในการเรียนรู้

3.2 พัฒนาโปรแกรมประยุกต์การรู้จำที่มีการประมวลผลแบบปกติเพื่อฝึกสอนและรู้จำโดย งานวิจัยนี้เลือกการรู้จำตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ A - Z จำนวน 26 ตัวอักษร โดยมี

คุณลักษณะที่สำคัญในการวิจัยคือ สีตัวอักษรและสีพื้นหลังมีความแตกต่างอย่างชัดเจน ขนาดของรูปภาพตัวอักษรที่ใช้ในการวิจัยมีขนาด 100 x 100 พิกเซล โดยเป็นไฟล์ภาพแบบ BMP

3.3 พัฒนาโปรแกรมประยุกต์การรู้จำที่มีการประมวลผลแบบกริด โดยทำการพัฒนาโปรแกรมที่มีการกระจายงานด้วย MPI เพื่อใช้งานทรัพยากรในการประมวลผลและหน่วยความจำของเครื่องต่าง ๆ

3.4 วัดประสิทธิภาพด้านความเร็วในการประมวลผล โดยเปรียบเทียบผลลัพธ์ด้านเวลาของการประมวลผลการเรียนรู้ของทั้ง 2 ระบบที่แตกต่างกัน

**4. ขั้นตอนการศึกษา**

ในงานวิจัยนี้สามารถแบ่งขั้นตอนในการศึกษาได้ดังนี้

4.1 เก็บรวมรวมข้อมูล ทฤษฎี และศึกษางานวิจัยต่างๆ ที่ใช้แนวคิดเกี่ยวกับเครือข่ายใยประสาท ด้วยการเรียนรู้แบบ Backpropagation

4.2 เก็บรวมรวมข้อมูล ทฤษฎี และศึกษางานวิจัยที่ใช้แนวคิดเกี่ยวกับการประมวลผลแบบกริด และการพัฒนาโปรแกรมแบบขนาน

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ลิขสิทธิ์

4.3 พัฒนาโปรแกรมประยุกต์การรู้จำด้วยเทคนิคเครือข่ายใยประสาท ที่มีการเรียนรู้แบบ Backpropagation ด้วยการประมวลผลแบบปกติ

4.4 พัฒนาโปรแกรมประยุกต์การรู้จำด้วยเทคนิคเครือข่ายใยประสาท ที่มีการเรียนรู้แบบ Backpropagation โดยมีการกระจายงานแบบขนานเพื่อการประมวลผลแบบกริด

4.5 วิเคราะห์ผลได้จากการวัดประสิทธิภาพด้านความเร็วในการประมวลผลการเรียนรู้ โดยเปรียบเทียบผลลัพธ์ด้านเวลาของการประมวลผลการเรียนรู้ของทั้ง 2 ระบบที่ใช้ในการทดลอง และหาค่าอัตราเร็วที่เกิดขึ้น

4.6 สรุปผลการทดลอง

4.7 รวบรวมข้อเสนอแนะ

**5. ประโยชน์ที่คาดว่าจะได้รับ**

ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย มีดังต่อไปนี้

5.1 สามารถนำเทคนิคและแนวคิดที่ได้มาประยุกต์ใช้ในการพัฒนาระบบงานอื่น ๆ ที่มีการทำงานด้วยเทคนิคเครือข่ายใยประสาท (Neural Network) ให้สามารถทำการเรียนรู้ได้เร็วขึ้น

5.2 สามารถนำเทคนิคและแนวคิดการทำงานบนระบบที่มีการประมวลผลแบบกระจายไปประยุกต์ใช้กับระบบที่มีการทำงานด้วยเทคนิคอื่น ๆ

5.3 ได้เรียนรู้หลักการ แนวคิด ตลอดจนเทคนิคต่าง ๆ เกี่ยวกับเครือข่ายใยประสาท (Neural Network) เพื่อการประยุกต์ใช้ในงานด้านต่าง ๆ

5.4 ได้เรียนรู้หลักการกระจายงานตามแนวคิดการทำงานแบบกระจายโดยใช้ MPI และการประมวลผลแบบกริด เพื่อประยุกต์ใช้ในงานด้านต่าง ๆ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# บทที่ 2
# วรรณกรรมที่เกี่ยวข้อง

งานวิจัยนี้มุ่งเน้นการประยุกต์ใช้องค์ความรู้ที่สำคัญ 2 ด้านให้สามารถทำงานร่วมกันได้ เพื่อการพัฒนาโปรแกรมประยุกต์ด้านการรู้จำให้มีประสิทธิภาพ วรรณกรรมที่เกี่ยวข้องจึงแบ่งออกเป็น 2 ส่วน คือ วรรณกรรมที่เกี่ยวกับเทคนิคเครือข่ายใยประสาทในการรู้จำตัวอักษร วรรณกรรมเกี่ยวกับเทคโนโลยีกริดและการพัฒนาระบบแบบกริด โดยมีรายละเอียดดังนี้

## 1. วรรณกรรมเกี่ยวกับเทคนิคเครือข่ายใยประสาทในการรู้จำตัวอักษร

วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับเทคนิคเครือข่ายใยประสาท จะมุ่งเน้นงานที่มีการใช้แนวคิดแบบ Backpropagation ในการเรียนรู้ เพื่อศึกษาถึงขั้นตอนและวิธีการดำเนินงานวิจัย โดยมีวรรณกรรมและงานวิจัยที่เกี่ยวข้อง ดังนี้

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

Sae-Tang and Methaste (2000) เป็นงานวิจัยเกี่ยวกับการรู้จำลายมือตัวอักษรภาษาไทยแบบออนไลน์ด้วยเทคนิคเครือข่ายใยประสาทแบบหลายชั้น (Multi Layer Perceptron : MLP) และเรียนรู้ด้วยแนวคิดแบบ Backpropagation งานวิจัยนี้ใช้ตัวอักษรภาษาไทย แยกเป็นพยัญชนะ 44 ตัว สระ 14 ตัว วรรณยุกต์ 5 ตัว สัญลักษณ์ 2 ตัว และตัวเลขไทย 10 ตัว โดยมองถึงปัญหาของภาษาไทย เช่น ช่องว่างระหว่างตัวอักษร ลายเส้นโค้ง กลม และการที่ตัวอักษรมีหัว เป็นต้น จากการจัดวางพยัญชนะ สระ และวรรณยุกต์ร่วมกันทำให้แบ่งระดับของตัวอักษรได้ 4 ระดับ ทั้งนี้ได้มีการวิจัยเกี่ยวกับตัวอักษรเขียนด้วย ทำให้เมื่อนำเข้าข้อมูลแล้วจะมีการประมวลผลเบื้องต้นคือ การตัดส่วนเกินที่เกิดจากการลากเส้นออก และการหาตำแหน่งของการหักมุม ตัวอักษรในแต่ละตัวจะถูกแบ่งออกเป็น 6 ส่วนคือ 3 ส่วนตามแนวตั้งและ 3 ส่วนตามแนวนอน โดยแต่ละส่วนจะถูกนำไปคำนวนเพื่อหาอัตราส่วนของพิกเซลสีดำ การประมวลผลจะนำข้อมูลในแต่ละเฟรมมาซ้อนกัน การทดลองจะมีการแบ่งพื้นที่ออกเป็น 3 ส่วน คือ บน กลาง และล่าง ตามตำแหน่งของพยัญชนะ สระ และวรรณยุกต์ จากนั้นให้กลุ่มผู้เข้าทดลองเขียนตัวอักษรและทำการทดสอบการรู้จำ ผลที่ได้สามารถจดจำตัวอักษรได้เป็นอย่างดี

กฤติกา วงศาวณิช และศุภชัย ตั้งบุญญะศิริ (2549) เรื่องระบบรู้จำป้ายทะเบียนรถ เป็นโครงงานที่ใช้การประมวลผลภาพ (Image Processing) สำหรับปรับปรุงข้อมูลภาพดิจิตอลโดยมีขั้นตอนคือการนำไฟล์รูปภาพชนิด BMP ที่เป็นภาพสีในโหมด RGB ซึ่งได้จากกล้องดิจิตอลเข้าสู่ระบบและอาศัยเทคนิคการแปลงภาพสีให้เป็นภาพขาว-ดำ (Threshold) เพื่อให้ง่ายต่อการแปลงค่าสีเป็นค่าตัวเลข 1,0 เพื่อนำไปประมวลผล จากนั้นทำการหาขอบภาพ (Edge Detection) โดยหาจากแนวตั้งและแนวนอน นำส่วนที่ได้มาทำการแบ่งภาพเป็นส่วนย่อย (Segmentation) และปรับปรุงคุณภาพของภาพ (Image Enhancement) เมื่อได้ขอบเขตที่ต้องการแล้วทำการเปลี่ยนข้อมูลภายในเป็นค่าตัวเลข 1 สำหรับตัวอักษร และเลข 0 สำหรับสีพื้นของป้ายทะเบียน นำค่าตัวเลขที่ได้เข้าสู่ขั้นตอนของเทคนิคเครือข่ายใยประสาท (Neural Network) สำหรับสอนให้คอมพิวเตอร์รู้จำลักษณะของตัวอักษรและตัวเลขเพื่อระบุหมายเลขทะเบียนรถยนต์ ขั้นตอนสุดท้ายคือการนำผลลัพธ์ที่ได้ไปสืบค้นในฐานข้อมูลเพื่อหาเจ้าของรถ

นราพงษ์ อภิรัตน์วรากุล และนเรศ เกื้อปัญญากูล (2548) การศึกษาหลักการการจดจำตัวอักษรไทยเป็นการศึกษาถึงขั้นตอนการจดจำตัวอักษร โดยรวบรวมข้อมูลที่เกี่ยวข้องจากแหล่งข้อมูลต่าง ๆ เพื่อให้ทราบถึงขั้นตอนต่าง ๆ โดยสรุปดังนี้ การจดจำตัวอักษรจะนำเข้าข้อมูลรูปภาพตัวอักษรที่ได้จากการสแกนหรือจากแหล่งภาพต่าง ๆ โดยภาพที่ใช้ควรเป็นภาพที่มีความแตกต่างระหว่างสีของตัวอักษรกับสีพื้นหลังอย่างชัดเจนเพื่อให้ได้ข้อมูลตัวอักษรที่แท้จริง ทำการแปลงค่าสีของภาพให้เป็นค่าตัวเลขทางดิจิตอล เพื่อให้ง่ายต่อการนำไปใช้งานหรือประมวลผลในระบบคอมพิวเตอร์ เมื่อได้ข้อมูลดังกล่าวจะทำการประมวลผลภาพเบื้องต้นได้แก่ การหาขอบภาพ การลดสัญญาณรบกวน การทดสอบการเอียง การทำให้ตัวอักษรบาง และการปรับขนาดตัวอักษร ซึ่งเมื่อผ่านขั้นตอนต่าง ๆ แล้วจำทำให้ได้ข้อมูลตามต้องการ จากนั้นนำข้อมูลเข้าสู่ขั้นตอนของการรู้จำโดยเลือกใช้เครือข่ายใยประสาท ผลการศึกษาและทดลองทำให้ทราบถึงขั้นตอนต่าง ๆ ในการรู้จำตัวอักษรและสามารถรู้จำตัวอักษรได้

ชนินทร์ ฮุนตระกูล (2543) เรื่อง การรู้จำตัวอักษรภาษาไทยและภาษาอังกฤษโดยใช้เครือข่ายใยประสาท เป็นงานวิจัยเพื่อรู้จำตัวอักษรจำนวน 94 ตัว แบ่งเป็น พยัชนะ สระ สัญลักษณ์และวรรณยุกต์ภาษาไทยรวม 68 ตัว และพยัญชนะภาษาอังกฤษจำนวน 26 ตัว โดยนำเข้าข้อมูลรูปภาพแบบขาวดำ (Bi-Level) ตัวอักษรจะได้จากการสแกนเอกสารที่พิมพ์ด้วยตัวอักษรในแบบ AngsanaUPC, BrowaliaUPC และ CordiaUPC กำหนดให้มีขนาด 16, 22, 36 พิกเซล และเป็นตัวอักษรแบบปกติ วิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA) เพื่อสร้างเมทริกของความแปรปรวนร่วม (Covariance Matrix) จากข้อมูลภาพที่ได้ ถูกนำไปใช้ในการบีบอัด

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

ข้อมูล และการสร้างภาพใบหน้าไอเกน (Eigen Faces) ทำการปรับขนาดภาพให้มีขนาด 64 x 64 พิกเซล เพื่อให้ได้เวกเตอร์ขนาด 4096 สำหรับส่งผ่านขั้นตอนการรู้จำซึ่งเลือกใช้เครือข่ายใยประสาทเทียมแบบเพอร์เซฟตรอนหลายชั้น (Multilayer Neural Network: MLP) หรือเครือข่ายใยประสาทแบบแพร่ย้อนกลับ (Feed-Forward Backpropagation Neural Network) ซึ่งเป็นเครือข่ายใยประสาทแบบที่มีผู้สอน (Supervised Learning) ซึ่งกำหนดให้มีชั้นของอินพุต, ชั้นซ่อนจำนวน 1 ชั้น และชั้นของเอาต์พุต และผลลัพธ์ที่ได้จะเป็นค่ารหัสแอสกี งานวิจัยชั้นนี้คำนึงถึงประสิทธิภาพในการรู้จำเท่านั้น

**2. วรรณกรรมเกี่ยวกับเทคโนโลยีกริดและการพัฒนาระบบแบบกระจาย**

Harwood Senyard and Minh (2004) จาก University of Melbourne Victoria, Australia เป็นงานวิจัยที่นำเสนอการพัฒนาให้เทคนิคเครือข่ายใยประสาทสามารถกระจายการประมวลผลไปยังเครื่องต่าง ๆ ที่เชื่อมต่อในระบบเครือข่ายแบบ Peer – to – Peer โดยใช้เครื่องมือ ANN Grid Toolset เพื่อช่วยในการกระจายทรัพยากรในการประมวลผล ซึ่งโครงสร้างการทำงานของ Toolset ดังกล่าวจะมีการทำงานของ Virtual Machine มาช่วยในการทำงานที่ชั้นกลาง ผลการวิจัยทำให้ระบบมีการประมวลผลการเรียนรู้เร็วขึ้น

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

Ribert Ennaji and Lecourtier (1999) เป็นงานวิจัยเกี่ยวกับการทดลองการเรียนรู้ด้วยเทคนิคเครือข่ายใยประสาท ที่มีการทำงานแบบ Multiple-Layer Perceptrons (MLP) โดยเปรียบเทียบระหว่างการเรียนรู้ของ MLP แบบเดี่ยวและแบบที่มีการกระจายงานซึ่งในงานวิจัยนี้ใช้ K-NN เมื่อทำการทดลองจะทำให้ได้ผลลัพธ์เกี่ยวกับค่าความผิดพลาด โดย MLP แบบเดี่ยวจะมีค่าความผิดพลาดมากกว่า K-NN เมื่อเทียบตามจำนวนครั้งของการเรียนรู้

Czauderna and Seiffert (2004) เป็นการทดสอบการประมวลผลเครื่องมือเครือข่ายใยประสาท ที่มีการกระจายงานบนระบบคลัสเตอร์แบบต่าง ๆ โดยระบุถึงปัจจัยที่มีผลต่อปัญหาในการทำงานคือ โทโปโลยีของเครือข่ายใยประสาทในเรื่องของขนาดและโครงสร้างการทำงานของแต่ละชั้น การควบคุมการเรียนรู้ในเรื่องของความถูกต้องและความซับซ้อนของการทำงาน การใช้ความสามารถในการใช้งานหน่วยประมวลผลและหน่วยความจำร่วมกัน และภาษาที่ใช้ในการเขียนโปรแกรมเพื่อเรียกใช้ไลบรารี่ที่จำเป็น ในการทดลองได้กำหนดสภาพแวดล้อมในการทดลองคือ 1) ชุดข้อมูลที่ใช้ในการทดลองและการทดสอบ เป็นชุดข้อมูลจากฐานข้อมูลรูปสัญลักษณ์ที่เขียนด้วยลายมือที่มีขนาด 32x32 พิกเซล จำนวน 3471 ภาพ โดยแบ่งเป็นตัวอย่างในการเรียนรู้ 2280 ภาพและตัวอย่างในการทดสอบ 1191 ภาพ 2) ฮาร์ดแวร์ที่ใช้เป็นระบบคอมพิวเตอร์ที่มีสถาปัตยกรรมแบบ

ขนานแตกต่างกันที่เชื่อมกันผ่านระบบเครือข่ายหลายประเภท เช่น SunV880 HP9000Superdome HP GS1280 และBeowulf Cluster เป็นต้น 3) ซอฟต์แวร์ที่ใช้ในงานวิจัยนี้เลือกใช้ภาษา C ในการพัฒนาและใช้การการกระจายงานด้วย MPI ซึ่งเป็นการรับส่งข้อความที่สามารถทำงานได้ทุกรูปแบบ สามารถทำงานได้ดีบนระบบขนานที่มีการกระจายหน่วยประมวลผลและหน่วยความจำ การรู้จำได้เลือกใช้การเรียนรู้แบบ MLP ในส่วนของการทำงานแบบขนานได้เลือกการแบ่งงานอย่างง่าย เนื่องจากการแบ่งงานที่ซับซ้อนจะทำให้เสียเวลา ตลอดจนเรื่องการสืบทอดลักษณะการเรียนรู้ให้ทำงานแบบกระจายได้ สำหรับงานวิจัยนี้ได้มีการเลือกใช้แนวคิดในการกระจายงาน โดยการให้หน่วยประมวลผลรับงานของแต่ละโหนดไปประมวลผล เนื่องจากการทำงานของแต่ละโหนดมีความเป็นอิสระต่อกัน ทำให้สามารถขยายการทำงานออกไปได้ ผลการทดลองจะวัดจากความแตกต่างด้านประสิทธิภาพของเครื่องคอมพิวเตอร์แบบขนานที่มีหน่วยประมวลผลตั้งแต่ 1 ถึง 8 ตัว โดยใช้การกำหนดระยะการเรียนรู้ด้วยซิกมอยด์ฟังก์ชั่น ซึ่งพบว่ากลุ่มเครื่องที่เป็น SMP และ Beowulf Clusters Pentium IV ที่เชื่อมต่อด้วย Myrinet ได้ผลลัพธ์ที่ดีสามารถลดเวลาในการเรียนรู้ได้โดย SMP ทั้ง 3 กลุ่มที่ใช้ทดลองสามารถใช้งานหน่วยประมวลผลได้ครบทั้ง 8 ตัวโดยมีเวลาที่ลดลงตามลำดับและ Beowulf Cluster Pentium IV ที่เชื่อมต่อด้วย Ethernet สามารถใช้งานหน่วยประมวลผลได้ดีที่ 7 ตัวโดยมีเวลาที่ลดลงตามลำดับแต่การใช้หน่วยประมวลผลตัวที่ 8 ใช้เวลาเพิ่มขึ้น ส่วน Beowulf Clusters Pentium III และ Beowulf Clusters IV ที่เชื่อมต่อด้วย Ethernet ไม่เหมาะสำหรับการกระจายงานการเรียนรู้แบบ MLP เพราะว่าจะสามารถใช้งานหน่วยประมวลผลได้ดีที่สุดที่ 3 ตัวถ้าเมื่อเพิ่มหน่วยประมวลผลจะใช้เวลาในการประมวลผลเพิ่มขึ้น ซึ่งสิ่งที่ทำให้เกิดผลดังกล่าวเกิดจาก 2 กรณีคือ ศักยภาพในการติดต่อสื่อสารของโหนดต่าง ๆ เนื่องจากข้อมูลที่ส่งมีขนาดค่อนข้างเล็กซึ่งเล็กกว่า 8 กิโลบิตและเครื่องที่มีการเชื่อมต่อด้วย Ethernet ไม่สามารถชดเชยเวลาที่เสียไปในการติดต่อสื่อสารได้

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

# บทที่ 3
# ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้ได้เลือกใช้เทคนิคเครือข่ายใยประสาท (Neural Network) ในการรู้จำตัวอักษรภาษาอังกฤษ A – Z และเลือกใช้แนวคิดการประมวลผลแบบกริด (Grid Computing) ซึ่งเป็นลักษณะการประมวลผลแบบกระจาย ในการเพิ่มความสามารถการประมวลผลในขั้นตอนของการเรียนรู้หรือการฝึกสอนให้เร็วขึ้น โดยมีรายละเอียดเกี่ยวกับทฤษฎีต่าง ๆ ดังต่อไปนี้

## 1. การรู้จำตัวอักษรด้วยเทคนิคเครือข่ายใยประสาท

(B.Muller, J.Reinhardt, M.T.Strickland 1995 : 3 - 23)

หลักการของเครือข่ายใยประสาท (Neural Networks) คือการจำลองรูปแบบการทำงานของเซลล์สมองมนุษย์ที่เรียกว่า ตัวเซลล์ (Cell Body) หรือนิวรอล (Neural) สมองของมนุษย์เปรียบได้กับเครือข่ายของแต่ละเซลล์ประสาท ซึ่งประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า เดนไดรท์ (Dendrite) และปลายในการส่งกระแสประสาทเรียกว่า แอคซอน (Axon) เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอก หรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอนของมัน เซลล์ประสาทในสมองมนุษย์สามารถทำงานได้หลากหลายและมีความสลับซับซ้อนและทำงานได้เร็ว

เครือข่ายใยประสาทเป็นเครือข่ายที่มีรูปแบบในการประมวลผลที่ประกอบด้วย โหนด (Node) โดยจะมีค่าตัวเลขสำหรับการกระตุ้นในแต่ละโหนด การเชื่อมต่อระหว่างโหนดในเครือข่ายจะมีค่าตัวเลขที่เรียกว่า ค่าถ่วงน้ำหนัก (Weight) สิ่งเหล่านี้เป็นพื้นฐานของอัตราการส่งค่าของแต่ละโหนด และความแข็งแกร่งของไซแนป (Synapse) ที่อยู่ระหว่างโหนด 2 โหนด

สำหรับในคอมพิวเตอร์ได้มีการจำลองการทำงานของเซลล์ประสาทเทียม โดยอาศัยโครงสร้างที่ประกอบไปด้วยหน่วยประมวลผลที่เรียกว่า นิวรอล (Neural) จำนวนมากเชื่อมต่อกัน (Connection) ทำให้การทำงานของเครือข่ายใยประสาทเป็นไปในรูปแบบของการประมวลผลจำนวนมากพร้อม ๆ กัน

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

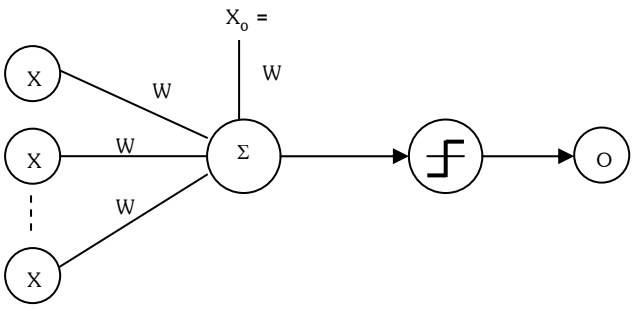
เซลล์ประสาทเทียม (Artificial Neural) หรือยูนิต (Unit) ประกอบด้วย ส่วนนำเข้าข้อมูล (Input) และส่วนผลลัพธ์ (Output) โดยการเชื่อมต่อถึงกันของการประมวลผลนั้นจะมีค่าน้ำหนัก (Weight) ของแต่ละการเชื่อมต่อ และเมื่อมีการให้ตัวอย่างที่ใช้ในการเรียนรู้เครือข่ายใยประสาทจะทำการปรับค่าน้ำหนักให้เหมาะสม จนได้ผลลัพธ์ที่ถูกต้องหรือมีข้อผิดพลาดน้อยที่สุด และสามารถนำค่าน้ำหนักที่ ถูกปรับนี้ไปใช้ในการเรียนรู้ครั้งต่อไปได้ ถ้าค่าน้ำหนักมีค่ามากกว่าค่าที่กำหนดนิวรอนจะส่ง ส่วนผลลัพธ์ออกไปยังส่วนนำเข้าข้อมูลของนิวรอลอื่น ๆ ที่เชื่อมกันในเครือข่าย ถ้าค่าน้ำหนัก น้อยกว่าค่าที่กำหนดก็จะไม่เกิดส่วนผลลัพธ์ของนิวรอนนั้น การรู้จำของคอมพิวเตอร์ทำได้โดยการให้เครื่องคอมพิวเตอร์ปรับค่าน้ำหนักที่ใช้เหล่านั้น ด้วยการสอนให้รู้จักรูปแบบของสิ่งที่ต้องการให้รู้จำ โดยมีการรู้จำที่เรียกว่า "Backpropagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ซึ่งการสอนต้องทำซ้ำ หลายๆ ครั้ง และต้องสอนให้รู้จักหลาย ๆ รูปแบบของแต่ละสิ่ง ยิ่งสอนมากเท่าไหร่ค่าน้ำหนัก ก็จะมีความถูกต้องมากขึ้นเท่านั้น

### 1.1 โครงสร้างของเพอร์เซปตรอน (Perceptron)

(Khanna, Tarun. 1990 : 50)

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

เพอร์เซปตรอน เป็นเครือข่ายใยประสาทที่มีหน่วยที่ใช้ทำงานตามลักษณะของเซลล์ประสาทอย่างง่าย โดยทำการรับค่าข้อมูลเข้าเป็นข้อมูลแบบเวกเตอร์ (Vector) จำนวนจริง $x_1$, $x_2$,..., $x_n$ แล้วคำนวณหาผลรวมกับค่าน้ำหนักของค่าข้อมูลเข้าโดยให้ค่า $w_1$, $w_2$,..., $w_n$ เป็นค่าน้ำหนักของข้อมูลเข้าและ $w_0$ เป็นค่า bias และ $x_0$ เป็นข้อมูลเข้าเทียมซึ่งกำหนดให้เป็น 1 เสมอ ผลรวมที่ได้จะถูกคำนวณในฟังก์ชั่นกระตุ้นเพื่อหาค่าผลลัพธ์ (o)




รูปที่ 1 เพอร์เซปตรอน (Perceptron)

เมื่อได้ผลลัพธ์ (Output) จะถูกส่งผ่านฟังก์ชันกระตุ้น (Activation Function) ซึ่งมีทั้งชนิดฟังก์ชันสองขั้ว (Bipolar Function) ที่ให้ผลลัพธ์เป็น 1 กับ -1 และฟังก์ชันไบนารี (Binary Function) ที่ให้ผลลัพธ์เป็น 1 กับ 0 เช่น ค่าที่ได้จะเข้าใกล้ 1 ถ้าผลรวมที่ได้มีค่าเกินค่าที่ยอมรับได้ และเข้าใกล้ 0 ถ้าค่าที่ได้ไม่เกินค่าที่ยอมรับได้

เอาต์พุตได้จากฟังก์ชันในรูปของผลรวมเชิงเส้นแบบถ่วงน้ำหนัก ซึ่งเป็นตัวกำหนดว่าอินพุตตัวใดที่มีความสำคัญในการกำหนดค่าผลลัพธ์ โดยดูจากค่าสัมบูรณ์ของค่าเวกเตอร์ถ่วงน้ำหนัก ซึ่งสามารถหาได้โดยใช้กฎการเรียนรู้เพอร์เซปตรอน (Perceptron Learning Rule) ซึ่งทำงานโดยสุ่มค่าเวกเตอร์ถ่วงน้ำหนัก เพื่อใช้ในการเปรียบเทียบระหว่างเพอร์เซปตรอนกับตัวอย่างการทำ งานจากนั้นนำค่าผลลัพธ์ที่ได้ไปเทียบกับผลลัพธ์เป้าหมาย ถ้าค่าไม่ตรงกันสามารถปรับค่าน้ำหนัก ตามอัตราส่วน การเรียนรู้ซึ่งจะเป็นตัวเลขที่มีค่าน้อยและมีผลต่อความเร็วในการเรียนรู้

**1.2 เพอร์เซปตรอนแบบหลายชั้น (Multilayer perceptron)**

จากลักษณะการทำงานของเพอร์เซปตรอนสามารถทำงานได้ในขอบเขตที่จำกัด จึงได้มีการนำเพอร์เซปตรอนหลาย ๆ ตัวมาทำการเชื่อมต่อกัน เพื่อสร้างเป็นเครือข่ายใยประสาทหลายชั้น (Multilayer Neural Network) ซึ่งทำให้สามารถทำงานและแสดงผลลัพธ์ในแบบ Linear Decision Surface และ Non – Linear Decision Surface การทำงานของเครือข่ายใยประสาทจะมีการทำงานโดยให้ผลลัพธ์ของเพอร์เซปตรอนหนึ่งเพอร์เซปตรอน หรือหลาย ๆ เพอร์เซปตรอนถูกส่งไปเป็นส่วนนำเข้าข้อมูลของเพอร์เซปตรอนต่อไป ซึ่งทำหน้าที่ในการประมวลผลโดยใช้ข้อมูลที่ได้รับมาและผลลัพธ์ที่ได้จะนำไปเปรียบเทียบน้ำหนักของเวกเตอร์น้ำหนักที่เหมาะสม ทั้งนี้การคำนวณการเรียนรู้สำหรับข่ายงานหลายชั้นต้องใช้ฟังก์ชันกระตุ้นที่หาอนุพันธ์ได้ ซึ่งอาจเป็นฟังก์ชันเชิงเส้นหรือฟังก์ชันซิกมอยด์ (Sigmoid Function)

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

**1.3 คุณสมบัติของเครือข่ายใยประสาท**

(LiMin Fu 1994 : 18-19)

โทโปโลยีของเครือข่ายใยประสาทมีเค้าโครงการการทำงานที่เชื่อมต่อกัน ซึ่งสามารถแบ่งออกเป็นระดับชั้นต่าง ๆ และในแต่ละชั้นจะประกอบไปด้วยการทำงานของโหนดย่อย ๆ โดยระดับชั้นต่าง ๆ มีรายละเอียดดังนี้

- ชั้นข้อมูลเข้า (Input Layer) เป็นชั้นเริ่มต้นโดยค่าของโหนดจะเป็นตัวอย่างที่ส่งเข้ามาในเครือข่ายเพื่อใช้เริ่มต้นการประมวลผล โดยจะถูกกระจายไปยังโหนดต่าง ๆ ที่อยู่ในชั้นซ่อน

- ชั้นซ่อน (Hidden Layer) เป็นชั้นสำหรับการประมวลผลโดยอยู่ระหว่างชั้นข้อมูลเข้าและชั้นผลลัพธ์ซึ่งชั้นนี้สามารถมีได้มากกว่า 1 ชั้น

- ชั้นผลลัพธ์ (Output Layer) เป็นชั้นที่จะได้ผลลัพธ์จากการคำนวนของเครือข่ายใยประสาท

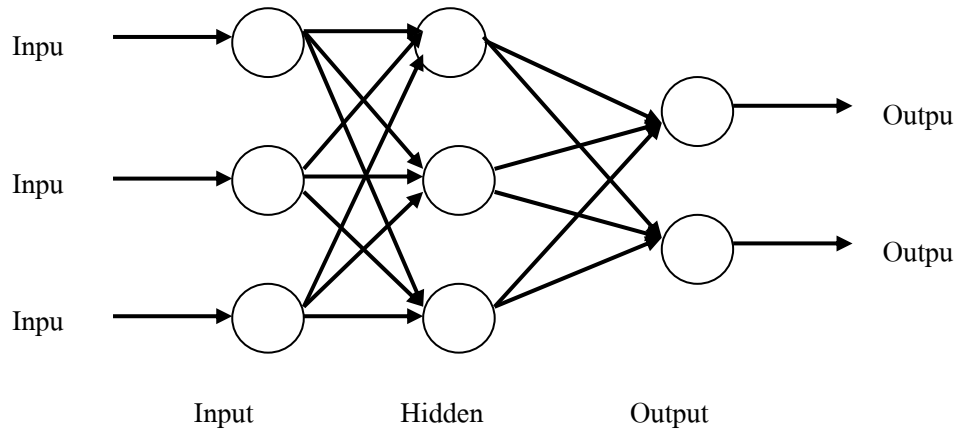
การทำงานของเครือข่ายใยประสาทสามารถเป็นได้ทั้งการส่งค่าไปด้านหน้า (Feed-Forward) หรือการส่งค่าย้อนกลับ (Recurrent) ก็ได้ ซึ่งมีรายละเอียดดังนี้

- เครือข่ายที่มีการส่งค่าไปด้านหน้า (Feed-Forward Network) ทุก ๆ โหนดที่เชื่อมต่อกันจะมีการส่งค่าไปในทิศทางเดียวกัน คือ ส่งจากชั้นข้อมูลเข้าไปยังชั้นซ่อนและชั้นผลลัพธ์ตามลำดับ

- เครือข่ายที่มีการส่งค่าย้อนกลับ (Recurrent Network) มีการส่งค่าย้อนกลับไปยังระดับชั้นก่อนหน้า หรือเกิดการวนรอบการทำงานของเครือข่าย

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

องค์ประกอบหลักของเครือข่ายใยประสาทแบบ Feed-Forward ได้แก่ ชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นผลลัพธ์ (Output Layer) ซึ่งจะมีการเชื่อมต่อระหว่างชั้นต่างๆ โดยโหนดในชั้นข้อมูลเข้าจะส่งสัญญาณไปยังทุก ๆ โหนดในชั้นซ่อน และทุกโหนดในชั้นซ่อนจะส่งสัญญาณไปยังทุกๆโหนดในชั้นซ่อนถัดไป (ในกรณีที่มีชั้นซ่อนมากกว่า 1 ชั้น) จนกระทั่งถึงชั้นซ่อนชั้นสุดท้ายซึ่งจะส่งสัญญาณไปยังทุกๆโหนดในชั้นผลลัพธ์



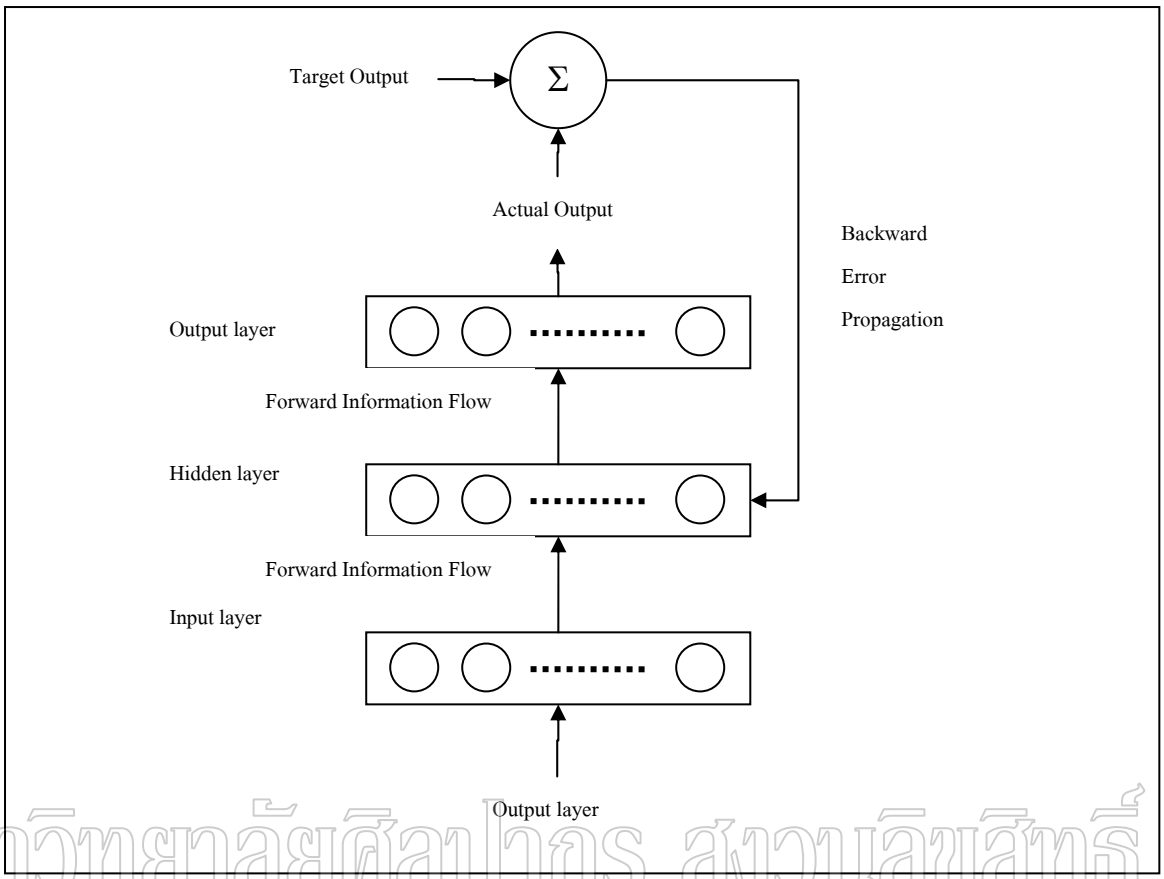

รูปที่ 2 แสดงสถาปัตยกรรมของ Feed-Forward Multilayer Perceptron

**1.4 Back-Propagation Algorithm**

(Fu, LiMin. 1994 : 80-84)

Backpropagation เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครือข่ายใยประสาทแบบที่มีการสอน (Supervised Learning) เป็นวิธีหนึ่งที่นิยมใช้ในเพอร์เซปตรอนแบบหลายชั้น (Multilayer Perceptron) เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้เหมาะสม โดยการปรับค่านี้จะ ขึ้นกับความแตกต่างของค่าผลลัพธ์ที่คำนวณได้กับค่าผลลัพธ์ที่ต้องการ ด้วยวิธีการประมวลผลเป็นชั้น ๆ โดยการนำชุดของข้อมูลเข้าและชุดของข้อมูลผลลัพธ์เป้าหมายที่ต้องการ (Target Output) ซึ่งเรียกรวมกันว่า ชุดการสอนควบคู่ (Training Pair) ป้อนเข้าสู่ระดับชั้นของข้อมูลเข้าผ่านระดับชั้นซ่อน จนถึงระดับชั้นผลลัพธ์ โดยในขั้นตอนเริ่มต้นนั้นค่าน้ำหนักเชื่อมโยง (Connection Weights) ทั้งหมดจะถูกกำหนดขึ้นโดยใช้วิธีการสุ่มค่าขึ้นมา (Random) และทำการป้อนเข้าสู่เครือข่ายใยประสาท หลังจากนั้นจะทำการคำนวณค่าของผลลัพธ์จริงที่ได้เปรียบเทียบกับผลลัพธ์เป้าหมายที่ต้องการเพื่อหาค่าผิดพลาด โดยค่าผิดพลาดที่คำนวณได้จะถูกป้อนส่งย้อนกลับเข้าไปในระดับชั้นก่อนหน้า ดังนั้น ถ้าสามารถคำนวณค่าผิดพลาดที่ข้อมูลออกของแต่ละระดับชั้นได้ ก็จะทำให้สามารถลดค่าผิดพลาดให้เหลือน้อยที่สุดของแต่ละระดับชั้นได้ แนวคิดของ Backpropagation มีรายละเอียดดังต่อไปนี้

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 3 แสดงรูปแบบ Backpropagation Neural Network

ที่มา : Fu, LiMin, Neural Network in Computer Intelligence (Singapore: McGraw Hill, 1994), 81.

