การวิเคราะห์เสียงโดยใช้โครงข่ายประสาทเทียมสำหรับการประมวลผลคำ

Artificial Neural Network based speech synthesis for word processing

นางสาวกาญจนา	ดีรัศมี	รหัส 45360047
นายพีรพล	สืบสุรีย์กุล	รหัส 45360328
นายวิทยา	ทองอ่อน	รหัส 45360419

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การวิเคราะห์เสียงโดยใช้โครงข่ายประสาทเทียมสำหรับการประมวลผลคำ		
ผู้ดำเนินโครงงาน	นางสาวกาญจนา	ดีรัศมี	รหัส 45360047
	นายพีรพล	สืบสุรีย์กุล	รหัส 45360328
	นายวิทยา	ทองอ่อน	รหัส 45360419
อาจารย์ที่ปรึกษา	ดร. สมยศ	เกียรติวนิชวิไล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.สมยศ เกียรติวนิชวิไล)

.....กรรมการ

(ดร.สุรเชษฐ์ กานต์ประชา)

.....กรรมการ

(อาจารย์พนัส นัถฤทธิ์)

หัวข้อโครงงาน	การวิเคราะห์เสียงโดยใช้โครงข่ายประสาทเทียมสำหรับการประมวลผลคำ		
ผู้ดำเนินโครงงาน	นางสาวกาญจนา	ดีรัศมี	รหัส 45360047
	นายพีรพล	สืบสุรีย์กุล	รหัส 45360328
	นายวิทยา	ทองอ่อน	รหัส 45360419
อาจารย์ที่ปรึกษา	ดร.สมยศ	เกียรติวนิชวิไล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

บทคัดย่อ

โครงงานนี้พัฒนาโปรแกรมสังเคราะห์เสียงพูดของมนุษย์ เพื่อนำไปใช้ประโยชน์ในการประมวลผลคำ การจำแนกและเปรียบเทียบเสียง โดยใช้ทฤษฎีโครงข่ายประสาทเทียม ทำให้เปรียบเทียบและจำแนกความแตกต่างระหว่างคำ อย่างไรก็ตามเสียงพูดของคำคำเดียวกันในเวลาและบุคคลที่พูดต่างกัน จะให้สัญญาณเสียงที่ไม่เหมือนกัน แต่จะมีความคล้ายคลึงกันเป็นส่วนใหญ่ ซึ่งสามารถรู้จำ และเปรียบเทียบความคล้ายของสัญญาณ ได้โดยใช้ทฤษฎีโครงข่ายประสาทเทียม โปรแกรมที่พัฒนาขึ้นอาศัยภาษาวิชวลเบสิค ในการสร้างโปรแกรมเพื่อรับอินพุตเป็นเสียงพูดจากไมโครโฟน ผ่านกระบวนการสังเคราะห์เสียงด้วยโครงข่ายประสาทเทียม จากผลที่ได้แสดงให้เห็นถึงความสามารถของโปรแกรมที่พัฒนาขึ้นในการสังเคราะห์เสียง

Project title	Artificial Neural Network based speech synthesis for word processing
Name	Miss Kanjana Deerussamee ID. 45360047
	Mr. Peerapol Suebsureekul ID. 45360328
	Mr. Wittaya Thong-on ID. 45360419
Project advisor	Dr. Somyot Kiattivanichvilai
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2005

.....

Abstract

This project develops a speech synthesis program for word processing. A neural network is used for classify and compare the different between words. However, the speech of the same word in the different time or different speaker is often not exactly same but has a few different. In this project, neural network is applied to solve the problem of recognition and similarity comparison. The developed program is using Visual Basic. Several speeches from a microphone is synthesized and used to train a neural network. The result of program show the ability of speech signal analyzer.

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร. สมยศ เกียรติวนิชวิไล อาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษา ความช่วยเหลือตลอดจนคำแนะนำต่างๆในการทำโครงงานชิ้นนี้ สุดท้ายต้องขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆทุกคนที่ยังไม่ได้เอ่ยนามที่ให้การสนับสนุนผู้จัดทำโครงงานให้สามารถทำโครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

นางสาวกาญจนา ดีรัศมี

นายพีรพล สืบสุรีย์กุล

นายวิทยา ทองอ่อน

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญรูป

สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงการ

เนื่องจากความเจริญก้าวหน้าทางด้านเทคโนโลยีในปัจจุบันมีความเจริญก้าวหน้ามากขึ้น โดยเฉพาะความเจริญก้าวหน้าเกี่ยวกับการนำคอมพิวเตอร์มาใช้งานในด้านต่างๆ คอมพิวเตอร์ในปัจจุบัน จึงเป็นอุปกรณ์ที่มีความสำคัญสำหรับบุคคลในทุกเพศทุกวัย และในปัจจุบันการออกแบบโปรแกรมต่างๆ ที่ใช้งานกับคอมพิวเตอร์มีการออกแบบโดยให้มีการใช้คีย์บอร์ดและเมาส์ในการใช้งานเพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายและสะดวกขึ้น ฉะนั้นการใช้งานคอมพิวเตอร์สำหรับคนบางกลุ่มอย่างเช่น คนที่มีความพิการเกี่ยวกับมือ ก็จะเป็นการยากในการในการใช้งานทางคอมพิวเตอร์เพราะ ในการควบคุมการทำงานของคอมพิวเตอร์นั้นจะใช้เมาส์และคีย์บอร์ด ในการควบคุมส่วนใหญ่ เป็นหลักซึ่งอาจจะเป็นปัญหาสำหรับผู้ที่มีความพิการทางระบบสัมผัส นอกจากนี้ผู้ที่ไม่มีความชำนาญในการใช้คีย์บอร์ดและเมาส์ก็อาจจะมองว่าการทำงานทางคอมพิวเตอร์โดยใช้คีย์บอร์ดและเมาส์ใช้งานไม่สะดวกสบาย จากแนวความคิดข้างต้น คณะผู้จัดทำได้มีแนวความคิดที่จะทำให้บุคคลกลุ่มดังกล่าว สามารถใช้งานคอมพิวเตอร์ได้ง่ายขึ้น และสะดวกสบายยิ่งขึ้น

อนึ่ง จากการที่ได้ศึกษาลักษณะของคำพูดในภาษาไทยพบว่า แต่ละคำมีลักษณะที่แตกต่างกันซึ่งมาจากพยัญชนะและสระที่ต่างกัน จึงสามารถนำมาวิเคราะห์ความแตกต่าง เพื่อนำไปใช้ประโยชน์ได้ จากแนวความรู้นี้ คณะผู้จัดทำจึงมีแนวความคิดที่จะใช้ลักษณะของความแตกต่างของคำพูดมาใช้ประโยชน์ในการใช้งานทางคอมพิวเตอร์ได้ง่ายขึ้น แทนการป้อนอินพุตผ่านคีย์บอร์ดและเมาส์ เพื่อให้ผู้ที่มีความพิการเกี่ยวกับมือและผู้ที่ไม่ชำนาญในการใช้คีย์บอร์ดและเมาส์มีความสะดวกในการใช้งานคอมพิวเตอร์มากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาหลักการของโครงข่ายประสาทเทียมเพื่อนำมาใช้ในการวิเคราะห์คำ

1.2.2 เพื่ออำนวยความสะดวกให้แก่ผู้ที่ไม่สะดวกในการใช้งานคอมพิวเตอร์ผ่านทางคีย์บอร์ดหรือเมาส์

1.2.3 เพื่อศึกษาและนำความรู้เกี่ยวกับการเขียนโปรแกรมภาษาวิชวลเบสิค มาใช้ประโยชน์ให้เกิดประโยชน์สูงสุด

1.3 ขอบข่ายของโครงงาน

1.3.1 สร้างโปรแกรมจากภาษาวิชวลเบสิคเพื่อวิเคราะห์คำพูดและนำผลลัพธ์ของคำพูดที่ได้เปิดใช้งานโปรแกรม

1.3.2 ประยุกต์ใช้ทฤษฎีโครงข่ายประสาทเทียมมาใช้ในการรู้จำและจำแนกคำพูด

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาความเป็นไปได้ของโครงงาน

1.4.2 รวบรวมข้อมูลที่เกี่ยวข้องกับโครงงานเรื่องโครงข่ายประสาทเทียมและการวิเคราะห์คำ

1.4.3 ทดสอบความเป็นไปได้ของการจัดทำโครงงาน

1.4.4 ออกแบบและจัดทำโปรแกรมโดยใช้ภาษาวิชวลเบสิค

1.4.5 ทดสอบการทำงานของโปรแกรมที่พัฒนาขึ้น

1.4.6 สรุปผลของโครงงานและจัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

กิจกรรม	ปี2547		ปี2548									
	พ..ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1.ศึกษาค้นคว้าข้อมูลเกี่ยวกับทฤษฎี โครงข่ายประสาทเทียม ในส่วนที่เกี่ยวข้องกับการวิเคราะห์เสียงพูดของมนุษย์	⟷	⟷										
2. ออกแบบรูปแบบการวิเคราะห์และจำแนกความแตกต่างของคำ			⟷	⟷								
3. ทดสอบความเป็นไปได้ในการจัดทำโครงงาน					⟷	⟷						
4. เขียนโปรแกรมในส่วนของการรับอินพุตเสียง และการวิเคราะห์คำ							⟷	⟷	⟷			
5. ทดสอบการทำงาน										⟺		
6. สรุปผลการทำงานและจัดทำรูปเล่มโครงงาน											⟷	⟷

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถเข้าใจหลักการทำงานของโครงข่ายประสาทเทียมในส่วนที่เกี่ยวข้องกับการวิเคราะห์และจำแนกความแตกต่างของคำ

1.6.2 สามารถนำทฤษฎีโครงข่ายประสาทเทียมไปใช้ในการออกแบบและพัฒนาโปรแกรมวิเคราะห์เสียงพูดภาษาไทยได้

1.6.3 สามารถสร้างและพัฒนาโปรแกรมโดยใช้ภาษาวิชวลเบสิค เพื่อนำมาใช้ในการวิเคราะห์คำภาษาไทยได้

1.7 งบประมาณของโครงงาน

1.7.1 ค่าหนังสือเกี่ยวกับทฤษฎีโครงข่ายประสาทเทียม

1.7.2 ค่าถ่ายเอกสารและค่าจัดทำรูปเล่มโครงงาน

1.7.3 ค่าหมึกพิมพ์

1.7.4 ค่าใช้จ่ายอื่นๆที่เกี่ยวข้อง

รวมเป็นเงิน 3,000 บาท (สามพันบาทถ้วน)

บทที่ 2
ทฤษฎีโครงข่ายประสาทเทียม

ดร. โรเบิร์ท เฮช์ท-นิลเซน (Dr. Robert Hecht-Nielsen) ซึ่งเป็นผู้ประดิษฐ์คอมพิวเตอร์เชิงปัญญา (newrocomputer) เครื่องแรกๆ ของโลก ได้ให้คำจำกัดความโครงข่ายประสาทเทียมไว้ว่า "ระบบการคำนวณชนิดหนึ่งซึ่งสร้างขึ้นจากหน่วยประมวลผลอย่างง่าย จำนวนมากที่เชื่อมต่อเข้าด้วยกันอย่างหนาแน่น และประมวลผลสารสนเทศโดยการตอบสนองต่อข้อมูลจากภายนอก ในสถานะที่ไม่คงตัว" [จากบทความเรื่อง Neural Network Primer: Part 1 โดย Maureen Candill วารสาร AI Expert เดือนกุมภาพันธ์ ค.ศ. ๑๙๘๙]

โครงข่ายประสาทเทียม คือ โครงสร้างของหน่วยประเมินผล (ซึ่งอาจหมายถึงโมดูลซอฟต์แวร์ ซึ่งระบุขั้นตอนการทำงานในลักษณะของโปรแกรมคอมพิวเตอร์ หรือเป็นอุปกรณ์ฮาร์ดแวร์ก็ได้) จำนวนมาก ที่ถูกจำลองขึ้นมาอย่างคร่าวๆ ตามอย่างโครงสร้างของระบบประสาทของสมองส่วนเซเรบรัลคอร์เท็กซ์ (cerebral cortex) ของสัตว์เลี้ยงลูกด้วยนม แต่จะมีขนาดที่เล็กกว่ามาก โครงข่ายประสาทเทียมขนาดใหญ่อาจมีจำนวนหน่วยประมวลผลได้เป็นหลายร้อยหลายพันหน่วย ในขณะที่สมองของสัตว์เลี้ยงลูกด้วยนมมีเซลล์ประสาทนับเป็นพันๆ ล้านเซลล์ การสร้างแบบจำลองของโครงข่ายประสาทขึ้นมาเป็นโครงข่ายประสาทเทียมนั้น ทำได้โดยการพยายามทำความเข้าใจกับกระบวนการทำงานของสมองแล้วพยายามอธิบายการทำงานนั้นด้วยแบบจำลองเชิงคณิตศาสตร์ จากนั้นจึงออกแบบระบบคอมพิวเตอร์ หรือเขียนโปรแกรมคอมพิวเตอร์ที่จะทำงานตามแบบจำลองเชิงคณิตศาสตร์ที่ได้

2.1 ประวัติความเป็นมาของโครงข่ายประสาทเทียม

ในปี พ.ศ. 2486 อาจถือได้ว่า เป็นปีแห่งการกำเนิดของสาขาโครงข่ายประสาทเทียมในวงการวิทยาศาสตร์ โดย แม็คคัลลอช (Mc Culloch) และ พิทส์ (Pitts) ได้เสนอแบบจำลองของเซลล์ประสาทและได้แสดงให้เห็นว่า ในทางทฤษฎีแล้วโครงข่ายของแบบจำลองเซลล์ประสาทดังกล่าวสามารถทำงานเป็นโปรแกรมคอมพิวเตอร์ใดๆ ก็ได้

ปี พ.ศ. 2492 โดนัลด์ เฮบบ์ (Donald Hebb) ได้เสนอผลงานวิจัยว่า การเรียนรู้ของสมองสามารถอธิบายได้ด้วยรูปแบบของการประกอบเซลล์ประสาทเข้าด้วยกันเป็นโครงข่าย และได้เสนอกฎการเรียนรู้ของเฮบบ์ (Hebb's rule) ที่ทำให้โครงข่ายของเซลล์ประสาทเทียมที่แม็คคัลลอชและพิทส์ได้เสนอไว้ สามารถเรียนรู้ปัญหาง่ายๆ ได้สำเร็จ การเรียนรู้ในแบบของเฮบบ์บนเซลล์ประสาทเทียมของแม็คคัลลอชและพิทส์นั้นเป็นการเรียนรู้แบบ "ไม่มีผู้สอน" ซึ่งในทางปฏิบัติแล้วโครงข่ายประสาทเทียมที่ทำการเรียนรู้จะพยายามทำการจัดกลุ่มข้อมูลที่โครงข่ายมองว่าคล้ายคลึง

กันนำไปไว้ในกลุ่มเดียวกัน ซึ่งไม่เหมาะสมกับปัญหาประเภทที่ต้องมีการควบคุมกระบวนการเรียนรู้

ในช่วงพุทธทศวรรษ 2490 คอมพิวเตอร์ที่ทำงานเลียนแบบสมองเครื่องแรกของโลกถูกสร้างและทดสอบโดยมินสกี้ (Minsk) ซึ่งได้เสนอผลงานดังกล่าวในปี พ.ศ. 2511 เมื่อคอมพิวเตอร์ดังกล่าวได้รับการป้อนตัวอย่างสำหรับการเรียนรู้เข้าไป ก็จะสามารถปรับอัตราการขยายสัญญาณในการเชื่อมโยงหรือ "ความแข็งแรงของการเชื่อมโยง" ระหว่างเซลล์ประสาทเทียมได้เองโดยอัตโนมัติ ซึ่งเป็นการแสดงการเรียนรู้ตัวอย่างที่ถูกป้อนเข้าไป

ในปี พ.ศ. 2501 แฟรงค์ โรเซ็นแบลทท์ (Frank Rosenblatt) ได้พัฒนาสถาปัตยกรรมโครงข่ายประสาทเทียมขึ้น โดยใช้แบบจำลองของแม็คคัลลอชและพิทส์เป็นแนวทาง รวมทั้งเสนอวิธีการเรียนรู้แบบใหม่สำหรับสถาปัตยกรรมโครงข่ายประสาทเทียมดังกล่าวด้วย โครงข่ายประสาทเทียมดังกล่าวเรียกว่า เพอร์เซพตรอน(Perceptron) ซึ่งมีการเรียนรู้แบบ "มีผู้สอน" (supervised learning) โดยการปรับความแข็งแรงของการเชื่อมโยง ซึ่งจะพิจารณาได้จากการเปรียบเทียบความรู้ของโครงข่ายประสาทเทียมกับความรู้ของ "ผู้สอน" (teacher) เพอร์เซพตรอนมีความเหมาะสมกับงานประเภท "การระบุชนิด" ซึ่งในระหว่างการเรียนรู้นั้น เพอร์เซพตรอนจะถูกสอนว่าข้อมูลตัวอย่างที่สอนเข้าไปแต่ละแบบนั้นจัดเป็นชนิดใดบ้าง หากปัญหาและข้อมูลตัวอย่างมีความเหมาะสม เพอร์เซพตรอนจะสามารถระบุชนิดของข้อมูลที่ไม่เคยเห็นมาก่อนได้ถูกต้อง

ในช่วงต้นพุทธทศวรรษ 2500 เบอร์นาร์ด วิโดรว (Bernard Widrow) และมาร์เซียน ฮอฟฟ์ (Marcian Hoff) ได้พัฒนาอุปกรณ์ที่เรียกว่า อดาไลน์ (ADALINE;Adaptive Linear combiner) และกฎการเรียนรู้แบบใหม่ที่มีประสิทธิภาพสูงเรียกว่า กฎการเรียนรู้ของวิโดรว-ฮอฟฟ์ (Windrow-Hoff learning rule) ที่เป็นการเรียนรู้แบบ "มีผู้สอน" ซึ่งในเวลาต่อมาอุปกรณ์ดังกล่าวได้รับการขยายแนวคิดไปเป็นมาดาไลน์ (MADALINE; Many ADALINEs) และได้ถูกนำไปประยุกต์ใช้ในการรู้จำรูปแบบ (pattern recognition) การพยากรณ์อากาศ และระบบควบคุมที่จำเป็นต้องมีการปรับเปลี่ยนระบบไปตามสภาพแวดล้อมต่างๆ

การพัฒนาระบบประมวลผลแบบโครงข่ายประสาทเทียมนั้น จะอิงกับแนวทางการประมวลผล ของสมองของสิ่งมีชีวิต ดังนั้น ความเข้าใจในคุณลักษณะเชิงกายภาพ และเชิงพฤติกรรมขององค์ประกอบต่างๆ ในสมองของสิ่งมีชีวิตจึงเป็นสิ่งจำเป็น

หน่วยรากฐานของสมองคือ เซลล์ประสาท (neuron) สมองของมนุษย์ประกอบไปด้วยเซลล์ประสาทจำนวนอย่างน้อยในระดับแสนๆ ล้านเซลล์ในแง่ของการทำงานนั้น เซลล์ประสาทแต่ละเซลล์ คือ หน่วยประมวลผลอย่างง่ายๆ ซึ่งรับสัญญาณและรวมสัญญาณที่ถูกส่งมาจากเซลล์ประสาทอื่นๆ แต่ละเซลล์ประสาทจะมีส่วนหลักๆ อยู่ 3 ส่วน คือ

1. ตัวเซลล์ซึ่งเรียกว่า โซมา (soma) มีลักษณะเป็นรูปทรงพีระมิด หรือทรงกระบอก

2. เดนไดรท์ (dendrite)

เดนไดรท์คือ เส้นใยบางๆ ที่เซลล์ประสาทใช้รับสัญญาณ ไฟฟ้าเข้าสู่เซลล์ แต่ละเซลล์ประสาทจะมีเดนไดรท์จำนวนมากจัดตัวเป็นลักษณะเหมือนกิ่งไม้

3. แอกซอน (Axon)

แอกซอนคือ สายส่งผ่านสัญญาณทรงกระบอกขนาดยาวและใหญ่ ที่เซลล์ประสาทใช้เป็นทางส่งสัญญาณไปยังเซลล์ประสาทอื่นๆ ส่วนปลายของแอกซอนจะแตกออกเป็นกิ่งก้านย่อยๆ โดยที่ส่วนปลายของแต่ละกิ่งก้านเหล่านี้ลักษณะเป็นปม และจะไปจ่ออยู่จนเกือบสัมผัสกับปลายของเดนไดรท์หนึ่งของเซลล์ประสาทเซลล์อื่น

บริเวณที่เป็นรอยต่อระหว่างปลายของแอกซอนกับปลายของเดนไดรท์เรียกว่า ไซแนปส์ (Synapse) สัญญาณไฟฟ้าที่ถูกส่งมาถึงปลายของแอกซอนจะกระตุ้นให้เกิดการส่งผ่านสัญญาณในเชิงเคมีผ่านไซแนปส์ สัญญาณเชิงเคมีดังกล่าวจะถูกเดนไดรท์ตีความเป็นสัญญาณไฟฟ้าวิ่งเข้าสู่เซลล์ประสาทต่อไป

คุณลักษณะสำคัญของไซแนปส์ คือ ความแรงของสัญญาณที่ถูกส่งผ่านจะขึ้นอยู่กับความเหนียวแน่นของการเชื่อมต่อ และสัญญาณที่ถูกส่งผ่านไซแนปส์อาจถูกทำให้มีสภาพเป็นสัญญาณกระตุ้น (excitory) หรือสัญญาณกด (inhibitory) ก็ได้ ขึ้นอยู่กับชนิดของสัญญาณเชิงเคมีที่ถูกกระตุ้นให้เคลื่อนผ่านรอยต่อ ซึ่งแต่ละประสาทอาจรับสัญญาณมาจากหนึ่งหมื่นไซแนปส์ หรือมากกว่า

เซลล์ประสาทเทียมคือ หน่วยรากฐานของโครงข่ายประสาทเทียม เซลล์ประสาทเทียมไม่สามารถใช้เป็นแบบในการอธิบายการทำงานของเซลล์ประสาทของสิ่งมีชีวิตได้ถูกต้อง แต่เป็นการนำเอาแนวคิดที่ได้จากความเข้าใจการทำงานของเซลล์ประสาทของสิ่งมีชีวิตมาประยุกต์ใช้

แบบจำลองพื้นฐานของเซลล์ประสาทเทียมถูกนำเสนอโดยแม็คคัลลอชและพิทส์ ตั้งแต่ปี พ.ศ. 2486 โดยมีการทำงานคร่าวๆ แบบเซลล์ประสาทของสิ่งมีชีวิตคือ ทำหน้าที่รวมสัญญาณที่เข้ามายังเซลล์ประสาทเทียม ซึ่งเสมือนว่าเป็นสัญญาณที่เข้ามาตามแดนไดรท์ของเซลล์ประสาทของสิ่งมีชีวิต แล้วยิงสัญญาณกระตุ้นออกไป หากผลรวมของสัญญาณเข้านั้นมีค่าสูงเกินค่าระดับ (threshold) ซึ่งก็เสมือนการยิงสัญญาณไฟฟ้าออกทางแอกซอนจากเซลล์ประสาทของสิ่งมีชีวิตนั่นเอง

อย่างไรก็ตาม สิ่งที่สำคัญในการจำลองเซลล์ประสาทคือ การจำลองไซแนปส์ทั้งหลายในโครงข่ายประสาทเทียม ซึ่งเปรียบเสมือนแหล่งสะสมความรู้ของสมอง การจำลองไซแนปส์นั้น ใช้หลักการที่ว่า แต่ละไซแนปส์ทำหน้าที่เป็นตัวปรับเปลี่ยนสภาพสัญญาณไฟฟ้าที่ส่งมาจากเซลล์ประสาทตัวอื่นๆ ก่อนส่งสัญญาณนั้นผ่านเดนไดรท์เข้าสู่ตัวเซลล์ประสาท และการปรับเปลี่ยนสัญญาณดังกล่าว จะขึ้นอยู่กับความเหนียวแน่นของการเชื่อมต่อบริเวณรอยต่อไซแนปส์ โดยความ

แข็งแกร่งนี้จะเปลี่ยนไปตามความรู้ที่สมองได้เรียนเข้าไป แม็คคัลลอชและพิทส์เสนอให้ใช้ตัวแปรตัวหนึ่งเรียกว่า "ค่าน้ำหนัก" (weight) ในการจำลองไซแนปส์ หากค่าน้ำหนักนี้มีขนาดใหญ่ก็จะหมายความว่า ความเหนียวแน่นของรอยต่อไซแนปส์มีค่าสูง นั่นคือส่งผ่านสัญญาณได้มาก หากค่าน้ำหนักนี้มีขนาดเล็กก็หมายความว่าสัญญาณจะส่งผ่านรอยต่อไซแนปส์ได้น้อย นอกจากนั้น ความเป็นบวกหรือลบของค่าน้ำหนักก็มีความหมายเช่นกัน หากค่าน้ำหนักมีค่าเป็นบวกจะหมายความว่าสัญญาณที่วิ่งผ่านรอยต่อไซแนปส์เข้าสู่เซลล์ประสาทเทียมจะเป็นสัญญาณกระตุ้น แต่หากค่าน้ำหนักมีค่าเป็นลบ จะหมายความว่าสัญญาณที่ผ่านรอยต่อไซแนปส์เข้าสู่เซลล์ประสาทเทียมจะมีผลเป็นสัญญาณกด

2.2 การเรียนรู้ของโครงข่ายประสาทเทียม

ผลการวิจัยทางพฤติกรรมศาสตร์พบว่า การเรียนรู้ของสิ่งมีชีวิตชนิดต่างๆ นั้นมีกระบวนการแตกต่างกันไปหลายๆ แบบ แต่ละแบบก็อาจเหมาะสมกับแต่ละเผ่าพันธุ์ของสิ่งมีชีวิตนั้นๆ ในสาขาโครงข่ายประสาทเทียมนั้น แนวคิดของกระบวนการเรียนรู้จะประยุกต์มาจากผลการศึกษาทางพฤติกรรมศาสตร์ อาจกล่าวโดยทั่วไปได้ว่าการเรียนรู้คือ กระบวนการซึ่งระบบประสาทปรับตัวเองไปตามสิ่งเร้า จนกระทั่งสามารถให้ผลตอบได้ตามต้องการ โดยใช้การปรับตัวแปรที่ควบคุมสภาพของตัวระบบเอง

การเรียนรู้ยังสามารถถูกมองได้ว่า เป็นกระบวนการจัดชนิดของสิ่งเร้าทั้งหลายที่เข้ามาอย่างต่อเนื่องด้วย นั่นคือ เมื่อได้รับสิ่งเร้า หากระบบประสาทรู้จักสิ่งเร้านั้น ก็จะให้ผลตอบได้ตามที่เคยเข้าใจไว้ แต่หากไม่รู้จัก ก็พยายามปรับความเข้าใจในการจัดชนิดขึ้นใหม่ ในทางปฏิบัตินั้น ระบบประสาทของสิ่งมีชีวิตจะปรับความเหนียวแน่นของการเชื่อมต่อที่ไซแนปส์ จนสร้างผลตอบต่อสิ่งเร้าได้ตามที่ต้องการ สถานะที่กระบวนการของการเรียนรู้ก็จะสิ้นสุดลง เป็นสถานะที่ถือว่าระบบประสาทได้รับความรู้ไปแล้ว

คำจำกัดความของกระบวนการเรียนรู้หมายถึงขั้นตอนต่อไปนี้

<u>ขั้นที่หนึ่ง</u> โครงข่ายประสาทถูกกระตุ้นด้วยสิ่งแวดล้อม

<u>ขั้นที่สอง</u> โครงข่ายประสาทเกิดการเปลี่ยนแปลง อันเป็นผลมาจากการกระตุ้นดังกล่าว

<u>ขั้นที่สาม</u> โครงข่ายประสาทตอบสนองต่อสิ่งแวดล้อมในแนวทางใหม่ อันเป็นผลมาจากการเปลี่ยนแปลงที่เกิดขึ้นในโครงสร้างภายในโครงข่าย

เมื่อพิจารณาไปที่เฉพาะบริเวณหนึ่งๆ ของโครงข่ายประสาท จะพบว่า การเชื่อมต่อของเซลล์ประสาทที่บริเวณต่างๆ นั้นสามารถมีรูปแบบที่แตกต่างกันได้หลายๆ แบบ และกระบวนการเรียนรู้ของแต่ละบริเวณก็ไม่เหมือนกันด้วย ในทำนองเดียวกัน เทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมก็แตกต่างกันไปสำหรับแต่ละชนิดของโครงข่าย

รูปแบบการเรียนรู้แบบมีผู้สอนเริ่มด้วยการส่งสิ่งเร้าที่ใช้ในการสอนเข้าไปเป็นอินพุต (Input) ในโครงข่ายประสาทเทียม เพื่อให้โครงข่ายประสาทเทียมสร้างผลตอบออกมาเป็นเอาท์พุท (Output) ซึ่งผลตอบจะเป็นอย่างไร ก็ขึ้นอยู่กับสภาวะในตอนที่เริ่มเรียนรู้ของโครงข่ายประสาทเทียม ผลตอบดังกล่าวจะถูกนำมาเปรียบเทียบกับผลตอบเป้าหมาย (target response) ซึ่งผู้สอน (teacher) จะเป็นผู้สร้างขึ้น หากผลตอบทั้งสองมีความแตกต่างกัน นั่นคือ มีความคลาดเคลื่อน (error) เกิดขึ้น ความคลาดเคลื่อนดังกล่าวจะถูกนำไปคำนวณการปรับแต่งค่าน้ำหนักต่างๆ ในโครงข่ายประสาทเทียม เพื่อลดความคลาดเคลื่อนลงให้เหลือน้อยที่สุด การปรับแต่งค่าน้ำหนักโดยพิจารณาจากความคลาดเคลื่อนนี้ จะขึ้นอยู่กับกฎการเรียนรู้หรือขั้นตอนการคำนวณซึ่งเรียกว่า "อัลกอริทึม" (Algorithm) ที่แตกต่างกัน โดยแต่ละอัลกอริทึมจะมีคุณลักษณะและสมรรถนะแตกต่างกัน อย่างไรก็ตาม วิธีการส่วนใหญ่ของการเรียนรู้แบบมีผู้สอนนี้จะดัดแปลงมาจากวิธีการทางคณิตศาสตร์ในเรื่องของเทคนิคการหาค่าเหมาะสม (optimization technique) นั่นเอง

เนื่องจากเจตนาของมนุษย์ในการพัฒนาเครื่องมือขึ้นมาใช้งานนั้น จะอิงอยู่กับการที่มนุษย์ต้องการสั่งการและควบคุมเครื่องมือนั้นๆ ให้ทำงานได้ตามต้องการ จึงทำให้โครงข่ายประสาทเทียมประเภทที่ใช้การเรียนรู้แบบมีผู้สอนได้รับความนิยมในการนำไปประยุกต์ใช้มากที่สุด เนื่องจากเป็นแบบที่สามารถควบคุมได้ การสั่งการโครงข่ายประสาทเทียมจะเป็นไปโดยทางอ้อมในลักษณะของการฝึกสอนโครงข่ายประสาทเทียม โดยการสร้างข้อมูลตัวอย่าง (รวมทั้งค่าเป้าหมาย) ที่จะให้โครงข่ายเรียนรู้ เมื่อโครงข่ายประสาทเทียมเรียนรู้ข้อมูลตัวอย่างได้ถูกต้องหมดแล้ว ความรู้ที่โครงข่ายประสาทเทียมได้เก็บไว้ในลักษณะของค่าน้ำหนักต่างๆ จะเป็นสิ่งที่ถูกนำไปใช้งานจริง เพื่อสร้างผลตอบต่อข้อมูลใหม่ๆ ที่โครงข่ายไม่เคยเห็นมาก่อน ดังนั้น สำหรับโครงข่ายประสาทเทียมหนึ่งๆ และวิธีการเรียนรู้แบบมีผู้สอนวิธีการหนึ่งๆ นั้น ความรู้ของโครงข่ายประสาทเทียมจะสามารถนำไปใช้งานจริงได้เพียงใด ก็ขึ้นอยู่กับคุณภาพของข้อมูลตัวอย่างที่นำใช้สอนนั้น อาจจะกล่าวได้ว่า หากข้อมูลตัวอย่างมีจำนวนมากพอ โครงข่ายประสาทเทียมก็จะสามารถสร้างความรู้ได้อย่างถูกต้อง อย่างไรก็ตาม ด้วยระดับความเจริญก้าวหน้าด้านโครงข่ายประสาทเทียมในปัจจุบันเทคนิคการวิเคราะห์ปัญหาเพื่อสร้างข้อมูลตัวอย่าง สำหรับการเรียนรู้ที่มีประสิทธิภาพนั้นยังคงเป็นงานวิจัยที่ต้องมีการค้นคว้ากันต่อไป

การเรียนรู้แบบไม่มีผู้สอนนั้น ไม่จำเป็นต้องมีค่าเป้าหมายของแต่ละข้อมูลตัวอย่าง ในระหว่างการเรียนรู้ โครงข่ายประสาทเทียมจะได้รับข้อมูลกระตุ้นในรูปแบบต่างๆ และจะทำการจัดกลุ่มรูปแบบต่างๆ เหล่านั้นเองตามต้องการ ผลตอบของโครงข่ายประสาทเทียมที่ใช้การเรียนรู้แบบไม่มีผู้สอนนี้ จะเป็นการระบุกลุ่มของข้อมูลที่ใส่เข้าไป โดยจะอิงกับวิธีการจัดกลุ่มซึ่งได้เรียนรู้จากข้อมูลที่โครงข่ายเคยพบมา

ตัวอย่างการเรียนรู้แบบนี้ในมนุษย์คือ การให้เด็กเล็กๆ จัดเก็บสิ่งของไว้บนชั้นวางของให้เป็นระเบียบเรียบร้อย สมมุติว่า เด็กคนหนึ่งเลือกเก็บหนังสือต่างๆ ไว้ที่ชั้นบน เก็บตุ๊กตาไว้ที่ชั้นล่าง และเก็บของเล่นอื่นๆ ไว้ที่ชั้นกลางๆ หลังจากนั้น หากเด็กคนนั้นซื้อตุ๊กตามาใหม่ ก็จะนำไปเก็บไว้ที่ชั้นล่าง เป็นต้น

แม้ว่าการเรียนรู้แบบไม่มีผู้สอนนี้จะไม่ต้องการผู้สอน แต่ก็ต้องการแนวทางในการจัดกลุ่ม เช่น การจัดกลุ่มอาจจะจัดตามรูปทรง สีหรือวิธีการใช้งานของวัตถุต่างๆ ที่จะนำมาจัดเป็นต้น ดังนั้น หากไม่มีการให้แนวทางที่ชัดเจนว่าการจัดกลุ่มควรเป็นไปตามคุณลักษณะใด การจัดกลุ่มอาจไม่ประสบความสำเร็จในแง่การนำมาใช้งานจริงก็ได้ ตัวอย่างเช่น การให้เด็กจัดของไว้บนชั้นวางของนั้น เด็กอาจจะจัดตามใจชอบและไม่เป็นหมวดหมู่ ทำให้ไม่สะดวกต่อการนำสิ่งของใหม่ๆ เข้าไปเก็บรวมด้วยก็ได้ การใช้งานโครงข่ายประสาทเทียมที่ใช้การเรียนรู้แบบนี้ จึงมักต้องมีการดำเนินการปรับแต่งข้อมูล เพื่อให้เกิดการเน้นสภาพของคุณลักษณะสำคัญ ที่ต้องการนำมาเป็นแนวทางในการจัดกลุ่มให้เด่นชัดขึ้นหรืออาจเป็นการปรับกฎการเรียนรู้ เพื่อให้เน้นไปที่คุณลักษณะที่ต้องการก็ได้

เนื่องจากขั้นตอนการเรียนรู้แบบไม่มีผู้สอนนี้จะมีการระบุกลุ่มของข้อมูลตัวอย่างก่อน เมื่อตัดสินใจได้แล้วว่า ข้อมูลใหม่มีลักษณะที่ควรจะจัดรวมเข้ากลุ่มใด (หรืออาจถือเป็นกลุ่มใหม่ก็ได้ในกรณีที่เห็นว่าไม่ควรจัดเข้ากลุ่มใดเลย) หลังจากนั้น จึงมีการปรับคุณลักษณะของกลุ่ม โดยการนำลักษณะของข้อมูลใหม่นี้มาช่วยกำหนดแนวทางการจัดด้วย ในการตัดสินว่าข้อมูลใหม่นี้ควรจัดรวมเข้ากลุ่มใด ในโครงข่ายประสาทเทียมบางชนิดอาจจะใช้วิธีการแข่งขันกันของกลุ่มต่างๆ ว่ากลุ่มใดควรได้ข้อมูลดังกล่าวไป การเรียนรู้ในลักษณะนี้จะถูกเรียกว่า การเรียนรู้แบบแข่งขันกัน (Competitive Learning)

ในอีกแง่มุมหนึ่ง นับจากจุดเริ่มต้นของการเรียนรู้ซึ่งไม่มีการจัดกลุ่มข้อมูลในแบบใดๆ เลย จนถึงเวลาที่การจัดเสร็จสิ้นแล้ว จะพบว่า การจัดกลุ่มข้อมูลเกิดขึ้นตามคุณลักษณะบางอย่างของข้อมูลตัวอย่าง ซึ่งการจัดกลุ่มนี้เกิดจากการที่โครงข่ายประสาทเทียมประเมินข้อมูลต่างๆ ที่ถูกป้อนเข้าไปในระหว่างการเรียนรู้ จนสร้างเป็นวิธีการจัดกลุ่มขึ้นมาได้ ดังนั้น การเรียนรู้ในลักษณะดังกล่าวจึงถูกเรียกว่า การเรียนรู้แบบจัดตัวเอง (Self-organizing) ด้วย

2.3 สถาปัตยกรรมโครงข่ายประสาทเทียม

Neural Network มี 3 แบบคือ

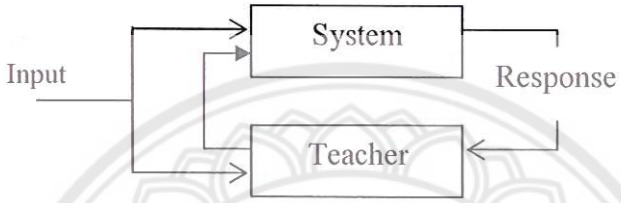
- Auto Associator เป็น Cell ประสาทที่ใช้จดจำว่าตัวไหนเข้ามาจะให้ผลลัพธ์เป็นตัวนั้น เช่น (JIM ให้ผลลัพธ์ JIM) ใช้ในกรณีที่ Input มีการผิดเพี้ยนไป Neural Cell จะหาคำตอบที่ถูกต้องให้ เช่น ใช้ในการแปลจาก ลายมือคนเป็น Text File

- Hetero Association เข้าตัวหนึ่งไปออกอีกตัวหนึ่ง ซึ่งจะนำมาใช้ในโปรแกรมนี้ ตัวอย่างเช่น JIM ออกผลลัพธ์เป็น Smith ซึ่งผลลัพธ์จะออกอะไรเราจะเป็นผู้กำหนด

- Classifier เป็น Hetero Association อย่างหนึ่ง และจะบอกถึงประเภทของ Input เช่น JIM ให้ผลเป็น Male

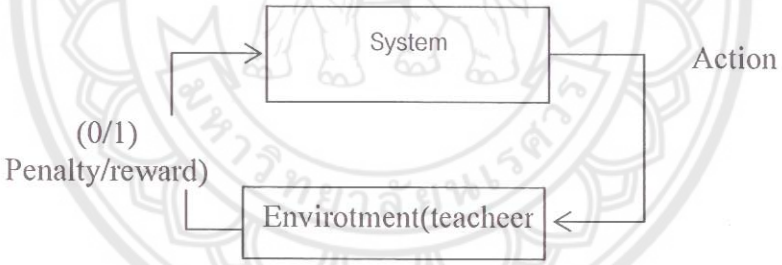
การที่ Matrix M มีการปรับค่าเพื่อให้การที่ นำ Vector มาคูณ จะได้คำตอบที่ต้องการเรียกว่า การเรียนรู้ ประเภทของการเรียนรู้มี 3 แบบคือ

1. Classical Supervised learning



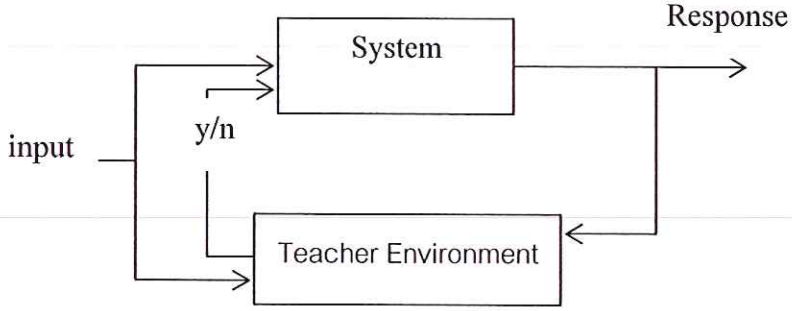
รูปที่ 2.1 ภาพแสดงการเรียนรู้แบบ Classical Supervised learning

2. Learning Automata (adaptive control Psychology)



รูปที่ 2.2 ภาพแสดงการเรียนรู้แบบ Learning Automata (adeptive control Psychology)

3. Combined รวม 2 ระบบเข้าด้วยกัน



รูปที่ 2.3 ภาพแสดงการเรียนรู้แบบ Combined รวม 2 ระบบเข้าด้วยกัน

การเชื่อมโยงเซลล์ประสาทเทียมจำนวนหนึ่งเข้าด้วยกันเป็นโครงข่ายประสาทเทียมนั้น สามารถเชื่อมโยงแบบใดก็ได้อย่างไม่ขอบเขตจำกัด อย่างไรก็ตาม ในทางปฏิบัติแล้ว เทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมมักจะถูกออกแบบมาให้ใช้งานได้กับสถาปัตยกรรมโครงข่ายประสาทเทียมที่มีลักษณะเฉพาะเท่านั้น สถาปัตยกรรมโครงข่ายประสาทเทียมที่พบทั่วไปจะมีลักษณะหลักๆ คือ มีการจัดเซลล์ประสาทเทียมเป็นชั้นๆ (layer) ชั้นที่รับข้อมูลเข้าเรียกว่า ชั้นอินพุต (input layer) ชั้นที่ผลิตผลตอบของโครงข่ายเรียกว่า ชั้นเอาท์พุท (output layer) ส่วนชั้นอื่นๆ ที่มีส่วนในการช่วยทำการประมวลผลอยู่ภายในเรียกว่า ชั้นซ่อน (hidden layer) ในโครงข่ายประสาทเทียมอาจมีชั้นซ่อนได้หลายชั้น โครงสร้างพื้นฐานจะมีลักษณะเป็นการประกอบกันของรูปแบบดังต่อไปนี้

2.3.1 แบบป้อนไปข้างหน้า (feed forward) อาจจัดได้เป็นสองแบบย่อยคือ แบบมีชั้นของเซลล์ประสาทชั้นเดียว และแบบมีชั้นของเซลล์ประสาทหลายชั้น โดยปกติแล้ว การเชื่อมโยงจะถูกกำหนดขึ้นระหว่างชั้นที่ติดกัน โดยจะมีการเชื่อมโยงระหว่างเซลล์ประสาทเทียมทุกตัว จากชั้นหนึ่งๆ ไปยังเซลล์ประสาทเทียมทุกตัวในชั้นต่อไป ในบางสถาปัตยกรรมอาจมีการเชื่อมโยงข้ามชั้นก็ได้

2.3.2 แบบมีการป้อนไปเวียนกลับ (recurrent) ในสถาปัตยกรรมบางแบบ โครงข่ายประสาทเทียมอาจมีการเชื่อมโยงที่ถูกกำหนดขึ้นระหว่างเซลล์ประสาทเทียมในชั้นหนึ่งๆ ย้อนกลับไปยังชั้นอื่นๆ ก่อนหน้านั้น หรือแม้แต่ภายในชั้นเดียวกันเอง

2.4 หลักการทำงานของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม หรือ Neural Network เป็นกระบวนการจัดการข้อมูลที่อาศัยการทำงานที่เลียนแบบมาจากสมองของมนุษย์ คือจะมีการจดจำและสามารถที่จะเรียนรู้ในสิ่งใหม่ๆได้ แต่ทว่าการทำงานของสมองจริงเป็นการทำงานที่อาศัยเซลล์เป็นจำนวนหลายล้านเซลล์ซึ่งการที่มนุษย์จะสามารถหาอุปกรณ์จำนวนมากๆ ที่ทำงานร่วมกันได้ดีดังเช่นสมองของมนุษย์นั้นเป็นการยาก ดังนั้นแทนที่การเลียนแบบจะเป็นในรูปของส่วนประกอบก็เป็นการเลียนแบบในรูปของการทำงานแทน และแทนที่จะอาศัยอุปกรณ์หรือ hardware เป็นจำนวนมากแล้ว ก็หันมาใช้การจำลองการทำงานโดยสมการทางคณิตศาสตร์แทน

รูปแบบของ Neural Network ที่มีการจำลองโดยกระบวนการทางคณิตศาสตร์จะอยู่บนเงื่อนไขดังนี้

1. ข้อมูลจะถูกประมวลผลบน simple element หลายๆอัน ที่เรียกว่า neuron หรือ node
2. ข้อมูลจะถูกส่งผ่านระหว่าง node ผ่านการเชื่อมโยงที่เรียกว่า link
3. แต่ละ link มีค่าแสดงความสัมพันธ์ (associated weight) ซึ่งโดยทั่วไปจะถูกนำไปคูณกับค่าของข้อมูลที่อยู่บน link นั้น

4. node จะใช้สมการที่เรียกว่า activated function ในการคำนวณหาผลลัพธ์ (output) ของ node นั้นและการจะบ่งบอกถึง Neural Network หนึ่งมีสิ่งที่เป็นลักษณะเฉพาะของ Network นั้นๆ ดังนี้

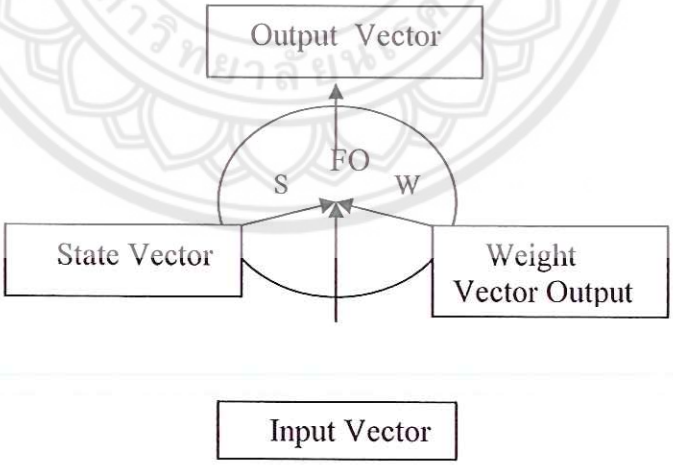
4.1 รูปแบบของการเชื่อมต่อของ node (หรือเรียกว่า architecture)

4.2 วิธีการที่ใช้ในการคำนวณค่า weight หรือ training, learning, algorithm

4.3 รูปแบบของ activated function

Neural Network เป็นการเชื่อมต่อกันด้วยรูปแบบต่างๆ โดยการทำงานอาจมีการต่อป้อนกลับ หรือเชื่อมต่อกันเพียงบางส่วน เมื่อมองย้อนกลับมาพิจารณาถึงสมองมนุษย์ เอาท์พุตของโหนดหนึ่งจะถูกนำไปเป็นอินพุตของโหนดอื่นๆ ซึ่งโหนดๆหนึ่งอาจรับอินพุตมาจากหลายๆ โหนด โดยอินพุตจากโหนดเดียวกันป้อนไปยังโหนดอื่นๆ อาจจะมีผลต่อโหนดอื่นๆ นั้นไม่เท่ากัน ซึ่งการที่เป็นเช่นนี้ก็เพราะว่าคุณสมบัติของเส้นทาง (link) นั้นไม่เท่ากัน โดยคุณสมบัตินี้สามารถนำมาจำลองใน Neural Network ได้ โดยค่าน้ำหนัก (weight value) กับเส้นทางแต่ละเส้น โดยอินพุตใดจะมีผลมากน้อยกับเอาท์พุตที่จะได้นั้นก็ขึ้นอยู่กับค่าน้ำหนักนี้

รูปแบบโครงสร้างทั่วไปของ Neural Network แต่ละโหนดจะต้องมีอย่างน้อย 1 อินพุต และ 1 เอาท์พุต และสิ่งที่เกิดขึ้นในแต่ละโหนด โดยกำหนดให้ f() เป็น activated function โดยปกติจะเขียนในรูปแบบ f (I,W,S) ซึ่งเป็นฟังก์ชั่นของ I (input), W (weight) และ S (state vector) ดังรูปที่แสดงไว้



General Node

รูปที่ 2.4 ภาพแสดงรายละเอียดส่วนประกอบโดยทั่วไปของโหนด

โครงข่ายจะทำการเก็บและจดจำข้อมูลโดยอาศัยการเปลี่ยนแปลงค่า weight ของโหนดในโครงข่าย ถ้าต้องการให้โครงข่ายมีการทำงานและจดจำได้ถูกต้องจำเป็นมากที่จะต้องกำหนดค่าของ weight ที่เหมาะสมให้กับทุกโหนดในโครงข่าย โดยขั้นตอนของการกำหนดค่า weight จะได้จากการคำนวณที่เรียกว่า training (การฝึกสอน) ซึ่งวิธีของการฝึกสอนก็มีอยู่มากมาย แต่ก็มีหลักการที่เหมือนกันคือ จะทำการให้ตัวอย่างแก่โครงข่ายก็จะทำการปรับค่า weight ของโหนดต่างๆเอง เพื่อให้ได้ผลที่ถูกต้องตามที่ต้องการจากตัวอย่างนั้นๆ

2.5 การแบ่งประเภทของโครงข่ายประสาทเทียม

2.5.1 แบ่งตามลักษณะของ input ที่เข้ามา สามารถแบ่งได้เป็น 2 ประเภท คือ

- Binary Input มีอินพุตเข้ามาเป็นไบนารี่ คือมีค่าการตัดสินใจได้ 2 แบบคือ 1 กับ 0 หรือค่าที่เป็นลักษณะของ bipolar คือ 1 กับ -1

- Continuous Valued Input รูปแบบนี้อินพุตที่เข้ามาจะมีค่ามากกว่า 2 ค่า โดยค่าเป็นในลักษณะที่ต่อเนื่อง เช่น เป็นค่าต่อเนื่องที่ค่าตั้งแต่ 0 ถึง 1 หรือ ค่าตั้งแต่ -1 ถึง 1 ซึ่งการที่อินพุตมีค่าต่อเนื่อง จึงทำให้มีการตัดสินใจมีมากกว่า 2 แบบ

2.5.2 แบ่งตามลักษณะการหาค่าน้ำหนัก (weight) หรือแบ่งตามวิธีการTraining สามารถแบ่งได้เป็น 3 ประเภท คือ

- Supervised training เป็นการเรียนรู้ของ network นั้นเป็นการเรียนรู้ระหว่าง input vector และ associated target output vector ค่าของน้ำหนักจะถูกเปลี่ยนไปตาม learning algorithm

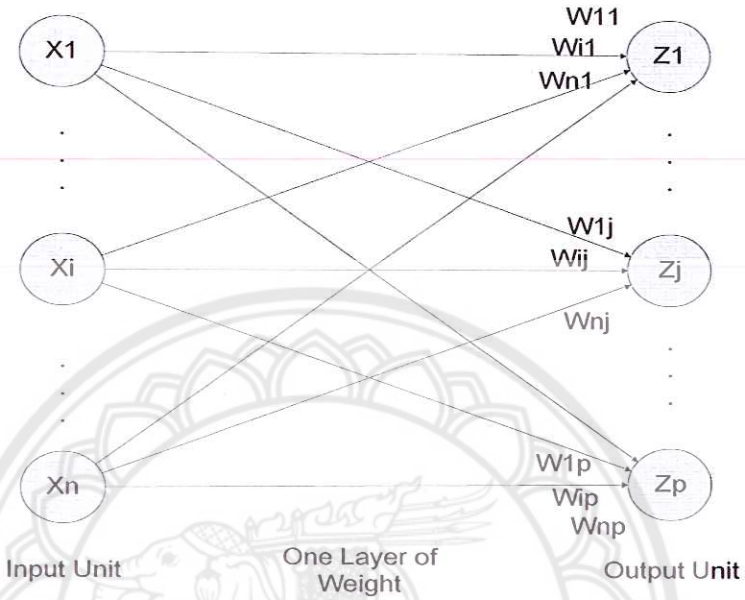
ในกรณีของ pattern recognition โดยที่ Neural Network นั้นเป็นการเรียนรู้ระหว่าง input และ output จะเรียกวิธีนี้ว่า "associative memory" และหากว่าค่าของ output มีค่าที่แตกต่างกับค่า input ที่ส่งเข้าไปจะเรียกว่า "heteroassciative memory"

- Unsupervised training ในการเรียนรู้แบบนี้โครงข่ายจะไม่มีการกำหนดค่าของ target output แต่จะมีการใส่ค่าของ input vector แล้วโครงข่ายจะทำงานในลักษณะของ self organization โดยที่โครงข่ายมีหน้าที่จัดกลุ่มของ input ที่มีลักษณะที่คล้ายกันเอง เพื่อใช้กำหนดกลุ่มของ output ที่เหมือนกัน (same output unit)

- Fixed-weight Net ในการออกแบบโครงข่ายบางครั้งจะมีกำหนดค่าของน้ำหนักลงไป เช่น Boltzman machine, Continuous Hopfield Net โดยที่ไม่มีขั้นตอนของการเรียนรู้ ซึ่งการกำหนดค่าของน้ำหนักแบบนนี้มักใช้กับปัญหาที่มีการแก้ไขที่มีขอบเขตจำกัด

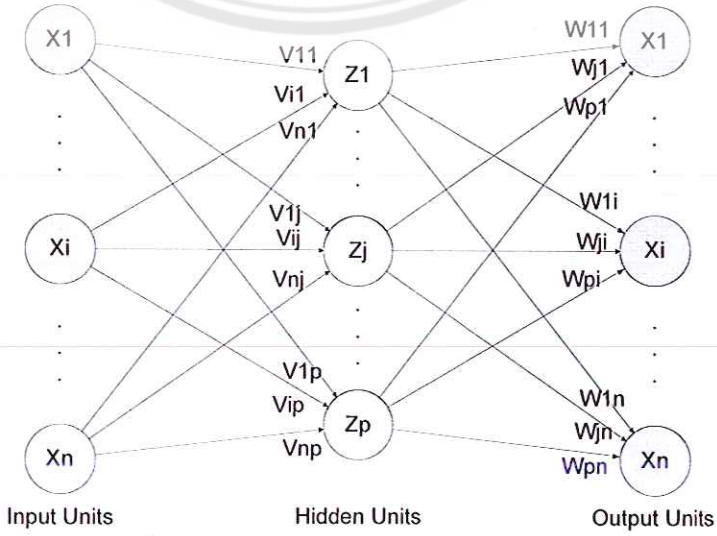
2.5.3 แบ่งตามรูปแบบของโครงข่าย สามารถแบ่งได้เป็น 3 ประเภท คือ

- Single Layer Net โครงข่ายแบบนี้จะมีเพียง 2 Layer เท่านั้น คือ input layer และ output layer โดยที่ input unit จะเชื่อมต่อไปยัง output unit ในการเชื่อมครบทุกโหนด (fully connected)



รูปที่ 2.5 ภาพแสดงรูปแบบโครง Single-Layer

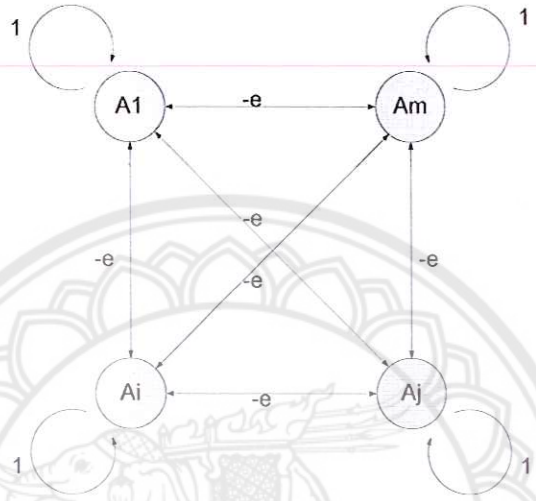
- Multi Layer Net โครงข่ายแบบนี้จะมีจำนวน layer มากกว่า 2 layer ที่เพิ่มขึ้นมาเรียกว่า Hidden layer โดยจะเป็น layer ที่อยู่ระหว่าง input layer และ output layer ซึ่งในแบบนี้จะสามารถแก้ไขปัญหาที่ซับซ้อนได้ดีกว่าแบบ single layer net



