การวิเคราะห์ลายเซนต์โดยใช้นิวรอนเน็ตเวิร์ก

SIGNATURE ANALYSIS BY USING NEURAL NETWORK

โดย

นาย ปวีณ ศรีทอง รหัส 35104263

นาย ปัญญา ว่องไว รหัส 35104266

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2538

วัน เดือน ปี 29 ก.ค. 2540
เลขทะเบียน 036084
เลขเรียกหนังสือ T 38727 ปวณ ก

ปริญญานิพนธ์ปีการศึกษา 2538

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์ลายเซนต์โดยใช้นิวรอนเน็ตเวิร์ก

ผู้จัดทำ

1. นาย ปวีณ ศรีทอง **รหัสประจำตัว** 35104263

2. นาย ปัญญา ว่องไว **รหัสประจำตัว** 35104266

..... **อาจารย์ที่ปรึกษา**

(ผศ.ดร. บุญวัฒน์ อัตชู)

ปริญญานิพนธ์การวิเคราะห์ลายเซนต์โดยใช้นิวรอนเน็ตเวิร์ก

นักศึกษา	นาย	ปวีณ	ศรีทอง	รหัสประจำตัว	35104263
	นาย	ปัญญา	ว่องไว	รหัสประจำตัว	35104266
อาจารย์ที่ปรึกษา	ผศ.ดร.	บุญวัฒน์		อัตชู	
ปีการศึกษาพ.ศ.	2538				

บทคัดย่อ

ลายเซนต์ถูกใช้เพื่อแทนตัวบุคคลมาเป็นเวลานานมีการทดลองเพื่อใช้คอมพิวเตอร์ในการตรวจสอบแทนที่จะใช้มนุษย์เพราะคอมพิวเตอร์มีความแม่นยำสูงกว่า ส่วนที่ยากที่สุดคือการทำให้ระบบมีความยืดหยุ่นเนื่องจากลายเซนต์ของคนจะไม่เหมือนกันทุกครั้งที่เซนต์แต่จะคล้ายๆกัน ความคล้ายนี้จะขึ้นกับความบ่อยของการใช้ลายเซนต์ของผู้เซนต์และระยะเวลาจากการเซนต์ครั้งสุดท้าย

โครงงานนี้ใช้ GRNN มาประยุกต์ใช้ในลักษณะของ mapping function จากอินพุทมายังเอาท์พุทเพื่อทำการพิสูจน์ลายเซนต์ โดยพารามิเตอร์แต่ละตัวและค่า threshold ของระบบจะหาจากข้อมูลลักษณะเด่นของลายเซนต์แต่ละลายเซนต์จากลายเซนต์ตัวอย่างที่เก็บไว้ในฐานข้อมูลเพื่อใช้ได้เหมาะสมกับแต่ละลายเซนต์ที่เราพิสูจน์ ระบบนี้เป็นระบบ on-line ข้อมูลจะได้จากdigitizer และ stylus pen ในรูปของคู่ลำดับ x y จากการทดลองพบว่าความถูกต้องในการตรวจสอบลายเซนต์ของเจ้าของลายเซนต์แท้จริงพบว่ามีความถูกต้อง 90 เปอร์เซ็นต์ และความถูกต้องในการตรวจสอบลายเซนต์ปลอมมีประมาณ 76 เปอร์เซ็นต์

Thesis Title **Signature Analysis using by neural network**

Student **Mr. Paween Sritong**

and Mr. Panya Wongwai

Thesis Advisor **Assisstant Prof. Dr. Boonwat Attachoo**

Year **1996**

Abstract

Signature has been used to indentify person for a long time.Many attemp has been made to verify signature by computer instead of human because computer is much more accurate.But the hardest part is to make the system flexible because one's signatures is not indentical ,they are similar and how similar they are is depended on how often the signer use it and how long he sign it for the last time till now.

This project uses GRNN as mapping Function from input to output to verify signature.Each parameter of the network and system threshold is found by features of the signatures in the database themselves. Because each one person's signatures are not the same as the others.This project is the on-line system it means that every input is recieved on-line via digitizer and stylus pen in forms of x and y coordinates.We obtain 90 percents acceptance of real signature and 76 percents rejectance of fake signatures.

สารบัญ

บท	หน้า
บทที่ 1 บทนำ	1
1.1 อุปกรณ์ที่ใช้ในโครงงานพิสูจน์ลายเซนต์	2
1.2 การเชื่อมต่ออุปกรณ์ที่ใช้งานในโปรเจค	3
1.3 ลักษณะข้อมูลอินพุทของอุปกรณ์ที่รับข้อมูลลายเซนต์	4
บทที่ 2 ทฤษฎี	6
2.1 การเตรียมข้อมูลอินพุท(preprocessing)	6
2.1.1 การหาความเอียงของลายเซนต์และการหมุนลายเซนต์	7
2.1.2 การทำให้ลายเซนต์อยู่ในขนาดมาตราฐาน	12
2.1.3 การเพิ่มข้อมูลจุดในลายเซนต์	15
2.2 นิวรอนเน็ตเวิร์ก	26
2.2.1 แนวคิดพื้นฐานของนิวรอนเน็ตเวิร์ก	26
2.2.2 ทฤษฎี Radial Basis Function	33
2.2.3 การเลือกลักษณะเด่น	37
2.2.4 การตัดสินใจ	38
บทที่ 3 การคำนวณและสร้าง	39
3.1 ลักษณะเด่นของลายเซนต์ที่ใช้	39
3.2 ลักษณะฐานข้อมูลที่ใช้	41
บทที่ 4 การทดลองและผลการทดลอง	44
บทที่ 5 สรุปและวิจารณ์การทดลอง	45
ภาคผนวก ก. คู่มือการใช้งานโปรแกรม	46
ภาคผนวก ข. ไดอะแกรม (Flow chart) ของแนวคิดและการทำงาน	56
กฤติกรรมประกาศ	60
เอกสารอ้างอิง	61

บทที่ 1
บทนำ

ที่มาและความสำคัญของปัญหา

ในปัจจุบันได้มีการนำเอาคอมพิวเตอร์มาใช้ในงานต่างๆ มากมาย เพื่อช่วยเพิ่มประสิทธิภาพ และความรวดเร็วในการทำงาน จุดหนึ่งที่ควรมีการนำคอมพิวเตอร์มาช่วยในการทำงานก็คือ การตรวจสอบลายเซนต์ ซึ่งเป็นวิธีการพิสูจน์ตัวบุคคลที่นิยมใช้กันทั่วไป ไม่ว่าจะเป็นการติดต่อกับหน่วยงาน สถาบันต่างๆ แต่ในปัจจุบันการตรวจสอบลายเซนต์ยังคงทำด้วยสายตา ซึ่งบางครั้งเกิดความล่าช้าและเกิดความผิดพลาดได้ ดังนั้นเพื่อความสะดวกและความถูกต้องในการตรวจสอบลายเซนต์ จึงควรนำคอมพิวเตอร์ซึ่งเป็นเทคโนโลยีที่มีประสิทธิภาพมาใช้ นอกจากนี้ ความเจริญเติบโตอย่างรวดเร็วทางเทคโนโลยีทางคอมพิวเตอร์ยังทำให้เรามีอุปกรณ์ต่อพ่วงที่มีประสิทธิภาพมาช่วยเพิ่มความสะดวกในการทำงานอีกด้วย ได้แก่ กระดาษอิเล็กทรอนิกส์ (DIGITIZER) ซึ่งเป็นอุปกรณ์รับข้อมูลเข้าสู่เครื่องคอมพิวเตอร์แบบออนไลน์ (On-line) ทำให้ได้ข้อมูลสำหรับใช้ในการการตรวจสอบลายเซนต์มากพอสมควรสามารถเทียบใช้เวลาได้ ซึ่งจะช่วยเพิ่มประสิทธิภาพในการตรวจสอบและโอกาสในการเกิดข้อผิดพลาดลงได้

1.1 อุปกรณ์ที่ใช้ในโครงงานการพิสูจน์ลายเซนต์ที่เกี่ยวข้องกับการรับอินพุทมีดังนี้

1. เครื่องคอมพิวเตอร์ (PC) ที่อยู่ในลักษณะ stand alone เพราะต้องมีการใช้ device driver สำหรับ digitizer ซึ่งใช้งานในโปรแกรมวินโดว์ โดยเครื่องคอมพิวเตอร์นี้ต้องสามารถใช้งานกับโปรแกรมวินโดว์ได้ด้วย และต้องมี serial port สำหรับการต่อใช้งานกับ digitizer ด้วย

2. เครื่อง digitizer ใช้สำหรับรับค่าอินพุทข้อมูลลายเซนต์(ตำแหน่งจุดลายเซนต์)ที่ถูกเซนต์โดยปากกาแสง

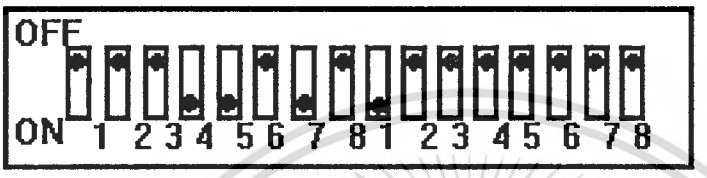
3. ปากกาแสงใช้สำหรับทำการเซนต์ลายเซนต์ให้อินพุทลายเซนต์แก่ digitizer โดยปากกาแสงจะต่ออยู่กับ digitizer ด้วย

4. ภาษา Delphi ที่ใช้งานกับโปรแกรมวินโดว์ใช้สำหรับสร้าง โปรแกรมที่ใช้ในการการพิสูจน์ลายเซนต์บนวินโดว์

1.2การใช้งานอุปกรณ์แต่ละชนิดที่ต้องใช้และนำอุปกรณ์เหล่านั้นมาทำการเชื่อมต่อเข้าด้วยกันซึ่งทำการเชื่อมต่อดังนี้

- ทำการเชื่อมต่อ digitizer เข้ากับเครื่องคอมพิวเตอร์โดยต่อสายข้อมูลจาก digitizer เข้ากับพอร์ตอนุกรมของเครื่องคอมพิวเตอร์

- ทำการตั้งค่า dip switch ของ digitizer ให้มีลักษณะตามรูปข้างล่าง



ซึ่งการตั้งค่า DIP SWITCH เช่นนี้มีความหมายดังนี้

- BAUD RATE = 9600 bps.
- มีการตรวจสอบพาริตี้บิตคี่
- มีการใช้ XOFF / XON
- ส่งข้อมูลทีละ 8 บิต

- ทำการ setup โปรแกรมวินโดว์ โดยแทน device driver ของเมาส์ด้วยการเลือก OTHER MOUSE .. ภายในเมนูที่มีให้เลือกไดรเวอร์ แต่เราเลือกที่เมนูนี้เพราะเรายังเคยใช้ device driver นี้กับวินโดว์หลังจากนั้นวินโดว์ก็จะถามถึง PATH ของไฟล์ device driver ว่าอยู่ที่ไหนก็ให้ใส่ไปเช่น C:\wintab\ ซึ่งภายในไดเร็กทอรี่ wintab ของไดรว์ C นี้จะมีไฟล์ชื่อ mwtoce.drv ซึ่งเป็นไฟล์ device driver ของ digitizer อยู่ เมื่อทำการ setup วินโดว์ลักษณะนี้แล้วเราสามารถเปิด digitizer และใช้โปรแกรมวินโดว์ได้เลย โดยการทำงานของเมาส์จะถูกแทนที่ด้วยการใช้งานปากกาแสงบน digitizer แทนโดยเมื่อขยับปากกาแสงจะทำให้ CURSOR ของเมาส์ขยับและเมื่อกดปากกาแสงจะเหมือนกับกดเมาส์ เมื่อเราจะใช้งานเมาส์ตามปกติเราก็จะต้องทำการ setup วินโดว์ใหม่โดยตั้ง driver ให้เป็นของเมาส์ดังเดิม

1.3 ลักษณะอินพุทของอุปกรณ์ที่ทำการรับข้อมูลลายเซนต์

จากการที่เราใช้ Delphi เราสามารถใช้ฟังก์ชั่นเกี่ยวกับเมาส์เพื่อใช้งานกับการอ่านตำแหน่งของ digitizer ได้ โดยใช้ที่ใช้หาตำแหน่งของเมาส์ เพื่อหาตำแหน่งของปากกาแสงได้โดยได้เป็นตำแหน่ง X , Y ของตำแหน่งที่ปากกาลากลายเซนต์ในหน้าจอของโปรแกรมในวินโดว์แต่ข้อมูลที่ได้มีความละเอียดน้อย จึงพยายามเพิ่มความละเอียดโดยวิธีต่อไปนี้

ทดลองปรับ DIP switch ของ digitizer ให้มี baud rate = 19200 แต่ไม่ได้ผลเป็นที่น่าพอใจ

ในที่สุดเราก็ทำการเพิ่มจุดข้อมูลในแนวลายเซนต์เอาเองซึ่งเป็นขั้นตอน normalize ขั้นตอนหนึ่งซึ่งจะอธิบายภายหลัง

สำหรับข้อมูลที่ได้จาก digitizer ที่ยังไม่ได้เพิ่มจุดเป็นดังนี้ตามลายเซนต์ตัวอย่าง

ข้อมูลในลักษณะการตำแหน่งในแนวแกน X และแนวแกน Y เป็นดังนี้

ค่าพิกัดในแนวแกน X	ค่าพิกัดในแนวแกน Y
140.00000	88.00000
141.00000	88.00000
141.00000	87.00000
141.00000	84.00000
141.00000	80.00000
141.00000	76.00000
141.00000	74.00000
141.00000	72.00000
141.00000	70.00000
141.00000	66.00000
141.00000	64.00000
141.00000	60.00000
142.00000	58.00000
142.00000	57.00000
143.00000	57.00000
144.00000	57.00000
145.00000	58.00000
147.00000	60.00000
148.00000	62.00000

150.00000	66.00000
151.00000	70.00000
152.00000	71.00000
153.00000	71.00000
153.00000	70.00000
155.00000	66.00000
156.00000	62.00000
158.00000	58.00000
159.00000	56.00000
161.00000	54.00000
163.00000	52.00000
164.00000	51.00000
165.00000	50.00000
166.00000	50.00000
167.00000	50.00000
168.00000	52.00000
170.00000	55.00000
171.00000	58.00000
172.00000	62.00000
174.00000	66.00000
174.00000	67.00000
175.00000	69.00000
175.00000	70.00000

บทที่ 2
ทฤษฎี

2.1 การทำการเตรียมข้อมูลอินพุท (Preprocessing)

การนอมัลไลซ์ (normalization)ข้อมูลอินพุทคือ การลด error ที่มีในข้อมูลอินพุทเพื่อที่จะได้ข้อมูลที่จะไปทำการประมวลผลได้ง่ายและถูกต้อง

การนอมัลไลซ์ข้อมูลอินพุทในการพิสูจน์ลายเซนต์มีดังนี้

1. การหาความเอียงของลายเซนต์และการหมุนลายเซนต์ให้เอียงน้อยที่สุด
2. การทำให้ลายเซนต์อยู่ในขนาดมาตราฐาน
3. การเพิ่มข้อมูลจุดในลายเซนต์

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

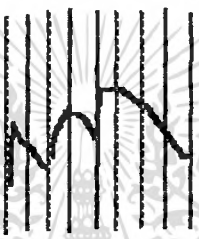
2.1.1 การหาความเอียงของลายเซนต์และการหมุนลายเซนต์ให้เอียงน้อยที่สุด

เนื่องจากคนเราเซนต์แต่ละครั้งไม่สามารถที่จะเซนต์โดยเอียงเป็นมุมที่ใกล้เคียงกันได้ ดังนั้นเราจึงต้องวิธีที่จะทำให้ลายเซนต์ที่แต่ละครั้งในการเซนต์ของคนมีความเอียงแตกต่างกันน้อยที่สุดเพื่อจะทำให้การเปรียบเทียบลายเซนต์ได้ง่ายขึ้น

