

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เรื่อง

การพัฒนาโครงข่ายประสาทเทียมสำหรับใช้ในการรู้จำตัวอักษร  
Development of Neural Network for Character Recognition



เลขหม.....  
เลขทะเบียน 34086  
วัน, เดือน, ปี 5 ต.ค. 2542

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง

การพัฒนาโครงข่ายประสาทเทียมสำหรับใช้ในงานการรู้จำตัวอักษร  
Development of Neural Network for Character Recognition



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2541

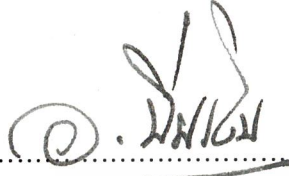
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนานิวรอลเน็ตเวิร์คสำหรับใช้ในการรู้จำตัวอักษร

ผู้จัดทำ

นายพงศ์พันธ์ กิจสนาโยธิน

 (อาจารย์ที่ปรึกษา)

(ผศ.ดร.เอื้อน ปิ่นเงิน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การพัฒนานิวรอลเน็ตเวิร์คสำหรับใช้ในงานการรู้จำตัวอักษร

นายพงศ์พันธ์ กิจสนาโยธิน

ผศ.ดร.เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา

ปีการศึกษา 2541

### บทคัดย่อ

ปริญญานิพนธ์นี้ทำการพัฒนานิวรอลเน็ตเวิร์คสำหรับงานในการรู้จำตัวอักษร โดยใช้การเรียนรู้แบบ Backpropagation ในนิวรอลเน็ตเวิร์คแบบ Feed-forward ซึ่งมีการเพิ่มเติมการปรับค่าของจำนวน Hidden unit เพื่อความรวดเร็วและได้ผลลัพธ์ที่ถูกต้องในการเรียนรู้ ซึ่งภายในงานวิจัยนี้ยังมีการอธิบายถึงกระบวนการทำงานในการจัดรูปแบบของข้อมูลที่จะเข้าอย่างคร่าว ๆ ประกอบไปด้วยการตัดแบ่งตัวอักษร การทำให้เส้นบาง การเปลี่ยนขนาด ผลการทดลองสามารถทำงานได้ตามวัตถุประสงค์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Development of Neural Network for Character Recognition

Mr.Phongphun Kijsanayothin

Ass.Prof.Dr. Ouen Pin-ngern

1998

### Abstract

This thesis presents the development of neural network for character recognition using Backpropagation in feed forward neural network. That have adjusts hidden unit and produces exact result, which reduces computation time. Within this research we also explain a process of rearranging the input data such as character segmentation, character thinning, character scaling, etc. The result of this research is very promising regarding to the planned objectives



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

บทคัดย่อ	I
Abstract	II
สารบัญภาพ	V
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์และขอบเขตของงานวิจัย	2
1.2 อุปกรณ์ที่ใช้ในงานวิจัย	2
1.3 โครงร่างของงานวิจัย	3
บทที่ 2 อาร์ติฟิเชียลนิวรอลเน็ตเวิร์กและการเรียนรู้แบบ Backpropagation	4
2.1 Backpropagation Neural Network	5
2.1.1 โครงสร้างของ Neural Network	6
2.1.2 กระบวนการทำงาน	6
2.2 Activation function	7
2.3 ขั้นตอนการทำงานของกระบวนการ Training	8
บทที่ 3 การเรียนรู้ Backpropagation แบบสามารถปรับ Hidden Unit ได้	11
3.1 Algorithm	11
3.2 ขั้นตอนการทดลอง	13
บทที่ 4 การจัดการข้อมูลก่อนป้อนเข้านิวรอลเน็ตเวิร์ก	16
4.1 Segmentation	16
4.2 การทำลายเส้นให้บาง	18
4.2.1 ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ Zhang and Suen	19
4.2.2 ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ	
A.DATTA and S.K.PARUI	21
4.3 การทำให้ตัวอักษรอยู่ในขนาดมาตรฐาน	23
บทที่ 5 แนะนำวิธีการใช้งาน โปรแกรมตัวอย่างเบื้องต้น	25
5.1 เริ่มแรกการใช้งาน โปรแกรมตัวอย่าง	25
5.2 การทำงานในรูปแบบฝึกสอน	26
5.3 การทำงานในรูปแบบการแบ่งแยกตัวอักษร	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

บทที่ 6 บทสรุปและวิจารณ์	31
บรรณานุกรม	32
ภาคผนวก Source code	33



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

ภาพที่ 2.1.1 แสดงโครงสร้างของ Neural network แบบหลายเลเยอร์	5
ภาพที่ 3.1.1 แสดงการทำงานของ การ Train Backpropagation แบบสามารถปรับค่า Hidden Unit ได้	12
ภาพที่ 3.2.1 แสดงตัวอักษร A-F ชุดที่ 1	14
ภาพที่ 3.2.2 แสดงตัวอักษร A-F ชุดที่ 2	15
ภาพที่ 3.2.3 แสดงตัวอักษร A-F ชุดที่ 3	15
ภาพที่ 4.1.1 แสดงการหาเส้นขอบด้านบนของตัวอักษร โดยเริ่มจากเส้นกรอบด้านบนของภาพ	17
ภาพที่ 4.1.2 แสดงการหาเส้นขอบด้านล่างของตัวอักษร โดยเริ่มจากเส้นขอบด้านบนของตัวอักษร	17
ภาพที่ 4.1.3 แสดงการหาเส้นขอบด้านซ้ายของตัวอักษร โดยเริ่มจากเส้นกรอบด้านซ้ายของภาพ	17
ภาพที่ 4.1.4 แสดงการหาเส้นขอบด้านขวาของตัวอักษร โดยเริ่มจากเส้นขอบด้านซ้ายของตัวอักษร	18
ภาพที่ 4.1.5 แสดงภาพของตัวอักษรที่ถูกเลือกแต่ยังไม่ได้รับการตัดเฉพาะตัวอักษร	18
ภาพที่ 4.1.6 แสดงภาพของตัวอักษรและภาพดั้งเดิมหลังจากทำงานเสร็จสำหรับตัวอักษรหนึ่ง ๆ แล้ว	18
ภาพที่ 4.2.2.1 แสดงแม่แบบ (ก) ถึง (ง) ที่ใช้ในการทำลายเส้นให้บาง ทั้ง 4 แบบและวินโดว์ (จ) ขนาดเมตริก 3 x 3	21
ภาพที่ 5.1 ภาพแสดงรูปของการทำงาน โปรแกรมตอนเริ่มต้น	25
ภาพที่ 5.2 แสดงหน้าต่างของ User Interface ใน Mode Train	26
ภาพที่ 5.2.1 แสดงผลเมื่อมีการกดปุ่ม Load Button	27
ภาพที่ 5.2.2 แสดงผลหลังจากมีการเลือกภาพเสร็จ	28
ภาพที่ 5.2.3 แสดงผลหลังจากมีการกดปุ่ม Next	28
ภาพที่ 5.2.4 แสดงผลหลังจากมีการกดปุ่ม Train	29
ภาพที่ 5.3.1 หน้าตา User Interface ของโปรแกรมในรูปแบบการแบ่งแยกตัวอักษร	29
ภาพที่ 5.3.2 แสดงผลเมื่อกดปุ่ม Run	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

ในปัจจุบัน คอมพิวเตอร์มีความสามารถในการคำนวณเชิงคณิตศาสตร์สูงมาก และให้ความแม่นยำในการคำนวณได้รวดเร็ว แต่ยังมีข้อจำกัดบางประการ เช่น การตีความหมายเชิงวัตถุของภาพ ความสามารถในการเรียนรู้และตัดสินใจโดยใช้ประสบการณ์ เราจึงต้องการคอมพิวเตอร์ที่มีความสามารถในการทำงานในลักษณะข้างต้นที่จะต้องไม่ทำงานตามคำสั่งหรือโปรแกรมที่กำหนดให้เท่านั้น

จากเหตุผลดังกล่าวข้างต้น จึงทำให้นักวิชาการพยายามให้ความสำคัญในการออกแบบและสร้างระบบคอมพิวเตอร์ให้มีโครงสร้างการทำงานเลียนแบบการทำงานของเซลล์ในสมองของมนุษย์ หรือที่เรียกกันว่า นิวรอล ซึ่งนิวรอลจะทำงานในแง่ของสตรีวิทยาประสาท ที่จะต่อเชื่อมโยงกันเป็นโครงข่ายของหน่วยประสาท หรือ นิวรอลเน็ตเวิร์ค (Neural Network) โดยใช้วิธีการประมวลผลแบบขนาน และการกระจายการทำงาน (Parallel Distributed Processing) เพื่อแปลงข้อมูลจากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่งตามที่ต้องการ

นอกจากนั้นคุณสมบัติของนิวรอลเน็ตเวิร์คยังแตกต่างจากหลักการทำงานในเชิงคณิตศาสตร์โดยทั่วไปในเรื่องของ Massively parallel processing ซึ่งคุณสมบัติประการแรก คือ Fault tolerance หมายถึง ระบบประมวลผลจะไม่หยุดทำงานกลางคัน ถึงแม้ว่าหน่วยประมวลผลบางหน่วยได้รับความเสียหายจนทำงานไม่ได้ก็ตามคุณสมบัติอีกประการหนึ่งที่สำคัญคือ Generalisation หมายถึง นิวรอลเน็ตเวิร์คยังคงให้ผลลัพธ์ที่ต้องการ ได้ถึงแม้จะมีสัญญาณรบกวนหรือข้อมูลขาดความสมบูรณ์หรือข้อมูลที่ไม่เคยเห็นมาก่อนแต่คล้ายกับข้อมูลที่เคยเข้ามาก่อนแล้ว คุณสมบัติประการสุดท้ายคือ Adaptability หมายถึง นิวรอลเน็ตเวิร์คสามารถเรียนรู้สภาพแวดล้อมใหม่ที่แตกต่างจากสภาพแวดล้อมเดิมได้อย่างรวดเร็ว

การทำงานของนิวรอลเน็ตเวิร์คจะอยู่ในรูปแบบของการฝึกสอน (Training) ทั้งนี้เพื่อปรับพื้นฐานความรู้ด้วยค่าตัวเลขน้ำหนัก (Weight) ทำให้ค่า input และ output ของระบบสอดคล้องกัน จากนั้นจึงจะสามารถนำฐานความรู้ (Knowledge base) ที่ได้ไปใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.1 วัตถุประสงค์และขอบเขตของการวิจัย

ศึกษาการทำงานของนิวรอลเน็ตเวิร์คและสามารถปรับความสามารถในการเรียนรู้ของนิวรอลเน็ตเวิร์คได้ และสามารถใช้ในการวิเคราะห์ตัวอักษรไม่จำกัดรูปแบบได้ โดยในเบื้องต้นจะอธิบายการทำงานอย่างคร่าว ๆ ของ Neural Network จากนั้นจะเข้าไปถึงในส่วนของการเรียนรู้แบบ Backpropagation ซึ่งเป็นการเรียนรู้ที่นิยมใช้กันมากที่สุดสำหรับปัญหาในแนวการทำงานในลักษณะนี้แต่การเรียนรู้แบบ Backpropagation ก็ยังมีข้อเสียตรงที่ไม่สามารถรู้ค่าของ Hidden unit ที่เหมาะสมได้จึงมีการอธิบายการพัฒนาการเรียนรู้แบบที่สามารถปรับค่าของ Hidden unit ได้ เมื่อมาถึงขั้นตอนนี้ก็จะอธิบายการทำ pre-processing เป็นการจัดรูปแบบต่าง ๆ ของค่า input ที่จะเข้าไปทำงานใน Neural Network และสุดท้ายก็จะไปการอธิบายการใช้การ Software DEMO นี้อย่างละเอียด

การทำงานของ Software DEMO นี้จะมีการทำงานคล้ายกับการทำงานของ Software อื่นตามทั่วไปแต่จะไม่มีตกแต่งความสวยงามมากนักเนื่องจากไม่มีความจำเป็นเพราะต้องการศึกษาเฉพาะในแง่ของการทำงานของนิวรอลเน็ตเวิร์คและความเร็วที่เพิ่มขึ้นสำหรับการเรียนรู้แบบใหม่นี้เท่านั้นส่วนในแง่ของความถูกต้องของการทำงานนั้นเนื่องจากการทำวิจัยครั้งนี้ใช้จำนวนของ Hidden unit เพียงขั้นเดียวเท่านั้นจึงมีผลอย่างมากในการแบ่งแยกความแตกต่างของตัวอักษรผลที่ได้จึงอาจจะดีไม่เท่า Software ตามท้องตลาดอื่น แต่ Software นี้จะมีส่วนดีตรงที่ใครต้องการจะทำการแบ่งแยกอะไรก็ได้ขอเพียงแค่มีตัวอักษรต้นแบบเท่านั้นก็สามารถที่จะสร้าง Neural Network ตามที่ตนต้องการได้ทันที

## 1.2 อุปกรณ์ที่ใช้ในการวิจัย

1. เครื่องคอมพิวเตอร์แบบใดก็ได้ขอเพียงมี JAVA RUNTIME ENVIRONMENT ติดตั้งอยู่
2. ตัวอย่างตัวอักษรที่ต้องการแบ่งแยกในที่นี้เนื่องจากเขียนโดยภาษา JAVA ภาพที่ต้องการจึงต้องเป็น Format แบบ JPG หรือ GIF เท่านั้น

### 1.3 โครงร่างของงานวิจัย

รายละเอียดของงานวิจัยประกอบด้วยบทต่าง ๆ ดังนี้

บทที่ 1 บทนำจะกล่าวถึงความเป็นมาและวัตถุประสงค์ของงานวิจัยนี้

บทที่ 2 อาร์ตทิฟิเชียลนิวรอลเน็ตเวิร์คและการเรียนรู้แบบ Backpropagation จะแสดงการทำงานของการเรียนรู้แบบ Backpropagation และ ประวัติความเป็นมาของอาร์ตทิฟิเชียลนิวรอลเน็ตเวิร์ค

บทที่ 3 การเรียนรู้แบบ Backpropagation แบบที่ผสมกับการปรับค่า Hidden unit แบบอัตโนมัติ (Automatic change Hidden unit) จะแสดงวิธีในการเรียนรู้แบบใหม่ที่สามารถหาค่าของ Hidden unit แบบมี Hidden Layer เพียง Layer เดียว

บทที่ 4 แสดงการทำงานในการประมวลผลข้อมูลการเข้า Neural Network ประกอบไปด้วยการทำงาน 3 แบบ คือ การแบ่งแยกตัวอักษรออกเป็นตัว ๆ ออกจากกัน, การทำลายเส้นในบาง, การเปลี่ยนขนาด

บทที่ 5 แนะนำวิธีการใช้งาน โปรแกรมตัวอย่างเบื้องต้น

บทที่ 6 บทสรุป วิเคราะห์ และข้อเสนอแนะ เป็นการสรุปการศึกษาที่ได้จากการทำงานชิ้นนี้ พร้อมทั้งมีข้อเสนอแนะในการปรับปรุงประสิทธิภาพ

## บทที่ 2

### อาร์ติฟิเชียลนิวรอลเน็ตเวิร์คและการเรียนรู้แบบ Backpropagation

ประวัติความเป็นมาของโครงข่ายหน่วยประสาทหรือ Artificial neural networks อาจกล่าวได้ว่า ปี 1943 เป็นปีที่เริ่มต้นของการพัฒนาระบบ Artificial neural networks เนื่องจาก McCulloch และ Pitts ได้เสนอรูปแบบหน่วยประมวลผลทางนิวรอล แต่การนำแนวความคิดไม่ใช้งานยังไม่เกิดผลเท่าที่ควร ต่อมาในปี 1949 Donald Hebb ได้เสนอทฤษฎีการเรียนรู้สำหรับ Artificial neural networks ซึ่งนักวิทยาศาสตร์สมัยนั้นเห็นว่าเน็ตเวิร์คของนิวรอลสามารถเรียนรู้ได้

ในปี 1950 ถึง 1960 มีกลุ่มผู้วิจัยทั้งนักชีววิทยาและนักฟิสิกส์ได้ผลิตอุปกรณ์วงจรอิเล็กทรอนิกส์ด้วยความรู้ทางด้าน Artificial neural networks เป็นครั้งแรก โดยการพัฒนาในครั้งนี้ Marvin Minsky ในปี 1954 และ Frank Rosenblatt ในปี 1958 และ Bernard Widrow and Marcian Hoff ในปี 1960 มีส่วนสำคัญยิ่ง ซึ่งได้มีส่วนในการพัฒนาในครั้งนี้ในแบบ Network แบบ เลเยอร์เดียวของแบบจำลองนิวรอล ที่เรียกว่า เพอเซ็ปตรอน ที่นำมาใช้กับปัญหาต่าง ๆ แต่ต่อมากลับพบว่า โครงสร้าง network ดังกล่าวไม่สามารถแก้ปัญหาผิวเหินบางอย่างได้สำเร็จ ซึ่งต่อมา Minsky และ Seymour Papert ได้เขียนหนังสือขึ้นเพื่อพิสูจน์ว่า นิวรอลเน็ตเวิร์คแบบชั้นเดียวมีข้อจำกัดที่ไม่สามารถแก้ปัญหาบางประการได้

การพัฒนา Artificial neural networks ได้หยุดชะงักเป็นเวลานาน จนกระทั่ง Paul Werbos ในปี 1974 และ Parker ในปี 1986 และ David Rumelhart, James McClelland, Geoffrey Hinton, Ronald Williams ในปี 1986 ได้มีผลจากการค้นคว้าเกี่ยวกับเน็ตเวิร์คแบบหลายเลเยอร์ ที่สามารถเรียนรู้ได้ด้วยความสามารถของ Backpropagation โดยเฉพาะ McClelland and Rumelhart ในปี 1986 และคณะผู้วิจัย ได้รวบรวมทฤษฎีและแนวทางการเรียนรู้ใหม่ ๆ ในหนังสือ 2 เล่ม เรื่อง "Parallel Distributed Processing" (PDP) ซึ่งเป็นการเปิดศักราชใหม่ในการคำนวณศักยภาพของเน็ตเวิร์คแบบหลายเลเยอร์ทำให้เกิดผลงานต่าง ๆ ขึ้นมากมาย