1.4.1 กำหนดค่าน้ำหนัก (Weight Initialization)

กำหนดค่าน้ำหนักทุก ๆ ค่าที่ต้องใช้ และค่า Thresholds ของแต่ละโหนดที่ได้จากการสุ่มค่า ตัวเลขที่มีค่าน้อย ๆ โดยค่า Thresholds จะเป็นค่าติดลบของค่าน้ำหนักที่ได้จาก Bias (ในส่วนของ Activation Level จะกำหนดให้เป็น 1)

1.4.2 ประมวลผลการกระตุ้น (Calculation of Activation)

- Activation Level ในชั้นนำเข้า (Input Unit) จะเป็นการตัดสินใจโดยการนำตัวอย่างเข้าสู่เครือข่าย

- Activation Level ในชั้นซ่อน (Hidden Unit) ที่ $O_j$ และชั้นผลลัพธ์ (Output Unit) จะได้จากการตัดสินใจโดย

$$O_j = F(\sum W_{ji} O_i - \theta_j) \quad \text{.....(3.1)}$$

เมื่อ $W_{ji}$ เป็นค่าน้ำหนักจากส่วนนำเข้า $O_i$

$\theta_j$ เป็นค่า Thresholds ของแต่ละโหนด

F เป็นซิกมอยด์ฟังก์ชั่น (Sigmoid Function)

$$F(a) = \frac{1}{1+ e^{-a}}, \quad \text{..... (3.2)}$$

1.4.3 การปรับค่าน้ำหนัก (Weight Training)

- เริ่มต้นที่ชั้นผลลัพธ์ (Output) และทำงานย้อนกลับไปสู่ชั้นซ่อน (Hidden) อีกครั้ง ซึ่งจะมีการปรับค่าน้ำหนัก

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \quad \text{.....(3.3)}$$

มหาวิทยาลัยศิลปากร ลงวันลิขสิทธิ์

เมื่อ $W_{ji}$ เป็นค่าน้ำหนักตั้งแต่ i ถึง j ที่เวลา t (หรือวนซ้ำ t ครั้ง)

$\Delta W_{ji}$ เป็นค่าน้ำหนักที่ได้จากการปรับค่า

- การเปลี่ยนแปลงค่าน้ำหนักสามารถหาได้จากการคำนวนโดย

$$\Delta W_{ji} = \eta \delta_j O_i \quad \text{..... (3.4)}$$

เมื่อ $\eta$ เป็นค่าการเรียนรู้อิสระ (Learning Rate) ($0<\eta<1$)

$\delta_j$ เป็นค่าผิดพลาดของ j

1.4.4 การหาค่าผิดพลาด

- สำหรับชั้นผลลัพธ์ (Output Unit)

$$\delta_j = O_j (1-O_j) (T_j - O_j) \quad \text{..... (3.5)}$$

เมื่อ Tj เป็นผลลัพธ์ที่คาดหวังหรือเป้าหมาย

Oj เป็นผลลัพธ์ที่คำนวนได้จากการกระตุ้นที่ j

- สำหรับชั้นซ่อน (Hidden Unit)

$$\delta_j = O_j (1-O_j) \sum \delta_k W_{kj} \quad \text{..... (3.6)}$$

เมื่อ $\delta_k$ เป็นค่าผิดพลาดจากหน่วยที่ k ถึงแต่ละจุดเชื่อมต่อจากชั้นซ่อนหน่วยที่ j

1.4.5 ทำการวนซ้ำการทำงาน จนกว่าค่าที่ได้จะมีค่าใกล้เคียงกับค่าผิดพลาดที่เลือกไว้ การวนซ้ำจะเป็นการแสดงค่าการตัดสินใจ การคำนวนค่าการกระตุ้น และการปรับค่าน้ำหนัก

**2. เทคโนโลยีกริด**

การประมวลผลกริด (Grid Computing) ได้มุ่งเน้นถึงแนวความคิดของการใช้ทรัพยากรของการประมวลผลหรือหน่วยสำรองข้อมูลร่วมกันระหว่างองค์กรที่มีความตกลงกัน เพื่อทำให้เกิดการใช้งานทรัพยากรเหล่านั้นได้เต็มประสิทธิภาพ (Foster และ คณะ, 2000) Grid Computing เป็นวิวัฒนาการของ Distributed Computing โดยมีจุดมุ่งหมายที่การแบ่งปันทรัพยากรในการประมวลผลในระดับใหญ่ (large scale resource sharing) เพื่อที่จะรองรับการประมวลผลที่ต้องการประสิทธิภาพในการคำนวณสูง ( High Performance Computing; HPC )

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

**2.1 ความสามารถของ Computing Grid**

กริดเป็นเทคโนโลยีที่กำลังได้รับความนิยม โดยมีการทำความร่วมมือระหว่างองค์กรต่าง ๆ เพิ่มขึ้น เห็นได้จากซอต์ฟแวร์ต่าง ๆ ได้เริ่มมีการเพิ่มความสามารถในการทำงานร่วมกับกริดมากขึ้น ทั้งนี้ส่วนหนึ่งเนื่องจากความสามารถของกริดที่มีอยู่มากกมายนั่นเอง ซึ่งผู้วิจัยได้เลือกใช้การประมวลผลแบบ กริดเนื่องจากความสามารถของกริด ดังนี้

การใช้ประโยชน์จากทรัพยากรที่ไม่ได้ใช้งาน เช่น การเข้าใช้หน่วยประมวลผลกลางของเครื่องที่ไม่ได้ถูกใช้งานหรือใช้งานหน่วยประมวลผลกลางไม่เต็มประสิทธิภาพ ซึ่งอาจเป็นเครื่องที่มีการทำงานที่แตกต่างกัน เมื่อมีหน่วยประมวลผลจำนวนมากก็สามารถแบ่งงานออกเป็นส่วนย่อยเพื่อกระจายไปประมวลผลยังหน่วยประมวลผลต่าง ๆ โดยกำหนดให้งานมีความเป็นอิสระต่อกัน ทำให้เกิดประสิทธิภาพในการประมวลผลสูง นอกจากนี้ การประมวลผลแบบกริดยังจำกัดการเข้าใช้ทรัพยากรเพื่อควบคุมระบบให้สามารถทำงานได้อย่างสอดคล้องกันได้ การเข้าใช้ทรัพยากรทางด้านการประมวลผลอาจสามารถทำได้โดยการนำงานไปทำงานบนเครื่องต่าง ๆ ที่อยู่

ภายในระบบ หรือนำงานที่สามารถทำงานบนระบบที่มีการประมวลผลแบบขนานมาทำงานบนเครื่องที่มีหน่วยประมวลผลที่แตกต่างกันได้

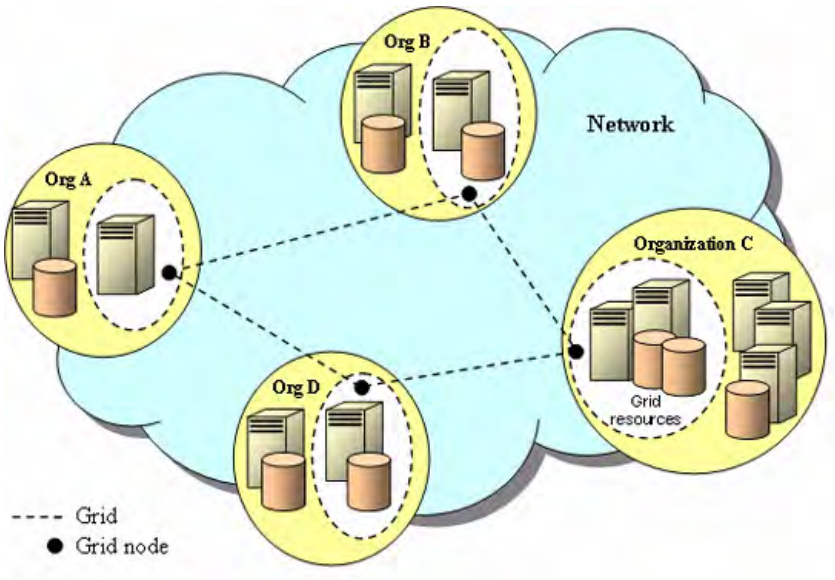
การทำงานของกริดจะมีมุมมองทางด้านทรัพยากรเสมือนว่าทุก ๆ เครื่องทำงานเป็นเครื่องเดียวกัน หรืออยู่ในองค์กรเดียวกัน ซึ่งถือเป็นองค์กรเสมือนทำให้แต่ละเครื่องมีทรัพยากรเสมือนขนาดใหญ่เป็นของตนเอง เช่น หน่วยประมวลผลกลาง ทำให้สามารถใช้หน่วยประมวลผลที่มีความสามารถสูงได้ ซึ่งเป็นลักษณะของการประมวลผลแบบกระจาย (Distributed Computing) ทั้งนี้ผู้ใช้ไม่จำเป็นต้องรู้ว่าข้อมูลที่ทำการประมวลผลถูกส่งมาจากเครื่องใด หรือส่งไปประมวลผลที่เครื่องใด กรณีของข้อมูลที่ทำงานบนกริดระบบจะมองข้อมูลทั้งหมดเป็นข้อมูลเดียวกันที่มีขนาดใหญ่ ซึ่งเรียกว่า “Data Grid” และไม่ว่าจะใช้อุปกรณ์ชนิดใด ระบบปฏิบัติการค่ายใด หรือมีข้อมูลรูปแบบต่างกันก็สามารถร้องขอ และใช้ข้อมูลร่วมกันได้ โดยจะต้องมีข้อตกลงที่จะใช้ข้อมูลร่วมกัน นอกจากนี้ในแต่ละเครื่องที่อยู่ในระบบจะบอกถึงจำนวนของหน่วยความจำที่มีอยู่ และที่สามารถให้เข้าใช้ได้ การเข้าใช้ข้อมูลหรือไฟล์ข้อมูลในระบบกริดอาจใช้การจัดการไฟล์ด้วยวิธีต่าง ๆ ของ Mountable Network File System เข้ามาช่วยทำให้มีประสิทธิภาพที่สูงขึ้น เช่น Andrew File System (AFS), Network File System (NFS), Distributed File System (DFS) หรือ General Parallel File System (GPFD) เป็นต้น

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

สามารถจัดตารางการใช้งานทรัพยากรอย่างมีประสิทธิภาพ และสามารถเพิ่มทรัพยากรต่าง ๆ ได้ตามต้องการ เช่น การเพิ่มจำนวนหรือความเร็วในการประมวลผลโดยการเพิ่มหน่วยประมวลผล หรือการเพิ่มความจุหน่วยความจำซึ่งจะทำให้เกิดความน่าเชื่อถือ รวมทั้งประสิทธิภาพในการใช้งานด้วย ทรัพยากรในที่นี้สามารถเป็นได้ทั้งฮาร์ดแวร์หรือซอต์ฟแวร์ก็ได้

**2.2 รูปแบบการใช้งานระบบ Computing Grid**

Computing Grid เป็นเทคโนโลยีที่นิยมใช้อย่างแพร่หลายมากที่สุดอย่างหนึ่ง โดยเป็นการใช้งานทรัพยากรด้านการประมวลผลภายในกลุ่มร่วมกัน ทำให้เกิดลักษณะการประมวลผลขนาดใหญ่ โดยทั่วไปจะเหมาะกับงานทางด้านวิทยาศาสตร์ที่เน้นงานด้านการประมวลผลเป็นหลัก

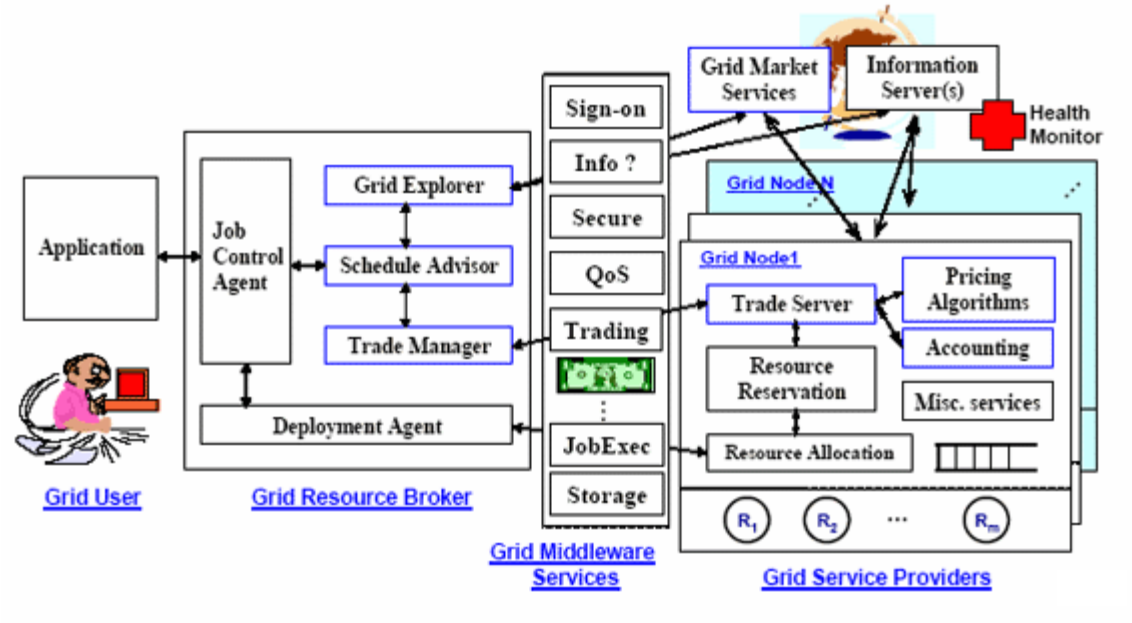



รูปที่ 4 Computing Grid

ที่มา : Thai grid, Grid Technology and Application [Online], accessed 15 April 2008. Available from http://rswg.thaigrid.or.th/book/export/html/75

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

(Vladimir Silva 2005 : 39) สถาปัตยกรรมเกี่ยวกับการประมวลผลของกริด จะใช้ Grid Middleware Service สำหรับให้บริการที่ช่วยให้ผู้ใช้สามารถเข้าใช้ทรัพยากรต่าง ๆ ผ่านทาง Grid Resource Broker ได้ โดยจะรวมถึงบริการเกี่ยวกับการจัดการโปรเซส การจัดสรรทรัพยากรร่วมกัน การเข้าถึงหน่วยความจำหรือหน่วยจัดเก็บข้อมูล การใช้งานข้อมูลต่าง ๆ การรักษาความปลอดภัย การยืนยันตัวบุคคล และคุณภาพของบริการ (QoS) เช่นเดียวกับเป็นทรัพยากรของตนเอง โดยความแตกต่างของ Grid Middleware จะขึ้นอยู่กับความต้องการขององค์กร




รูปที่ 5 Grid Architecture for Computational Economy

ที่มา : Thai grid, Grid Architecture for Computational Economy [Online], accessed 15 April 2008. Available from http://www.thaigrid.or.th/html/about/contents.php

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

**3. การเขียนโปรแกรมแบบขนานโดยการส่งผ่านข้อความด้วย MPI**

(Vladimir Silva 2005 : 431-458)

การเขียนโปรแกรมแบบขนานโดยทั่วไปจะมีการเขียนได้หลายวิธี แต่วิธีที่ได้รับความนิยมมากวิธีหนึ่งคือ การเขียนโดยการส่งผ่านข้อความ (Message Passing Interface: MPI) ซึ่งได้เตรียมส่วนติดต่อในการเขียนโปรแกรมส่งผ่านข้อความแบบเดียวกันในภาษาฟอร์แทรนและภาษาซี ซึ่งออกแบบมาให้สามารถทำงานได้บนเครื่องที่มีการทำงานแบบขนาน โดยมีพื้นฐานมาจากความต้องการที่จะมีการทำงานร่วมกันขององค์กรต่าง ๆ ในอเมริกาและยุโรป ซึ่งเครื่องคอมพิวเตอร์ประสิทธิภาพสูงส่วนมากจะมีการใช้งานร่วมกับ MPI

**3.1 ลักษณะทั่วไปของ MPI**

MPI เป็นมาตรฐานในการเขียนโปรแกรมที่มีการส่งผ่านข้อความ และสะดวกในการนำไปใช้ร่วมกับงานหลัก MPI เป็นแนวคิดพื้นฐานในการติดต่อสื่อสารระหว่างโปรเซสผ่านทางข้อความ การส่งผ่านข้อความเป็นตัวอย่างหนึ่งที่นิยมใช้บนเครื่องแบบขนาน โดยเฉพาะกับเครื่องที่มีการกระจายหน่วยความจำ ระบบการส่งผ่านข้อความสามารถทำงานได้อย่างมีประสิทธิภาพ

ระบบหน่วยความจำจะเป็นการใช้หน่วยความจำชั่วคราวหรือบัฟเฟอร์ (Buffer) และเก็บอยู่ภายในซึ่งจะเห็นเป็นวัตถุของ MPI เช่น กลุ่ม เครื่องสื่อสาร ประเภทข้อมูล เป็นต้น โดยทั่วไป MPI จะมีลักษณะ ดังนี้

3.1.1 การติดต่อสื่อสารที่มีประสิทธิภาพ และรองรับส่วนติดต่อในการเขียนโปรแกรมประยุกต์ ที่หลีกเลี่ยงการคัดลอกจากหน่วยความจำไปยังหน่วยความจำ และยอมให้มีการทำงานซ้อนกันของการประมวลผลและการติดต่อสื่อสาร

3.1.2 สามารถใช้ในสิ่งแวดล้อมที่แตกต่างกันได้

3.1.3 การติดต่อสื่อสารที่เชื่อถือได้ ซึ่งความล้มเหลวในระหว่างการติดต่อสื่อสารจะถูกจัดการอยู่ภายใต้การติดต่อสื่อสารของระบบย่อย

3.1.4 สามารถพัฒนาบนการทำงานหลายรูปแบบ โดยไม่เปลี่ยนนัยสำคัญ

3.1.5 มีส่วนติดต่อกับภาษาต่างๆ ที่มีความอิสระ นอกจากนี้ยังเตรียมความสามารถในการปรับประสิทธิภาพบนเครื่องคอมพิวเตอร์แบบขนาน

3.1.6 มีความสามารถมาตรฐานมีดังนี้

3.1.6.1 การติดต่อสื่อสารแบบเครื่องต่อเครื่อง (Peer-To-Peer)

3.1.6.2 การคำนวณเป็นหมู่คณะ

3.1.6.3 กลุ่มโปรเซส

3.1.6.4 การติดต่อสื่อสาร

3.1.6.5 โปรเซสโทโปโลยี

3.1.6.6 เข้ากันกับภาษาฟอร์แทรนและภาษาซี

3.1.6.7 การค้นหาและจัดการสิ่งแวดล้อม

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**3.2 โพรซิเดอร์และอาร์กิวเมนต์**

โพรซิเดอร์ของ MPI อาจเป็นได้ทั้งแบบบล็อค (Blocking) เพื่อให้เกิดการทำงานที่ผสานจังหวะกัน (Synchronous) หรือแบบไม่บล็อค (Non-Blocking) ที่การทำงานอาจไม่ผสานจังหวะกันก็ได้ (Asynchronous) โดยมีการคืนค่ากลับก่อนที่การทำงานจะเสร็จสิ้น

โพรซิเดอร์อาจเป็นแบบที่สามารถเรียกใช้ได้เฉพาะโปรเซสปัจจุบัน หรือเป็นแบบที่สามารถใช้งานร่วมกับโปรเซสอื่น หรือแบบที่ทุกโปรเซสในกลุ่มต้องการให้โพรซิเดอร์ในการทำงานร่วมกัน

อาร์กิวเมนต์สามารถใช้ได้ทั้งแบบ IN, OUT หรือ INOUT โดย IN เป็นการใช้งานโดยโพรซิเดอร์เท่านั้น ส่วน OUT เป็นการแก้ไข ปรับปรุง และ INOUT เป็นได้ทั้งการใช้งานและการแก้ไข ปรับปรุง

**3.3 ตัวแปรอะเรย์และค่าคงที่**

ตัวแปรอะเรย์จะใช้อาร์กิวเมนต์ในการแสดงถึงจำนวนสมาชิกในอะเรย์ และมีการระบุตำแหน่งของตัวชี้เพื่อเรียกใช้ข้อมูลในอะเรย์ตำแหน่งต่าง ๆ เช่นเดียวกับภาษาอื่น ๆ นอกจากนี้ตัวแปรที่เป็นค่าคงที่จะไม่สามารถเปลี่ยนแปลงค่าได้อีกเมื่อมีการทำงานอยู่ระหว่าง MPI_INIT และ MPI_FINALIZE

**3.4 ประเภทของตัวแปร**

ในโปรแกรมแบบ MPI จะมีการกำหนดประเภทของข้อมูลเบื้องต้นตามภาษาหลักที่เขียนร่วมกัน โดย MPI จะมีการกำหนดขนาดของข้อความในรูปของจำนวนอิลิเมนต์

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

ตารางที่ 1 ตารางเปรียบเทียบประเภทตัวแปรที่ใช้ใน MPI และภาษาซี

| MPI | C |
|---|---|
| MPI_CHAR | Signed char |
| MPI_SHORT | Signed short int |
| MPI_INT | Signed int |
| MPI_LONG | Signed long int |
| MPI_UNSIGNED_CHAR | Unsigned char |
| MPI_UNSIGNED_SHORT | Unsigned short int |
| MPI_UNSIGNED | Unsigned int |
| MPI_UNSIGNED_LONG | Unsigned long int |
| MPI_FLOAT | Float |
| MPI_DOUBLE | Double |
| MPI_LONG_DOUBLE | Long double |
| MPI_BYTE | |

**3.5 ฟังก์ชั่นในการส่งผ่านข้อความด้วย MPI**

การส่งผ่านข้อความด้วย MPI มีอยู่หลายฟังก์ชั่น ซึ่งสามารถใช้ในการแบ่งข้อความเพื่อส่งหรืออาจใช้ในการเลือกข้อความเพื่อรับก็ได้ การใช้งานฟังก์ชั่นต่าง ๆ จะจำกัดจำนวนฟิลด์และระบุตำแหน่งของอาร์กิวเมนต์ต่าง ๆ โดยส่วนหนึ่งของฟังก์ชั่นที่ใช้ในการส่งผ่านข้อความด้วย MPI มีดังนี้

3.5.1 ฟังก์ชั่น MPI_INIT และ MPI_FINALIZE

เป็นฟังก์ชั่นที่ใช้กำหนดขอบเขตการทำงานของโปรแกรม MPI โดยก่อนการใช้งาน MPI จะต้องเริ่มต้นที่การทำงานของ MPI_INIT ก่อนเสมอ และสิ้นสุดการทำงานด้วย MPI_FINALIZE ซึ่งมีรูปแบบ ดังนี้

- MPI_INIT (argc, argv[])
- MPI_FINALIZ (void)

3.5.2 ฟังก์ชั่น MPI_Comm_rank และ MPI_Comm_size

MPI_Comm_rank เป็นฟังก์ชั่นที่บอกถึงลำดับของโปรเซส หรือหมายเลขโปรเซส หรือจำนวน Rank ในการทำงานและ MPI_Comm_size เป็นฟังก์ชั่นที่บอกถึงจำนวนโปรเซสที่มีการทำงานทั้งหมด โดยทุก ๆ Rank จะรู้ว่ามีจำนวนโปรเซสในการทำงานทั้งหมดเท่าไหร่ ซึ่งมีรูปแบบ ดังนี้

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

- MPI_Comm_rank(MPI_Comm, rank)
- MPI_Comm_size(MPI_Comm, size)

3.5.3 ฟังก์ชั่น MPI_Bcast

เป็นฟังก์ชั่นที่ใช้ในการส่งข้อมูลในลักษณะการกระจายข้อมูลไปยัง Rank ต่างๆ ในกลุ่มการสื่อสารเดียวกัน ทำให้ทุก ๆ Rank ได้รับข้อมูลที่เหมือนกัน ซึ่งมีรูปแบบ ดังนี้

- MPI_Bcast(msg, count, datatype, source, tag, comm, status)
- msg เป็นตำแหน่งเริ่มต้นของข้อความที่ส่ง
- count จำนวนหรือขนาดของข้อความที่ส่ง
- datatype ชนิดของข้อมูลที่จะส่ง
- source กำหนดหมายเลข Rank ต้นทางที่ส่งข้อความ
- comm ตัวแปรที่ใช้ในการติดต่อสื่อสาร

3.5.4 ฟังก์ชั่น MPI_Scatter

เป็นฟังก์ชั่นที่ใช้ในการส่งข้อมูลในลักษณะของการกระจายข้อมูลไปยัง Rank ต่าง ๆ ในกลุ่มการสื่อสารเดียวกัน โดยในแต่ละ Rank จะได้รับข้อมูลที่ไม่เหมือนกันแต่มีขนาดของข้อมูลเท่ากัน หรือถ้าต้องการกระจายข้อมูลในลักษณะของ MPI_Scatter แต่ขนาดของข้อมูลไม่เท่ากันก็สามารถเลือกใช้ MPI_Scatterv แทนได้ ซึ่ง MPI_Scatter มีรูปแบบ ดังนี้

- MPI_ Scatter(msg_send, count_send, datatype_send, msg_recv, count_recv, datatype_recv, root, comm)

- msg_send เป็นตำแหน่งเริ่มต้นของข้อความที่ส่ง
- count_send จำนวนหรือขนาดของข้อความที่ส่ง
- datatype_send ชนิดของข้อมูลที่จะส่ง
- msg_recv เป็นตำแหน่งเริ่มต้นของข้อความที่รับ
- count_recv จำนวนหรือขนาดของข้อความที่รับ
- datatype_recv ชนิดของข้อมูลที่จะรับ
- root กำหนดหมายเลข Rank ที่กระจายข้อมูล
- comm ตัวแปรที่ใช้ในการติดต่อสื่อสาร

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

3.5.5 ฟังก์ชั่น MPI_Gather

เป็นฟังก์ชั่นที่ใช้ในการรับข้อมูลในลักษณะของการรวมข้อมูลที่ถูกกระจายไปยัง Rank ต่าง ๆ ในกลุ่มการสื่อสารเดียวกันกลับมา โดยในแต่ละ Rank จะส่งข้อมูลกลับมายัง Rank ที่กำหนด โดยมีขนาดของข้อมูลเท่ากัน หรือถ้าต้องการรวมข้อมูลในลักษณะของ MPI_Gather แต่ขนาดของข้อมูลไม่เท่ากันก็สามารถเลือกใช้ MPI_Gatherv แทนได้ซึ่ง MPI_Gather มีรูปแบบ ดังนี้

- MPI_ Gather(msg_send, count_send, datatype_send, msg_recv, count_recv, datatype_recv, root, comm)

- msg_send เป็นตำแหน่งเริ่มต้นของข้อความที่ส่ง
- count_send จำนวนหรือขนาดของข้อความที่ส่ง
- datatype_send ชนิดของข้อมูลที่จะส่ง
- msg_recv เป็นตำแหน่งเริ่มต้นของข้อความที่รับ
- count_recv จำนวนหรือขนาดของข้อความที่รับ
- datatype_recv ชนิดของข้อมูลที่จะรับ

- root กำหนดหมายเลข Rank ที่รวมข้อมูล
- comm ตัวแปรที่ใช้ในการติดต่อสื่อสาร

3.5.6 ฟังก์ชั่น MPI_Wtime

เป็นฟังก์ชั่นที่ใช้จับเวลาในการทำงาน โดยดึงเวลาจากระบบปฏิบัติการมาใช้งาน ซึ่งเป็นการเรียกใช้ข้อมูลจากสภาพแวดล้อม ซึ่งมีรูปแบบ ดังนี้

- MPI_Wtime(void)

นอกจากนี้ยังมีฟังก์ชั่นอื่น ๆ ให้เลือกใช้ให้เหมาะกับงานที่ทำ เช่น การใช้ MPI_Isend และ MPI_Irecv ที่เป็นการทำงานแบบไม่มีการบล็อคถ้าต้องการให้มีการหยุดรอเพื่อประสานจังหวะการทำงานสามารถใช้ฟังก์ชั่น MPI_Barrier ร่วมด้วยได้

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# บทที่ 4
# วิธีดำเนินการวิจัย

การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริดพัฒนาขึ้นเพื่อหาผลลัพธ์การทำงานสำหรับประกอบผลการวิจัย ซึ่งสามารถสรุปถึงขั้นตอนและวิธีดำเนินการวิจัย โดยมีรายละเอียดดังนี้

**1. การเตรียมตัวอย่างสำหรับการรู้จำ**

งานวิจัยนี้ใช้รูปภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ A – Z จำนวน 26 ตัวอักษร เป็นตัวอย่างสำหรับใช้ในการทดลองการเรียนรู้ รูปภาพทั้งหมดสามารถสร้างได้จากโปรแกรมประยุกต์ทางด้านกราฟิกทั่วไป หรือภาพตัวอักษรที่ได้จากการสแกนเอกสารตัวอย่าง โดยมีคุณสมบัติเบื้องต้นดังนี้

1.1 เป็นตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ A - Z

1.2 มีรูปแบบของตัวอักษรแต่ละตัวที่แตกต่างกัน เพื่อความหลากหลายของการเรียนรู้

1.3 รูปภาพตัวอักษรแต่ละรูป จะต้องมีสีของตัวอักษร และสีพื้นหลังที่แตกต่างกัน เพื่อให้ง่ายต่อการอ่านค่าสีสำหรับแปลงเป็นสีขาว-ดำ

1.4 ไฟล์ภาพที่ใช้เป็นไฟล์ภาพชนิด BMP ขนาดกว้าง 100 พิกเซล และสูง 100 พิกเซล

**2. เครื่องมือและอุปกรณ์**

2.1 เครื่องสแกนแบบ All in one Epson Stylus CX3700

2.2 เครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ในการวิจัย ซึ่งมีคุณสมบัติดังนี้

| | |
|---|---|
| - หน่วยประมวลผล | 2 GHz |
| - หน่วยความจำ DDR RAM | 2 GB |
| - หน่วยความจำ HDD | 160 GB |
| - การแสดงผล | Mobile Intel (R) 965 Graphic Media |

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**3. ซอร์ฟแวต์ที่ใช้ในการวิจัย**

3.1 ระบบปฏิบัติการ Rocks Cluster 4.2 (Hallasan)

3.2 Grid Rocks with architecture i386, x86_64

3.3 Putty A Free Telnet/SSH Client สำหรับเข้าใช้งาน Rocks Frontend Node ภายในกลุ่มของ SU Grid

3.4 WinSCP : Free SFTP, FTP and SCP client for Windows สำหรับรับหรือส่งไฟล์ต่าง ๆ ไปยัง Rocks Frontend Node

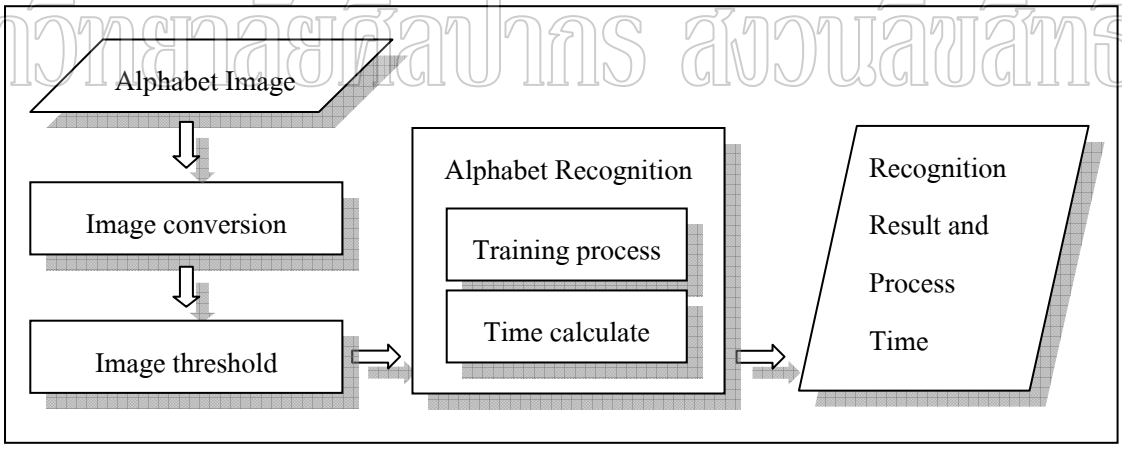
3.5 GCC สำหรับพัฒนาโปรแกรมประยุกต์ ด้วยภาษา C/C++

3.6 MPICH-2 หรือ LAM/MPI สำหรับพัฒนาโปรแกรม MPI

**4. ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ**

เป็นขั้นตอนการรู้จำด้วยเทคนิคเครือข่ายใยประสาทที่มีการประมวลผลการเรียนรู้ด้วยหน่วยประมวลผลเพียงตัวเดียว ซึ่งจะทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล มีรายละเอียดดังนี้

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

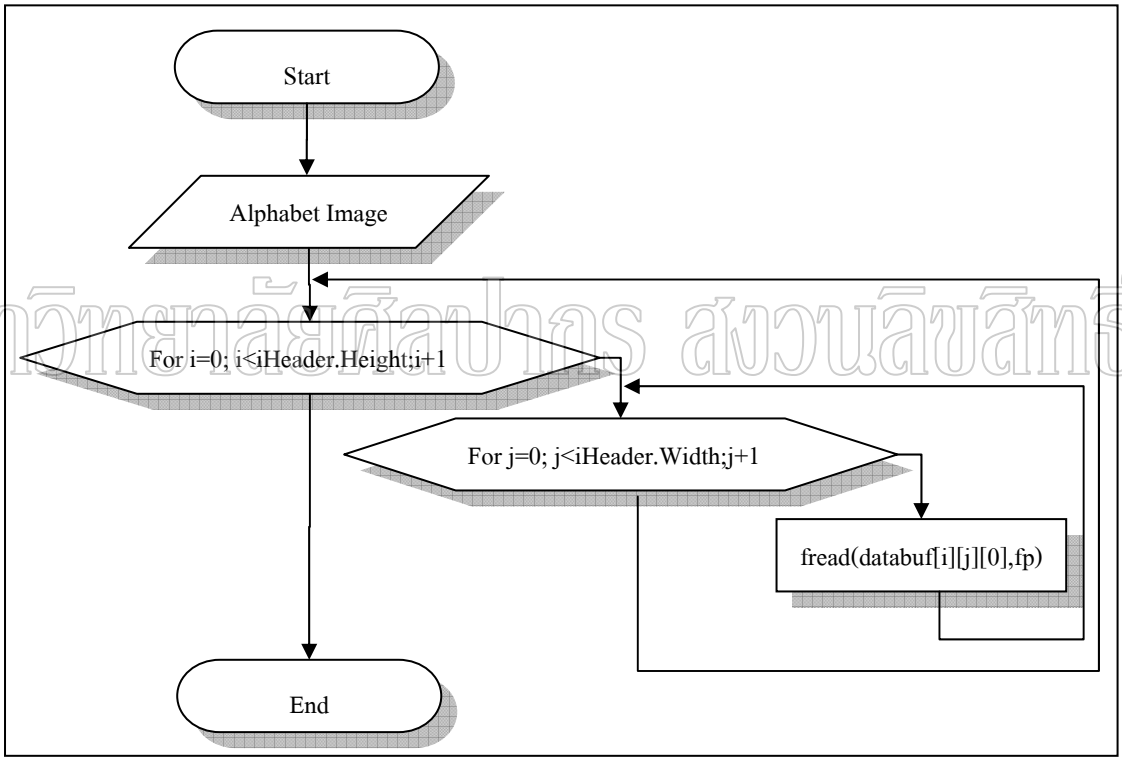



รูปที่ 6 ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ

จากรูปที่ 6 แสดงให้เห็นถึงขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ ซึ่งสามารถแบ่งออกเป็นขั้นตอนที่เกี่ยวข้อง 4 ขั้นตอน โดยมีรายละเอียดดังนี้

**4.1 ขั้นตอนการนำเข้ารูปภาพสำหรับใช้ในการเรียนรู้**

เป็นขั้นตอนการนำเข้าข้อมูลรูปภาพตัวอักษรที่ได้จากแหล่งต่าง ๆ เช่น การสร้างด้วยโปรแกรมประยุกต์ หรือการสแกนจากเอกสาร จะทำการเลือกไฟล์ภาพที่ต้องการซึ่งเป็นไฟล์ภาพแบบ BMP และควรมีความแตกต่างระหว่างสีของตัวอักษรและสีพื้นหลังอย่างชัดเจน (รายละเอียดเพิ่มเติมตามหัวข้อการเตรียมตัวอย่างสำหรับการรู้จำ) เมื่อได้ภาพที่ต้องการ จะทำการอ่านข้อมูลจากไฟล์ภาพ BMP ซึ่งประกอบด้วยส่วนหัวไฟล์ที่เก็บรายละเอียดต่าง ๆ ซึ่งไม่ได้ใช้ในการเรียนรู้ จากนั้นทำการอ่านค่าข้อมูลกลุ่มของพิกเซลที่เก็บค่าสีต่าง ๆ เอาไว้ออกมา ดังแสดงในรูปที่ 7




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 7 ขั้นตอนการอ่านค่าจากไฟล์รูปภาพตัวอักษรภาษาอังกฤษ

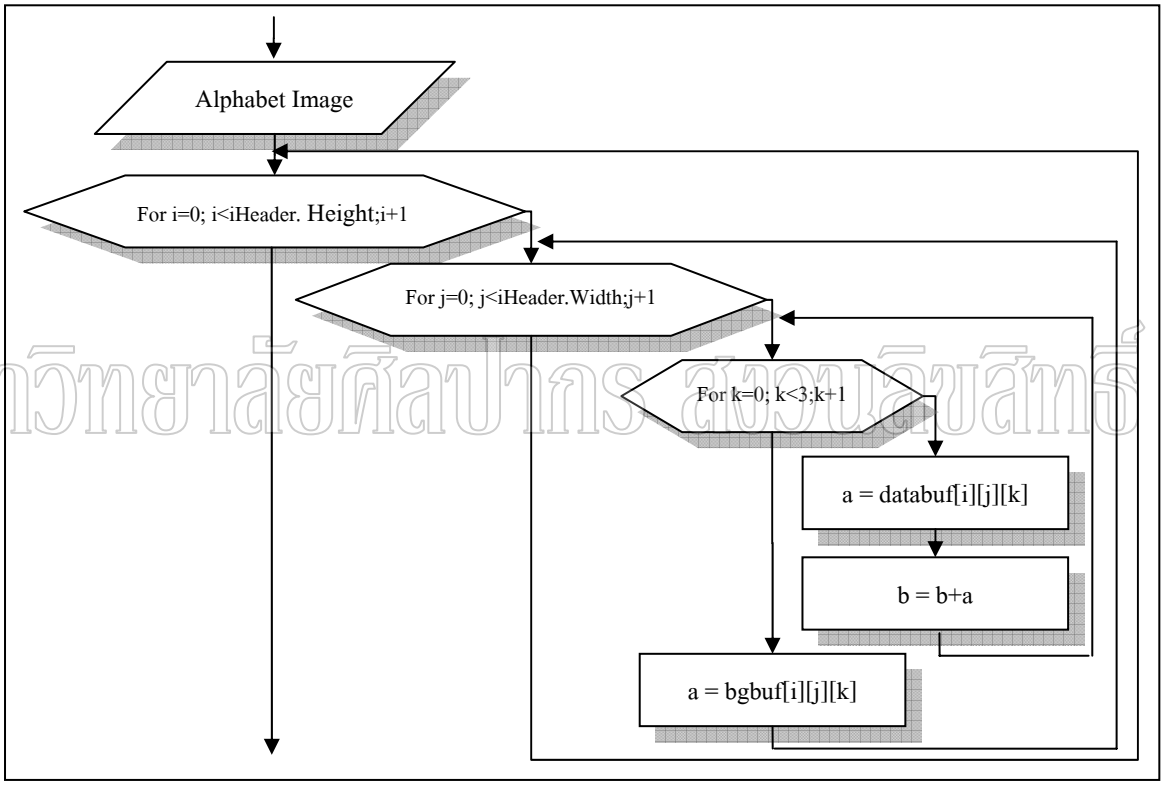
**4.2 ขั้นตอนการประมวลผลภาพเบื้องต้น**

เป็นขั้นตอนของการแปลงค่าข้อมูลจากไฟล์ภาพสีเป็นภาพขาว - ดำ ที่เป็นค่าตัวเลขอย่างง่ายเพื่อสะดวกต่อการนำไปเรียนรู้ในเครือข่ายใยประสาท สำหรับงานวิจัยนี้จะทำการ

อ่านค่าสีในแต่ละพิกเซลลงสู่อะเรย์เพื่อนำไปแปลงเป็นค่าตัวเลขอย่างง่ายที่ใช้ในการเรียนรู้ต่อไป ซึ่งมีรายละเอียดแต่ละขั้นตอนดังนี้

4.2.1 การอ่านค่าสีในแต่ละพิกเซลเพื่อแปลงเป็นข้อมูลนำเข้าที่ต้องการ

ในแต่ละพิกเซลประกอบไปด้วยค่าสีหลัก 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน ซึ่ง ค่าสีของแต่ละพิกเซลจะมีความเป็นไปได้ตั้งแต่ 0 ถึง 255 โดยค่าที่เข้าใกล้ 0 จะให้ค่าสีเข้ม และค่าที่เข้าใกล้ 255 จะให้ค่าสีที่เข้าใกล้สีขาว งานวิจัยนี้จึงได้ทำการรวมค่าสีทั้ง 3 สีเข้าด้วยกันโดยทำการวนรอบเพื่ออ่านค่าสีทั้ง 3 สีจากแต่ละพิกเซล ดังแสดงในรูปที่ 8

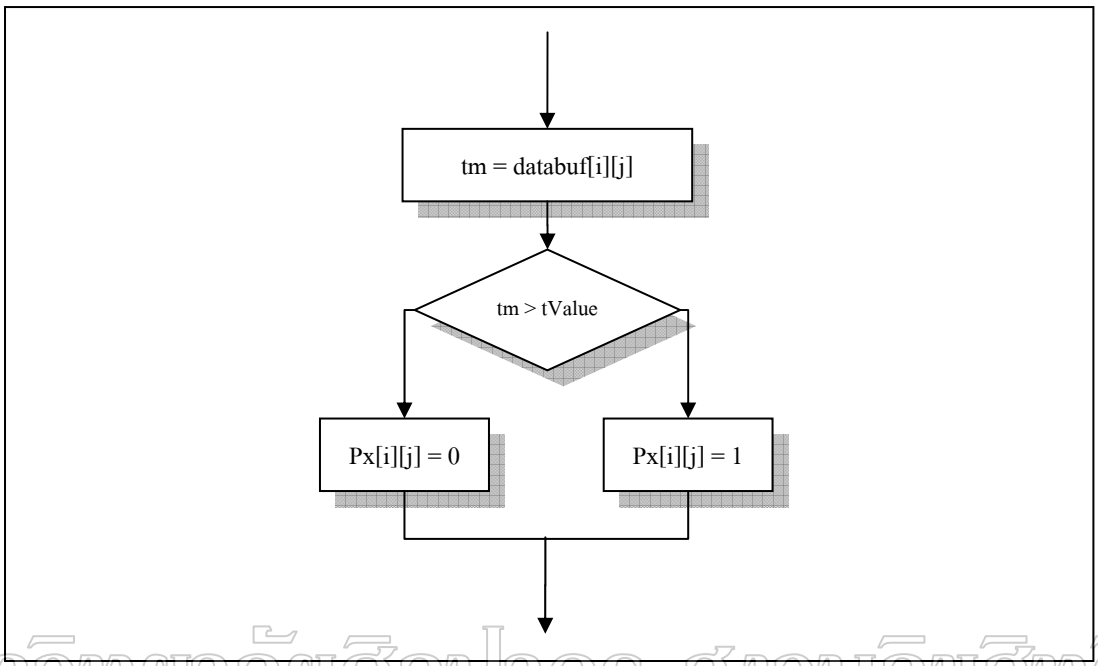



มหาวิทยาลัยศิลปากร ลิขสิทธิ์

รูปที่ 8 ขั้นตอนการอ่านค่าสีในแต่ละพิกเซลเพื่อทำการรวมค่าสี

จากรูปที่ 8 เมื่อทำการอ่านค่าสีในแต่ละพิกเซลเรียบร้อยแล้วจะทำการแปลงค่าสีที่ได้ให้อยู่ในลักษณะของรูปขาว-ดำหรือการเปลี่ยนให้แต่ละพิกเซลเป็นค่าตัวเลข 1 หรือ 0 นั่นเอง โดยการกำหนดค่าตัวแปร tValue สำหรับใช้ในการแยกความแตกต่างระหว่างสีตัวอักษรกับสีพื้นหลัง การเปรียบเทียบค่าสีในแต่ละพิกเซลกับค่าตัวแปร tValue ที่กำหนด จะได้ว่า ถ้าค่าสีในแต่ละพิกเซลมีค่ามากกว่าค่าที่กำหนดแสดงว่ามีความน่าจะเป็นสีขาวจะกำหนดค่าของพิกเซลนั้นเป็น 0