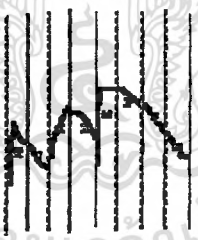
● การหาความเอียงของลายเซนต์

เราหาความเอียงของลายเซนต์เพื่อที่จะได้ทำให้ลายเซนต์หายเอียงได้โดยมีขั้นตอนการหาความเอียงของลายเซนต์ดังนี้

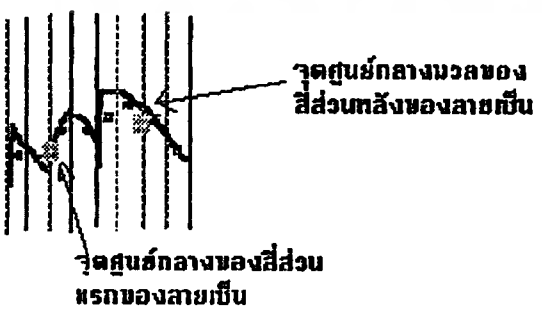
- ทำการแบ่งส่วนลายเซนต์ออกเป็น แปดส่วนเท่าๆ กันตามความยาวของลายเซนต์ตามรูปข้างล่าง



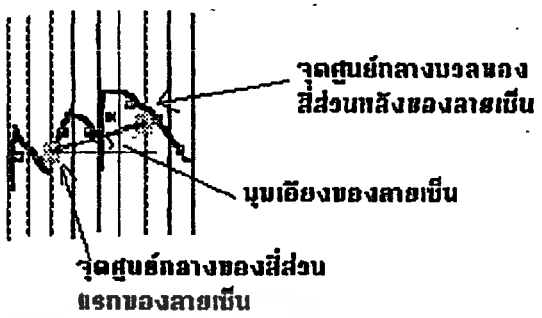
- หาจุดศูนย์กลางมวลของแต่ละส่วนของลายเซนต์ โดยตำแหน่งจุดศูนย์กลางมวลหาจากผลรวมของตำแหน่งของจุดตามค่าในแกน x และในแกน y ที่ถูกเขียนลายเซนต์ภายในพื้นที่นั้นเฉลี่ยได้ค่าเฉลี่ยเป็นค่าตำแหน่งจุดศูนย์กลางดังรูปข้างล่าง



- หาตำแหน่งจุดศูนย์กลางมวลของพื้นที่สี่ส่วนแรกที่ติดกันของลายเซนต์ และหาตำแหน่งจุดศูนย์กลางมวลของพื้นที่สี่ส่วนหลัง โดยหาจากค่าเฉลี่ยของจุดศูนย์กลางมวลของสี่ส่วนที่ติดกันนั้น



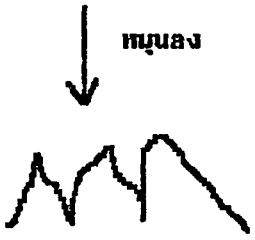
- หาความเอียงของเส้นที่เชื่อมต่อระหว่างจุดศูนย์กลางของสี่ส่วนแรกและหลังของลายเซนต์เมื่อหาก็จะสามารถหาความชันหรือความเอียงของลายเซนต์ ที่สามารถนำมาคำนวณหาการหมุนลายเซนต์ได้



●การหมุนลายเซนต์ให้เอียงน้อยที่สุด

ก่อนอื่นต้องดูก่อนว่าลายเซนต์นั้นเอียงมากพอที่จะหมุนไหม ซึ่งมีวิธีการหมุนลายเซนต์ได้ตามขั้นตอนดังนี้

- ดูว่าลายเซนต์เอียงเป็นมุมเท่าไหร่ ถ้าเอียงเป็นมุมที่มีค่ามุมเอียงเป็นบวกแต่มีค่าไม่ถึง 90 องศา(เอียงขึ้น) ก็จะทำการหมุนลายเซนต์ลง โดยจะต้องหาจุดอ้างอิงที่อยู่ในแนวเส้นที่ใช้วัดความเอียง(เส้นที่ลากผ่านจุดศูนย์กลางมวลของสี่ส่วนแรกและหลังของลายเซนต์) ที่มีค่าความสูงของจุดน้อยกว่าจุดต่ำสุดของลายเซนต์และมีค่าตามแนวนอนน้อยกว่าค่าต่ำสุดของค่า x ที่มีในลายเซนต์ เพื่อเป็นจุดเปรียบการหมุนโดยจะใช้คำนวณหาระยะทางจากจุดนี้ไปยังจุดในลายเซนต์ทีละจุด และใช้หามุมระหว่างจุดนี้กับจุดของลายเซนต์ เพื่อนำมุมที่คำนวณได้มาลบด้วยมุมความเอียงของลายเซนต์แล้วนำมุมผลลัพธ์ที่ได้จากการลบนั้นมาหาจุดเทียบกับกับจุดอ้างอิงโดยยึดเอาระยะทางจากจุดเท่าเดิมแต่มุมเปลี่ยนไปเนื่องจากการหมุน



ถ้าลายเซนต์เอียงเป็นมุมลบ(ลายเซนต์เอียงลง) ก็ ต้องทำการหมุนลายเซนต์ขึ้น โดยต้องหาจุดอ้างอิงในการทำการหมุน โดยจุดอ้างอิงจะมีค่าในตำแหน่งตามแกน Y มากกว่าค่าสูงสุดตามแกน Y ของจุดในลายเซน

เซนต์และมีค่าตามแนวนอนน้อยกว่าค่าต่ำสุดของค่า x ที่มีในลายเซนต์ เพื่อเป็นจุดเปรียบการหมุนโดยจะใช้คำนวณหาระยะทางจากจุดนี้ไปยังจุดในลายเซนต์ทีละจุด และใช้หามุมระหว่างจุดนี้กับจุดของลายเซนต์ เพื่อนำมุมที่คำนวณได้มาบวกกับมุมความเอียงของลายเซนต์แล้วนำมุมผลลัพธ์ที่ได้จากการลบนั้นมาหาจุดเทียบกับกับจุดอ้างอิงโดยยึดเอาระยะทางจากจุดเท่าเดิมแต่มุมเปลี่ยนไปเนื่องจากการหมุน

ในกรณีที่มุมเอียง 90 องศา เราจะกำหนดจุดอ้างอิงเป็นจุดอยู่ต่ำกว่าจุดต่ำสุดของลายเซนต์ และทำการเปรียบเทียบการหมุนคล้ายกับกรณีก่อนหน้านี้

ตัวอย่างข้อมูลหลังจากที่ทำการรับอินพุทตามตัวอย่างการรับข้อมูลก่อนหน้านี้มาแล้วทำการหมุนลายเซนต์เป็นดังข้างล่างนี้

ภาพลายเซนต์หลังจากหมุน

ข้อมูลของจุดในลายเซนต์ตามตำแหน่งในแนวนอนและแนวตั้ง

ค่าพิกัดในแนวแกน X	ค่าพิกัดในแนวแกน Y
214.00000	61.00000
215.00000	62.00000
215.00000	61.00000
217.00000	58.00000
219.00000	55.00000
221.00000	51.00000

222.00000	50.00000
223.00000	48.00000
224.00000	46.00000
227.00000	43.00000
228.00000	41.00000
230.00000	38.00000
232.00000	37.00000
232.00000	36.00000
233.00000	36.00000
234.00000	37.00000
234.00000	38.00000
235.00000	41.00000
235.00000	43.00000
234.00000	48.00000
233.00000	52.00000
233.00000	53.00000
234.00000	54.00000
235.00000	53.00000
238.00000	50.00000
241.00000	48.00000
245.00000	45.00000
247.00000	44.00000
250.00000	43.00000
253.00000	43.00000
254.00000	42.00000
255.00000	42.00000
256.00000	43.00000
257.00000	43.00000
257.00000	45.00000
257.00000	49.00000
256.00000	52.00000
255.00000	56.00000
254.00000	60.00000

254.00000	61.00000
254.00000	64.00000
253.00000	64.00000

2.1.2 การทำให้ลายเซนต์อยู่ในขนาดมาตราฐาน

การทำให้ลายเซนต์อยู่ในขนาดมาตราฐาน คือ การที่ทำให้ทุกลายเซนต์อยู่ในขนาดเดียวกันโดยใช้หลักการเทียบสัดส่วน

ในการทำให้ลายเซนต์มีขนาดมาตราฐานเราจะให้ขนาดมาตราฐานตามแนวนอนเป็น 200 และในแนวตั้งเป็น 60 โดยใช้วิธีการ non - linear normalize มีขั้นตอนดังต่อไปนี้

1. กำหนดขนาดของภาพลายเซนต์กว้าง M ยาว N โดยแต่ละจุดในภาพลายเซนต์จะมีฟังก์ชั่น f(x,y) เป็นฟังก์ชั่นแบบ binary ระบุว่าลายเซนต์ที่จุด (x,y) ในภาพลายเซนต์นั้นถูกปากกาเซนต์หรือไม่ ถ้าถูกเซนต์จะมีค่าเอาท์พุทของฟังก์ชั่นเป็น 1 ถ้าไม่ถูกเซนต์จะมีค่าเอาท์พุทเป็น 0

2. หาค่าของฟังก์ชั่นในแนวนอน H(i) ของแต่ละจุด i ในระนาบที่ให้เซนต์ลายเซนต์โดย

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

โดย i เป็นค่าระบุตำแหน่งในแนวนอนของจุด และ j เป็นค่าระบุตำแหน่งในแนวตั้งของจุดในระนาบที่ให้เซนต์ลายเซนต์ N คือ จุดที่มีค่ามากที่สุดในแนวตั้งของพื้นที่ที่เซนต์ และ α_H เป็นค่าคงที่ซึ่งในทฤษฎีกำหนดให้เป็น 0 หรือ 2 แต่เราใช้ 0

หาค่าของฟังก์ชั่นในแนวตั้ง V(j) ของแต่ละจุด j ในระนาบที่ให้เซนต์ลายเซนต์โดย

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

โดย i เป็นค่าระบุตำแหน่งในแนวนอนของจุด และ j เป็นค่าระบุตำแหน่งในแนวตั้งของจุดในระนาบที่ให้เซนต์ลายเซนต์ M คือ จุดที่มีค่ามากที่สุดในแนวนอนของพื้นที่ที่เซนต์ และ α_V เป็นค่าคงที่ซึ่งในทฤษฎีกำหนดให้เป็น 0 หรือ 2 แต่เราใช้ 0

3. ทำการหาตำแหน่งของจุดหลังจากที่ทำการ normalize ขนาดแล้วโดยหาได้จากสูตรดังต่อไปนี้

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

โดยที่

X คือ ตำแหน่งตามแนวนอนของจุดหลังถูกแปลงให้อยู่ในขนาดมาตราฐาน

x คือ ตำแหน่งตามแนวนอนของจุดก่อนถูกแปลงให้อยู่ในขนาดมาตราฐาน

M คือ จุดที่มีค่ามากที่สุดในแนวนอนของพื้นที่ที่เซนต์

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

โดยที่

Y คือ ตำแหน่งตามแนวตั้งของจุดหลังถูกแปลงให้อยู่ในขนาดมาตราฐาน

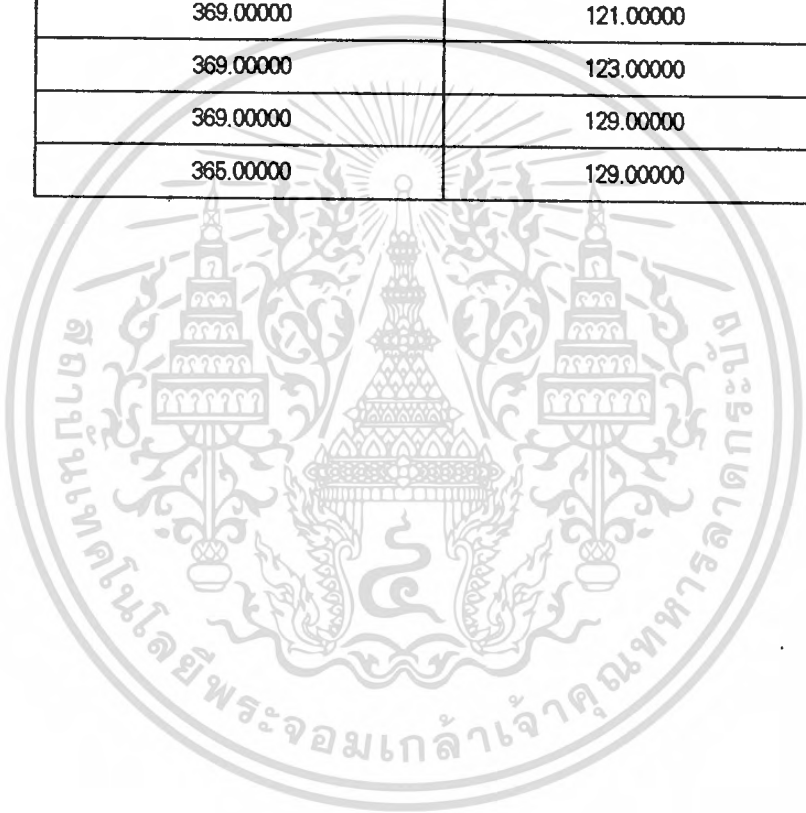
y คือ ตำแหน่งตามแนวตั้งของจุดก่อนถูกแปลงให้อยู่ในขนาดมาตราฐาน

N คือ จุดที่มีค่ามากที่สุดในแนวตั้ งของพื้นที่ที่เซนต์

จะมีข้อมูลที่ตำแหน่งแนวนอนและแนวตั้งดังนี้

202.00000	123.00000
206.00000	125.00000
206.00000	123.00000
216.00000	117.00000
223.00000	111.00000
231.00000	104.00000
235.00000	102.00000
240.00000	98.00000
244.00000	94.00000
256.00000	88.00000
261.00000	84.00000
269.00000	79.00000
277.00000	77.00000
277.00000	75.00000
282.00000	75.00000
286.00000	77.00000
286.00000	79.00000
290.00000	84.00000
290.00000	88.00000
286.00000	98.00000
282.00000	106.00000
282.00000	108.00000
286.00000	110.00000
290.00000	108.00000
302.00000	102.00000
315.00000	98.00000
332.00000	92.00000
340.00000	90.00000
353.00000	88.00000
365.00000	88.00000
369.00000	86.00000

374.00000	86.00000
378.00000	88.00000
382.00000	88.00000
382.00000	92.00000
382.00000	100.00000
378.00000	106.00000
374.00000	113.00000
369.00000	121.00000
369.00000	123.00000
369.00000	129.00000
365.00000	129.00000



2.1.3 การเพิ่มข้อมูลจุดในลายเซนต์

เนื่องจากข้อมูลลายเซนต์ที่ได้แต่ละครั้งมีปริมาณเพื่อจะได้มีข้อมูลที่เท่ากันในการประมวลผลในนิวรอนเน็ตเวิร์กเพราะนิวรอนเน็ตเวิร์กที่ใช้ในการพิสูจน์ลายเซนต์นี้ถ้าข้อมูลมีปริมาณไม่เท่ากันจะประมวลผลได้ไม่ค่อยดี

หลักการของการเพิ่มจุด

- ทำการจับเวลาที่ใช้ในการเซนต์ลายเซนต์

- ทำการแบ่งเวลาเช่น แบ่งเวลาเป็น 250 ช่วงที่เท่าๆ กัน ตัวอย่างเช่น ลายเซนต์ใช้เวลาในการเซนต์ 250 วินาที จะแบ่งช่วงเวลา 250 ช่วงได้ช่วงละ 1 วินาที

- ให้จุดแรกของลายเซนต์เดิมเป็นจุดแรกของลายเซนต์ที่เพิ่มจุดเข้าไปใหม่ แล้วหาจุดต่อไปเมื่อเวลาผ่านไปหนึ่งช่วงจากลายเซนต์เดิมนั้นซึ่งอาจจะไม่ใช่จุดในลายเซนต์เดิมนั้นแต่จะเป็นที่คิดขึ้นใหม่ก็ได้โดยจุดนั้นต้องอยู่ในแนวเส้นที่ลากเพื่อเซนต์ของลายเซนต์เดิมดังรูปข้าง