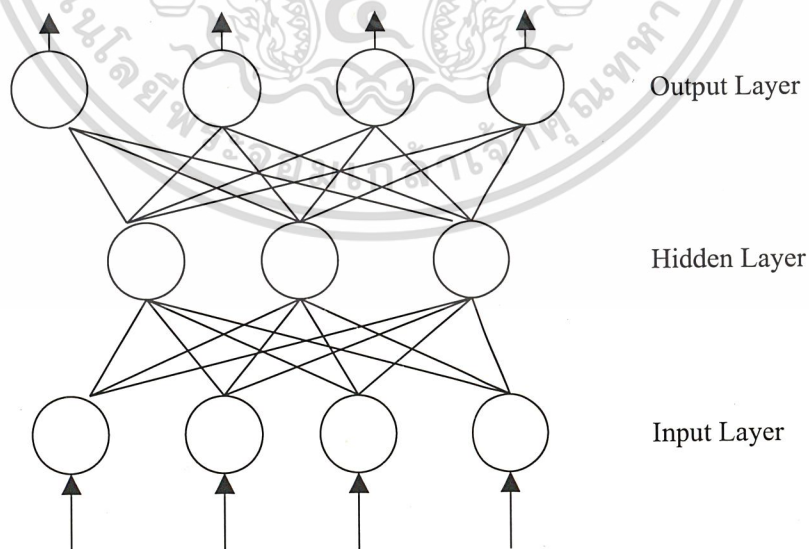
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1 Backpropagation neural network

Backpropagation neural network หรือ Feedforward Backpropagation Network เป็น Neural network แบบหนึ่งที่ได้รับคามนิยมในการใช้มากแบบหนึ่งในจำนวน Neural network ที่มีใช้กันอยู่ในปัจจุบัน ภายในของ Backpropagation neural network ไม่มีการเชื่อมต่อระหว่าง node ต่อ node (Connection link) แบบ Feedback เลยแต่ค่า Error ที่ได้จากการคำนวณในกระบวนการ Training จะถูกส่งกลับมาคำนวณ (Backpropagated) ในระหว่างขั้นตอนการ Training

ในกระบวนการ Training ของ Backpropagation จะประกอบไปด้วยขั้นตอนที่สำคัญทั้งหมด 3 ขั้นตอน คือ การทำกระบวนการ Feedforward ของ Input pattern ที่ต้องการ Train , คำนวณค่าของ Error ที่ได้รับการ Train , ปรับปรุงค่าต่าง ๆ ใน Backpropagation จากค่า Error ที่คำนวณได้จากกระบวนการที่ผ่านมา

โดยทั่วไปแล้ว เราจะกำหนด Hidden layer ของ Backpropagation neural network เป็นแบบ Single layer ซึ่งจะง่ายต่อการคำนวณต่าง ๆ และรวดเร็วในการเรียนรู้แต่บางงานการใช้เพียง Layer เดียวของ Hidden layer นั้นอาจจะไม่พอในการใช้งานเราจึงอาจจะต้องมีการเพิ่มจำนวนชั้นของ Hidden layer ในมากยิ่งขึ้น เพื่อประสิทธิภาพในการทำงานของ Backpropagation neural network แต่ทั้งนี้และทั้งนั้นก็ขึ้นอยู่กับขั้นตอนกระบวนการ การทำงาน และความซับซ้อนของงานซึ่งในรูปภาพที่ 2.1.1 เป็นการแสดงโครงสร้างของ neural network แบบหลายเลเยอร์ที่จะนำมาใช้ในการเรียนรู้แบบ Backpropagation



รูปภาพที่ 2.1.1 แสดงโครงสร้างของ neural network แบบหลายเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.1 โครงสร้างของ Neural network

จาก ภาพที่ 2.1.1 เป็นภาพของ Neural network แบบหลาย Layer (Multi-layer neural network) ซึ่งประกอบด้วยจำนวน Hidden layer จำนวน 1 ชั้น และจำนวน Bias ในชั้นจำนวน 1 ตัว ในการทำงานปกติ Backpropagation จะทำงานเฉพาะในลักษณะของ Feedforward จากนั้นจะนำค่าที่ได้จาก Output layer ไปใช้ในการทำงานได้ทันทีอาจจะต้องมีการปรับค่าของ Output ที่ได้เล็กน้อย แต่เมื่อมาถึงในการทำงานในลักษณะของการ Learning นั้น เมื่อค่าไปที่ Output แล้วก็จะนำค่ามาคำนวณค่าของ Error กลับไปใช้ในการปรับปรุงค่าต่าง ๆ ภายใน Backpropagation neural network อีกต่อหนึ่ง

### 2.1.2 กระบวนการทำงาน

ในขั้นตอนการ Train Backpropagation จะประกอบด้วย ขั้นตอน 3 ขั้นตอน คือ การทำกระบวนการ Feedforward ของ Input pattern ที่ต้องการ Train, คำนวณค่าของ Error ที่ได้รับการ Train, ปรับปรุงค่าต่าง ๆ ใน Backpropagation จากค่า Error ที่คำนวณได้จากกระบวนการที่ผ่านมา

ในขั้นตอนแรก (Feedforward) ค่าใน Input unit แต่ละตัวจะรับค่า Input pattern จากนั้นจะส่งค่าที่ได้รับมาไปแบบ Broadcast ไปที่ชั้นของ Hidden layer ใน Hidden unit แต่ละตัวจะทำการคำนวณค่า Activation และส่งค่าที่ได้ไปยัง output layer เช่นเดียวกับใน output unit แต่ละตัวก็จะทำการคำนวณ Activation เช่นเดียวกับในชั้นของ Hidden layer

ในระหว่างกระบวนการ Training ภายใน output node แต่ละ node จะทำการเปรียบเทียบค่าของตัวเอง (Activation) ที่ได้รับการ Feedforward กับค่าที่ต้องการให้เป็น (Target output) เพื่อนำมากำหนดค่าของ Error สำหรับ Pattern unit นั้น ๆ ที่กำลัง Training อยู่ เมื่อลงมาดูในการคำนวณ Error ค่าของ Factor  $\delta_k$  ( $k=1, \dots, M$ ) จะถูกคำนวณ ซึ่ง Factor ตัวนี้จะนำมาใช้ในการกระจายค่า Error ของ Output unit กลับไปให้กลับ Unit ทุกตัวในชั้นที่ผ่านมา นอกจากนี้ค่า Factor นี้ยังมีผลต่อการปรับปรุงเปลี่ยนแปลงค่าของ Weight ของการเชื่อมต่อระหว่าง Output layer และ Hidden layer นอกจากนี้ยังมี Factor  $\delta_j$  ซึ่ง Factor นี้จะถูกคำนวณใน Hidden unit ค่าของ Factor  $\delta_j$  ตัวนี้ไม่จำเป็นต้องส่งค่า Error กลับไปที่ Input layer แต่ Factor  $\delta_j$  จะถูกใช้ในการปรับปรุงเปลี่ยนแปลงค่าของ Weight ของการเชื่อมต่อระหว่าง Hidden layer กับ Input layer

หลังจากค่าของ Factor ทั้งหมดได้ถูกกำหนดแล้ว Weight ต่าง ๆ ของทุก ๆ Layer จะถูกปรับปรุงไปพร้อม ๆ กัน การปรับปรุง Weight ของชั้นการเชื่อมต่อระหว่าง Hidden layer กับ Input layer นั้น จะใช้ค่าของ Factor  $\delta_j$  และค่าของ Factor  $\delta_k$  ในการคำนวณค่าของ Weight ใหม่ ซึ่งค่าของ Weight ใหม่จะขึ้นอยู่กับค่าของ Factor  $\delta_j$  และค่าของ Factor  $\delta_k$  ที่คำนวณได้ก่อนหน้านี้

Output layer จะขึ้นอยู่กับ Factor  $\delta_k$  และค่าของ Activation ของ Hidden unit ส่วนในการปรับปรุง Weight ของชั้นการเชื่อมต่อระหว่าง Input layer กับ Hidden layer จะขึ้นอยู่กับ Factor  $\delta_j$  และค่าของ Activation ของ Input unit แต่ละตัว

ตัวแปรต่าง ๆ ที่ใช้ใน Backpropagation neural network ในการ Training จะมีดังต่อไปนี้

x Input training vector

$$x = (x_1, x_2, \dots, x_n)$$

t Output target vector

$$t = (t_1, t_2, \dots, t_m)$$

$\delta_k$  ตัวกำหนดความถูกต้องที่ได้จาก Error ใช้สำหรับปรับปรุงเปลี่ยนแปลงค่าของ Weight ที่  $W_{jk}$  ซึ่ง  $W_{jk}$  เป็นค่าของ Weight ที่เชื่อมระหว่าง Output unit  $Y_k$  กับ Hidden unit  $Z_j$  ในชั้นของ Hidden layer

$\delta_j$  ตัวกำหนดความถูกต้องที่ได้จาก Error ใช้สำหรับปรับปรุงเปลี่ยนแปลงค่าของ Weight ที่  $V_{ij}$  ซึ่ง  $V_{ij}$  เป็นค่าของ Weight ที่เชื่อมระหว่าง Hidden unit  $Z_j$  กับ Input unit  $X_i$  ในชั้นของ Input layer

$\alpha$  Learning rate (อัตราการเรียนรู้)

$X_i$  Input unit ที่ตำแหน่ง I

$V_{0j}$  Bias ของ Hidden unit ที่ j

$Z_j$  Hidden unit ที่ตำแหน่ง j ซึ่ง

$$z_{in_j} = V_{0j} + \sum X_i V_{ij}$$

ซึ่งค่า Output sigmoid (Activation) ของ  $z_{in_j}$  จะเป็น  $Z_j$

$$Z_j = f(z_{in_j})$$

$W_{ok}$  Bias ของ Output unit ที่ตำแหน่ง k

$Y_k$  Output unit ที่ตำแหน่ง k ซึ่ง

$$y_{in_k} = W_{ok} + \sum Z_i W_{ik}$$

ซึ่งค่า Output sigmoid (Activation) ของ  $y_{in_k}$  จะเป็น  $Y_k$

$$Y_k = f(y_{in_k})$$

## 2.2 Activation function

Activation function สำหรับ Backpropagation neural network นั้นจะต้องมีข้อกำหนดในการเป็น Function ที่สำคัญ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
1. Function ที่จะเป็น Activation Function จะต้องต่อเนื่อง  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จะต้องสามารถหาค่าของ Function differential ได้
3. ต้องเป็น Monotonically non-decreasing

ทั้งนี้ที่่ต้องกำหนดข้อจำกัดนี้เพื่อต้องการประสิทธิภาพและความสะดวกในการคำนวณ

รูปแบบหนึ่งที่นิยมใช้เป็น Activation function คือ Binary sigmoid function ซึ่ง binary sigmoid function นี้จะให้ค่าของ Function ออกมาอยู่ในระหว่าง 0 ถึง 1 เท่านั้น

$$f_1(x) = 1/(1+\exp(-x)) \quad (\text{Binary sigmoid function})$$

และค่า Differential function ของ Binary sigmoid function

$$f_1'(x) = f_1(x)[1-f_1(x)]$$

จาก Binary sigmoid function ข้างต้น เรายังมี Activation function ที่มีค่าของ Function อยู่ระหว่าง 1 ถึง -1 Function นี้เราจะเรียกว่า Bipolar sigmoid function

$$f_2(x) = 2/(1 + \exp(-x)) - 1$$

และค่า Differential function ของ bipolar sigmoid function

$$f_2'(x) = 0.5 [1 + f_2(x)] [1 - f_2(x)]$$

## 2.3 ขั้นตอนการทำงานของกระบวนการ Training

จากหัวข้อที่กล่าวผ่านมาในเรื่องของ Activation function เราจะนำกลับมาใช้ใน Backpropagation algorithm ซึ่งรูปแบบของข้อมูลก็สำคัญมากสำหรับการเลือก Activation function ในส่วนต่อไปนี่จะเป็นส่วนของกระบวนการ Training Algorithm ซึ่งจะมีขั้นตอนดังต่อไปนี้

**Step 0** กำหนดค่าของ Weight เริ่มต้น

(อาจจะกำหนดโดยการ Random ค่าของ Weight เป็นจำนวนน้อย ๆ)

**Step 1** ในขณะที่ Stopping condition ยังไม่เป็น false ทำ Step 2 ถึง Step 9

**Step 2** สำหรับการ Training ของ Input แต่ละ Pattern ทำ Step 3 ถึง Step 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Feedforward

**Step 3** สำหรับ Input unit ( $X_i, i=1 \dots n$ ) แต่ละตัวจะได้รับ Input signal  $X_i$  และทำการ Broadcast signal ที่ได้รับ ไปยังทุก unit ในชั้นที่อยู่สูงกว่า (Hidden layer)

**Step 4** แต่ละ Hidden unit ( $Z_j, j=1 \dots p$ )

$$z_{in_j} = V_{oj} + \sum X_i V_{ij}$$

จากนั้นนำค่าที่ได้มาเข้า Activation function จะได้

$$Z_j = f(z_{in_j})$$

จากนั้นจะส่งค่าไปยังชั้นที่อยู่สูงกว่า (Output layer)

**Step 5** แต่ละ Output unit ( $Y_k, k=1 \dots m$ )

$$y_{in_k} = W_{ok} + \sum Z_j W_{jk}$$

จากนั้นนำค่าที่ได้มาเข้า Activation function จะได้

$$Y_k = f(y_{in_k})$$

### Backpropagation of Error

**Step 6** แต่ละตัวของ Output unit ( $Y_k, k=1 \dots m$ ) จะนำค่า Target pattern มาเปรียบเทียบกับ แล้วทำการหาค่า Error

$$\delta_k = (t_k - Y_k) f'(y_{in_k})$$

คำนวณค่าของความถูกต้องของ Weight (Weight correction) เพื่อใช้ในการปรับปรุงค่าของ  $W_{jk}$

$$\Delta W_{ok} = \alpha \delta_k Z_j$$

คำนวณค่า Bias correction สำหรับใช้ในการปรับปรุงค่าของ  $W_{ok}$  จากนั้นจะส่งค่า  $\delta_k$  ไปยัง unit ที่อยู่ในชั้น Layer ที่ต่ำกว่า (Hidden layer)

**Step 7** แต่ละตัวของ Hidden unit ( $Z_j, j=1 \dots p$ ) รวมค่าของ Delta input (ค่าที่ถูกส่งมาโดยชั้นที่สูงกว่า)

$$\delta_{in_j} = \sum \delta_k W_{jk}$$

นำค่าของ Delta input มาคูณด้วยค่า Differential activation function ของ  $Z_{in_j}$  เพื่อนำค่ามาหาค่าของ Error information term

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณค่า Weight correction term ไว้ใช้ในการปรับปรุ่ค่าของ  $V_{ij}$  ต่อไป

$$\Delta V_{ij} = \alpha \delta_j x_i$$

จากนั้นคำนวณค่า Bias correction term สำหรับในการปรับปรุ่ค่าของ  $V_{0j}$  ต่อไป

$$\Delta V_{0j} = \alpha \delta_j$$

### Update weight and bias

**Step 8** แต่ละ Output unit ( $Y_k, k=1 \dots m$ ) ปรับปรุ่ค่าของ Bias และ Weight ( $j=0 \dots p$ )

$$W_{jk \text{ (new)}} = W_{jk \text{ (old)}} + \Delta W_{jk}$$

แต่ละ Hidden unit ( $Z_j, j=1 \dots p$ ) ปรับปรุ่ค่าของ Bias และ Weight ( $i=0 \dots n$ )

$$V_{ij \text{ (new)}} = V_{ij \text{ (old)}} + \Delta V_{ij}$$

**Step 9** ทดสอบ Stopping condition



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การเรียนรู้ Backpropagation แบบสามารถปรับ Hidden Unit ได้

ในตามทฤษฎีของการเรียนรู้แบบ Backpropagation นั้นเป็นการเรียนรู้ที่ได้รับความนิยมมาก เนื่องจากได้ผลเร็วกว่าวิธีอื่น ๆ เพราะนำค่า Error ที่ได้จากการ Train มา คำนวณจริง ๆ ซึ่งค่าจะเปรียบเทียบกับ Genetic Algorithm ซึ่งใช้เพื่อหาค่าของ Weight ที่เหมาะสมแล้วนั้น ทั้งรวดเร็วและไวกว่ามากมายนัก ซึ่ง Genetic Algorithm ยังไม่ Guarantee ที่จะสามารถหาค่าได้จริง ๆ เนื่องจากการทำงานแบบ Random ส่วนการเรียนรู้แบบ Backpropagation จะ Guarantee ที่จะหาค่าได้อย่างแน่นอน

ในการหาค่าที่เหมาะสมสำหรับ Weight ในการ Train ของ Data ชุดหนึ่ง ๆ องค์ประกอบที่มีความสำคัญมากองค์ประกอบหนึ่งก็เห็นจะเป็นจำนวนของ Hidden Unit เนื่องจากการกำหนดจำนวน Hidden Unit ที่เหมาะสมสามารถจะลดเวลาในการ Train ได้อย่างมากมายนัก ซึ่งแต่ละวิธีการของการคำนวณ Hidden Unit ไม่ได้ระบุว่า Data ชุดนี้จะต้องมี Hidden Unit เท่านี้ แต่จะคำนวณในลักษณะที่ว่าจำนวน Input เท่านี้ต้องการ Output เท่านี้ จะควรมี Hidden Unit เท่าใด (ซึ่งในกรณีนี้เราจะคิดเฉพาะในกรณีที่มี Layer ทั้งหมด 3 Layer คือ Input Layer, Hidden Layer, Output Layer โดยจะมีจำนวน Hidden Layer เพียง 1 Layer เท่านั้น)

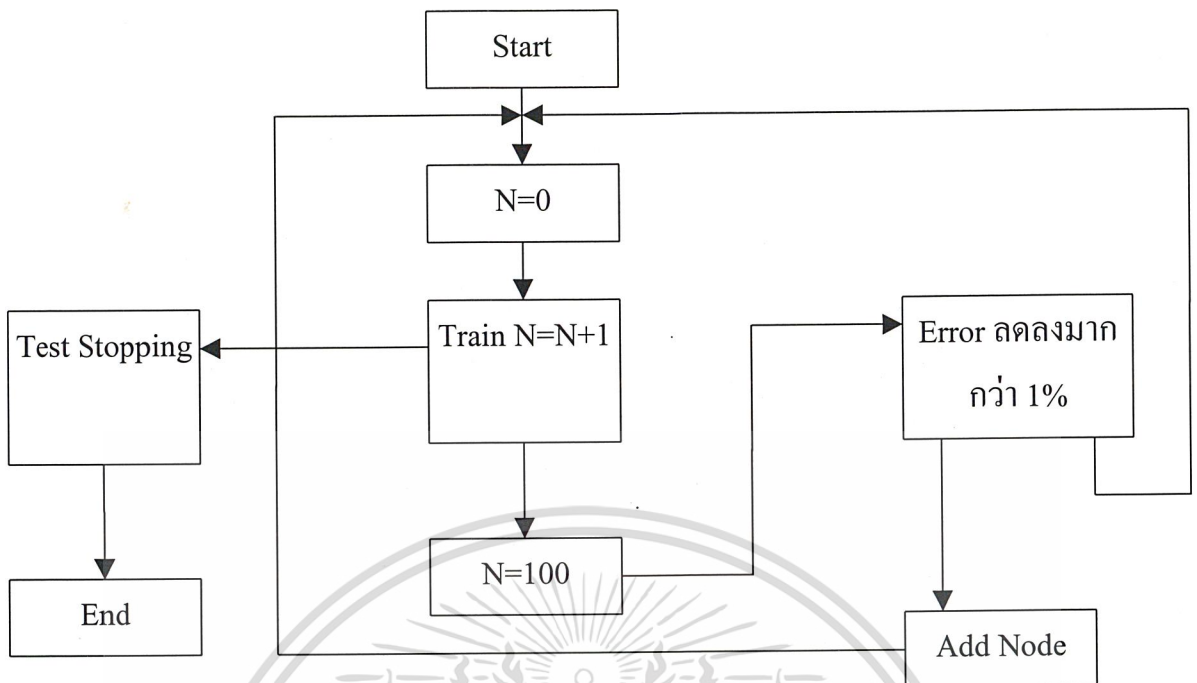
ฉะนั้นในการพัฒนาความสามารถในการ Train เพื่อหาค่าของ Hidden Unit ที่เหมาะสม จึงเป็นความสำคัญอย่างยิ่งเพื่อจะลดเวลาในการ Train Data Pattern ที่ต้องการ