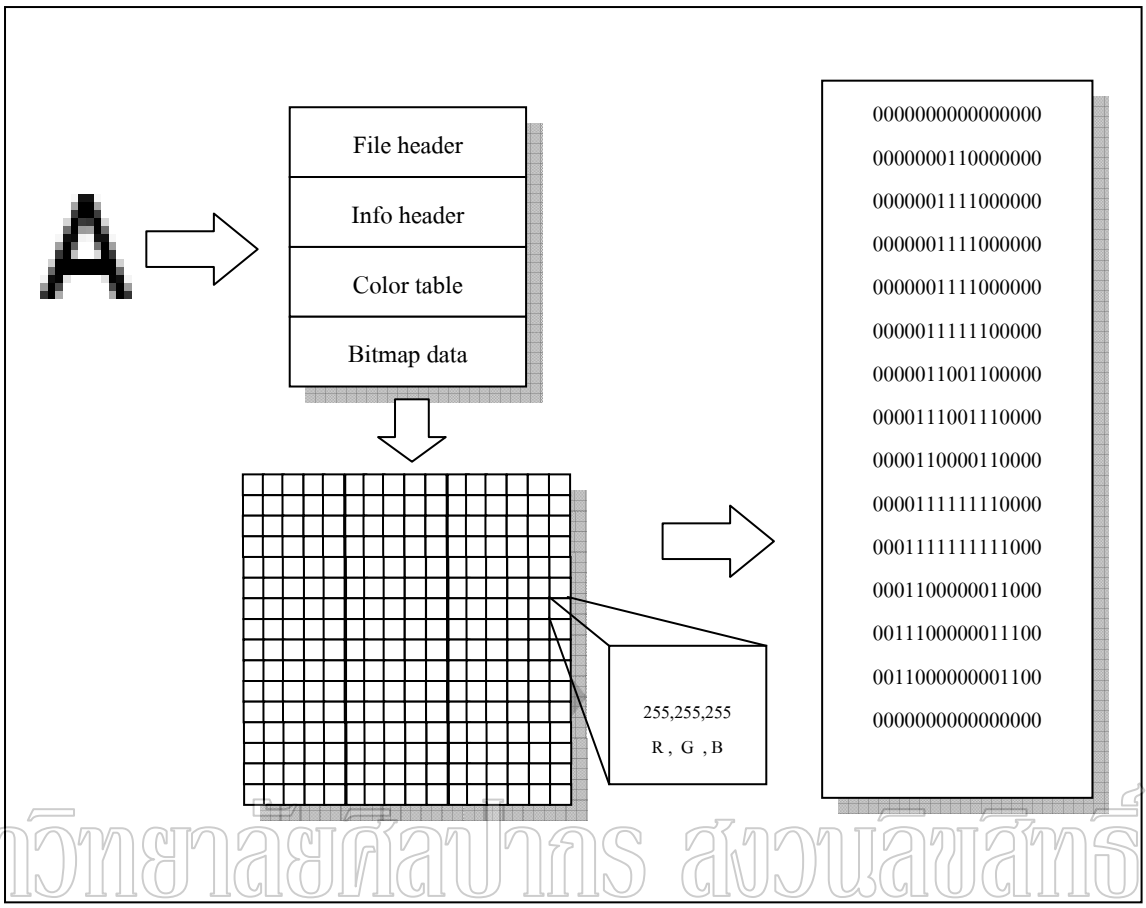
และถ้าค่าสีในแต่ละพิกเซลมีค่าน้อยกว่าค่าที่กำหนดแสดงว่ามีความน่าจะเป็นสีดำจะกำหนดค่าของพิกเซลนั้นเป็น 1 ดังแสดงในรูปที่ 9




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 9 ขั้นตอนการปรับค่าสีเพื่อเพิ่มความแตกต่างระหว่างตัวอักษรกับพื้นหลัง

เนื่องจากงานวิจัยนี้เลือกใช้ไฟล์รูปภาพประเภทบิทแมบหรือ BMP ซึ่งโดยทั่วไปจะมีการจัดเก็บข้อมูล 4 ส่วนคือ File header, Info header, Color table, Bitmap Data ซึ่งบรรจุข้อมูลรายละเอียดเกี่ยวกับไฟล์ เช่น ประเภทไฟล์ ขนาดไฟล์ ความกว้าง ความสูงของภาพ เป็นต้น เมื่อนำรายละเอียดเกี่ยวกับไฟล์ภาพมารวมกับขั้นตอนของการประมวลผลภาพเบื้องต้น สามารถแสดงได้ดังรูปที่ 10




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

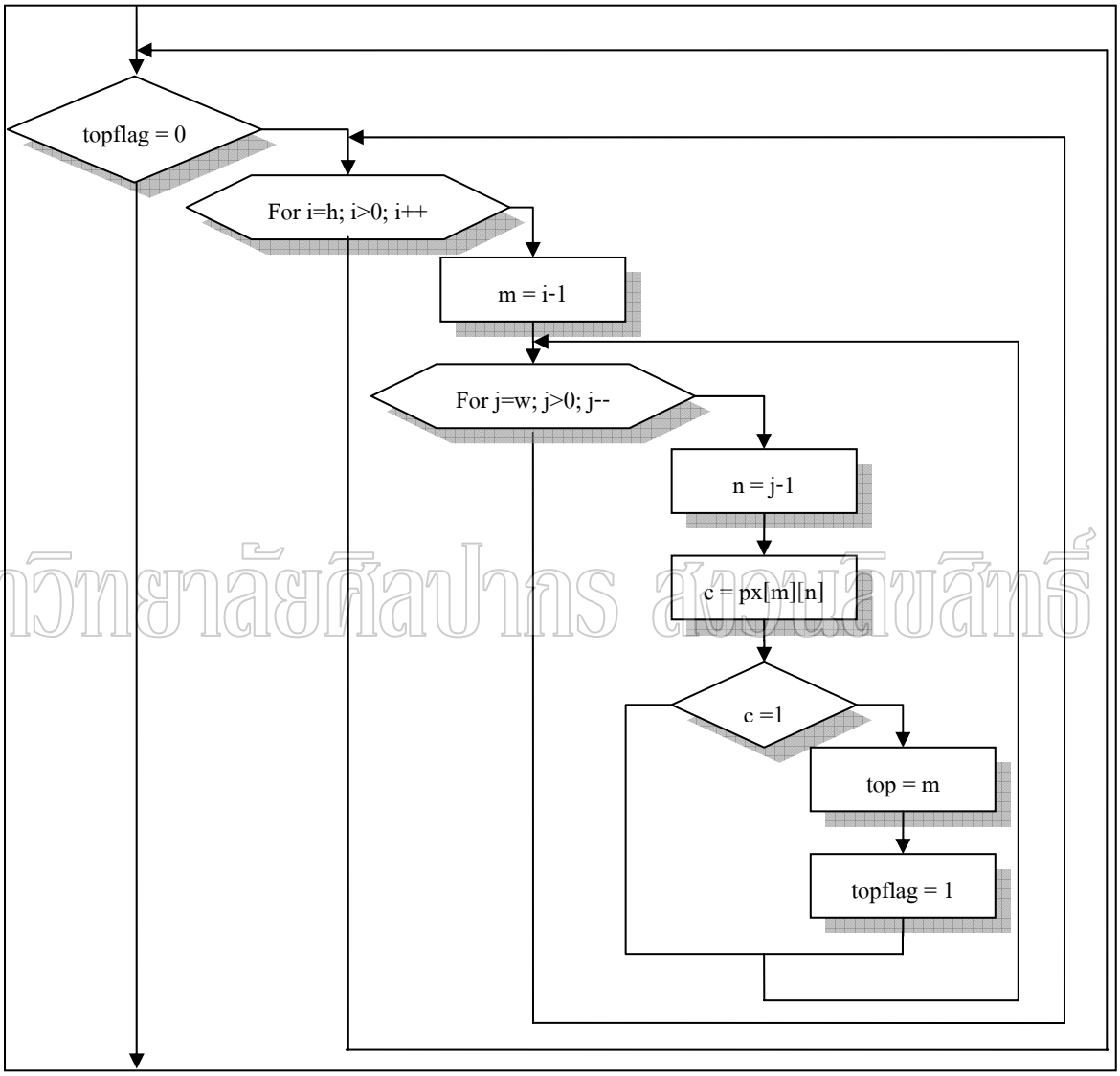
รูปที่ 10 ขั้นตอนการแทนค่าตัวอักษร

4.2.2 การเลือกเฉพาะบริเวณที่เก็บภาพตัวอักษร

จากขั้นตอนของการแทนค่าตัวอักษรจนได้ค่าสีขาว – ดำ ซึ่งเป็นค่าไบนารี่ (0,1) ดังรูปที่ 10 จะพบความแตกต่างระหว่างตัวอักษรและพื้นหลังอย่างชัดเจน ทำให้สามารถจำกัดขอบเขตของพื้นที่เฉพาะตัวอักษรด้วยแนวคิดการหาขอบเขตตัวอักษร โดยมีรายละเอียดขั้นตอนการหา ดังนี้

- การหาขอบเขตบน
- การหาขอบเขตล่าง
- การหาขอบเขตซ้าย
- การหาขอบเขตขวา
- ขั้นตอนการหาขอบเขตบนของตัวอักษร จะทำการตรวจสอบค่าไบนารี่ในแต่ละพิกเซลว่ามีค่าเท่ากับ 1 หรือไม่ โดยจะเรียงลำดับจากซ้ายไปขวาจากแถวด้านบนลงมาด้านล่าง

เมื่อพบค่าข้อมูลดังกล่าวจะทำการบันทึกค่าตัวแปร top เป็นค่าลำดับของแถวที่พบและบันทึกค่าตัวแปร topflag ซึ่งเป็นตัวแปรสถานะขอบเขตบนเป็น 1 เพื่อบอกว่าพบขอบเขตบน ดังแสดงในรูปที่ 11

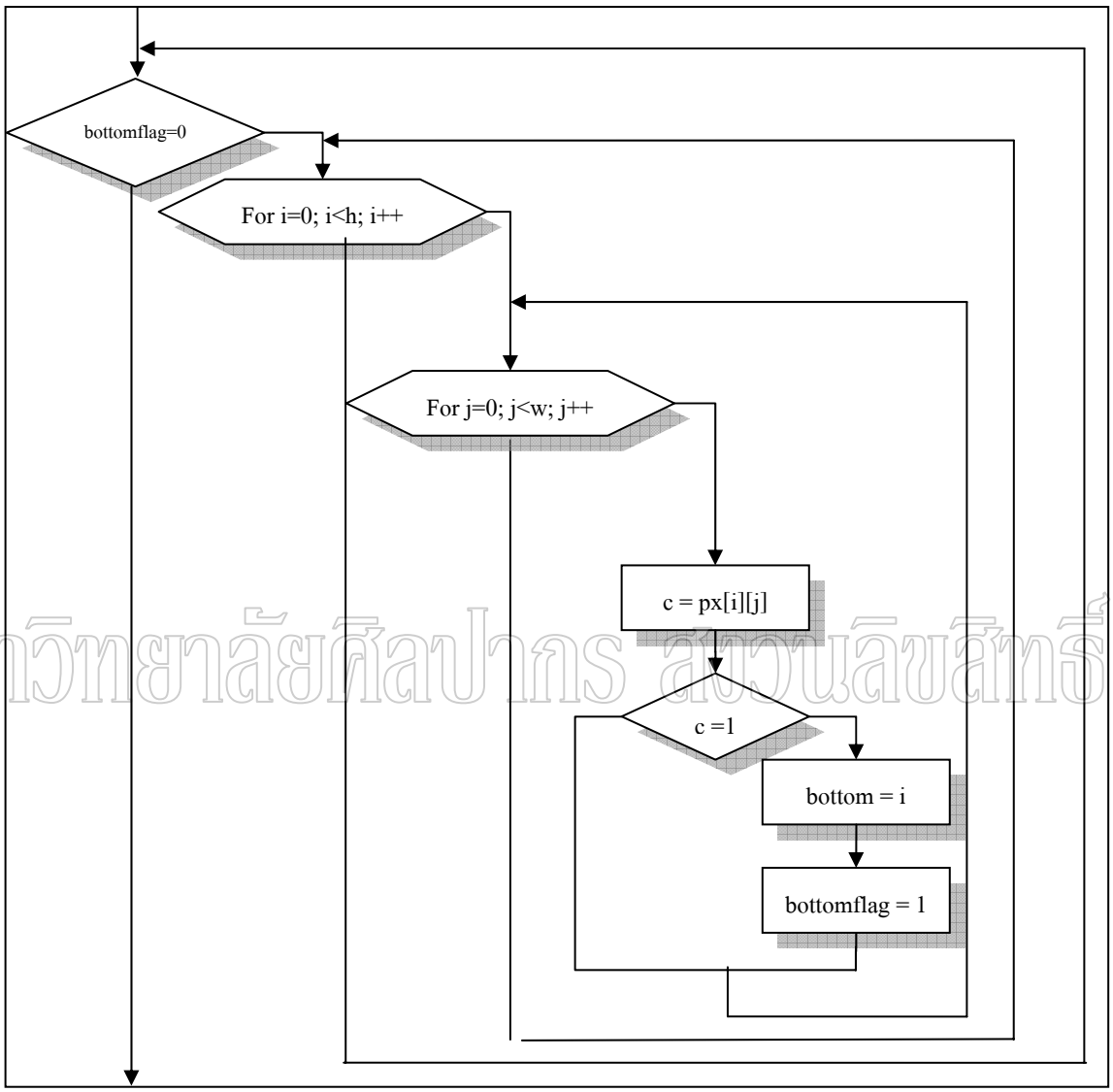



มหาวิทยาลัยศิลปากร ลิขสิทธิ์

รูปที่ 11 แสดงขั้นตอนการหาค่าขอบเขตบนของตัวอักษร

- ขั้นตอนการหาขอบเขตล่างของตัวอักษร จะทำการตรวจสอบค่าไบนารี่ในแต่ละพิกเซลว่ามีค่าเท่ากับ 1 หรือไม่ โดยจะเรียงลำดับจากซ้ายไปขวาจากแถวด้านล่างขึ้นมาด้านบน เมื่อพบค่าข้อมูลดังกล่าวจะทำการบันทึกค่าตัวแปร bottom เป็นค่าลำดับของแถวที่พบและ

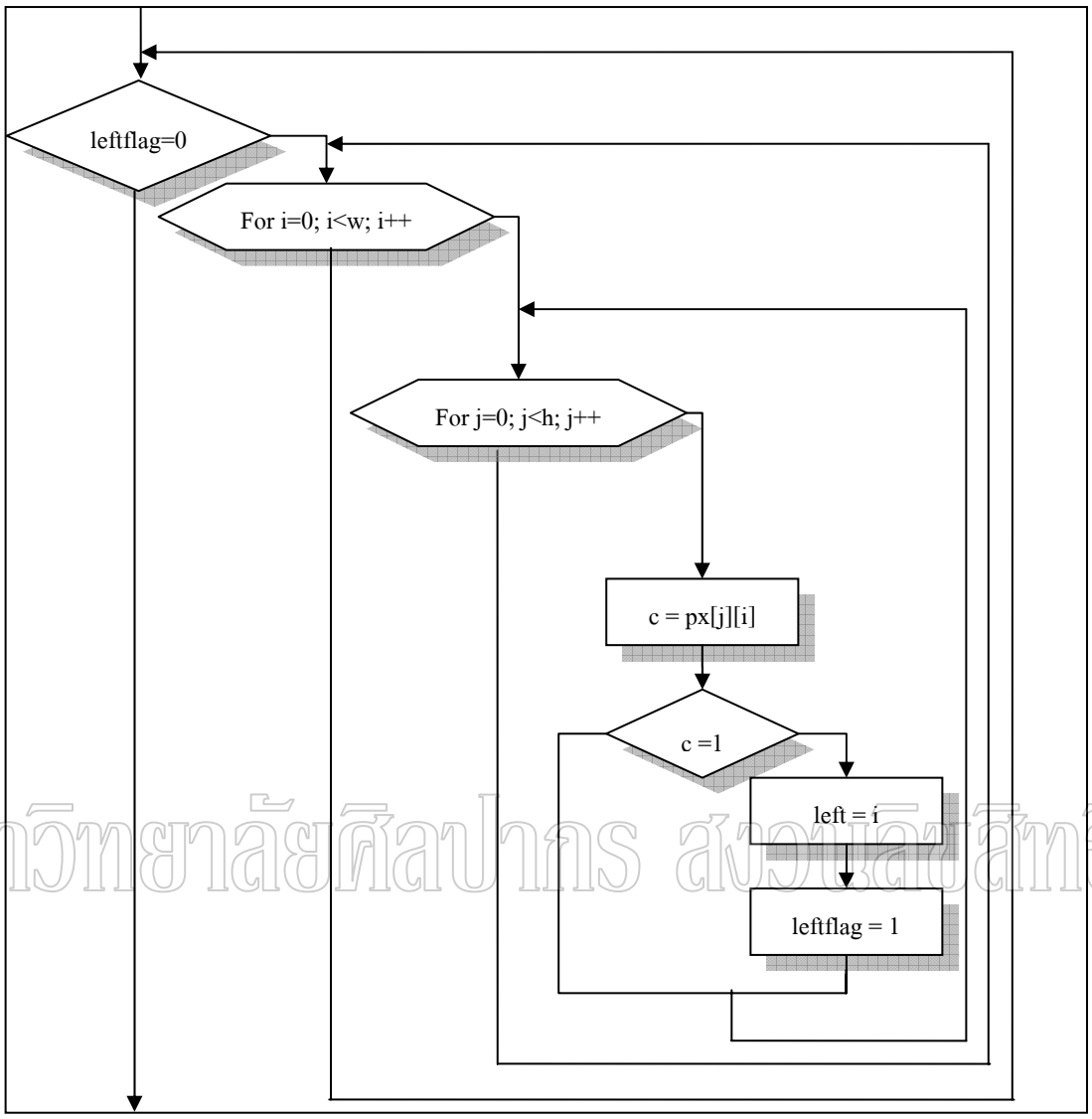
บันทึกค่าตัวแปร bottomflag ซึ่งเป็นตัวแปรสถานะขอบเขตล่างเป็น 1 เพื่อบอกว่าพบขอบเขตล่าง ดังแสดงในรูปที่ 12




มหาวิทยาลัยศิลปากร ลิขสิทธิ์

รูปที่ 12 แสดงขั้นตอนการหาค่าขอบเขตล่างของตัวอักษร

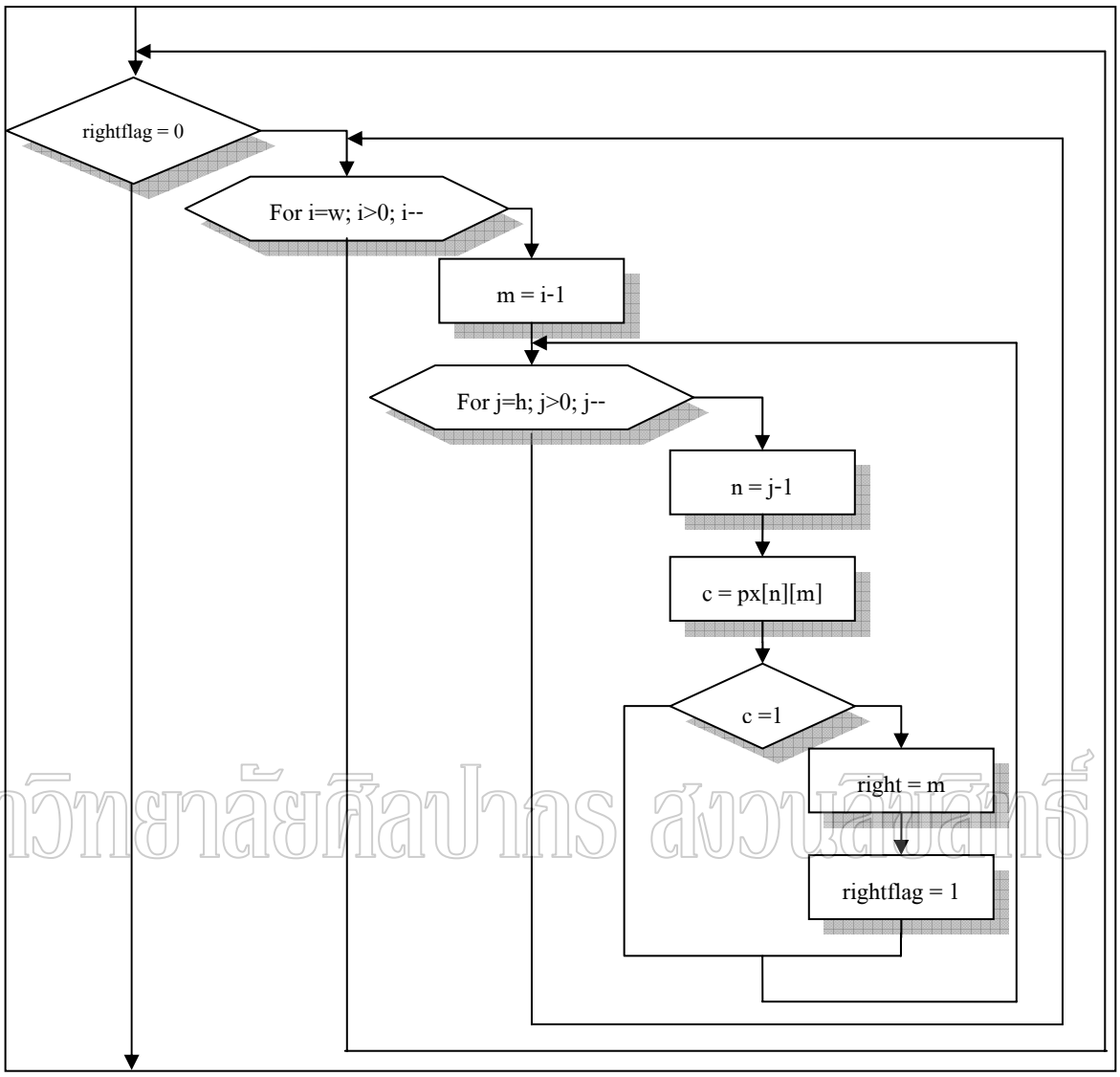
- ขั้นตอนการหาขอบเขตซ้ายของตัวอักษรจะทำการตรวจสอบค่าไบนารี่ในแต่ละพิกเซลว่ามีค่าเท่ากับ 1 หรือไม่ โดยจะเรียงลำดับจากซ้ายไปขวาจากแถวด้านล่างขึ้นมาด้านบน เมื่อพบค่าข้อมูลดังกล่าวจะทำการบันทึกค่าตัวแปร left เป็นค่าลำดับของแถวที่พบและบันทึกค่าตัวแปร leftflag ซึ่งเป็นตัวแปรสถานะขอบเขตซ้ายเป็น 1 เพื่อบอกว่าพบขอบเขตซ้าย ดังแสดงในรูปที่ 13




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 13 แสดงขั้นตอนการหาค่าขอบเขตซ้ายของตัวอักษร

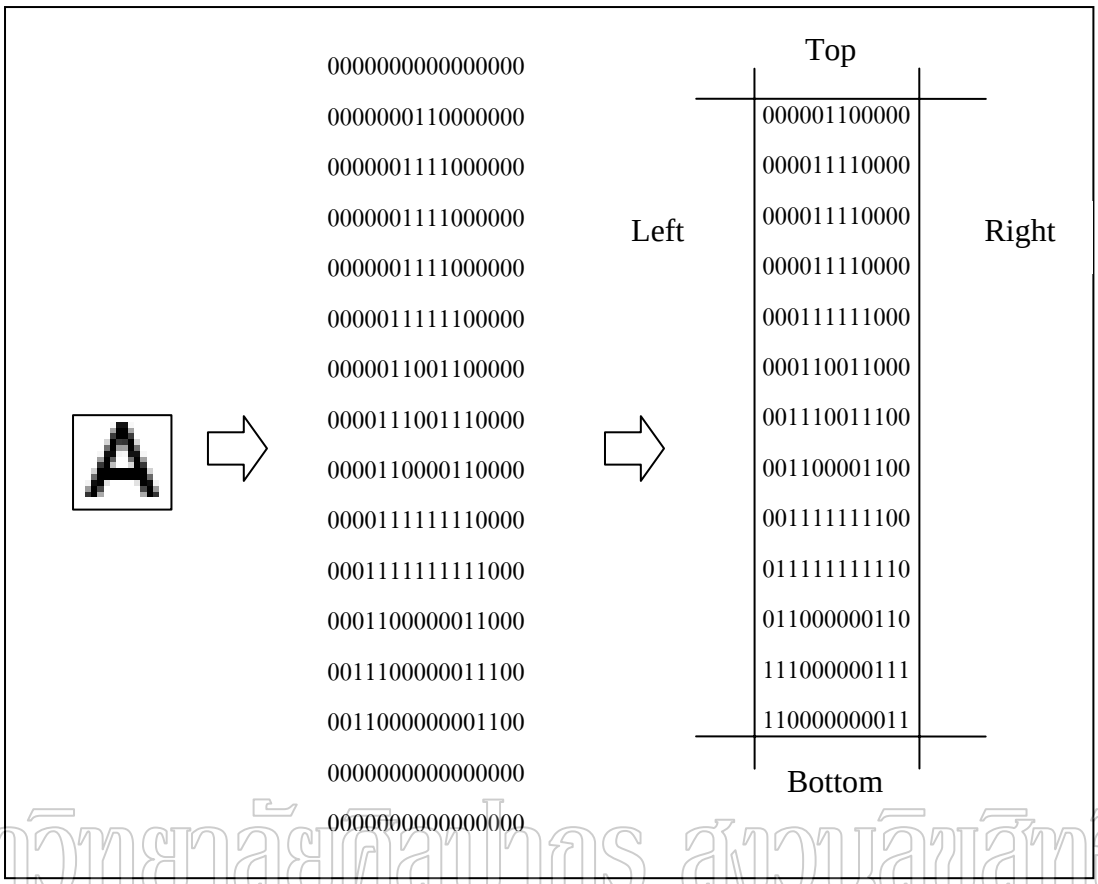
- ขั้นตอนการหาขอบเขตขวาของตัวอักษรจะทำการตรวจสอบค่าไบนารี่ในแต่ละพิกเซลว่ามีค่าเท่ากับ 1 หรือไม่ โดยจะเรียงลำดับจากขวาไปซ้ายจากแถวด้านบนลงมาด้านล่าง เมื่อพบค่าข้อมูลดังกล่าวจะทำการบันทึกค่าตัวแปร right เป็นค่าลำดับของแถวที่พบและบันทึกค่าตัวแปร rightflag ซึ่งเป็นตัวแปรสถานะขอบเขตขวาเป็น 1 เพื่อบอกว่าพบขอบเขตขวา ดังแสดงในรูปที่ 14




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 14 แสดงขั้นตอนการหาค่าขอบเขตขวาของตัวอักษร

เมื่อผ่านขั้นตอนการประมวลผลเบื้องต้นแล้วจะทำให้ได้ข้อมูลนำเข้าสำหรับใช้ในการเรียนรู้ ซึ่งอยู่ในรูปของตัวแปรชุดอะเรย์ 2 มิติ ดังแสดงในรูปที่ 15



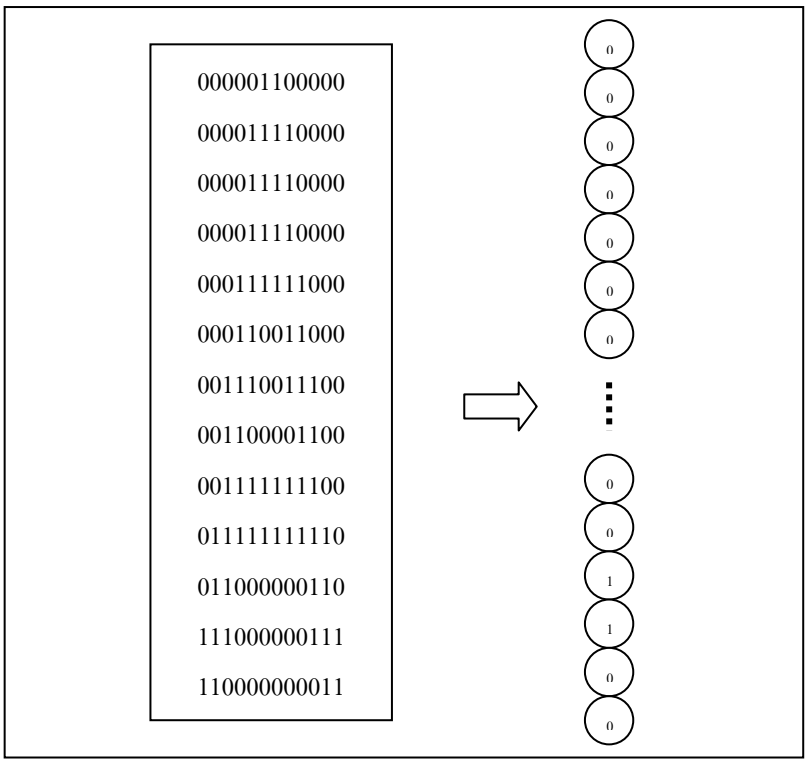

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 15 แสดงขั้นตอนการจำกัดขอบเขตของตัวอักษร

**4.3 ขั้นตอนการเรียนรู้ตัวอักษรภาษาอังกฤษ**

เริ่มจากการนำข้อมูลที่ได้จากการแปลงค่าข้อมูลสีแต่ละพิกเซลซึ่งเป็นค่าไบนารี่ (0,1) ที่ถูกเก็บไว้ในรูปแบบของตัวแปรอะเรย์ เพื่อใช้ในการคำนวนตามแนวคิดของเทคนิคเครือข่ายใยประสาทที่ทำงานแบบ Multilayer Perceptron หรือ MLP ซึ่งแบ่งการประมวลผลออกเป็นชั้นต่าง ๆ ประกอบด้วย

4.3.1 ชั้นข้อมูลเข้า (Input Layer) ได้จากการนำค่าไบนารี่ที่ได้มาใช้เป็นข้อมูลนำเข้าเบื้องต้น กำหนดให้มีจำนวนโหนดในชั้นนี้เท่ากับจำนวนของค่าไบนารี่ที่ได้ โดยสามารถคำนวณได้จากผลคูณระหว่างความกว้างกับความสูงของตัวอักษรที่ผ่านการประมวลผลเบื้องต้นแล้ว โดยตัวอักษรแต่ละตัวจะมีจำนวนโหนดในชั้นข้อมูลเข้าไม่เท่ากัน



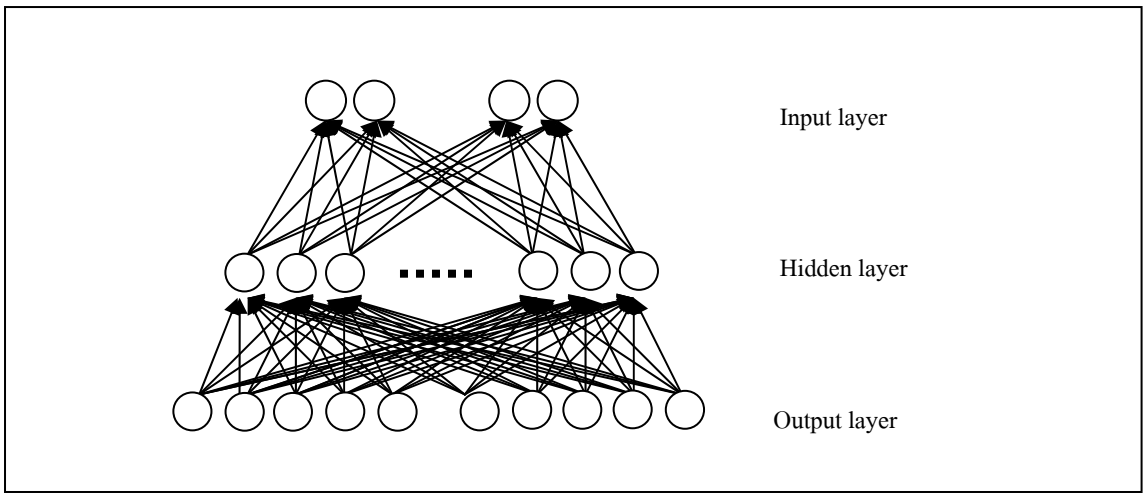

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 16 การอ่านค่าข้อมูลในชั้นนำเข้าข้อมูล

4.3.2 ชั้นซ่อน (Hidden Layer) เป็นชั้นที่นำค่าของโหนดต่างๆในชั้นข้อมูลเข้ามาใช้เป็นข้อมูลเริ่มต้นในการประมวลผล โดยกำหนดให้มีจำนวนโหนดในชั้นซ่อน 512 โหนด ซึ่งผลลัพธ์ที่ได้จากชั้นซ่อนจะใช้เป็นข้อมูลเริ่มต้นในการประมวลผลในชั้นผลลัพธ์ต่อไป

4.3.4 ชั้นผลลัพธ์ (Output Layer) เป็นชั้นสุดท้ายของการประมวลผลในแต่ละรอบของการเรียนรู้ โดยกำหนดให้มีจำนวน 8 โหนด

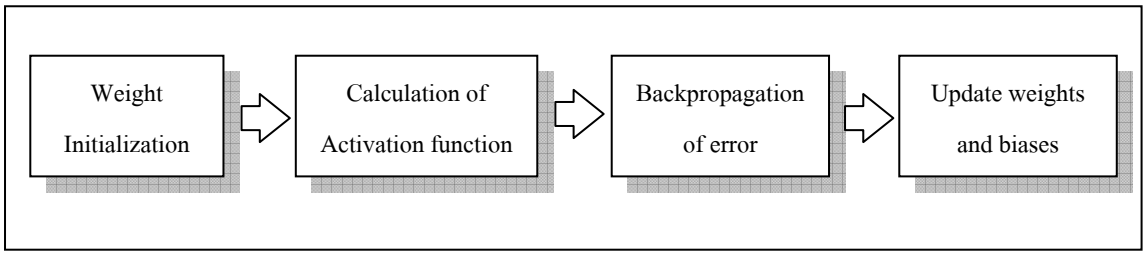
จากการทำงานของเทคนิคเครือข่ายใยประสาทดังกล่าวข้างต้น เป็นการทำงานแบบ Multilayer Feed Forward Network ซึ่งสามารถแสดงดังรูปที่ 17




รูปที่ 17 แสดงชั้นของ Multilayer Feed Forward Network

จะเห็นได้ว่าการทำงานแบบ Feed Forward Network เป็นการทำงานแบบที่มีการส่งค่าผลลัพธ์ไปด้านหน้า ทั้งนี้เพื่อให้เกิดการเรียนรู้จึงได้กำหนดให้มีการฝึกสอนด้วย Backpropagation ซึ่งมีแนวคิดในการส่งค่าผิดพลาดที่เกิดจากความแตกต่างระหว่างผลลัพธ์กับค่าเป้าหมายที่คาดหวังกลับไปยังระดับชั้นก่อนหน้าเพื่อทำการปรับค่าน้ำหนักให้เกิดความเหมาะสมดังแสดงได้ดังรูปที่ 18

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร




รูปที่ 18 แสดงขั้นตอนการฝึกสอนตามแนวคิด Backpropagation Algorithm

**4.4 ขั้นตอนการวัดเวลาการประมวลผลการเรียนรู้**

เนื่องจากงานวิจัยนี้เป็นการวัดประสิทธิภาพการประมวลผลด้วยการเปรียบเทียบเวลาที่ใช้ในการประมวลผล จึงได้มีการจับเวลาการประมวลผลในการเรียนรู้ โดยการเรียกใช้ฟังก์ชั่นเกี่ยวกับเวลาของ MPI คือ ฟังก์ชั่น MPI_Wtime เพื่ออ่านค่าเวลา ณ ขณะที่มีการเริ่มต้นการ

เรียนรู้ และอ่านค่าเวลาอีกครั้ง ณ ขณะที่การเรียนรู้เสร็จสิ้น จากนั้นนำค่าเวลาที่ได้มาลบกัน โดยมีหน่วยเป็นวินาที

เวลาเริ่มต้นการเรียนรู้ หาได้จาก Start time = MPI_Wtime();

เวลาสิ้นสุดการเรียนรู้ หาได้จาก End time = MPI_Wtime();

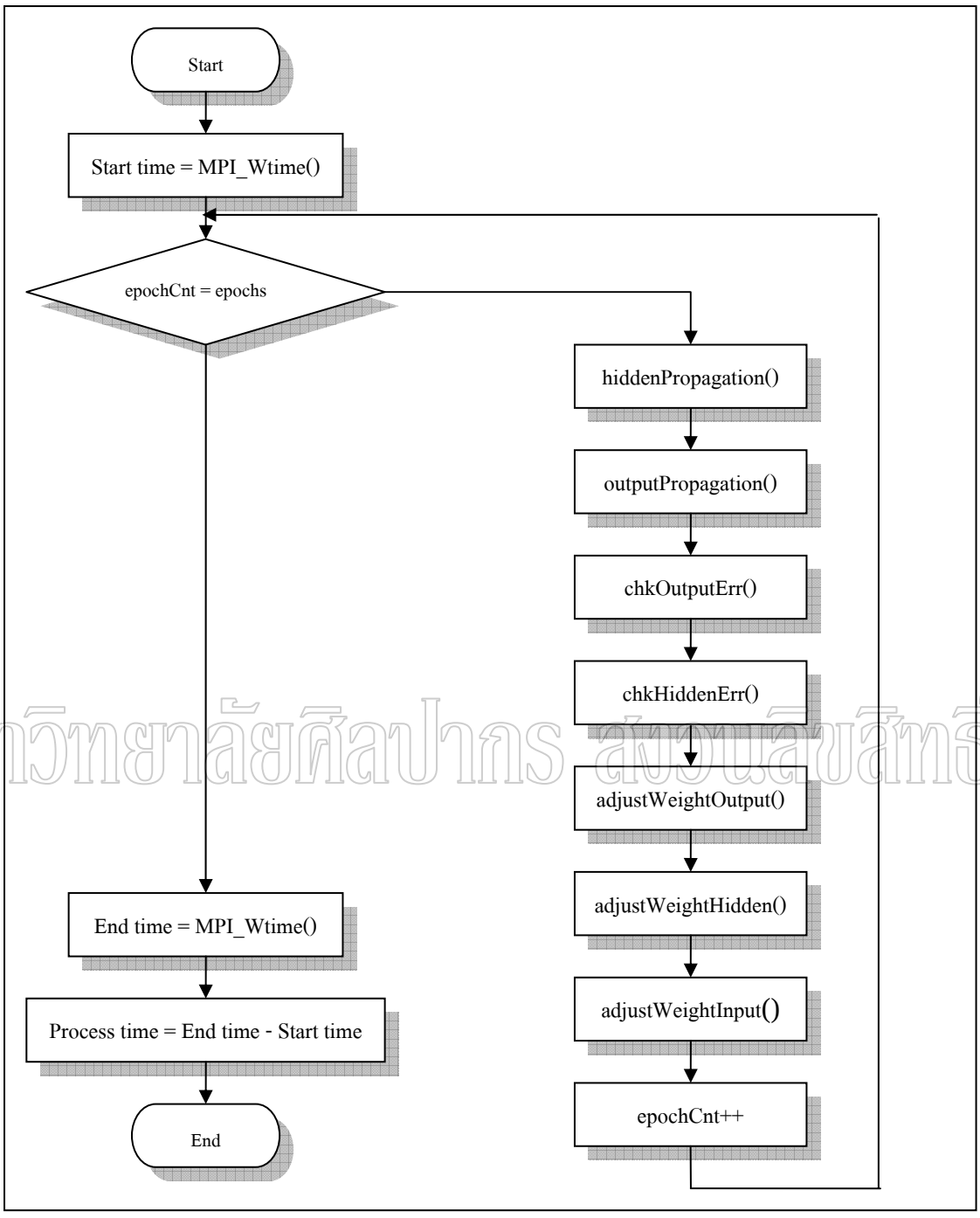
เวลาที่ใช้ในการประมวลผล หาได้จาก Process time = End time – Start time

**4.5 ขั้นตอนการทดสอบการรู้จำ**

เนื่องจากการเรียนรู้ด้วยเครือข่ายใยประสาท เป็นการปรับค่าน้ำหนักที่เชื่อมอยู่ในแต่ละ ชั้น ของการทำงาน ผลลัพธ์ที่ได้จึงเป็นชุดของค่าน้ำหนักที่ได้จากการเรียนรู้ ซึ่งจะเป็นค่าน้ำหนักเฉพาะ ของแต่ละรูปแบบ การทดสอบความถูกต้องของการรู้จำทำได้โดยการนำข้อมูลภาพตัวอักษรที่ต้องการทดสอบมาผ่านเครือข่ายใยประสาทที่ผ่านการเรียนรู้แล้ว ค่าข้อมูลที่นำเข้าจะผ่านการประมวลผลกับค่าน้ำหนักที่ได้จากการเรียนรู้ เมื่อถึงชั้นผลลัพธ์จะได้ค่าผลลัพธ์ซึ่งนำมาเปรียบเทียบกับค่าของรูปแบบที่ ได้มีการเก็บไว้ ถ้าตรงกับค่าผลลัพธ์ของรูปแบบตัวอักษรใด ก็จะสามารถบอกได้ว่าภาพที่ใช้ทดสอบเป็นตัวอักษรนั้น แต่ถ้าผลลัพธ์ที่ได้ไม่ตรงกับรูปแบบใด ๆ เลย ก็จะสามารถบอกได้ว่าไม่รู้จักภาพดังกล่าว โดยจะทำการเรียนรู้ครั้งละ 1 ภาพ และนำภาพตัวอักษรเดิมและตัวอักษร ๆ อื่น มาทำการทดสอบ จากนั้นทำการนับจำนวนความถูกต้องและความผิดพลาดที่ได้เพื่อหาเปอร์เซ็นต์ความถูกต้อง ซึ่งผลลัพธ์ที่ได้จะเป็นเปอร์เซ็นต์ความถูกต้องซึ่งแสดงในภาคผนวก ก