ดังรูปเวลาผ่านในช่วงเวลาที่เราหาได้ในช่วงเวลาหนึ่งแล้ว เราจะได้จุดที่หาได้ในช่วงเวลาอันแรกอยู่ในแนวของแนวเส้นลายเซนต์เดิมที่เชื่อมต่อระหว่างจุดเริ่มต้นและจุดถัดไปในลายเซนต์โดยเราหาจุดนี้ได้จาก การหาความเร็วระหว่างจุดทั้งสองในลายเซนต์(หาได้โดยหาระยะทางระหว่างจุดทั้งสองในลายเซนต์แล้วหารด้วยเวลาระหว่างจุดทั้งสองในลายเซนต์นั้น)แล้วนำความเร็วนั้นไปคูณด้วยช่วงเวลาที่หาได้ก็จะได้ระยะทางในช่วงเวลานั้นแล้วค่อยไปหาตำแหน่งของจุดอีกที สำหรับจุดในช่วงเวลาอันถัดมาอันที่สองก็ทำได้ในลักษณะเดียวกัน แต่ในจุดที่หาได้จากช่วงเวลาอันที่สามนี้เวลาระหว่างจุดที่หาได้ในช่วงเวลาอันที่สองไปยังจุดถัดไปจากจุดเริ่มต้นของลายเซนต์นั้นมีค่าน้อยกว่าเวลาในช่วงเวลาที่คิดได้ ดังนั้นมันยังต้องไล่หาเวลาไปตามแนวเส้นลายเซนต์ที่เลยจุดที่สองของลายเซนต์ไปอีกเป็นเวลา = เวลาในหนึ่งช่วงเวลา - เวลาระหว่างจุดที่หาได้ในช่วงเวลาอันที่สองและจุดที่สองของลายเซนต์ จะได้เป็นเวลาส่วนที่เหลือสำหรับหาจุดในช่วงเวลา ซึ่งเราหาจุดที่หาได้ในช่วงเวลาอันที่สามได้จากการหาความเร็วระหว่างจุดที่สองและจุดที่สามในลายเซนต์นั้น แล้วคูณด้วยเวลาส่วนที่เหลือสำหรับหาจุดในช่วงเวลา เราก็จะได้จุดที่หาได้ในช่วงเวลาอันที่สามซึ่งอยู่ในแนวเส้นระหว่างจุดที่สองและจุดที่สามในลายเซนต์ซึ่งเราจะเอาข้อมูลจุดเริ่มต้นของลายเซนต์และจุดแต่ละอันที่หาได้จากช่วงเวลาเก็บไว้

ประมวลผลเท่านั้น ดังนั้นบางจุดในลายเซนต์ที่จึงถูกตัดไปซึ่งจะทำให้แนวเส้นลายเซนต์เปลี่ยนไปเล็กน้อยเช่นดังรูป แต่จะไม่ผลเสียหายต่อการพิสูจน์ลายเซนต์เพราะช่วงเวลาที่เราหาได้นั้นมีค่าน้อยกว่าเวลาในระหว่างจุดของลายเซนต์ที่วัดได้จาก digitizer มาก ส่วนที่จุดอื่นๆ ในลายเซนต์ก็ทำการเพิ่มจุดในลักษณะที่คล้ายกัน

ข้อมูลลายเซนต์ที่ทำการเพิ่มจุดในแนวนอนและแนวตั้งมีดังนี้และได้ลักษณะของลายเซนต์ที่เพิ่มจุดดังรูปข้างล่าง

ซึ่งข้อมูลที่ได้หลังจากการเพิ่มจุดในแนวแกน Xและแนวแกน Y เป็นดังนี้

ค่าพิกัดในแนวแกน X	ค่าพิกัดในแนวแกน Y
1.00000	49.00000
1.38286	49.19143
1.76573	49.38286
2.14859	49.57430
2.53146	49.76573
2.91432	49.95716
3.29719	50.14859
3.68005	50.34003
4.06292	50.53146
4.44578	50.72289
4.82865	50.91432
4.83952	50.79285
4.85920	50.57297
4.87888	50.35310
4.89857	50.13322
4.91825	49.91334
4.93793	49.69346
4.95761	49.47358
4.97729	49.25371
4.99697	49.03383
5.72613	48.54614
6.58787	47.96761
7.44960	47.39007
8.31134	46.81254
9.17308	46.23500
10.03481	45.65747
10.89655	45.07993
11.75828	44.50240
12.62002	43.92486
13.48175	43.34733
22.36627	36.67952

23.28514	35.87650
24.20402	35.07149
25.12289	34.26747
26.04177	33.46345
26.96064	32.65944
27.87952	31.85542
28.79839	31.05141
29.71727	30.24739
39.26506	23.73494
39.64793	23.35207
40.03079	22.96921
40.41365	22.58635
40.79652	22.20348
41.17938	21.82062
41.56225	21.43775
41.94511	21.05489
42.32798	20.67202
42.71084	20.28916
55.14056	13.88755
55.71486	13.42811
56.28916	12.96867
56.86345	12.50924
57.43775	12.04980
58.01205	11.59036
58.58635	11.13092
59.16064	10.67149
59.73494	10.21205
68.41232	4.89692
69.17805	4.70549
69.94378	4.51406
70.70950	4.32262
71.47523	4.13119

72.24096	3.93976
73.00669	3.74833
73.77242	3.55689
74.53815	3.36546
75.30388	3.17403
76.01187	1.00000
76.14239	1.00000
76.27291	1.00000
76.40343	1.00000
76.53395	1.00000
76.66448	1.00000
76.79500	1.00000
76.92552	1.00000
77.05604	1.00000
77.18656	1.00000
77.31709	1.00000
77.44761	1.00000
77.57813	1.00000
77.70865	1.00000
77.83917	1.00000
77.96970	1.00000
78.10022	1.00000
78.23074	1.00000
78.36126	1.00000
78.49179	1.00000
78.62231	1.00000
78.75283	1.00000
78.88335	1.00000
79.01387	1.00000
79.14440	1.00000
79.27492	1.00000
79.40544	1.00000

79.53596	1.00000
79.66648	1.00000
79.79701	1.00000
79.92753	1.00000
80.05805	1.00000
80.18857	1.00000
80.31909	1.00000
80.44962	1.00000
80.58014	1.00000
80.71066	1.00000
80.84118	1.00000
80.97171	1.00000
85.00000	3.17992
85.00000	3.40964
85.00000	3.63936
85.00000	3.86908
85.00000	4.09880
85.00000	4.32851
85.00000	4.55823
85.00000	4.78795
88.97055	14.07363
88.58768	15.03079
88.20482	15.98795
87.82195	16.94511
87.43909	17.90228
87.05622	18.85944
86.67336	19.81660
86.29050	20.77376
85.90763	21.73092
85.52477	22.68809
85.14190	23.64525
85.06572	35.96714

85.17014	35.91493
85.27455	35.86272
85.37897	35.81051
85.48339	35.75831
85.58781	35.70610
85.69222	35.65389
85.79664	35.60168
85.90106	35.54947
86.00548	35.49726
86.10989	35.44505
86.21431	35.39284
86.31873	35.34064
86.42315	35.28843
86.52756	35.23622
86.63198	35.18401
86.73640	35.13180
86.84082	35.07959
86.94524	35.02738
87.04965	34.97517
87.15407	34.92296
87.25849	34.87076
87.36291	34.81855
87.46732	34.76634
87.57174	34.71413
87.67616	34.66192
87.78058	34.60971
87.88499	34.55750
87.98941	34.50529
88.09383	34.45309
88.19825	34.40088
88.30267	34.34867
88.40708	34.29646

88.51150	34.24425
88.61592	34.19204
88.72034	34.13983
88.82475	34.08762
88.92917	34.03541
101.48032	27.85221
102.97349	27.39277
104.46667	26.93333
105.95984	26.47390
107.45301	26.01446
108.94618	25.55502
110.43936	25.09558
111.93253	24.63614
113.42570	24.17671
131.47122	17.88220
132.23695	17.69076
133.00268	17.49933
133.76841	17.30790
134.53414	17.11647
135.29987	16.92503
136.06560	16.73360
136.83133	16.54217
137.59705	16.35074
138.36278	16.15930
152.23133	14.00000
153.60964	14.00000
154.98795	14.00000
156.36627	14.00000
157.74458	14.00000
159.12289	14.00000
160.50120	14.00000
161.87952	14.00000

163.25783	14.00000
168.26506	12.00000
168.83936	12.00000
169.41365	12.00000
169.98795	12.00000
170.56225	12.00000
171.13655	12.00000
171.71084	12.00000
172.28514	12.00000
172.85944	12.00000
173.16876	12.14458
173.55507	12.33601
173.95139	12.52744
174.34771	12.71888
174.74403	12.91031
175.14035	13.10174
175.53667	13.29317
175.93299	13.48461
176.32931	13.67604
176.72563	13.86747
181.00000	14.14137
181.00000	14.60080
181.00000	15.06024
181.00000	15.51968
181.00000	15.97912
181.00000	16.43855
181.00000	16.89799
181.00000	17.35743
181.00000	17.81687
180.76975	26.34536
180.38688	26.91968
180.00402	27.49398

179.62115	28.06827
179.23829	28.64257
178.85542	29.21687
178.47256	29.79116
178.08969	30.36546
177.70683	30.93976
177.32396	31.51406
168.00000	47.03534
168.00000	47.26506
168.00000	47.49478
168.00000	47.72450
168.00000	47.95422
168.00000	48.18394
168.00000	48.41365
168.00000	48.64337
168.00000	48.87309
167.82865	55.00000
167.44578	55.00000
167.06292	55.00000
166.68005	55.00000
166.29719	55.00000
165.91432	55.00000
165.53146	55.00000
165.14859	55.00000
164.76573	55.00000
164.38286	55.00000
164.00000	55.00000

ผลจากการเพิ่มข้อมูลจุดจะทำให้คิดข้อมูลอื่นๆ ที่มาจากข้อมูลจุดได้ละเอียดยิ่งขึ้น และทำให้สามารถตรวจสอบลายเซนต์ได้ละเอียดขึ้นมากกว่าเดิมช่วยเพิ่มประสิทธิภาพในการป้องกันลายเซนต์ปลอมได้ดีขึ้น

2.2 นิวรอนเน็ตเวิร์ก

2.2.1 แนวคิดพื้นฐานของ Neural Network

Information Processing

Normal ในทางชีววิทยานั้นส่งสัญญาณทางไฟฟ้าเคมี(Electrochemical) ผ่าน Neuron อื่นๆ เป็นทางไป แต่ละ Neuron รับสัญญาณจาก Neuron อื่นๆ เป็นทางไปแต่ละ Neuron รับสัญญาณจาก Neuronอื่นๆ ทางจุดเชื่อมต่อพิเศษเรียกว่า Synapses บางอินพุทจะกระตุ้น Neuron บางอินพุทหยุดการทำงานของมัน เมื่อผลกระทบที่สะสมเกินกว่า Threshold Neuron จะ fire และส่งสัญญาณต่อไปใน neural networks แต่ละ neuron จะรับเซตของอินพุท แต่ละอินพุทจะถูกคูณกับ weight ที่เปรียบได้กับความแข็งแรงของ Snapses ผลรวมของอินพุทที่คูณกับ Weight จะเป็นตัวกำหนดระดับของการ fire เรียกว่า Activation level

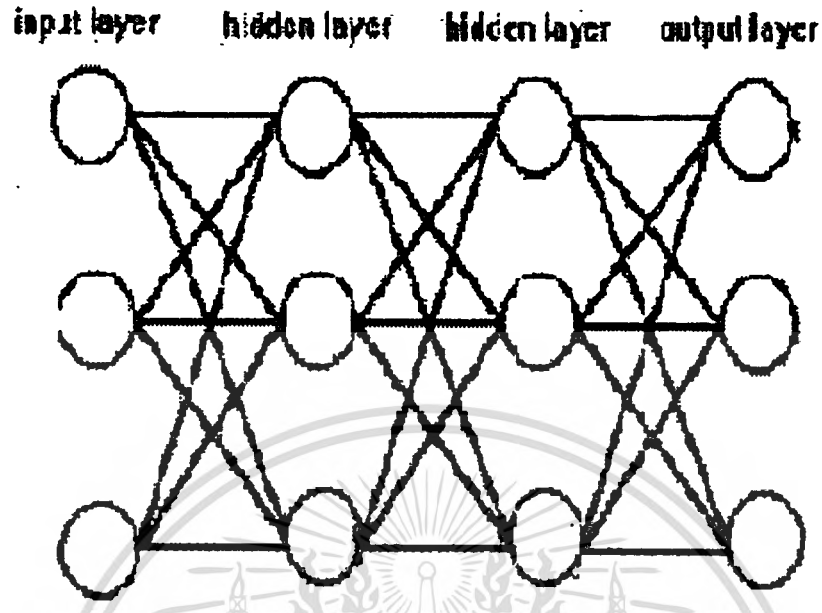
$\sum x_i w_i$

สัญญาณอินพุทจะถูกดำเนินการต่อโดย Activation Function เพื่อจะได้สัญญาณเอาท์พุท ซึ่งถ้าไม่เท่ากับ 0 จะถูกส่งต่อไป Activation Function อาจเป็น Threshold Function หรือ smooth function เช่น Sigmoid Function หรือ Hyperbolic Tangent Function

Neural Network แก้ปัญหาโดยการเรียนรู้ด้วยตนเอง (Self - Learning) และการจัดโครงสร้างตัวเองใหม่ (Self - organization) มันได้ความชาญฉลาดมาจาก การรวบรวมพฤติกรรมการคำนวณอย่างง่ายๆ ในแต่ละ Neuron

แนวคิดพื้นฐาน

Neural Network มีสถาปัตยกรรมกระจายแบบขนาน (Parallel - Distribution) กับ nodeและ connections จำนวนมาก แต่ละ Connection ชี้จาก node หนึ่งไปอีก node หนึ่งและมี Weight ติดอยู่กับแต่ละ Connection



การสร้าง Neural Network ทำดังต่อไปนี้

- พิจารณาคุณสมบัติของ Network : topology ของ Network , ชนิดของ Connection order ของ Connection และขอบเขตของ weight

- พิจารณาคุณสมบัติของ Node : ขอบเขตของ Activation และ Activation Function

- พิจารณา system dynamic : วิธีการ initialize weight , สูตรการคำนวณการ Activate และกฎการเรียนรู้ (learning rule)

คุณสมบัติของ Network

topology ของ network หมายถึง โครงสร้างและการเชื่อมต่อระหว่างกัน โครงสร้างอาจกำหนดโดยจำนวน layer และจำนวน node ในแต่ละ layer ประเภท layer มีดังนี้

Input layer

Hidden layer

Output layer

Input unit ไม่ประมวลข้อมูลแต่จะส่งผ่านข้อมูลไปยัง layer อื่น Hidden layer จะเป็น non-linearity ให้ระบบ ส่วน Output layer เป็นการแสดงผล

ในเรื่องของการเชื่อมต่อระหว่างกัน Network อาจเป็นแบบ Feed Forward หรือ Recurrent และอาจเชื่อมต่อแบบ Symmetrical หรือ Asymmetrical ดังนิยามต่อไปนี้

- FeedForward Network : แต่ละ connection ชี้ไปยังทิศทางเดียวกัน (จาก Input Layer ไปยัง Output Layer)

- Recurrent Network : มีการ Feedback หรือ Loop

- Symmetrical connection : การสมมาตรคือ ถ้ามี connection จาก node i ไป j จะมี connection จาก j ไป i เสมอและ weight บนสอง connections นั้นจะเท่ากัน ($w_{ij} = w_{ji}$)