#### 3.1 Algorithm

ในระหว่างการ Train N รอบ เราจะดูว่าอัตราการลดลงของ Error นั้นเป็นอย่างไร ถ้าอยู่ในระหว่างที่เราพอใจเราก็จะ Train ต่ออีก N รอบ แต่ตัวต่ำกว่าที่ต้องการก็จะมีกระบวนการ Add Node เข้าไป

ในที่นี้เราจะกำหนดให้ Train ทุก ๆ 100 รอบจะดูว่า อัตราการลดลงของค่า Error ลดลงถึง 1% หรือเปล่า ถ้าไม่ก็จะมีกระบวนการ Add Hidden Unit 1 Unit ลงใน Hidden Layer ดังแสดงในรูปภาพที่ 3.1.1 เป็นการแสดงกระบวนการการเรียนรู้แบบสามารถปรับค่าของจำนวน Hidden unit ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 3.1.1 ภาพแสดงการทำงานของ การ Train Backpropagation แบบ เพิ่ม Hidden Unit

อธิบายภาพที่ 3.1.1 ได้ตามการทำงานข้างล่างนี้

Step 1 Start

Step 2  $N=0$  Feed for Ward () Error 1= Average Error While (Not Stop)

Step 3 Train Data Pattern

Check stopping if

Accept Error > Average Error

Then Stop

$N=N+1$

It  $N=100$  Then Error 2=Average Error

If  $\text{Error 1} - \text{Error 2} < 0.01 * \text{Error 1}$

Then Add Node ()

$N=0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

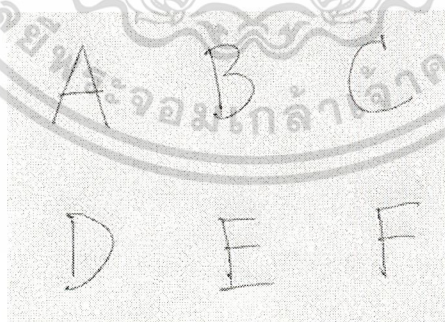
จากขั้นตอนทั้งหมดข้างต้นจะเห็นได้ว่าการทำงานของ Algorithm จะมีการ Train Data เป็นรอบ ๆ ซึ่งจะใช้ Data ที่ต้องการ Train จริง ๆ ซึ่งแต่ละรอบของการ Train จะตรวจดูว่าอัตราการลดลง Error ลดลงในอัตราที่น่าพอใจหรือไม่ ถ้าไม่ก็จะมีการ Add Node เข้าไปเพิ่มเพื่อจะต้องการเพิ่มอัตราการลดลงของ Error เมื่อได้ค่าที่เหมาะสมกับความต้องการแล้วก็จะหยุดการทำงานเหมือน Backpropagation ทุกอย่างยกเว้นเพียงแต่เพิ่มกระบวนการ Check จำนวนรับกับการวัดอัตราการลดลงของ Error และกระบวนการ Add Hidden Unit เท่านั้น

ในตอนเริ่มต้นของการทำงานเราจะระบุให้จำนวน Hidden Unit มีค่าเท่ากับ 1 ซึ่งเป็นตัวที่น้อยที่สุด จากนั้นเมื่อต้องการจะเพิ่ม Hidden Unit ในแต่ละรอบจะเพิ่มเพียง 1 ตัวเท่านั้น ซึ่ง Weight ของการเชื่อมต่อระหว่าง Hidden Unit ตัวนี้กับ Input Layer และ Output Layer จะกำหนดให้มี Weight การเชื่อมต่อเป็น 0 ที่กำหนดเช่นนี้เพื่อที่จะไม่ให้มีผลกระทบกับค่าของ Average Error ที่ได้ Train ผ่าน ๆ มา ซึ่งค่าของ Average Error ก็จะลดลงเรื่อย ๆ

- การ Add Node ก็เชื่อมต่อของ Node กับ Input Layer และ Output Layer จะมีค่าของ Weight = 0

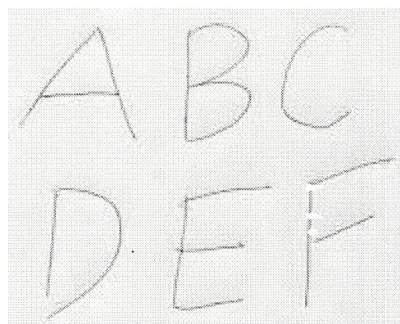
### 3.2 ขั้นตอนการทดลอง

เราจะนำตัวอักษร A-F มาทดลองโดยการใช้ Backpropagation แบบนี้กับ Algorithm ของการเปลี่ยนแปลงค่า Hidden Unit เพื่อเปรียบเทียบกับ การ Train แบบธรรมดาโดยใช้เพียง Backpropagation ธรรมดาเท่านั้น โดยภาพที่เราจะใช้ที่นี่จะเป็นภาพในรูปภาพที่ 3.2.1, รูปภาพที่ 3.2.2 และ รูปภาพที่ 3.2.3 ตามลำดับ



รูปภาพที่ 3.2.1 แสดงตัวอักษร A-F ชุดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 3.2.2 แสดงตัวอักษร A-F ชุดที่ 2



รูปภาพที่ 3.2.3 แสดงตัวอักษร A-F ชุดที่ 3

เราจะทำการ Train โดยใช้ ภาพตัวอักษร A-F ทั้ง 3 ชุดซึ่งแต่ละในการ Train เราจะใช้จะ Neural Network ที่มีจำนวน Hidden unit แตกต่างกัน (1,2,3,4,5) แล้วนับจำนวนรอบของการ Train ดังใน

จำนวน Hidden unit	จำนวนรอบของการ Train
1	50
2	30
3	12
4	18
5	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าเรานำมาเพิ่มกระบวนการ Train แบบสามารถปรับเปลี่ยนจำนวน Hidden unit ได้ เราจะได้ผลตามตารางข้างล่าง

แบบปรับเปลี่ยนจำนวน Hidden unit	จำนวนรอบ
1	4
2	6
3	5

จากผลการทดลองจะเห็นได้ว่า ผลของ Train แบบปรับเปลี่ยนจำนวน Hidden unit ได้ นั้นจะรวดเร็วกว่าการ Train แบบธรรมดาบางกรณี ซึ่งจำนวนรอบทั้งหมดของการ Train แบบปรับเปลี่ยนจำนวน Hidden unit ได้ นั้นจะมีรอบการ Train ทั้งหมด 15 รอบ ซึ่งจะน้อยกว่าการ Train แบบธรรมดาแบบที่มีจำนวน Hidden unit 1 2 และ 4 ตัว แต่จะมากกว่าในกรณี 3 และ 5 ตัว ซึ่งถ้าเราสามารถเลือกถูกว่าจำนวน Hidden unit เป็นเท่าไรก็จะดีแต่เราไม่สามารถเลือกจำนวนที่ถูกต้องได้ อีกประเด็นหนึ่งที่สำคัญในการ Train แบบ ปรับค่า Hidden unit ได้ นี้จะได้ค่าของ Hidden unit ที่เหมาะสมและมีค่าที่น้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การจัดการข้อมูลก่อนป้อนเข้านิรอลเน็ตเวิร์ค

การทำ Pre-Processing หรือ การจัดการข้อมูลก่อนป้อนเข้านิรอลเน็ตเวิร์คเป็นการจัดรูปแบบของข้อมูลทั้งหมดให้อยู่ในรูปแบบ (Format) เดียวกัน เพื่อให้สามารถวิเคราะห์และทำงานในการแบ่งแยกตัวอักษรให้มีความละเอียดถูกต้องและเพื่อเป็นการสะดวกในการแบ่งแยกความแตกต่างของตัวอักษรอีกด้วย

กระบวนการทำกับข้อมูลก่อนทำการประมวลผล (Pre - Processing ) ประกอบด้วยขั้นตอนต่าง ๆ กันตามงานวิจัยนี้จะแบ่งออกเป็น 3 ขั้นตอน

1. Segmentation เป็นการแบ่งแยกตัวอักษรออกจากกันเพื่อสะดวกในการทำงาน โดยจะทำงานการแยกตัวอักษรทีละตัว
2. การทำให้เส้นบาง เนื่องจากตัวอักษรมีความหนาและบางไม่เท่ากันฉะนั้นเราจึงต้องทำให้มีขนาดความหนาเท่ากัน
3. การเปลี่ยนขนาด เนื่องจากตัวอักษรมีขนาดไม่เท่ากัน

สาเหตุที่ต้องมีการทำ pre-processing

- ขนาดตัวอักษรมีขนาดไม่เท่ากัน
- ขนาดความหนาของตัวอักษรเดียวกันบางครั้งไม่เท่ากัน
- ข้อมูลที่จะนำมาใช้ไม่มีการแบ่งแยกออกมาเป็นตัว ๆ อย่างชัดเจน
- ต้องการการทำงานโดยมีประสิทธิภาพในการประมวลผลเร็ว

จากสาเหตุดังกล่าวจึงจำเป็นต้องมีการทำ Pre-processing

#### 4.1 Segmentation

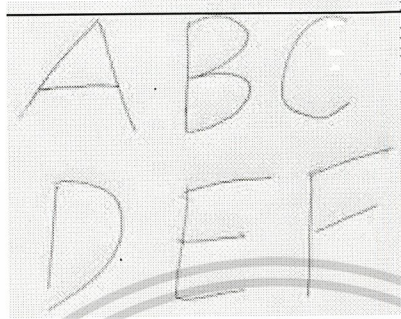
เป็นขั้นตอนการแบ่งแยกตัวอักษรออกเป็นตัว ๆ ออกจากชุดตัวอักษรอื่น ๆ ที่อยู่ในรูปภาพเดียวกันเพื่อนำตัวอักษรนั้นออกมาทำกระบวนการอื่น ๆ ต่อไป

กระบวนการทำงานของการแบ่งแยกตัวอักษร

- 1) จะเริ่มจากหาเส้นขอบด้านบนของตัวอักษร
- 2) จากเส้นขอบบนของตัวอักษรจะหาเส้นขอบด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

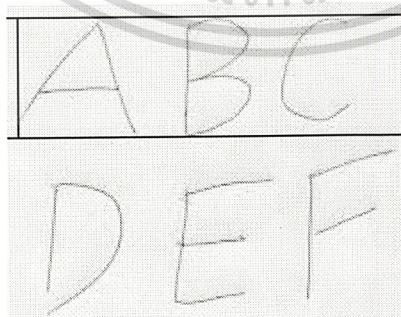
- 3) ระหว่างเส้นขอบด้านบนและด้านล่างหาเส้นขอบที่ชิดริมด้านซ้ายของตัวอักษร
- 4) ระหว่างเส้นขอบด้านบนและด้านล่าง โดยเริ่มต้นจากเส้นชิดขอบด้านซ้ายหาเส้นขอบด้านขวาของตัวอักษร
- 5) ทำข้อ 1) และ 2) ใหม่อีกรอบ



รูปภาพที่ 4.1.1 แสดงการหาเส้นขอบด้านบนของตัวอักษร โดยเริ่มจากเส้นกรอบด้านบนของภาพ



รูปภาพที่ 4.1.2 แสดงการหาเส้นขอบด้านล่างของตัวอักษร โดยเริ่มจากเส้นขอบด้านบนของตัวอักษร



รูปภาพที่ 4.1.3 แสดงการหาเส้นขอบด้านซ้ายของตัวอักษร โดยเริ่มจากเส้นกรอบด้านซ้ายของเอกสาร เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สเกลเลตันที่ได้จะยังคงรักษาค่า โทโปโลยี ของวัตถุไว้และแสดงข้อมูลรูปทรงของรูปแบบไว้ สเกลเลตันที่ได้จะต้องมีความกว้างของจุดภาพเพียง หนึ่งจุดภาพและเป็นเส้นแกนกลางของวัตถุในภาพ วิธีที่แตกต่างกันจะให้สเกลเลตันที่แตกต่างกันซึ่งตามปกติขั้นตอนการทำวิธีลายเส้นให้บางจะแบ่งออกเป็น 2 แบบ คือ แบบลำดับ (Sequential) และแบบขนาน (Parallel) ซึ่งมักเชื่อกันว่าขั้นตอนวิธีแบบตามลำดับจะทำงานได้เร็วกว่าขั้นตอนวิธีแบบขนาน เมื่อขั้นตอนวิธีทั้ง 2 แบบถูกใช้งานในเครื่องคอมพิวเตอร์แบบ อนุกรม (Serial Computer) และขั้นตอนวิธีแบบขนานจะทำงานได้อย่างมีประสิทธิภาพสูงสุดเมื่อถูกใช้งานบนเครื่องคอมพิวเตอร์แบบขนาน (Parallel Computer) หรือเครื่องคอมพิวเตอร์แบบ Pipeline Computer

สเกลเลตันดิจิทัล (Digital Skeleton) จะเกิดขึ้นภายหลังใช้ขั้นตอนวิธีการทำลายเส้นให้บางซึ่งมักจะใช้แสดงวัตถุในภาพขาวดำดิจิทัล (Binary Digital Image) ของการวิเคราะห์รูปทรงและการจำแนกประเภท (Classification) มีขั้นตอนการทำลายเส้นให้บางเป็นจำนวนมากที่ได้ถูกสร้างขึ้นโดยทั่วไป ซึ่งในการทำลายเส้นให้บางในงานวิจัยนี้ได้ทำการศึกษาละเอียดของขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีที่แตกต่างจำนวน 2 ขั้นตอนซึ่งมีขั้นตอนวิธีที่ 1 เป็นขั้นตอนวิธีของ Zhang and Suen , ขั้นตอนวิธีที่ 2 เป็นขั้นตอนวิธีของ A.DATTA and S.K. PARUI

#### 4.2.1 ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ Zhang and Suen

ในขั้นตอนวิธีทำลายเส้นให้บางตามวิธีของ Zhang and Suen ที่นำมาแสดงนี้กระทำกับภาพขาวดำโดยให้ 1 แสดงแทนจุดภาพของวัตถุในภาพ และ 0 แสดงแทนจุดภาพของพื้นฉากในภาพ ขั้นตอนวิธีการทำลายเส้นให้บางวิธีนี้ประกอบด้วย 2 ขั้นตอนที่กระทำกับจุด Contour ของเนื้อวัตถุในภาพ โดยจุด Contour เป็นจุดที่มีค่าเป็น 1 และมีจุดรอบข้างของ  $x_0$  อย่างน้อย 1 จุดที่มีค่าเป็น 0

ขั้นตอนที่ 1 จุด Contour ของ  $x_0$  จะลบมีเงื่อนไขดังนี้

- (1) น้อยกว่าหรือเท่ากับ  $N(x_0)$  น้อยกว่าหรือเท่ากับ 6
- (2)  $S(x_0)$  เท่ากับ 1
- (3)  $x_1 * x_3 * x_7 = 0$
- (4)  $x_1 * x_5 * x_7 = 0$

โดยที่  $N(x_0)$  เป็นจำนวนจุดรอบข้างของ  $x_0$  ที่ไม่เป็น 0 นั่นคือ

$$N(x_0) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$$

$S(x_0)$  เป็นจำนวนจุดที่มีการเปลี่ยนแปลงจาก 0 เป็น 1 ตามลำดับของ  $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$  ดังตัวอย่าง  $N(x_0) = 4$  และ  $S(x_0) = 3$  ดังข้างล่าง

$$0 \ 0 \ 1$$

$$1 \ x_0 \ 0$$

$$1 \ 0 \ 1$$

แสดงเงื่อนไข (1) และ (2) ในกรณีนี้

$$N(x_0) = 4 \text{ และ } S(x_0) = 3$$

โดยเงื่อนไขของ (3) เป็นการตรวจสอบผลคูณของ  $x_1 * x_3 * x_7 = 0$  และ เงื่อนไขของ (4) การตรวจสอบผลคูณของ  $x_1 * x_5 * x_7 = 0$  ซึ่งเงื่อนไขของ (3) และ (4) จะเป็นจริง เมื่อมีค่าใดค่าหนึ่งตามนี้คือ  $x_1 = 0$  หรือ  $x_7 = 0$  หรือ ( $x_3 = 0$  และ  $x_5 = 0$ )

ขั้นตอนที่ 2 เงื่อนไข (1) และ (2) ยังคงเหมือนเดิมแต่เงื่อนไข (3) และ (4) ถูกเปลี่ยนเป็น

(1) น้อยกว่าหรือเท่ากับ  $N(x_0)$  น้อยกว่าหรือเท่ากับ 6

(2)  $S(x_0)$  เท่ากับ 1

(3)  $x_1 * x_3 * x_5 = 0$

(4)  $x_3 * x_5 * x_7 = 0$

เงื่อนไข (1) และ (2) จะมีวิธีในการหาค่าเหมือนกับ เงื่อนไข (1) และ (2) ที่อยู่ในขั้นตอนที่ 1 โดยเงื่อนไขขั้นตอนที่ 2 (3) เป็นการตรวจสอบผลคูณของ  $x_1 * x_3 * x_5 = 0$  และเงื่อนไขขั้นตอนที่ 2 (4) เป็นการตรวจสอบผลคูณของ  $x_3 * x_5 * x_7 = 0$  ซึ่งเงื่อนไขของขั้นตอนที่ 2 (3) (4) จะเป็นจริงเมื่อมีค่าใดค่าหนึ่งตามนี้คือ  $x_3 = 0$  หรือ  $x_5 = 0$  หรือ ( $x_1 = 0$  และ  $x_7 = 0$ )