ทั้งนี้ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติสามารถแสดงได้ดังรูปที่ 19

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

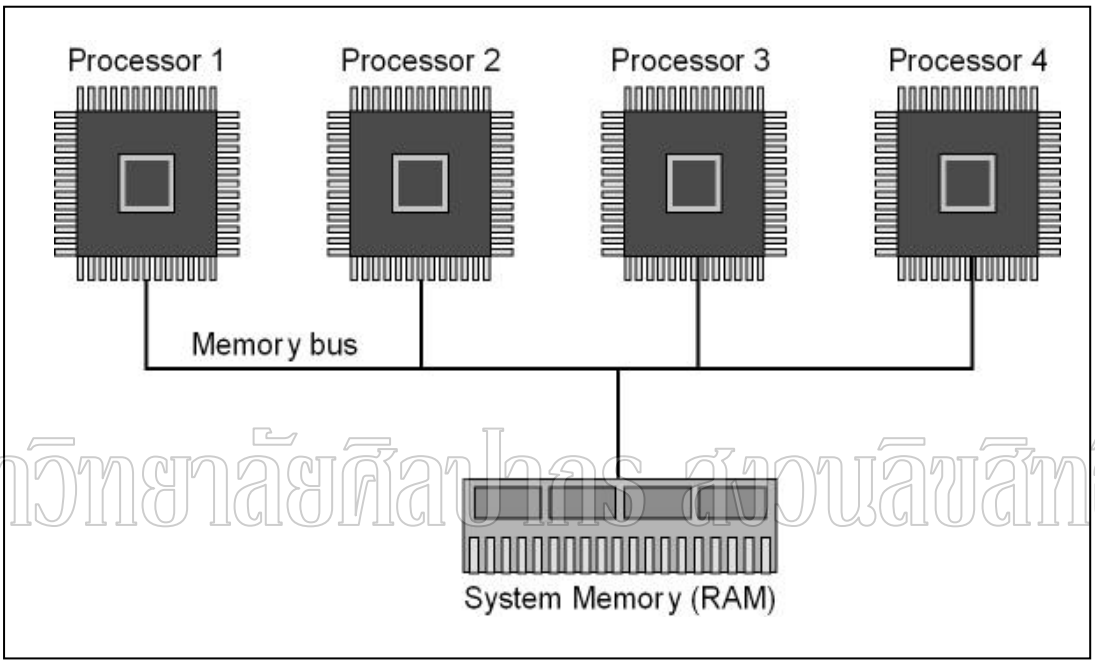



มหาวิทยาลัยศิลปากร ลิขสิทธิ์

รูปที่ 19 ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำแบบปกติ

**5. ขั้นตอนการทำงานของโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลกริด**

การประมวลผลแบบกริดเป็นการประมวลผลที่มีพื้นฐานมาจากระบบคลัสเตอร์ ซึ่งเป็นกลุ่มของเครื่องคอมพิวเตอร์หรือกลุ่มของหน่วยประมวลผลที่เชื่อมต่อกันผ่านเครือข่ายความเร็วสูง โดยแต่ละโหนดสมาชิกในคลัสเตอร์นิยมทำงานแบบหน่วยประมวลผลหลายตัว (Multiprocessor) ซึ่งสามารถใช้หน่วยความจำร่วมกันผ่านระบบบัสภายใน ดังรูปที่ 20




มหาวิทยาลัยศิลปากร ลิขสิทธิ์

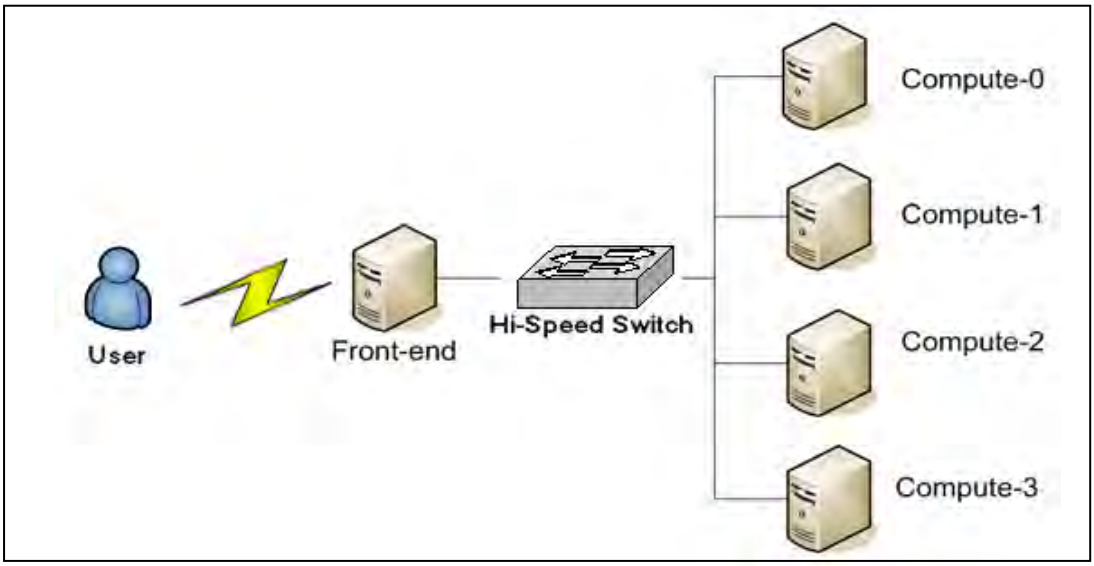
รูปที่ 20 โครงสร้างการทำงานของหน่วยประมวลผลหลายตัว (Multiprocessor)

สำหรับงานวิจัยนี้ได้เลือกใช้ Tubkaew Grid Cluster ในการทดลองซึ่งใช้ระบบปฏิบัติการ ROCKS OS และใช้ Grid Rolls ในการทำให้เครื่องคอมพิวเตอร์ในกลุ่มมีการทำงานแบบกริด โดยแบ่งเครื่องที่ใช้ในการทำงานออกเป็น 2 ประเภท คือ

- Front-End Node สำหรับติดต่อกับผู้ใช้และเป็นเครื่องที่กระจายข้อมูลไปยัง Compute Node

- Compute Node สำหรับประมวลผลตามข้อมูลที่ได้รับมาจากเครื่อง Front-End Node

ทั้งนี้ ลักษณะการเชื่อมต่อและการทำงานร่วมกันของ Front-End Node และ Compute Node แสดงได้ดังรูปที่ 21 และรายละเอียดทรัพยากรของ Tubkaew Grid Cluster แสดงได้ดังตารางที่ 2




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 21 โครงสร้างการเชื่อมต่อของ Front-End Node และ Compute Node

ตารางที่ 2 รายละเอียดทรัพยากรที่ใช้ในการประมวลผลของ Tubkaew Grid Cluster

| **Tubkaew Grid Cluster** | **Processor** | | | | **Memory** | **OS** | |
|---|---|---|---|---|---|---|---|
| | **Model** | **Clock (MHz)** | **Number** | **Detail** | **Size (MB)** | **Name** | **Release** |
| **compute-0-0** | X86_64 | 2800 | 4 | Intel(R) Xeon(TM) CPU 2.80GHz | 3946 | Rocks | 4.2 (Hallasan) (2.6.9-34.0.2. ELsmp) |
| **compute-0-1** | X86_64 | 2800 | 4 | | 3946 | Rocks | |
| **compute-0-2** | X86_64 | 2800 | 4 | | 3946 | Rocks | |
| **compute-0-3** | X86_64 | 2800 | 4 | | 3946 | Rocks | |
| **sugrid** | X86_64 | 2800 | 4 | | 3946 | Rocks | |

จากตารางที่ 2 เครื่อง sugrid เป็นเครื่อง Front-End Node ที่ทำหน้าที่กระจายข้อมูลไปยัง Compute Node ต่าง ๆ สำหรับงานวิจัยนี้ได้ทำการทดลองแบ่งจำนวนโปรเซสในการทำงานเป็น 4 รูปแบบ คือ 1, 2, 4 และ 8 ตามลำดับ เพื่อส่งไปประมวลผลยัง Compute Node ต่าง ๆ และวัดเวลาเพื่อเปรียบเทียบประสิทธิภาพด้านความเร็ว ทั้งนี้ในการประมวลผลแบบกริด

เมื่อเตรียมทรัพยากรในการประมวลผลต่าง ๆ พร้อมแล้ว ขั้นตอนในการทำงานของโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด จะมีดังนี้

**5.1 ขั้นตอนการนำเข้ารูปภาพสำหรับใช้ในการเรียนรู้**

เป็นขั้นตอนการนำเข้าข้อมูลรูปภาพตัวอักษรที่ได้จากแหล่งต่าง ๆ เช่น การสร้างด้วยโปรแกรมประยุกต์ และการสแกนโดยการเลือกไฟล์ภาพที่ต้องการซึ่งเป็นไฟล์ภาพแบบ BMP และควรมีความแตกต่างระหว่างสีของตัวอักษรและสีพื้นหลังอย่างชัดเจน (รายละเอียดเพิ่มเติมตามหัวข้อการเตรียมตัวอย่างสำหรับการรู้จำ) เมื่อได้ภาพที่ต้องการ จะทำการอ่านข้อมูลจากไฟล์ภาพ BMP ซึ่งประกอบด้วยส่วนหัวไฟล์ที่เก็บรายละเอียดต่าง ๆ จากนั้นทำการอ่านค่าข้อมูลกลุ่มของพิกเซลที่เก็บค่าสีต่าง ๆ เอาไว้ออกมาเพื่อใช้ในขั้นตอนต่อไป

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

**5.2 ขั้นตอนการประมวลผลภาพเบื้องต้น**

เป็นขั้นตอนของการแปลงค่าข้อมูลจากไฟล์ภาพให้อยู่ในลักษณะของภาพขาว-ดำ โดยการแทนค่าด้วย 1 หรือ 0 เพื่อสะดวกต่อการนำไปเรียนรู้ในเครือข่ายใยประสาท โดยมีขั้นตอนดังนี้

5.2.1 การอ่านค่าสีในแต่ละพิกเซลเพื่อแปลงเป็นข้อมูลนำเข้าที่ต้องการ

การอ่านค่าข้อมูลค่าสีของแต่ละพิกเซลจากไฟล์ภาพ โดยการวนรอบเพื่ออ่านค่าสีทีละสีแล้วหาค่าผลรวมของค่าสีทั้ง 3 สีในแต่ละพิกเซล จากนั้นทำการแปลงค่าสีที่ได้ให้เป็นค่าของภาพขาว-ดำ (0,1) โดยนำค่าที่ได้มาเปรียบเทียบกับค่าที่กำหนด เพื่อแยกความแตกต่างระหว่างสีตัวอักษรกับสีพื้นหลัง ถ้าค่าสีในแต่ละพิกเซลมีค่ามากกว่าค่าที่กำหนดแสดงว่ามีความน่าจะเป็นสีขาวจะกำหนดค่าของพิกเซลนั้นเป็น 0 และถ้าค่าสีในแต่ละพิกเซลมีค่าน้อยกว่าค่าที่กำหนดแสดงว่ามีความน่าจะเป็นสีดำจะกำหนดค่าของพิกเซลนั้นเป็น 1

5.2.2 การเลือกเฉพาะบริเวณที่เก็บภาพตัวอักษร

จากขั้นตอนของการแทนค่าตัวอักษรจนได้ค่าไบนารี่ (0,1) จะพบความแตกต่างระหว่างตัวอักษรและพื้นหลังอย่างชัดเจน ทำให้สามารถจำกัดขอบเขตของพื้นที่เฉพาะตัวอักษรด้วยแนวคิดการหาขอบเขตตัวอักษร โดยมีรายละเอียดขั้นตอนการหา ดังนี้

- การหาขอบเขตบน
- การหาขอบเขตล่าง
- การหาขอบเขตซ้าย
- การหาขอบเขตขวา

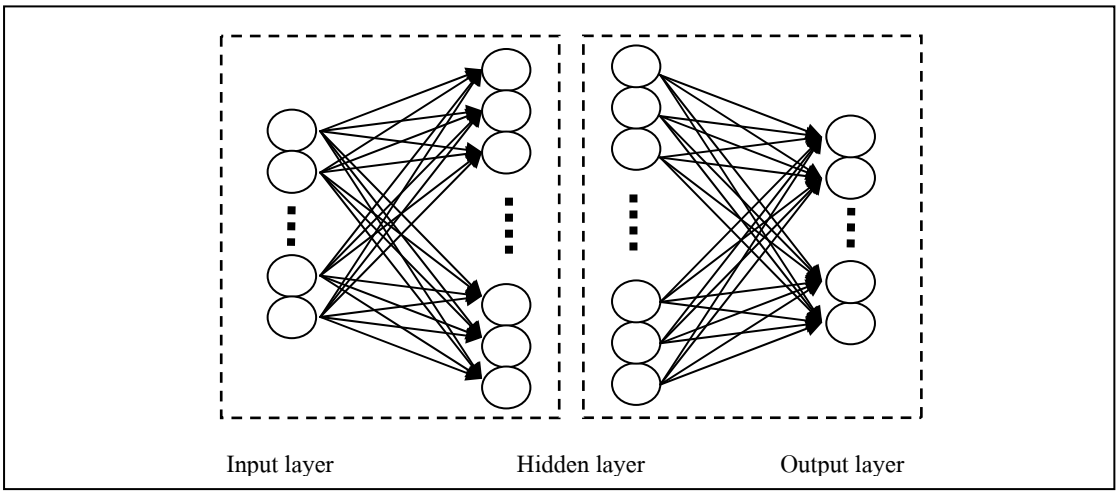
ทั้งนี้ขั้นตอนการนำเข้ารูปภาพสำหรับใช้ในการเรียนรู้และขั้นตอนการประมวลผลภาพเบื้องต้นจะเป็นการประมวลผลด้วยโปรเซสเดียวจึงกำหนดให้มีการประมวลผลที่เครื่อง Front-End Node ที่มี Process ID หรือ Rank หมายเลข 0 เท่านั้น

**5.3 ขั้นตอนการเรียนรู้ตัวอักษรภาษาอังกฤษ**

ขั้นตอนการเรียนรู้เริ่มจากการนำข้อมูลที่ได้จากขั้นตอนการประมวลผลภาพเบื้องต้น ซึ่งค่าข้อมูลเข้าจะเป็นค่าไบนารี่ (0,1) ที่ถูกเก็บในรูปแบบของตัวแปรอะเรย์ มาใช้ในการคำนวน โดยใช้เทคนิคเครือข่ายใยประสาทที่มีการทำงานแบบ Multilayer Perceptron และมีการเรียนรู้ด้วย Backpropagation การทำงานจะแบ่งออกเป็นระดับชั้นต่าง ๆ ซึ่งในแต่ละระดับชั้นจะมีโหนดย่อย ๆ ที่มีการประมวลผลเพื่อหาค่าผลลัพธ์ของแต่ละโหนด ซึ่งจะถูกนำไปใช้เป็นข้อมูลเริ่มต้นในระดับชั้นถัดไป

ทั้งนี้ ผู้วิจัยได้ทำการแบ่งส่วนการทำงานของการรู้จำที่ใช้การประมวลผลแบบกระจายออกเป็น 2 ส่วน คือ ส่วนของการประมวลผลระหว่างชั้นข้อมูลเข้ากับชั้นซ่อน และส่วนของการประมวลผลระหว่างชั้นซ่อนกับชั้นผลลัพธ์ ดังรูปที่ 22

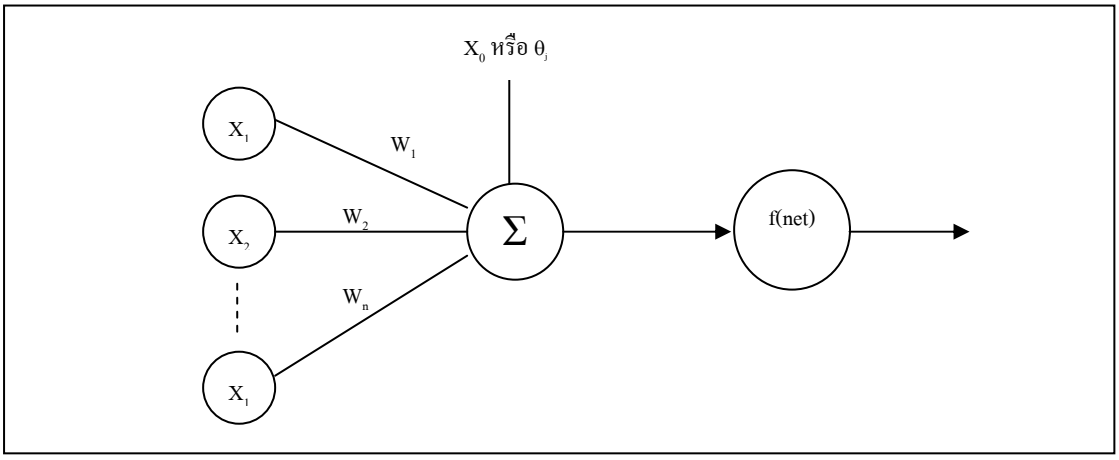
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี




รูปที่ 22 แสดงการแบ่งส่วนการทำงานของการรู้จำที่ใช้การประมวลผลแบบกระจาย

จากรูปที่ 22 ค่าข้อมูลของแต่ละโหนดในชั้นซ่อนได้จากการคำนวณเพื่อหาค่าผลรวมของการคูณระหว่างข้อมูลของโหนดต่าง ๆ ในชั้นข้อมูลเข้ากับค่าน้ำหนักของแต่ละโหนด เพื่อนำผลลัพธ์ที่ได้ไปคำนวณหาค่าฟังก์ชั่นกระตุ้นเพื่อใช้ในชั้นถัดไป และค่าข้อมูลของแต่ละโหนดในชั้นผลลัพธ์จะได้จากการคำนวณเพื่อหาค่าผลรวมของการคูณระหว่างข้อมูลของโหนดต่าง ๆ ในชั้นซ่อนกับค่าน้ำหนักของแต่ละโหนด ซึ่งการประมวลผลในแต่ละโหนดสามารถแสดงได้ ดังรูปที่ 23

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




รูปที่ 23 แสดงลักษณะการประมวลผลในแต่ละโหนดของเครือข่ายใยประสาท

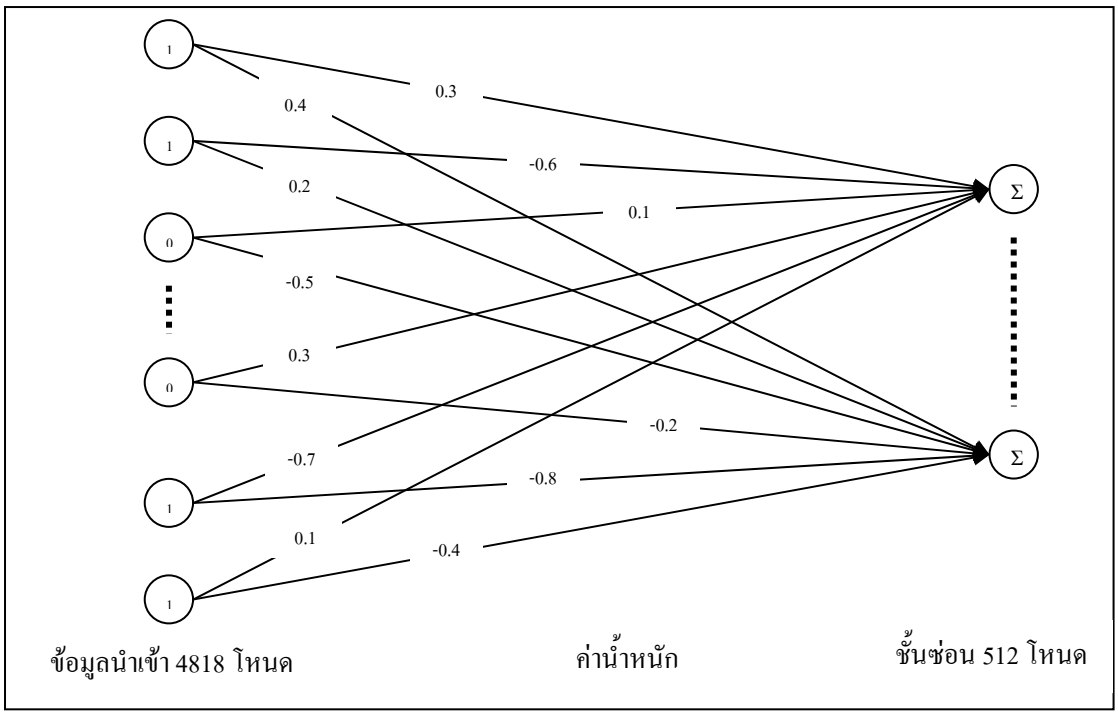
จากรูปที่ 23 กำหนดให้ $x_i$ แทนด้วยค่าข้อมูลของโหนดต่าง ๆ ในชั้นข้อมูลเข้าหรือชั้นซ่อน $w_i$ แทนด้วยค่าน้ำหนักที่อยู่ระหว่างแต่ละโหนดในชั้นข้อมูลเข้ากับชั้นซ่อน ทำการหาค่าผลรวมของการคำนวนระหว่าง $x_i$ และ $w_i$ เพื่อนำไปหาค่า f(net) หรือค่าผลลัพธ์ที่จะถูกใช้เป็นข้อมูลเข้าในชั้นถัดไป จากลักษณะดังกล่าว ทำให้สามารถอธิบายการเรียนรู้ในระดับชั้นต่าง ๆ ได้ดังนี้

5.3.1 ชั้นข้อมูลเข้า (Input Layer) ข้อมูลเข้าในชั้นนี้ได้จากการนำค่าไบนารี่ที่ได้จากการประมวลผลภาพเบื้องต้น กำหนดให้มีจำนวนโหนดในชั้นนี้เท่ากับจำนวนสมาชิกตัวแปรอะเรย์ที่เก็บค่าไบนารี่ โดยสามารถคำนวณได้จากผลคูณระหว่างความกว้างกับความสูงของตัวอักษรที่ผ่านการประมวลผลเบื้องต้นแล้ว ซึ่งตัวอักษรแต่ละตัวจะมีจำนวนโหนดในชั้นข้อมูลเข้าไม่เท่ากัน

จากเทคนิคการทำงานของเครือข่ายใยประสาท พบว่า ในแต่ละโหนดของระดับชั้นซ่อน จะได้รับข้อมูลนำเข้าชุดเดียวกัน แต่จะได้รับค่าน้ำหนักที่ต้องใช้ในการคำนวณของแต่ละ โหนดต่างกันนซึ่งค่าน้ำหนักดังกล่าวจะได้จากการสุ่ม โดยแบ่งเป็นชุดซึ่งมีจำนวนชุดของค่าน้ำหนักที่ ใช้เท่ากับ จำนวนโหนดในชั้นซ่อนหรือ 512 ชุด แต่ละชุดจะมีจำนวนสมาชิกเท่ากับจำนวนโหนดในชั้นข้อมูลเข้า ซึ่งตัวอักษรแต่ละตัวจะมีจำนวนโหนดในชั้นข้อมูลเข้าไม่เท่ากัน

กรณีของภาพตัวอักษร A เมื่อผ่านขั้นตอนการประมวลผลภาพเบื้องต้นแล้วจะได้จำนวนโหนดในชั้นข้อมูลเข้า 4818 โหนด และมีจำนวนโหนดในชั้นซ่อน 512 โหนด ในการหาผลลัพธ์ของการคำนวณระหว่างชั้นข้อมูลเข้าและชั้นซ่อนจะได้ว่า แต่ละโหนดของชั้นซ่อนจะต้องใช้ข้อมูลนำเข้า 4818 โหนดคำนวณร่วมกับค่าน้ำหนักจำนวน 4818 ค่า แล้วนำผลลัพธ์ที่ได้มาหาค่าผลรวมเพื่อนำไปหาค่าของฟังก์ชั่นกระตุ้นต่อไป ซึ่งทำให้มีการประมวลผลในลักษณะดังกล่าวในชั้นซ่อนเป็น 512 ชุด แต่ละชุดจะมีจำนวน 4818 ค่า

มหาวิทยาลัยศิลปากร ลิขสิทธิ์




รูปที่ 24 ตัวอย่างการส่งค่าระหว่างโหนดในชั้นข้อมูลนำเข้าไปยังชั้นซ่อนกรณีภาพตัวอักษร A

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

จากรูปที่ 24 โหนดแรกของชั้นข้อมูลเข้าจะมีค่าเป็น 1 ทำให้โหนดต่าง ๆ ในชั้นซ่อนที่เชื่อมต่อกับโหนดแรกของชั้นข้อมูลเข้าดังกล่าวได้รับค่าเป็น 1 เหมือนกันทุกโหนดแต่ค่าน้ำหนัก ที่ใช้ ในการคำนวณ จะได้จากการสุ่มทำให้ได้ค่าน้ำหนักไม่เหมือนกัน ทำให้มีลักษณะการคำนวณ ดังนี้

$x_{00}w_{00} = 1 \times 0.3 = 0.3$

$x_{10}w_{10} = 1 \times -0.6 = -0.6$

…

$x_{n-10}w_{n-10} = 1 \times -0.7 = -0.7$

$x_{n0}w_{n0} = 1 \times -0.4 = -0.4$

จากตัวอย่างการคำนวณหาค่าผลรวมของโหนดแรกในชั้นซ่อนหาได้จาก

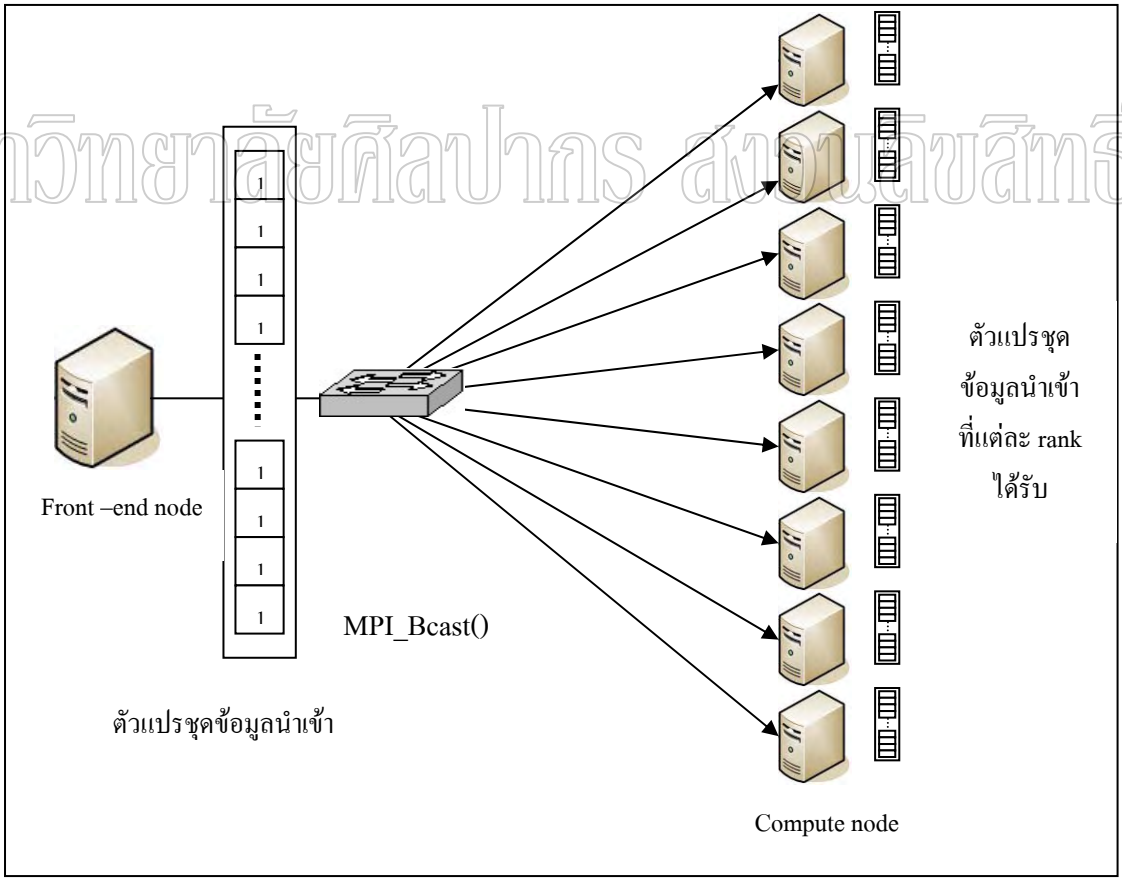
0.3+ (-0.6)+ 0+**…**+0+(-0.7)+0.4

จากการคำนวณในลักษณะดังกล่าว ผู้วิจัยจึงมีแนวคิดในการทำงาน ดังนี้

สำหรับข้อมูลเข้าจะถูกเก็บในตัวแปรอะเรย์ 1 มิติ ซึ่งผู้วิจัยได้เลือกใช้การกระจายตัวแปรอะเรย์ที่เก็บค่าข้อมูลเข้าด้วยฟังก์ชั่น MPI_Bcast ซึ่งเป็นฟังก์ชั่นที่จะทำการกระจาย

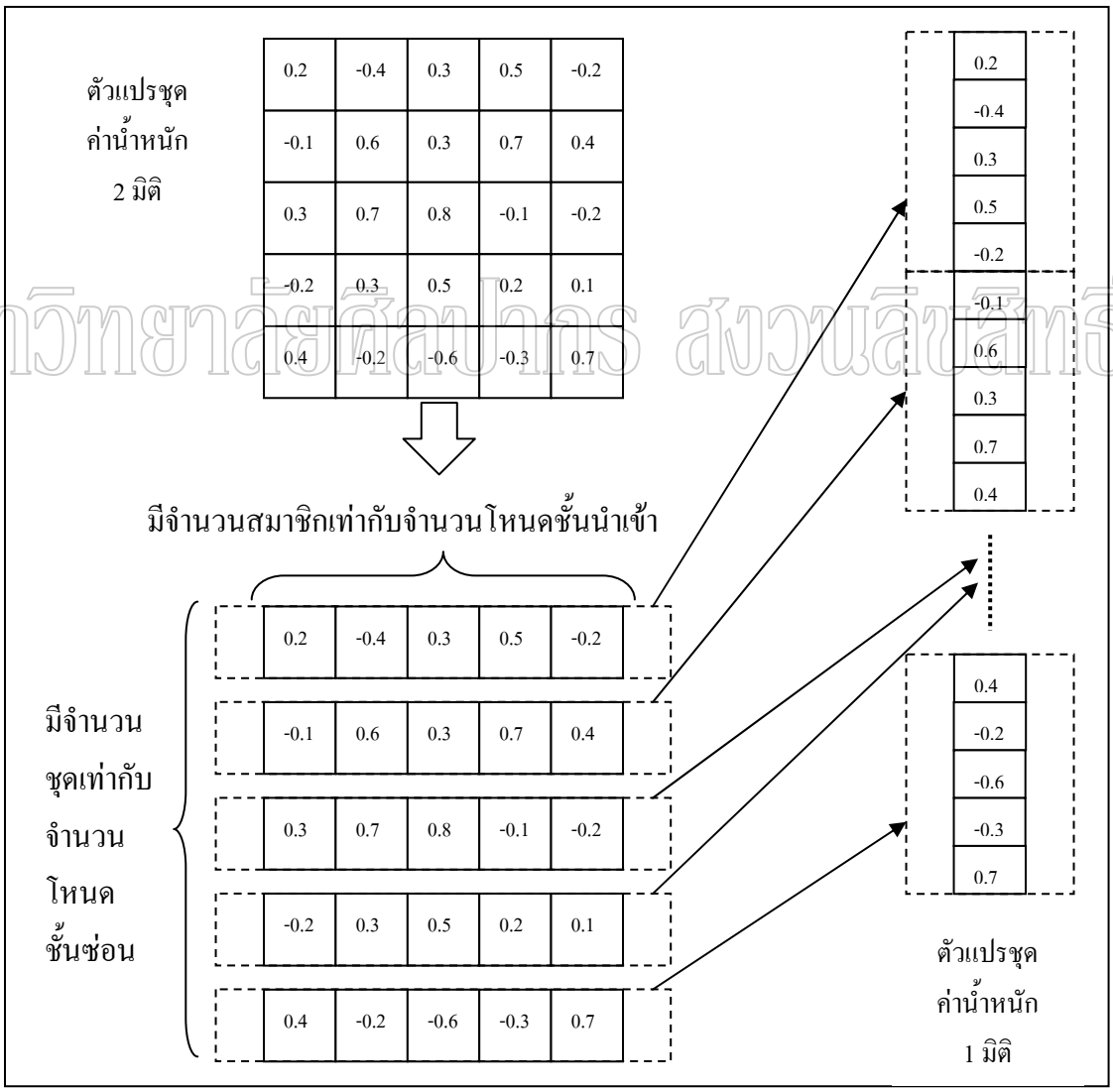
ตัวแปรอะเรย์ที่ต้องการไปยังทุก ๆ Rank โดยจะได้รับค่าของตัวแปรอะเรย์ที่มีค่าเหมือนกัน สาเหตุที่เลือกกระจายงานด้วย MPI_Bcast เนื่องจากการกระจายงานดังกล่าวจะช่วยลดขั้นตอนของการส่งและรับข้อมูลไปยัง Rank ต่าง ๆ ทั้งนี้นอกจาก MPI_Bcast ยังมีฟังก์ชั่นที่สามารถส่งข้อมูลไปยัง Rank ต่าง ๆ เช่น MPI_Send และ MPI_Isend ซึ่งเป็นการส่งข้อมูลระหว่าง 2 Rank ถ้ามีการส่งข้อมูลด้วยฟังก์ชั่นดังกล่าวจะต้องกำหนดให้ Rank ปลายทางมีการรับข้อมูลด้วย MPI_Recv หรือ MPI_Irecv ซึ่งกรณีที่ต้องการกระจายข้อมูลไปยังหลาย ๆ Rank จะต้องมีการเรียกใช้ฟังก์ชั่นดังกล่าวหลายรอบ นอกจากนี้ยังมี MPI_Sendrecv และ MPI_Sendrecv_replace ซึ่งเป็นฟังก์ชั่นที่สามารถกำหนด Rank ที่เป็นผู้ส่งและ Rank ที่เป็นผู้รับได้ด้วยการเรียกใช้ฟังก์ชั่นเดียว แต่ทั้ง 2 ฟังก์ชั่นเป็นการรับส่งข้อมูลระหว่าง 2 Rank ทำให้ไม่สะดวกในการกระจายงานไปยังหลาย ๆ Rank ทั้งนี้การกระจายงานด้วย MPI_Bcast ไม่จำเป็นต้องเรียกใช้ฟังก์ชั่นในการรับข้อมูล ทำให้สะดวกในการใช้งานและประหยัดเวลาในการส่งและรับข้อมูล โดยสามารถแสดงดังรูปที่ 25

มหาวิทยาลัยศิลปากร ลิขสิทธิ์




รูปที่ 25 การกระจายตัวแปรชุดข้อมูลนำเข้าด้วย MPI_Bcast

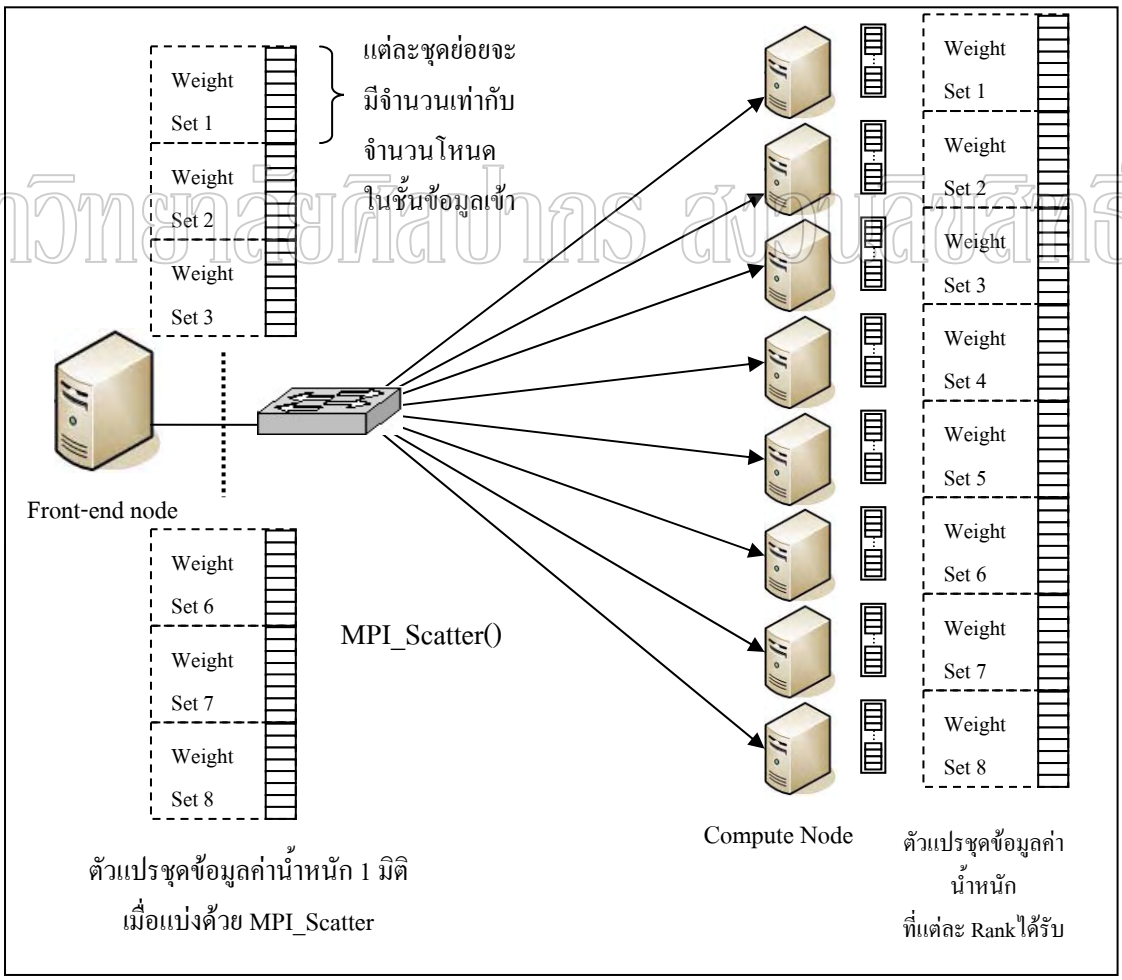
ในการวิจัยนี้ได้ทำการสุ่มค่าน้ำหนักที่ต้องใช้ในการวิจัย ตั้งแต่เริ่มต้นการประมวลผล ซึ่ง อยู่ในขั้นตอนของ Weight Initialization โดยเก็บค่าน้ำหนักไว้ในตัวแปรอะเรย์ 2 มิติ ทำให้สามารถแบ่งค่าน้ำหนักออกเป็นชุดย่อย ที่มีจำนวนสมาชิกในแต่ละชุดเท่ากับจำนวนสมาชิกของตัว แปรอะเรย์ที่เก็บค่าข้อมูลเข้าหรือเท่ากับจำนวนโหนดในชั้นข้อมูลเข้านั่นเอง โดยจำนวนชุดของค่าน้ำหนักจะเท่ากับจำนวนโหนดในชั้นซ่อน ข้อมูลตัวแปรอะเรย์ชุดย่อยที่เก็บค่าน้ำหนักจะถูกกระจายไปยัง Rank ต่าง ๆ ตามต้องการ ลักษณะการแบ่งตัวแปรอะเรย์ค่าน้ำหนักแสดงดังรูปที่ 26




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

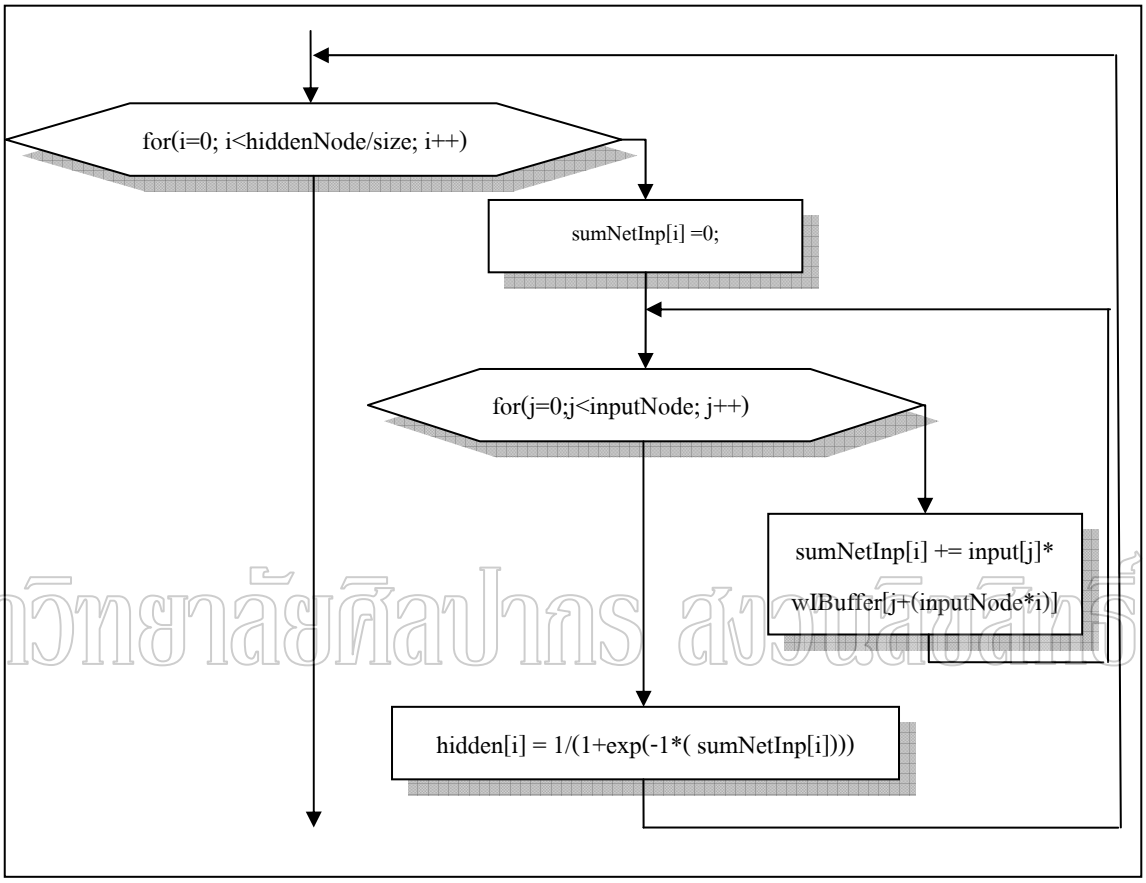
รูปที่ 26 การแบ่งข้อมูลจากตัวแปรชุดข้อมูลค่าน้ำหนัก 2 มิติ เป็นตัวแปรชุด 1 มิติ