- Asymmetrical connection : ไม่สมมาตร

connection ระหว่าง node ต่าง Layer เรียกว่า Interlayer connection ส่วนระหว่าง node ใน layer เดียวกันเรียกว่า intralayer connection ,ที่ชี้จาก node มายังตัวเองเรียกว่า Self - conection และ connection ระหว่าง node ในต่าง layer ที่อยู่ใกล้กัน (ไม่อยู่ติดกัน) เรียกว่า supralayer connection Connectivity หมายถึง ทุกๆ node ใน layer ใดๆ จะติดต่อกับทุกๆ node ใน layer ถัดไป

High - order connection หมายถึง connection ที่รวมอินพุทจากหลายๆ node โดยการคูณกัน จำนวนของอินพุทเป็นตัวบอก order ของ Connection นั้น order ของ Neural Network คือ order ที่สูงที่สุด โดยส่วนมาก Neural Network เป็น First order

Connection Weight อาจเป็นจำนวนจริงหรือจำนวนเต็ม และอาจอยู่ในช่วงใดช่วงหนึ่ง Weight ปรับได้ในขณะที่ Train Network เมื่อ Train แล้วทั้งหมดควรจะมีค่าคงที่

คุณสมบัติของ Node

Activation Level ของ node อาจเป็นแบบ discrete หรือแบบต่อเนื่องภายในช่วงเช่น [0,1] หรือไม่จำกัดขึ้นอยู่กับ Activation Function ที่เลือกใช้ ถ้าเป็น Hard - limiting function , Activation level จะเป็น 0 (หรือ -1.) หรือ 1 ถ้าเป็น sigmoid function , Activation จะอยู่ในช่วง [0,1]

Sigmoid Function : $F(x) = 1/(1 + e^{-x})$

ถ้าเป็น Activation Function เชิงเส้นจะไม่จำกัดค่า

System Dynamics

การกำหนดค่า weight จะขึ้นกับแบบของ Neural Network ที่เลือกแต่ส่วนมากจะ random

Learning rule จะบ่งบอกวิธีการปรับเวท เพื่อยกระดับการทำงานและการคำนวณเวทในแต่ละวงรอบของการ train

เมื่อใช้Neural Network แก้ปัญหา ผลลัพธ์จะอยู่ที่ Activation Level ของเอาท์พุทยูนิต เช่นเมื่อต้องการแบ่งแยกผลไม้ออกเป็น Lemon,orange และ apple Nueral Network จะมี 3 เอาท์พุทยูนิตแทน 3 ชนิด เมื่อเราให้ระบบแยกผลไม้ที่ไม่ทราบชนิด ข้อมูลจะผ่านเข้าทาง input layer และแพร่ไปข้างหน้า ถ้าเอาท์พุทยูนิตที่แทน apple มีค่าระดับ Activation มากที่สุดผลไม้นั้นคือ apple

ในการ train จะต้องใช้การคำนวณบางอย่างเพราะเราต้องรู้ค่าระดับของ Activation ที่เกิดขึ้นจริงและค่าที่ต้องการเพื่อหา error นำมาใช้ในการปรับเวท

ระดับของ Activation ของอินพุทยูนิตไม่ต้องคำนวณแต่จะได้จากข้อมูลส่วนของอิตเดนและเอาท์พุทยู

นิต จะคำนวณจาก Activationฟังก์ชั่น ที่ใช้เช่น Sigmoidฟังก์ชั่น, ระดับ Activation (o_j) ของยูนิต j จะคำนวณจาก

$$o_j = 1/[\ 1 + e^{-(\Sigma_i w_{ji} x_i - \theta_j)}]$$

เมื่อ X_i คืออินพุทจากยูนิต i W_{ji} คือ weight บน connection จากยูนิต i มายังยูนิต j และ θ_j คือ Threshold ของยูนิต j

ในกรณีของ Hard - Limiting ฟังก์ชั่น เอาท์พุทจะได้จาก

$$O_j = 1, \Sigma_i w_{ji} x_i > \theta_j$$

$$0, \Sigma_i w_{ji} x_i \text{ น้อยกว่าหรือเท่ากับ } \theta_j$$

ในบางโอกาสเราเก็บค่าระดับของ Activation ของบาง node เอาไว้

Inference and Learning

ในการสร้างระบบ AI ที่ใช้นิวรอนเน็ตเวิร์ก จะทำตามขั้นตอนต่อไปนี้

1. เลือกแบบของนิวรอนเน็ตเวิร์กที่เหมาะสมกับปัญหา
2. สร้างนิวรอนเน็ตเวิร์ก
3. ทำการ Train นิวรอนเน็ตเวิร์ก ตามวิธีการเรียนรู้ของรูปแบบของเน็ตเวิร์ก ที่เลือก
4. ใช้นิวรอนเน็ตเวิร์กที่ Train แล้วหาข้อวินิจฉัย (Inference) หรือแก้ปัญหา ถ้าผลที่ได้ไม่พอใจ ให้กลับไปทำขั้นตอนใดขั้นตอนหนึ่งใหม่

การคุ้นเคยกับแอพพลิเคชั่นที่มีอยู่จะช่วยให้พิจารณา network ที่เหมาะสมตามหลักการออกแบบได้ และจะเลือกรูปแบบการคำนวณที่เหมาะสมที่สุดสำหรับการ learn และการวินิจฉัย

ใน training phase (หรือ learning) ชุดของ Training instance ที่ถูกกำหนดแต่ละ training instance ที่ถูกกำหนด แต่ละ training instance ถูกบรรยายโดยลักษณะแบบเวคเตอร์ (ถูกเรียกว่า อินพุทเวคเตอร์) มันอาจจะสัมพันธ์กับสิ่งที่ต้องการต่อๆ ไป(a concept, a class และอื่นๆ) ซึ่งถูกเข้ารหัสเป็นเวคเตอร์อันอื่นๆ (เรียกว่า desired output vector) ในตัวอย่างข้างบน สาม class ที่เป็นไปได้ (Lemon, orange และ apple) สามารถที่จะถูกเสนอในลักษณะ [100],[010] และ[001] ได้โดยเริ่มต้นด้วยการกำหนดค่าเวทแบบเจาะจงหรือ แบบสุ่มค่า (random) นิวรอนเน็ตเวิร์กจะถูก train เพื่อที่จะปรับตัวเอง เพื่อให้ได้ลักษณะการวินิจฉัย(training instance) โดยการเปลี่ยนค่าเวทในแต่ละรอบการ train เราจะเสนอการวินิจฉัยให้กับเน็ตเวิร์ก ซึ่งจะทำให้เกิดเวคเตอร์เอาท์พุท ซึ่งจะนำมาเปรียบเทียบกับเวคเตอร์เอาท์พุทที่ต้องการ(desired output vector) ว่าตรงหรือใกล้เคียงกันไหม ด้วยวิธีการนี้ error สำหรับแต่ละเอาท์พุทยูนิตจะถูกคำนวณและใช้ในการ update ค่า relevant weight ใน multilayer network error จาก hidden unit จะไม่ถูกสังเกตเห็นได้โดยตรง แต่จะถูกประเมินได้ด้วย heuristic บางอย่าง แต่ละการเปลี่ยนแปลงค่าเวทจะถูกหวังว่าจะทำให้ error ลดลง เมื่อทุกข้อวินิจฉัย(instance) ถูกทดสอบ network จะเริ่มทำกับ ข้อวินิจฉัยแรกและทำซ้ำคล้ายๆ กันกับอันอื่นๆ จะทำซ้ำไปซ้ำมาอย่างต่อเนื่องจนกว่าประสิทธิภาพของระบบ (ในเทอมของ error magnitude) จะอยู่ในระดับที่น่าพอใจ ในการทดลองเราเลือก

error criterion ซึ่งจะถูกทำให้ลดลงระหว่างการ training สอง criteriaที่ถูกใช้สำหรับการดู error ได้แก่

อย่างแรกคือ sum of squared error

$$\sum_p \sum_j (T_{jp} - O_{jp})^2$$

โดย T_{jp} และ O_{jp} เป็นเอาท์พุทที่ต้องการ และค่าเอาท์พุทจริงๆที่ได้จาก unit j ในการวินิจฉัยแบบแบบ p นอกจาก Sum of squared error แล้วยังมี criteria อื่นๆที่เช่น cross - entropy ตามข้างล่าง

$$-\sum_p \sum_j P_{jp} \log_2(O_{jp}) + (1 - P_{jp}) \log_2(1 - O_{jp})$$

โดย P_{jp} และ O_{jp} เป็นเอาท์พุทที่ต้องการของยูนิต j ในการใช้ instance P และค่า probability ของเอาท์พุทที่แท้จริงของยูนิต j ด้วยการใช้ใน instance P

ในบางรูปแบบของ network error ไม่สามารถสังเกตได้เพราะ training instance นั้นยังไม่ถูกค้นพบด้วยผลลัพธ์ที่ถูกต้องการภายใต้โอกาสนี้วิธีการ learning ต่างๆ ซึ่งจะทำการ update เวทโดยการใช้ heuristic ที่ต่างๆ กันไปควรที่จะถูกนำมาใช้

ใน inference phase, เน็ตเวิร์กจะแพร่ข้อมูลจากอินพุทไปยัง output layer เมื่อการแพร่เสร็จลงเอาท์พุทยูนิตจะเก็บผลลัพธ์ของ inference ไว้ อย่างไรก็ตามข้อแตกต่างก็ควรที่จะถูกทำระหว่าง feed forward และ recurrent network การที่ไม่มี feedback ในกรณีก่อนทำให้สามารถที่จะกำหนด clear endpoint ของ inference ได้ และยังสามารถยืนยันได้ว่า network นั้น stable

Recurrent network นั้นไม่เหมือนกัน โดยจะซับซ้อนกว่าเพราะว่ามี feedback ที่ป้อนจากเอาท์พุทกลับมาเป็นอินพุท หลังจากที่ได้ทำงานโดยอินพุทหนึ่ง เอาท์พุทจะคำนวณและจะถูก feedback กลับไปรวมกับอินพุทเพื่อเป็นอินพุทแก่ระบบ เอาท์พุทจะถูกคำนวณใหม่และจะ process กับตัวมันเองซ้ำไปซ้ำมาที่ประสบผลสำเร็จจะผลิตเอาท์พุทที่มีการเปลี่ยนแปลงน้อยลงและเกือบที่จะเป็นเอาท์พุทที่คงที่ ซึ่งจุดนั้นถูกกำหนดเป็นจุด endpoint (จุดสิ้นสุด) ของ inference network ที่ process ไม่จบจะถูกเรียกว่า Unstable(ไม่เสถียร)

ทั้งที่มีการเปลี่ยนแปลงโดยเฉพาะ model ที่ต่างกัน , general learning และ inference procedure สามารถคิดได้ดังนี้

อัลกอลิทึมที่ใช้ในการ learn ของ Neural โดยทั่วไป

ตัวอย่างของการ train แบบ n

1. กำหนดค่า network weight ตั้งต้นกำหนดให้ i = 1
2. เสนอตัวอย่างอันที่ i ไปให้แก่ network
3. สังเกตระดับของเอาท์พุทยูนิต ที่ทำงานโดยการใช้อัลกอลิทึม inference (ซึ่งจะบรรยายต่อไป) ถ้าประสิทธิภาพของเน็ตเวิร์กอยู่ในขั้น predefined standard (หรือในขั้นที่หยุดการวิเคราะห์แล้ว) ก็ exit ได้
4. ทำการ update ค่า weight โดยกฎการ learn ของเน็ตเวิร์ก

5. ถ้า i = n ดังนั้นให้ i = 1 ใหม่ ถ้ายังไม่เท่ากับ n ก็ให้เพิ่มค่า i อีกหนึ่งแล้วไปทำ ในขั้นตอนที่ 2 ใหม่

อัลกอลิทึม Inference ของ Neural Network (โดยทั่วไป)

1. เสนอตัวอย่างแก่ network ที่ input layer
2. คำนวณระดับ activation ของ node ที่อยู่ในเน็ตเวิร์ก
3. สำหรับ feedforward network ถ้าระดับ Activation ของทุกเอาท์พุทยูนิตถูกคำนวณแล้ว exit ได้ แต่สำหรับ recurrent network ถ้าระดับ Activation ของทุกเอาท์พุทยูนิตค่อนข้างจะคงที่ก็ exit ได้ ถ้ายังไม่คงที่ต้องไปทำในขั้นตอนที่ 2 ใหม่ อย่างไรก็ตามถ้าเน็ตเวิร์กถูกพบว่าไม่มีความเสถียรก็จะ exit และถือว่า fail

การจัดรูปแบบข้อมูล(Data Representation)

ค่าที่มีลักษณะ discrete เช่น ลักษณะของสีเช่น สีแดง, สีเหลือง, สีน้ำเงินและสีอื่นๆ สามารถที่จะถูกเข้ารหัสโดย input unit เดียวกันได้ ในกรณีนี้ค่า activation ของ 1 จะหมายถึง 'yes' หรือ 'true' ขณะที่ค่า activation ของ 0 จะหมายถึง 'no' หรือ 'false' ค่า activation อาจจะมีค่าอยู่ระหว่าง 0 และ 1 ก็ได้ซึ่งจะเป็นการระบุค่าความเป็นไปได้ของค่านั้น

ค่าที่มีลักษณะต่อเนื่อง (contineous) เช่น ลักษณะของอายุที่มีช่วงตั้งแต่ 1 ถึง 150 ถูกเข้ารหัสโดยอินพุทยูนิต และค่าของลักษณะต่างๆ นี้จะถูกจัด(map) อยู่ใน unit activation ในการทดลองช่วงค่าที่ต่อเนื่องนี้จะถูก normalize ให้อยู่ในช่วงระหว่างค่า 0 ถึง 1

ในการเสนอแบบ discrete ช่วงค่าที่ต่อเนื่องจะถูกแบ่งออกเป็นหลายช่วง โดยแต่ละช่วงจะถูกเข้ารหัสโดยอินพุทยูนิตพิจารณาช่วงระหว่าง a และ b (a และ b เป็นจำนวนจริง, b>a) เราจะแบ่งค่าระหว่างมันออกเป็น k ช่วงโดยมี k - 1 จุดอยู่ระหว่าง a และ b : $m_1, \ldots, m_{k-1}$ เราอาจจะกำหนด $m_0 = a$ และ $m_k = b$ ในการเข้าถึงกำหนดให้ค่า feature x ถ้า $m_i \leq x < m_{i+1}$ ดังนั้นยูนิตจะเข้ารหัสที่ช่วงที่มี activation ของ 1 และส่วนที่เหลือของยูนิตจะมี zero activation ในการเข้าถึงแบบอื่นๆ ยูนิตจะมีค่า activation ของ 1 ถ้าขอบที่ต่ำกว่าของช่วงที่สัมพันธ์นั้นน้อยกว่าหรือเท่ากับค่า x นั่นคือ $m_i \leq x$ มันจึงชัดแจ้งว่าเป็นการเข้าถึงแบบที่สอง จะมีหนึ่งหรือมากว่าหนึ่งยูนิตที่จะถูก activate ที่เวลาเดียวกัน สำหรับเข้ารหัสสำหรับค่าเพียงค่าเดียว เพราะการเข้ารหัสของข้อมูลมีการกระจายมากขึ้น การเข้าถึงแบบที่สองจึงดูว่ามี noise น้อยกว่าการเข้าถึงแบบแรก

ในวิธีที่ 3 ยูนิตจะถูก activate เพื่อถูกพิจารณาเพิ่มขึ้นโดยคิดจากช่วงใดมีความใกล้กับค่านั้น ใกล้กันเพียงใด นี่ก็เป็นเทคนิคการเข้ารหัสแบบแพร่กระจาย(distribution)

ในวิธีที่ 4 ลักษณะค่าที่ต่อเนื่องจะถูกเข้ารหัสโดย fuzzy set สำหรับ fuzzy inference

มันเป็นไปได้ที่จะเสนอค่าต่อเนื่องเป็นเช่นค่า binary (ดีกว่าเป็นค่า decimal) ตัวอย่างเช่น 8 ถูกเสนอโดย 4 บิทเป็น1000 , 7 ถูกแสดงเป็น 0111 วิธีนี้ไม่น่าเป็นจริงในลักษณะกายภาพ เพราะค่าที่ใกล้เคียงกันสอง