โดยจะมีขั้นตอนการทำงานอย่างคร่าว ๆ คือ

ในขั้นตอนที่ 1 จะประมวลผลกับจุดขอบภาพ(Border Point) ของเนื้อวัตถุในภาพขาวดำภายใต้การพิจารณา ถ้าไม่ตรงกับเงื่อนไขอันใดอันหนึ่งของ (1) ถึง (4) จุดภาพที่พิจารณาอยู่ จะไม่ถูกลบ แต่ถ้าถูกต้องตามเงื่อนไข (1)-(4) จุดภาพที่พิจารณานั้นจะถูกลบ แต่จุดภาพที่ถูกพิจารณาทั้งหมดจะไม่ถูกลบจนกระทั่งจุดขอบภาพของทั้งภาพถูกประมวลผล โดยจะทำเครื่องหมาย (flag) ของจุดที่จะไม่ถูกลบไว้ก่อน ซึ่งจะป้องกันการเปลี่ยนแปลงโครงสร้างของข้อมูลระหว่างการดำเนินการของขั้นตอนวิธีการทำลายเส้นในบางตามวิธีของ Zhang and Suen ภายหลังจากขั้นตอนที่ 1 ได้ถูกประมวลผลกับทุกจุดขอบภาพแล้ว จุดที่ถูกทำเครื่องหมายไว้ จะถูกลบ(กล่าวคือเปลี่ยนค่าเป็น 0) จากนั้นขั้นตอนที่ / จะประมวลผลกับภาพที่ได้จากการประมวลผลในขั้นตอนที่ 1 แล้วด้วยวิธีการที่เหมือนกับขั้นตอนที่ 1

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยพื้นฐานของขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ Zhang and Suen นี้ ดังเกิดได้ว่า ประกอบด้วยขั้นตอนอย่างคร่าว ๆ ดังนี้

1. ทำตามขั้นตอนที่ 1
2. ลบจุดภาพ
3. ทำตามขั้นตอนที่ 2 กับจุดขอบภาพที่ยังคงเหลือ
4. ลบจุดขอบภาพ

ซึ่งขั้นตอนทั้ง 4 ข้อนี้อาจถูกกระทำซ้ำไปเรื่อย ๆ จนกระทั่งไม่มีจุดภาพใด ๆ จะถูกลบอีก ซึ่งจะเป็นช่วงเวลาสิ้นสุดการทำงานของขั้นตอนการทำลายเส้นให้บางตามวิธีของ Zhang and Suen ที่เราจะได้สเกลเลตันของเนื้อวัตถุในภาพ

#### 4.2.2 ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI

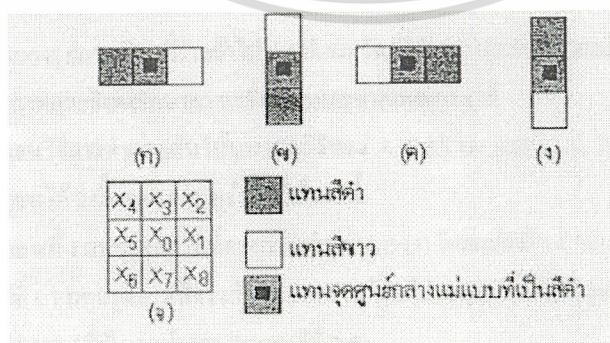
ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI จะทำการลบจุดภาพชั้นนอกในกรอบต่าง ๆ และผลลัพธ์ที่ได้ในแต่ละรอบจะถูกนำไปใช้กับรอบถัดไปตามลำดับ ซึ่งจุดของภาพจะไม่ถูกลบ ถ้ามีเงื่อนไขตามนี้ คือ

1. ถ้าถูกลบ จะไปเปลี่ยนแปลงค่าความเชื่อมต่อ (Connectivity)
2. ถ้าถูกลบ จะไปทำให้ขาของสเกลเลตันสั้นขึ้น

และจะไม่ทำการลบจุดภาพ ที่มีเงื่อนไขตามนี้ คือ

1. เป็นส่วนที่เรียกว่า จุดภาพวิกฤต
2. เป็นส่วนที่เรียกว่า จุดปลาย

ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI นี้ จะใช้แม่แบบขนาดเมตริกซ์  $1 \times 3$  จำนวน 2 อัน แม่แบบขนาดเมตริกซ์  $3 \times 1$  จำนวน 2 อัน และการใช้วินโดว์ขนาดเมตริกซ์  $3 \times 3$  จำนวน 1 อัน ดังในภาพที่ 4.2.2.1



รูปภาพ 4.2.2.1 แสดงแม่แบบ (ก) ถึง (ง) ที่ใช้ในการทำลายเส้นให้บาง ทั้ง 4 แบบและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาเวกเตอร์  $3 \times 3$  ซึ่งอยู่รอบจุดภาพ  $x_0$  จะเรียกแทนด้วย  $W$  โดยให้  $S$  เป็นเซตของ 8 จุดภาพในเวกเตอร์ ยกเว้นจุด  $x_0$  นั่นคือ

ถ้าเราสนใจภายในเวกเตอร์  $W$  เท่านั้น เราสามารถหาจุดวิกฤติได้จากทฤษฎีที่ 1 และจุดปลายได้จากทฤษฎีที่ 2

ทฤษฎีที่ 1 ถ้า  $x_0$  ตรงกับแม่แบบอันใดอันหนึ่งของทั้ง 4 แบบจาก ตรงกับแม่แบบอันใดอันหนึ่งของทั้ง 4 แบบดังในรูปแสดงแม่แบบ (ก) ถึง (ง) แล้วจุดวิกฤติจะมีค่าตามนี้

1.  $x_0$  เป็นจุดวิกฤติ ถ้า  $(x_3 = 0 \text{ และ } x_2 = 1)$  หรือ  $(x_7 = 0 \text{ และ } x_8 = 1)$
2.  $x_0$  เป็นจุดวิกฤติ ถ้า  $(x_5 = 0 \text{ และ } x_4 = 1)$  หรือ  $(x_1 = 0 \text{ และ } x_2 = 1)$
3.  $x_0$  เป็นจุดวิกฤติ ถ้า  $(x_3 = 0 \text{ และ } x_4 = 1)$  หรือ  $(x_7 = 0 \text{ และ } x_6 = 1)$
4.  $x_0$  เป็นจุดวิกฤติ ถ้า  $(x_5 = 0 \text{ และ } x_6 = 1)$  หรือ  $(x_1 = 0 \text{ และ } x_8 = 1)$

ทฤษฎีที่ 2 ถ้า  $x_0$  ตรงกับแม่แบบอันใดอันหนึ่งของทั้ง 4 แบบ แล้วจุดปลายจะมีค่าตามนี้

1.  $x_0$  เป็นจุดปลาย ถ้า  $x_2, x_3, x_4, x_6, x_7, x_8$  เป็น 0 (มีค่าเท่ากับ  $x_2+x_3+x_4+x_6+x_7+x_8=0$ )
2.  $x_0$  เป็นจุดปลาย ถ้า  $x_1, x_2, x_4, x_5, x_6, x_8$  เป็น 0 (มีค่าเท่ากับ  $x_1+x_2+x_4+x_5+x_6+x_8=0$ )
3.  $x_0$  เป็นจุดปลาย ถ้า  $x_2, x_3, x_4, x_6, x_7, x_8$  เป็น 0 (มีค่าเท่ากับ  $x_2+x_3+x_4+x_6+x_7+x_8=0$ )
4.  $x_0$  เป็นจุดปลาย ถ้า  $x_1, x_2, x_4, x_5, x_6, x_8$  เป็น 0 (มีค่าเท่ากับ  $x_1+x_2+x_4+x_5+x_6+x_8=0$ )

ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI ตามธรรมชาติ จะมีการกระทำซ้ำ (Iteration) โดยจะดำเนินการกำจัดจุดขอบภาพในแต่ละรอบการกระทำซ้ำทั้ง 4 รอบย่อย ในแต่ละรอบย่อยจะกำจัดจุดภาพที่ตรงกับแต่ละแม่แบบ กล่าวคือ แม่แบบ ก ถึง ง จะถูกใช้ในการกำจัดจุดขอบภาพ เวกเตอร์ขนาดเมตริกซ์  $3 \times 3$  จะใช้ในการตรวจสอบการลบจุดภาพนั้นเพื่อให้ยังรักษาความเชื่อมต่อและขาสเกลเลตันจะไม่หลุดสิ้น ภายหลังจากการกระทำซ้ำ (จะขึ้นอยู่กับความหนาของรูปแบบวัตถุที่รับเข้าไป) รูปแบบวัตถุที่รับเข้าไปจะเข้าสู่สเกลเลตันที่มีความหนาอยู่เพียง 1 จุดภาพและยังคงรักษาความเชื่อมต่อและขาสเกลเลตันไว้

ขั้นตอนวิธีการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI สามารถแบ่งออกเป็นขั้นตอนได้ 2 ขั้นตอนอย่างคร่าว ๆ ได้ตามนี้

ขั้นตอนที่ 1 ลบทุกจุดภาพที่ตรงกับแม่แบบ 1 ถึง 4 โดยแบ่งเป็นรอบย่อย ๆ คือ

รอบที่ 1 ลบทุกจุดภาพที่ตรงกับแม่แบบ 1 โดยไม่ลบจุดภาพที่เป็นจุดวิกฤติตามทฤษฎีที่ 1 ในแม่แบบ 1 และไม่ลบจุดภาพที่เป็นจุดปลายตามทฤษฎีที่ 2 ในแม่แบบ 1

รอบที่ 2 ลบทุกจุดภาพที่ตรงกับแม่แบบ 2 โดยไม่ลบจุดภาพที่เป็นจุดวิกฤติตามทฤษฎีที่

1 ในแม่แบบ 2 และไม่ลบจุดภาพที่เป็นจุดปลายตามทฤษฎีที่ 2 ในแม่แบบ 2

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอบที่ 3 ลบทุกจุดภาพที่ตรงกับแม่แบบ 3 โดยไม่ลบจุดภาพที่เป็นจุดวิกฤติตามทฤษฎีที่ 1 ในแม่แบบ 3 และไม่ลบจุดภาพที่เป็นจุดปลายตามทฤษฎีที่ 2 ในแม่แบบ 3

รอบที่ 4 ลบทุกจุดภาพที่ตรงกับแม่แบบ 4 โดยไม่ลบจุดภาพที่เป็นจุดวิกฤติตามทฤษฎีที่ 4 ในแม่แบบ 1 และไม่ลบจุดภาพที่เป็นจุดปลายตามทฤษฎีที่ 2 ในแม่แบบ 4

ขั้นตอนที่ 2 กลับไปทำขั้นตอนที่ 1 ไปเรื่อย ๆ จนกระทั่งไม่มีการเปลี่ยนแปลงใด ๆ เกิดขึ้นในภาพ กล่าวคือไม่มีการลบจุดภาพ

จากขั้นตอนวิธีการทำลายเส้นในบางที่กล่าวมาแล้วทั้ง 2 ขั้นตอนในวิธีของ Zhang and Suen และวิธีของ A.DATTA and S.K.PARUI มีขั้นตอนที่ง่ายไม่ซับซ้อน อีกทั้งในขั้นตอนของการทำลายเส้นในบางตามแบบของ A.DATTA and S.K.PARUI ใช้แบบแม่ในการทำลายเส้นให้บางจำนวนน้อย ซึ่งทำให้เราคาดคะเนได้ว่า การทำลายเส้นให้บางทั้ง 2 วิธีที่กล่าวมา ใช้เวลาในการประมวลผลน้อย แต่เมื่อพิจารณาจากภาพลายเส้นที่ได้จากการทำลายเส้นในบางตามวิธีของ Zhang and Suen พบว่า มีบางส่วนของลายเส้นที่หนา และเมื่อพิจารณาจากภาพลายเส้นที่ได้จากการทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI พบว่า จะให้ลายเส้นที่บาง จึงทำให้เราทราบว่า การทำลายเส้นให้บางตามวิธีของ A.DATTA and S.K.PARUI นั้นให้ผลที่ดีกว่าการทำลายเส้นให้บางตามวิธีของ Zhang and Suen

#### 4.3 การทำให้ตัวอักษรอยู่ในขนาดมาตรฐาน

การทำให้ตัวอักษรอยู่ในขนาดมาตรฐาน คือ การที่ทำให้ทุกตัวอักษรมีขนาดเดียวกันโดยใช้หลักการเทียบสัดส่วน

ในการทำให้ตัวอักษรมีขนาดมาตรฐานมีขั้นตอนดังต่อไปนี้

1. การกำหนดขนาดของภาพกว้าง  $M$  ยาว  $N$  โดยแต่ละจุดในภาพจะมีฟังก์ชัน  $f(x,y)$  เป็นฟังก์ชันแบบ Binary ระบุว่าตัวอักษรที่จุด  $(x,y)$  ในภาพเป็นส่วนหนึ่งของตัวอักษรหรือไม่ถ้าเป็นจะมีค่าของฟังก์ชัน เป็น 1 แต่ถ้าไม่อยู่ในตัวอักษรจะมีค่าเป็น 0
2. หาค่าของฟังก์ชันในแนวนอน  $H(i)$  ของแต่ละจุด  $i$  ในระนาบโดยที่

$$H(i) = \sum_{j=1}^N f(i, j) + \alpha_H$$

โดยที่  $i$  เป็นค่าระบุตำแหน่งของจุดในแนวนอน และ  $j$  เป็นค่าระบุตำแหน่งในแนวตั้งของจุดในระนาบ  $N$  คือ จุดที่มีค่ามากที่สุดที่แนวตั้งของพื้นที่ และ  $\alpha_H$  เป็นค่าคงที่ซึ่งในทฤษฎีกำหนดให้เป็น 0 หรือ 2 แต่เราจะใช้ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาค่าของฟังก์ชันในแนวตั้ง  $V(j)$  ของแต่ละจุด  $j$  ในระนาบที่ให้ตัวอักษรโดย

$$V(j) = \sum_{i=1}^M f(i, j) + \alpha_v$$

โดย  $i$  เป็นค่าระบุตำแหน่งในแนวนอนของจุด และ  $j$  เป็นค่าระบุตำแหน่งในแนวตั้งของจุดในแนวนอนของพื้นที่ที่ตัวอักษร และ  $\alpha_v$  เป็นค่าคงที่ซึ่งในทฤษฎีกำหนดให้เป็น 0 หรือ 2 แต่เราใช้ 0

3. ทำการหาค่าตำแหน่งของจุดหลังจากที่ทำการ Normalize ขนาดแล้วโดยหาได้จากสูตรดังต่อไปนี้

$$x = (\sum_{i=1}^X H(i) * \text{ขนาดมาตรฐานตามแนวนอน}) / \sum_{i=1}^M H(i) \text{ โดยที่}$$

$x$  คือ ตำแหน่งตามแนวนอนของจุดหลังถูกแปลงให้อยู่ในขนาดมาตรฐาน

$X$  คือ ตำแหน่งตามแนวนอนของจุดก่อนแปลงให้อยู่ในขนาดมาตรฐาน

$M$  คือ จุดที่มีค่ามากที่สุดที่ในแนวนอนของพื้นที่ที่ตัวอักษร

$$y = (\sum_{j=1}^Y V(j) * \text{ขนาดมาตรฐานตามแนวตั้ง}) / \sum_{j=1}^N V(j) \text{ โดยที่}$$

$y$  คือ ตำแหน่งตามแนวตั้งของจุดหลังถูกแปลงให้อยู่ในขนาดมาตรฐาน

$Y$  คือ ตำแหน่งตามแนวตั้งของจุดก่อนถูกแปลงให้อยู่ในขนาดมาตรฐาน

$N$  คือ จุดที่มีค่ามากที่สุดที่ในแนวตั้งของพื้นที่ที่ตัวอักษร

## บทที่ 5

### แนะนำวิธีการใช้งานโปรแกรมตัวอย่างเบื้องต้น

ในบทนี้จะกล่าวถึงการใช้งานของโปรแกรมตัวอย่างที่ทำขึ้นมาเพื่อแนะนำการทำงานของ Neural Network เพื่อใช้ในงานการแบ่งแยกตัวอักษรตามที่ได้กล่าวมาถึงในบทก่อนหน้า ซึ่งจะเป็นการใช้งานอย่างละเอียดทุกขั้นตอนแบบ Step - to - Step

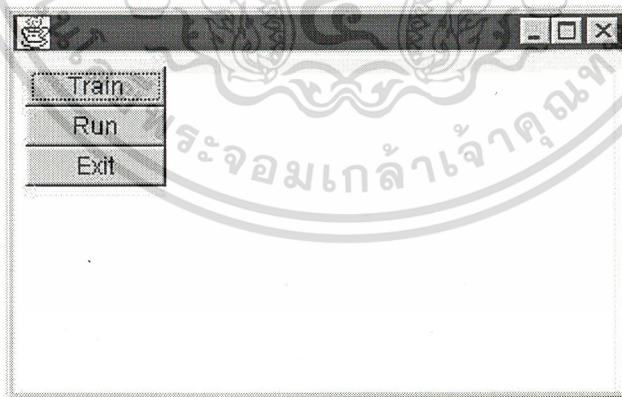
การใช้งานโปรแกรมตัวอย่างนี้ใช้งานได้ง่ายมากและไม่ขึ้นกับระบบปฏิบัติการอีกด้วยไม่ว่าจะใช้ระบบปฏิบัติการ Window 95 หรือ Window 98 หรือ X-Window ใน Linux ขอเพียงมีตัว Java Runtime Environment เพื่อใช้ในการ Run แบบ Interpreter ตอนทำงานเท่านั้น

โปรแกรมตัวอย่างนี้ประกอบด้วยการทำงานใน 2 รูปแบบ คือ

1. การทำงานในรูปแบบฝึกสอน (Training Mode)
2. การทำงานในรูปแบบการทำงานเพื่อแบ่งแยกตัวอักษร (Classification Mode)