สำหรับข้อมูลค่าน้ำหนักผู้วิจัยเลือกใช้ฟังก์ชั่น MPI_Scatter ซึ่งเป็นฟังก์ชั่นที่สามารถแบ่งข้อมูลออกเป็นชุดย่อยที่มีขนาดหรือจำนวนสมาชิกของข้อมูลเท่ากัน และกระจายข้อมูลดังกล่าวไปยัง Rank ต่าง ๆ โดยการส่งข้อมูลดังกล่าวจะเป็นการกระจายข้อมูลชุดย่อยแต่ละชุดไปยัง Rank ต่าง ๆ ไม่ซ้ำกัน ทั้งนี้ขนาดของชุดย่อยที่แบ่งจะมีขนาดเท่ากับผลคูณของจำนวนโหนด ในชั้นข้อมูลเข้ากับผลหารที่ได้จากการหารจำนวนโหนดในชั้นซ่อนด้วยจำนวนโปรเซส เช่น กรณีของตัวอักษร A ขนาดในชั้นข้อมูลเข้า 4818 โหนด แบ่งการประมวลผลออกเป็น 8 โปรเซส จะมีการกระจายชุดของค่าน้ำหนักที่มีขนาดเท่ากับ 4818 x (512/8) เพื่อให้แต่ละ Rank สามารถนำไปประมวลผลร่วมกับตัวแปรชุดของค่าข้อมูลเข้าที่ได้รับไปก่อนหน้านี้ ทำให้ในแต่ละ Rank จะทำการประมวลผลเพื่อหาผลลัพธ์ของโหนดในชั้นซ่อน 64 โหนด ซึ่งการกระจายข้อมูลค่าน้ำหนักสามารถแสดงได้ดังรูปที่ 27




รูปที่ 27 การกระจายตัวแปรชุดข้อมูลค่าน้ำหนักด้วย MPI_Scatter

เมื่อแต่ละ Rank ได้รับชุดของตัวแปรอะเรย์ที่เก็บข้อมูลเข้าและค่าน้ำหนัก แล้วจะทำการประมวลผลเพื่อหาค่าของฟังก์ชั่นกระตุ้น ซึ่งสำหรับงานวิจัยนี้เลือกใช้ซิกมอยด์ ฟังก์ชั่น ดังรูปที่ 28



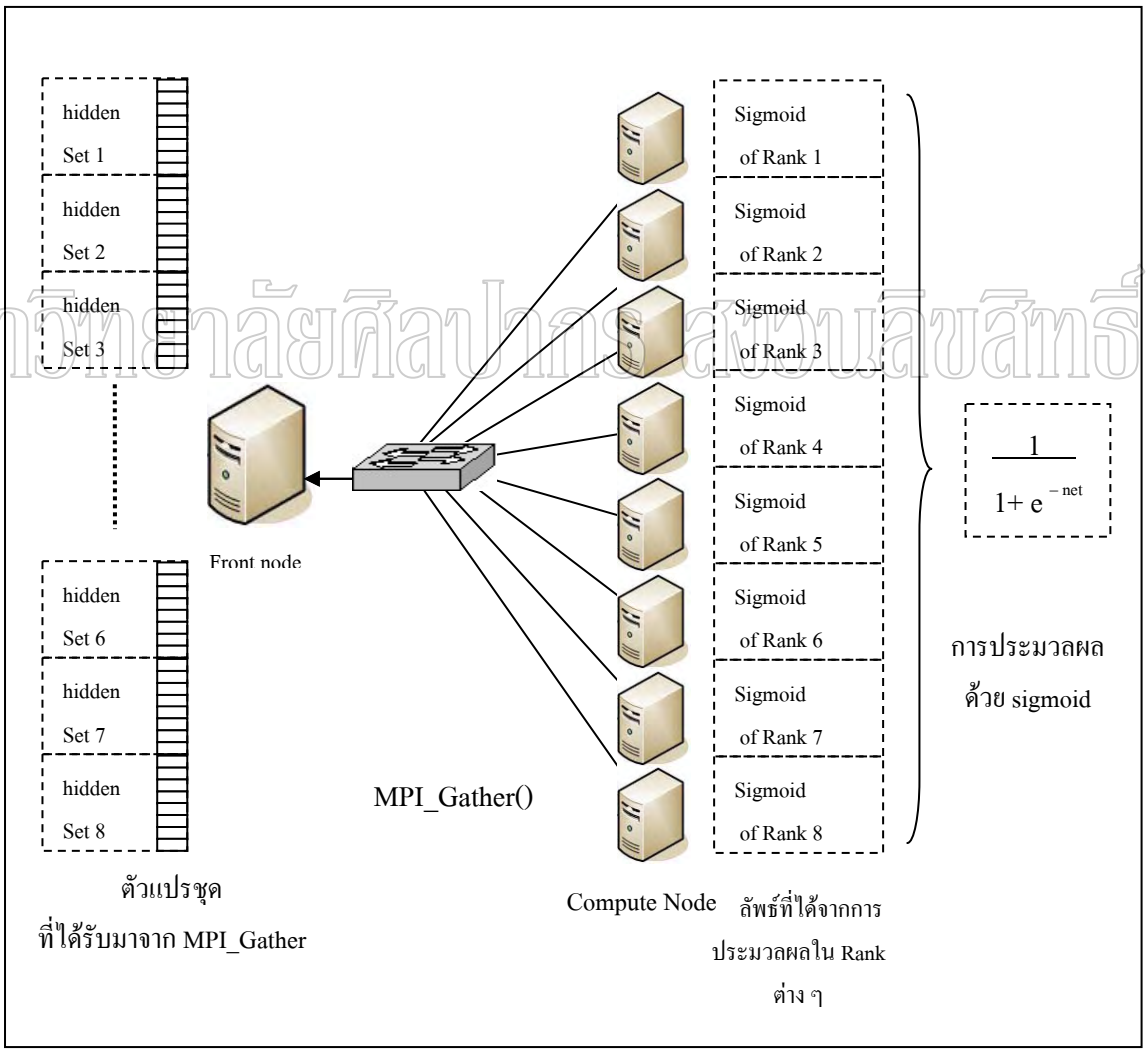

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 28 การคำนวณของแต่ละโปรเซสที่กระจายกันประมวลผลตาม Rank ต่างๆ

จากรูปที่ 28 เป็นการประมวลผลในแต่ละ Rank โดยมีการวนรอบเพื่อหาผลรวมของการคำนวณระหว่างค่าข้อมูลเข้ากับค่าน้ำหนักที่ได้รับมาจาก Rank 0 โดยกำหนดให้ input[j] เป็นตัวแปรอะเรย์ของข้อมูลเข้าและ wIBuffer[i] เป็นตัวแปรอะเรย์ของค่าน้ำหนัก เมื่อได้ค่าผลรวมแล้วในแต่ละ Rank จะประมวลผลเพื่อหาค่าของฟังก์ชั่นกระตุ้นของแต่ละโหนดในชั้นซ่อน ทำให้ในแต่ละ Rank จะได้ค่าผลลัพธ์เท่ากับผลหารระหว่างจำนวนโหนดของชั้นซ่อนกับจำนวนโปรเซสที่ประมวลผล

ในงานวิจัยนี้กำหนดให้มีจำนวนโหนดในชั้นซ่อน 512 โหนด โดยแต่ละ Rank จะทำการประมวลผลในส่วนของข้อมูลที่ได้รับ เช่น ถ้ามีจำนวน Rank ในการทำงานเท่ากับ 8 จำนวนโหนดของชั้นซ่อนที่แต่ละ Rank จะประมวลผลจะได้จากจำนวนโหนดในชั้นซ่อนคือ 512 โหนด หารด้วยจำนวน Rank คือ 8 ทำให้แต่ละ Rank จะทำการประมวลผลเพื่อหาผลลัพธ์ของโหนดในชั้นซ่อนจำนวน 64 โหนด

การส่งค่าผลลัพธ์กลับมายัง Rank 0 ผู้วิจัยได้เลือกใช้ฟังก์ชั่น MPI_Gather() ซึ่งมีการทำงานโดยจะนำค่าผลลัพธ์ที่ได้ในแต่ละ Rank ส่งกลับมาเก็บเป็นตัวแปรอะเรย์ที่ Rank 0 ดังแสดงในรูปที่ 29




มหาวิทยาลัยศิลปากรลิขสิทธิ์

รูปที่ 29 การรับค่าตัวแปรชุดที่ส่งมาจาก Rank ต่างๆด้วย MPI_Gather

5.3.2 ชั้นซ่อน (Hidden Layer) เป็นชั้นที่นำผลลัพธ์จากชั้นข้อมูลเข้ามาใช้เป็นข้อมูลเริ่มต้นในการประมวลผล โดยกำหนดให้มีจำนวนโหนดในชั้นซ่อนเป็น 512 โหนด ซึ่งผลลัพธ์ที่ได้จากการประมวลผลของชั้นซ่อนจะมีจำนวนเท่ากับจำนวนของชั้นผลลัพธ์ คือ 8 โหนด

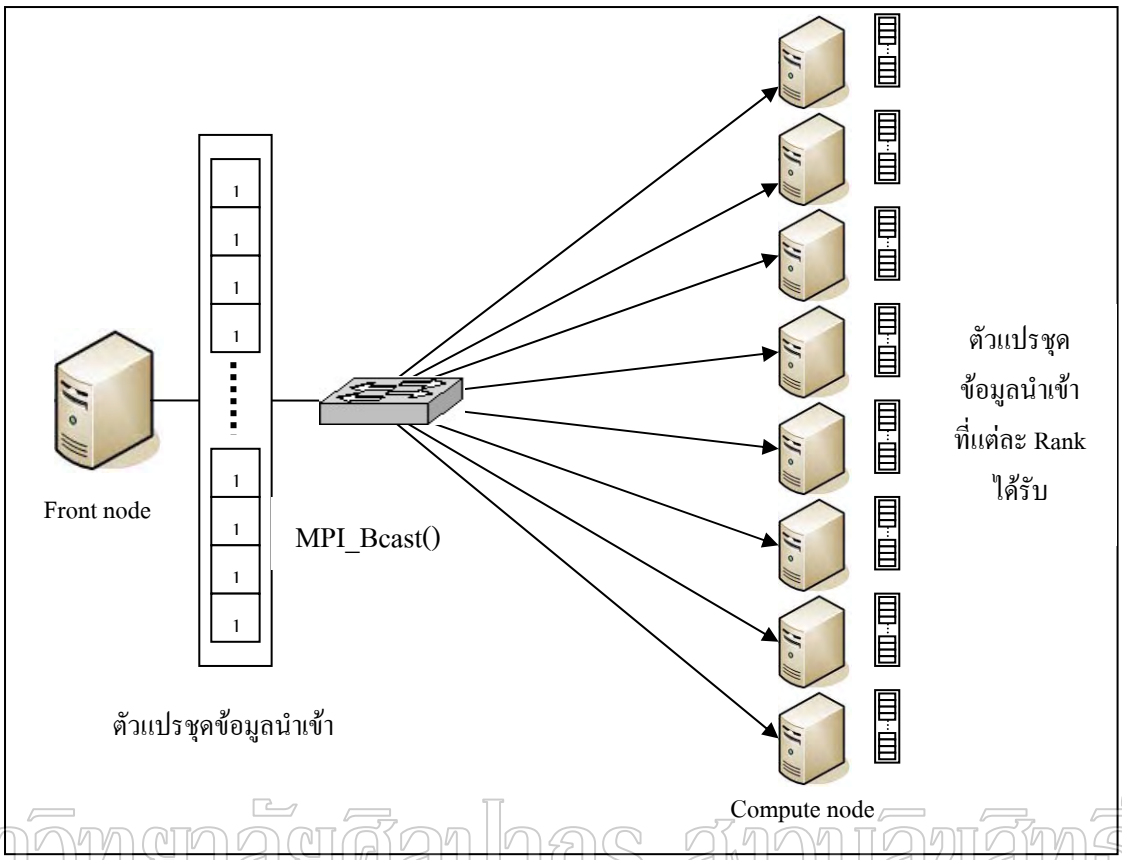
สำหรับการประมวลในชั้นซ่อนจะมีขั้นตอนการกระจายข้อมูลของโหนดในชั้นซ่อนด้วย MPI_Bcast การแบ่งและกระจายข้อมูลค่าน้ำหนักด้วย MPI_Scatter และทำการรวบรวมผลลัพธ์จาก Rank ต่างๆ ด้วย MPI_Gather เช่นเดียวกับการประมวลผลในชั้นข้อมูลเข้า แต่เปลี่ยนจำนวนโหนดเริ่มต้นเป็นจำนวนโหนดของชั้นซ่อน และจำนวนผลลัพธ์จะเท่ากับจำนวนโหนดของชั้นผลลัพธ์

5.3.3 ชั้นผลลัพธ์ (Output Layer) เป็นชั้นสุดท้ายของการประมวลผลในแต่ละรอบของการเรียนรู้ โดยกำหนดให้มีจำนวน 8 โหนด

เมื่อประมวลผลมาจนถึงชั้นผลลัพธ์ ผลลัพธ์ที่ได้จะถูกนำมาเปรียบเทียบกับค่าเป้าหมายที่คาดหวังเพื่อหาข้อผิดพลาด และทำการส่งค่าผิดพลาดที่ได้กลับไปยังระดับชั้นก่อนหน้าเพื่อทำการปรับค่าถ่วงน้ำหนักให้เกิดความเหมาะสม และถ้าค่าผิดพลาดที่ได้ยังมากกว่าค่าที่ยอมรับได้ จะทำการวนรอบเพื่อทำการประมวลผลซ้ำโดยใช้ค่าถ่วงน้ำหนักที่ปรับค่าแล้ว

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

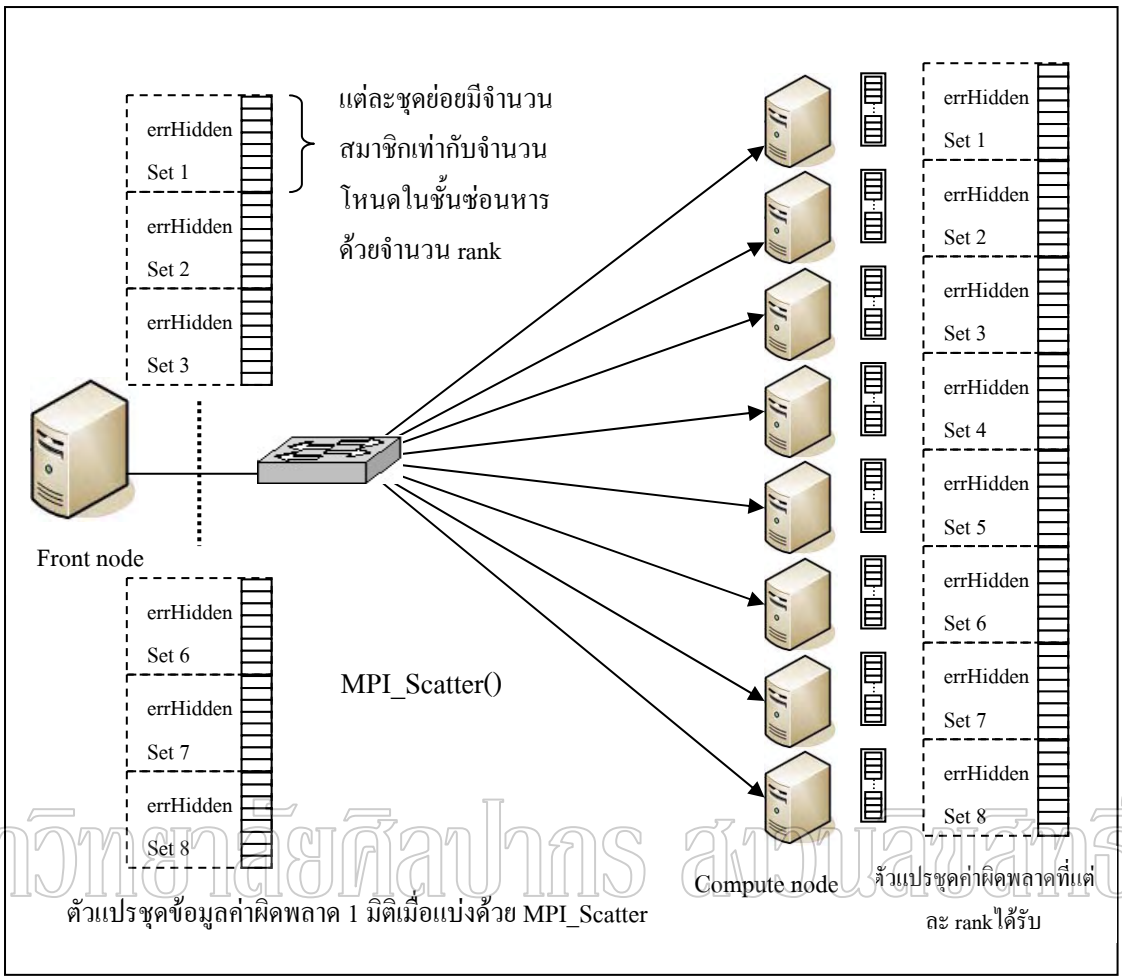
ในการทำงานของ Backpropagation จะต้องมีการปรับค่าน้ำหนักเพื่อให้ผลลัพธ์ที่ ได้เข้า ใกล้เป้าหมายมากที่สุด แต่เนื่องจากมีข้อมูลนำเข้าจำนวนมาก ทำให้สูญเสียเวลาการปรับค่าน้ำหนักที่อยู่ ระหว่างชั้นข้อมูลเข้ากับชั้นซ่อนมาก ผู้วิจัยจึงทำการปรับค่าน้ำหนักบนระบบกระจาย ซึ่ง เป็นการ คำนวณระหว่างค่าอัตราการเรียนรู้ ค่าผิดพลาดที่ได้ และค่าข้อมูลเข้า โดยค่าอัตราการเรียนรู้จะเป็นค่าตัวเลขที่ใช้สำหรับการเรียนรู้ ซึ่งถ้ากำหนดมากไปจะทำให้มีการเรียนรู้เร็วขึ้นแต่ถ้ากำหนดน้อยไปจะทำให้การเรียนรู้ช้า ซึ่งค่าอัตราการเรียนรู้สำหรับงานวิจัยนี้กำหนดให้มีค่าเป็น 0.35 ซึ่งเป็นค่าที่ได้จากการทดลอง โดยการกระจายตัวแปรอะเรย์ของข้อมูลเข้าได้เลือกใช้ฟังก์ชั่น MPI_Bcast() เนื่องจากทุก ๆ Rank จะต้องใช้ข้อมูลเข้าชุดเดียวกันในการปรับค่าถ่วงน้ำหนักทั้งหมด ดังแสดงในรูปที่ 30




มหาวิทยาลัยศิลปากร ลิขสิทธิ์

รูปที่ 30 การกระจายตัวแปรชุดข้อมูลเข้าด้วย MPI_Bcast

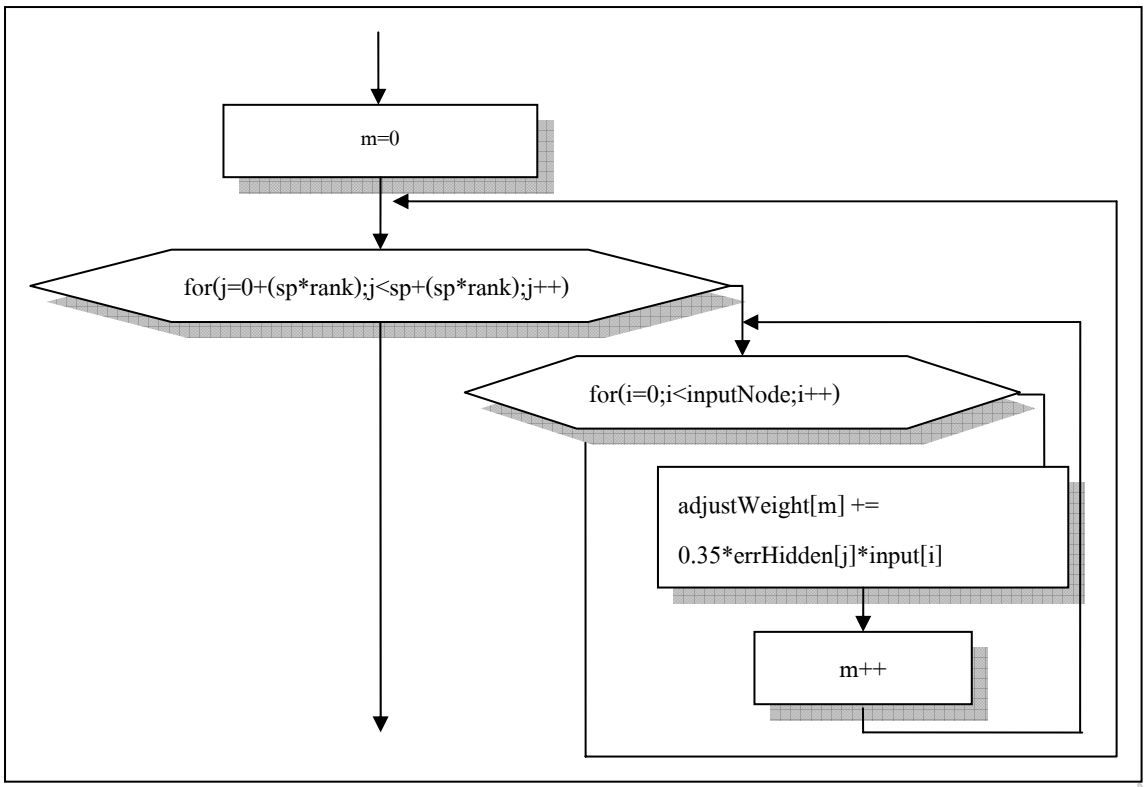
สำหรับค่าผิดพลาดที่ได้จากการประมวลผลในโหนดต่างๆ ซึ่งถูกเก็บอยู่ในตัวแปรอะเรย์ซึ่งจะมีขนาดเท่ากับจำนวนโหนดของชั้นซ่อน คือ 512 โหนด ผู้วิจัยได้เลือกใช้ฟังก์ชั่น MPI_Scatter() ในการแบ่งและกระจายตัวแปรชุดค่าผิดพลาดไปยัง Rank ต่าง ๆ โดยแต่ละ Rank จะได้รับค่าผิดพลาดที่มีขนาดหรือจำนวนสมาชิกเท่ากับผลหารของจำนวนโหนดของชั้นซ่อนกับจำนวน Rank ที่ประมวลผล เช่น ถ้ามีจำนวน Rank เท่ากับ 8 แต่ละ Rank จะได้รับค่าผิดพลาดที่มีขนาดหรือจำนวนสมาชิกเท่า 64 เนื่องจากค่าผิดพลาดในแต่ละชุดที่ใช้สำหรับคำนวณในแต่ละโหนดของชั้นซ่อน จะมีค่าไม่เหมือนกัน เพื่อให้แต่ละ Rank สามารถนำไปประมวลผลร่วมกับตัวแปรชุดของข้อมูลเข้าที่ได้รับไปก่อนหน้านี้ โครงสร้างการกระจายข้อมูลค่าผิดพลาดสามารถแสดงได้ดังรูปที่ 31




รูปที่ 31 การกระจายตัวแปรชุดข้อมูลค่าผิดพลาดด้วย MPI_Scatter

เมื่อแต่ละ Rank ได้รับชุดของตัวแปรชุดข้อมูลเข้าและค่าผิดพลาดแล้ว จะทำการประมวล ผลเพื่อปรับค่าน้ำหนัก ดังรูปที่ 32

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

รูปที่ 32 การประมวลผลเพื่อปรับค่าถ่วงน้ำหนัก

**5.4 ขั้นตอนการวัดเวลาในการประมวลผลการเรียนรู้**

การจับเวลาในการประมวลผลทำได้โดยการเรียกใช้ฟังก์ชั่นเกี่ยวกับเวลาของ MPI คือ MPI_Wtime เพื่ออ่านค่าเวลา ณ ขณะที่มีการเริ่มต้นการเรียนรู้ และอ่านค่าเวลาอีกครั้ง ณ ขณะที่การเรียนรู้เสร็จสิ้น จากนั้นนำค่าเวลาที่ได้มาลบกัน โดยมีหน่วยเวลาเป็นวินาที

**6. การประเมินผล**

เนื่องจากงานวิจัยนี้มีการวัดประสิทธิภาพด้านความเร็วในการประมวลผลจึงต้องมีการเก็บเวลาที่ใช้ในการประมวลผลการเรียนรู้เมื่อใช้จำนวนหน่วยประมวลผลที่ต่างกัน ทั้งนี้ในการทดลองได้ทำการแบ่งจำนวนโปรเซสเป็น 1, 2, 4 และ 8 ตามลำดับ และวัดค่าเวลาด้วยขั้นตอนที่ (4.1) โดยทำการเก็บค่าเวลาในการประมวลผลการเรียนรู้ของระบบเป็นจำนวน 10 ครั้งของการทำงานที่มีหน่วยประมวลผลในจำนวนต่าง ๆ จากนั้นทำการหาค่าผลรวมและหารด้วยจำนวนครั้งที่ทำการเก็บเวลา ด้วยขั้นตอนที่ (4.2) เพื่อให้เห็นถึงค่าเวลาเฉลี่ยที่เกิดขึ้น

$$\text{Process time}_i = \text{End time}_i - \text{Start time}_i \quad \text{.....(4.1)}$$

$$\text{Average process time} = \sum_{i=0}^{n} \text{Process time}_i/\text{round} \quad \text{.....(4.2)}$$

โดย Process time เป็นค่าเวลาที่ได้จากการประมวลผลการเรียนรู้ในแต่ละครั้ง ซึ่งได้จากผลต่างระหว่าง End time ที่เป็นค่าเวลาเมื่อการประมวลผลเสร็จสิ้น และ Start time ที่เป็นค่าเวลาที่เริ่มต้นประมวลผลการเรียนรู้

Average process time เป็นค่าเวลาที่ได้จากการประมวลผลการเรียนรู้เฉลี่ย โดยการนำผลรวมของค่าเวลาจากการประมวลผลการเรียนรู้ในแต่ละครั้งหารด้วยจำนวนที่เก็บค่าเวลาดังกล่าว

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# บทที่ 5
# ผลการดำเนินงานวิจัย

สำหรับผลการดำเนินงานวิจัย ทางผู้วิจัยได้ทำการศึกษา วิเคราะห์ และทำการทดลองตลอดจนพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด ซึ่งมีรายละเอียดของการทดลองและผลการทดลอง ดังต่อไปนี้

## 1. ตัวอย่างภาพตัวอักษรที่ใช้ในการวิจัย

เพื่อทำการวิจัยเกี่ยวกับการฝึกสอนและการรู้จำตัวอักษร ผู้วิจัยจึงได้เลือกใช้ตัวอักษรภาษาอังกฤษโดยมีคุณลักษณะ ดังนี้

1.1 ตัวอักษรภาษาอังกฤษ ตัวพิมพ์ใหญ่ A - Z

1.2 รูปภาพตัวอักษรแต่ละภาพ จะมีสีของตัวอักษรและสีพื้นหลังที่แตกต่างกัน เพื่อให้ง่ายต่อการอ่านค่าสีสำหรับการแปลงเป็นภาพขาว-ดำ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

1.3 ไฟล์ภาพที่ใช้เป็นไฟล์ภาพชนิด BMP

1.4 ตัวอักษรที่กำหนดมีขนาดกว้าง 100 พิกเซล และสูง 100 พิกเซล

1.5 จำนวนภาพตัวอักษรที่ใช้ในการเรียนรู้มีทั้งหมด 26 รูปแบบ จำนวน 26 ภาพ

1.6 จำนวนภาพตัวอักษรที่ใช้ในการทดสอบมีทั้งหมด 260 ภาพ

## 2. ผลการวัดประสิทธิภาพการประมวลผล

เมื่อผ่านการประมวลผลภาพเบื่องต้นแล้วจะได้จำนวนโหนดในชั้นข้อมูลเข้า (Input layer) ของตัวอักษรแต่ละตัว และเนื่องจากงานวิจัยนี้ต้องการวัดประสิทธิภาพการประมวลผลของโปรแกรมประยุกต์ที่พัฒนาขึ้น จึงทำการทดลองโดยให้มีการประมวลผลการเรียนรู้ ซึ่งสามารถสรุปผลการวัดประสิทธิภาพการประมวลผล ดังนี้

ตารางที่ 3 สรุปผลการวัดประสิทธิภาพการประมวลผลการเรียนรู้

| ตัวอักษร | เวลาที่ใช้ในการเรียนรู้ (Training Time) | | | | อัตราความเร็วในการเรียนรู้ (Speedup) | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 1 | 2 | 4 | 8 |
| A | 19.198 | 11.257 | 6.023 | 3.031 | 1.00 | 1.71 | 3.19 | 6.33 |
| B | 17.168 | 10.638 | 5.705 | 2.796 | 1.00 | 1.61 | 3.01 | 6.14 |
| C | 18.998 | 11.684 | 6.152 | 3.091 | 1.00 | 1.63 | 3.09 | 6.15 |
| D | 18.953 | 11.295 | 6.417 | 3.104 | 1.00 | 1.68 | 2.95 | 6.11 |
| E | 16.11 | 9.524 | 5.084 | 2.794 | 1.00 | 1.69 | 3.17 | 5.77 |
| F | 15.977 | 10.013 | 6.021 | 3.172 | 1.00 | 1.60 | 2.65 | 5.04 |
| G | 21.121 | 12.015 | 7.019 | 3.674 | 1.00 | 1.76 | 3.01 | 5.75 |
| H | 18.236 | 11.23 | 6.317 | 3.106 | 1.00 | 1.62 | 2.89 | 5.87 |
| I | 7.83 | 4.31 | 3.069 | 2.241 | 1.00 | 1.82 | 2.55 | 3.49 |
| J | 7.323 | 4.613 | 3.223 | 2.523 | 1.00 | 1.59 | 2.27 | 2.90 |
| K | 17.32 | 9.769 | 5.84 | 3.014 | 1.00 | 1.77 | 2.97 | 5.75 |
| L | 13.336 | 7.84 | 4.891 | 2.711 | 1.00 | 1.70 | 2.73 | 4.92 |
| M | 19.81 | 12.173 | 6.369 | 3.318 | 1.00 | 1.63 | 3.11 | 5.97 |
| N | 17.251 | 10.873 | 5.506 | 3.17 | 1.00 | 1.59 | 3.13 | 5.44 |
| O | 21.2 | 13.091 | 7.024 | 3.445 | 1.00 | 1.62 | 3.02 | 6.15 |
| P | 17.376 | 10.176 | 5.676 | 3.026 | 1.00 | 1.71 | 3.06 | 5.74 |
| Q | 22.305 | 13.601 | 7.75 | 3.721 | 1.00 | 1.64 | 2.88 | 5.99 |
| R | 17.278 | 10.53 | 6.103 | 3.108 | 1.00 | 1.64 | 2.83 | 5.56 |
| S | 17.578 | 10.395 | 6.13 | 3.98 | 1.00 | 1.69 | 2.87 | 4.42 |
| T | 18.241 | 11.241 | 6.531 | 3.137 | 1.00 | 1.62 | 2.79 | 5.81 |
| U | 18.715 | 10.715 | 5.815 | 3.171 | 1.00 | 1.75 | 3.22 | 5.90 |
| V | 19.659 | 11.63 | 6.311 | 3.121 | 1.00 | 1.69 | 3.12 | 6.30 |

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

ตารางที่ 3 สรุปผลการวัดประสิทธิภาพการประมวลผลการเรียนรู้ (ต่อ)

| ตัวอักษร | เวลาที่ใช้ในการเรียนรู้ (Training Time) | | | | อัตราความเร็วในการเรียนรู้ (Speedup) | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 1 | 2 | 4 | 8 |
| W | 23.732 | 14.201 | 8.193 | 4.217 | 1.00 | 1.67 | 2.90 | 5.63 |
| X | 18.21 | 10.891 | 6.31 | 3.221 | 1.00 | 1.67 | 2.89 | 5.65 |
| Y | 17.56 | 9.612 | 5.7061 | 3.011 | 1.00 | 1.83 | 3.08 | 5.83 |
| Z | 16.027 | 8.793 | 5.463 | 3.104 | 1.00 | 1.82 | 2.93 | 5.16 |

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# บทที่ 6
# อภิปรายผล สรุป และข้อเสนอแนะ

ในบทนี้จะเป็นการอภิปรายถึงผลการวิจัย อันนำไปสู่การสรุปผลการวิจัยการพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด พร้อมทั้งข้อเสนอแนะในการนำระบบไปปรับปรุง เพิ่มเติม หรือใช้เป็นแนวทางในการพัฒนาระบบในอนาคต โดยมีรายละเอียดดังนี้

## 1. อภิปรายผลการวิจัย

จากสมมติฐานของผู้วิจัยก่อนทำการวิจัยที่มีอยู่ว่า เนื่องจากการเรียนรู้ของเทคนิคเครือข่ายใยประสาทเป็นการเรียนรู้โดยแบ่งโครงสร้างการเรียนรู้ออกเป็นระดับชั้น และในแต่ละระดับชั้นมีการประมวลผลของโหนดต่าง ๆ ซึ่งเป็นการประมวลผลที่มีความเป็นอิสระต่อกัน ทำให้มีแนวคิดว่า การนำความสามารถในการประมวลผลการเรียนรู้ดังกล่าวมาทำการกระจายไปยังหน่วยประมวลผลจำนวนมากที่มีการทำงานอิสระต่อกัน เพื่อให้เกิดการแบ่งงานหรือช่วยกันทำงานจะทำให้การเรียนรู้ของเทคนิคเครือข่ายใยประสาทมีความเร็วเพิ่มขึ้น ทั้งนี้ขึ้นอยู่กับปัจจัยที่สำคัญ 2 ประการ คือ จำนวนหน่วยประมวลผลภายในระบบที่มีการแบ่งงานกัน และการสูญเสียเวลาในการรับส่งข้อมูลบนระบบเครือข่าย จากสมมติฐานดังกล่าว นำไปสู่การวิจัยและพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด

การพัฒนาโปรแกรมประยุกต์ตามที่ได้ตั้งสมมติฐานไว้ เริ่มจากการศึกษาองค์ความรู้เกี่ยวกับขั้นตอนการเรียนรู้ด้วยเทคนิคเครือข่ายใยประสาท และหลักการทำงานของเทคโนโลยีกริด นำไปสู่การประมวลผลแบบกริด ทำให้ทราบว่า การพัฒนาโปรแกรมประยุกต์ที่มีการประมวลผลแบบ กริด เป็นการพัฒนาโปรแกรมประยุกต์ให้มีความสามารถในการกระจายงานไปประมวลผลยังหน่วยประมวลผลต่างๆ ซึ่งเทคนิคที่นิยมใช้ในการทำงาน คือ เทคโนโลยี Message Passing Interface หรือ MPI

สำหรับเครื่องมือในการพัฒนาโปรแกรมประยุกต์ผู้วิจัยได้เลือกใช้ซอร์ฟแวร์ ตลอดจนระบบปฏิบัติการที่ไม่ข้องเกี่ยวกับเรื่องลิขสิทธิ์ และสามารถเปิดเผยขั้นตอนการทำงานได้ ทั้งนี้เพื่อให้งานวิจัยนี้สามารถใช้เป็นแนวทางในการพัฒนาความสามารถของโปรแกรมประยุกต์แบบ Open Source ต่อไป

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

จากสมมติฐานและแนวความคิดในการเลือกใช้เครื่องมือในการพัฒนา ทำให้เกิดอุปสรรคในการวิจัยหลายด้าน เช่น

- งานวิจัยนี้ผู้วิจัยได้เลือกพัฒนาโปรแกรมประยุกต์การรู้จำภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ เนื่องจากภาพมีขนาดเล็กทำให้ใช้เวลาในการประมวลไม่มากนัก ทำให้ผลการทดลองที่ได้ไม่เกิดเห็นปรสิทธิภาพมากนัก

- งานวิจัยนี้ได้ทำการวิจัยผ่านเครือข่ายอินเตอร์เน็ตทำให้ผลลัพธ์ที่ได้จากการจับเวลามีความคลาดเคลื่อนอยู่บ้าง เนื่องจากความไม่แน่นอนของแบรนด์วิดของผู้ให้บริการเชื่อมต่ออินเตอร์เน็ตที่ผู้วิจัยใช้

- ผู้วิจัยใช้เวลาในการทำความเข้าใจเกี่ยวกับการรู้จำด้วยเครือข่ายใยประสาทและการพัฒนาระบบงานแบบกระจายเป็นเวลานาน เนื่องจากมีความรู้เดิมเกี่ยวกับเรื่องดังกล่าวค่อนข้างน้อย

จากสมมติฐานและอุปสรรคที่พบระหว่างการวิจัยทำให้ผู้วิจัยได้หาแนวทางในการแก้ไขอุปสรรคต่าง ๆ ที่เกิดขึ้น โดยการศึกษาข้อมูลจากแหล่งต่าง ๆ เพิ่มเติมเพื่อใช้ประกอบผลการทดลอง จากผลการดำเนินงานวิจัยในบทที่ 5 และการอภิปรายในเบื้องต้นทำให้ทราบถึงข้อเท็จจริงเกี่ยวกับสมมติฐานที่ผู้วิจัยได้ตั้งไว้ ดังนี้

- จากขั้นตอนของเทคนิคเครือข่ายใยประสาทที่มีอยู่หลายขั้นตอน พบว่าขั้นตอนที่มีการประมวลผลระหว่างค่าข้อมูลเข้ากับค่าน้ำหนักซึ่งจะถูกส่งไปหาค่าผลรวมของโหนด ในระดับชั้นถัดไป เป็นการประมวลผลที่ความเป็นอิสระที่สุด ผู้วิจัยจึงเลือกขั้นตอนดังกล่าวในการกระจายไปประมวลผลยังหน่วยประมวลผลต่าง ๆ และส่งผลลัพธ์กลับมาเพื่อหาค่าผลลัพธ์ต่อไป

- ขั้นตอนอื่น ๆ ที่เกี่ยวข้องยังคงเป็นการประมวลผลบนเครื่องหลัก เนื่องจากเป็นขั้นตอนพื้นฐานที่ไม่จำเป็นต่อทำการกระจายการประมวลผล เช่น

- ขั้นตอนการนำเข้าภาพตัวอักษร เนื่องจากเป็นขั้นตอนที่ไม่มีการประมวลผลและเป็นขั้นตอนที่ไม่มีส่วนในการวัดประสิทธิภาพการทำงานด้านความเร็ว

- ขั้นตอนการประมวลผลภาพเบื้องต้น ซึ่งเป็นขั้นตอนที่จะเปลี่ยนไปตามความต้องการของผู้พัฒนาโปรแกรมแต่ละคน เช่น บางระบบอาจเพิ่มขั้นตอนการค้นหาขอบภาพ หรือบางระบบอาจมีการเพิ่มขั้นตอนการลดสัญญาณรบกวน หรือขั้นตอนการปรับขนาดของภาพ เป็นต้น สำหรับงานวิจัยนี้เลือกใช้การประมวลผลภาพอย่างง่าย เช่น การแปลงเป็นภาพขาว-ดำ การตัดขอบภาพโดยเลือกตัดขอบด้านต่าง ๆ คือ ด้านบน ล่าง ซ้าย และขวา เนื่องจากขั้นตอนต่าง ๆ ไม่ได้อยู่ในส่วนสำคัญสำหรับการเพิ่มประสิทธิภาพการเรียนรู้

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

- ขั้นตอนการแสดงผลลัพธ์ที่ได้จากการเรียนรู้ และรู้จำ

- ถ้าขนาดของข้อความที่ส่งและรับบนระบบมีขนาดเล็กเกินไป จะทำให้การกระจายงานไม่เกิดประสิทธิภาพเท่าที่ควร เนื่องจากการสูญเสียเวลาในการส่งและรับข้อความ

- ขนาดของข้อความที่ส่งและรับในชั้นข้อมูลเข้าจะมีขนาดของข้อความน้อยที่สุดคือ ตัวเลขจำนวนเต็มที่เป็นข้อมูลเข้า 803 จำนวน และตัวเลขจำนวนจริงที่เป็นค่านำหนัก 411136 จำนวน ซึ่งคำนวนได้จากจำนวนของข้อมูลเข้าของตัวอักษร I ซึ่งมีจำนวนน้อยที่สุดคูณกับจำนวนโหนดชั้นซ่อนคือ 512 โหนดและขนาดของข้อความมากที่สุดคือ ตัวเลขจำนวนเต็มที่เป็นข้อมูลเข้า 6789 จำนวน และตัวเลขจำนวนจริงที่เป็นค่าน้ำหนัก 3475968จำนวน ซึ่งคำนวนได้จากจำนวนของข้อมูลเข้าของตัวอักษร W ซึ่งมีจำนวนมากที่สุดคูณกับจำนวนโหนดชั้นซ่อนคือ 512 โหนด