ค่ามีการเสนอได้แตกต่างกันมาก การกำหนดความสัมพันธ์ในการที่จะแสดงข้อมูลที่ถูกเก็บไว้นั้นเป็นการทำ preprocessing ขั้นตอนนี้เกี่ยวข้องกับการเลือกและการแปลงข้อมูล

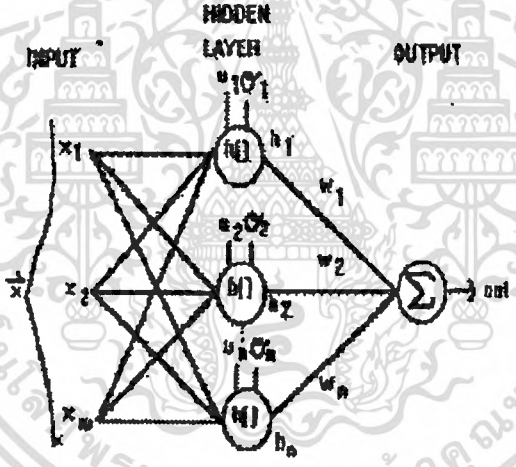
มันไม่ควรถูกมองข้าม ในความแม่นยำในการคำนวณ อาจจะมีผลต่อคุณสมบัติ convergence ของนิวรอนเน็ตเวิร์ก ความถูกต้องของ network weight หรือ weight - update calculation ความแม่นยำที่ต่ำกว่าเป็นไปได้ที่จะต้องใช้เทคนิคพิเศษอื่นมาใช้อีก

2.2.2 ทฤษฎีของ Radial Basis - Function Network

Radial Basis - Function Network เป็นที่สนใจอย่างมากจากการที่มันสามารถ train ได้อย่างรวดเร็ว, generality (ความสามารถในการประมวลผลอินพุทได้ดี ทั้งอินพุทที่เคยเห็นมาแล้ว และยังไม่เคยเห็น) และความไม่ซับซ้อนของมัน Radial Basis - Function Network มักจะถูกเปรียบเทียบกับ Backpropagation feed-forward network ซึ่งเป็นแบบที่ใช้กันอย่างกว้างขวาง backpropagation มีปัญหาเกี่ยวกับการ training มากมาย เช่น paralysis , local minima ฯลฯ Radial Basis - Function Network สามารถ train ได้อย่างรวดเร็วและไม่มีปัญหาเหล่านั้น แต่มีข้อเสียอยู่คือ หลังจาก training แล้วมันจะทำงานได้ช้ามากเนื่องจากต้องใช้การคำนวณมาก

Radial Basis - Function Network ได้รับการพิสูจน์แล้วว่ามันเป็น universal approximator นั่นคือเมื่อให้ Neuron ใน hidden layer มากพอ มันสามารถประมาณค่าฟังก์ชั่นใดๆ ก็ได้ โดยมีความแม่นยำตามความพอใจ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



จากรูปจะเห็นว่า RBF Network จะคล้ายกับ Feed - Forward Network โดยทั่วๆ ไปเช่น Backpropagation แต่ว่าจะแตกต่างกันในการทำงาน แทนที่จะเริ่มต้นด้วยการสุ่มค่าของเซตของเวทของ hidden unit จะถูกออกแบบเพื่อให้มีผลตอบสนองครอบคลุมอาณาบริเวณที่ต้องการของ Input vector space โดยส่วนใหญ่แล้วฟังก์ชั่นที่ใช้จะใช้ Radial basis function

แต่ละ Hidden unit จะให้ค่าตามเอ็กโพเนนเชียลฟังก์ชั่น

$h_i = \exp[-D_i^2 /(2\sigma^2)]$

x = อินพุทเวกเตอร์

u_i = เวทเวกเตอร์ของ hidden unit I โดยจำนวนของ hidden unit จะเท่ากับจำนวนของ training vector

$D_i^2 = (x - u_i)^T (X - u_i)$

โดย x และ u เป็นคอลัมม์เวคเตอร์

u_i จะมีค่าเท่ากับ $y = \sum_i h_i w_i$

โดยที่ w_i = เวทจาก hidden unit ไปยัง output layer

ในการ training จะมี 2 stage

ในครั้งแรกจะตั้งค่าให้ u_i และ σ_i แล้วจึง train เวทเมตริก w โดย supervised training Generalized Regression Neural Network (GRNN)

GRNN เป็นระบบที่เพิ่งได้รับการพัฒนาขึ้นมาเมื่อไม่นานมานี้ (Specht 1991) คุณสมบัติพิเศษของ GRNN คือ มันไม่ต้องการการ train มันสามารถประมาณค่าฟังก์ชั่นใดๆ ระหว่างอินพุทเวคเตอร์และเอาท์พุทเวคเตอร์ได้โดยการประมาณค่าโดยตรงจากข้อมูลที่ใช้ในการ train ยิ่งกว่านั้นมันมีความอยู่ตัว (consistent) คือเมื่อเซตของข้อมูลที่ในการ train ใหญ่ขึ้นมากๆ ค่าความผิดพลาดในการประมาณจะเข้าใกล้ 0 โดยมีข้อจำกัดน้อยมากกับฟังก์ชั่น

สถาปัตยกรรมของ GRNN

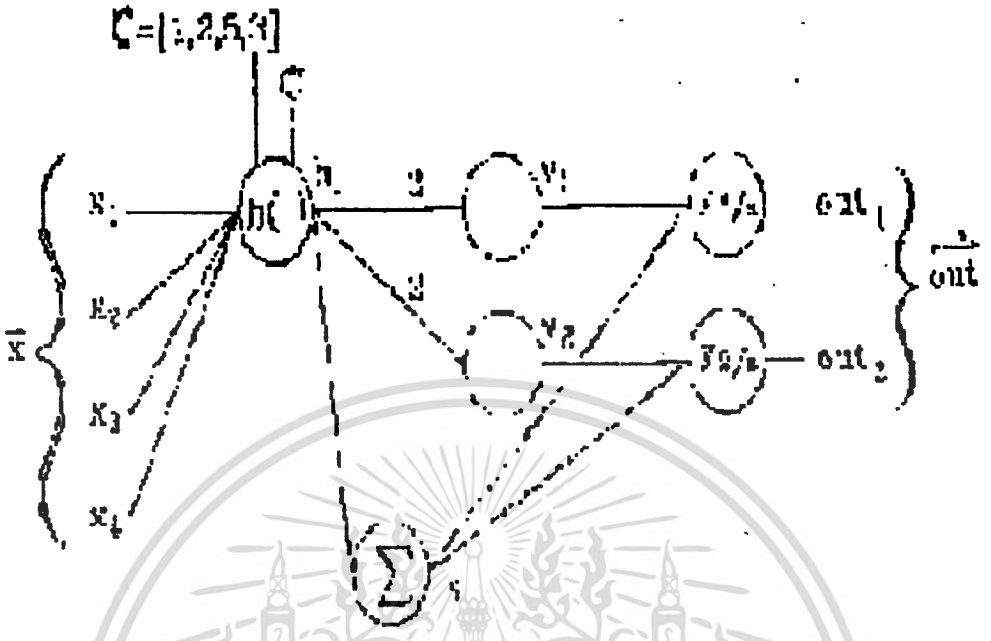
GRNN จะประกอบด้วย 1 อินพุทเลเยอร์ 1 ฮิดเดนเลเยอร์ และ 1 เอาท์เลเยอร์หรือมากกว่า ทุกอินพุทนิวรอนจะต่อเข้ากับแต่ละฮิดเดนนิวรอน โดยเวทที่เชื่อมต่อระหว่างอินพุทเลเยอร์และฮิดเดนเลเยอร์ จะมีค่าเท่ากับ 1 แต่ละฮิดเดนนิวรอนจะเก็บค่าข้อมูลที่ใช้ในการ train ไว้ในรูปเวคเตอร์ 1 x Nเมื่อ N เป็นจำนวนเต็มบวก เพราะฉะนั้นจำนวนฮิดเดนนิวรอนจะขึ้นอยู่กับจำนวนข้อมูลที่ใช้ในการ train จากทุกฮิดเดนนิวรอนจะมีการเชื่อมต่อไปยังแต่ละเอาท์พุทยูนิตในเอาท์พุทเลเยอร์โดยเวทที่แต่ละการเชื่อมต่อจะมีค่าเท่ากับเอาท์พุทที่ต้องการที่เอาท์ยูนิตนั้นดังจะยกตัวอย่างดังนี้

สมมติว่าเซตที่ใช้ในการ train เป็นดังนี้

อินพุทเวคเตอร์ : 1 2 5 3

เอาท์พุทเวคเตอร์ที่ต้องการ : 3 2

จะได้ GRNN ดังรูปต่อไปนี้



การใช้ GRNN เป็น Classifier

เนื่องจาก GRNN สามารถประมาณค่าฟังก์ชั่นที่ต่อเนื่องใดๆ ได้ ทำให้มันสามารถเปลี่ยนเป็น Classifier ได้อย่างง่ายดาย โดยการให้แต่ละเอาท์พุทยูนิตแทนแต่ละ class

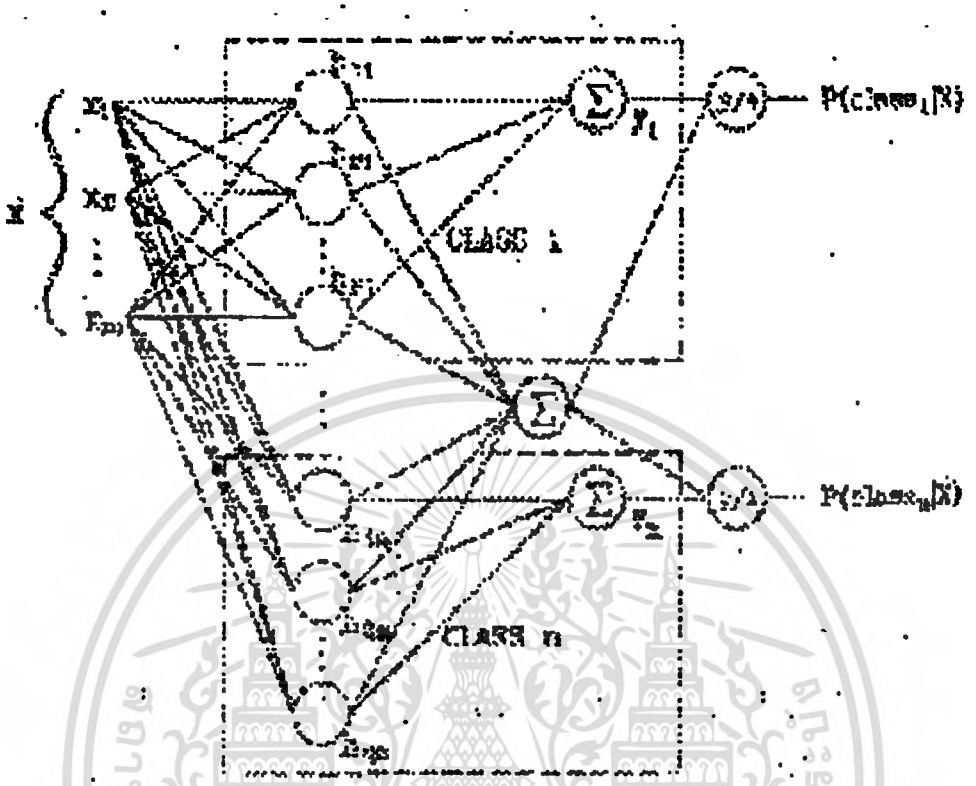
แต่ละ output layer weight จะมีค่า 1 หรือ 0 แต่ละ hidden unit จะถูกแทนด้วย input training vector

ในการตั้งค่าเวทของ GRNN ค่าของอินพุทเวทจะเป็น 1 หมด

สำหรับแต่ละ hidden unit

1. ให้ค่าเวทเป็น 1 สำหรับเวทที่ต่อไปยังเอาท์พุทยูนิตของ class เดียวกัน

2. เวทที่ต่อไปยังเอาท์พุทยูนิตอื่นๆ ให้เป็นค่า 0 และสามารถเอาออกจาก network ได้เมื่อทำตามวิธีนี้แล้วจะได้รูปดังต่อไปนี้



ค่าของแต่ละฮิดเดนยูนิตจะถูกรวมกันแล้วนำไปหารกับแต่ละเอาท์พุทยูนิต เป็นการ normalization

2.2.3การเลือกลักษณะเด่น

การเลือกลักษณะเด่นถือเป็นส่วนสำคัญที่สุดในกระบวนการรู้จำ เนื่องจากลักษณะเด่นจะถูกใช้เป็นข้อมูลแทนรูปแบบ (pattern) ที่กำลังพิจารณาอยู่ และถูกนำไปใช้เปรียบเทียบรูปแบบที่ต้องตรวจสอบกับรูปแบบที่มีอยู่ ดังนั้น การเลือกลักษณะเด่นจึงจำเป็นจะต้องทำอย่างรอบคอบเพื่อให้ได้ลักษณะเด่นที่ค่อนข้างคงที่ มีการเปลี่ยนแปลงน้อย และสามารถใช้แทนรูปแบบได้ ลักษณะเด่นที่ใช้จะขึ้นกับลักษณะของปัญหา เปอร์เซนต์ความถูกต้องที่ต้องการ และอาจจะขึ้นกับงบประมาณในการทำงานด้วย ในงานวิจัยทางด้านกระบวนการรู้จำ มีการเลือกใช้ลักษณะเด่นต่างๆมากมาย ซึ่งแต่ละอย่างก็มีข้อดีข้อเสียแตกต่างกัน แต่ส่วนใหญ่แล้ว จะใช้ลักษณะเด่นหลายๆอย่างมาประกอบกัน จะไม่เลือกใช้ลักษณะเด่นอย่างใดอย่างหนึ่งแต่เพียงอย่างเดียว

ลักษณะเด่นที่นิยมใช้กันทั่วไป

จากการศึกษาวิจัยเกี่ยวกับลายมือเขียนและลายเซนต์พบว่า ลักษณะเด่นโดยทั่วไปของลายมือเขียนและลายเซนต์สามารถแบ่งได้เป็นประเภทใหญ่ๆ ได้ดังนี้

1. ตำแหน่ง หมายถึง ค่าพิกัด (X_k,Y_k) ณ จุด k ใดๆ ซึ่งได้จากกลุ่มสัญญาณ $U_x(t)$ และ $U_y(t)$ ในช่วงเวลาใดๆ โดยในการเก็บข้อมูลเกี่ยวกับตำแหน่งนี้ อาจจะเก็บข้อมูลเกี่ยวกับตำแหน่งที่ยกปากกาไว้ด้วย เนื่องจากข้อมูลในส่วนนี้สามารถปลอมแปลงหรือเลียนแบบได้ยาก ถึงแม้จะมีตัวอย่างลายเซนต์ที่ถูกต้องอยู่ก็ตาม นอกจากนี้ ข้อมูลเวลาของตำแหน่งในการเซนต์ เช่น ระยะเวลาในการเซนต์ลายเซนต์ เวลาในขณะที่เซนต์ลายเซนต์ ณ แต่ละพิกัด ก็มีความสำคัญมากเช่นกัน ข้อมูลส่วนนี้ทำให้สามารถแยกข้อมูลตำแหน่งของลายเซนต์ออกเป็นส่วนจริง (real part) และจินตภาพ (imaginary part) ของฟังก์ชั่นเชิงซ้อนได้ในรูปของ

$$F(t) = U_x(t) + iU_y(t)$$

ฟังก์ชั่นที่ได้นี้สามารถนำไปทำการ Normalization เป็นตำแหน่งในระนาบที่เป็นอิสระจากกัน นำไปทำการหมุน , ปรับขนาด และดูค่าแนวโน้มของการเคลื่อนที่จากซ้ายไปขวาได้

2. แรงกดปากกา เป็นลักษณะเด่นอีกอย่างหนึ่งซึ่งนิยมนำไปใช้ร่วมกับลักษณะเด่นอื่นๆ เนื่องจากช่วยทำให้การวิเคราะห์ลายเซนต์มีประสิทธิภาพมากยิ่งขึ้น แต่ก็มีปัจจัยหลายอย่างที่มีผลต่อแรงกดปากกา เช่น ความเอียงของปากกาและแรงกดนิ้วบนปากกาหรือกระดาษอิเล็กทรอนิกส์ เป็นต้น

3. แรง เป็นลักษณะเด่นซึ่งเกิดจากสิ่งเร้าซึ่งส่งจากสมองของผู้เซนต์ไปยังระบบกล้ามเนื้อ และมีขนาดแตกต่างกันไปตามขนาดของตัวอักษรหรือลายเซนต์ที่เขียนด้วย ลักษณะเด่นนี้จะไม่วัดด้วยเครื่องมือวัด แต่จะคำนวณจากค่าพิกัดและแรงกดปากกา โดยใช้โมเดลทางคณิตศาสตร์

4. ความเร็ว ข้อมูลนี้เป็นปัจจัยสำคัญในกระบวนการพิสูจน์ลายเซนต์แบบออนไลน์ เนื่องจากความเร็วในการเซนต์ลายเซนต์หรือเขียนข้อความของบุคคลแต่ละคนย่อมจะมีความแตกต่างกัน โดยค่าความเร็วนี้อาจจะคำนวณจากค่าพิกัด หรือใช้อุปกรณ์พิเศษวัดก็ได้

5. ความเร่ง ในทฤษฎีเกี่ยวกับลายมือเขียนถือว่าความเร่งเป็นฟังก์ชั่นที่มีความสำคัญมาก ซึ่งอาจจะได้จากการคำนวณ การใช้ปากกาที่สามารถวัดความเร่งได้ หรือการใช้อุปกรณ์ฮาร์ดแวร์สำหรับวัดความเร่งก็ได้

2.2.4 การตัดสินใจ

ความแตกต่างของลายเซนต์

โดยทั่วไป ลายเซนต์แต่ละลายเซนต์จะมีความแตกต่างกัน ถึงแม้จะเป็นลายเซนต์ของบุคคลคนเดียวกันก็ตาม ทำให้สามารถแบ่งความแตกต่างของลายเซนต์ได้เป็น 2 ประเภท คือ

1. ความแตกต่างภายนกลุ่ม (intraclass or intrapersonal variability) คือ ความแตกต่างระหว่างภายในกลุ่ม (ϖ^{i}_{j}) ของลายเซนต์ของบุคคลคนเดียวกันคือ คนที่ i

2. ความแตกต่างระหว่าง (interclass or interpersonal variability) คือ ความแตกต่างระหว่างกลุ่มของลายเซนต์สองกลุ่ม (ϖ^{i}_{j}) และ (ϖ^{i}_{j}) ซึ่งได้จากบุคคลคนที่ i และ

ในทางทฤษฎีความแตกต่างระหว่างกลุ่มก็ควรจะมีมากพอที่จะสามารถแบ่งแยกลายเซนต์ของแต่ละบุคคลได้ แต่ในทางปฏิบัตินั้นมักพบว่าไม่สามารถแบ่งแยกความแตกต่างทั้งสองกลุ่มนี้ได้อย่างชัดเจน

ในกระบวนการรู้จำมีโอกาสจะยอมรับและปฏิเสธลายเซนต์ในกรณีต่อไปนี้

1. การยอมรับลายเซนต์ที่ถูกต้อง คือ การยอมรับว่าลายเซนต์ที่ต้องการตรวจสอบ ซึ่งเป็นลายเซนต์ของบุคคลคนเดียวกันกับลายเซนต์ในฐานข้อมูล เหมือนกับลายเซนต์ในฐานข้อมูล

2. การยอมรับลายเซนต์ปลอม คือ การยอมรับลายเซนต์ที่ต้องการตรวจสอบ ซึ่งเกิดจากบุคคลคนละคนกันว่ามีความคล้ายกับลายเซนต์ในฐานข้อมูล

3. การปฏิเสธลายเซนต์จริง คือการยอมรับว่าลายเซนต์ที่ตรวจสอบมีความแตกต่างกับลายเซนต์ในฐานข้อมูล ทั้งๆ ที่ได้จากบุคคลคนเดียวกัน

4. การปฏิเสธลายเซนต์ปลอม คือ การปฏิเสธว่าลายเซนต์ที่ตรวจสอบซึ่งเกิดจากบุคคลคนละคนกันเป็นลายเซนต์เดียวกันกับในฐานข้อมูล

อัตราความผิดพลาดทั้งประเภท คือ การปฏิเสธลายเซนต์จริง (type 1 error หรือ FRR false rejection rate) และการยอมรับลายเซนต์ปลอม (type 2 error หรือ FAR false acception rate) จะเป็นเครื่องกำหนดประสิทธิภาพของเครื่องวิเคราะห์ลายเซนต์ ซึ่งอัตราความผิดพลาดทั้งสองจะขึ้นกับค่าขอบเขตในการยอมรับหรือปฏิเสธลายเซนต์

บทที่ 3
การคำนวณและสร้าง

3.1 ลักษณะเด่นของลายเซนต์ที่ใช้

เราเลือกใช้ลักษณะเด่นของลายเซนต์ 2 แบบคือ

1.ระยะเวลาและจำนวนครั้งของการยกวางปากกา

2.ความเร่งตามแนวแกน x และแกน y

โครงสร้างของโครงข่ายประสามเทียมแบบจีอาร์เอ็นเอ็น

ในโครงงานนี้เราได้นำโครงข่ายประสามเทียมแบบจีอาร์เอ็นเอ็นมาประยุกต์ โดยโครงสร้างของโครงข่ายเป็นดังนี้

อินพุทยูนิท 746 ยูนิทเท่ากับจำนวนข้อมูลลักษณะเด่นของลายเซนต์

จำนวนฮิดเดนยูนิท 5 ยูนิท โดยที่ค่า u ของแต่ละยูนิทจะเป็นลักษณะเด่นของลายเซนต์ตัวอย่างแต่ละอัน ส่วนค่า σ หาจากค่าระยะทาง RMS จากยูนิทนั้นไปยังยูนิทที่ใกล้ที่สุด

อินพุทยูนิท 746 ยูนิทเท่ากับจำนวนข้อมูลลักษณะเด่นของลายเซนต์

เอาท์พุทยูนิท 1 ยูนิท

การทำงานและการคำนวณ

การทำงานของโปรแกรมจะแบ่งเป็น 2 เฟสคือ

1.เฟสการรับข้อมูลตัวอย่าง,ทำนอมัลไลซ์และจัดเก็บ

2.เฟสการประมวลผลข้อมูลลายเซนต์อินพุท

แต่ละเฟสมีการทำงานและการคำนวณดังนี้

1.เฟสการรับข้อมูลตัวอย่าง,ทำนอมัลไลซ์และจัดเก็บ

โปรแกรมจะทำการรับข้อมูลจากดิจิไทเซอร์และนำมาทำการนอมัลไลเซชัน (บทที่ 2)และจัดเก็บลงในฐานข้อมูล

2.เฟสการประมวลผลข้อมูลลายเซนต์อินพุท

เป็นการตัดสินใจว่าจะรับลายเซนต์อินพุทหรือไม่มีขั้นตอนดังนี้

2.1 เลือกข้อมูลลักษณะเด่นของลายเซนต์จากฐานข้อมูลตามชื่อของผู้ใช้

2.2 รับลายเซนต์อินพุทจากดิจิไทเซอร์

2.3 ทำการนอมัลไลซ์ลายเซนต์อินพุท

2.4 สร้างโครงข่ายประสามเทียมแบบจีอาร์เอ็นเอ็นจากข้อมูลในฐานข้อมูลตามขั้นตอนต่อ

ไปนี้

2.4.1 ใส่ค่าลักษณะเด่นของลายเซนต์จากฐานข้อมูลแต่ละชุดลงในค่า μ ของแต่ละฮิดเดนยูนิท ยูนิทละ 1 ชุด

2.4.2 คำนวณค่า σ จากค่าระยะทาง RMS จากแต่ละยูนิทไปยังยูนิทที่ใกล้ที่สุด โดยค่าระยะทาง RMS หาได้จาก

- หาระยะทางจากแต่ละยูนิทไปยังยูนิทอื่นๆทุกยูนิท ระยะทางหาได้จาก d
- หาระยะทางที่ใกล้ที่สุด
- หาค่ารากที่ 2 ของระยะทางนั้น

2.5 หาค่า threshold (ค่าเอาท์พุทต่ำสุดที่จะยอมรับลายเซนต์) โดยการนำค่าลักษณะเด่นของลายเซนต์จากฐานข้อมูลแต่ละชุดไปเป็นอินพุทของโครงข่ายประสามเทียมแล้วหาค่าเอาท์พุทที่น้อยที่สุดมาเป็นค่า threshold

2.6 นำค่าลักษณะเด่นของลายเซนต์อินพุทมาเป็นอินพุทของโครงข่ายประสามเทียมแล้วหาค่าเอาท์พุท

2.7 ตัดสินใจโดยถ้าค่าเอาท์พุทของลายเซนต์อินพุทมากกว่าค่า threshold จะยอมรับลายเซนต์นั้น

3.2 ลักษณะของฐานข้อมูลที่ใช้ในการพิสูจน์ลายเซนต์

เราใช้ฐานข้อมูลในงานพิสูจน์ลายเซนต์ดังนี้

- เก็บข้อมูลตัวอย่างลายเซนต์ 5 ครั้งโดยในแต่ละตัวอย่างเก็บข้อมูลเกี่ยวกับลายเซนต์ในครั้งนั้นประมาณครั้งละ 250 เรคคอร์ด

- ใช้ดึงข้อมูลตัวอย่างลายเซนต์มาใช้ในการพิสูจน์ลายเซนต์

ลักษณะของฐานข้อมูล

ฐานข้อมูลที่ใช้ประกอบไปด้วยตาราง 4 ตาราง

ตารางแรก ใช้ในการเก็บข้อมูลชื่อของผู้ใช้เพื่อใช้การให้ผู้เลือกลายชื่อเพื่อใช้จะได้นำข้อมูลตามชื่อนั้นมาพิสูจน์ข้อมูล และใช้ในการตรวจสอบดูว่ามีข้อมูลของคนๆ นั้นอยู่หรือไม่ ซึ่งมีลักษณะตารางดังนี้

Name	ID
ปัญญา ว่องไว	1

ตารางแรกนี้ใช้มีฟิลด์ name ใช้เก็บชื่อและนามสกุลของผู้ใช้และต้องใส่ข้อมูลชื่อและนามสกุลพร้อมกันเสมอโดยเก็บชื่อและมีช่องว่างหนึ่งช่องกั้นระหว่างชื่อและนามสกุล โดยเก็บข้อมูลเป็น string

ฟิลด์ ID ใช้ระบุถึงผู้ใช้คนใดคนหนึ่งอย่างย่อๆ เพื่อจะได้ไม่ต้องใช้ชื่อและนามสกุลระบุเพราะชื่อและนามสกุลนั้นเป็นข้อมูลที่มากไป และฟิลด์ ID นี้ใช้เป็นฟิลด์ที่เชื่อมระหว่างตารางทั้งสองที่เราใช้เก็บข้อมูลเป็น long integer

ตารางที่สองมีฟิลด์ในตาราง 6 ฟิลด์ คือ ID,node,cnt และ rad

ฟิลด์ ID ใช้เก็บข้อมูล เลขที่ระบุผู้ใช้ โดยเหมือนกับฟิลด์ ID ในตารางแรกมีไว้เพื่อใช้สร้างความสัมพันธ์ให้ทั้งสองตาราง เก็บข้อมูลเป็น long integer

ฟิลด์ node ใช้เก็บข้อมูลว่าเป็น node อันที่เท่าไหร่ของลายเซนต์นั้นหรือข้อมูลลายเซนต์ที่เซนต์ครั้งที่เท่าไหร่ของลายเซนต์ของคนนั้น เก็บข้อมูลเป็น integer

ฟิลด์ cnt เป็นข้อมูลบอกว่าเป็นข้อมูลที่เท่าไหร่ของลายเซนต์ครั้งนั้น เก็บข้อมูลในฟิลด์นี้เป็น integer

ฟิลด์ rad ใช้เก็บข้อมูลของลายเซนต์ที่ใช้ในการประมวลผลพิสูจน์ลายเซนต์ เก็บข้อมูลในฟิลด์นี้เป็นค่าจำนวนจริง

ฟิลด์ ax ใช้เก็บข้อมูลความเร่งในแนวนอนของลายเซนต์ที่ใช้ในการประมวลผลพิสูจน์ลายเซนต์ เก็บข้อมูลในฟิลด์นี้เป็นค่าจำนวนจริง

ฟิลด์ ay ใช้เก็บข้อมูลความเร่งในแนวตั้งของลายเซนต์ที่ใช้ในการประมวลผลพิสูจน์ลายเซนต์ เก็บข้อมูลในฟิลด์นี้เป็นค่าจำนวนจริง

เราใช้ฟิลด์ ID,node,cnt ใช้เป็นprimary key เพื่อใช้ระบุข้อมูลที่ใช้ประมวลผล (rad,ax,ay) อันต่างๆ ในลักษณะที่ระบุเหมือนเช่น ข้อมูลนี้เป็นข้อมูลที่เท่าไหร่ของลายเซนต์ครั้งใดของผู้ใด

ข ลักษณะของโครงสร้างของตารางที่ 2 นี้มีดังรูปข้างล่างนี้ ID	Node	Cnt	Rad	Ax	Ay
1	1	1	0.00054	10.05	5.5

ตารางที่สามมีฟิลด์ในตาราง 5 ฟิลด์ คือ ID,npoint,px,py และ bpoint เป็นตารางใช้เก็บข้อมูลที่ใช้สร้างรูปลายเซนต์ที่แท้จริงของผู้ใช้

ฟิลด์ ID ใช้เก็บข้อมูล เลขที่ระบุผู้ใช้ โดยเหมือนกับฟิลด์ ID ในตารางแรกมีไว้เพื่อใช้สร้างความสัมพันธ์กับตารางอื่น เก็บข้อมูลเป็น long integer

ฟิลด์ npoint ใช้เก็บข้อมูลเพื่อระบุลำดับจุดที่ใช้สร้างรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ px ใช้เก็บข้อมูลตำแหน่งพิกัดในแนวนอนของจุดที่ใช้สร้างรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ py ใช้เก็บข้อมูลตำแหน่งพิกัดในแนวนอนของจุดที่ใช้สร้างรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ bpoint ใช้เก็บข้อมูลว่าจุดนั้นเป็นจุดเริ่มกดปากกาในการเซนต์หลังจากที่มีการยกปากกาในการเซนต์เก็บข้อมูลเป็น small integer ถ้าค่าเป็น 1 แสดงว่าจุดนั้นเป็นจุดเริ่มกดปากกา ถ้าค่าเป็น 0 แสดงว่าไม่ใช่

ลักษณะของโครงสร้างของตารางที่ 3 นี้มีดังรูปข้างล่างนี้

ID	npoint	px	py	bpoint
1	1	44	70	1

ตารางที่สี่มีฟิลด์ในตาราง 4 ฟิลด์ คือ npoint,px,py และ bpoint เป็นตารางใช้เก็บข้อมูลที่ใช้สร้างรูปลายเซนต์ที่นำมาพิสูจน์ของผู้ใช้ เป็นการเก็บข้อมูลชั่วคราวเก็บรูปลายเซนต์ที่นำมาพิสูจน์ลายเซนต์เดียวเท่านั้น

ฟิลด์ npoint ใช้เก็บข้อมูลเพื่อระบุลำดับจุดที่ใช้สร้างรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ px ใช้เก็บข้อมูลตำแหน่งพิกัดในแนวนอนของจุดที่ใช้สร้ายรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ py ใช้เก็บข้อมูลตำแหน่งพิกัดในแนวนอนของจุดที่ใช้สร้ายรูปลายเซนต์ เก็บข้อมูลเป็น small integer

ฟิลด์ bpoint ใช้เก็บข้อมูลว่าจุดนั้นเป็นจุดเริ่มกดปากกาในการเซนต์หลังจากที่มีการยกปากกาในการเซนต์เก็บข้อมูลเป็น small integer ถ้าค่าเป็น 1 แสดงว่าจุดนั้นเป็นจุดเริ่มกดปากกา ถ้าค่าเป็น 0 แสดงว่าไม่ใช่ ลักษณะของโครงสร้างของตารางที่ 4 นี้มีดังรูปข้างล่างนี้

npoint	px	py	bpoint
1	10	34	1

ตารางที่ห้ามีฟิลด์ในตาราง 4 ฟิลด์ คือ ID,node,cnt และ point

ฟิลด์ ID ใช้เก็บข้อมูล เลขที่ระบุผู้ใช้ โดยเหมือนกับฟิลด์ ID ในตารางแรกมีไว้เพื่อใช้สร้างความสัมพันธ์ให้ทั้งสองตาราง เก็บข้อมูลเป็น long integer

ฟิลด์ node ใช้เก็บข้อมูลว่าเป็น node อันที่เท่าไหร่ของลายเซนต์นั้นหรือข้อมูลลายเซนต์ที่เซ็นต์ครั้งที่เท่าไหร่ของลายเซนต์ของคนนั้น เก็บข้อมูลเป็น integer

ฟิลด์ cnt เป็นข้อมูลบอกว่าเป็นข้อมูลที่เท่าไหร่ของลายเซนต์ครั้งนั้น เก็บข้อมูลในฟิลด์นี้เป็น integer

ฟิลด์ point ใช้เก็บข้อมูลของลายเซนต์ระบุว่ากดหรือยกปากกา โดยถ้าค่าเป็น1 แสดงว่ากดปากกา ถ้าเป็น 0 แสดงว่ายกปากกา ข้อมูลในฟิลด์นี้เป็นจำนวนเต็มบวก

เราใช้ฟิลด์ ID,node,cnt ใช้เป็นprimary key เพื่อใช้ระบุข้อมูลที่ใช้ประมวลผล (point) อันต่างๆ ในลักษณะที่ระบุเหมือนเช่น ข้อมูลนี้เป็นข้อมูลที่เท่าไหร่ของลายเซนต์ครั้งใดของผู้ใด

ซึ่งลักษณะของโครงสร้างของตารางที่ 2 นี้มีดังรูปข้างล่างนี้

ID	Node	Cnt	Point
1	1	1	1

บทที่ 4

การทดลองและผลการทดลอง

ลักษณะการเก็บข้อมูล

การเก็บข้อมูลเราทำการเก็บข้อมูลโดยให้ผู้ใช้เซนต์ลายเซนต์ตัวอย่างให้แก่โปรแกรม 5 ครั้ง แล้วทำการประมวลผลข้อมูลตัวอย่างนั้นกับข้อมูลลายเซนต์ที่ผู้ใช้เซนต์ทดสอบ โดยให้ผู้ใช้เซนต์ลายเซนต์ที่แท้จริงไปให้โปรแกรมทดสอบ 5 ครั้ง และมีการทดสอบปลอมลายเซนต์ของผู้ใช้คนนั้นอีก 5 ครั้ง ได้ผลการทดลอง การตรวจสอบลายเซนต์จริงและปลอมดังตารางข้างล่าง

ผลการทดลอง

ผู้เซนต์คนที่	ยอมรับลายเซนต์จริง	ปฏิเสธลายเซนต์ปลอม
1	4	4
2	4	4
3	5	4
4	5	3
5	5	4
6	5	5
7	4	3
8	4	4
9	5	4
10	4	3

จะเห็นได้ว่าอัตราการยอมรับลายเซนต์ที่แท้จริงที่ถูกต้องมีค่าประมาณ 90 เปอร์เซ็นผิดพลาดไปประมาณ 10 เปอร์เซ็นและอัตราการปฏิเสธลายเซนต์ปลอมมีค่าประมาณ 76 เปอร์เซ็น ผิดพลาดไปประมาณ 24 เปอร์เซ็น

จากการทดลองเก็บข้อมูลพบว่าตอนเก็บข้อมูลลายเซนต์ 5 ครั้งลงฐานข้อมูลทำได้ช้ามากเพราะเก็บข้อมูลประมาณ 1,000 เรคคอร์ดต่อคน

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

ระบบยังมีข้อผิดพลาดอยู่บ้างเนื่องจากสาเหตุต่อไปนี้

1.การรับข้อมูลจาก digitizer ยังไม่เป็นที่น่าพอใจเนื่องจากวินโดว์มีการใช้อิเทอร์รัพจำนวนมากและมีการทำงานที่ซับซ้อน ทำให้ timer object ของ Delphi ทำงานไม่ถูกต้องตามที่ต้องการ มีอัตราการสุ่มข้อมูลต่ำกว่าที่ต้องการ

2. ข้อจำกัดของระบบ กระดานอิเล็กทรอนิกส์ที่ใช้ยังไม่สะดวกต่อการใช้งาน เนื่องจากต้องใช้แรงกดมากทำให้มีปัญหาแก่ผู้ใช้ เกิดปัญหาเส้นขาด ไม่สามารถบังคับขนาดของลายเซนต์ได้ ใช้ระยะเวลาในการเซนต์มาก แม้แต่คนที่ใช้ลายเซนต์อยู่บ่อยก็ไม่สามารถใช้ลายเซนต์ให้เหมือนเดิมได้

3. ผลการทดลองการหมุนลายเซนต์ที่ไม่ตรงสามารถที่จะหมุนลายเซนต์ที่ไม่ตรงให้เป็นไปตามต้องได้ในทุกกรณี

4.บางครั้งลายเซนต์ในฐานข้อมูลมีความต่างมากเกินไปทำให้มีค่า threshold ต่ำและจะรับลายเซนต์ปลอมได้ง่าย

ข้อแนะนำ

1.ควรเขียนmoduleเพื่อใช้รับข้อมูลจากdigitizerเองโดยอาจใช้ภาษา assembly เพื่อให้มีอัตราการสุ่มมากขึ้น

2.ควรมีการเลือกเฟ้นข้อมูลที่จะเก็บลงฐานข้อมูลก่อนลายเซนต์ใดมีค่าต่างจากตัวอื่นมากควรตัดออก

ภาคผนวก ก.

คู่มือการใช้โปรแกรม

พิสูจน์ลายเซนต์ด้วยนิวรอนเน็ตเวิร์ก

คู่มือการใช้งาน

โปรแกรมโครงงาน

พิสูจน์ลายเซนต์ด้วยนิวรอนเน็ตเวิร์ก

โดย

ปวีณ ศรีทอง

และ

ปัญญา ว่องไว

อุปกรณ์ที่ใช้กับโปรแกรม
พิสูจน์ลายเซนต์ด้วยนิวรอนเนตเวิร์ก

อุปกรณ์ที่จำเป็นต้องใช้กับโปรแกรมมีดังนี้

1. เครื่องคอมพิวเตอร์ แบบ PC ตั้งแต่รุ่น DX-33 ขึ้นไป มี RAM 8 MB. ขึ้นไปมี serial port แบบ 25 ขา

2. โปรแกรมวินโดว์เวอร์ชั่น 3.1 หรือ 3.11 เพราะโปรแกรมนี้ใช้งานในวินโดว์

3. เครื่องดิจิไตเซอร์และปากกาแสงใช้สำหรับรับข้อมูลลายเซนต์โดยเซนต์ด้วยปากกาแสง

4. ไดรเวอร์สำหรับใช้งานเครื่องดิจิไตเซอร์และปากกาแสงที่ใช้งานในวินโดว์ แทนการใช้งานของเมาส์ในวินโดว์

5. แผ่น install โปรแกรมพิสูจน์ลายเซนต์ด้วยนิวรอนเน็ตเวิร์ก

การติดตั้งโปรแกรมพิสูจน์ลายเซนต์ ด้วยนิวรอนเน็ตเวิร์ก

ในแผ่นโปรแกรมจะมีไฟล์ vsign.zip อยู่ให้ทำการติดตั้งโปรแกรมลงสู่เครื่องคอมพิวเตอร์โดยวิธีการดังต่อไปนี้

1. สร้างไดเร็กทอรี่ ชื่อ vsign ลงในฮาร์ดดิสก์ที่ต้องการทำการติดตั้ง

2. ใช้คำสั่ง cd vsign เพื่อเข้าไปในไดเร็กทอรี่ vsign แล้วพิมพ์คำสั่งดังข้างล่างนี้

a:pkunzip -d a:vsign.zip d:

พิมพ์ในลักษณะคล้ายกันนี้ถ้าไดรว์ต้นทางและปลายทางเปลี่ยนไปก็ให้แทนไดรว์ที่ระบุต้นทางและปลายทางเสียใหม่ในลักษณะเดียวกันนี้

3. ทำการแก้ไขไฟล์ WIN.INI ของโปรแกรมวินโดรว์โดยเติมข้อความต่อไปนี้ต่อท้ายไฟล์ WIN.INI ดังนี้

[IDAPI]

DLLPATH=c:\vsign\dll

[Borland Language Drivers]

LDPath=c:\vsign\langdrv

โดยอาจจะระบุไดรว์เป็นไดรว์อื่นที่ไม่เป็นไดรว์ c: ก็ได้ถ้าไม่ได้ลงโปรแกรมในไดรว์ c:

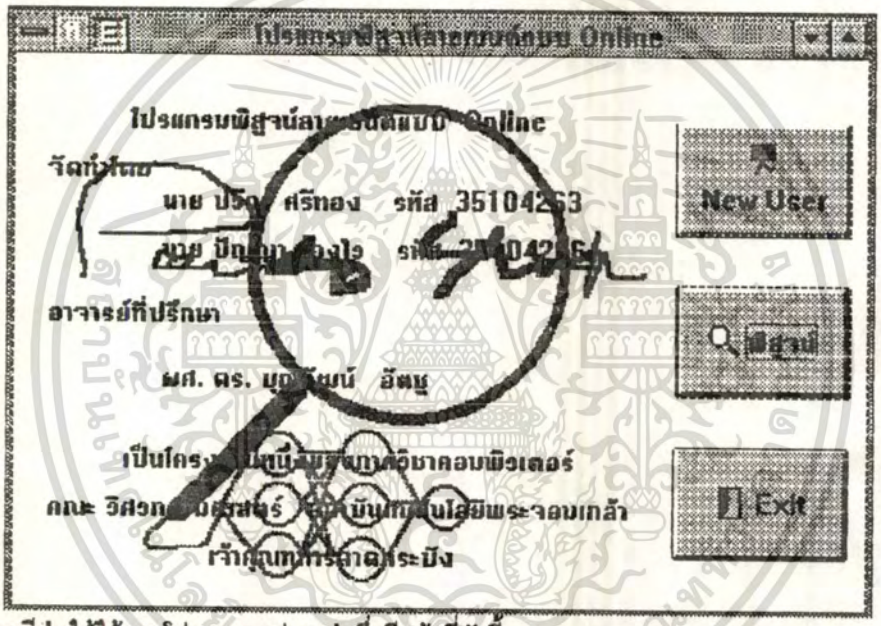
4. หลังจากนั้นก็ทำการ New program item โดยเลือกที่เมนูบาร์ของ Program manager ของโปรแกรมวินโดว์ที่ File แล้วเลือก New... แล้วเลือก program item แล้วไปเลือกเอาไฟล์ชื่อ finalprj.exe จากในไดเร็กทอรี่ vsign มา

5. หลังจากเลือกไอคอนและติดตั้งโปรแกรมในวินโดว์แล้วเราสามารถใช้โปรแกรมในวินโดว์โดยต้องทำการติดตั้งไดรเวอร์ของดิจิไตเซอร์โดยเรียก setup ของวินโดว์แล้วทำการเพิ่มไดรเวอร์เมาส์โดยเลือกที่หัวข้อเมาส์แล้วเลือก Other mouse แล้วใส่ pathname ที่มีไดรเวอร์ของดิจิไตเซอร์นั้นอยู่ลงไปเพื่อจะได้ลงไดรเวอร์แก่วินโดว์ หลังจากลงแล้วเลือกไดรเวอร์เมาส์ของวินโดว์เป็น Oce G6400 ทำการติดตั้งไดรเวอร์เสร็จแล้วปิดเครื่องดิจิไตเซอร์ก่อนเรียกใช้วินโดว์ เมื่อเรียกใช้ในโดว์แล้วจะสามารถใช้งานปากกาแสงแทนเมาส์ได้ โดยการกดปากกาแสงจะเหมือนกับการคลิกเมาส์ หลังจากนี้สามารถใช้งานโปรแกรมได้ เมื่อใช้งานโปรแกรมเสร็จก็เปลี่ยนไดรเวอร์จากดิจิไตเซอร์เป็นเมาส์ตามเดิมได้

การใช้งานโปรแกรมพิสูจน์ลายเซนต์

หลังจากที่ลงโปรแกรมนี้เรียบร้อยแล้ว สามารถใช้โปรแกรมได้โดยใช้งานโปรแกรมในวินโดว์โดยทำการดับเบิ้ลคลิกส์ที่ไอคอนดังรูปข้างล่างนี้

หลังจากที่กดใช้งานโปรแกรมที่ไอคอนแล้วจะปรากฏฟอร์ม ดังนี้ขึ้น



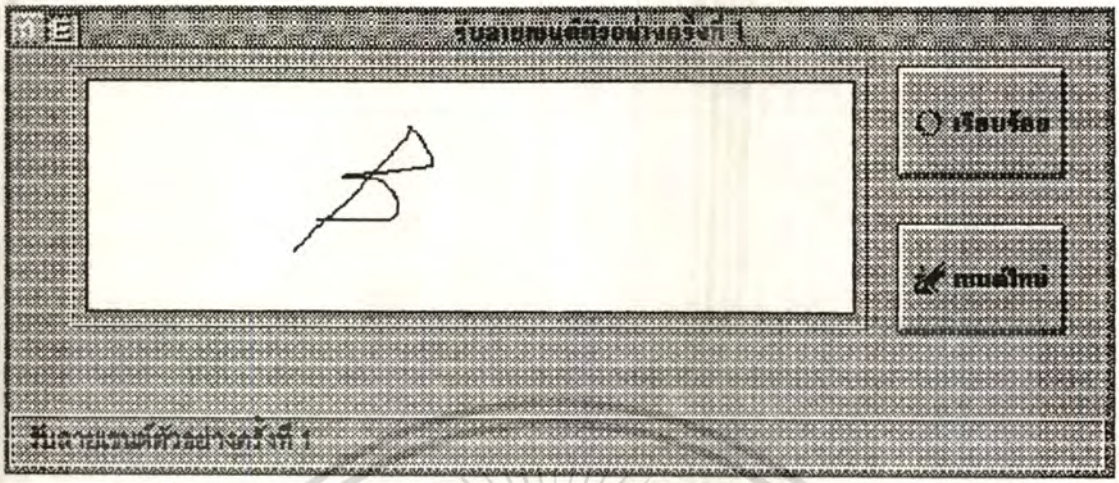
โดยมีปุ่มให้ใช้งานโปรแกรมอยู่ 3 ปุ่มซึ่งมีหน้าที่ดังนี้

- ปุ่ม 'New user' ใช้สำหรับเพิ่มลายเซนต์ของผู้ใช้ใหม่ เข้าไปในฐานข้อมูลลายเซนต์
- ปุ่ม 'พิสูจน์' ใช้สำหรับทำการพิสูจน์ลายเซนต์ที่ผู้ใช้เซนต์ทดสอบกับข้อมูลลายเซนต์ที่ผู้ใช้อ้างถึง
- ปุ่ม 'Exit' ใช้เมื่อต้องการเลิกใช้งานโปรแกรม

การเพิ่มข้อมูลลายเซนต์ใหม่

สามารถที่จะเพิ่มลายเซนต์ใหม่ได้โดยกดปุ่ม New User จะทำการเพิ่มลายเซนต์ใหม่ที่ยังไม่มีในฐานข้อมูลโดยผู้ใช้จะต้องกรอกชื่อและนามสกุลในแบบฟอร์มข้างล่างนี้

โดยเมื่อใส่ชื่อและนามสกุลเสร็จแล้วก็กดปุ่มที่มีข้อความว่าเรียบร้อย เพื่อจะได้สามารถตรวจสอบได้ว่ามีข้อมูลเดิมของคนชื่อนี้นามสกุลนี้อยู่หรือไม่ ถ้าไม่มีก็จะเริ่มทำการเก็บข้อมูลลายเซนต์โดยเก็บใช้แบบฟอร์มข้างล่างนี้

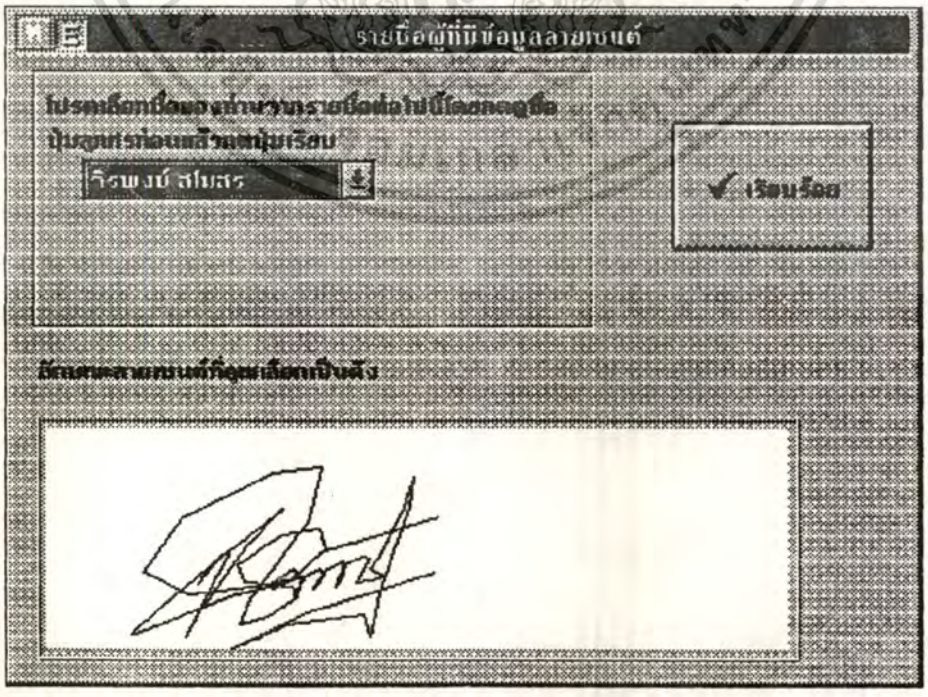


สำหรับแบบฟอร์มรับลายเซนต์ตัวอย่างนี้ใช้งานโดยผู้ใช้เซนต์ลายเซนต์ลงไปในกรอบสี่เหลี่ยมขาว เมื่อเซนต์เสร็จแล้วถ้าลายเซนต์ไม่ถูกต้องสามารถเซนต์ใหม่ได้โดยกดปุ่ม 'เซนต์ใหม่' จะสามารถเซนต์ใหม่ได้อีกจนกว่าจะพอใจลายเซนต์นั้น ถ้าลายเซนต์ที่เซนต์นั้นถูกต้องแล้วก็กดปุ่ม 'เรียบร้อย' เพื่อเก็บข้อมูลตัวอย่างลายเซนต์ ทำเช่นนี้ไป 5 ครั้งเพื่อเก็บตัวอย่างลายเซนต์ เมื่อเก็บข้อมูลลายเซนต์ครบ 5 ครั้งแล้วสามารถก็เป็นอันเสร็จขั้นตอนการเก็บข้อมูลลายเซนต์ สามารถนำข้อมูลลายเซนต์นี้ไปใช้ในขั้นตอนพิสูจน์ได้ต่อไป

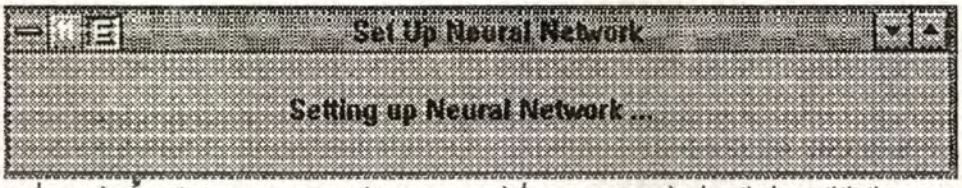
การทำการพิสูจน์ลายเซนต์

ผู้ใช้จะทำการพิสูจน์ลายเซนต์ได้โดยกดปุ่ม 'พิสูจน์' ในแบบฟอร์มข้างล่าง

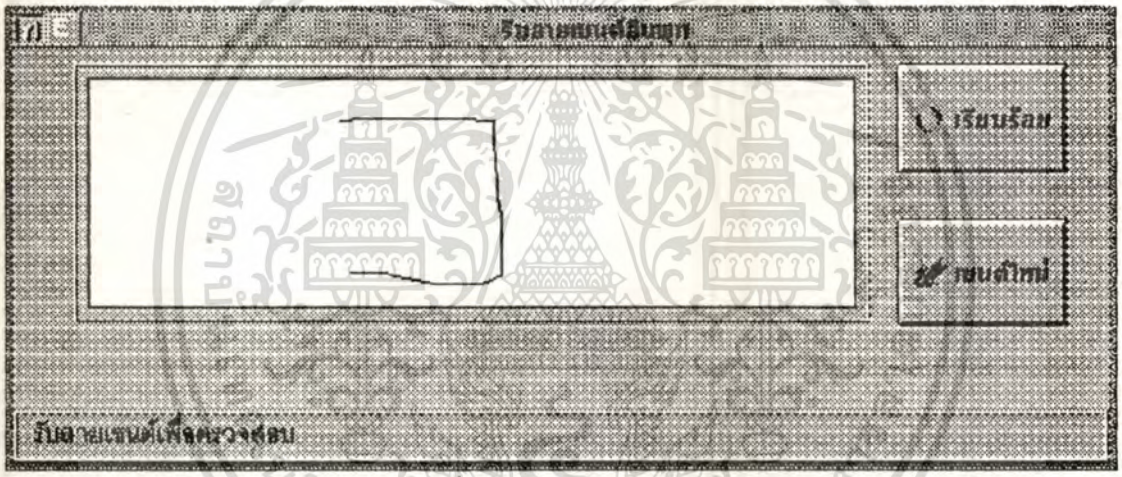
เมื่อกดปุ่มพิสูจน์แล้วผู้ใช้จะต้องเลือกรายชื่อของผู้ใช้จากแบบฟอร์มข้างล่างเพื่อทำการพิสูจน์ลายเซนต์ของผู้ใช้ชื่อและนามสกุลนั้นๆ



เมื่อเลือกรายชื่อผู้ใช้เสร็จแล้วให้กดปุ่ม 'เรียบร้อย' เพื่อทำในขั้นตอนการเตรียมข้อมูลลายเซนต์สำหรับใช้ในการพิสูจน์ลายเซนต์โดยขณะที่ทำการเตรียมข้อมูลจะปรากฏฟอร์มข้างล่างนี้ขึ้น



เมื่อฟอร์มนี้หายไปแสดงว่าเตรียมข้อมูลลายเซนต์เพื่อทำการพิสูจน์เสร็จแล้วก็จะให้ผู้ใช้ click หนึ่งครั้งที่ฟอร์มแล้วจะปรากฏแบบฟอร์มที่ทำการเซนต์ลายเซนต์เพื่อพิสูจน์ตามแบบฟอร์มข้างล่างนี้



โดยแบบฟอร์มนี้จะใช้งานเหมือนกับแบบฟอร์มที่ใช้เก็บข้อมูลลายเซนต์ โดยให้ผู้ใช้เซนต์ลายเซนต์ลงในกรอบสีขาว ถ้าเซนต์ใช้ได้แล้วให้กดปุ่ม 'เรียบร้อย' เพื่อทำการพิสูจน์ลายเซนต์แต่ถ้ายังเซนต์ไม่ดีก็สามารถเซนต์ใหม่ได้โดยกดปุ่ม 'เซนต์ใหม่' เมื่อเซนต์ลายเซนต์และกดปุ่ม 'เรียบร้อย' แล้วโปรแกรมก็จะแสดงผลการพิสูจน์ลายเซนต์ออกมาดังในแบบฟอร์มข้างล่างนี้

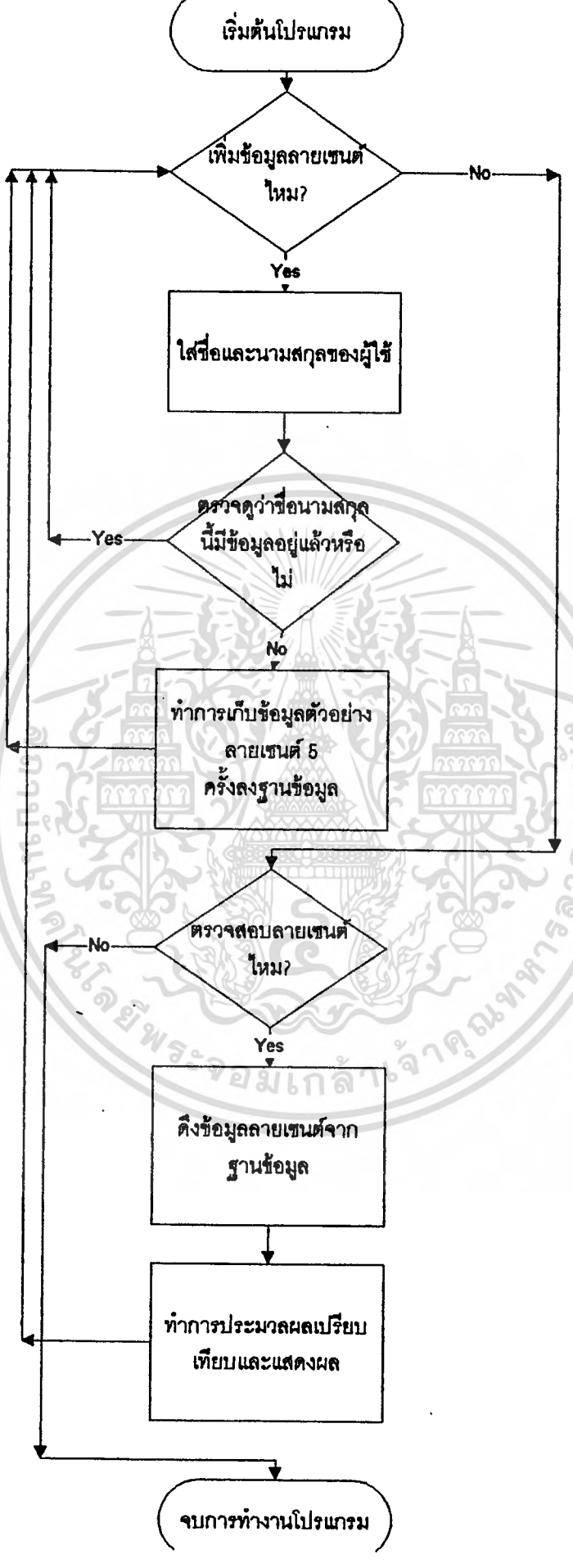
เมื่อกดปุ่มตรวจสอบโปรแกรมก็จะแสดงผลการตรวจสอบว่ายอมรับหรือไม่ เมื่อทราบผลแล้วท่านผู้ใช้สามารถกดปุ่ม 'เรียบร้อย' เพื่อไปทำการเพิ่มข้อมูลลายเซนต์ใหม่หรือทำการพิสูจน์ลายเซนต์อีกครั้งได้ในฟอร์มที่ปรากฏขึ้นตอนแรกๆ ได้อีก หรือถ้าจะดูภาพลายเซนต์ตัวอย่างอันอื่นๆ ก็เลือกเบอร์ภาพได้จากลายชื่อจำนวนของลายเซนต์ที่แท้จริงได้ด้วย และสามารถดูผลการเปรียบเทียบระหว่างลายเซนต์ที่ใช้ทดสอบกับค่าเกณฑ์เปรียบเทียบของแต่ละลายเซนต์ตัวอย่างได้ดังรูปข้างล่าง

ผลการตรวจสอบ
ค่าเกณฑ์ต่ำสุดของลายครั้งถัดต่าง
ค่าเกณฑ์ต่ำสุด 7.15702458
ดีคือ 0.64417928
ค่าของลายเซนต์ที่ป้อนเข้ามา 2.23929649
ผลการตรวจสอบ : ไม่ยอมรับ
ตรวจสอบ
เรียบร้อย
ตัวอย่างลายเซนต์ในฐานข้อมูล
3
ลายเซนต์ที่ให้ป้อน

ภาคผนวก ข.

ไดอะแกรมแสดงการทำงาน

และแนวความคิด


เริ่มต้นโปรแกรม
เพิ่มข้อมูลลายเซนต์ไหม?
No
Yes
ใส่ชื่อและนามสกุลของผู้ใช้
ตรวจดูว่าชื่อนามสกุลนี้มีข้อมูลอยู่แล้วหรือไม่
Yes
No
ทำการเก็บข้อมูลตัวอย่างลายเซนต์ 5 ครั้งลงฐานข้อมูล
ตรวจสอบลายเซนต์ไหม?
No
Yes
ดึงข้อมูลลายเซนต์จากฐานข้อมูล
ทำการประมวลผลเปรียบเทียบและแสดงผล
จบการทำงานโปรแกรม

ไดอะแกรมของการหมุนลายเซนต์

ไดอะแกรมแสดงการทำลายเซนต์ให้อยู่ในขนาดมาตราฐาน

ไดอะแกรมแสดงการเพิ่มข้อมูลจุด

กฤติกรรมประกาศ

ขอขอบคุณอาจารย์ บุญวัฒน์ อัตชู สำหรับความช่วยเหลือและความเมตตา ตลอด 2 เทอมที่ผ่านมาเป็นระยะเวลายาวนานพอควร

ปวีณขอขอบคุณ

คุณนุช,น้องแตง สำหรับกำลังใจที่ทำให้กระเสือกกระสนจนงานเสร็จจนได้

น้องเก่ที่ทำให้พี่ (ปวีณ) นอนไม่หลับทำให้ต้องลุกขึ้นมาทำโปรเจคอยู่เนืองๆ

ปัญญาขอขอบคุณ

เพื่อนๆ ห้องโปรเจคอาจารย์ ศุภมิตร ที่เอื้อเฟื้อการใช้เครื่องและคำแนะนำในการใช้ Delphi ตลอดมา โดยเฉพาะสมาน (ณัฐ)

Special Thank นายทาร์ที่เอื้อเฟื้อข้อมูลทางคณิตศาสตร์ และคอยช่วยซื้ออาหารมาให้ในตอนดึกขณะที่ทำโปรเจคกำลังหิวอยู่

และขอขอบคุณทุกคนที่เป็นแรงใจที่ช่วยให้กระเสือกกระสนทำโปรเจคนี้

เอกสารอ้างอิง

ก. <u>เอกสารอ้างอิงที่เป็นวารสารภาษาอังกฤษ</u> จัดเรียงตามลำดับดังนี้

1. Wacef Guerfali and Rejean Plamondon , 'Normalize and Restoring on - line handwriting', *Pattern Recognition* , Vol. 26 , No. 3 , pp. 419-431 , 1993

ข. <u>เอกสารอ้างอิงที่เป็นหนังสือภาษาไทยและภาษาอังกฤษ</u> จัดเรียงตามลำดับดังนี้

1. Philip D. Wasserman , ' Advanced Methods in Neural Computing' , VAN NOSTRAND REIHOLD , New York, 255 p. , 1993.
2. ปิยมาศ โง้วมณี , "การวิเคราะห์ลายเซนต์" , คณะวิทยาศาสตร์ สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง , 117 หน้า , 2538
3. Ullmann J.R.Ullmann , "PATTERN RECOGNITION TECHNIQES", Butterworths , London , 403 p., 1973