#### 5.1 เริ่มแรกการใช้งานโปรแกรมตัวอย่าง

เมื่อเริ่มรันโปรแกรมตัวอย่างนี้จะปรากฏหน้าจอดังรูปข้างล่างนี้



รูปภาพ 5.1 ภาพแสดงรูปของการทำงานโปรแกรมตอนเริ่มต้น

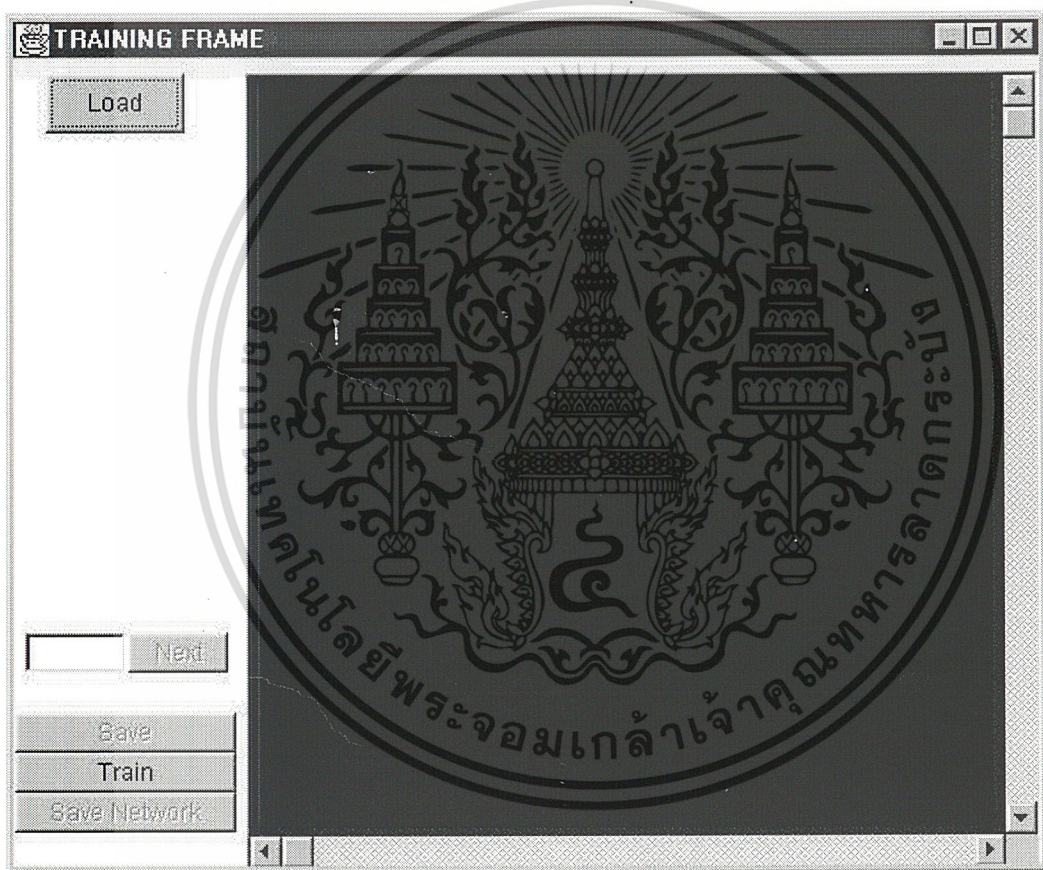
จากภาพ 5.1 จะประกอบไปด้วยปุ่ม 3 ปุ่ม คือ Train , Run , Exit

■ Train Button มีไว้สำหรับใน Mode Train จะ Click เมื่อต้องการที่จะ Train เมื่อ Train เสร็จแล้วก็จะได้ Neural Network ที่ต้องการเพื่อแบ่งแยกตัวอักษร

- Run Button มีไว้สำหรับใน Mode Run จะ Click เมื่อต้องการจะทดสอบค่าที่ได้จากการ Train
- Exit Button มีไว้สำหรับต้องการออกจากโปรแกรมตัวอย่าง

## 5.2 การทำงานในรูปแบบฝึกสอน

เมื่อ Click ใน Train Button ตามภาพตัวอย่างในรูป 5.1 ก็จะเข้ามาทำงานในรูปแบบการฝึกสอนจากภาพข้างล่างจะเป็นหน้าต่างของ User Interface ใน Mode การทำงานในรูปแบบการฝึกสอน



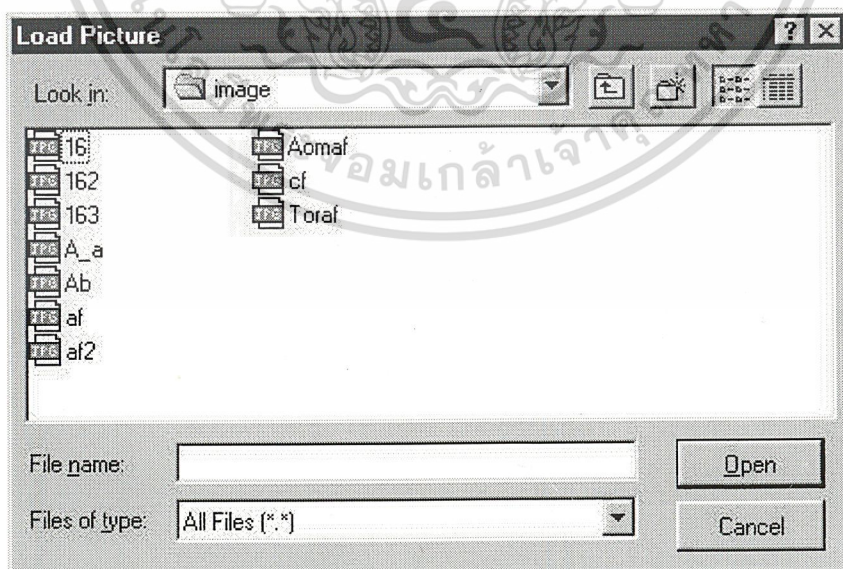
รูปภาพ 5.2 แสดงหน้าต่างของ User Interface ใน Mode Train

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

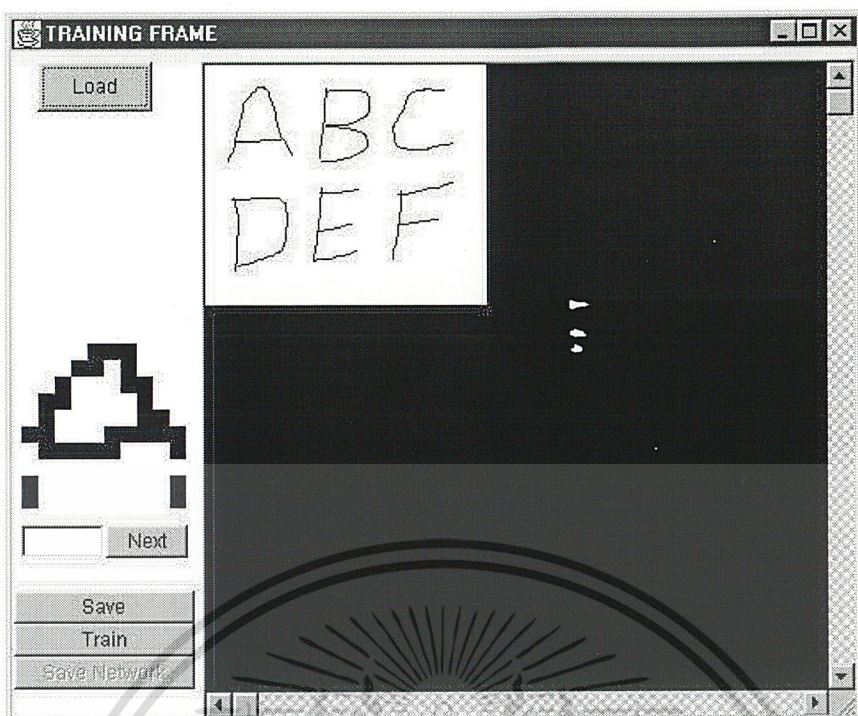
จากภาพรูปที่ 5.2 จะประกอบด้วยปุ่มทั้งหมด 5 ปุ่ม คือ

- Load Button เป็นปุ่มที่ใช้ในการโหลดภาพที่เป็นรูปแบบของภาพตระกูล Gif และ Jpg เมื่อนำมาเป็น Input ของ Neural Network เมื่อมีการกดปุ่มนี้จะได้หน้าต่างดังแสดงในรูปภาพที่ 5.2.1
- Next Button เป็นปุ่มที่ใช้ในการแสดงตัวอักษรที่จะเป็นตัว Input ตัวต่อไปมาแสดงให้ User ดูเพื่อป้อนข้อมูลบอกกับ Neural Network ได้อย่างถูกต้อง ซึ่งในการแสดงผลนั้นจะเป็นการแสดงผลที่จะเข้าไปใน Neural Network จริง ๆ
- Save Button เป็นการบันทึกข้อมูลของภาพที่ได้เมื่อป้อนบอก Neural Network แล้วไว้เป็น File เพื่อสามารถนำมาใช้ในการสอนใหม่อีกครั้งก็ได้
- Train Button ก่อนที่จะ Click ปุ่มนี้ได้จะต้องมีการโหลดข้อมูลที่ได้จาก File ที่บันทึกโดย Save Button ก่อน ปุ่มนี้จะเป็นการเริ่มต้นกระบวนการ Train ของ Neural Network
- Save Network เป็นการบันทึกข้อมูลของ Neural Network ที่ได้จากการ Train แล้ว

เมื่อมีการกดปุ่ม Load Button จะมีหน้าต่างขึ้นมาดังในรูปที่ 5.2.1 และเมื่อมีการเลือกภาพที่ต้องการแล้วจะแสดงผลของการเลือกภาพนั้นดังในรูปที่ 5.2.2

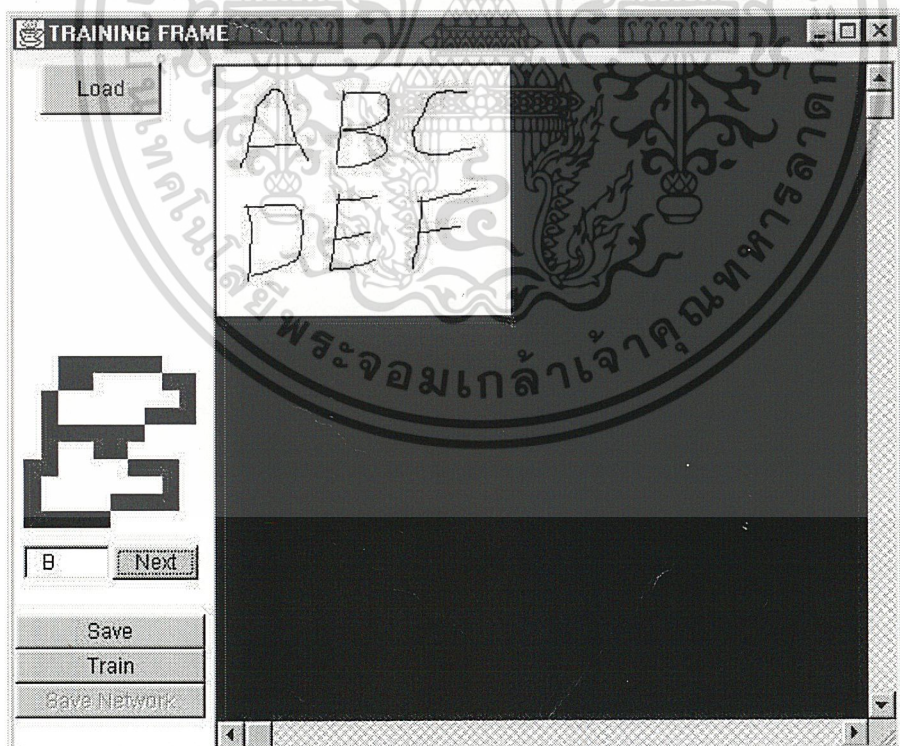


เอกสารนี้เป็นเอกสารที่สงวนไว้รูปภาพที่ 5.2.1 แสดงผลเมื่อมีการกดปุ่ม Load Button ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพที่ 5.2.2 แสดงผลหลังจากมีการเลือกภาพเสร็จ

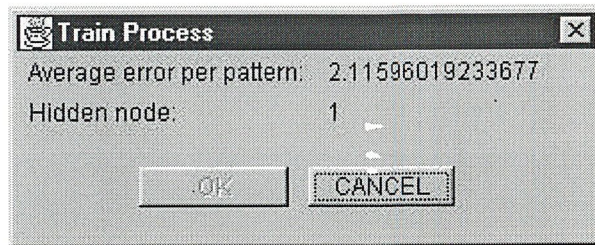
ก่อนที่เราจะกดปุ่ม Next นั้นเราจะต้องป้อนค่าของรูปที่เราเห็น ซึ่งจะแสดงให้เห็นในกรอบสี่เหลี่ยมด้านบนในการป้อนค่าจะป้อนลงในช่องว่างตามที่กำหนดไว้ในโปรแกรมตัวอย่างนี้



รูปภาพที่ 5.2.3 แสดงผลหลังจากมีการกดปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังมีการ Save ข้อมูลแล้วก็จะเป็นการ Train เมื่อกดปุ่ม Train ก็จะขึ้นหน้าต่างมา เพื่อให้เราเลือกข้อมูลที่จะเข้าไป Train ใน Neural Network



รูปภาพที่ 5.2.4 แสดงผลหลังจากมีการกดปุ่ม Train

ในรูปภาพที่ 5.2.4 จะเป็นหน้าต่างย่อยแสดงผลของการ Train เพื่อติดตามการทำงาน ประกอบไปด้วย ค่าของ Error เฉลี่ย และจำนวน Hidden Unit ในขณะนั้น ซึ่งบางครั้งเมื่อค่าของ Error เฉลี่ยลดลงในระดับที่พอใจแล้วเราสามารถหยุดได้เอง

เมื่อ Train ได้ผลลัพธ์ที่ต้องการแล้วก็จะ Save Neural Network ลง File

### 5.3 การทำงานในรูปแบบการแบ่งแยกตัวอักษร

เมื่อ Click ใน Run Button ตามภาพตัวอย่างในรูป 5.1 ก็จะเข้ามาทำงานในรูปแบบของการ แบ่งแยกตัวอักษรจากภาพ

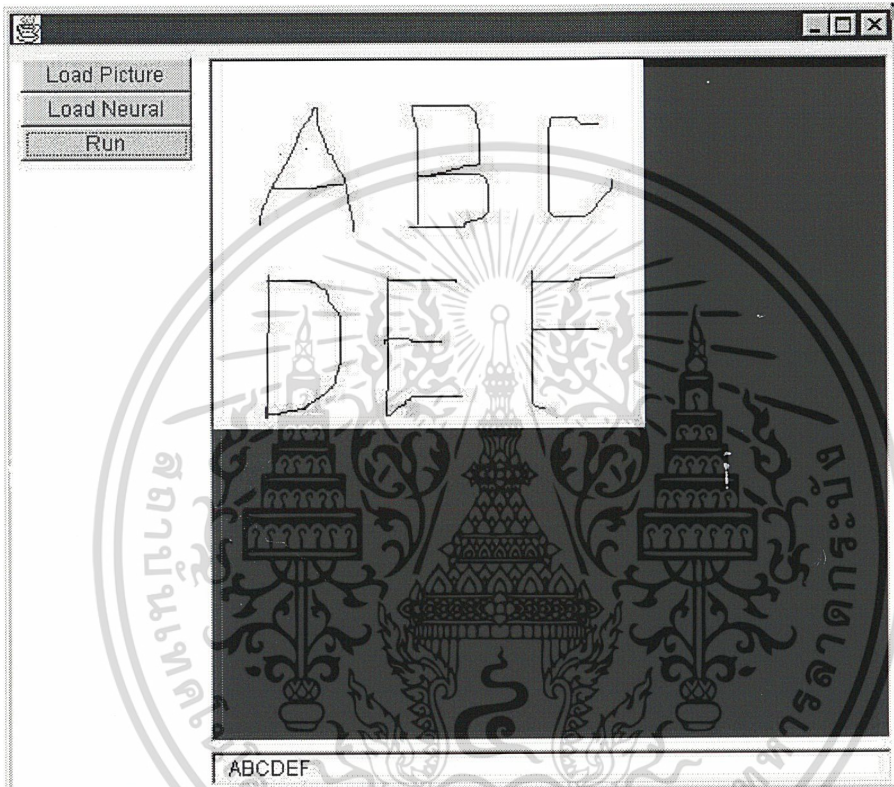


รูปภาพที่ 5.3.1 หน้าตา User interface ของ โปรแกรมในรูปแบบการแบ่งแยกตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 5.3.1 จะประกอบไปด้วยปุ่ม 3 ปุ่ม

- Load Picture ใช้สำหรับ โหลดภาพที่ต้องการจะแบ่งแยกตัวอักษร
- Load Neural Network ใช้สำหรับ โหลด Neural Network ที่ต้องการจาก File
- Run ใช้สำหรับ ประมวลผลใน Neural Network เพื่อแบ่งแยกตัวอักษร



รูปภาพที่ 5.3.2 แสดงผลเมื่อกดปุ่ม Run

จากภาพที่ 5.3.2 จะเป็นการแสดงผลการรู้จำตัวอักษรจากภาพที่ให้เข้ามาโดยมีการกำหนดหาของ Neural Network ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### บทสรุปและวิจารณ์

#### สรุป

จากการศึกษาการทำงานของ Neural Network แบบ Feed forward ซึ่งใช้วิธีการเรียนรู้แบบ Backpropagation นั้นมีความสามารถในการทำงานสูงแต่มีข้อเสียตรงที่ไม่สามารถหาค่าของ Hidden unit ที่เหมาะสมได้จึงต้องเพิ่มกระบวนการทำการปรับเปลี่ยนค่าของ Hidden unit ซึ่งผลที่ได้รับนั้นเป็นที่น่าพอใจเป็นอย่างมากในเรื่องความเร็วในการค้นหาจำนวน Hidden unit ที่เหมาะสม

#### วิจารณ์

จากการศึกษาที่ผ่านมาคิดว่าน่าในประเทศไทยน่าจะมีการให้ความสนใจในเรื่องของการศึกษาในสาขาวิชาปัญญาประดิษฐ์มากกว่านี้ซึ่งในความเป็นจริงแล้วยังมีผู้ที่ไม่ค่อยให้ความสนใจเท่าไรนักทั้ง ๆ ที่ในอนาคตการทำงานเกือบทุกอย่างจำเป็นต้องใช้คอมพิวเตอร์เมื่อมีการใช้มากโอกาสที่จะผลิตบุคลากรรองรับนั้นก็น้อยฉะนั้นงานบางอย่างจึงจำเป็นต้องในคอมพิวเตอร์ตัดสินใจเองเพื่อประสิทธิผลและประสิทธิผลในการทำงาน

#### ข้อเสนอแนะ

ในการรู้จำตัวอักษรที่ทำมาในปริยญาณิพนธ์นี้มีการใช้วิธีการ pre-processing ที่ไม่ยากเท่าไรนักจนบางครั้งไม่สามารถหาความแตกต่างของตัวอักษรที่คล้ายกันมาก ๆ ได้ เพื่อประสิทธิผลในการทำงานจึงอาจจะต้องมีการทำการ pre-processing ในวิธีที่ดีกว่านี้

### บรรณานุกรม

- [1] นายปวีณ ศรีทอง, นายปัญญา ว่องไว, “การทำให้ลายเส้นอยู่ในขนาดมาตรฐาน” การวิเคราะห์ลายเส้นต์ โดยใช้นิวรอลเน็ตเวิร์ค, หน้า 12 ปรียญานิพนธ์ปีการศึกษา 2538 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [2] นายเอกรัตน์ จุลวรรณ์, “ขั้นตอนวิธีการทำลายเส้นให้บางแบบวิธีต่าง ๆ” การประเมินผลของขั้นตอนวิธีการทำลายเส้นให้บางเพื่อนำไปใช้กับภาพพิมพ์ลายนิ้วมือ ,หน้า 83-90 วิทยานิพนธ์ปีการศึกษา 2541 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

**Source Code**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The logo of Rajabhat Buriram University is a circular emblem. It features a central five-tiered umbrella (parasol) with a sunburst above it. The emblem is flanked by two traditional Thai stupa-like structures. The entire design is set against a background of stylized floral and flame patterns. The Thai text around the border of the emblem reads "มหาวิทยาลัยราชภัฏบรจรม" (Mahavithayalai Rajabhat Buriram).