- ขนาดของข้อความที่ส่งและรับในชั้นซ่อนจะมีขนาดของข้อความคือ จำนวนจริงที่เป็นค่าข้อมูลของชั้นซ่อน 512 จำนวน และจำนวนจริงที่เป็นค่าน้ำหนัก 4096 จำนวน ซึ่งคำนวนได้จากจำนวนของข้อมูลในชั้นซ่อนจำนวน 512 โหนดคูณกับจำนวนโหนดในชั้นผลลัพธ์ 8 โหนด

- เนื่องจากการประมวลผลแบบกริดอยู่บนพื้นฐานของระบบแบบกระจาย การเชื่อมต่อระหว่างโหนดต่าง ๆ หรือหน่วยประมวลผลต่าง ๆ ยังคงอาศัยการเชื่อมต่อผ่านระบบเครือข่ายที่มีการใช้งานทั่วไป ทำให้ในบางครั้งอาจเกิดการสูญเสียเวลาในช่วงของการรับส่งข้อมูลระหว่างเครื่องต่าง ๆ ภายในเครือข่าย

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

**2. สรุปผลการวิจัย**

จากผลการดำเนินงานวิจัย สรุปได้จากการเปรียบเทียบค่าเวลาที่ใช้ในการประมวลผลในขั้นตอนการเรียนรู้ของโปรแกรมประยุกต์ทั้ง 2 แบบ พบว่าการประมวลผลแบบกริดใช้เวลาในการประมวลผลน้อยกว่าการประมวลผลด้วยหน่วยประมวลผลเพียงตัวเดียว โดยถ้ามีหน่วยประมวลผลจำนวน 2 ตัว จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 1.41 ถึง 1.74 เท่า สำหรับหน่วยประมวลผลจำนวน 4 ตัว จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 1.53 ถึง 1.86 เท่า และหน่วยประมวลผลจำนวน 8 ตัว จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 1.59 ถึง 1.96 เท่า ทั้งนี้ขึ้นกับตัวอักษรที่นำมาใช้เพื่อการเรียนรู้ อีกทั้งยังพบว่าการประมวลผลแบบกริดจะไม่เกิดประโยชน์เท่าที่ควรถ้าตัวอักษรที่นำมาใช้มีขนาดเล็กเกินไป

ทั้งนี้หากนำผลการวิจัยนี้ไปประยุกต์ใช้กับระบบการรู้จำที่มีความซับซ้อนจะทำให้เห็นถึงประสิทธิภาพได้อย่างชัดเจนยิ่งขึ้น

**3. ข้อเสนอแนะเกี่ยวกับงานวิจัย**

งานวิจัยนี้ผู้วิจัยได้มุ่งเน้นเกี่ยวกับการประยุกต์ใช้องค์ความรู้ 2 ด้านให้สามารถทำงานร่วมกันได้ เพื่อเพิ่มประสิทธิภาพในการประมวลผลที่รวดเร็วขึ้นตามสมมติฐานที่ได้ตั้งไว้ในเบื้องต้น ด้วยเหตุนี้จึงเลือกการรู้จำตัวอักษรภาษาอังกฤษมาใช้ในงานวิจัย ซึ่งได้ผลเป็นน่าพอใจระดับหนึ่ง ทั้งนี้ผู้วิจัยมีข้อเสนอแนะเกี่ยวกับงานวิจัยนี้เพื่อให้เกิดการพัฒนา ปรับปรุง ในโอกาสต่อไป ดังนี้

3.1 เนื่องจากตัวอย่างที่ใช้ในการรู้จำเป็นภาพตัวอักษรภาษาอังกฤษที่ไม่มีความซับซ้อนมากนัก จึงไม่ได้มีการเลือกใช้ทฤษฎีขั้นสูงประกอบการวิจัย หากมีการพัฒนาโปรแกรมการรู้จำในอนาคตสามารถเพิ่มขั้นตอนต่าง ๆ ตามคำอภิปรายเบื้องต้น

3.2 ข้อมูลที่ใช้ในการทดลองมีขนาดเล็ก ทำให้ได้ค่าสำหรับการวัดประสิทธิภาพด้านความเร็วที่ไม่สูงมากนัก หากต้องการผลลัพธ์ที่แตกต่างมากขึ้นควรใช้ตัวอย่างที่มีขนาดใหญ่พอ

3.3 การวิจัยใช้ตัวอย่างที่มีคุณลักษณะแบบเดียวกันในหลาย ๆ ด้าน เช่น เป็นตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ และสีตัวอักษรและสีพื้นหลังมีความแตกต่างกันอย่างชัดเจน หากต้องการเพิ่มความหลากหลายของกลุ่มตัวอย่าง หรือต้องการแยกข้อมูลตัวอักษรกับพื้นหลังที่มีความแตกต่างกันไม่มากก็สามารถทำได้ โดยการเพิ่มขั้นตอนการวิเคราะห์ค่าสี หรือการประมวลผลภาพเบื้องต้นในรูปแบบต่าง ๆ ได้

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

**4. แนวทางการพัฒนางานวิจัยในอนาคต**

4.1 สามารถนำแนวคิดและผลลัพธ์ที่ได้ ไปประกอบการตัดสินใจในการพัฒนาโปรแกรมประยุกต์การรู้จำข้อมูลด้านอื่น ๆ ได้

4.2 สามารถนำความรู้เกี่ยวกับการใช้เครื่องมือ หรือเทคนิคต่าง ๆ ที่ได้รับระหว่างการวิจัยไปประยุกต์ใช้ในการพัฒนาโปรแกรมที่อยู่ภายใต้ข้อตกลง Open Source ได้

4.3 สามารถนำคำอภิปรายและอุปสรรคที่เกิดขึ้นระหว่างวิจัยไปปรับแก้ และพัฒนาโปรแกรม ประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริดที่สมบูรณ์ขึ้นได้

# บรรณานุกรม


## ภาษาไทย

กฤติกา วงศาวณิช และศุภชัย ตั้งบุญญะศิริ. “ระบบรู้จำป้ายทะเบียนรถยนต์.” เอกสารโครงงาน ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น, 2549

นราพงษ์ อภิรัตน์วรากุล และนเรศ เกื้อปัญญากูล. “หลักการการจดจำตัวอักษรไทย.” เอกสารสัมมนาภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น, 2548

## ภาษาต่างประเทศ

Fox,G,C., Furmanski,W., “Load Balancing Loosely Synchronous Problems with a Neural Network.”

**California Institute of Technology**, 1988

Holland, J. H., “Adaptation in Natural and Artificial Systems”. **The University of Michigan Press, Ann Arbor,** 1975.

Harwood,A., Senyard,A., and Minh,T., “Artificial Neural Network Development Using P2P Networks”, **University of Melbourne**, 2004.

LiMin, Fu. Neural Network in Computer Intelligence. Singapore: McGraw Hill, 1994.

Levi B. Larkey, “Distributed Asynchronous Neural Encoding of Compositional Representations”, **The University of Texas at Austin**,2002.

Michell, T. M. Machine Learning. Singapore: McGraw Hill, 1997.

Muller,B., Reinhardt,J., and M.T.Strickland. Neural Network An Introduction. New York : Springer-Verlag Berlin Heidelberg, 1995.

Ribert, Ennaji and Lecourtier. “Building and Evaluation of a Distributed Neural Classifier”. **P.S.I. Faculté des Sciences, Université de Rouen**,1999.

Riccardo Poli, “Discovery of Symbolic, Neuro-Symbolic and Neural Networks with Parallel Distributed Genetic Programming”, **The University of Birmingham**, 1999.

Schikuta, E. Weishaupl, T**. “**N2Grid: neural networks in the grid**”, Neural Networks**, 2004.


มหาวิทยาลัยขอนแก่น ลิขสิทธิ์

Sae-Tang,S., Methaste,I., “Thai Online Handwritten Character Recognition Using Windowing Backpropagation Neural Networks”. **National Electronics and Computer Technology Center**, 2000**.**

Thai grid. Grid Architecture for Computational Economy [Online], accessed 15 April 2008. Available from http://www.thaigrid.or.th/html/about/contents.php

Thai grid. Grid Technology and Application [Online], accessed 15 April 2008. Available from http://rswg.thaigrid.or.th/book/export/html/75

Udo Seiffert, “Artificial Neural Networks on Massively Parallel Computer Hardware”. **University of Magdeburg, Germany,**2004

Vladimir Silva. Grid Computing For Developers. USA: Charles River Media, 2005

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# ภาคผนวก

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# ภาคผนวก ก

## รายละเอียดการวัดประสิทธิภาพการประมวลผลการเรียนรู้ของแต่ละตัวอักษร

มหาวิทยาลัยศิลปากร ลิขสิทธิ์

**รายละเอียดการวัดประสิทธิภาพการประมวลผลการเรียนรู้ตัวอักษร**

1. การเรียนรู้ตัวอักษร A

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 4818
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 66
- ความสูงตัวอักษร : 73

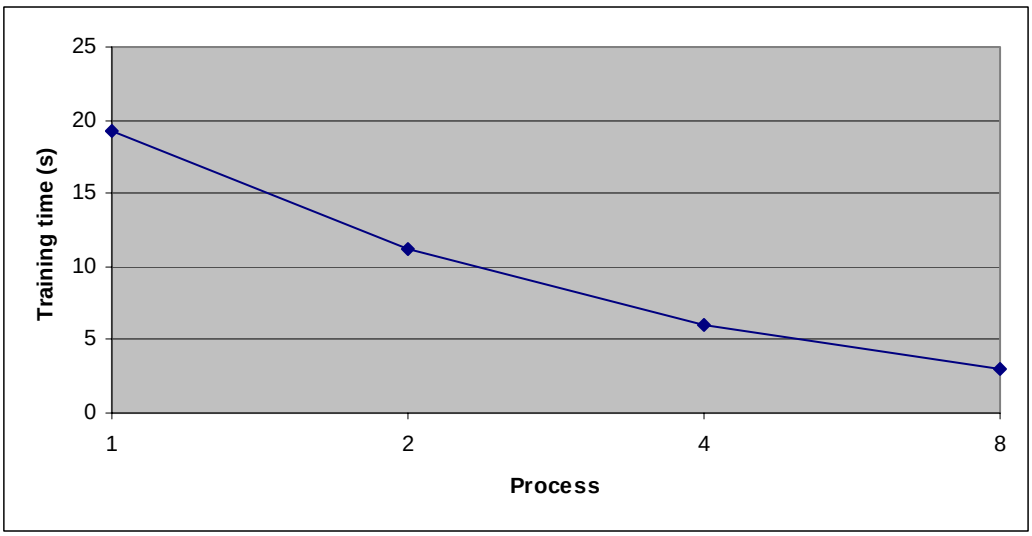
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 4

ตารางที่ 4 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร A

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 19.198 | 11.257 | 6.023 | 3.031 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.71 | 3.19 | 6.33 |
| ความถูกต้องในการรู้จำ | 90.5% | | | |

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

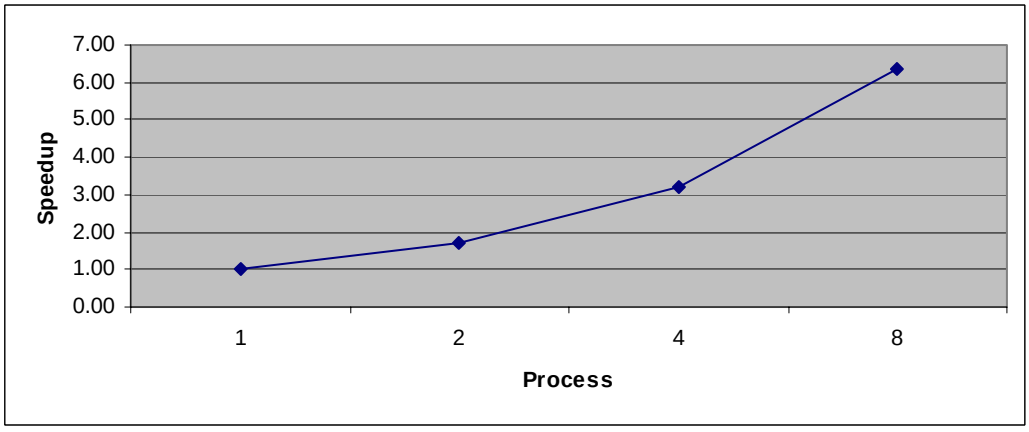
จากตารางที่ 4 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร A ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร A ดังแสดงในแผนภูมิที่ 1 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร A ดังแสดงในแผนภูมิที่ 2




แผนภูมิที่ 1 แสดงเวลาในการเรียนรู้ตัวอักษร A

จากแผนภูมิที่ 1 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซส จะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 2 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร A

จากแผนภูมิที่ 2 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซส จะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

2. การเรียนรู้ตัวอักษร B

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 3869
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 53
- ความสูงตัวอักษร : 73

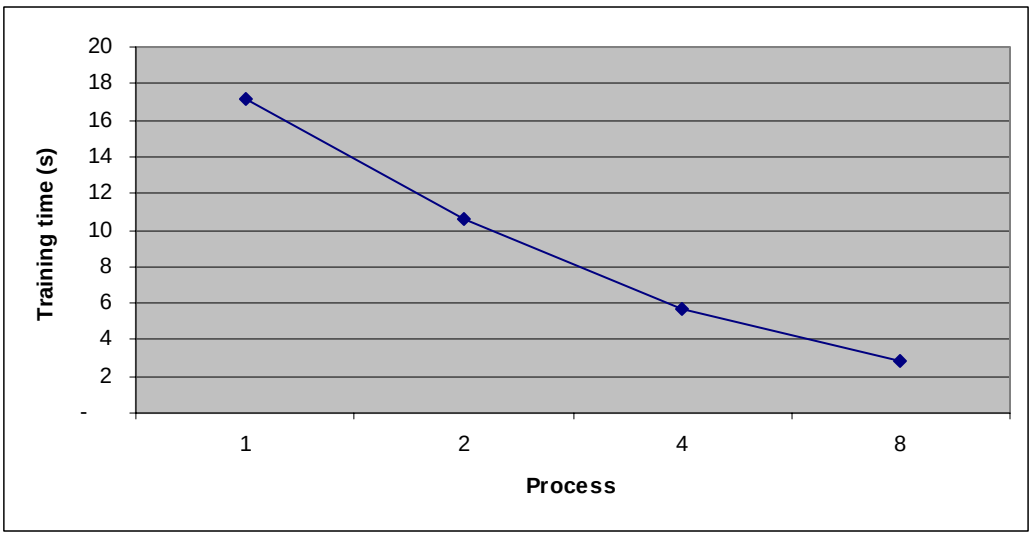
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 5

ตารางที่ 5 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร B

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.168 | 10.638 | 5.705 | 2.796 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.61 | 3.01 | 6.14 |
| ความถูกต้องในการรู้จำ | 92.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

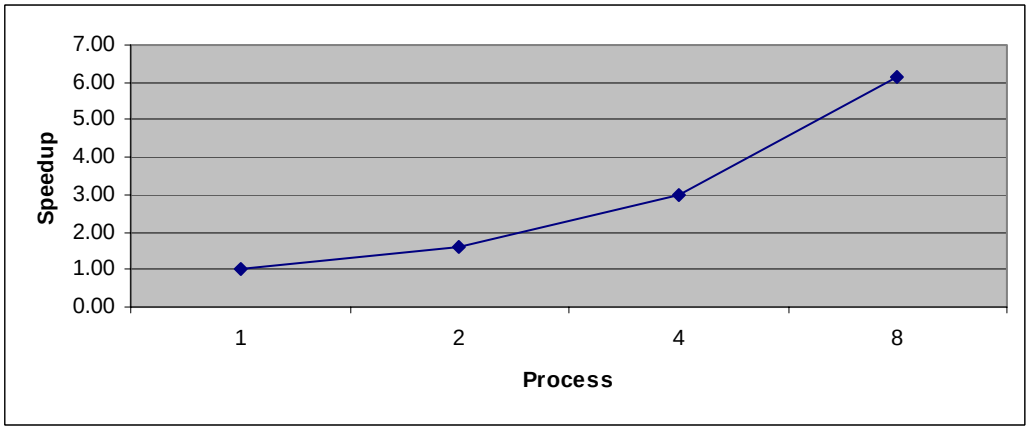
จากตารางที่ 5 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร B ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร B ดังแสดงในแผนภูมิที่ 3 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร B ดังแสดงในแผนภูมิที่ 4




แผนภูมิที่ 3 แสดงเวลาในการเรียนรู้ตัวอักษร B

จากแผนภูมิที่ 3 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซส จะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 4 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร B

จากแผนภูมิที่ 4 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซส จะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

3. การเรียนรู้ตัวอักษร C

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 4621
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 60
- ความสูงตัวอักษร : 77

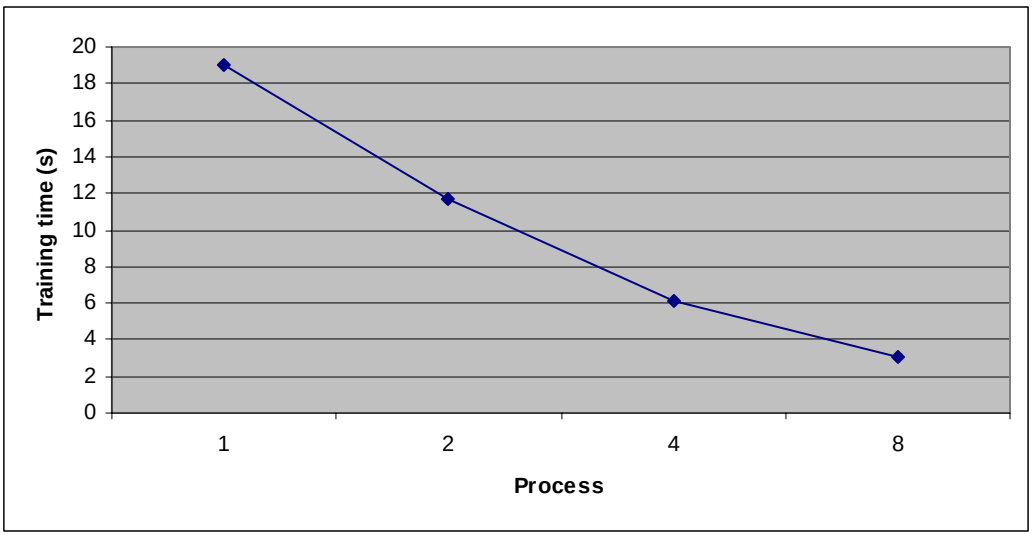
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 6

ตารางที่ 6 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร C

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.998 | 11.684 | 6.152 | 3.091 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.63 | 3.09 | 6.15 |
| ความถูกต้องในการรู้จำ | 89.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

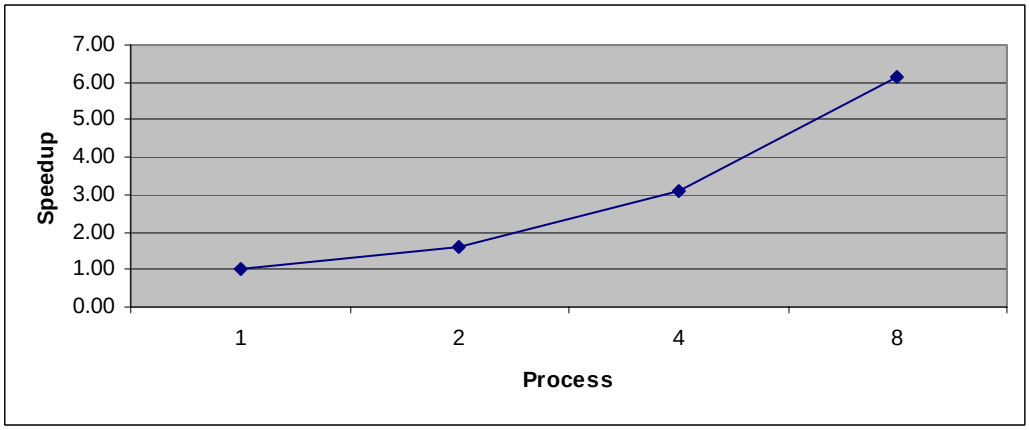
จากตารางที่ 6 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร C ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร C ดังแสดงในแผนภูมิที่ 5 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร C ดังแสดงในแผนภูมิที่ 6




แผนภูมิที่ 5 แสดงเวลาในการเรียนรู้ตัวอักษร C

จากแผนภูมิที่ 5 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 6 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร C

จากแผนภูมิที่ 6 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

4. การเรียนรู้ตัวอักษร D

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 4599
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 63
- ความสูงตัวอักษร : 73

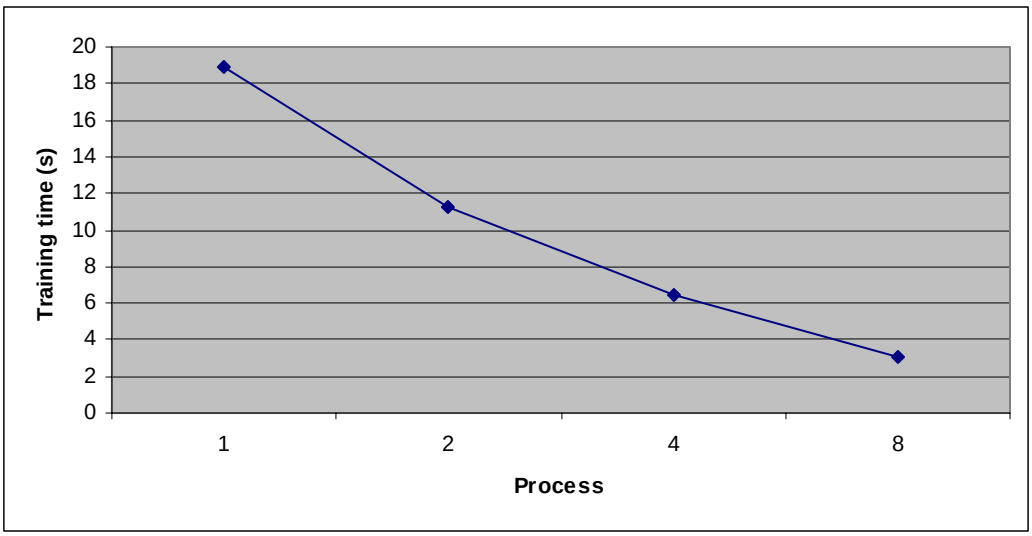
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 7

ตารางที่ 7 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร D

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.953 | 11.295 | 6.417 | 3.104 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.68 | 2.95 | 6.11 |
| ความถูกต้องในการรู้จำ | 94.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

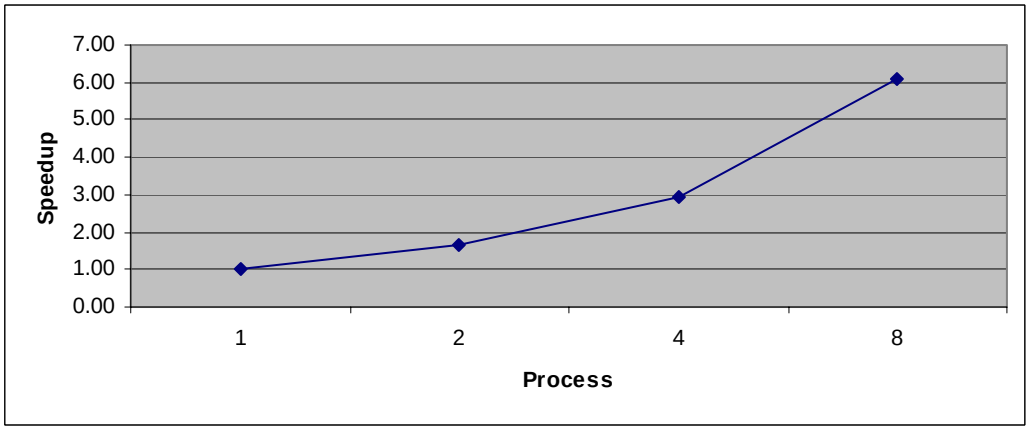
จากตารางที่ 7 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร D ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร D ดังแสดงในแผนภูมิที่ 7 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร D ดังแสดงในแผนภูมิที่ 8




แผนภูมิที่ 7 แสดงเวลาในการเรียนรู้ตัวอักษร D

จากแผนภูมิที่ 7 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 8 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร D

จากแผนภูมิที่ 8 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

5. การเรียนรู้ตัวอักษร E

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 3504
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 48
- ความสูงตัวอักษร : 73

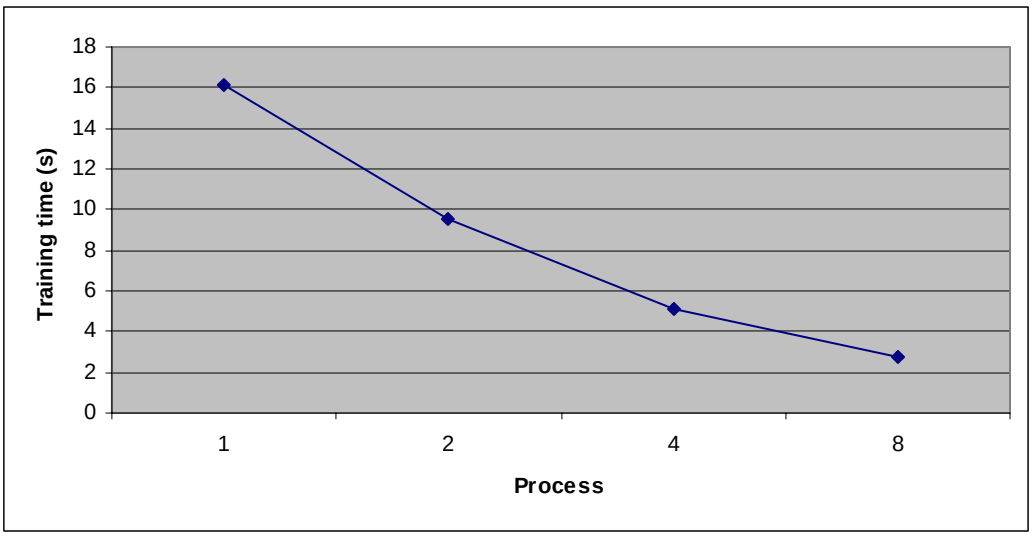
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 8

ตารางที่ 8 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร E

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 16.11 | 9.524 | 5.084 | 2.794 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.69 | 3.17 | 5.77 |
| ความถูกต้องในการรู้จำ | 95.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

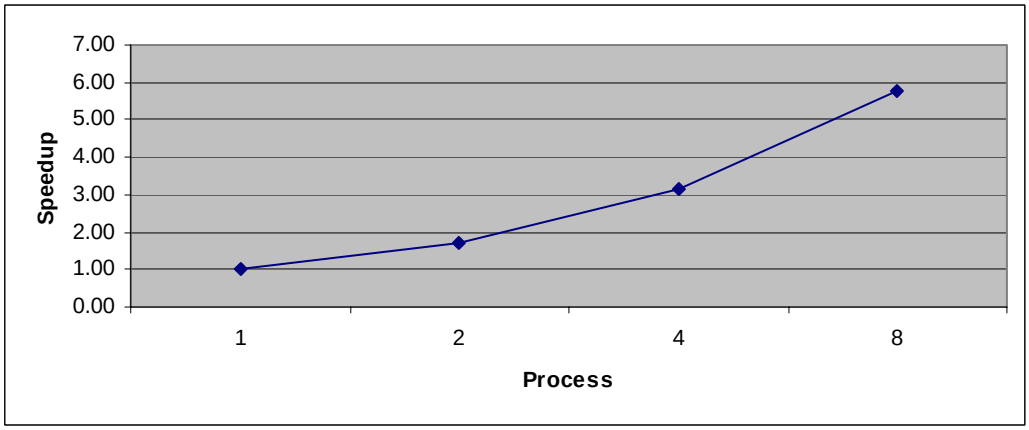
จากตารางที่ 8 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร E ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร E ดังแสดงในแผนภูมิที่ 9 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร E ดังแสดงในแผนภูมิที่ 10




แผนภูมิที่ 9 แสดงเวลาในการเรียนรู้ตัวอักษร E

จากแผนภูมิที่ 9 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 10 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร E

จากแผนภูมิที่ 10 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

6. การเรียนรู้ตัวอักษร F

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 3139
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 43
- ความสูงตัวอักษร : 73

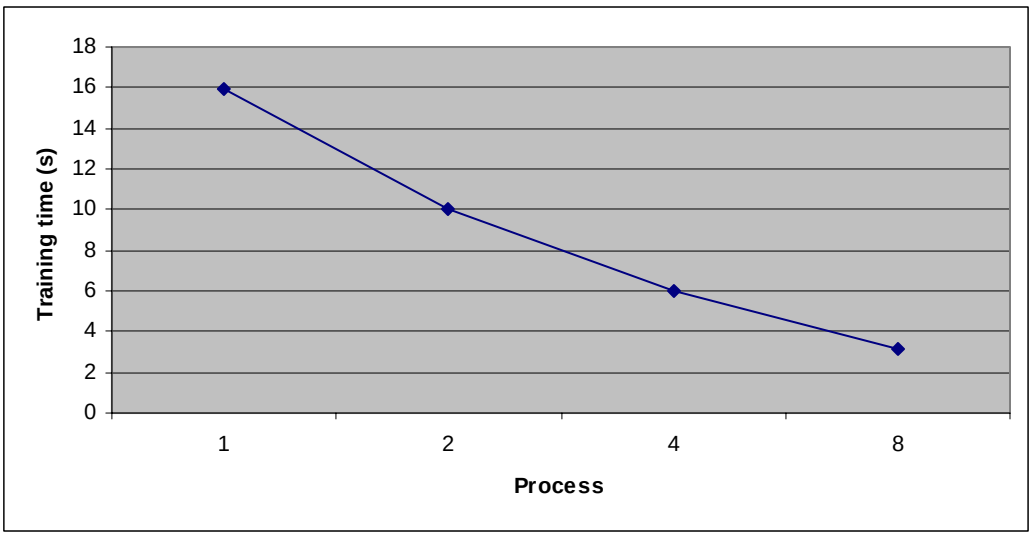
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 9

ตารางที่ 9 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร F

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 15.977 | 10.013 | 6.021 | 3.172 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.60 | 2.65 | 5.04 |
| ความถูกต้องในการรู้จำ | 96.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

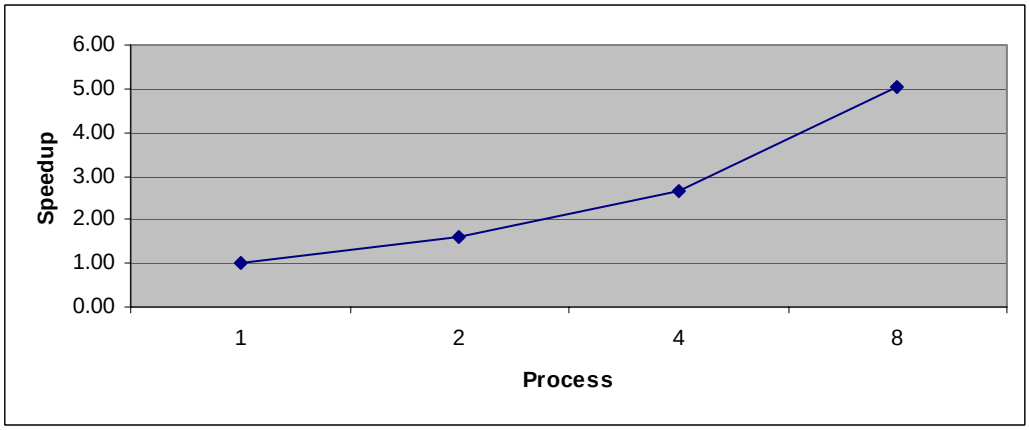
จากตารางที่ 9 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร F ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร F ดังแสดงในแผนภูมิที่ 11 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร F ดังแสดงในแผนภูมิที่ 12




แผนภูมิที่ 11 แสดงเวลาในการเรียนรู้ตัวอักษร F

จากแผนภูมิที่ 11 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 12 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร F

จากแผนภูมิที่ 12 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

7. การเรียนรู้ตัวอักษร G

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 5005
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 65
- ความสูงตัวอักษร : 77

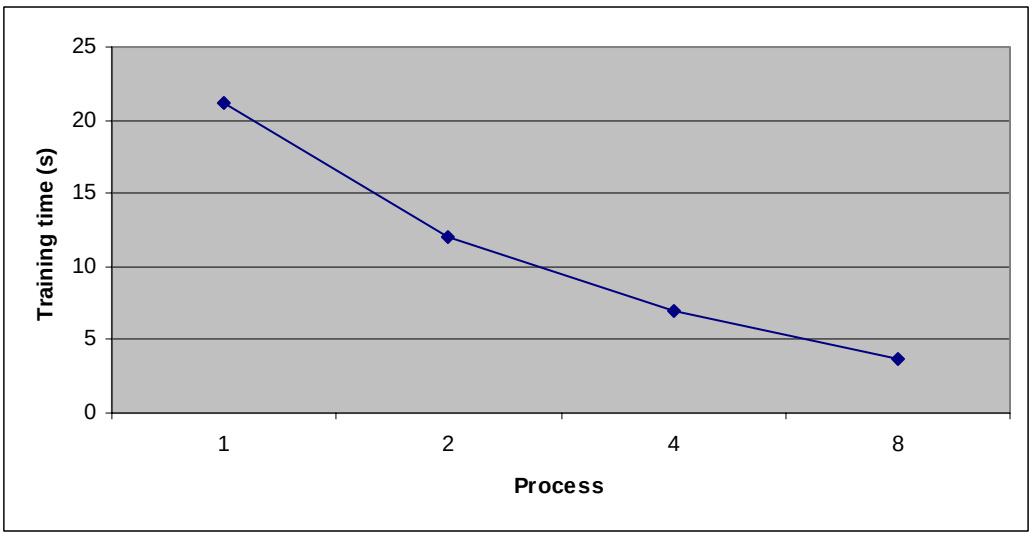
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 10

ตารางที่ 10 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร G

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 21.121 | 12.015 | 7.019 | 3.674 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.76 | 3.01 | 5.75 |
| ความถูกต้องในการรู้จำ | 92.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี

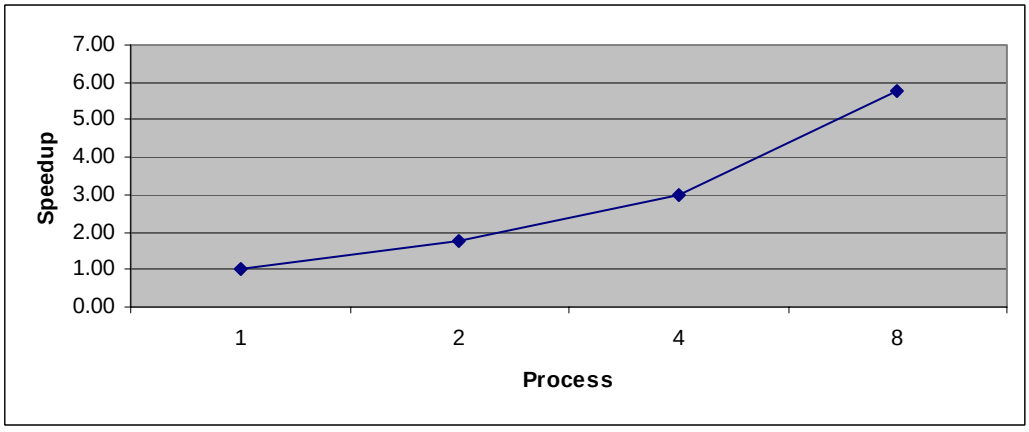
จากตารางที่ 10 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร G ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร G ดังแสดงในแผนภูมิที่ 13 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร G ดังแสดงในแผนภูมิที่ 14




แผนภูมิที่ 13 แสดงเวลาในการเรียนรู้ตัวอักษร G

จากแผนภูมิที่ 13 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 14 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร G

จากแผนภูมิที่ 14 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

8. การเรียนรู้ตัวอักษร H

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 4088
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 56
- ความสูงตัวอักษร : 73

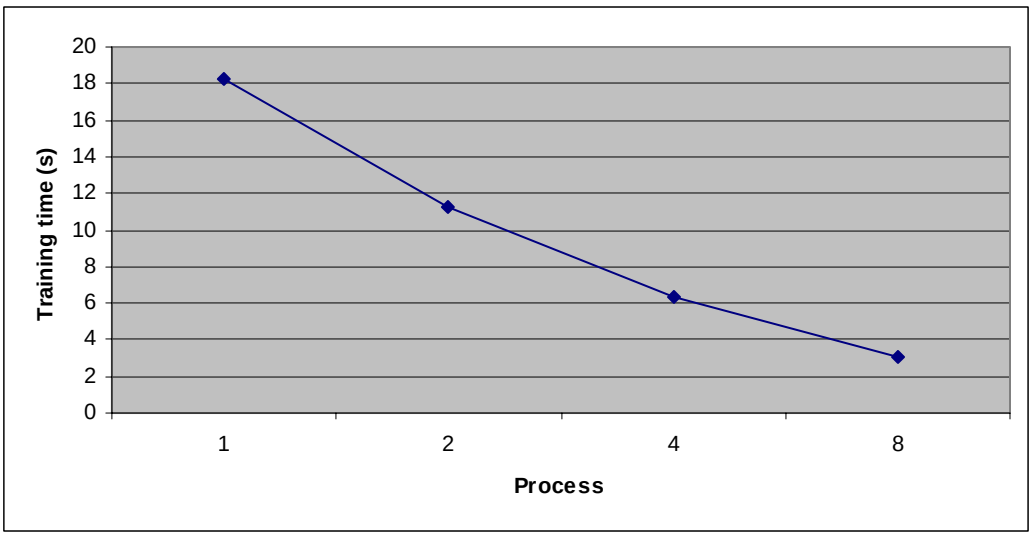
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 11

ตารางที่ 11 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร H

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.236 | 11.23 | 6.317 | 3.106 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.62 | 2.89 | 5.87 |
| ความถูกต้องในการรู้จำ | 96.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

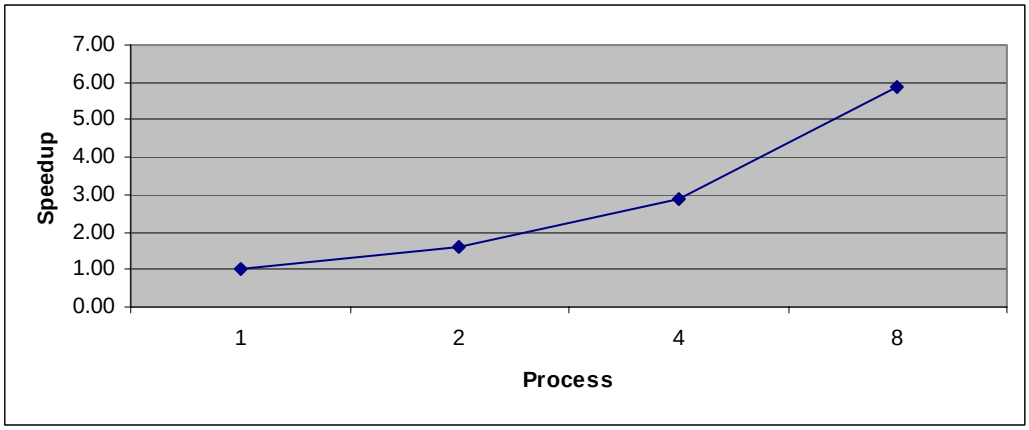
จากตารางที่ 11 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร H ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร H ดังแสดงในแผนภูมิที่ 15 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร H ดังแสดงในแผนภูมิที่ 16




แผนภูมิที่ 15 แสดงเวลาในการเรียนรู้ตัวอักษร H

จากแผนภูมิที่ 15 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 16 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร H

จากแผนภูมิที่ 16 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

9. การเรียนรู้ตัวอักษร I

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node) : 803
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 11
- ความสูงตัวอักษร : 73

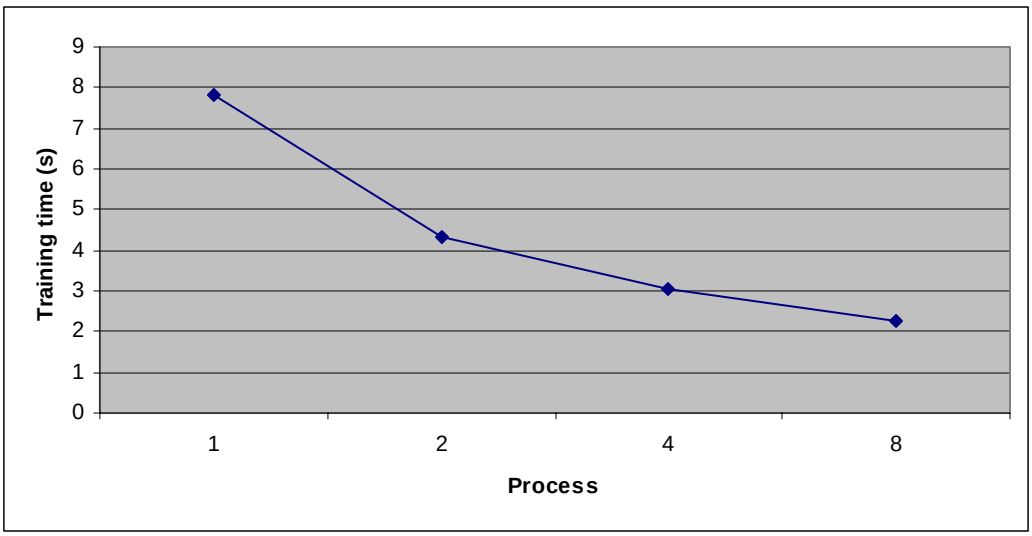
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 12