รูปที่ 2.6 ภาพแสดงรูปแบบโครง Multi-Layer

- Competitive layer โครงข่ายแบบนี้จะมีความแตกต่างไปจาก 2 แบบที่ได้กล่าวมาข้างต้นซึ่งทั้งแบบ single layer, multi layer นั้นการทำงานจะเป็นแบบ feed forward แต่ในแบบ competitive

Layer การทำงานจะเป็นแบบ recurrent คือ ทิศทางของข้อมูลสามารถวนกลับมาที่โหนดเดิมได้



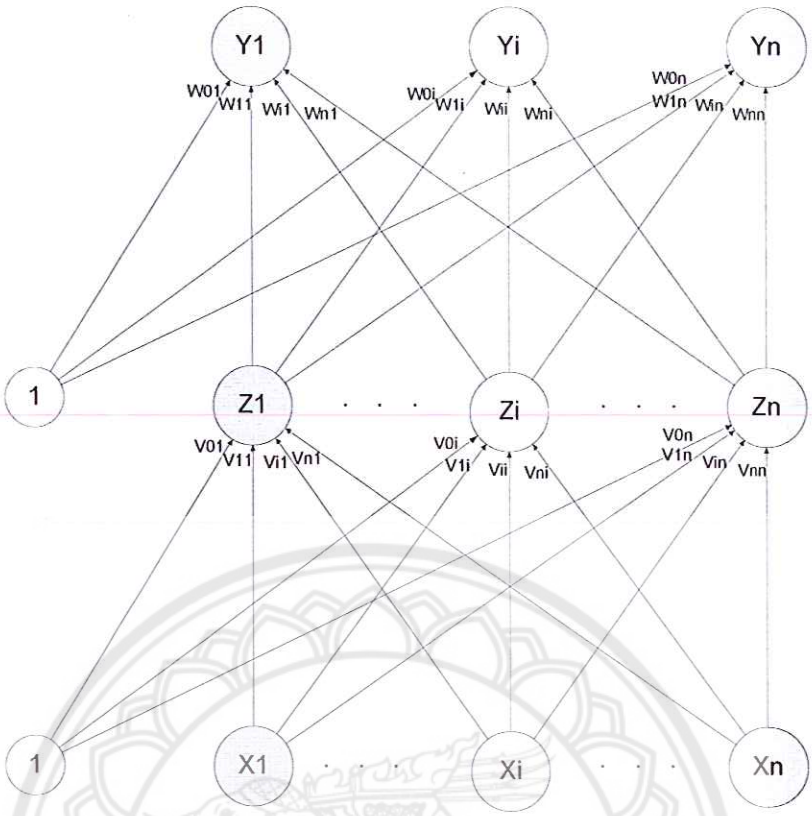
รูปที่ 2.7 ภาพแสดงรูปแบบโครง Competitive-Layer

รูปแบบของ Neural ที่ใช้ในโครงงาน Speech Recognition เป็น neural network แบบ Multi layer Perceptron using back error propagation ซึ่งมีการเรียนรู้แบบ Supervised

ในโครงข่ายแบบนี้ โครงข่ายจะถูกแบ่งเป็นชั้นๆ ที่เรียกว่า Layer ซึ่งแต่ละสามารถที่จะรับ input ได้ทีละหลายตัวและ output ของแต่ละชั้นจะเป็นไปเป็น input ของชั้นถัดไป โดยจะไม่มีการส่ง output ไปยังโหนดในชั้นก่อนหน้าหรือชั้นเดียวกันเลย layer ในโครงข่ายแบบนี้ต้องประกอบด้วย input layer ซึ่งไม่มีการคำนวณใดๆ และ output layer ระหว่าง 2 Layer นี้จะมี layer อื่นแทรกอยู่ที่เรียกว่า hidden layer ซึ่งโดยปกติมันจะมีอยู่หนึ่งหรือสองชั้น

โดยทั่วไปค่าของ input จะเป็นค่าต่อเนื่อง ซึ่งอาจเป็นค่าตั้งแต่ 0 ถึง 1 ที่เรียกว่าเป็นแบบ binary หรือ -1 ถึง 1 ที่เรียกว่าแบบ bipolar ซึ่งแต่ละโหนดในโครงข่ายจะสามารถรับค่า input ตามที่ค่าต่อเนื่องนี้ และค่าของ output ที่คำนวณได้ที่ละโหนดก็จะมีอยู่ระหว่างค่าต่อเนื่องนี้เช่นกัน

ค่าต่อเนื่องที่ใช้โดยส่วนใหญ่มักเป็นค่าต่อเนื่องที่เป็น bipolar มากกว่าเพราะมักจะใช้เวลาในการเรียนรู้น้อยกว่าในแบบค่า input ที่เป็นค่าต่อเนื่องที่เป็นแบบ binary



รูปที่ 2.8 ภาพแสดงรูปแบบโครงสร้างโดยทั่วไปของโครงข่ายประสาทเทียมแบบ Backpropagation

จากรูปที่ 2.8 แสดงลักษณะโครงสร้างของ Multilayer Neural Network ที่มี 1 hiddenlayer (โหนด Z) และ outputlayer (โหนด Y) และที่ส่วนของ hiddenlayer และ outputlayer จะมีค่าค่าหนึ่งที่จะใช้ในการคำนวณเรียกว่าค่า Bias ซึ่งค่านี้จะมีการทำงานที่คล้ายกับค่าน้ำหนัก (weight) โดยในการทำงานของบางโครงข่ายค่าของ Bias จะถูกกำหนดไว้ที่ค่าคงที่ค่าหนึ่ง ดังเช่นโครงข่ายที่แสดงดังภาพ จะเห็นได้ว่าค่า bias ของ outputlayer ซึ่งถูกกำหนดไว้ที่ค่าคงที่ค่าหนึ่ง ดังเช่นโครงข่ายที่แสดงดังภาพ จะเห็นได้ว่าค่า bias ของ outputlayer ซึ่งถูกกำหนดไว้โดย w_{0k} และค่า bias ของ hiddenlayer ซึ่งถูกำหนดไว้โดย v_{0j} จะมีค่าที่เป็น 1 ที่ทุกโหนดการทำงาน แต่ในการทำงานของโครงข่ายอื่นๆ ค่า bias อาจได้มาจากการคำนวณในระหว่างที่ทำการรู้อยู่ก็ได้

จากภาพการทำงานของโครงข่ายจะมีทิศทางการคำนวณไปตามทิศทางลูกศร หรือเรียกว่า feedforward phase operation และในช่วงการทำงานที่เป็นการคำนวณค่าผิดพลาดในระหว่างการเรียนรู้ทิศทางของการคำนวณจะกลับทิศทางของลูกศรที่แสดงดังภาพหรือที่เรียกว่า Backpropagation phase of learning

ในการทำงานของระบบโครงข่ายประสาทเทียมหรือ Neural network ของจำนวนของ hiddenlayer ที่ใช้ไม่จำเป็นว่าจะต้องได้ผลดีเสมอไป ในบางงานจำนวนของ hiddenlayer มากกว่าก็ได้

การทำงานของโครงข่ายประสาทเทียมนั้น เป็นการประมวลผลโดยการเลียนแบบการทำงานของสมองมนุษย์ ซึ่งได้อาศัยแบบจำลองเชิงคณิตศาสตร์ ในโครงงานนี้จะเป็นการนำทฤษฎีโครงข่ายประสาทเทียมนำมาใช้ในการประมวลผลคำซึ่งได้นำมาพัฒนาและทดลองในบทต่อไป

บทที่ 3
การพัฒนาโปรแกรมโดยใช้โครงข่ายประสาทเทียม

เนื้อหาในบทนี้กล่าวถึงขั้นตอนการดำเนินการในส่วนต่างๆ โดยจะเป็นรายละเอียดในส่วนของการพัฒนาโปรแกรมโดยใช้ทฤษฎีโครงข่ายประสาทเทียมในการประมวลผลคำเป็นหลัก

3.1 โค้ดการทำงานของโครงข่ายประสาทเทียม

3.1.1 การสร้าง Module โดยใช้ทฤษฎีโครงข่ายประสาทเทียม

โดยจะแบ่งเป็นฟังก์ชั่นการทำงานเป็น ส่วนย่อยๆ

```
Option Base 1
Option Explicit
Const e = 2.7183
```

- เป็นการประกาศเริ่มต้นให้กับภาษาวิชวลเบสิค คำสั่ง Option Base 1 เป็นการกำหนดให้นับ Array แรกเริ่มต้นที่ 1

```
Private Type Dendrite
Weight As Double
End Type
```

- สร้างชนิดข้อมูลเป็นชนิด Dendrite โดยโครงสร้าง Dendrite จะประกอบด้วย Weight เป็นข้อมูลชนิด Double

```
Private Type Neuron
Dendrites() As Dendrite
DendriteCount As Long
Bias As Double
Value As Double
Delta As Double
```

```
End Type
```

- สร้างชนิดข้อมูลเป็นชนิด Neuron โดยโครงสร้าง Neuron จะประกอบด้วย Dendrites เป็นชนิด Dendrite, Dendrite Count (จำนวน dendrite) เป็นตัวแปรชนิด Long และค่าตัวแปรชนิด Bias, Value, Delta เป็นชนิด Double

```
Private Type Layer
Neurons() As Neuron
NeuronCount As Long
End Type
```

- สร้างชนิดข้อมูลเป็นข้อมูลชนิด Layer โดยโครงสร้างแต่ละ Layer จะประกอบด้วย Neurons (Neuron หลายตัว) เป็นตัวแปรชนิด Neuron, NeuronCount (จำนวนNeuron) เป็นตัวแปรชนิด Long

```
Private Type NeuralNetwork
Layers() As Layer
LayerCount As Long
LearningRate As Double
End Type
```

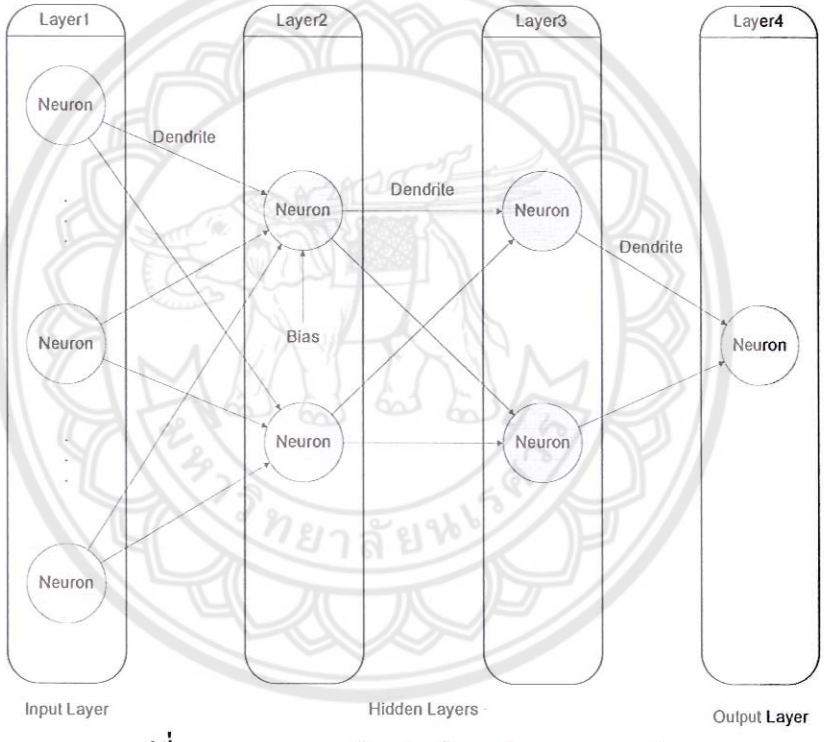
- โครงสร้างนี้เป็นโครงสร้างโดยรวมทั้งหมดโดยสร้างเป็นชนิด NeuralNetwork ซึ่งโครงสร้างภายในจะประกอบ Layers (Layer ในโครงข่ายประสาทเทียม), LayerCount (จำนวน Layer ในโครงข่ายประสาทเทียม), LearningRate (ค่า Learning rate สำหรับ Network)

```
Function CreateNet(LearningRate As Double, ArrayOfLayers As Variant) As Integer
Dim i, j, k As Integer
Network.LayerCount = UBound(ArrayOfLayers)
If Network.LayerCount < 2 Then
   CreateNet = 0
   Exit Function
```

```
End If
Network.LearningRate = LearningRate
ReDim Network.Layers(Network.LayerCount) As Layer
For i = 1 To UBound(ArrayOfLayers)
DoEvents
  Network.Layers(i).NeuronCount = ArrayOfLayers(i)
  ReDim Network.Layers(i).Neurons(Network.Layers(i).NeuronCount) As Neuron
  For j = 1 To ArrayOfLayers(i)
  DoEvents
    If i = UBound(ArrayOfLayers) Then
      Network.Layers(i).Neurons(j).Bias = GetRand
      Network.Layers(i).Neurons(j).DendriteCount = ArrayOfLayers(i - 1)
      ReDim
Network.Layers(i).Neurons(j).Dendrites(Network.Layers(i).Neurons(j).DendriteCount) As_
Dendrite
        For k = 1 To ArrayOfLayers(i - 1)
          DoEvents
          Network.Layers(i).Neurons(j).Dendrites(k).Weight = GetRand
        Next k
    ElseIf i = 1 Then
      DoEvents
    Else
      Network.Layers(i).Neurons(j).Bias = GetRand
      Network.Layers(i).Neurons(j).DendriteCount = ArrayOfLayers(i - 1)
      ReDim
Network.Layers(i).Neurons(j).Dendrites(Network.Layers(i).Neurons(j).DendriteCount) As_
Dendrite
        For k = 1 To ArrayOfLayers(i - 1)
          DoEvents
          Network.Layers(i).Neurons(j).Dendrites(k).Weight = GetRand
        Next k
    End If
```

```
   Next j
Next i
CreateNet = 1
End Function
```

- เป็นฟังก์ชั่นในส่วนของการสร้างโครงข่ายประสาทเทียมโดยจะสร้าง Layers ขึ้นมาภายใน Layer ประกอบด้วย Neurons แต่ละ Neuron จะมีค่า Bias และจะมีการเชื่อมต่อระหว่าง Neurons ด้วย Dendrite ซึ่งแต่ละ Dendrite จะมีค่า Weight อยู่โดยค่า Weight และค่า Bias ในตอนเริ่มต้นจะกำหนดให้เป็นค่าแบบสุ่มมีค่าอยู่ในช่วง -1 ถึง 1



รูปที่ 3.1 ภาพแสดงตัวอย่างโครงข่ายประสาทเทียม

จากรูปที่ 3.1 เป็นการยกตัวอย่างการทำงานของฟังก์ชั่น CreateNet โดยจากรูปทำการสร้าง NeuralNetwork ที่ประกอบด้วย Layers จำนวน 4 Layer โดยใน Layer ที่ 1 ซึ่งเป็น Input Layer จะมีจำนวน Neural ภายใน จำนวน 3 NeuralLayer ที่ 2 และ 3 ซึ่งเป็น Hidden Layers มี Neural ภายใน Layer จำนวน 2 Neural ส่วน Output Layer ซึ่งเป็นส่วนสุดท้าย มีจำนวน Neuron ภายใน จำนวน 1 Neuron โดยในทุกๆ Neuron จะมีค่า Bias และในทุกๆ Dendrite จะมีค่า Weight

```
Function Run(ArrayOfInputs As Variant) As Variant
Dim i, j, k As Integer
If UBound(ArrayOfInputs) <> Network.Layers(1).NeuronCount Then
   Run = 0
   Exit Function
End If
For i = 1 To Network.LayerCount - 1
DoEvents
   For j = 1 To Network.Layers(i).NeuronCount
   DoEvents
      If i = 1 Then
         Network.Layers(i).Neurons(j).Value = ArrayOfInputs(j)
      Else
         Network.Layers(i).Neurons(j).Value = 0
         For k = 1 To Network.Layers(i - 1).NeuronCount
            DoEvents
            Network.Layers(i).Neurons(j).Value = Network.Layers(i).Neurons(j).Value +_
Network.Layers(i - 1).Neurons(k).Value * Network.Layers(i).Neurons(j).Dendrites(k).Weight
         Next k
         Network.Layers(i).Neurons(j).Value = Sigmoid(Network.Layers(i).Neurons(j).Value +_
Network.Layers(i).Neurons(j).Bias)
      End If
   Next j
   Next i
   For i = 1 To Network.Layers(Network.LayerCount).NeuronCount
      For k = 1 To Network.Layers(Network.LayerCount).Neurons(i).DendriteCount_
Network.Layers(Network.LayerCount).Neurons(i).Value =_
Network.Layers(Network.LayerCount - 1).Neurons(i).Value *_
Network.Layers(Network.LayerCount).Neurons(i).Dendrites(k).Weight +_
Network.Layers(Network.LayerCount).Neurons(i).Bias
   Next k
   Next i
```

```
ReDim OutputResult(Network.Layers(Network.LayerCount).NeuronCount) As Double
For i = 1 To (Network.Layers(Network.LayerCount).NeuronCount)
   DoEvents
   OutputResult(i) = (Network.Layers(Network.LayerCount).Neurons(i).Value)
Next i
Run = OutputResult
End Function
```

- เป็นฟังก์ชั่นในส่วนของการทำงานภายในโครงข่ายประสาทเทียมโดยจะเขียนโปรแกรมเพื่อให้โครงข่ายประมาทเทียมได้เรียนรู้ฟังก์ชั่นของอินพุตกับเอาท์พุต โดยจะทำการกำหนด Layer แรกให้เป็น Input Layer และ Layer สุดท้ายเป็น Output Layer ส่วน Layer ที่อยู่ตรงกลางเป็นการทำงานของ Hidden Layer ฟังก์ชั่นที่ใช้ในการทำงานใน Hidden layer คือ log sigmoid และ linear โดยในตอนเริ่มต้นจะกำหนดค่าให้ Neural ที่อยู่ใน Hidden Layer และ Output Layer มีค่าเป็น 0 หลังจากนั้นจะเริ่มคำนวณค่าต่างๆ ให้แต่ละ Neuron โดยอ้างอิงจากสมการ 3.1

$$a^n = f(W^n p + b^n) \tag{3.1}$$

a = Neural Value

n = ตำแหน่ง Layer

f = ฟังก์ชั่นที่ใช้ในการทำงาน

W = ค่า Weight

p = Input สำหรับ Neural

b = ค่าBias

```
Function SupervisedTrain(inputdata As Variant, outputdata As Variant) As Integer
Dim i, j, k As Integer
If UBound(inputdata) <> Network.Layers(1).NeuronCount Then
   SupervisedTrain = 0
   Exit Function
End If
If UBound(outputdata) <> Network.Layers(Network.LayerCount).NeuronCount Then
SupervisedTrain = 0
   Exit Function
End If
Call Run(inputdata)
For i = 1 To Network.Layers(Network.LayerCount).NeuronCount
DoEvents
Network.Layers(Network.LayerCount).Neurons(i).Delta = (-2) * 1 *_
(outputdata(i) - Network.Layers(Network.LayerCount).Neurons(i).Value)
   For j = Network.LayerCount - 1 To 2 Step -1
   DoEvents
      For k = 1 To Network.Layers(j).NeuronCount
      DoEvents
         Network.Layers(j).Neurons(k).Delta = Network.Layers(j).Neurons(k).Value *_
         (1 - Network.Layers(j).Neurons(k).Value) * Network.Layers_
         (j + 1).Neurons(i).Dendrites(k).Weight * Network.Layers(j + 1).Neurons(i).Delta
      Next k
   Next j
Next i

For i = Network.LayerCount To 2 Step -1
DoEvents
   For j = 1 To Network.Layers(i).NeuronCount
   DoEvents
      Network.Layers(i).Neurons(j).Bias = Network.Layers(i).Neurons(j).Bias -_
(Network.LearningRate * 1 * Network.Layers(i).Neurons(j).Delta)
```

```
    For k = 1 To Network.Layers(i).Neurons(j).DendriteCount
    DoEvents

Network.Layers(i).Neurons(j).Dendrites(k).Weight =_
Network.Layers(i).Neurons(j).Dendrites(k).Weight -_
(Network.LearningRate * Network.Layers(i - 1).Neurons(k).Value *_
Network.Layers(i).Neurons(j).Delta)
    Next k
  Next j
Next i
SupervisedTrain = 1
End Function
```

- การทำงานของฟังก์ชั่น SupervisedTrain จะเป็นการทำงานในลักษณะ Back propagation ส่วนของการคำนวณความแตกต่างระหว่างเอาท์พุตที่ได้จากการคำนวณเปรียบเทียบกันเอาท์พุตจริง แล้วป้อนค่ากลับให้ Neural เรียนรู้โดยใช้สมการ

$$s^m = -2F^m(n^m)(t-a) \tag{3.2}$$

$$s^m = F^m(n^m)(W^{m+1})^t \cdot s^{m+1} \tag{3.3}$$

S = ค่า Delta

F = ฟังก์ชั่นการทำงาน

m = ตำแหน่ง Layer

t = ค่าเอาท์พุตจริง

a = เอาท์พุตที่ได้จากการคำนวณ

W = ค่า Weight

การคำนวณค่า Delta สำหรับ neural ที่อยู่ใน Output Layer คำนวณใช้สมการที่ 3.2 ส่วนค่า Delta สำหรับ neural ที่อยู่ใน Layer อื่นๆ คำนวณใช้สมการที่ 3.3 เมื่อได้ค่า Delta แล้วจะนำค่า Delta มาใช้ในการอัพเดทค่า Bias และค่า Weight ตามสมการที่ 3.4 และ3.5 ตามลำดับ

$$b^m(k+1) = b^m(k) - \alpha s^m \tag{3.4}$$

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \tag{3.5}$$

a = Neural Value

W = ค่า Weight

b = ค่าBias

S = ค่า Delta

α = ค่า Learning rate

m = ตำแหน่ง Layer

t = ค่าเอาท์พุตจริง

W = ค่า Weight

```
Function GetRand() As Double
Randomize
GetRand = 2 - (1 + Rnd + Rnd)
End Function
```

- เป็นฟังก์ชั่นในการ Random ค่าให้กับตัวแปรต่างๆ ให้มีค่าระหว่าง -1 ถึง 1

```
Private Function Sigmoid(Value As Double)
   Sigmoid = (1 / (1 + Exp(Value * -1)))
End Function
```

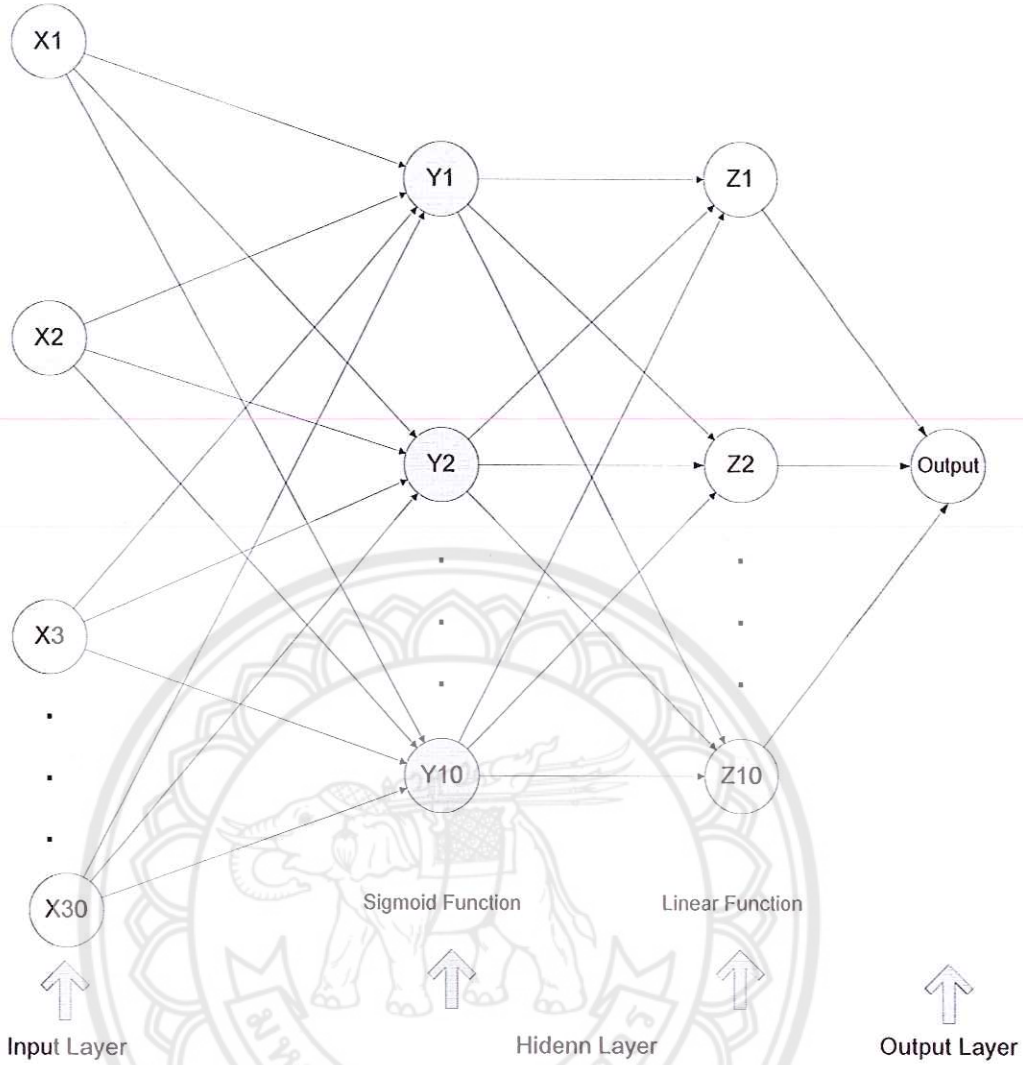
- ฟังก์ชั่นในการคำนวณค่า Sigmoid ฟังก์ชั่น

3.2 โค้ดการทำงานในส่วนของการประมวลผลคำ

```
Private Sub Form_Load()
    SetState False
    Call CreateNet(0.1, Array(30, 10, 10, 1))
    Set R = New SpInprocRecognizer
    Set c = R.CreateRecoContext
    Set G = c.CreateGrammar(16)
    Set V = New SpVoice
    Set V.Voice = V.GetVoices("gender=male").Item(0)
    G.DictationSetState SGDSActive

    Agent1.Characters.Load "Genie", DATAPATH
    Set Genie = Agent1.Characters("Genie")
    Genie.LanguageId = &H409
```

- จากโปรแกรมหลักจะเป็นการกำหนดค่าต่างๆ ที่ใช้ในการเรียกใช้งาน Speech SDK 5.1 หลังจากนั้นจะทำการสร้างโครงข่ายประสาทเทียมสำหรับประมวลผลคำ โดยมี จำนวน Layer เท่ากับ 4 Layer โดย Layer แรกจะประกอบด้วย 30 Neurons และ Layer ที่ 2 และ 3 จะประกอบด้วย 10 Neurons ส่วน Layer ที่ 4 จะมี 1 Neuron และกำหนดให้โครงข่ายประสาทเทียมนี้ มีค่า Learning rate = 0.1 ลักษณะโครงข่ายประสาทเทียมที่สร้างขึ้นแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 ภาพแสดงโครงข่ายประสาทเทียม

การทำงานโดยการเปลี่ยนเสียงพูดให้เก็บอยู่ในรูปของตัวเลขเพื่อนำมาใช้ให้โครงข่ายประสาทเทียมเรียนรู้แสดงได้จากโค๊ดด้านล่าง โดยจากตัวอย่างนี้จะทำการเปลี่ยนเสียงพูดแต่ละคำให้อยู่ในรูปของตัวเลขจำนวน 30 ตัวเลข

```
For Each e In Result.PhraseInfo.Elements
    Genie.Show
    A1 = e.AudioStreamOffset
    A2 = e.AudioSizeBytes
    X = Format(A1, "000000") & "" & Format(A2, "000000") & ""
    'Time data
    T1 = e.AudioTimeOffset
```

```
T2 = e.AudioSizeTime
X = X & Format(T1, "000000000") & "" & Format(T2, "000000000") & " "
For i = 1 To 30 Step 1
   it(i) = Mid(X, i, 1)
Next i
```

เมื่อได้ตัวเลขแล้วก็นำไปใช้ให้โครงข่ายประสาทเทียมเรียนรู้ โดยใช้คำสั่ง

Call Supervised Train (my Input(array of number), my Output(array of number))

เมื่อต้องการทดสอบการทำงาน ของโครงข่ายประสาทเทียม โดยการป้อนค่า Input ที่โครงข่ายประสาทเทียมไม่เคยมีการเรียนรู้เข้าไปจะใช้คำสั่ง Run (Array (array of number))

จากการทำงานของโปรแกรมนี้ได้ให้โครงข่ายประสาทเทียมเรียนรู้คำจำนวน 5 คำ เพื่อใช้ในการเปิดโปรแกรมคอมพิวเตอร์โดยคำที่ใช้มีดังนี้ คือ เพลง, วาด, เกมส์, โหลด, เนต

เพลง ใช้ในการเปิดโปรแกรม Win amp

วาด ใช้ในการเปิดโปรแกรม Paint

เกมส์ ใช้ในการเปิดโปรแกรมเกมส์ Solitaire

โหลด ใช้ในการเปิดโปรแกรม Flash get

เนต ใช้ในการเปิดโปรแกรม Internet Explorer

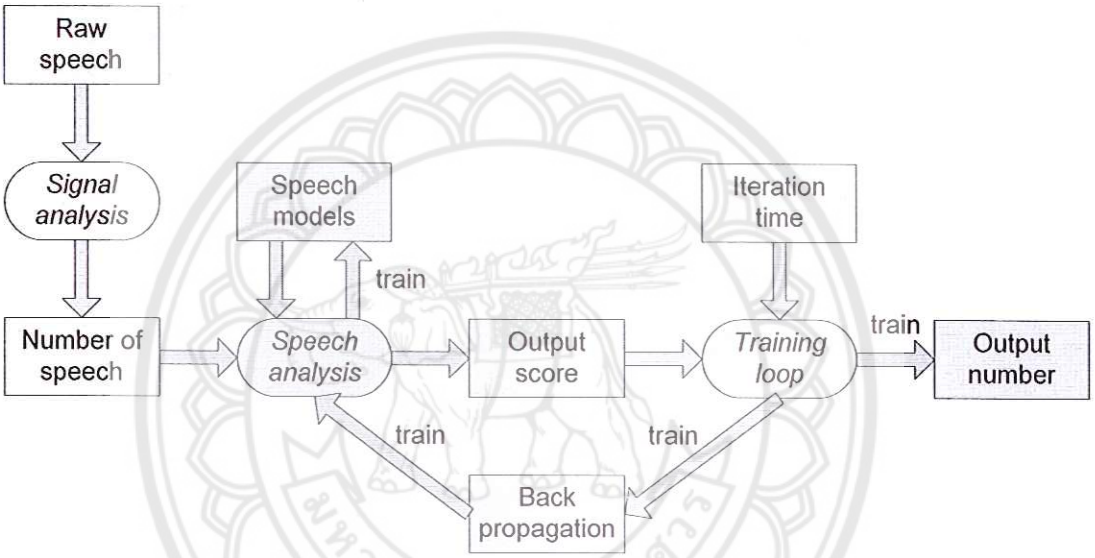
ขั้นตอนการดำเนินการดำเนินการในการพัฒนาโปรแกรมสามารถแบ่งการทำงานออกเป็นส่วนต่างๆ ดังนี้คือ ส่วนการรันเสียงพูด แล้วนำมาประมวลผลเป็นตัวเลข ส่วนของการเรียนรู้คำโดยใช้ทฤษฎีโครงข่ายประสาทเทียม และส่วนของโปรแกรมหลักที่ใช้ในการเชื่อมต่อการทำงานโดยการรับค่าอินพุตที่เป็นเสียงพูดแล้วนำเสียงพูดนั้นมาผ่านกระบวนการในการประมวลผลเพื่อให้ได้ค่าเอาท์พุตสำหรับเสียงพูดนั้นออกมา

บทที่ 4

ผลการทดลอง

จากการดำเนินการตามบทที่ 3 เนื้อหาในส่วนนี้จะแสดงผลของการทำงานจากขั้นตอนต่างๆ ดังที่ได้กล่าวมาแล้ว

4.1 ภาพรวมการทำงานของการประมวลผลคำ

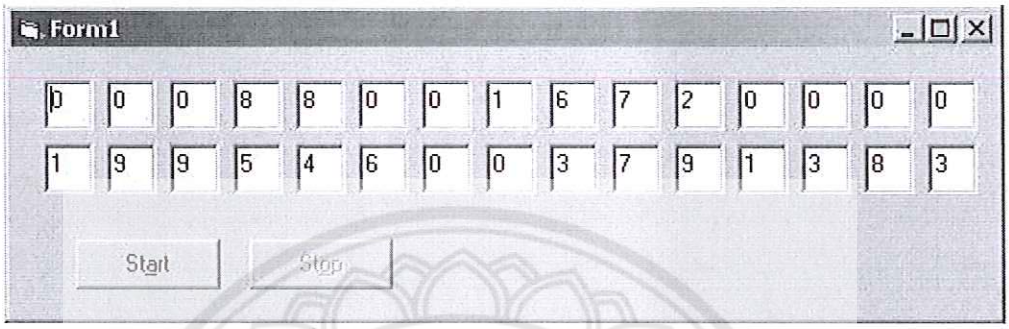


รูปที่ 4.1 ภาพแสดงภาพรวมการทำงานของการประมวลผลคำ [Joe Tebelskis 1995:372]

จากรูปที่ 4.1 จะแสดงภาพรวมในการทำงานหลังจากรับอินพุตที่เป็นเสียงพูดแล้วทำการประมวลผลสัญญาณเสียงที่ได้ยินออกมาในรูปของตัวเลข หลังจากนั้นนำไปเข้าสู่กระบวนการเรียนรู้ โดยการนำอินพุตที่รับมาไปวิเคราะห์เปรียบเทียบกับเสียงที่โครงข่ายประสาทเทียมรู้จักโดยทำการวนรอบการทำงานในการประมวลผลคำตามจำนวนที่กำหนด เมื่อทำการประมวลผลเสร็จจะได้ตัวเลขที่เป็นเอาท์พุตออกมา

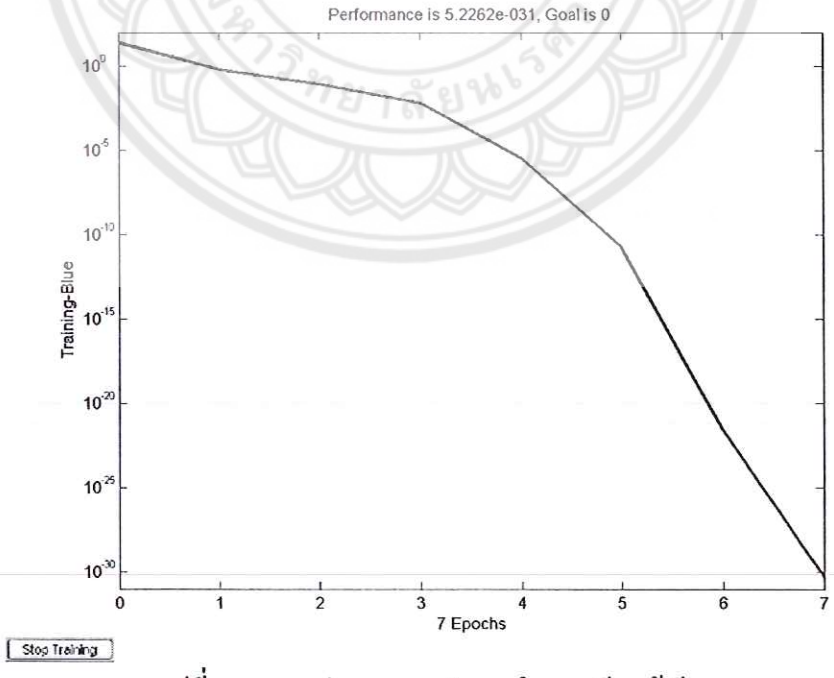
4.2 ผลที่ได้รับจากการประมวลผลคำ

ในส่วนของการเปลี่ยนค่าอินพุตที่เป็นเสียงพูดให้เป็นตัวเลข หลังจากใช้คำสั่งให้ Speech SDK5.1 ทำการเก็บค่าของคำพูดแต่ละคำให้อยู่ในรูปของตัวเลข ผลการทดลองที่ได้หลังจากทำการพูดคำหนึ่งคำจะออกมาในรูปของตัวเลขจำนวน 30 ตัว ตัวอย่างแสดงดังรูปที่ 4.2



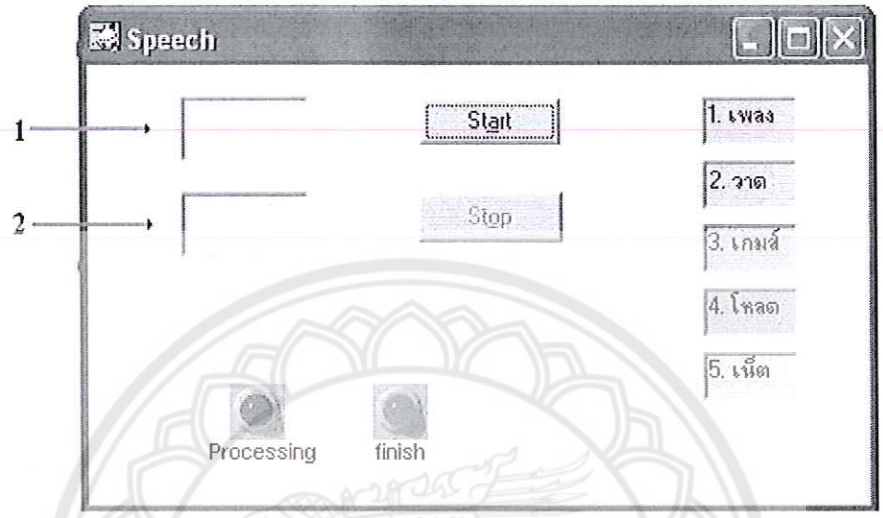
รูปที่ 4.2 ภาพแสดงตัวเลขที่ได้จากคำ 1 คำ

จากรูปที่ 4.2 แสดงตัวเลข 30 ตัวที่ได้จากการพูดคำ 1 คำผ่านไมโครโฟนซึ่งโปรแกรมจะนำตัวเลขดังกล่าว มาวิเคราะห์กับเสียงต้นแบบที่ได้ทำการบันทึกไว้โดยใช้ทฤษฎีโครงข่ายประสาทเทียม



รูปที่ 4.3 กราฟแสดงการทำงานในการเรียนรู้เสียงพูด

จากรูปที่ 4.3 เป็นการแสดงการทำงานในการเรียนรู้เสียงพูดโดยใช้โครงข่ายประสาทเทียม โดยจะสังเกตเห็นว่าค่าของกราฟจะลดลงเรื่อยจนเข้าใกล้ศูนย์ โดยถ้ากราฟลดลงจนถึงจุดที่น้อยที่สุดแล้วกราฟจะคงที่อยู่ที่ค่านั้น ซึ่งรูปแบบของโปรแกรมแสดงในรูปที่ 4.4

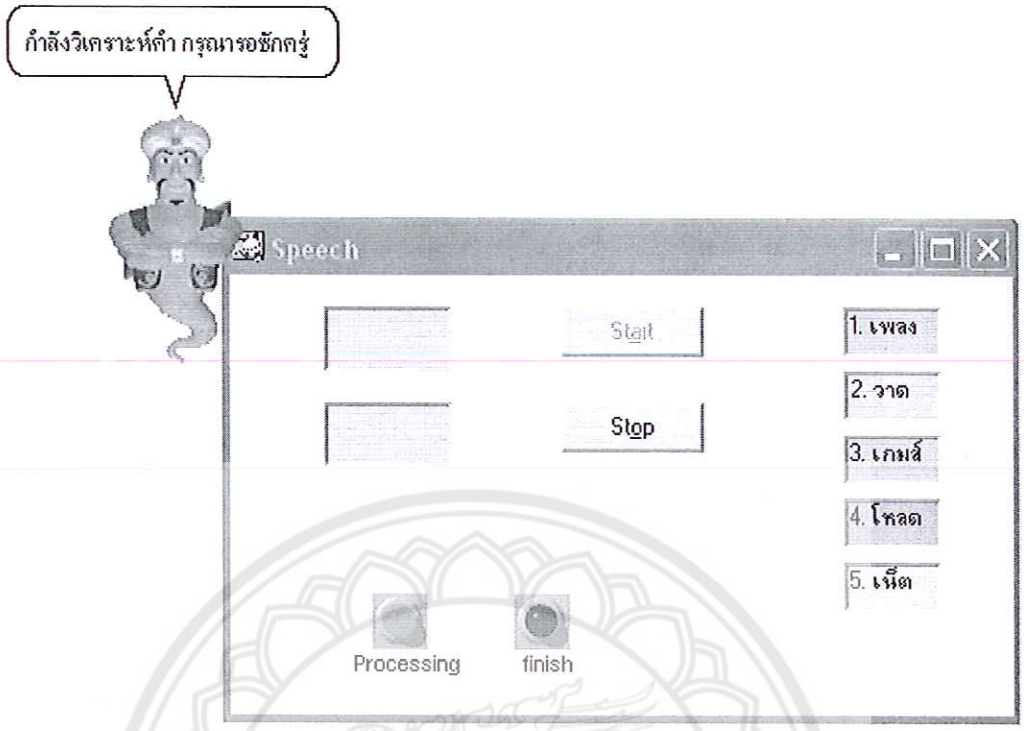


รูปที่ 4.4 ภาพแสดงโปรแกรมรับอินพุตจากไมโครโฟน

จากรูปที่ 4.4 แสดงภาพโปรแกรม ที่ใช้ในการรับอินพุตของเสียง เพื่อนำมาวิเคราะห์ด้วยโครงข่ายประสาทเทียม เมื่อกดปุ่ม Start เพื่อให้โปรแกรมเริ่มรับค่าอินพุตเข้ามาหลังจากนั้นก็ทำการกดปุ่ม Stop เพื่อให้โปรแกรมหยุดรับอินพุตค่าอื่นในระหว่างที่กำลังประมวลผลโปรแกรมจะทำการประมวลผลอินพุตที่รับเข้ามาโดยระหว่างที่โปรแกรมกำลังประมวลผลการทำงานหลอดไฟที่แสดงสถานะ การทำงานจะเป็นสีเขียว ดังแสดงในรูปที่ 4.4

โดยที่

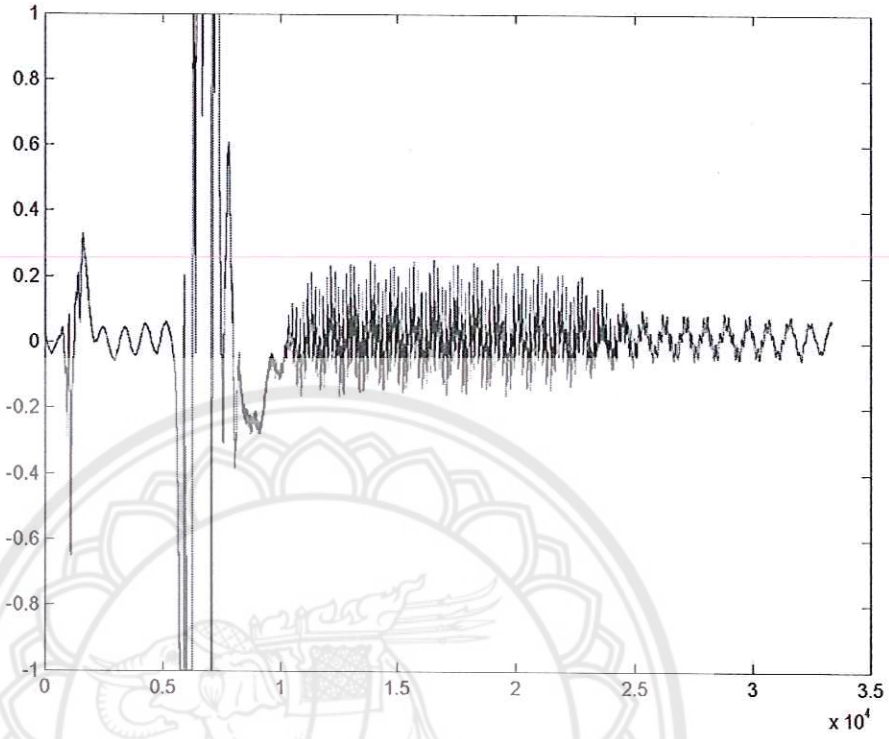
1. ค่าที่ได้จากการประมวลผลจริง
2. ค่าที่ได้จากการปัดเศษแล้ว



รูปที่ 4.5 ภาพแสดงโปรแกรมขณะทำการคำนวณ

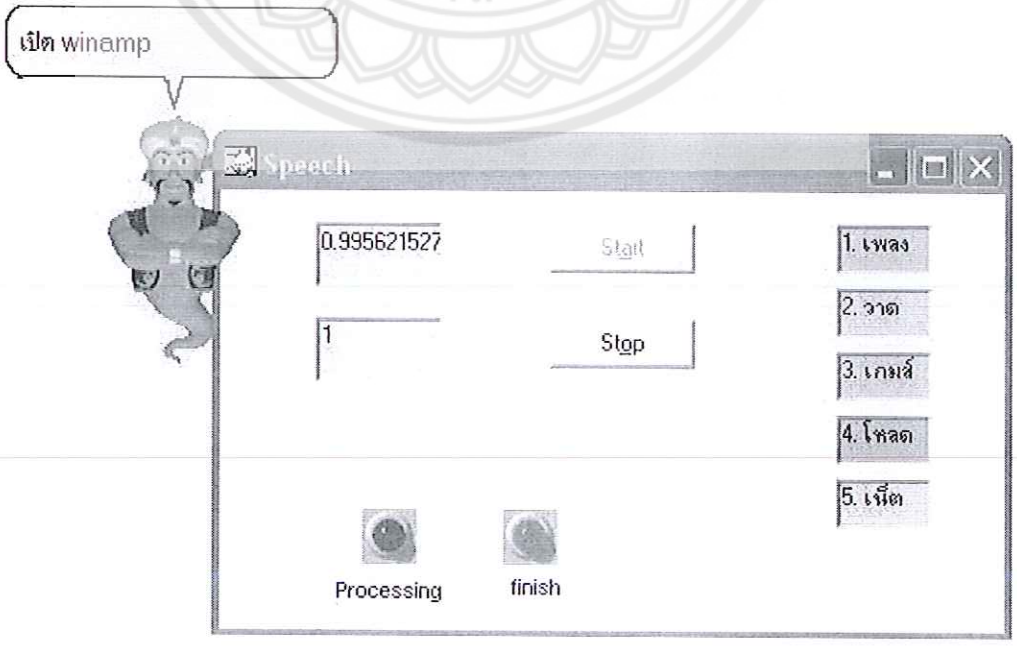
หลังจากที่ได้รับอินพุต และ ประมวลผลคำโดยใช้โครงข่ายประสาทเทียมแล้วไฟแสดงสถานะจะเปลี่ยนเป็นสีแดง โปรแกรมจะแสดงเอาท์พุต ที่ได้จากการคำนวณออกมาดังแสดงในรูปที่ 4.5 ในช่องที่ 1 โปรแกรมจะทำการเปรียบเทียบค่าโดยปัดค่าเลขที่ได้ให้ใกล้เคียงที่สุด และจะได้เอาท์พุตในการใช้งาน

ถ้าโปรแกรมได้รับอินพุตเป็นเสียงคำว่า “เพลง” โดยมีคลื่นเสียงแสดงดังรูปที่ 4.6



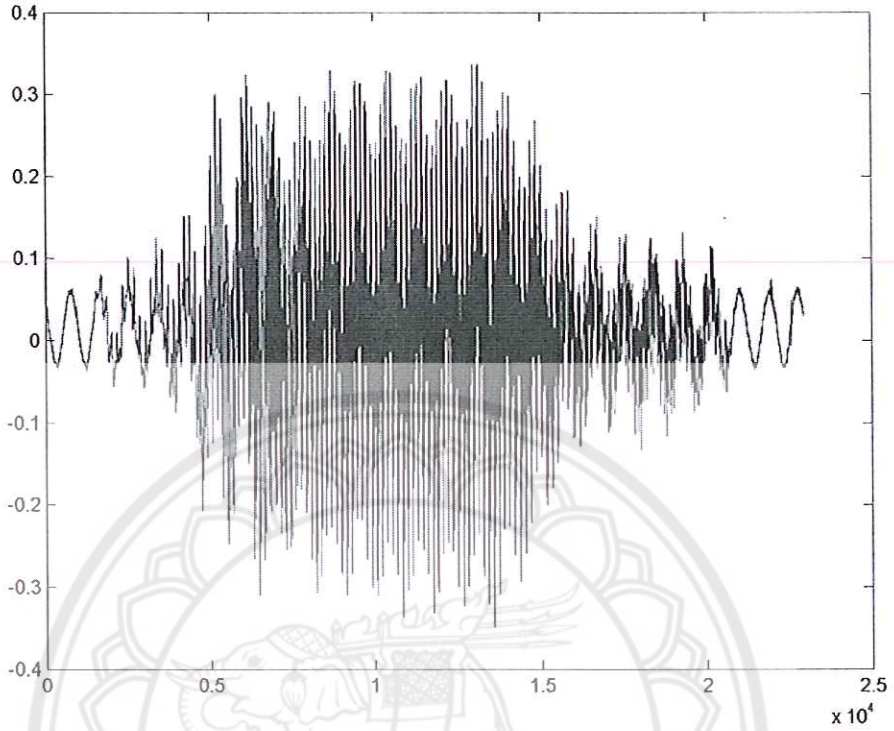
รูปที่ 4.6 ภาพแสดงคลื่นเสียงคำว่า “เพลง”

โปรแกรมจะทำการประมวลผลคำและได้เอาท์พุตในการประมวลผลเป็น**การสั่งงานเปิดโปรแกรม** Winamp ดังแสดงใน รูปที่ 4.7



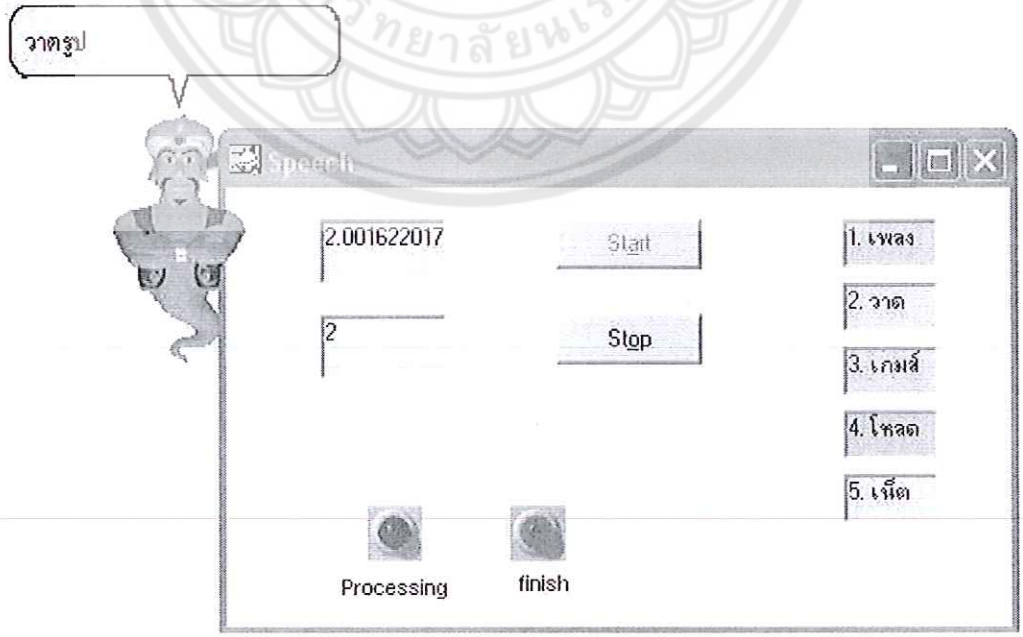
รูปที่ 4.7 ภาพแสดงโปรแกรมคำว่า “เพลง”

ถ้าโปรแกรมได้รับอินพุตเป็นเสียงคำว่า “วาด” โดยมีคลื่นเสียงแสดงดังรูปที่ 4.8



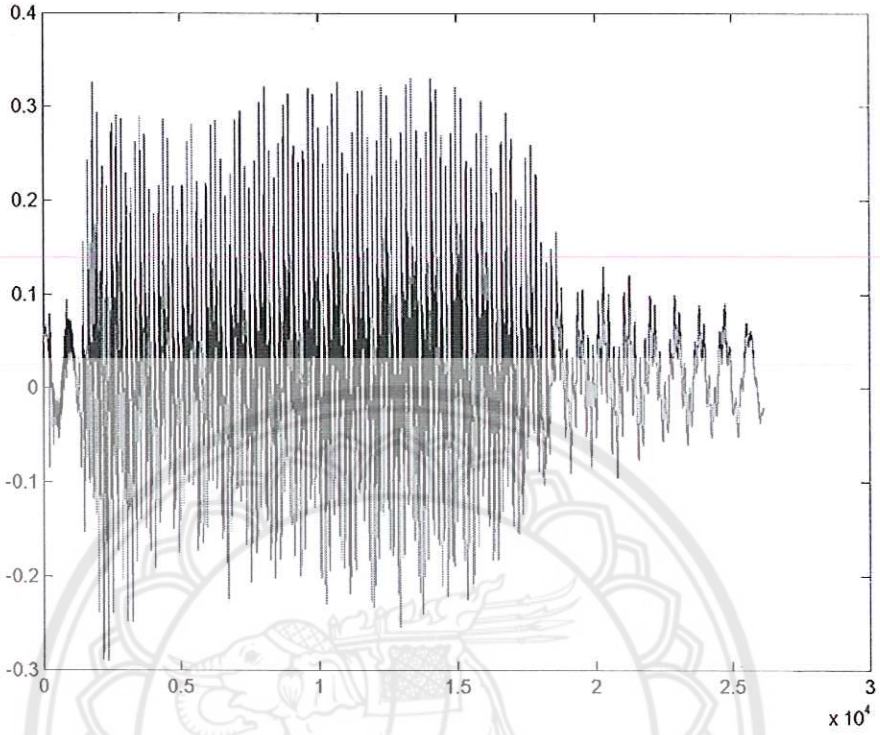
รูปที่ 4.8 ภาพแสดงคลื่นเสียงคำว่า “วาด”

โปรแกรมจะทำการประมวลผลคำและได้เอาท์พุตในการประมวลผลเป็นการสั่งงานเปิดโปรแกรมวาดรูป ดังแสดงใน รูปที่ 4.9



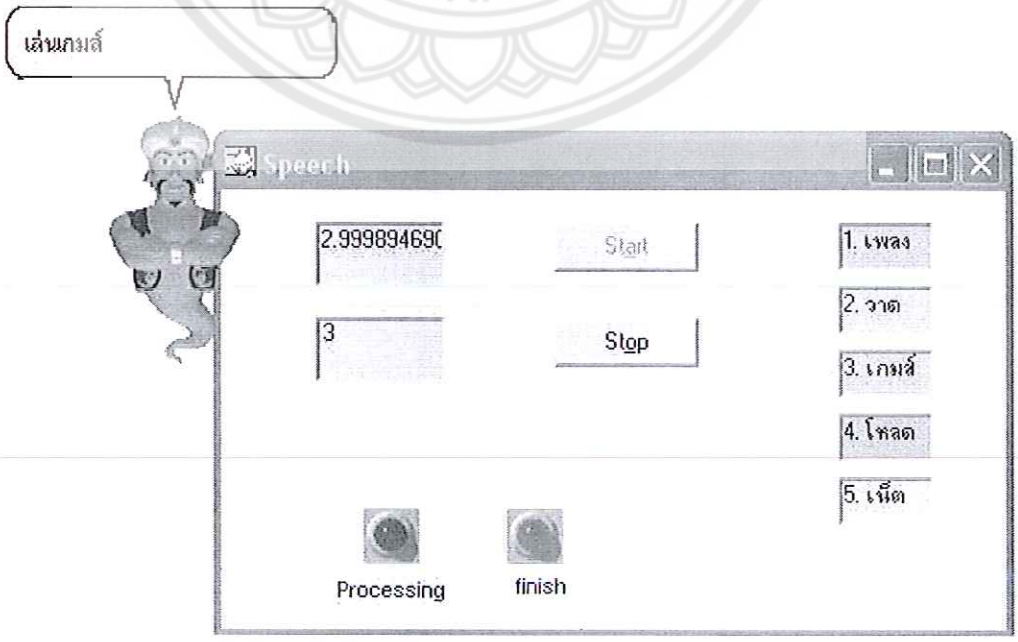
รูปที่ 4.9 ภาพแสดงโปรแกรมคำว่า “วาด”

ถ้าโปรแกรมได้รับอินพุตเป็นเสียงคำว่า “เกมส์” โดยมีคลื่นเสียงแสดงดังรูปที่ 4.10



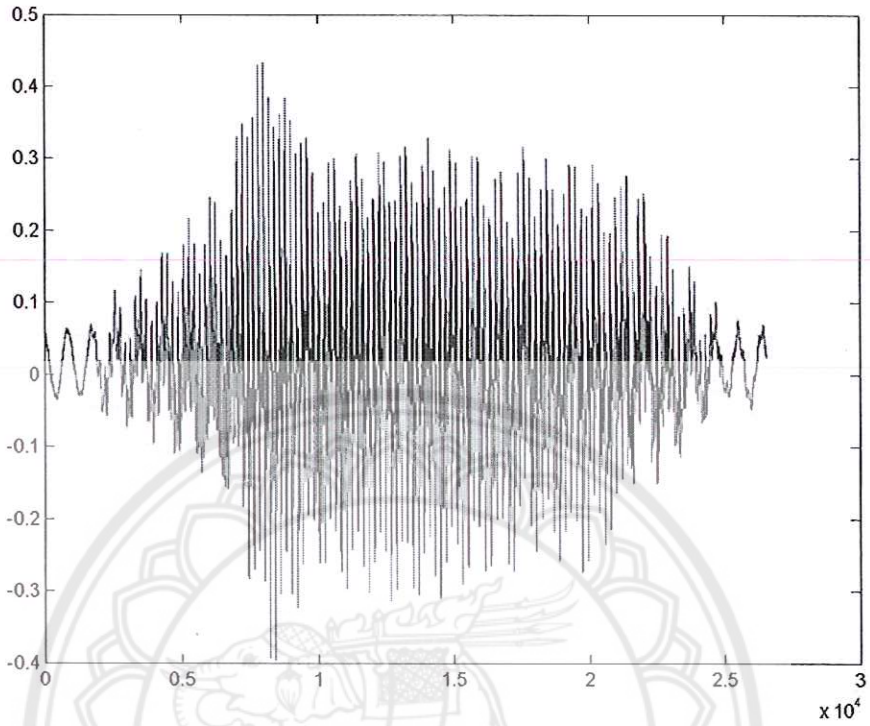
รูปที่ 4.10 ภาพแสดงคลื่นเสียงคำว่า “เกมส์”

โปรแกรมจะทำการประมวลผลคำและได้เอาท์พุตในการประมวลผลเป็นการสั่งงานเปิดโปรแกรมเกมส์ ดังแสดงใน รูปที่ 4.11



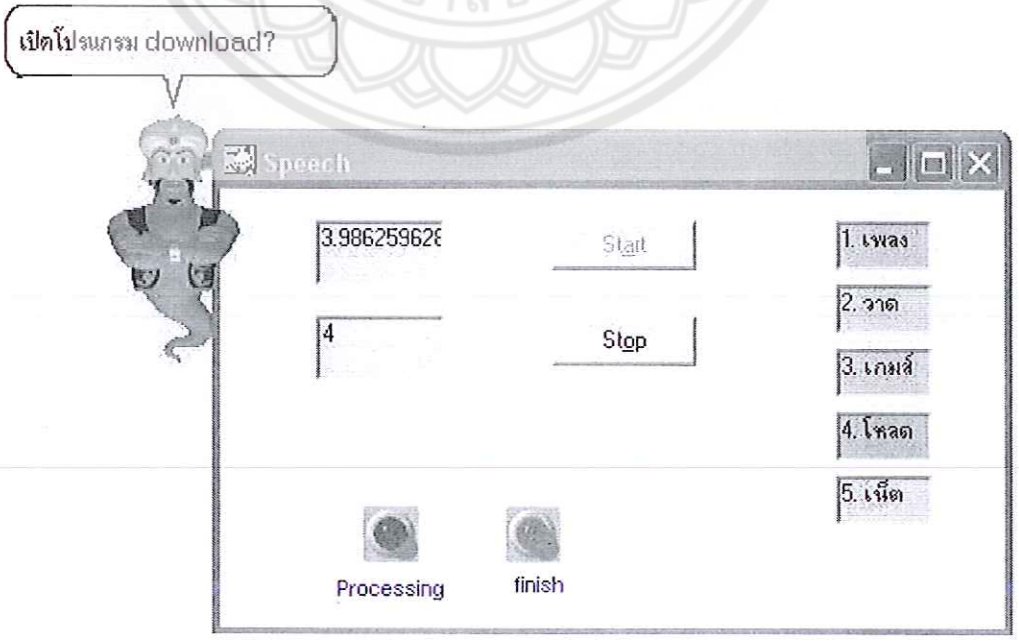
รูปที่ 4.11 ภาพแสดงโปรแกรมคำว่า “เกมส์”

ถ้าโปรแกรมได้รับอินพุตเป็นเสียงคำว่า “โหลด” โดยมีคลื่นเสียงแสดงดังรูปที่ 4.12



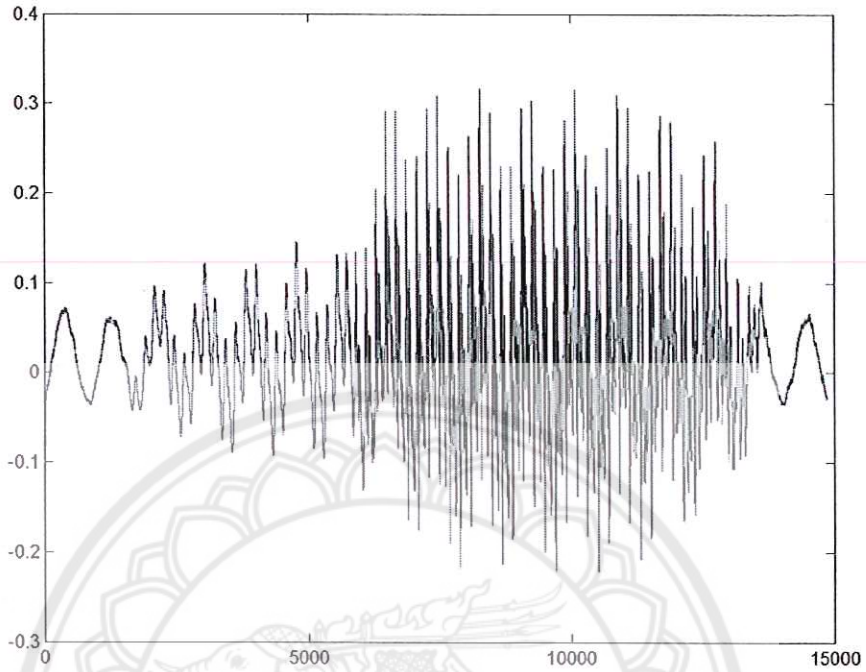
รูปที่ 4.12 ภาพแสดงคลื่นเสียงคำว่า “โหลด”

โปรแกรมจะทำการประมวลผลคำและได้เอาท์พุตในการประมวลผลเป็นการสั่งงานเปิดโปรแกรม download ดังแสดงใน รูปที่ 4.13



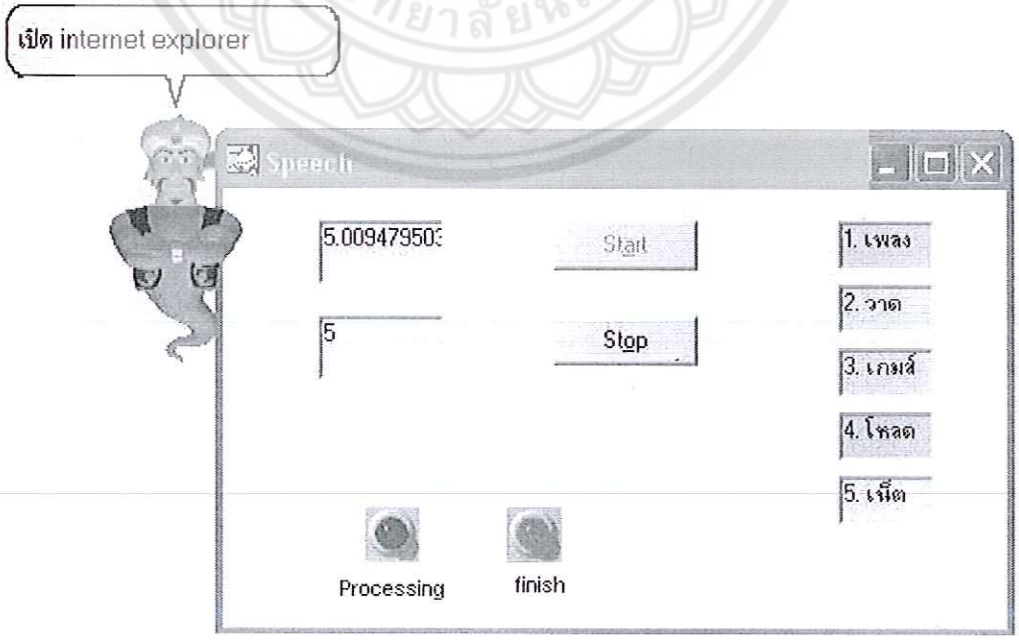
รูปที่ 4.13 ภาพแสดงโปรแกรมคำว่า “โหลด”

ถ้าโปรแกรมได้รับอินพุตเป็นเสียงคำว่า “เนต” โดยมีคลื่นเสียงแสดงดังรูปที่ 4.14



รูปที่ 4.14 ภาพแสดงคลื่นเสียงคำว่า “เนต”

โปรแกรมจะทำการประมวลผลคำและได้เอาท์พุตในการประมวลผลเป็นการสั่งงานเปิดโปรแกรม Internet explorer ดังแสดงใน รูปที่ 4.15



รูปที่ 4.15 ภาพแสดงโปรแกรมคำว่า “เนต”

หลังจากที่ได้เอาท์พุตของการประมวลผลแล้ว โปรแกรมจะทำการเปิดโปรแกรม ตามที่ได้กำหนดไว้

จากการทดสอบผลการทำงานของโปรแกรมที่พัฒนาขึ้นพบว่า โปรแกรมมีการเชื่อมต่อการทำงานกันในส่วนต่างๆ และสามารถนำผลลัพธ์ที่ได้จากการประมวลผลคำมาใช้ในการสั่งงานเปิดโปรแกรมคอมพิวเตอร์ได้ โดยการประมวลผลคำจะมีความผิดพลาดบ้าง ทั้งนี้ขึ้นอยู่กับปัจจัยหลายๆ ด้าน เช่น สภาพแวดล้อมขณะทำการทดลองใช้งานโปรแกรม มีเสียงรบกวน ผู้ทดสอบพูดไม่ชัดเจน และอาจเกิดจากตัวอย่างของเสียงพูด ที่ใช้ในการเรียนรู้สำหรับโครงข่ายประสาทเทียมยังไม่ใช่ตัวอย่างที่ดีพอ ซึ่งปัญหาต่างๆเหล่านี้ควรจะมีการพัฒนาและแก้ไข เพื่อให้โปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพในอนาคตต่อไป

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

การทำงานในส่วนของโครงข่ายประสาทเทียม จะประมวลผลโดยการเลียนแบบการทำงานของสมองมนุษย์ โดยใช้แบบจำลองเชิงคณิตศาสตร์ ลักษณะโครงสร้างและการทำงานของโครงข่ายประสาทเทียมโดยสรุปเป็นดังนี้คือ

- ภายในโครงข่ายประสาทเทียมจะประกอบไปด้วย Layer หลาย Layers ซึ่งในแต่ละ Layer จะประกอบไปด้วย neuron หรือ node หลายๆ ตัว
- ข้อมูลจะประมวลผลใน neuron หรือ node ซึ่งในแต่ละ Input จะรับค่าข้ามาคำนวณโดยอาศัยการจำลองทางคณิตศาสตร์
- ข้อมูลจะถูกส่งผ่านระหว่าง node ผ่านการเชื่อมโยงที่เรียกว่า Link
- โดยแต่ละ Link มีค่าแสดงความสัมพันธ์ (weight) ซึ่งโดยทั่วไปจะถูกนำไปคูณกับค่าของข้อมูลที่อยู่บน link นั้น
- Node แต่ละ Node จะใช้สมการ ในการคำนวณหาผลลัพธ์ของ node แต่ละ node นั้นจะแสดงเอาท์พุตของ Neural Network ซึ่งมีสิ่งที่เป็นลักษณะเฉพาะของ Network นั้นๆ ตามอัลกอริทึม ที่ได้ทำการเขียนโปรแกรมคอมพิวเตอร์ และ จะแสดงเอาท์พุตที่คำนวณได้

ซึ่งในทฤษฎีโครงข่ายประสาทเทียมที่นำมาใช้ในโครงงาน คือ การประมวลผลคำ เพื่อเป็นการแบ่งแยกคำให้เห็นความแตกต่างของคำแต่ละคำ และ นำผลที่ได้จากการประมวลผลมาใช้งาน โดยในโครงงานนี้ได้นำเอาท์พุตที่ได้มาเปิดโปรแกรมเพื่อใช้งาน

จากการประยุกต์ใช้ทฤษฎีโครงข่ายประสาทเทียม ในการพัฒนาโปรแกรมเพื่อวิเคราะห์เสียงพูดแล้วนำมาใช้ในการประมวลผลคำพบว่า เนื่องจากลักษณะความแตกต่างกันของเสียงพูดของคำแต่ละคำ ทำให้โครงข่ายประสาทเทียมสามารถเรียนรู้และประมวลผลคำที่ได้ยินออกมาได้ ทำให้สามารถนำไปประยุกต์ใช้ในการสั่งงานเปิดโปรแกรมคอมพิวเตอร์ได้ โดยจากการทดสอบประสิทธิภาพการทำงานพบว่า โปรแกรมมีความถูกต้องในการทำงานเฉลี่ยคิดเป็น 62.25% โดยทดสอบจากผู้ทดสอบ 4 คน สั่งเปิดโปรแกรมคอมพิวเตอร์โดยใช้เสียงพูดจำนวน 100 ครั้ง

5.2 ปัญหาในการทดลอง

- ทฤษฎีโครงข่ายประสาทเทียม การที่จะให้ผลที่ออกมาเป็นที่แน่นอนจะต้องทำงานวนลูปการทำงานกลับไปกลับมาหลายรอบ จึงเป็นเหตุให้การทำงานล่าช้า

- แต่ละคนมีความแตกต่างในการใช้ภาษาไม่ว่าจะเป็นระดับเสียงสูงต่ำ หรือความเร็วในการออกเสียง ทำให้อินพุตที่ได้มีความหลากหลายแตกต่างกันมาก จึงเป็นการยากในกระบวนการเรียนรู้เสียงของแต่ละคน ซึ่งมีผลทำให้การทำงานของโปรแกรมที่พัฒนามีความผิดพลาด

5.3 ข้อเสนอแนะ

- ในโครงงานที่ทางผู้พัฒนาได้พัฒนา จะใช้ภาษา Visual Basic หากมีการศึกษาและพัฒนาต่อไปอาจเลือกใช้ภาษาอื่นเพื่อพิจารณาข้อดีข้อเสีย เช่น ความเร็วและความถูกต้องแม่นยำของซอร์ฟแวร์

- ถ้าจัดให้มีการเรียนการสอนเรื่องนี้ในหลักสูตรได้ ก็จะเป็นการดีอย่างยิ่ง เพราะจะเป็นประโยชน์ต่อบุคคลอื่นได้อีกมาก

- เพื่อให้โปรแกรมมีความถูกต้องในการทำงานมากขึ้น อุปกรณ์ที่นำมาใช้ก็มีความสำคัญเช่นกัน เช่นไมโครโฟน ถ้าใช้ไมโครโฟนที่ไม่มีคุณภาพก็จะทำให้โปรแกรมทำงานผิดพลาดได้

บรรณานุกรม

[1] Joe Tebelskis. **Speech Recognition using Neural Networks** Pittsburgh, Pennsylvania :1995

[2] James A. Freeman, David M. Skapura. **Neural Networks Algorithms, Applications, and Programming Techniques**. Addison-Wesley Publishing Company. USA :1991

[3] Paras Chopra, **Neural networks: Under Standing Using Visual Basic** [Online]. Available: http://www.planet-source-code.com /vb/scripts/ShowCode.asp?txtCodeId=55011&lngWId=1

[4] Valluru B. Rao. **C++ Neural Networks and Fuzzy Logic.** IDG Books Worldwide, Inc. USA :1995

[5] **Software for Machine mission by MVTex Software GmbH** [Online]. Available: http://www.programmersheaven.com/zone22/cat165/1199.htm

ภาคผนวก ก
คำอธิบายโปรแกรม

การกำหนดค่าตัวแปร

```
Private Type Dendrite
Weight As Double
End Type

Private Type Neuron
Dendrites() As Dendrite
DendriteCount As Long
Bias As Double
Value As Double
Delta As Double
End Type

Private Type Layer
Neurons() As Neuron
NeuronCount As Long
End Type

Private Type NeuralNetwork
Layers() As Layer
LayerCount As Long
LearningRate As Double
End Type
```

ฟังก์ชั่นการสร้างโครงข่ายประสาทเทียม

```
Function CreateNet(LearningRate As Double, ArrayOfLayers As Variant) As Integer
Dim i, j, k As Integer
Network.LayerCount = UBound(ArrayOfLayers)
If Network.LayerCount < 2 Then
   CreateNet = 0
   Exit Function
End If
Network.LearningRate = LearningRate
ReDim Network.Layers(Network.LayerCount) As Layer
For i = 1 To UBound(ArrayOfLayers)
DoEvents
   Network.Layers(i).NeuronCount = ArrayOfLayers(i)
   ReDim Network.Layers(i).Neurons(Network.Layers(i).NeuronCount) As Neuron
   For j = 1 To ArrayOfLayers(i)
   DoEvents
      If i = UBound(ArrayOfLayers) Then
         Network.Layers(i).Neurons(j).Bias = GetRand
         Network.Layers(i).Neurons(j).DendriteCount = ArrayOfLayers(i - 1)
         ReDim
Network.Layers(i).Neurons(j).Dendrites(Network.Layers(i).Neurons(j).DendriteCount) As_
Dendrite
         For k = 1 To ArrayOfLayers(i - 1)
            DoEvents
            Network.Layers(i).Neurons(j).Dendrites(k).Weight = GetRand
         Next k
      ElseIf i = 1 Then
         DoEvents
      Else
         Network.Layers(i).Neurons(j).Bias = GetRand
         Network.Layers(i).Neurons(j).DendriteCount = ArrayOfLayers(i - 1)
```

```
        ReDim
Network.Layers(i).Neurons(j).Dendrites(Network.Layers(i).Neurons(j).DendriteCount) As_
Dendrite
        For k = 1 To ArrayOfLayers(i - 1)
          DoEvents
          Network.Layers(i).Neurons(j).Dendrites(k).Weight = GetRand
        Next k
      End If
    Next j
Next i
CreateNet = 1
End Function
```

ฟังก์ชั่นการทำงานของโครงข่ายประสาทเทียม

```
Function Run(ArrayOfInputs As Variant) As Variant
Dim i, j, k As Integer
If UBound(ArrayOfInputs) <> Network.Layers(1).NeuronCount Then
  Run = 0
  Exit Function
End If
For i = 1 To Network.LayerCount - 1
DoEvents
  For j = 1 To Network.Layers(i).NeuronCount
  DoEvents
    If i = 1 Then
      Network.Layers(i).Neurons(j).Value = ArrayOfInputs(j)
    Else
      Network.Layers(i).Neurons(j).Value = 0
      For k = 1 To Network.Layers(i - 1).NeuronCount
        DoEvents
```

```
            Network.Layers(i).Neurons(j).Value = Network.Layers(i).Neurons(j).Value +_
Network.Layers(i - 1).Neurons(k).Value * Network.Layers(i).Neurons(j).Dendrites(k).Weight
        Next k
        Network.Layers(i).Neurons(j).Value = Sigmoid(Network.Layers(i).Neurons(j).Value +_
Network.Layers(i).Neurons(j).Bias)
      End If
    Next j
    Next i
    For i = 1 To Network.Layers(Network.LayerCount).NeuronCount
      For k = 1 To Network.Layers(Network.LayerCount).Neurons(i).DendriteCount_
Network.Layers(Network.LayerCount).Neurons(i).Value =_
Network.Layers(Network.LayerCount - 1).Neurons(i).Value *_
Network.Layers(Network.LayerCount).Neurons(i).Dendrites(k).Weight +_
Network.Layers(Network.LayerCount).Neurons(i).Bias
    Next k
    Next i
ReDim OutputResult(Network.Layers(Network.LayerCount).NeuronCount) As Double
For i = 1 To (Network.Layers(Network.LayerCount).NeuronCount)
    DoEvents
    OutputResult(i) = (Network.Layers(Network.LayerCount).Neurons(i).Value)
Next i
Run = OutputResult
End Function
```

ฟังก์ชั่นการทำงานของโครงข่ายประสาทเทียมส่วนของ Back propagation

```
Function SupervisedTrain(inputdata As Variant, outputdata As Variant) As Integer
Dim i, j, k As Integer
If UBound(inputdata) <> Network.Layers(1).NeuronCount Then
    SupervisedTrain = 0
    Exit Function
End If
```

```
If UBound(outputdata) <> Network.Layers(Network.LayerCount).NeuronCount Then
SupervisedTrain = 0
   Exit Function
End If
Call Run(inputdata)
For i = 1 To Network.Layers(Network.LayerCount).NeuronCount
DoEvents
Network.Layers(Network.LayerCount).Neurons(i).Delta = (-2) * 1 *_
(outputdata(i) - Network.Layers(Network.LayerCount).Neurons(i).Value)
   For j = Network.LayerCount - 1 To 2 Step -1
   DoEvents
      For k = 1 To Network.Layers(j).NeuronCount
      DoEvents
         Network.Layers(j).Neurons(k).Delta = Network.Layers(j).Neurons(k).Value *_
(1 - Network.Layers(j).Neurons(k).Value) * Network.Layers_
(j + 1).Neurons(i).Dendrites(k).Weight * Network.Layers(j + 1).Neurons(i).Delta
      Next k
   Next j
Next i

For i = Network.LayerCount To 2 Step -1
DoEvents
   For j = 1 To Network.Layers(i).NeuronCount
   DoEvents
      Network.Layers(i).Neurons(j).Bias = Network.Layers(i).Neurons(j).Bias -_
(Network.LearningRate * 1 * Network.Layers(i).Neurons(j).Delta)
      For k = 1 To Network.Layers(i).Neurons(j).DendriteCount
      DoEvents

Network.Layers(i).Neurons(j).Dendrites(k).Weight =_
Network.Layers(i).Neurons(j).Dendrites(k).Weight -_
```

```
(Network.LearningRate * Network.Layers(i - 1).Neurons(k).Value *_
Network.Layers(i).Neurons(j).Delta)
      Next k
   Next j
Next i
SupervisedTrain = 1
End Function
```

ฟังก์ชั่นการสุ่มค่าของตัวแปร

```
Function GetRand() As Double
Randomize
GetRand = 2 - (1 + Rnd + Rnd)
End Function

Private Function Sigmoid(Value As Double)
   Sigmoid = (1 / (1 + Exp(Value * -1)))
End Function
```

ฟังก์ชั่นคำนวณค่า Sigmoid

```
Private Function Sigmoid(Value As Double)
   Sigmoid = (1 / (1 + Exp(Value * -1)))
End Function
```

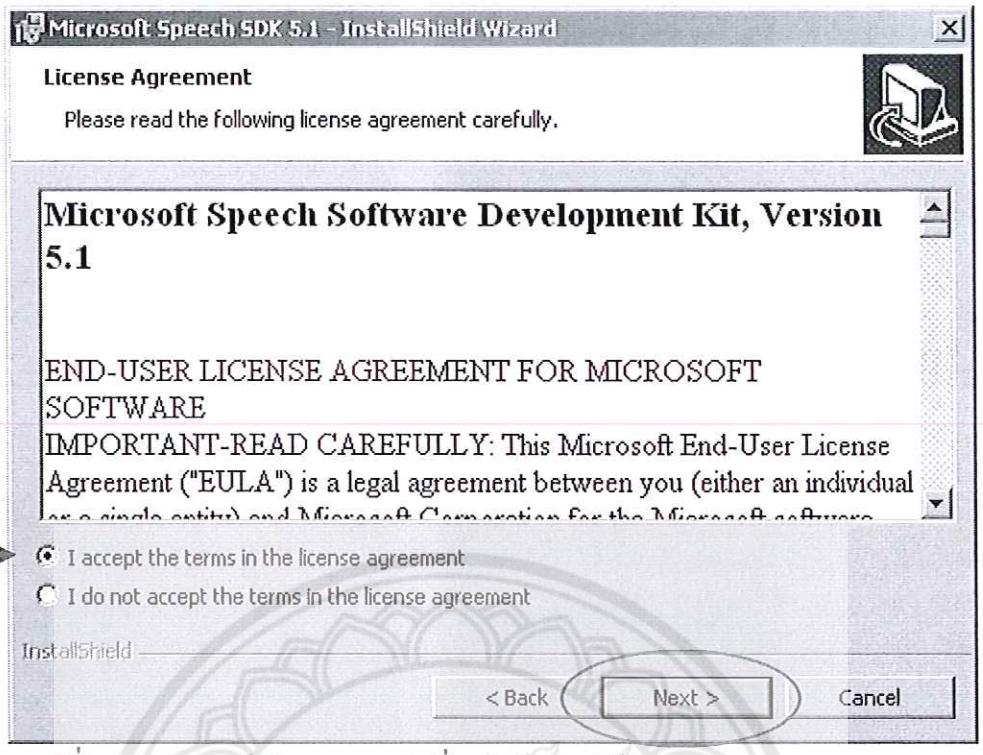
ภาคผนวก ข

การติดตั้งโปรแกรม Microsoft SDK 5.1

หลังจากที่ได้ดาวน์โหลดไฟล์ speech SDK 5.1มาจากเว็บไซท์ www.microsoft.com แล้ว ให้ทำการดับเบิ้ลคลิกที่ไฟล์ Setup.exe จะขึ้นหน้าต่าง ดังภาพที่ ข.1

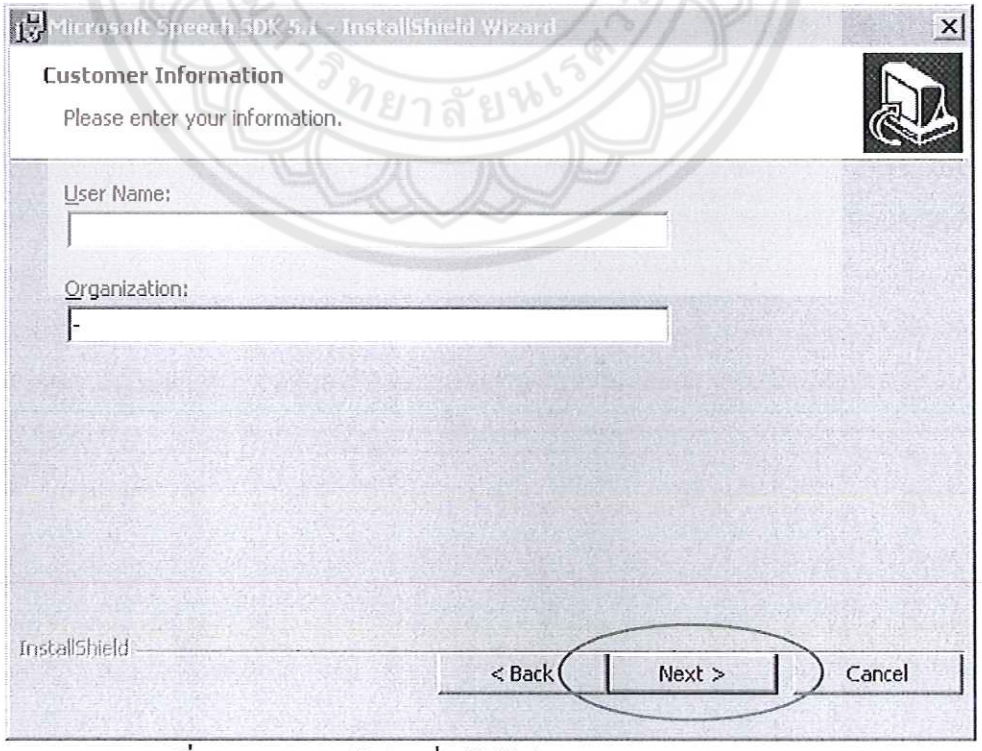
รูปที่ ข.1 แสดงหน้าต่าง Install ของโปรแกรม Microsoft SDK 5.1

หลังจากทำการที่คลิกที่ปุ่ม Next จะปรากฏดังภาพ ข.2



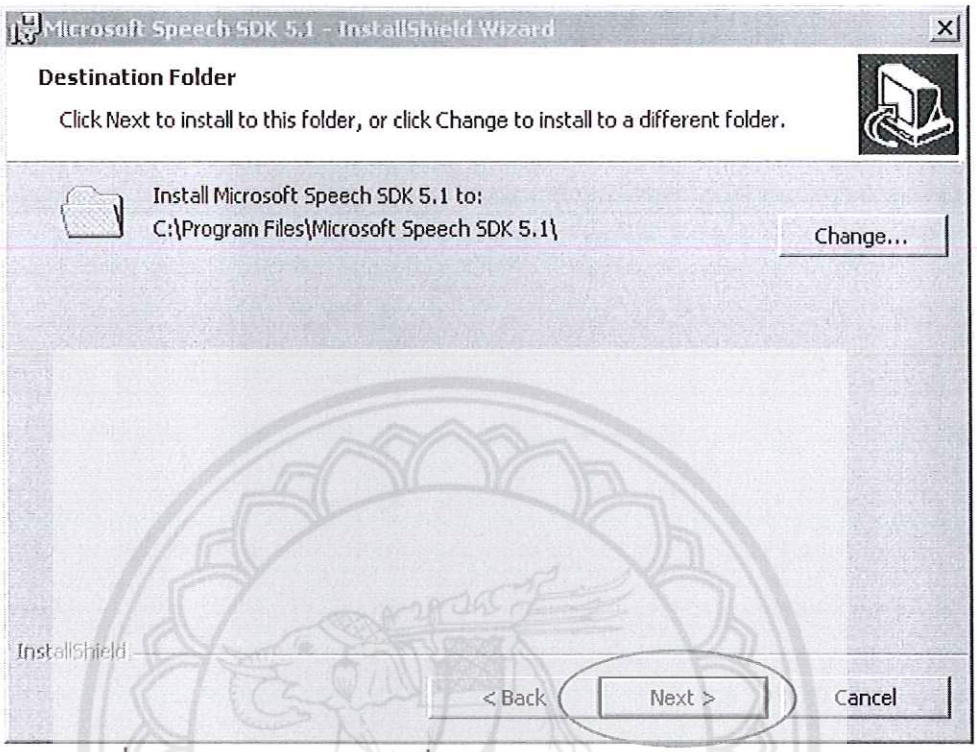
รูปที่ ข.2 แสดงหน้าต่างการยอมรับเงื่อนไขในการใช้โปรแกรม Microsoft SDK 5.1

ทำการคลิกที่ ปุ่ม I accept the term in the license agreement แล้วคลิก ปุ่ม Next จะปรากฏดังภาพที่ ข.3



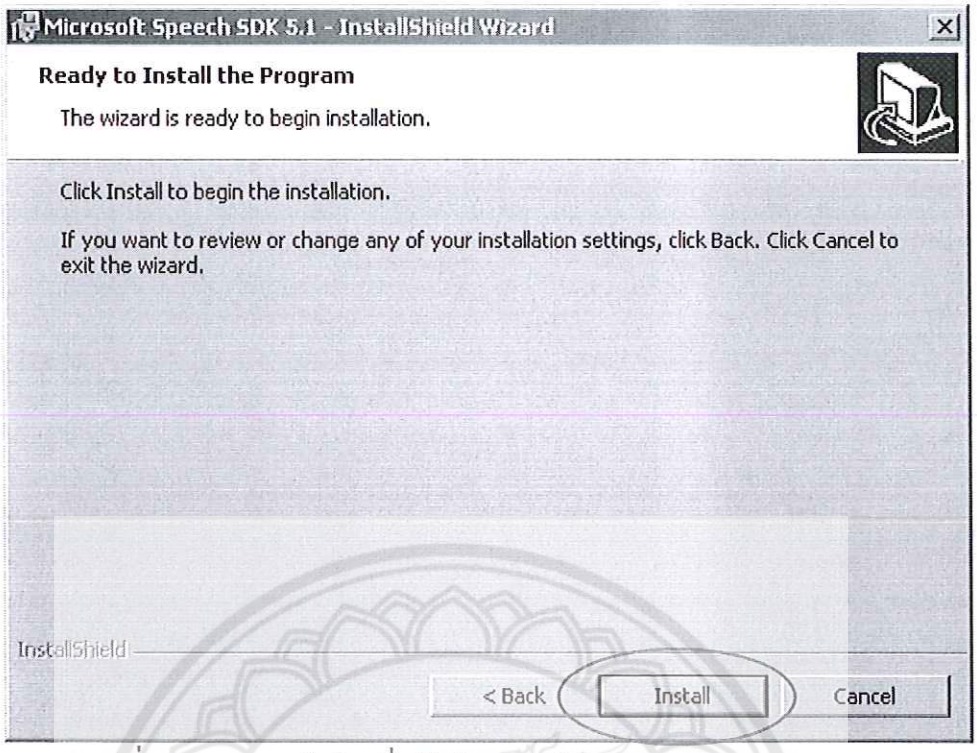
รูปที่ ข.3 แสดงหน้าต่าง ชื่อผู้ใช้โปรแกรม Microsoft SDK 5.1

หลังจากที่ได้ทำการใส่ชื่อผู้ใช้ลงในช่อง User Name แล้วคลิกปุ่ม Next แล้ว จะแสดงดังภาพที่ ข.4



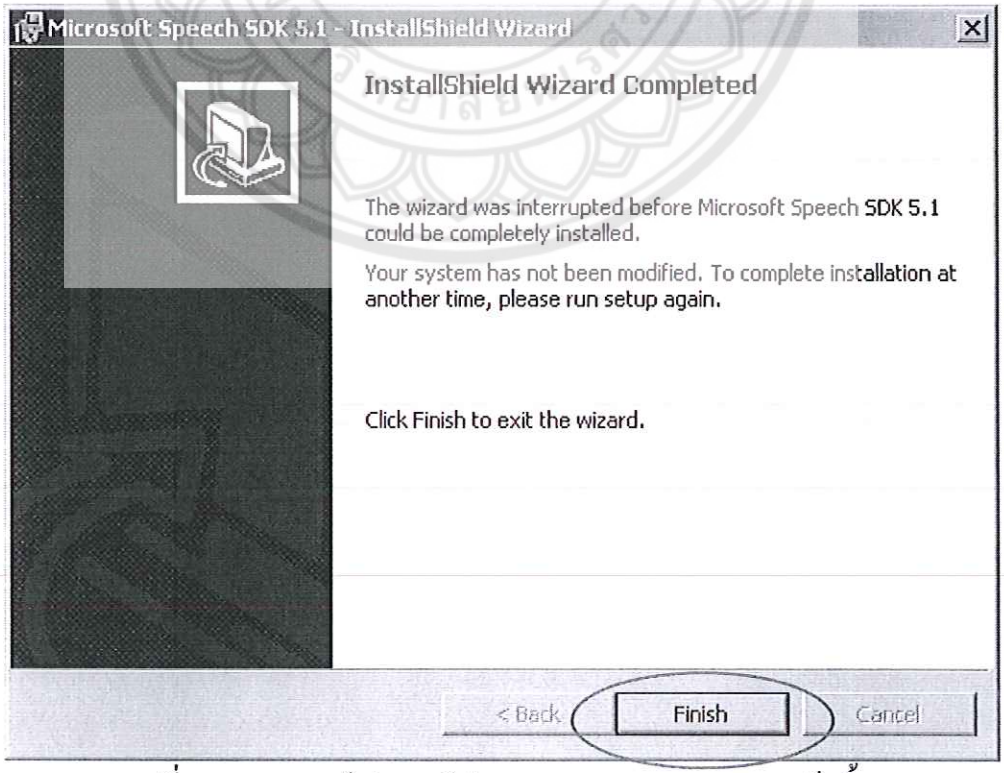
รูปที่ ข.4 แสดงหน้าต่าง แฟ้มที่จะทำการลงโปรแกรม Microsoft SDK 5.1

หลังจากที่ได้เลือกแฟ้มที่จะทำการลงโปรแกรมแล้วคลิกปุ่ม Next แล้ว **โปรแกรมจะเริ่มติดตั้งเมื่อ**ผู้ใช้กดปุ่ม install ดังภาพที่ ข.5



รูปที่ ข.5 แสดงหน้าต่าง เริ่มต้นทำการลงโปรแกรม Microsoft SDK 5.1

หลังจากที่ได้คลิกปุ่ม Install เพื่อทำการลงโปรแกรมแล้ว โปรแกรมจะทำการลงโปรแกรม และ หลังจาก เมื่อทำการลงโปรแกรมเสร็จแล้ว จะแสดงดังภาพที่ ข.6



รูปที่ ข.6 แสดงหน้าต่าง ลงโปรแกรม Microsoft SDK 5.1 เสร็จสิ้น