# Class Backpropagation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.lang.*;

public class Backpropagation implements Runnable {
    public int numberInput;
    public int numberOutput;
    public int numberHidden;
    public double[][] weightInputToHidden;
    public double[][] weightHiddenToOutput;
    public double[] biasHidden;
    public double[] biasOutput;
    public double[] inputLayer;
    public double[] outputLayer;
    public double[] activationOutput;
    public double[] hiddenLayer;
    public double[] activationHidden;
    public double[] desireOutput;
    public double learningRate=0.5;

    private double inputPattern[][];
    private double targetPattern[][];
    private double acceptError;

    private Thread trainThread;
    private double avgError;
    private boolean trainComplete=false;

```

```

public Backpropagation() {
}

```

```

public Backpropagation(int i,int o) {
    numberInput=i;

```

```

    numberOutput=o;
    numberHidden=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inputLayer=new double[numberInput];
hiddenLayer=new double[numberHidden];
outputLayer=new double[numberOutput];
desireOutput=new double[numberOutput];

activationOutput=new double[numberOutput];
activationHidden=new double[numberHidden];

biasOutput=new double[numberOutput];
biasHidden=new double[numberHidden];

weightInputToHidden=new double[numberInput][numberHidden];
weightHiddenToOutput=new double[numberHidden][numberOutput];
}

public void feedforward() {
int i,j;
reset();
for(i=0;i<numberHidden;i++)
for(j=0;j<numberInput;j++)
activationHidden[i]+=inputLayer[j]*weightInputToHidden[j][i];

for(i=0;i<numberHidden;i++)
hiddenLayer[i]=sigmoid(biasHidden[i]+activationHidden[i]);

for(i=0;i<numberOutput;i++)
for(j=0;j<numberHidden;j++)
activationOutput[i]+=hiddenLayer[j]*weightHiddenToOutput[j][i];

for(i=0;i<numberOutput;i++)
outputLayer[i]=sigmoid(biasOutput[i]+activationOutput[i]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**public void update() {**  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j;
double errorOutput[]=new double[numberOutput];

//Update Hidden To Output Weight
for(i=0;i<numberOutput;i++) {
    errorOutput[i]=(desireOutput[i]-outputLayer[i])*outputLayer[i]*
        (1-outputLayer[i]);
    for(j=0;j<numberHidden;j++)
        weightHiddenToOutput[j][i]+=learningRate*errorOutput[i]*hiddenLayer[j];
    biasOutput[i]=learningRate*errorOutput[i];
}

```

```

double errorIn[]=new double[numberHidden];
double errorHidden[]=new double[numberHidden];

for(i=0;i<numberHidden;i++)
    for(j=0;j<numberOutput;j++)
        errorIn[i]+=errorOutput[j]*weightHiddenToOutput[i][j];

for(i=0;i<numberHidden;i++) {
    errorHidden[i]=errorIn[i]*(1-hiddenLayer[i])*hiddenLayer[i];
    for(j=0;j<numberInput;j++)
        weightInputToHidden[j][i]+=learningRate*errorHidden[i]*inputLayer[j];
    biasHidden[i]=learningRate*errorHidden[i];
}
}

```

```

public void train(double input[],double desire[],double e) {
    inputPattern=input;
    targetPattern=desire;
    acceptError=e;

```

```

if(trainThread==null) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

trainThread.start();
}

}

public void randomWeight() {
    int i,j;
    for(i=0;i<numberInput;i++)
        for(j=0;j<numberHidden;j++)
            weightInputToHidden[i][j]=randRange();

    for(i=0;i<numberHidden;i++)
        for(j=0;j<numberOutput;j++)
            weightHiddenToOutput[i][j]=randRange();
}

```

```

private double randRange() {
    double result=Math.random();
    if(Math.random()>0.5) result=-result;
    return result;
}

public void setInputLayer(double[] input) {
    for(int i=0;i<numberInput;i++)
        inputLayer[i]=input[i];
}

```

```

public void setLearnRate(double rate) {
    learningRate=rate;
}

```

```

public void setDesireOutput(double[] desire) {
    for(int i=0;i<numberOutput;i++)
        desireOutput[i]=desire[i];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void setWeightInputToHidden(double[][] weight) {
    weightInputToHidden=weight;
}
```

```
public void setNumberHiddenLayer(int number) {
    numberHidden=number;
    activationHidden=new double[number];
    hiddenLayer=new double[number];
}
```

```
public void setWeightHiddenToOutput(double[][] weight) {
    weightHiddenToOutput=weight;
}
```

```
public void setBiasHidden(double[] weight) {
    biasHidden=weight;
}
```

```
public void setBiasOutput(double[] weight) {
    biasOutput=weight;
}
```

```
private double sigmoid(double s) {
    return 1/(1+Math.pow(Math.E,-s));
}
```

```
private void addNode() {
    int i,j;
    numberHidden++;
    double temp1[][]=new double[numberInput][numberHidden];
    double temp2[][]=new double[numberHidden][numberOutput];
    double tempActivation[]=new double[numberHidden];
    double tempBias[]=new double[numberHidden];
    double tempHidden[]=new double[numberHidden];
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int temp=numberHidden-1;

System.out.println("NUMBER OF HIDDEN "+numberHidden);
for(i=0;i<numberInput;i++)
    for(j=0;j<numberHidden;j++) {
        if(j!=temp)
            temp1[i][j]=weightInputToHidden[i][j];
        else
            temp1[i][j]=0.0;
    }

for(i=0;i<numberHidden;i++)
    for(j=0;j<numberOutput;j++)
        if(i!=temp)
            temp2[i][j]=weightHiddenToOutput[i][j];
        else
            temp2[i][j]=0.0;

for(i=0;i<numberHidden;i++)
    if(i!=temp)
        tempBias[i]=biasHidden[i];
    else
        tempBias[i]=0.0;

weightInputToHidden=temp1;
weightHiddenToOutput=temp2;
activationHidden=tempActivation;
biasHidden=tempBias;
hiddenLayer=tempHidden;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j;
numberHidden--;
double temp1[][]=new double[numberInput][numberHidden];
double temp2[][]=new double[numberHidden][numberOutput];
double tempActivation[]=new double[numberHidden];
double tempBias[]=new double[numberHidden];
double tempHidden[]=new double[numberHidden];

for(i=0;i<numberInput;i++)
    for(j=0;j<numberHidden;j++)
        temp1[i][j]=weightInputToHidden[i][j];

for(i=0;i<numberHidden;i++)
    for(j=0;j<numberOutput;j++)
        temp2[i][j]=weightHiddenToOutput[i][j];

for(i=0;i<numberHidden;i++)
    tempBias[i]=biasHidden[i];

weightInputToHidden=temp1;
weightHiddenToOutput=temp2;
activationHidden=tempActivation;
biasHidden=tempBias;
hiddenLayer=tempHidden;
}

```

```

public void displayWeight() {

```

```

    int i,j;

```

```

    for(i=0;i<numberInput;i++)

```

```

        for(j=0;j<numberHidden;j++)

```

```

            System.out.println(weightInputToHidden[i][j]);

```

```

        System.out.println();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public double getError() {
    double totalError=0;
    for(int i=0;i<numberOutput;i++)
//    totalError+=
//    (desireOutput[i]-outputLayer[i])*(desireOutput[i]-outputLayer[i]);
    totalError+=Math.abs(desireOutput[i]-outputLayer[i]);
    return totalError;
}

```

```

public double[] getOutputLayer() {
    return outputLayer;
}

```

```

private void reset() {
    int i;
    for(i=0;i<numberHidden;i++) {
        activationHidden[i]=0;
        hiddenLayer[i]=0;
    }
}

```

```

for(i=0;i<numberOutput;i++) {
    activationOutput[i]=0;
    outputLayer[i]=0;
}
}

```

```

public double[][] getWeightInputToHidden() {
    return weightInputToHidden;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public double[][] getWeightHiddenToOutput() {
    return weightHiddenToOutput;
}
```

```
public double[] getBiasHidden() {
    return biasHidden;
}
```

```
public double[] getBiasOutput() {
    return biasOutput;
}
```

```
public int getInput() {
    return numberInput;
}
```

```
public int getOutput() {
    return numberOutput;
}
```

```
public int getHidden() {
    return numberHidden;
}
```

```
public void run() {
    int numberPattern=inputPattern.length;
    int i,j;
```

```
    int count=0; ////
```

```
    double sumError=0;
```

```
    double sumError1=0;
```

```
    double sumError2=0;
```

```
    boolean converge=false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//check error input.length and desire.length

while(true) {
    while(!converge) {
        if(count!=100) {
            sumError=0;
            for(i=0;i<numberPattern;i++) {
                setInputLayer(inputPattern[i]);
                setDesireOutput(targetPattern[i]);
                feedforward();
                sumError+=getError();
                update();
            }

            sumError=sumError/numberPattern;
            if(sumError<acceptError) {
                System.out.println(sumError);

                converge=true;
                trainComplete=true;
                trainThread.stop();
            }
            count++;
            if(count==1)
                sumError1=sumError;
        } else {
            count=0;
            sumError2=sumError;
            avgError=sumError2;
            if(sumError1-sumError2<0.01*sumError1)
                addNode();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
}  
  
public double getAverageError() {  
    return avgError;  
}  
  
public int getNumberHiddenNode() {  
    return numberHidden;  
}  
  
public boolean trainComplete() {  
    return trainComplete;  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.lang.*;
import java.util.*;
public class Block {
    int dataBlock[][];
    int dataBlock_WIDTH;
    int dataBlock_HEIGHT;
    int scanline_TOP;
    int scanline_LEFT;
    int scanline_BUTTOM;
    int scanline_RIGHT;
    int tokenData[][];
    int numberOfToken=0;
    boolean haveToken=true;
    Vector tokenDisplayVector=new Vector();
    Vector tokenDataVector=new Vector();
    int numberOfDisplayData=0;

    public Block(int data[][]) {
        dataBlock=data;
        dataBlock_HEIGHT=dataBlock.length-1;
        dataBlock_WIDTH=dataBlock[0].length-1;
        while(haveAnyToken()) getToken();
    }

    public void getToken() {
        scanline_TOP=0;
        scanline_LEFT=0;
        scanline_BUTTOM=0;
        scanline_RIGHT=0;
        findToken(0,0,dataBlock_WIDTH,dataBlock_HEIGHT);
    }

    private void findToken(int left,int top,int right,int buttom) {

```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 //case top,left  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j;

//Find scanline_TOP
for(i=top;i<=buttom;i++)
for(j=left;j<=right;j++)
  if(dataBlock[i][j]==1) {
    scanline_TOP=i;
    i=buttom+1;
    j=right+1;
  }

//Find scanline_BUTTOM
for(i=scanline_TOP;i<=buttom;i++)
for(j=left;j<=right;j++)
  if(dataBlock[i][j]==1) j=right+1;
  else if(j==right) {
    scanline_BUTTOM=i-1;
    i=buttom+1;
  }

//Find scanline_LEFT
for(j=left;j<=right;j++)
for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
  if(dataBlock[i][j]==1) {
    scanline_LEFT=j;
    i=scanline_BUTTOM+1;
    j=right+1;
  }

//Find scanline_RIGHT
for(j=scanline_LEFT;j<=right;j++)
for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
  if(dataBlock[i][j]==1) i=scanline_BUTTOM+1;

```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    j=right+1;
}

```

```
//Cut blank Top and Buttom again
```

```

for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
for(j=scanline_LEFT;j<=scanline_RIGHT;j++)
if(dataBlock[i][j]==1) {
    scanline_TOP=i;
    j=scanline_RIGHT+1;
    i=scanline_BUTTOM+1;
}

```

```

for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
for(j=scanline_LEFT;j<=scanline_RIGHT;j++)
if(dataBlock[i][j]==1) j=scanline_RIGHT+1;
else if(j==scanline_RIGHT) scanline_BUTTOM=i-1;

```

```
haveToken=false;
```

```
//Check it Single Token
```

```
//first check left to right if it not have blank step 2
```

```

for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
for(j=scanline_LEFT;j<=scanline_RIGHT;j++)
if(dataBlock[i][j]==1) j=scanline_RIGHT+1;
else if(j==scanline_RIGHT) haveToken=true;

```

```
//step 2 check top to bottom
```

```

if(!haveToken)
for(j=scanline_LEFT;j<=scanline_RIGHT;j++)
for(i=scanline_TOP;i<=scanline_BUTTOM;i++)
if(dataBlock[i][j]==1) i=scanline_BUTTOM+1;
else if(i==scanline_BUTTOM) haveToken=true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//if it is a single token haveToken=false we want to clear
```

```
//block in the token position
```

```
//clear Block
```

```
if(!haveToken) {
```

```
    //copy dataBlock--> dataToken
```

```
    int datatoken[][]=new int[scanline_BOTTOM-scanline_TOP+1]
        [scanline_RIGHT-scanline_LEFT+1];
```

```
    for(i=scanline_TOP;i<=scanline_BOTTOM;i++)
```

```
        for(j=scanline_LEFT;j<=scanline_RIGHT;j++)
```

```
            datatoken[i-scanline_TOP][j-scanline_LEFT]=dataBlock[i][j];
```

```
    clearBlock(scanline_LEFT,scanline_TOP,scanline_RIGHT,scanline_BOTTOM);
```

```
    Token x=new Token(datatoken);
```

```
    tokenDisplayVector.addElement(x);
```

```
    tokenDataVector.addElement(x);
```

```
}
```

```
else
```

```
    findToken(scanline_LEFT,scanline_TOP,scanline_RIGHT,scanline_BOTTOM);
```

```
}
```

```
private void clearBlock(int left,int top,int right,int buttom) {
```

```
    int i,j;
```

```
    for(i=top;i<=buttom;i++)
```

```
        for(j=left;j<=right;j++)
```

```
            dataBlock[i][j]=0;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public boolean haveAnyDisplayData() {
    if(numberOfDisplayData==0)
        return false;
    else
        return true;
}

```

```

public int[][] getDisplayData() {
    Token token=null;
    if(!tokenDisplayVector.isEmpty()) {
        token =(Token)tokenDisplayVector.firstElement();
        tokenDisplayVector.removeElementAt(0);
    }
    numberOfDisplayData=tokenDisplayVector.size();
    return token.getDisplayData();
}

```

////////////////////////////////////

```

public double[] getTokenData() {
    Token token=null;
    if(!tokenDataVector.isEmpty()) {
        token =(Token)tokenDataVector.firstElement();
        tokenDataVector.removeElementAt(0);
    }
    return token.getData();
}

```

////////////////////////////////////

```

public boolean haveAnyToken() {
    int i,j;
    boolean temp=false;

```

```

    for(i=0;i<=dataBlock_HEIGHT;i++)

```

```

        for(j=0;j<=dataBlock_WIDTH;j++)

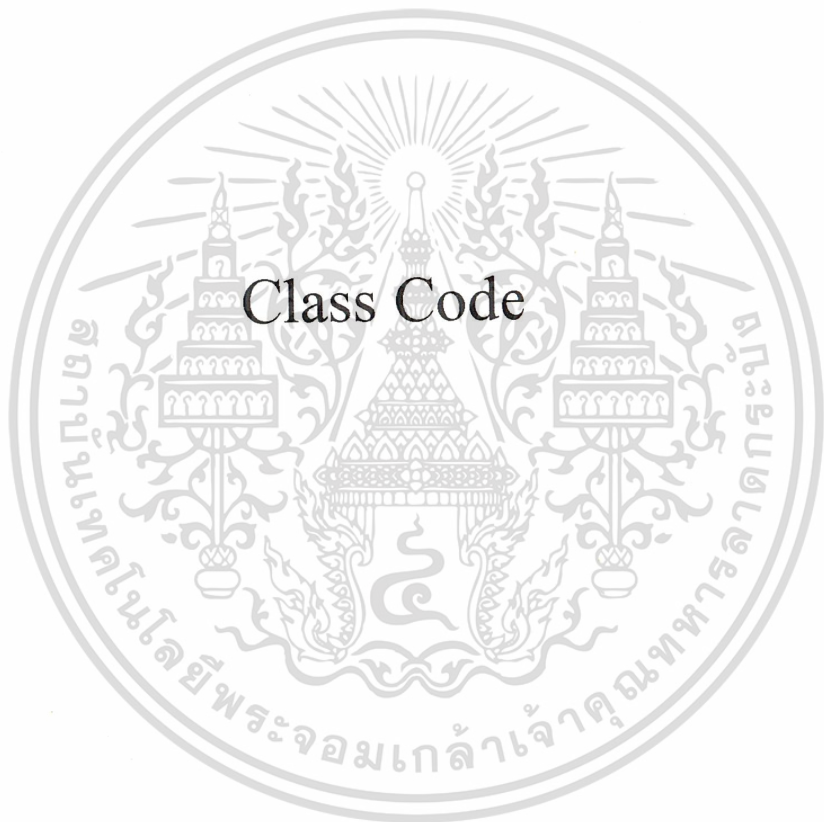
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(dataBlock[i][j]==1) {  
    j=dataBlock_WIDTH;  
    i=dataBlock_HEIGHT;  
    temp=true;  
}  
return temp;  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public class Code {
    public static final int SIZE=6;
    public static final double codeA[]={0,0,0,0,0,1};
    public static final double codeB[]={0,0,0,0,1,0};
    public static final double codeC[]={0,0,0,0,1,1};
    public static final double codeD[]={0,0,0,1,0,0};
    public static final double codeE[]={0,0,0,1,0,1};
    public static final double codeF[]={0,0,0,1,1,0};
    public static final double codeG[]={0,0,0,1,1,1};
    public static final double codeH[]={0,0,1,0,0,0};
    public static final double codeI[]={0,0,1,0,0,1};
    public static final double codeJ[]={0,0,1,0,1,0};
    public static final double codeK[]={0,0,1,0,1,1};
    public static final double codeL[]={0,0,1,1,0,0};
    public static final double codeM[]={0,0,1,1,0,1};
    public static final double codeN[]={0,0,1,1,1,0};
    public static final double codeO[]={0,0,1,1,1,1};
    public static final double codeP[]={0,1,0,0,0,0};
    public static final double codeQ[]={0,1,0,0,0,1};
    public static final double codeR[]={0,1,0,0,1,0};
    public static final double codeS[]={0,1,0,0,1,1};
    public static final double codeT[]={0,1,0,1,0,0};
    public static final double codeU[]={0,1,0,1,0,1};
    public static final double codeV[]={0,1,0,1,1,0};
    public static final double codeW[]={0,1,0,1,1,1};
    public static final double codeX[]={0,1,1,0,0,0};
    public static final double codeY[]={0,1,1,0,0,1};
    public static final double codeZ[]={0,1,1,0,1,0};
    public static final double codea[]={0,1,1,0,1,1};
    public static final double codeb[]={0,1,1,1,0,0};
    public static final double codec[]={0,1,1,1,0,1};
    public static final double coded[]={0,1,1,1,1,0};
    public static final double codee[]={0,1,1,1,1,1};
    public static final double codef[]={1,0,0,0,0,0};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public static final double codeg[]={1,0,0,0,0,1};
public static final double codeh[]={1,0,0,0,1,0};
public static final double codei[]={1,0,0,0,1,1};
public static final double codej[]={1,0,0,1,0,0};
public static final double codek[]={1,0,0,1,0,1};
public static final double codel[]={1,0,0,1,1,0};
public static final double codem[]={1,0,0,1,1,1};
public static final double coden[]={1,0,1,0,0,0};
public static final double codeo[]={1,0,1,0,0,1};
public static final double codep[]={1,0,1,0,1,0};
public static final double codeq[]={1,0,1,0,1,1};
public static final double coder[]={1,0,1,1,0,0};
public static final double codes[]={1,0,1,1,0,1};
public static final double codet[]={1,0,1,1,1,0};
public static final double codeu[]={1,0,1,1,1,1};
public static final double codev[]={1,1,0,0,0,0};
public static final double codew[]={1,1,0,0,0,1};
public static final double codex[]={1,1,0,0,1,0};
public static final double codey[]={1,1,0,0,1,1};
public static final double codez[]={1,1,0,1,0,0};
public static final double code1[]={1,1,0,1,0,1};
public static final double code2[]={1,1,0,1,1,0};
public static final double code3[]={1,1,0,1,1,1};
public static final double code4[]={1,1,1,0,0,0};
public static final double code5[]={1,1,1,0,0,1};
public static final double code6[]={1,1,1,0,1,0};
public static final double code7[]={1,1,1,0,1,1};
public static final double code8[]={1,1,1,1,0,0};
public static final double code9[]={1,1,1,1,0,1};
public static final double code0[]={1,1,1,1,1,0};