ตารางที่ 12 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร I

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 7.83 | 4.31 | 3.069 | 2.241 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.82 | 2.55 | 3.49 |
| ความถูกต้องในการรู้จำ | 98.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี

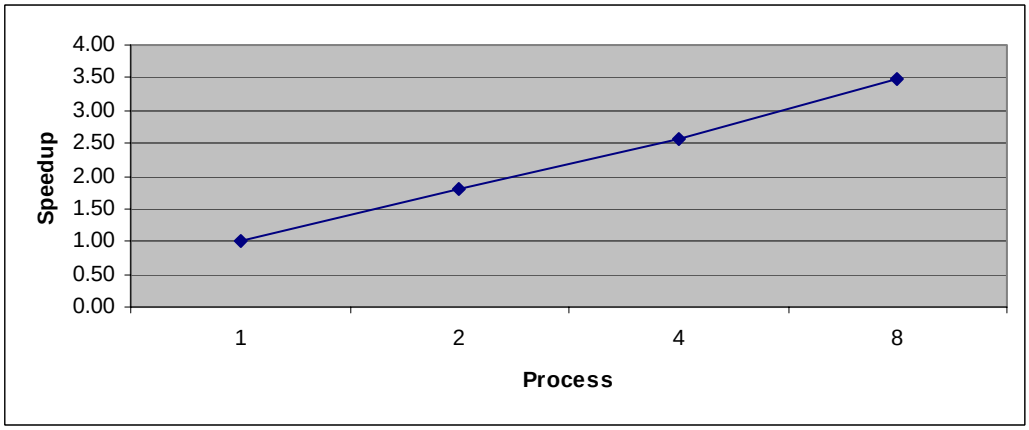
จากตารางที่ 12 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร I ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร I ดังแสดงในแผนภูมิที่ 17 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร I ดังแสดงในแผนภูมิที่ 18




แผนภูมิที่ 17 แสดงเวลาในการเรียนรู้ตัวอักษร I

จากแผนภูมิที่ 17 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 18 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร I

จากแผนภูมิที่ 18 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

10. การเรียนรู้ตัวอักษร J

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 2418
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 26
- ความสูงตัวอักษร : 93

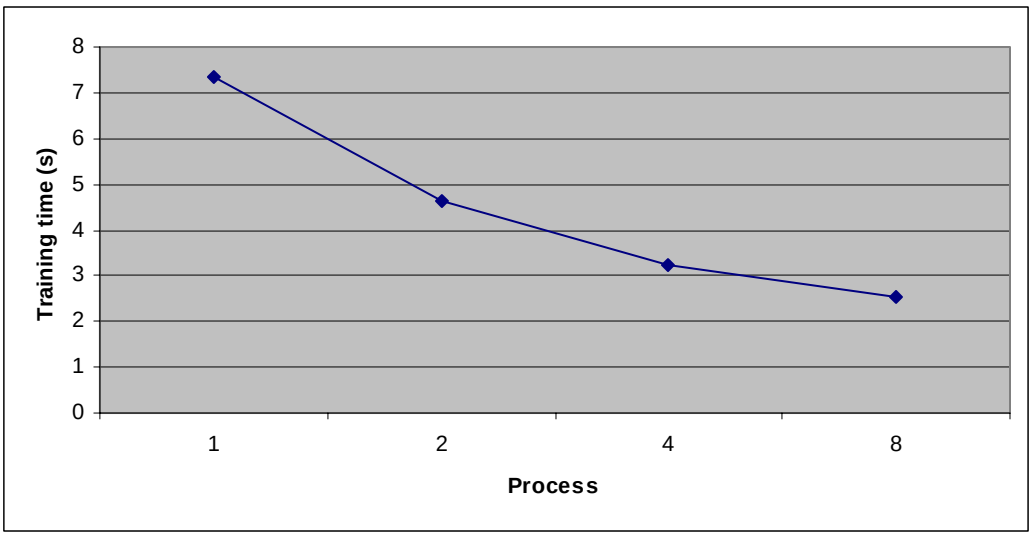
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 13

ตารางที่ 13 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร J

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 7.323 | 4.613 | 3.223 | 2.523 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.59 | 2.27 | 2.90 |
| ความถูกต้องในการรู้จำ | 96.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

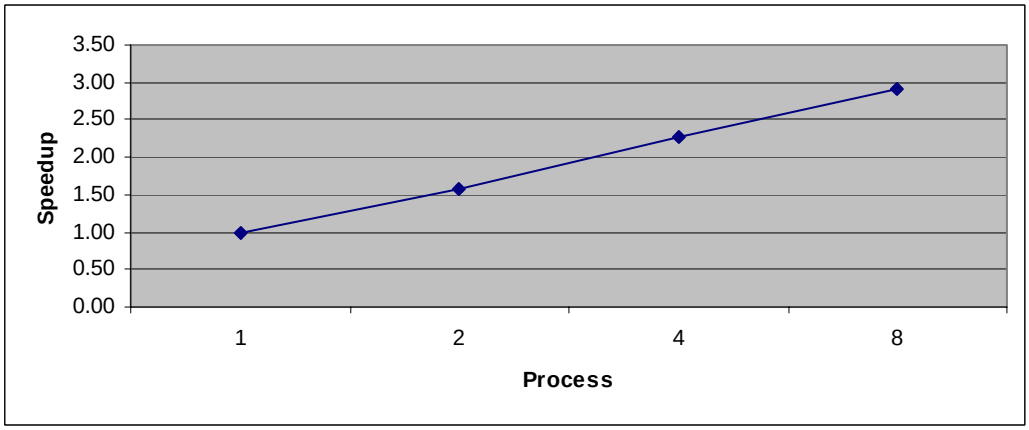
จากตารางที่ 13 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร J ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร J ดังแสดงในแผนภูมิที่ 19 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร J ดังแสดงในแผนภูมิที่ 20




แผนภูมิที่ 19 แสดงเวลาในการเรียนรู้ตัวอักษร J

จากแผนภูมิที่ 19 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 20 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร J

จากแผนภูมิที่ 20 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

11. การเรียนรู้ตัวอักษร K

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4307
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 59
- ความสูงตัวอักษร : 73

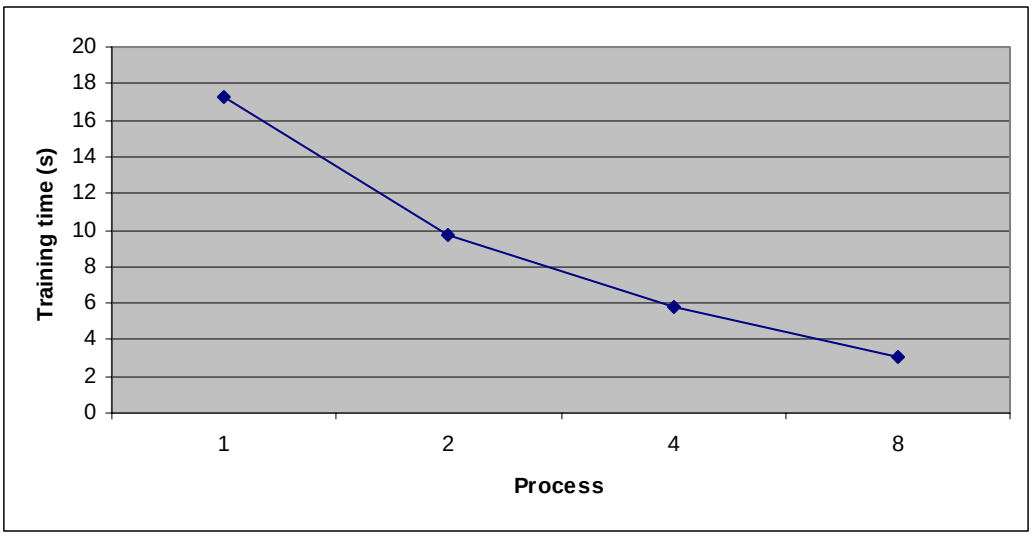
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 14

ตารางที่ 14 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร K

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.32 | 9.769 | 5.84 | 3.014 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.77 | 2.97 | 5.75 |
| ความถูกต้องในการรู้จำ | 89.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

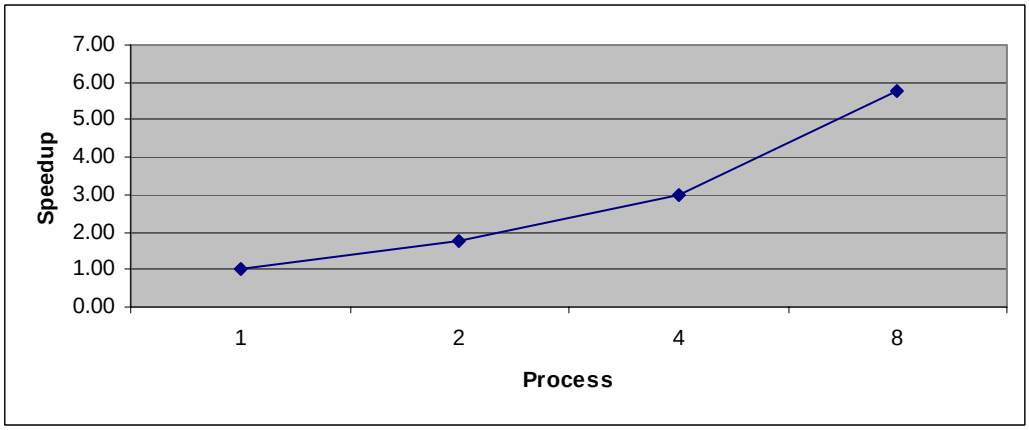
จากตารางที่ 14 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร K ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร K ดังแสดงในแผนภูมิที่ 21 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร K ดังแสดงในแผนภูมิที่ 22




แผนภูมิที่ 21 แสดงเวลาในการเรียนรู้ตัวอักษร K

จากแผนภูมิที่ 21 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 22 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร K

จากแผนภูมิที่ 22 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

12. การเรียนรู้ตัวอักษร L

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 3431
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 47
- ความสูงตัวอักษร : 73

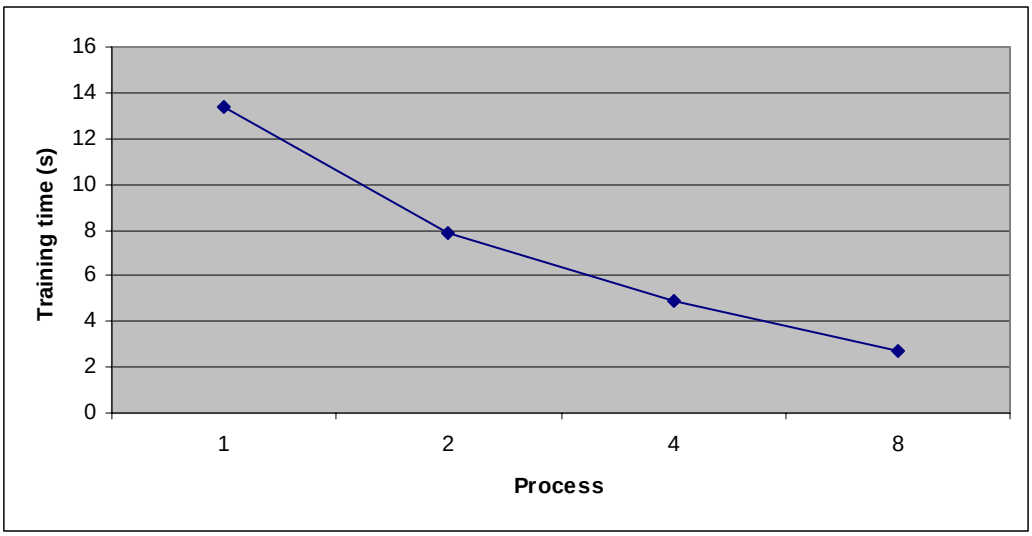
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 15

ตารางที่ 15 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร L

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 13.336 | 7.84 | 4.891 | 2.711 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.70 | 2.73 | 4.92 |
| ความถูกต้องในการรู้จำ | 97.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

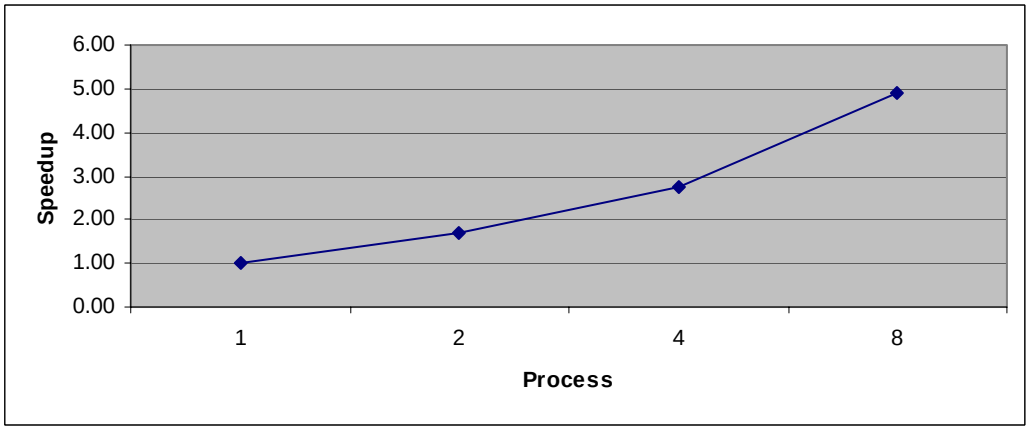
จากตารางที่ 15 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร L ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร L ดังแสดงในแผนภูมิที่ 23 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร L ดังแสดงในแผนภูมิที่ 24




แผนภูมิที่ 23 แสดงเวลาในการเรียนรู้ตัวอักษร L

จากแผนภูมิที่ 23 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 24 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร L

จากแผนภูมิที่ 24 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

13. การเรียนรู้ตัวอักษร M

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4964
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 68
- ความสูงตัวอักษร : 73

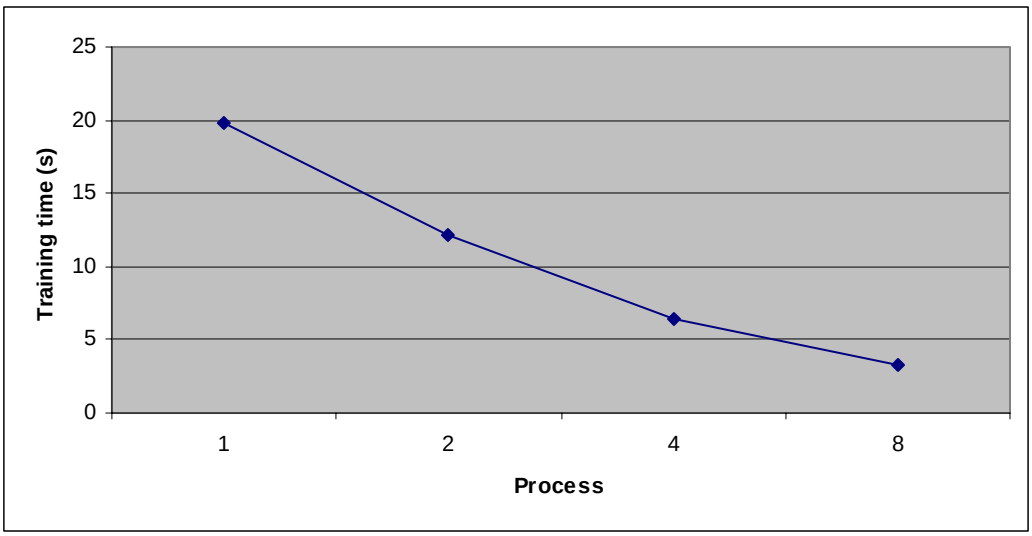
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 16

ตารางที่ 16 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร M

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 19.81 | 12.173 | 6.369 | 3.318 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.63 | 3.11 | 5.97 |
| ความถูกต้องในการรู้จำ | 92.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

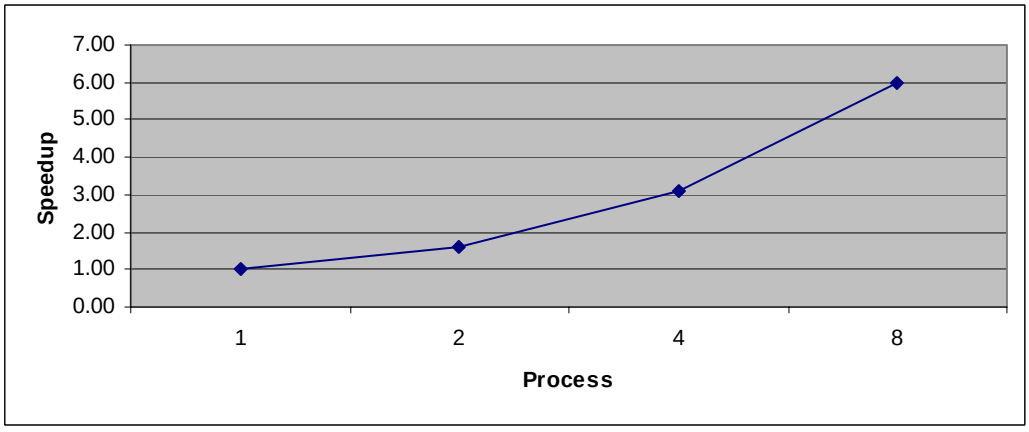
จากตารางที่ 16 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร M ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร M ดังแสดงในแผนภูมิที่ 25 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร M ดังแสดงในแผนภูมิที่ 26




แผนภูมิที่ 25 แสดงเวลาในการเรียนรู้ตัวอักษร M

จากแผนภูมิที่ 25 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 26 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร M

จากแผนภูมิที่ 26 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

14. การเรียนรู้ตัวอักษร N

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4088
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 56
- ความสูงตัวอักษร : 73

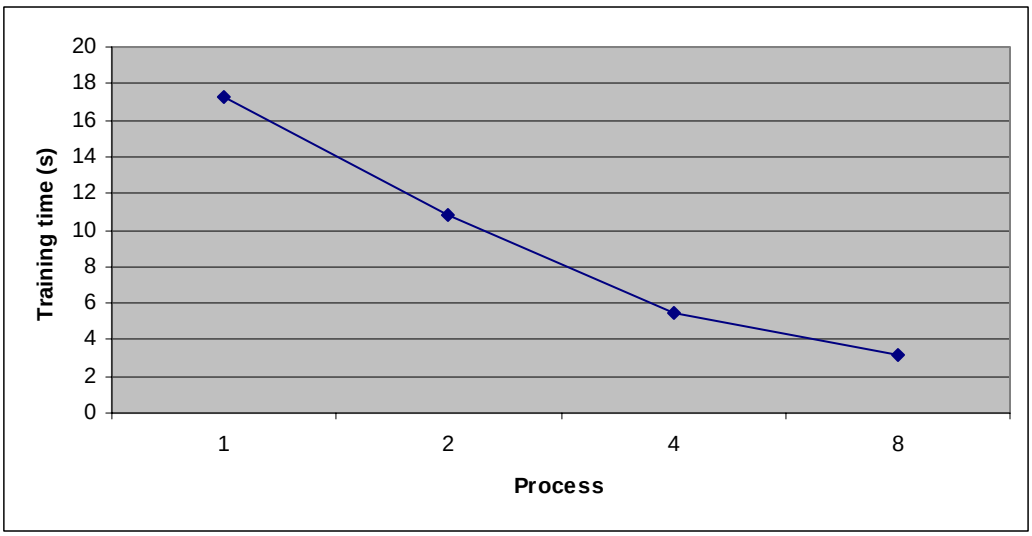
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 17

ตารางที่ 17 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร N

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.251 | 10.873 | 5.506 | 3.17 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.59 | 3.13 | 5.44 |
| ความถูกต้องในการรู้จำ | 93.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

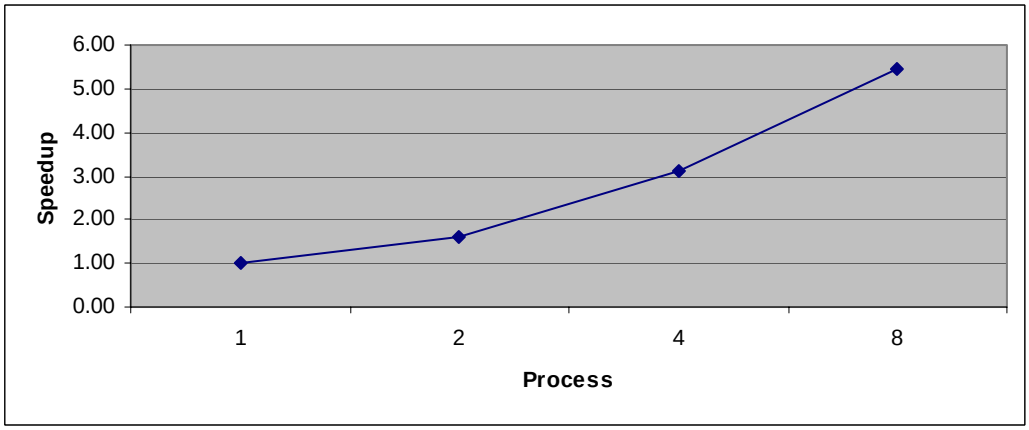
จากตารางที่ 17 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร N ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร N ดังแสดงในแผนภูมิที่ 27 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร N ดังแสดงในแผนภูมิที่ 28




แผนภูมิที่ 27 แสดงเวลาในการเรียนรู้ตัวอักษร N

จากแผนภูมิที่ 27 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 28 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร N

จากแผนภูมิที่ 28 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

15. การเรียนรู้ตัวอักษร O

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 5313
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 69
- ความสูงตัวอักษร : 77

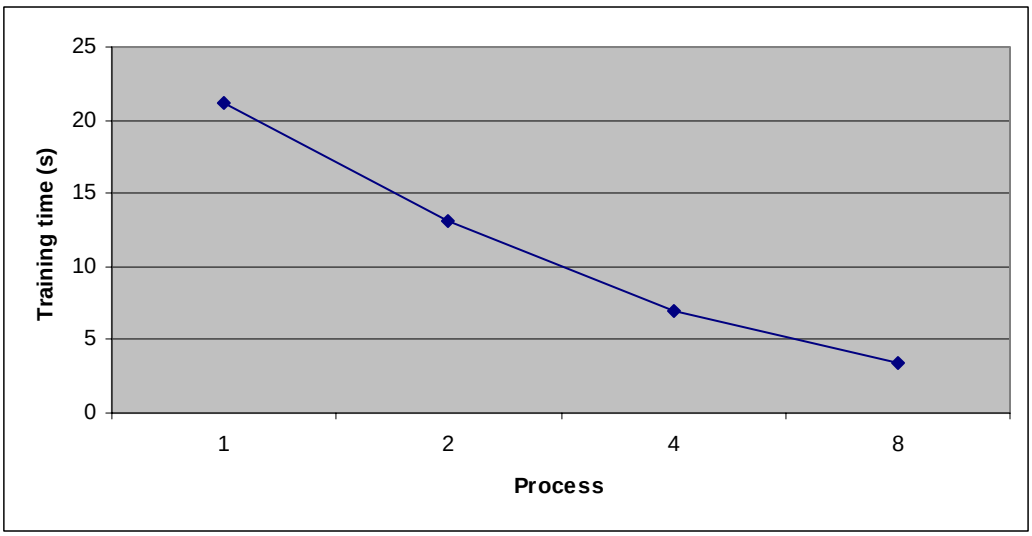
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 18

ตารางที่ 18 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร O

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 21.2 | 13.091 | 7.024 | 3.445 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.62 | 3.02 | 6.15 |
| ความถูกต้องในการรู้จำ | 95.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

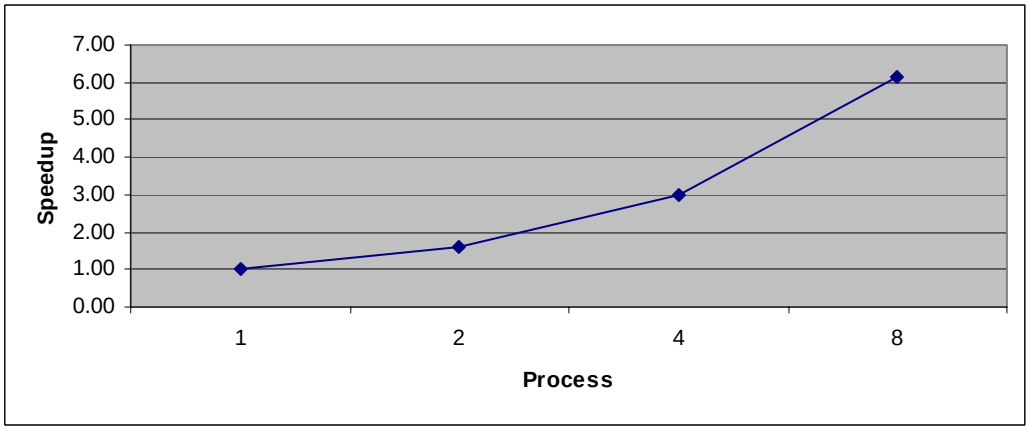
จากตารางที่ 18 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร O ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร O ดังแสดงในแผนภูมิที่ 29 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร O ดังแสดงในแผนภูมิที่ 30




แผนภูมิที่ 29 แสดงเวลาในการเรียนรู้ตัวอักษร O

จากแผนภูมิที่ 29 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 30 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร O

จากแผนภูมิที่ 30 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

16. การเรียนรู้ตัวอักษร P

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 3504
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 48
- ความสูงตัวอักษร : 73

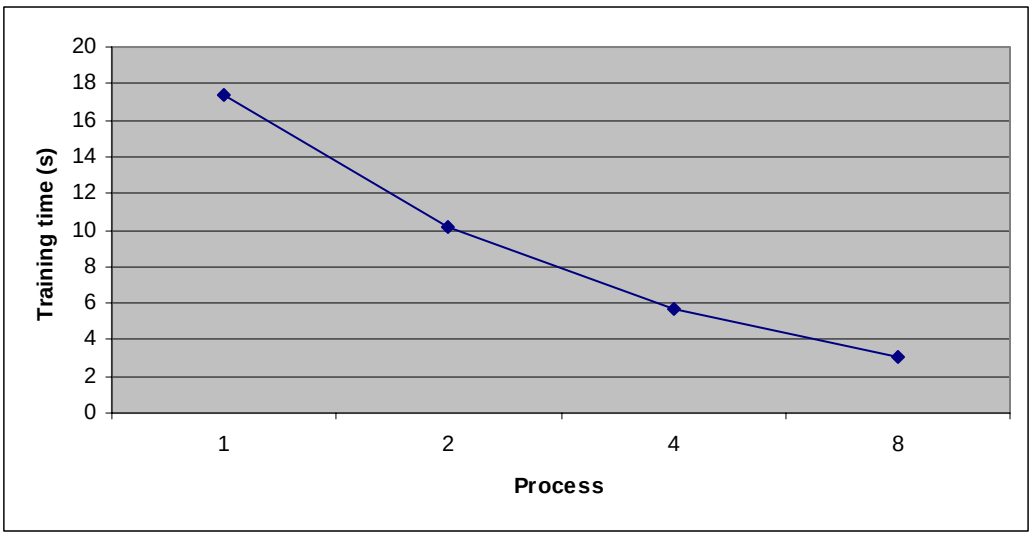
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 19

ตารางที่ 19 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร P

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.376 | 10.176 | 5.676 | 3.026 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.71 | 3.06 | 5.74 |
| ความถูกต้องในการรู้จำ | 92.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

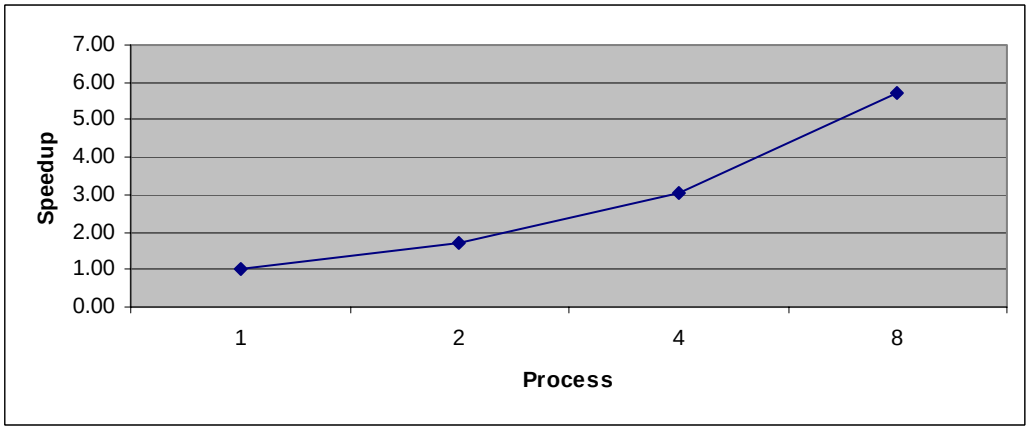
จากตารางที่ 19 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร P ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร P ดังแสดงในแผนภูมิที่ 31 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร P ดังแสดงในแผนภูมิที่ 32




แผนภูมิที่ 31 แสดงเวลาในการเรียนรู้ตัวอักษร P

จากแผนภูมิที่ 31 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 32 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร P

จากแผนภูมิที่ 32 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

17. การเรียนรู้ตัวอักษร Q

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 6072
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 69
- ความสูงตัวอักษร : 88

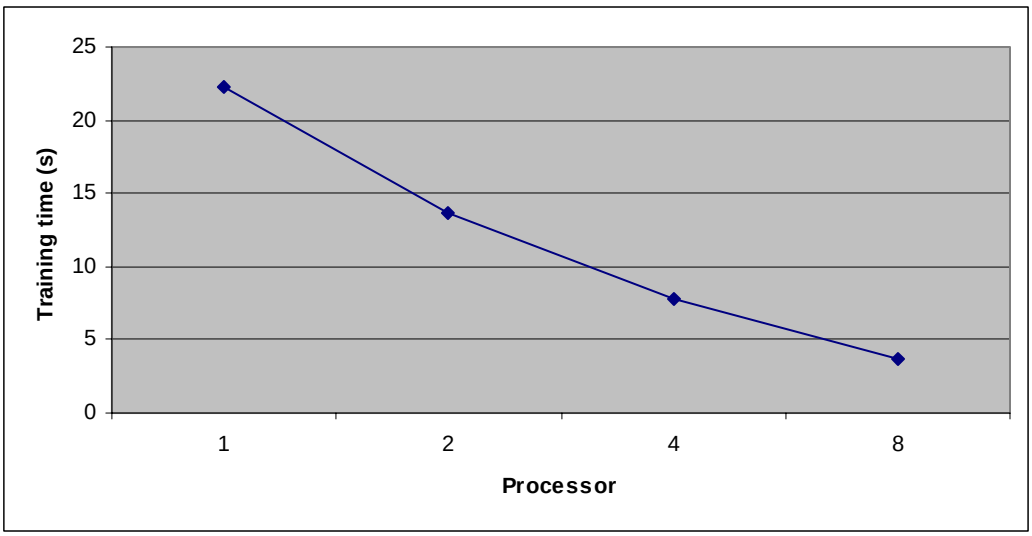
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 20

ตารางที่ 20 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Q

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 22.305 | 13.601 | 7.75 | 3.721 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.64 | 2.88 | 5.99 |
| ความถูกต้องในการรู้จำ | 94.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

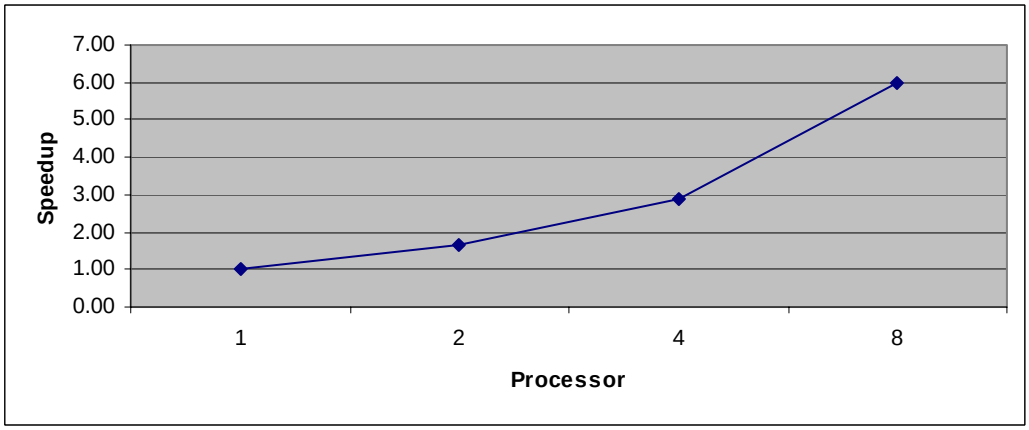
จากตารางที่ 20 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร Q ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร Q ดังแสดงในแผนภูมิที่ 33 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร Q ดังแสดงในแผนภูมิที่ 34




แผนภูมิที่ 33 แสดงเวลาในการเรียนรู้ตัวอักษร Q

จากแผนภูมิที่ 33 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 34 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Q

จากแผนภูมิที่ 34 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

18. การเรียนรู้ตัวอักษร R

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4161
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 57
- ความสูงตัวอักษร : 73

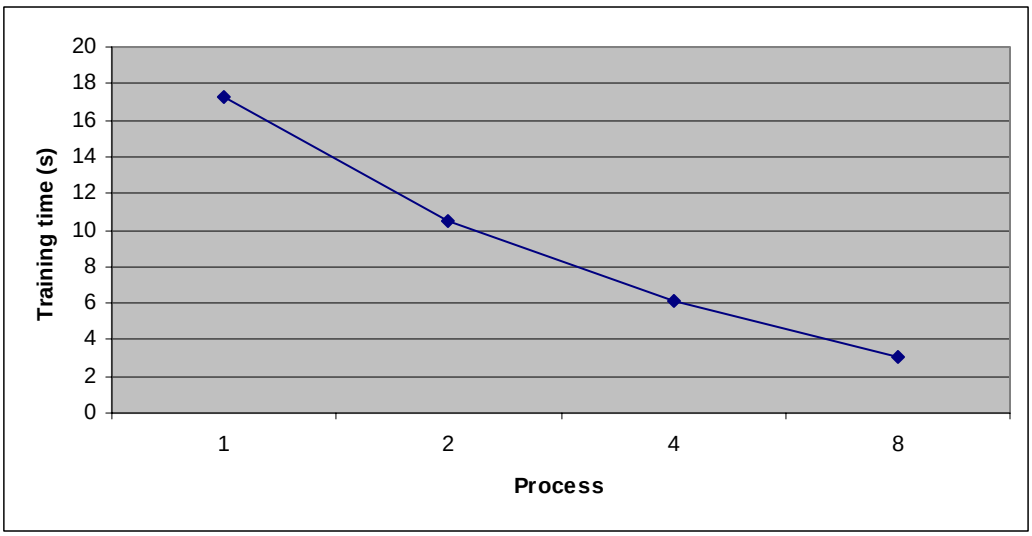
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 21

ตารางที่ 21 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร R

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.278 | 10.53 | 6.103 | 3.108 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.64 | 2.83 | 5.56 |
| ความถูกต้องในการรู้จำ | 93% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

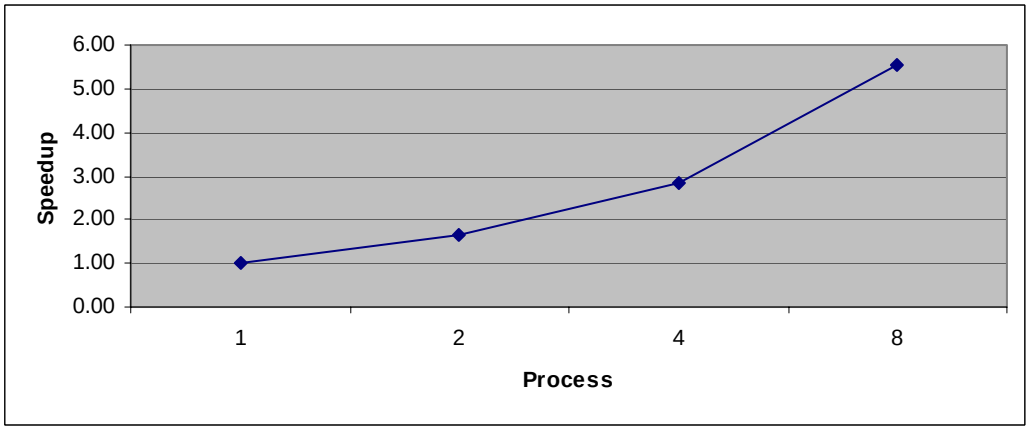
จากตารางที่ 21 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร R ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร R ดังแสดงในแผนภูมิที่ 35 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร R ดังแสดงในแผนภูมิที่ 36




แผนภูมิที่ 35 แสดงเวลาในการเรียนรู้ตัวอักษร R

จากแผนภูมิที่ 35 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 36 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร R

จากแผนภูมิที่ 36 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

19. การเรียนรู้ตัวอักษร S

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4004
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 52
- ความสูงตัวอักษร : 77

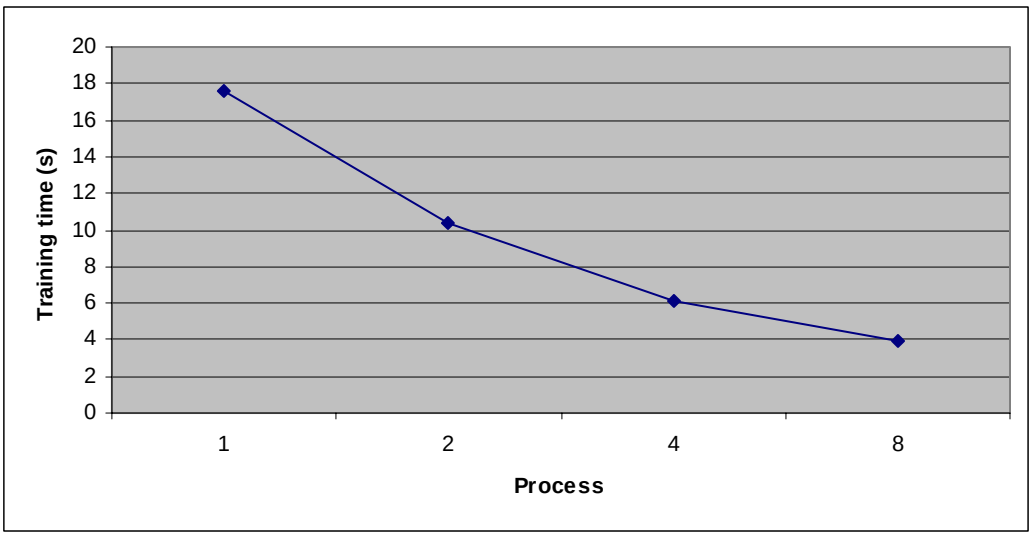
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 22

ตารางที่ 22 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร S

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.578 | 10.395 | 6.13 | 3.98 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.69 | 2.87 | 4.42 |
| ความถูกต้องในการรู้จำ | 86% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

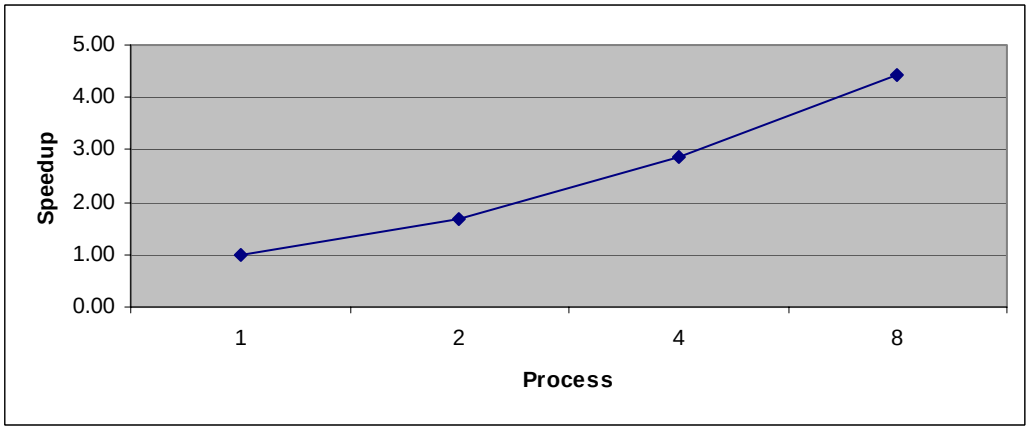
จากตารางที่ 22 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร S ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร S ดังแสดงในแผนภูมิที่ 37 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร S ดังแสดงในแผนภูมิที่ 38




แผนภูมิที่ 37 แสดงเวลาในการเรียนรู้ตัวอักษร S

จากแผนภูมิที่ 37 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 38 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร S

จากแผนภูมิที่ 38 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

20. การเรียนรู้ตัวอักษร T

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4599
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 63
- ความสูงตัวอักษร : 73