public static final double codeError[]={0,0,0,0,0,0};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**public static String decode(double[] code) {**  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(checkMatch(code,codeA)) return "A";
if(checkMatch(code,codeB)) return "B";
if(checkMatch(code,codeC)) return "C";
if(checkMatch(code,codeD)) return "D";
if(checkMatch(code,codeE)) return "E";
if(checkMatch(code,codeF)) return "F";
if(checkMatch(code,codeG)) return "G";
if(checkMatch(code,codeH)) return "H";
if(checkMatch(code,codeI)) return "I";
if(checkMatch(code,codeJ)) return "J";
if(checkMatch(code,codeK)) return "K";
if(checkMatch(code,codeL)) return "L";
if(checkMatch(code,codeM)) return "M";
if(checkMatch(code,codeN)) return "N";
if(checkMatch(code,codeO)) return "O";
if(checkMatch(code,codeP)) return "P";
if(checkMatch(code,codeQ)) return "Q";
if(checkMatch(code,codeR)) return "R";
if(checkMatch(code,codeS)) return "S";
if(checkMatch(code,codeT)) return "T";
if(checkMatch(code,codeU)) return "U";
if(checkMatch(code,codeV)) return "V";
if(checkMatch(code,codeW)) return "W";
if(checkMatch(code,codeX)) return "X";
if(checkMatch(code,codeY)) return "Y";
if(checkMatch(code,codeZ)) return "Z";
if(checkMatch(code,codea)) return "a";
if(checkMatch(code,codeb)) return "b";
if(checkMatch(code,codec)) return "c";
if(checkMatch(code,codee)) return "d";
if(checkMatch(code,codee)) return "e";
if(checkMatch(code,codef)) return "f";

```

```

if(checkMatch(code,codeg)) return "g";

```

```

if(checkMatch(code,codeh)) return "h";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(checkMatch(code,codei)) return "i";
if(checkMatch(code,codej)) return "j";
if(checkMatch(code,codek)) return "k";
if(checkMatch(code,code1)) return "l";
if(checkMatch(code,codem)) return "m";
if(checkMatch(code,coden)) return "n";
if(checkMatch(code,codeo)) return "o";
if(checkMatch(code,codep)) return "p";
if(checkMatch(code,codeq)) return "q";
if(checkMatch(code,coder)) return "r";
if(checkMatch(code,codes)) return "s";
if(checkMatch(code,codet)) return "t";
if(checkMatch(code,codeu)) return "u";
if(checkMatch(code,codev)) return "v";
if(checkMatch(code,codew)) return "w";
if(checkMatch(code,codex)) return "x";
if(checkMatch(code,codey)) return "y";
if(checkMatch(code,codez)) return "z";
if(checkMatch(code,code1)) return "1";
if(checkMatch(code,code2)) return "2";
if(checkMatch(code,code3)) return "3";
if(checkMatch(code,code4)) return "4";
if(checkMatch(code,code5)) return "5";
if(checkMatch(code,code6)) return "6";
if(checkMatch(code,code7)) return "7";
if(checkMatch(code,code8)) return "8";
if(checkMatch(code,code9)) return "9";
if(checkMatch(code,code0)) return "0";

else return "Error";
}

```

```

public static boolean checkMatch(double code1[],double code2[]) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int i=0;i<code1.length;i++)
    if(code1[i]!=code2[i]) return false;
return true;
}

```

```

public static double[] encode(String code) {
    if(code.equals("A")) return codeA;
    if(code.equals("B")) return codeB;
    if(code.equals("C")) return codeC;
    if(code.equals("D")) return codeD;
    if(code.equals("E")) return codeE;
    if(code.equals("F")) return codeF;
    if(code.equals("G")) return codeG;
    if(code.equals("H")) return codeH;
    if(code.equals("I")) return codeI;
    if(code.equals("J")) return codeJ;
    if(code.equals("K")) return codeK;
    if(code.equals("L")) return codeL;
    if(code.equals("M")) return codeM;
    if(code.equals("N")) return codeN;
    if(code.equals("O")) return codeO;
    if(code.equals("P")) return codeP;
    if(code.equals("Q")) return codeQ;
    if(code.equals("R")) return codeR;
    if(code.equals("S")) return codeS;
    if(code.equals("T")) return codeT;
    if(code.equals("U")) return codeU;
    if(code.equals("V")) return codeV;
    if(code.equals("W")) return codeW;
    if(code.equals("X")) return codeX;
    if(code.equals("Y")) return codeY;
    if(code.equals("Z")) return codeZ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(code.equals("c")) return codec;
if(code.equals("d")) return coded;
if(code.equals("e")) return codee;
if(code.equals("f")) return codef;
if(code.equals("g")) return codeg;
if(code.equals("h")) return codeh;
if(code.equals("i")) return codei;
if(code.equals("j")) return codej;
if(code.equals("k")) return codek;
if(code.equals("l")) return codeL;
if(code.equals("m")) return codem;
if(code.equals("n")) return coden;
if(code.equals("o")) return codeo;
if(code.equals("p")) return codep;
if(code.equals("q")) return codeq;
if(code.equals("r")) return coder;
if(code.equals("s")) return codes;
if(code.equals("t")) return codet;
if(code.equals("u")) return codeu;
if(code.equals("v")) return codev;
if(code.equals("w")) return codew;
if(code.equals("x")) return codex;
if(code.equals("y")) return codey;
if(code.equals("z")) return codez;
if(code.equals("1")) return code1;
if(code.equals("2")) return code2;
if(code.equals("3")) return code3;
if(code.equals("4")) return code4;
if(code.equals("5")) return code5;
if(code.equals("6")) return code6;
if(code.equals("7")) return code7;
if(code.equals("8")) return code8;
if(code.equals("9")) return code9;
if(code.equals("0")) return code0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else return codeError;
```

```
}
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# Class FileManager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.io.*;

public class FileManager {

    public static final String LOAD="r";
    public static final String SAVE="rw";
    public RandomAccessFile raf;

    public FileManager(String name,String mode) {
        try {
            raf=new RandomAccessFile(name,mode);
        } catch(IOException e) {}
    }

    public void appendData() {
        try {
            raf.seek(raf.length());
        } catch(IOException e) {}
    }

    public void setPosition(long position) {
        try {
            raf.seek(position);
        } catch(IOException e) {}
    }

    public void write(double data) {
        try {
            raf.writeDouble(data);
        } catch(IOException e) {}
    }

    public void writeCode(String data) {
        try {
            raf.writeUTF(data);
        } catch(IOException e) {}
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public void writeInt(int data) {
    try {
        raf.writeInt(data);
    } catch(IOException e) {}
}

```

```

public void writeString(String data) {
    try {
        raf.writeBytes(data);
    } catch(IOException e) {}
}

```

```

public double read() {
    double temp=0.0;
    try {
        temp=raf.readDouble();
    } catch(IOException e) {}
    return temp;
}

```

```

public int readInt() {
    int temp=0;
    try {
        temp=raf.readInt();
    } catch(IOException e) {}
    return temp;
}

```

```

public String readUTF() {
    String temp="";
    try {
        temp=raf.readUTF();
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
} catch(IOException e) {}  
return temp;  
}
```

```
public long getSize() {  
    long temp=0;  
    try {  
        temp=raf.length();  
    } catch(IOException e) {}  
    return temp;  
}
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The seal of Rajabhat Buriram University is a circular emblem. It features a central five-tiered umbrella (parasol) with a sunburst above it. The emblem is flanked by two smaller umbrellas and is surrounded by intricate floral and geometric patterns. The Thai text around the border of the seal reads "มหาวิทยาลัยราชภัฏบรจรม" (Mahavithayalai Rajabhat Buriram).

# Class FileNeuralManager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.io.*;

public class FileNeuralManager {

    public int numberOfPattern;

    public int numberOfOutput;

    public int numberOfData;

    public double patternInput[][];

    public double requireOutput[][];

    public Backpropagation b;

    public void loadDataForTrain(FileTrainData file) {

        int i,j;

        numberOfPattern=file.getNumberOfData();

        numberOfOutput=file.getNumberCode();

        numberOfData=file.getNumberPerData();

        patternInput=new double[numberOfPattern][numberOfData];

        requireOutput=new double[numberOfPattern][numberOfOutput];

        file.readFormat();

        for(i=0;i<numberOfPattern;i++) {

            for(j=0;j<numberOfData;j++)

                patternInput[i][j]=file.readData();

            for(j=0;j<numberOfOutput;j++)

                requireOutput[i][j]=file.readData();

            if(file.readCode().equals("\n")) {}

            else System.out.println("ERROR");

        }

    }

    public int getNumberOutput() {

        return numberOfOutput;

    }

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้มีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return numberOfData;
}

public double[][] getPatternInput() {
    return patternInput;
}

public double[][] getRequireOutput() {
    return requireOutput;
}

```

```

public void saveData(FileManager file, Backpropagation back) {
    int i,j;
    double wih[][]=back.getWeightInputToHidden();
    double who[][]=back.getWeightHiddenToOutput();
    double bh[]=back.getBiasHidden();
    double bo[]=back.getBiasOutput();

    file.writeInt(wih.length);
    file.writeInt(wih[0].length);
    file.writeInt(who[0].length);
    file.writeInt(bh.length);
    file.writeInt(bo.length);

    for(i=0;i<wih.length;i++)
        for(j=0;j<wih[0].length;j++)
            file.write(wih[i][j]);

    for(i=0;i<who.length;i++)
        for(j=0;j<who[0].length;j++)
            file.write(who[i][j]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

file.write(bh[i]);

for(i=0;i<bo.length;i++)
file.write(bo[i]);
}

public Backpropagation loadData(FileManager file) {
int i,j;
int numberOfInput=file.readInt();
int numberOfHidden=file.readInt();
int numberOfOutput=file.readInt();
int numberOfBiasH=file.readInt();
int numberOfBiasO=file.readInt();

double wih[][]=new double[numberOfInput][numberOfHidden];
double who[][]=new double[numberOfHidden][numberOfOutput];
double bh[]=new double[numberOfBiasH];
double bo[]=new double[numberOfBiasO];
for(i=0;i<numberOfInput;i++)
for(j=0;j<numberOfHidden;j++)
wih[i][j]=file.read();

for(i=0;i<numberOfHidden;i++)
for(j=0;j<numberOfOutput;j++)
who[i][j]=file.read();

for(i=0;i<numberOfBiasH;i++)
bh[i]=file.read();

for(i=0;i<numberOfBiasO;i++)
bo[i]=file.read();

```

Backpropagation back=new Backpropagation(numberOfInput, numberOfOutput);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
back.setWeightInputToHidden(wih);  
back.setWeightHiddenToOutput(who);  
back.setNumberHiddenLayer(numberOfHidden);  
back.setBiasHidden(bh);  
back.setBiasOutput(bo);  
return back;
```

}

}



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Class FileTrainData

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.io.*;

public class FileTrainData extends FileManager {
    String fileName;
    int fileMode;
    int count=0;
    int numberData=0;
    int numberPerData=0;
    int numberCode=6;
    double typeData[];
    public FileTrainData(String path,String name,String mode) {
        super(name,mode);
        if(getSize(>0)
            readFormat();
        else
            setFormat();
    }

    public void setFormat() {
        setPosition(0);
        writeInt(numberData);
        writeInt(numberPerData);
        writeInt(numberCode);
        writeCode("\n");
    }

    public void readFormat() {
        setPosition(0);
        numberData=readInt();
        numberPerData=readInt();
        numberCode=readInt();
        String s=readCode();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void setNumberPerData(int data) {
    numberPerData=data;
}
```

```
public int getNumberOfData() {
    return numberData;
}
```

```
public int getNumberCode() {
    return numberCode;
}
```

```
public int getNumberPerData() {
    return numberPerData;
}
```

```
public void save(double[] data,String s) {
    int i;
    appendData();
    for(i=0;i<data.length;i++)
        write(data[i]);

    typeData=Code.encode(s);

    for(i=0;i<typeData.length;i++)
        write(typeData[i]);

    writeCode("\n");
    setPosition(0);
    numberData++;
    writeInt(numberData);
}
```

```
public void append(double[] data,String s) {
    appendData();
    save(data,s);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public double readData() {  
    return read();  
}  
public String readCode() {  
    return readUTF();  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;

public class LoadCharacter extends Canvas {
    private int width=100;
    private int height=100;
    private int blockWidth;
    private int blockHeight;

    private int[][] characterImage;

    public LoadCharacter() {
        setSize(width,height);
    }

    public void load(int[][] imageData) {
        int i,j;
        characterImage=imageData;
        blockWidth=width/imageData[0].length;
        blockHeight=height/imageData.length;
        repaint();
    }

    public void paint(Graphics g) {
        g.setColor(Color.white);
        g.fillRect(0,0,width,height);
        if(characterImage!=null) {
            g.setColor(Color.black);
            for(int i=0;i<blockHeight;i++)
                for(int j=0;j<blockWidth;j++)
                    if(characterImage[j][i]==1)
                        g.fillRect(i*blockWidth,j*blockHeight,blockWidth,blockHeight);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.image.*;

public class LoadImage extends Canvas {

    private Image theImage;
    private int width=100;
    private int height=100;
    private boolean loadComplete=false;

    public LoadImage() {
    }

    public LoadImage(String fileImage) {
        theImage=getToolkit().getImage(fileImage);
        width=getWidth();
        height=getHeight();
        //loadComplete=true;
        this.setSize(width,height);
    }

    public void load(String fileImage) {
        theImage=getToolkit().getImage(fileImage);
        width=getWidth();
        height=getHeight();
        //loadComplete=true;
        this.setSize(width,height);
        repaint();
    }

    public boolean loadComplete() {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 }  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public int getHeight() {
    while(theImage.getHeight(this)==-1) {}
    return theImage.getHeight(this);
}

public int getWidth() {
    while(theImage.getWidth(this)==-1) {}
    return theImage.getWidth(this);
}

public Image getImage() {
    return theImage;
}

public void paint(Graphics g) {
    g.drawImage(theImage,0,0,this);
}

public int[][] convertToArray() {
    int Width=getWidth();
    int Height=getHeight();
    int pixelArray[]=new int[Width*Height];
    PixelGrabber pg=new
        PixelGrabber(theImage,0,0,Width,Height,pixelArray,0,Width);

    try {
        pg.grabPixels();
    } catch(InterruptedException e){};

    Color middle_color=new Color(125,125,125);
    int middle_color_int=middle_color.hashCode();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 int temp[][]=new int[Height][Width];  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int count=-1;

for(int i=0;i<Height;i++)
    for(int j=0;j<Width;j++) {
        count++;
        temp[i][j]=pixelArray[count];
//CONVERT COLOR HASHCODE TO BINARY IF COLOR > MIDDLE =1 ELSE =0
        if(temp[i][j]<middle_color_int)
            temp[i][j]=1;
        else temp[i][j]=0;
    }
return temp;
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.event.*;
public class MyFrame extends Frame implements WindowListener {
    public MyFrame() {
        this.addWindowListener(this);
    }
    public MyFrame(String st) {
        super(st);
        addWindowListener(this);
    }
    public void windowActivated(WindowEvent e) {}
    public void windowClosed(WindowEvent e) {
    }
    public void windowClosing(WindowEvent e) {
        this.dispose();
    }
    public void windowDeactivated(WindowEvent e) {}
    public void windowDeiconified(WindowEvent e) {}
    public void windowIconified(WindowEvent e) {}
    public void windowOpened(WindowEvent e) {}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;

public class Neural extends MyFrame implements ActionListener {
    Button trainButton=new Button("Train");
    Button runButton=new Button("Run");
    TrainFrame trainFrame;
    RunFrame runFrame;

    public Neural() {
        setLayout(null);
        setSize(400,400);

        trainButton.addActionListener(this);
        trainButton.setLocation(10,30);
        trainButton.setSize(70,20);

        runButton.addActionListener(this);
        runButton.setLocation(10,50);
        runButton.setSize(70,20);

        add(trainButton);
        add(runButton);
    }

```

```

public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("Train")) {
        trainFrame=new TrainFrame();
        trainFrame.show();
        //Button LOAD Clicked
    } else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 if(e.getActionCommand().equals("Run")) {  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

runFrame=new RunFrame();
runFrame.show();
}
}
}

```

```

public static void main(String arg[] ) {
    Neural neural=new Neural();
    neural.show();
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Class Picture Area

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.image.*;
public class PictureArea extends ScrollPane {

    Image characterImage;
    LoadImage image=new LoadImage();
    Block myBlock ;

    public PictureArea() {
        super(ScrollPane.SCROLLBARS_AS_NEEDED);
        this.setSize(400,400);
        setBackground(Color.black);
    }

    public void loadImage(String fileImage) {
        image.load(fileImage);
        add(image);
    }

    public void segmentation() {
        makeBlock();
    }

    public int[][] getCharacter() {
        int imageData[][]=myBlock.getDisplayData();
        return imageData;
    }

    public double[] getData() {
        return myBlock.getTokenData();
    }

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ปรากฏในเอกสารนี้ หากมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public Image getCharacterImage() {
    return characterImage;
}

private void makeBlock() {
    int[][] block=image.convertToArray();
    myBlock=new Block(block);
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Class RunFrame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.event.*;
public class RunFrame extends MyFrame implements ActionListener {
    Button runButton=new Button("Run");
    Button loadPictureButton=new Button("Load Picture");
    Button loadNeuralButton=new Button("Load Neural");
    TextField answerTextField=new TextField();
    PictureArea pictureArea=new PictureArea();
    Frame frameParentForFileDialog=new Frame();
    TheFileDialog theFileDialog;
    FileNeuralManager fnm=new FileNeuralManager();
    Backpropagation back=new Backpropagation();
    public RunFrame() {
        setLayout(null);
        setSize(600,500);
        loadPictureButton.setLocation(10,30);
        loadPictureButton.setSize(100,20);
        loadPictureButton.addActionListener(this);

        loadNeuralButton.setLocation(10,50);
        loadNeuralButton.setSize(100,20);
        loadNeuralButton.addActionListener(this);

        runButton.setLocation(10,70);
        runButton.setSize(100,20);
        runButton.addActionListener(this);
        runButton.setEnabled(false);

        pictureArea.setLocation(120,30);

        answerTextField.setLocation(120,435);
        answerTextField.setSize(400,20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

add(loadPictureButton);
add(loadNeuralButton);
add(runButton);
add(pictureArea);
add(answerTextField);
}

```

```

private void clickLoadPictureButton() {
    theFileDialog=new
        TheFileDialog(frameParentForFileDialog
            ,"Load Picture",FileDialog.LOAD);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        String thePictureFile=theFileDialog.getDirectory()
            +theFileDialog.getFile();
        pictureArea.removeAll();
        pictureArea.loadImage(thePictureFile);
        pictureArea.segmentation();
    }
}

```

```

private void clickLoadNeuralButton() {
    theFileDialog=new
        TheFileDialog(frameParentForFileDialog
            ,"Load Neural",FileDialog.LOAD);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        FileManager fileManager=new
            FileManager(theFileDialog.getFile(),"r");
        back=fnm.loadData(fileManager);
        runButton.setEnabled(true);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void clickRunButton() {
    back.setInputLayer(pictureArea.getData());
    back.feedforward();
    double temp[]=back.getOutputLayer();
    for(int i=0;i<temp.length;i++) {
        if(temp[i]>=0.8) temp[i]=1;
        else {
            if(temp[i]<=0.2) temp[i]=0;
            else System.out.print(2);
        }
    }
    answerTextField.setText(answerTextField.getText()+Code.decode(temp));
}

public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("Load Picture"))
        clickLoadPictureButton();
    else {
        if(e.getActionCommand().equals("Load Neural"))
            clickLoadNeuralButton();
        else {
            if(e.getActionCommand().equals("Run"))
                clickRunButton();
        }
    }
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The seal of Rajabhat Buriram University is a circular emblem. It features a central five-tiered stupa with a sunburst above it. The stupa is flanked by two smaller stupa-like structures. The entire emblem is surrounded by a decorative border with Thai script. The text around the border reads "มหาวิทยาลัยราชภัฏบรจรม" at the top and "วิทยาเขตบรจรม" at the bottom.

# Class TheFileDialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.io.*;
import java.awt.*;

public class TheFileDialog extends FileDialog implements FilenameFilter {

    public static final int LOAD=0;
    public static final int SAVE=1;

    File theFile;
    String theFilterSuffix;

    public TheFileDialog(Frame parent,String title,int mode) {
        super(parent,title,mode);
    }

    public void setFilterSuffix(String fileSuffix) {
        theFilterSuffix=fileSuffix;
        setFilenameFilter(this);
    }

    public boolean accept(File dir,String filter) {
        return filter.endsWith(theFilterSuffix);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;

public class Token {
    int dataToken[][];
    int height;
    int width;
    public static final int numberOfBlockX=10;
    public static final int numberOfBlockY=10;
    double encodeData[]=new double[100];
    int displayData[][]=new int[numberOfBlockY][numberOfBlockX];

    public Token(int[][] data) {
        dataToken=data;
        height=dataToken.length;
        width=dataToken[0].length;
        skeleton();
    }

    public double[] getData() {
        encode();
        return encodeData;
    }
    //////////////////////////////////////
    private void encode() {
        setSize();
        int i,j;
        for(i=0;i<10;i++)
            for(j=0;j<10;j++)
                encodeData[i*10+j]=displayData[i][j];
    }
    //////////////////////////////////////
    public int[][] getDisplayData() {
        setSize();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return displayData;
}

```

```

private void setSize() {
    int sumRow[]=new int[height];
    int sumCol[]=new int[width];
    int tempDisplayData[][]=new int[numberOfBlockY][numberOfBlockX];
    int totalRow=0;
    int totalCol=0;
    int i,j,k,l;
    int tempX=0;
    int tempY=0;
    for(j=0;j<width;j++)
        for(i=0;i<height;i++)
            if(dataToken[i][j]==1) sumCol[j]++;
    for(j=0;j<width;j++)
        totalCol=sumCol[j]+totalCol;
    for(i=0;i<height;i++)
        for(j=0;j<width;j++)
            if(dataToken[i][j]==1) sumRow[i]++;
    for(i=0;i<height;i++)
        totalRow=sumRow[i]+totalRow;
    for(j=0;j<width;j++)
        for(i=0;i<height;i++)
            if(dataToken[i][j]==1) {
                tempX=0;
                tempY=0;
                for(k=0;k<j;k++) tempX=sumCol[k]+tempX;
                for(l=0;l<i;l++) tempY=sumRow[l]+tempY;
                tempDisplayData[tempY*numberOfBlockY/totalRow]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[tempX*numberOfBlockX/totalCol]=1;
```

```
}
```

```
displayData=tempDisplayData;
```

```
}
```

```
private void skeleton() {
```

```
    int i,j;
```

```
    int window[];
```

```
    int tempdata[][]=dataToken;
```

```
    boolean erase=true;
```

```
    while(erase) {
```

```
        erase=false;
```

```
        //first step
```

```
        for(i=0;i<height;i++)
```

```
            for(j=0;j<width;j++)
```

```
                if(dataToken[i][j]==1) {
```

```
                    window=setIndexArray(i,j);
```

```
                    if(!((window[0]==0&&window[1]==1)||(window[4]==0&&window[3]==1)
```

```
                        ||(window[1]+window[0]+window[7]+window[5]+window[4]+
                            window[3]==0)))
```

```
                        if(window[6]==1&&window[2]==0) tempdata[i][j]=2;
```

```
                }
```

```
        // erase
```

```
        for(i=0;i<height;i++)
```

```
            for(j=0;j<width;j++)
```

```
                if(tempdata[i][j]==2) {
```

```
                    dataToken[i][j]=0;
```

```
                    tempdata[i][j]=0;
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    erase=true;
}

//seconde step
for(i=0;i<height;i++)
for(j=0;j<width;j++)
if(dataToken[i][j]==1) {
    window=setIndexArray(i,j);
    if(!((window[6]==0&&window[7]==1)||((window[2]==0&&window[1]==1)
        ||(window[2]+window[1]+window[7]+window[6]+window[5]+
            window[3]==0))))
        if(window[4]==1&&window[0]==0) tempdata[i][j]=2;
}

// erase
for(i=0;i<height;i++)
for(j=0;j<width;j++)
if(tempdata[i][j]==2) {
    dataToken[i][j]=0;
    tempdata[i][j]=0;
    erase=true;
}

//tird step
for(i=0;i<height;i++)
for(j=0;j<width;j++)
if(dataToken[i][j]==1) {
    window=setIndexArray(i,j);
    if(!((window[0]==0&&window[7]==1)||((window[4]==0&&window[5]==1)
        ||(window[1]+window[0]+window[7]+window[5]+window[4]+
            window[3]==0))))
        if(window[2]==1&&window[6]==0) tempdata[i][j]=2;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// erase
for(i=0;i<height;i++)
  for(j=0;j<width;j++)
    if(tempdata[i][j]==2) {
      dataToken[i][j]=0;
      tempdata[i][j]=0;
      erase=true;
    }

//four step
for(i=0;i<height;i++)
  for(j=0;j<width;j++)
    if(dataToken[i][j]==1) {
      window=setIndexArray(i,j);
      if(!((window[6]==0&&window[5]==1)||((window[2]==0&&window[3]==1)
        ||(window[2]+window[1]+window[7]+window[6]+window[5]+
          window[3]==0))))
        if(window[0]==1&&window[4]==0) tempdata[i][j]=2;
    }

// erase
for(i=0;i<height;i++)
  for(j=0;j<width;j++)
    if(tempdata[i][j]==2) {
      dataToken[i][j]=0;
      tempdata[i][j]=0;
      erase=true;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private int[] setIndexArray(int y,int x) {
    int temp[]=new int[8];
    if(y!=0) temp[0]=dataToken[y-1][x];
    else temp[0]=0;

    if(y!=0 && x!=width-1) temp[1]=dataToken[y-1][x+1];
    else temp[1]=0;

    if(x!=width-1) temp[2]=dataToken[y][x+1];
    else temp[2]=0;

    if(y!=height-1 && x!=width-1) temp[3]=dataToken[y+1][x+1];
    else temp[3]=0;

    if(y!=height-1) temp[4]=dataToken[y+1][x];
    else temp[4]=0;

    if(y!=height-1 && x!=0) temp[5]=dataToken[y+1][x-1];
    else temp[5]=0;

    if(x!=0) temp[6]=dataToken[y][x-1];
    else temp[6]=0;

    if(y!=0 && x!=0) temp[7]=dataToken[y-1][x-1];
    else temp[7]=0;

    // for(int i=0;i<8;i++) System.out.print(temp[i]);
    // System.out.println();
    return temp;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import java.lang.*;

public class TrainFrame extends MyFrame implements ActionListener,Runnable {

```

```

    PictureArea pictureArea;

```

```

    Button loadButton;

```

```

    Button nextButton;

```

```

    Button saveButton;

```

```

    Button trainButton;

```

```

    Button saveBackButton;

```

```

    TheFileDialog theFileDialog;

```

```

    String thePictureFile;

```

```

    TextField patternTextField;

```

```

    Frame frameParentForFileDialog=new Frame();

```

```

    LoadCharacter loadCharacter;

```

```

    FileTrainData fileData;

```

```

    Vector dataCharacter=new Vector();

```

```

    Vector typeCharacter=new Vector();

```

```

    boolean lastCharacter=false;

```

```

    double[] dataToken;

```

```

    int[] displayData;

```

```

    FileNeuralManager fnm=new FileNeuralManager();

```

```

    Backpropagation backpropagation;

```

```

    double errorRate=0.01*Code.SIZE;

```

```

    Thread trainThread;

```

```

    Dialog trainProcess=new Dialog(this,"Train Process");

```

```

    Label dialogLabel=new Label();

```

```

    Label numberHiddenNode=new Label("0");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label completeDialog=new Label();
Button okDialogButton=new Button("OK");

public TrainFrame() {
    super("TRAINING FRAME");
    setLayout(null);
    setSize(600,500);

    loadButton=new Button("Load");
    loadButton.addActionListener(this);
    loadButton.setLocation(20,30);
    loadButton.setSize(70,30);

    nextButton=new Button("Next");
    nextButton.addActionListener(this);
    nextButton.setLocation(61,310);
    nextButton.setSize(50,20);
    nextButton.setEnabled(false);

    saveButton=new Button("Save");
    saveButton.addActionListener(this);
    saveButton.setLocation(5,350);
    saveButton.setSize(110,20);
    saveButton.setEnabled(false);

    trainButton=new Button("Train");
    trainButton.addActionListener(this);
    trainButton.setLocation(5,370);
    trainButton.setSize(110,20);

    saveBackButton=new Button("Save Network");
    saveBackButton.addActionListener(this);
    saveBackButton.setLocation(5,390);
    saveBackButton.setSize(110,20);

```

```
saveBackButton.setEnabled(false);
```

```
patternTextField=new TextField();
```

```
patternTextField.setLocation(10,310);
```

```
patternTextField.setSize(50,20);
```

```
loadCharacter=new LoadCharacter();
```

```
loadCharacter.setLocation(10,200);
```

```
pictureArea=new PictureArea();
```

```
pictureArea.setLocation(120,30);
```

```
trainProcess.setLayout(null);
```

```
trainProcess.setSize(300,120);
```

```
trainProcess.setLocation(200,200);
```

```
Label displayAVE=new Label("Average error per pattern:");
```

```
trainProcess.add(displayAVE);
```

```
displayAVE.setLocation(10,25);
```

```
displayAVE.setSize(150,15);
```

```
trainProcess.add(dialogLabel);
```

```
dialogLabel.setLocation(160,25);
```

```
dialogLabel.setSize(150,15);
```

```
Label displayHN;
```

```
trainProcess.add(displayHN=new Label("Hidden node:"));
```

```
displayHN.setLocation(10,45);
```

```
displayHN.setSize(150,15);
```

```
trainProcess.add(numberHiddenNode);
```

```
numberHiddenNode.setLocation(160,45);
```

```
numberHiddenNode.setSize(150,15);
```

```
Button cancelDialogButton=new Button("CANCEL");
```

```
trainProcess.add(okDialogButton);
```

```
trainProcess.add(cancelDialogButton);
```

```
okDialogButton.setLocation(65,80);
```

```
okDialogButton.setSize(75,20);
```

```

cancelDialogButton.setLocation(150,80);
cancelDialogButton.setSize(75,20);
okDialogButton.addActionListener(this);
okDialogButton.setEnabled(false);
cancelDialogButton.addActionListener(this);
trainProcess.add(completeDialog);
completeDialog.setSize(150,15);
completeDialog.setLocation(65,65);

add(loadButton);
add(nextButton);
add(saveButton);
add(trainButton);
add(saveBackButton);
add(patternTextField);
add(loadCharacter);
add(pictureArea);
this.pack();
}

private void clickLoadButton() {
    theFileDialog=new
    TheFileDialog(frameParentForFileDialog,"Load Picture",FileDialog.LOAD);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        thePictureFile=theFileDialog.getDirectory()+theFileDialog.getFile();
        pictureArea.removeAll();
        pictureArea.loadImage(thePictureFile);
        this.setCursor(new Cursor(Cursor.WAIT_CURSOR));
        pictureArea.segmentation();
        this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));

        loadCharacter.load(pictureArea.getCharacter());
        dataCharacter.addElement(pictureArea.getData());
    }
}

```

```

lastCharacter=false;
nextButton.setEnabled(true);
saveButton.setEnabled(true);
}
}

```

```
private void clickNextButton() {
```

```

    if(!lastCharacter) {
        if(pictureArea.haveAnyCharacter()) {
            loadCharacter.load(pictureArea.getCharacter());
            dataCharacter.addElement(pictureArea.getData());
            typeCharacter.addElement(patternTextField.getText());
            if(!pictureArea.haveAnyCharacter())
                lastCharacter=true;
        }
        else {
            typeCharacter.addElement(patternTextField.getText());
            nextButton.setEnabled(false);
        }
    }
}

```

```
private void clickSaveButton() {
```

```

    if(!lastCharacter)
        typeCharacter.addElement(patternTextField.getText());
    theFileDialog=new TheFileDialog(frameParentForFileDialog,
        "Save Datatoken",FileDialog.SAVE);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        dataToken=(double[])dataCharacter.elementAt(0);
        int numberPerData=dataToken.length;
        fileData =new FileTrainData(theFileDialog.getDirectory()
            ,theFileDialog.getFile(),"rw");
    }
}

```

```

fileData.setNumberPerData(numberPerData);
fileData.setFormat();
String type;
for(int i=0;i<typeCharacter.size();i++) {
    dataToken=(double[])dataCharacter.elementAt(i);
    type=(String)typeCharacter.elementAt(i);
    fileData.save(dataToken,type);
}
typeCharacter.removeAllElements();
dataCharacter.removeAllElements();
}
}

private void clickTrainButton() {
    theFileDialog=new TheFileDialog(frameParentForFileDialog,
        "Load DataFromTrain",FileDialog.LOAD);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        fileData =new FileTrainData(theFileDialog.getDirectory()
            ,theFileDialog.getFile(),"r");
        fmm.loadDataForTrain(fileData);
//        this.setCursor(new Cursor(Cursor.WAIT_CURSOR));
        backpropagation=new Backpropagation(fmm.getNumberInput()
            ,fmm.getNumberOutput());
        if(trainThread==null) {
            trainProcess.show();

            trainThread=new Thread(this);
            trainThread.start();
        }

//        this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
//        saveBackButton.setEnabled(true);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

private void clickSaveBackButton() {
    theFileDialog=new TheFileDialog(frameParentForFileDialog,
        "Save NeuralNetwork",FileDialog.SAVE);
    theFileDialog.show();
    if(theFileDialog.getFile()!=null) {
        FileManager fileData =new FileManager(theFileDialog.getFile(),"rw");
        fnm.saveData(fileData,backpropagation);
    }
}
}

```

```

public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("Load"))
        clickLoadButton();
    else {
        if(e.getActionCommand().equals("Next"))
            clickNextButton();
        else {
            if(e.getActionCommand().equals("Save"))
                clickSaveButton();
            else {
                if(e.getActionCommand().equals("Train"))
                    clickTrainButton();
                else {
                    if(e.getActionCommand().equals("Save Network"))
                        clickSaveBackButton();
                    else {
                        if(e.getActionCommand().equals("CANCEL")) {
                            trainThread.stop();
                            trainProcess.dispose();
                            saveBackButton.setEnabled(true);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