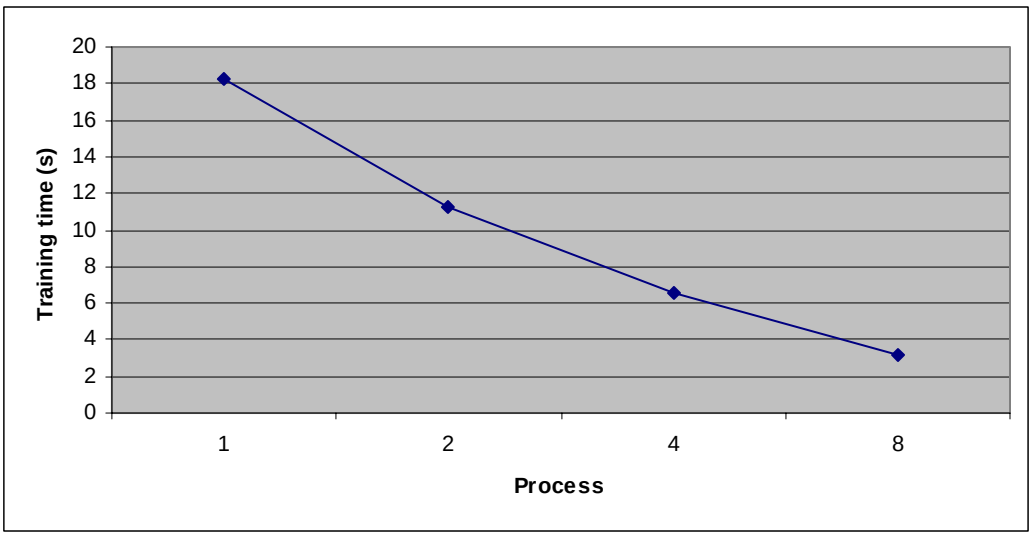
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 23

ตารางที่ 23 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร T

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.241 | 11.241 | 6.531 | 3.137 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.62 | 2.79 | 5.81 |
| ความถูกต้องในการรู้จำ | 93.5% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

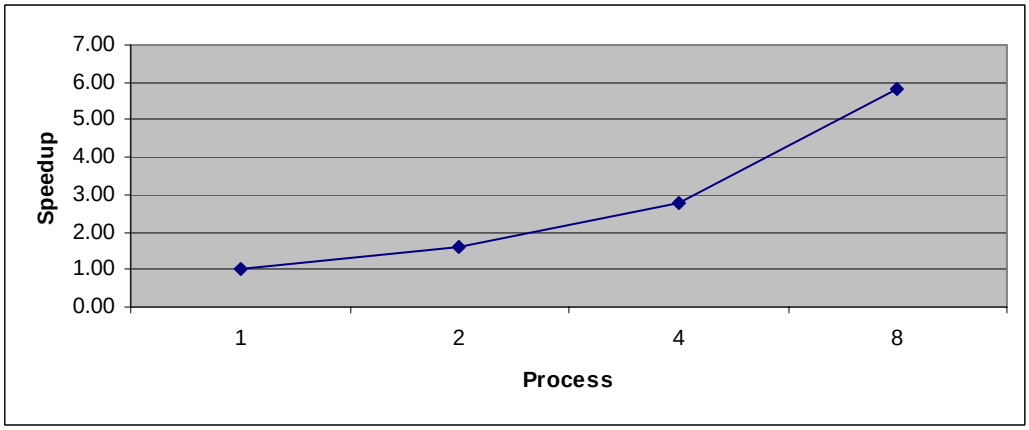
จากตารางที่ 23 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร T ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร T ดังแสดงในแผนภูมิที่ 39 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร T ดังแสดงในแผนภูมิที่ 40




แผนภูมิที่ 39 แสดงเวลาในการเรียนรู้ตัวอักษร T

จากแผนภูมิที่ 39 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 40 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร T

จากแผนภูมิที่ 40 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

21. การเรียนรู้ตัวอักษร U

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4275
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 57
- ความสูงตัวอักษร : 75

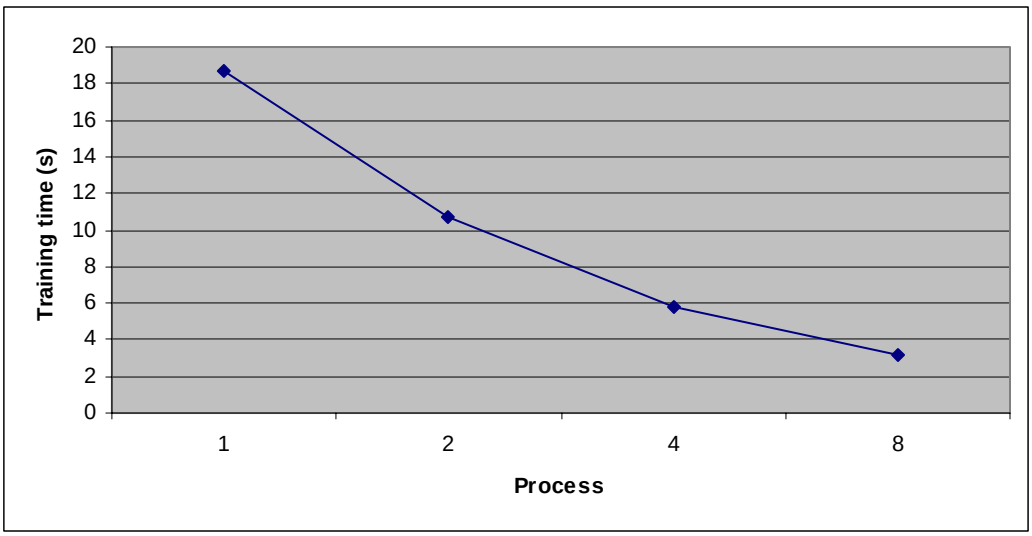
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 24

ตารางที่ 24 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร U

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.715 | 10.715 | 5.815 | 3.171 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.75 | 3.22 | 5.90 |
| ความถูกต้องในการรู้จำ | 92 % | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

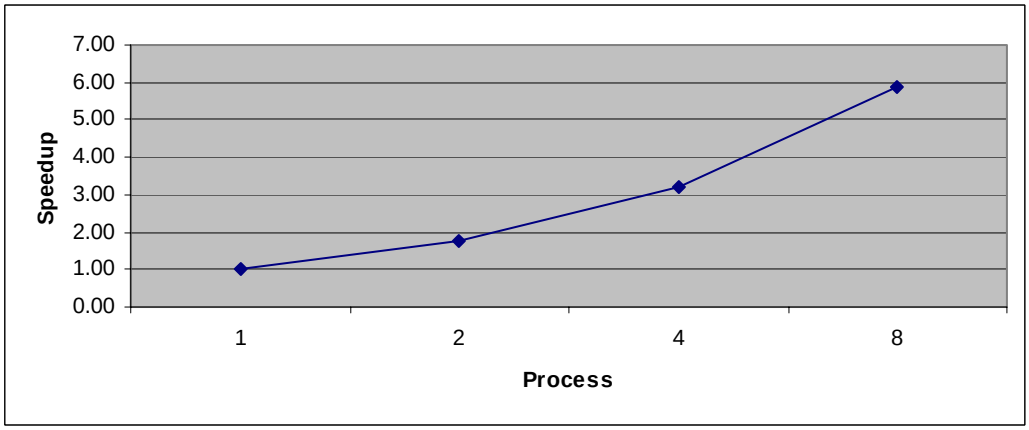
จากตารางที่ 24 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร U ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร U ดังแสดงในแผนภูมิที่ 41 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร U ดังแสดงในแผนภูมิที่ 42




แผนภูมิที่ 41 แสดงเวลาในการเรียนรู้ตัวอักษร U

จากแผนภูมิที่ 41 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 42 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร U

จากแผนภูมิที่ 42 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

22. การเรียนรู้ตัวอักษร V

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4818
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 66
- ความสูงตัวอักษร : 73

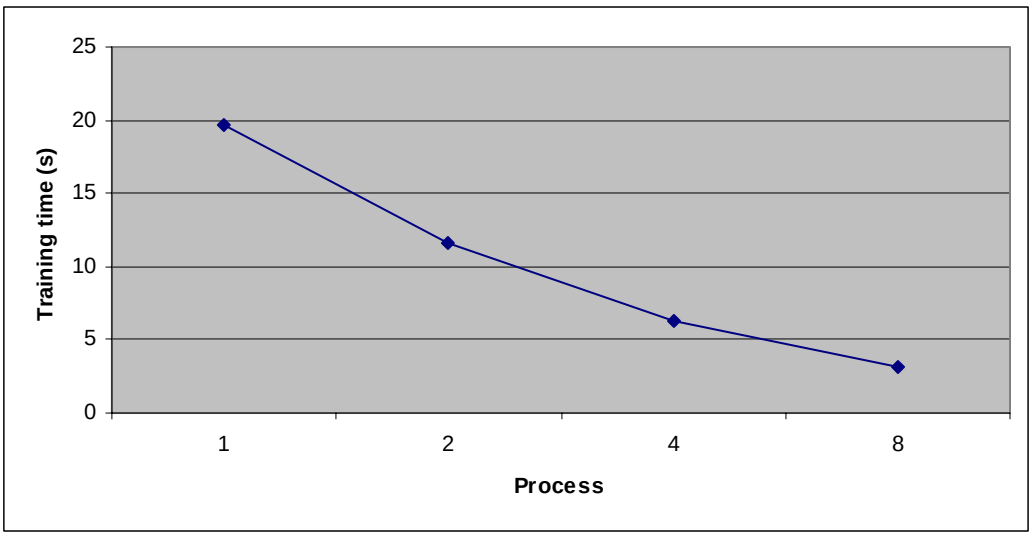
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 25

ตารางที่ 25 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร V

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 19.659 | 11.63 | 6.311 | 3.121 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.69 | 3.12 | 6.30 |
| ความถูกต้องในการรู้จำ | 96% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

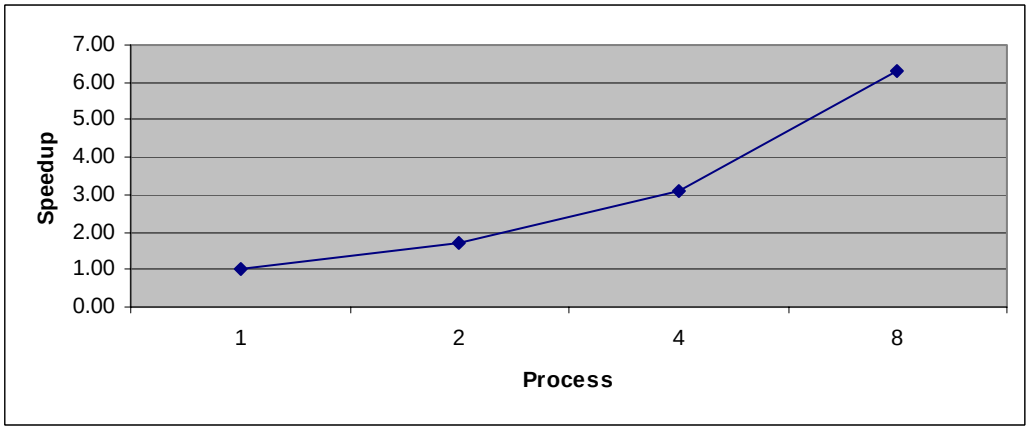
จากตารางที่ 25 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร V ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร V ดังแสดงในแผนภูมิที่ 43 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร V ดังแสดงในแผนภูมิที่ 44




แผนภูมิที่ 43 แสดงเวลาในการเรียนรู้ตัวอักษร V

จากแผนภูมิที่ 43 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 44 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร V

จากแผนภูมิที่ 44 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

23. การเรียนรู้ตัวอักษร W

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 6789
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 93
- ความสูงตัวอักษร : 73

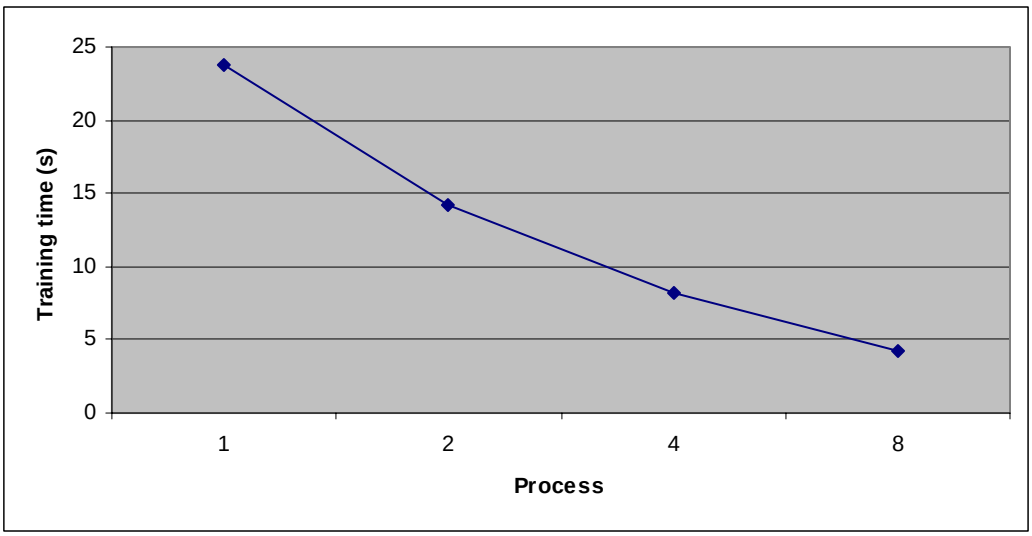
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 26

ตารางที่ 26 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร W

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 23.732 | 14.201 | 8.193 | 4.217 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.67 | 2.90 | 5.63 |
| ความถูกต้องในการรู้จำ | 86.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

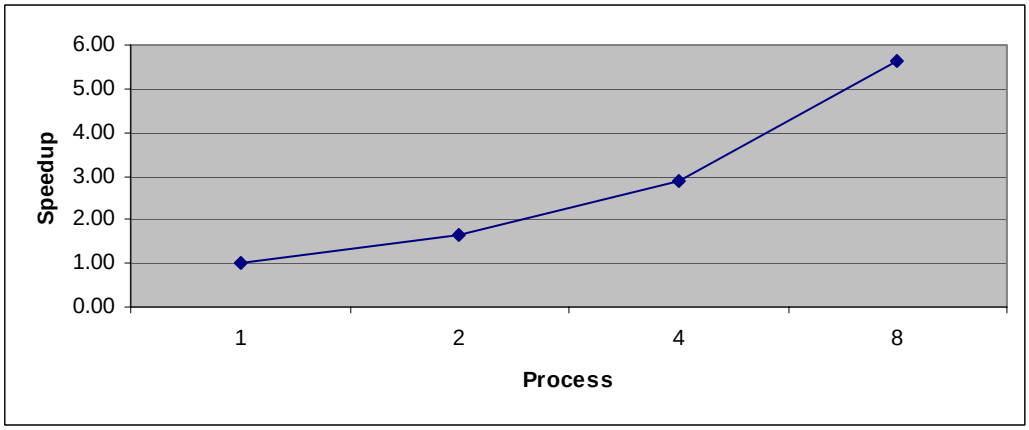
จากตารางที่ 26 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร W ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร W ดังแสดงในแผนภูมิที่ 45 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร W ดังแสดงในแผนภูมิที่ 46




แผนภูมิที่ 45 แสดงเวลาในการเรียนรู้ตัวอักษร W

จากแผนภูมิที่ 45 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 46 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร W

จากแผนภูมิที่ 46 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

24. การเรียนรู้ตัวอักษร X

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4599
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 63
- ความสูงตัวอักษร : 73

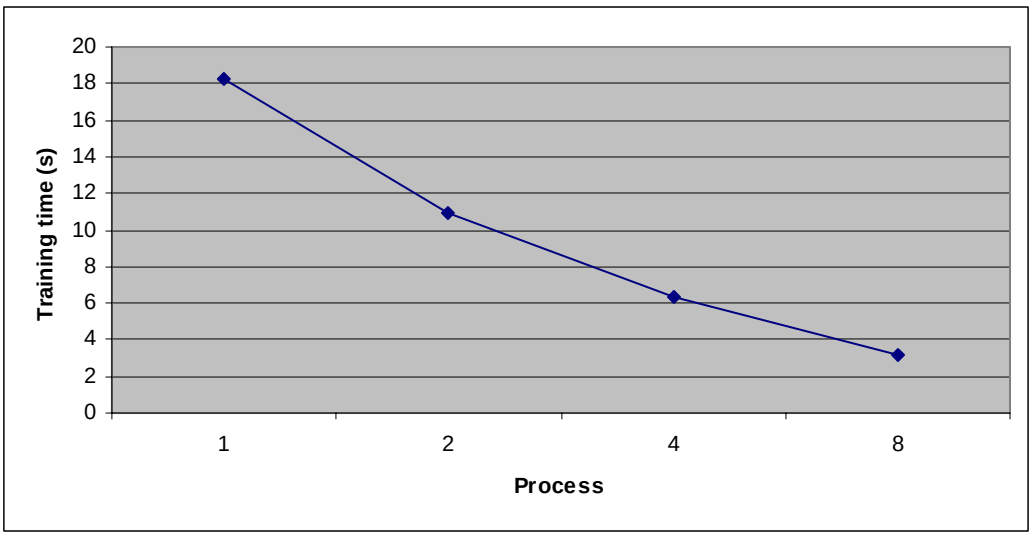
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 27

ตารางที่ 27 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร X

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 18.21 | 10.891 | 6.31 | 3.221 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.67 | 2.89 | 5.65 |
| ความถูกต้องในการรู้จำ | 91.5% | | | |

มหาวิทยาลัยศิลปากร ศูนย์วิทยทรัพยากร

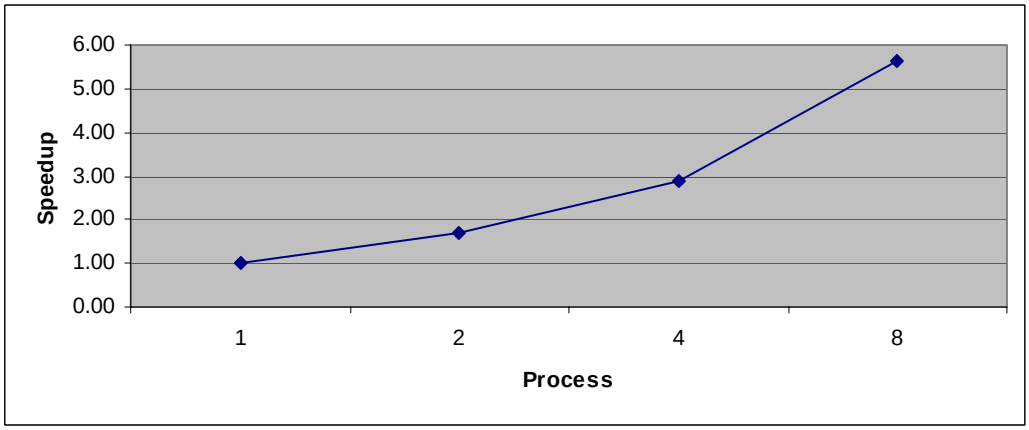
จากตารางที่ 27 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร X ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร X ดังแสดงในแผนภูมิที่ 47 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร X ดังแสดงในแผนภูมิที่ 48




แผนภูมิที่ 47 แสดงเวลาในการเรียนรู้ตัวอักษร X

จากแผนภูมิที่ 47 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 48 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร X

จากแผนภูมิที่ 48 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

25. การเรียนรู้ตัวอักษร Y

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4453
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 61
- ความสูงตัวอักษร : 73

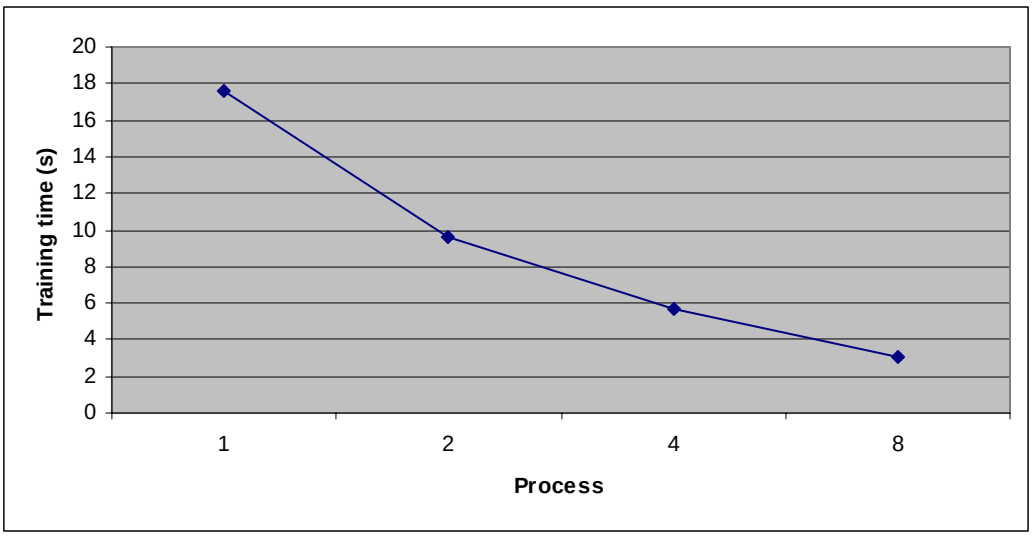
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 28

ตารางที่ 28 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Y

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 17.56 | 9.612 | 5.7061 | 3.011 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1 | 1.83 | 3.08 | 5.83 |
| ความถูกต้องในการรู้จำ | 94.5% | | | |

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

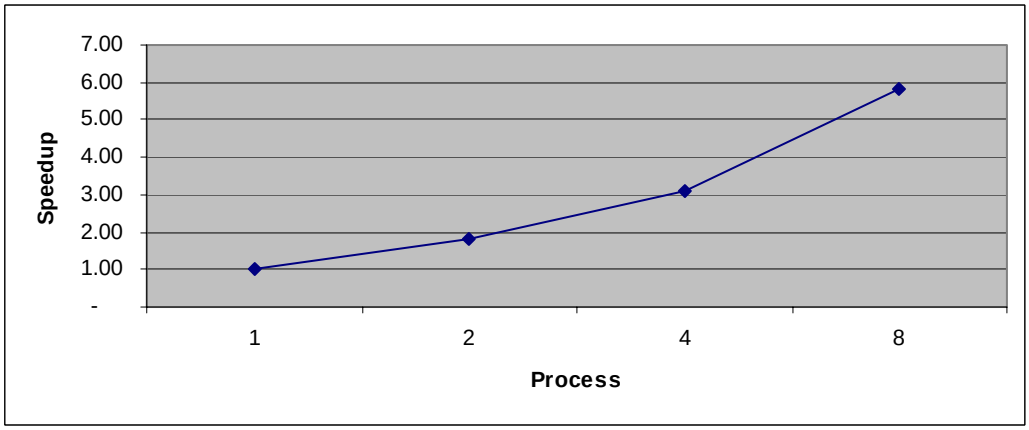
จากตารางที่ 28 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร Y ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร Y ดังแสดงในแผนภูมิที่ 49 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร Y ดังแสดงในแผนภูมิที่ 50




แผนภูมิที่ 49 แสดงเวลาในการเรียนรู้ตัวอักษร Y

จากแผนภูมิที่ 49 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




แผนภูมิที่ 50 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Y

จากแผนภูมิที่ 50 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซสและค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

26. การเรียนรู้ตัวอักษร Z

- จำนวนโหนดในชั้นข้อมูลเข้า (Input node): 4380
- จำนวนโหนดในชั้นซ่อน (Hidden node) : 512
- จำนวนโหนดในชั้นผลลัพธ์ : 8
- ความกว้างตัวอักษร : 60
- ความสูงตัวอักษร : 73

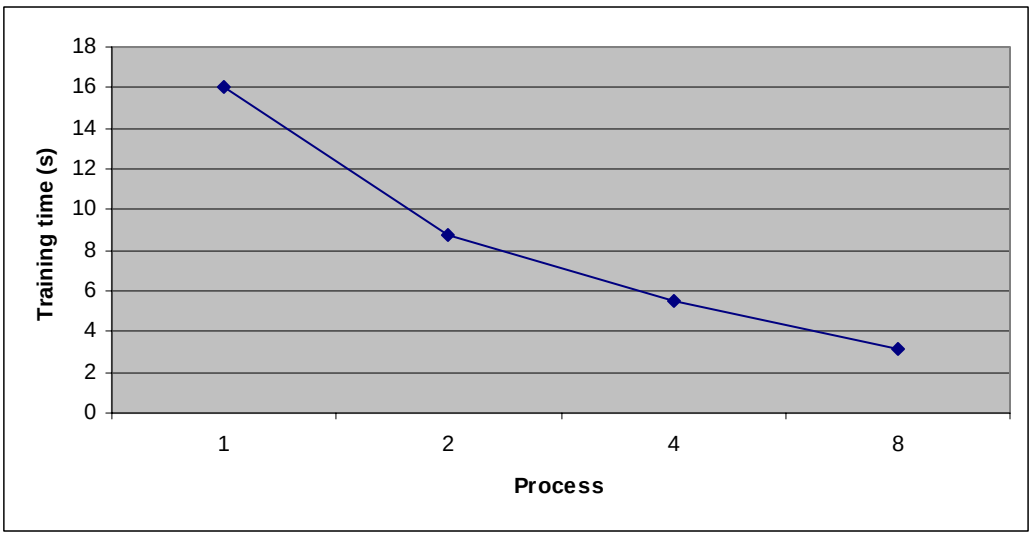
จากข้อกำหนดเบื้องต้นเมื่อนำตัวอย่างตัวอักษรมาผ่านการประมวลผลการฝึกสอนหรือเรียนรู้โดยใช้คุณสมบัติเฉพาะตัวพื้นฐานในการเรียนรู้ เช่น ความกว้างและความสูงร่วมด้วยแล้ว จะได้ผลลัพธ์ดังตารางที่ 29

ตารางที่ 29 เวลาและอัตราความเร็วที่ใช้ในการเรียนรู้ตัวอักษร Z

| จำนวนโปรเซส (Process number) | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| เวลาที่ใช้ในการเรียนรู้ (Training Time (second)) | 16.027 | 8.793 | 5.463 | 3.104 |
| อัตราความเร็วในการเรียนรู้ (Speedup) | 1.00 | 1.82 | 2.93 | 5.16 |
| ความถูกต้องในการรู้จำ | 92% | | | |

มหาวิทยาลัยเทคโนโลยีสุรนารี ลิขสิทธิ์

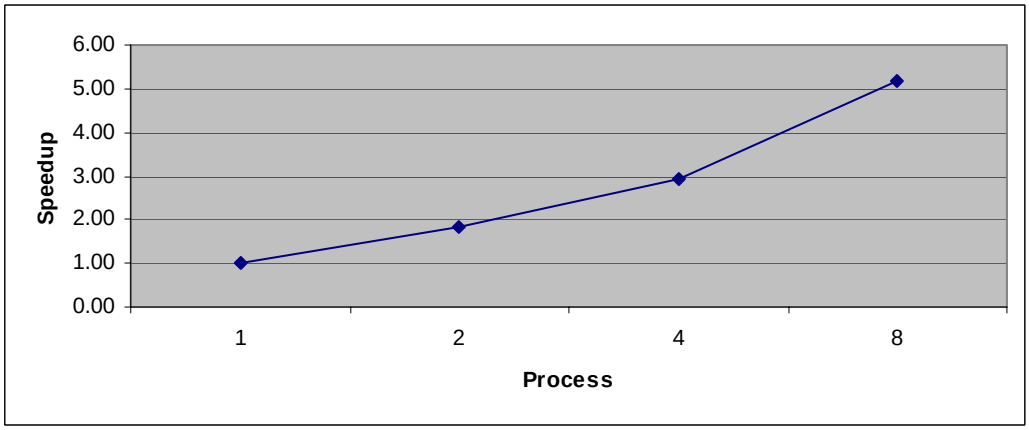
จากตารางที่ 29 เป็นการทดลองประมวลผลการเรียนรู้ตัวอักษร Z ที่มีขนาดกว้าง 100 พิกเซล สูง 100 พิกเซล สามารถเขียนแผนภูมิแสดงเวลาในการเรียนรู้ตัวอักษร Z ดังแสดงในแผนภูมิที่ 51 และสามารถวัดประสิทธิภาพด้านความเร็วโดยเขียนเป็นแผนภูมิแสดงอัตราเร็วในการเรียนรู้ตัวอักษร Z ดังแสดงในแผนภูมิที่ 52




แผนภูมิที่ 51 แสดงเวลาในการเรียนรู้ตัวอักษร Z

จากแผนภูมิที่ 51 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าเวลาที่ใช้ในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้เวลาในการเรียนรู้ลดลงตามลำดับ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์



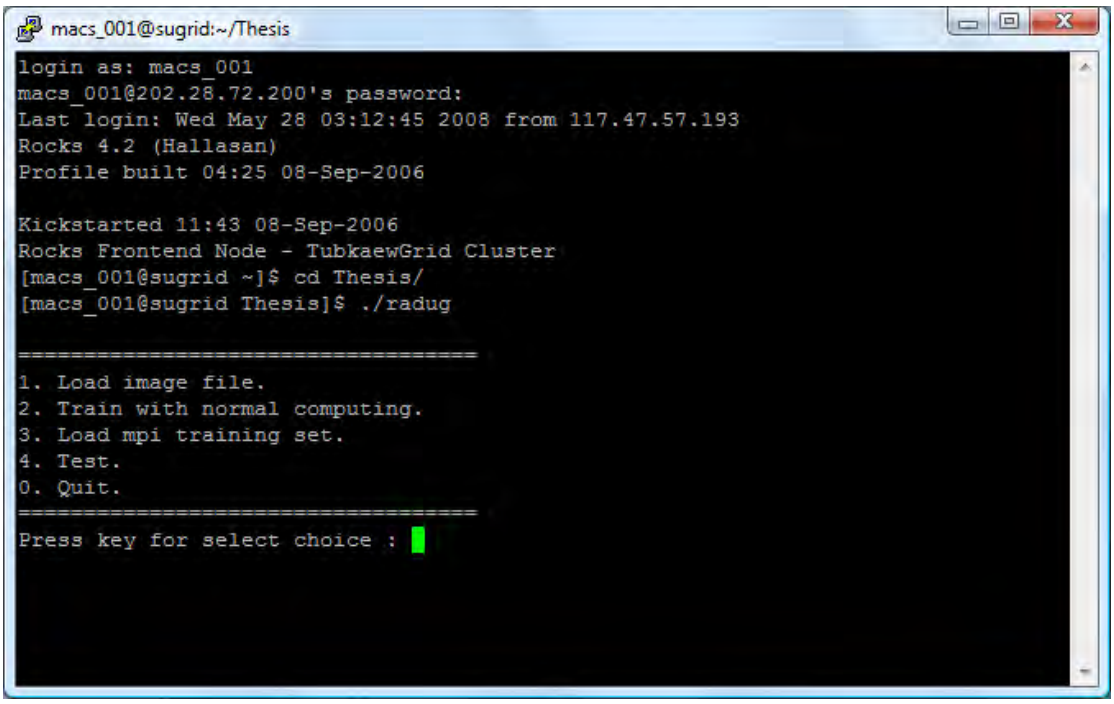

แผนภูมิที่ 52 แสดงอัตราเร็วในการเรียนรู้ตัวอักษร Z

จากแผนภูมิที่ 52 ค่าทางด้านแนวนอนหมายถึงจำนวนโปรเซส และค่าทางด้านแนวตั้งเป็นค่าอัตราเร็วในการเรียนรู้ โดยจะพบว่าการเพิ่มจำนวนโปรเซสจะทำให้อัตราการเร็วเพิ่มขึ้นตามลำดับ

# ภาคผนวก ข

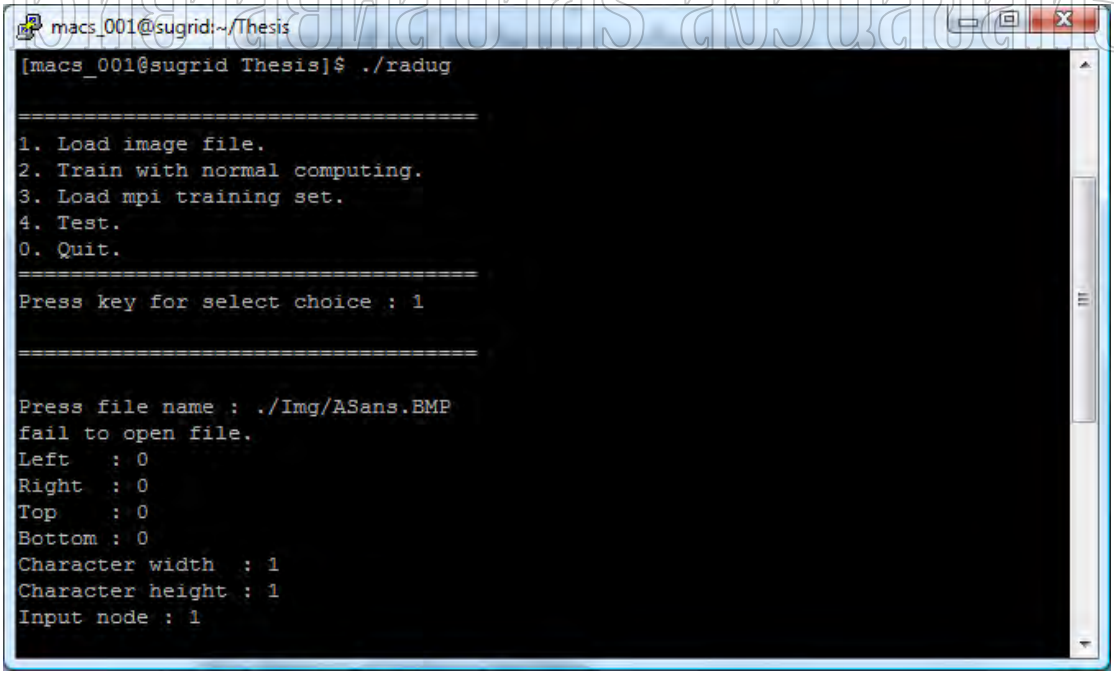
## คู่มือการใช้งานโปรแกรม

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

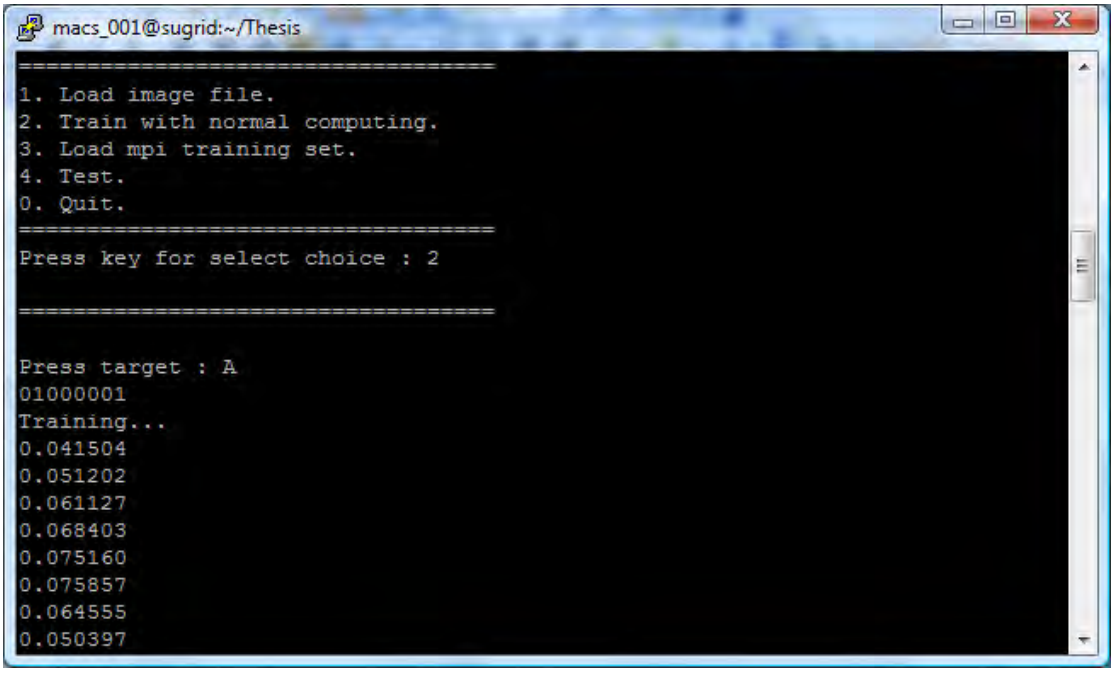



รูปที่ 33 เข้าสู่โปรแกรมจะพบเมนูสำหรับเข้าใช้งานโปรแกรมประยุกต์การรู้จำ

มหาวิทยาลัยศิลปากร ศูนย์วิทยบริการ

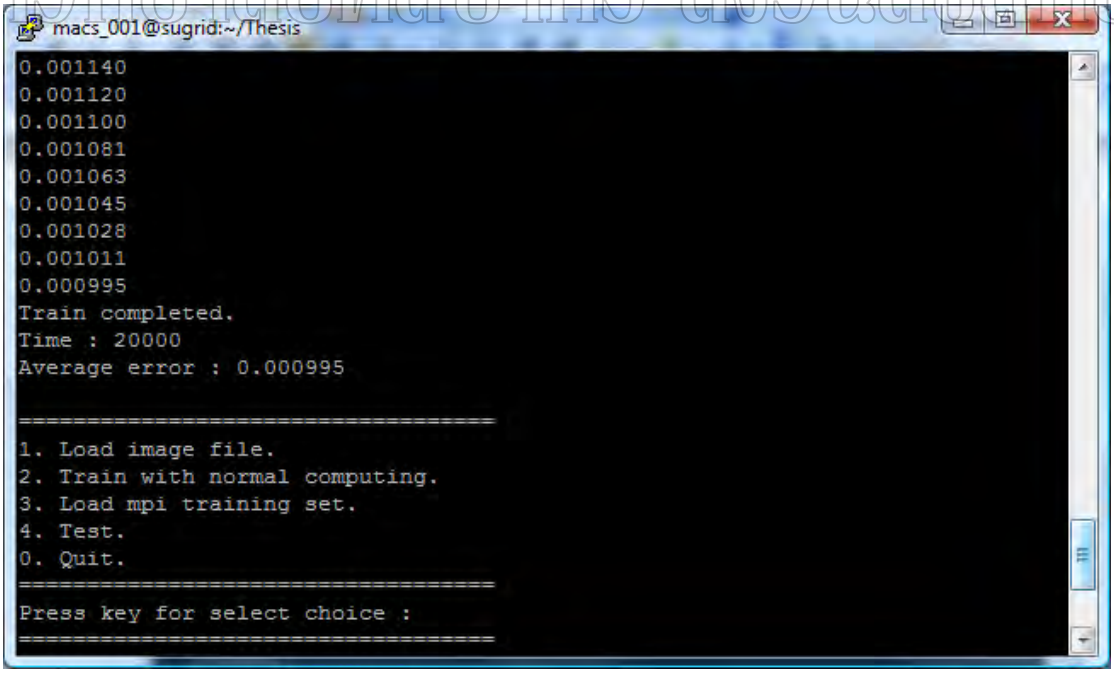



รูปที่ 34 เลือกไฟล์ภาพเข้าสู่ระบบเพื่อเริ่มต้นการประมวลผลภาพเบื้องต้น

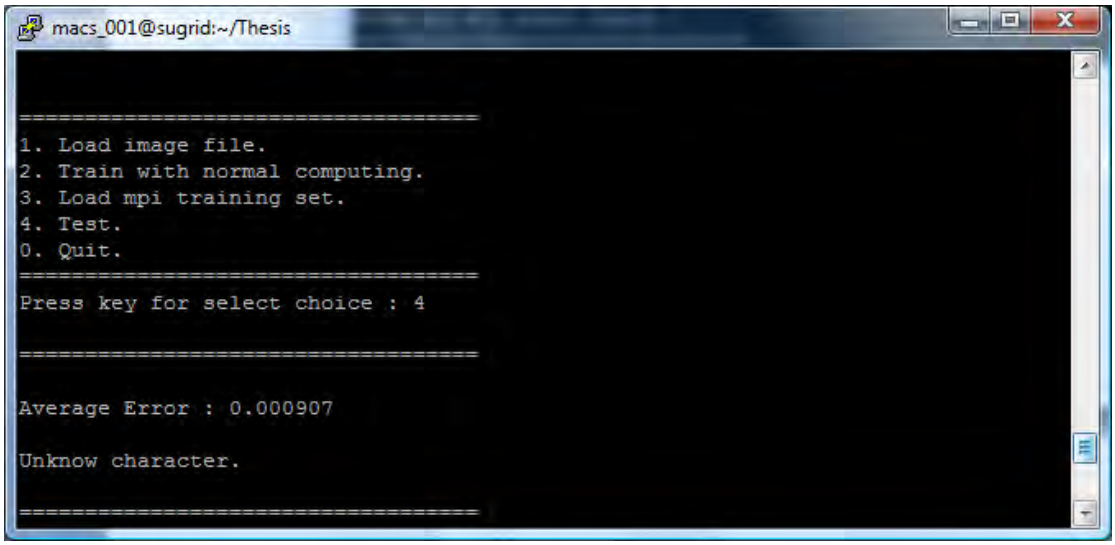



รูปที่ 35 เริ่มทำการประมวลผลการเรียนรู้

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์




รูปที่ 36 เมื่อประมวลผลเสร็จจะแสดงเวลาที่ใช้และค่าผิดพลาดที่ยอมรับได้




รูปที่ 37 แสดงผลเมื่อทำการทดสอบการรู้จำ

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

# ประวัติผู้วิจัย

| | |
|---|---|
| ชื่อ – สกุล | นายไพศาล สิมาเลาเต่า |
| ที่อยู่ | 45 ถนนนาสร้าง ตำบลนครปฐม อำเภอเมืองนครปฐม จังหวัดนครปฐม 73000 |
| ที่ทำงาน | 85 มหาวิทยาลัยราชภัฏนครปฐม ถนนมาลัยแมน อำเภอเมืองจังหวัดนครปฐม 73000 |

ประวัติการศึกษา

| | |
|---|---|
| พ.ศ. 2545 | สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏนครปฐม |
| พ.ศ. 2546 | ศึกษาต่อในระดับปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยศิลปากร |

มหาวิทยาลัยศิลปากร สงวนลิขสิทธิ์

ประวัติการทำงาน

| | |
|---|---|
| พ.ศ. 2545 – ปัจจุบัน | พนักงานมหาวิทยาลัย สายการสอน (อาจารย์) กลุ่มโปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏนครปฐม |