การแยกแยะและรู้จำข้อมูลเสียง

Speech Recognition

T 0 3 4 0 6 9

โดย

นางสาว ปวิณา เตโช

นางสาว ปิยะวรรณ ปิยะพรมดี

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่
เลขทะเบียน 34069
วัน, เดือน, ปี - 1 ต.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแยกแยะและรู้จำข้อมูลเสียง

Speech Recognition

โดย

นางสาว ปวิณา เตโช

นางสาว ปิยะวรรณ ปิยะพรมดี

อาจารย์ที่ปรึกษา

อ. มานิตย์ เกียรติกำจายขจร

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2541

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การแยกแยะและรู้จำข้อมูลเสียง

ผู้จัดทำ

1.นางสาว ปวิณา เตโช 38014285

2.นางสาว ปิยะวรรณ ปิยะพรมดี 38014301

อาจารย์ที่ปรึกษา

(อาจารย์ มานิตย์ เกียรติกำจายขจร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์	การแยกแยะและรู้จำข้อมูลเสียง (Speech Recognition)	
จัดทำโดย	น.ส. ปวิณา เตโช	เลขประจำตัว 38014285
	น.ส. ปิยะวรรณ ปิยะพรมดี	เลขประจำตัว 38014301

ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ .....

(อ.มานิตย์ เกียรติกำจายขจร)

อาจารย์ที่ปรึกษา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแยกแยะและรู้จำข้อมูลเสียง

นางสาว ปวิณา เตโช
นางสาว ปิยะวรรณ ปิยะพรมดี
อ.มานิตย์ เกียรติกำจายขจร อาจารย์ที่ปรึกษา
ปีการศึกษา 2541

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอวิธีการตรวจสอบข้อมูลเสียงเลข0-9 ด้วยนิวรอลเน็ตเวอร์คชนิด Back Propagation เพื่อให้รู้จำเสียงพูดของผู้พูดคนเดียว(single user) โดยเสียงพูดจะถูกบันทึกผ่านเข้ามาทางไมโครโฟน สัญญาณที่ได้ คือ แอมปลิจูดในแกนเวลา และแปลงเป็นสัญญาณดิจิตอลด้วย การ์ดเสียง (sound blaster card) จากนั้นจะผ่านส่วนปรีโปรเซสซิ่ง เพื่อลดขนาดข้อมูลเสียงโดยทำการปรับข้อมูลในแกนเวลา ข้อมูลเสียงนี้จะผ่านการแปลงฟูเรียร์อย่างเร็ว(FFT) ได้สัญญาณเสียงในโดเมนความถี่ จากนั้นทำการหาคุณลักษณะเฉพาะตัวของเสียงที่ต้องการทราบว่ามีความเหมือนหรือสอดคล้องกับเสียงในกลุ่มเสียงฐานข้อมูลหลักมากเพียงใด โดยใช้สมการสหสัมพันธ์ (Correlation) ค่าที่ได้จะนำไปใช้เทรนนิ่งบนเน็ตเวอร์คเพื่อคำนวณหาค่าถ่วงน้ำหนักที่เหมาะสมกับข้อมูลตัวอย่างที่สอดคล้องกับเอาท์พุทตามต้องการ เมื่อนำข้อมูลทดสอบป้อนให้กับอินพุทของเน็ตเวอร์ค แล้วใช้ค่าถ่วงน้ำหนักที่คำนวณได้จากการเทรนนิ่ง มาอ้างอิงกับข้อมูลทดสอบ จะทำให้เน็ตเวอร์คตรวจสอบและแยกแยะเสียงได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPEECH RECOGNITION

Miss Pawina Techo

Miss Piyawan Piyapromdee

MR Manit Kiatkumjaiykhajorn (Advisor)

2nd semester , Education year 1998

Abstract

This thesis present the application of Backpropagation neural network ,that use for numeral speech recognition . Speech signal will be detected from microphone ,then convert to digital signal using sound blaster card . All data sets are transformed to frequency domain by FFT (Fast Fourier Transform). The transformed data are brought to find correlation and be used as a standarad profile for matching. Then the correlated data are brought to train on neural network in order to accummulate the value of weight.The accummulated weight is used as a reference to the testing data.Consequencely, the neural network is able to test and classifies the numerul speech correctly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
บทนำ	1
บทที่2	3
การได้ยิน	3
2.2 การเกิดของเสียง	4
2.3 การเข้าใจ	5
2.4 ความดัง	5
บทที่3	
การวิเคราะห์เสียงพูด	7
3.1 การวิเคราะห์เสียงพูดในช่วงสั้น	7
3.2 ค่าพารามิเตอร์ในโดเมนความถี่	8
บทที่4	
โครงข่ายประสาทเทียม	15
4.1 โครงข่ายประสาทชีวภาพ	15
4.2 โครงข่ายประสาทเทียม	20
บทที่5	21
ผล วิจารณ์ และแนวทางการปรับปรุง	21
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

การพัฒนาคอมพิวเตอร์ในปัจจุบันมุ่งเน้นความสะดวกสบายในการใช้งานมากขึ้น การสามารถสั่งงานด้วยเสียงก็เป็นสิ่งที่มีการค้นคว้า และได้รับความสนใจอย่างกว้างขวาง ทำให้มีทฤษฎีและอัลกอริทึมในการตัดสินใจแยกแยะเสียง(Speech Recognition)ให้กับคอมพิวเตอร์อย่างมากมาย

การใช้โครงข่ายประสาทเทียมมาเป็นส่วนตัดสินใจ ก็เป็นแนวทางหนึ่งในการแยกแยะและรู้จำโดยลอกเลียนแบบโครงข่ายประสาททางชีวภาพของมนุษย์ ที่มีการเรียนรู้และตัดสินใจต่อสิ่งต่างๆได้เป็นอย่างดี

ในโครงงานนี้ออกแบบใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ(Back-propagation Neural Network)ซึ่งมีรูปแบบการสอนแบบมีผู้สอน(Supervise Training) มาเป็นส่วนตัดสินใจให้กับคอมพิวเตอร์ เพื่อแยกแยะ-รู้จำเสียงพูดตัวเลข 0-9 แบบผู้ใช้เพียงคนเดียว(Single User) และใช้การแปลงสัญญาณเสียงในโดเมนเวลา(Time Domain)ไปสู่แกนความถี่(Frequency Domain)ด้วยฟาสฟูเรียร์ ทรานส์ฟอร์ม(Fast Fourier Transform) เป็นส่วนปรี-โปรเซสซิ่ง เพื่อดึงเอาคุณสมบัติของเสียงพูดแต่ละเสียงออกมาส่งให้กับเน็ตเวิร์คต่อไป

ส่วนปรี-โปรเซสซิ่งจึงมีความสำคัญต่อความสามารถตัดสินใจอย่างมาก วิธีการในส่วนปรี-โปรเซสซิ่งที่สามารถบ่งความแตกต่าง และคุณสมบัติเฉพาะของเสียงได้ชัดเจนจะทำให้เน็ตเวิร์คสามารถตัดสินใจได้ดีและเร็วขึ้น

กระบวนการแยกแยะและรู้จำจึงลำดับได้ดังแสดงต่อไปนี้

รูปที่ 1.1 กระบวนการแยกแยะ-รูจำเสียง

โดยในส่วนปรับแกนเวลา จะใช้การกำหนดขอบเขตข้อมูลแต่ละเสียงให้เท่าๆกัน เพราะการพูดของคนแต่ละครั้ง แม้จะเป็นคำ คำเดียวกันจะใช้เวลาออกเสียงต่างกันไป ขั้นตอนนี้จึงเสมือนกำหนดเงื่อนไขบนแกนเวลาของทุกเสียงให้เสมอภาคกัน

ส่วนฟาสฟูเรียร์ ทรานส์ฟอร์มจะแปลงข้อมูลแกนเวลาสู่แกนความถี่ เพื่อวิเคราห์คุณสมบัติเสียงตามแกนความถี่แทน

จากนั้นจะปรับแกนความถี่ให้เป็นจำนวนเท่าของความถี่มูลฐาน เพื่อให้สอดคล้องกับกระบวนการได้ยินของหูคน โดยใช้ลอการิทึมฐานสอง แล้วผ่านเข้าสู่กระบวนการตัดสินใจด้วนนิวรอลเน็ตเวิร์คต่อไป

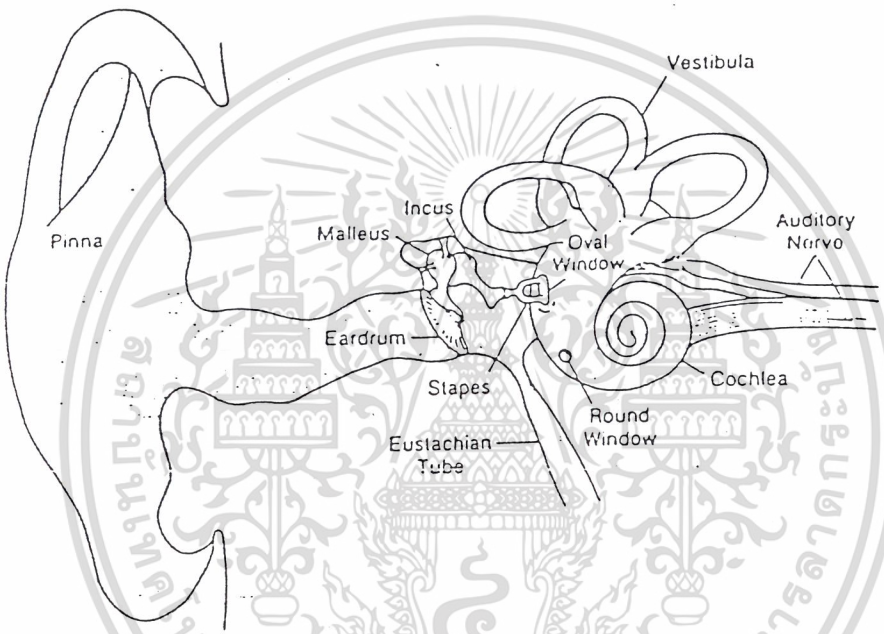
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งแม้ว่าวิธีการตัดสินใจที่เลือกใช้ จะยังไม่สามารถเลียนแบบการทำงานของใยประสาทมนุษย์ที่มีการพัฒนาและเรียนรู้เพิ่มเติมตลอดเวลาได้ ก็จัดว่าเป็นการพัฒนาขั้นพื้นฐานเพื่อการค้นคว้าต่อๆไป นอกจากนี้การเรียนรู้แบบแพร่ย้อนกลับยังเป็นแนวคิดที่ไม่ซับซ้อน เหมาะกับผู้ที่ศึกษาทำความเข้าใจเบื้องต้น และสามารถใช้กับงานเฉพาะขอบเขตระดับหนึ่งได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การได้ยิน (hearing)

ระบบการฟัง หมายถึง สิ่งที่ได้รับมาถูกเปลี่ยนไปเป็นสัญญาณกระตุ้นไปเป็นระบบประสาท เริ่มจากโครงสร้างของหูแบ่งเป็น 3 ส่วน ได้แก่ หูชั้นนอก หูชั้นกลาง และหูชั้นใน



รูปที่ 2.1 แสดงส่วนประกอบของหู

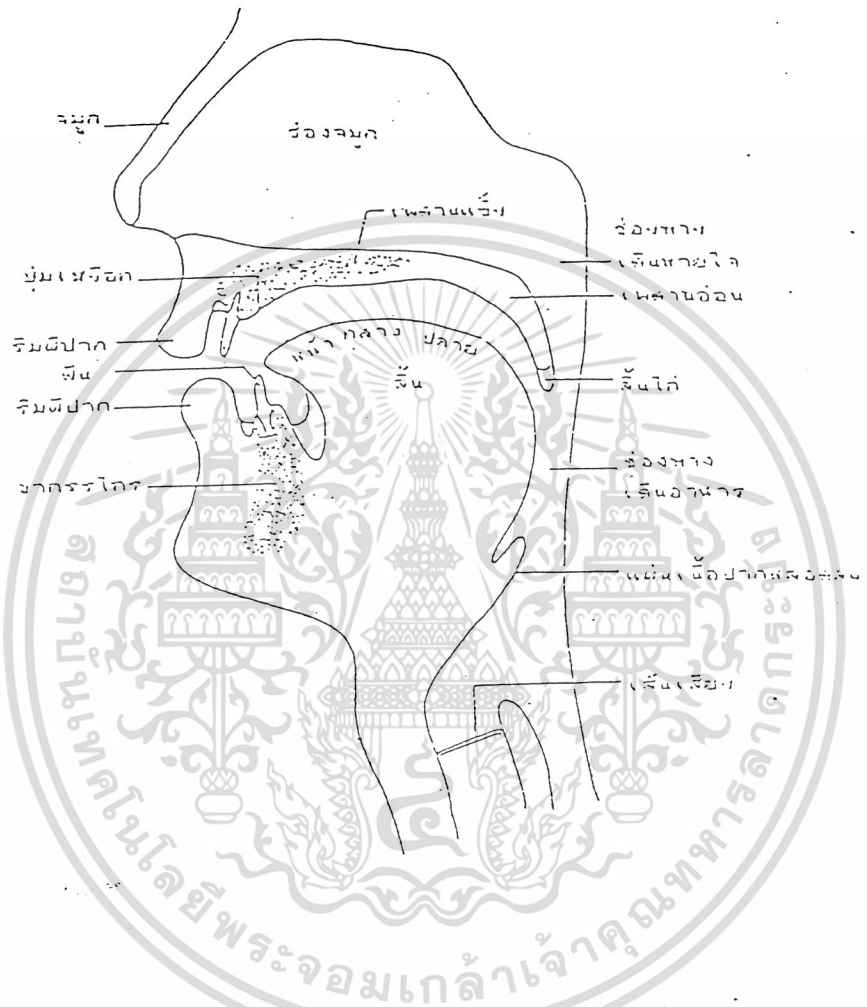
2.1 อวัยวะที่ใช้ในการออกเสียง (Organs of Speech)

อวัยวะที่ใช้ในการออกเสียงพูดมีอยู่หลายส่วน แต่ละส่วนสามารถทำให้เสียงพูดแตกต่างกันออกไปได้ อวัยวะเหล่านี้มีปาก และส่วนต่างๆในปาก ช่องคอ กล่องเสียง ช่องปาก และช่องจมูกดังรายละเอียดในรูปที่ 1.2 อวัยวะที่ใช้ในการออกเสียงมี 2 จำพวก คือ

1.อวัยวะที่ใช้ในการกระทำอาหาร (Articulator) หมายถึง อวัยวะส่วนที่เคลื่อนไหวเพื่อผลักลมไปยังส่วนต่างๆ อวัยวะตัวผลักดันอาหารที่สำคัญ คือ ลิ้น ซึ่งเป็นส่วนที่เคลื่อนไหวได้มากที่สุด

2.อวัยวะที่เป็นตำแหน่งที่เกิดเสียงต่างๆ (Point of Articulator) หมายถึง ตำแหน่งหรือ ฐานกรณ์ที่เกิดเสียงต่างๆ เช่น เพดาน ส่วนต่างๆที่ริมฝีปาก ฟัน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงอวัยวะที่ใช้ในการออกเสียงพูด

2.2 การเกิดของเสียง (Speech Production)

ขั้นตอนการเกิดของเสียงแบ่งได้เป็น 3 ขั้นตอนใหญ่คือ

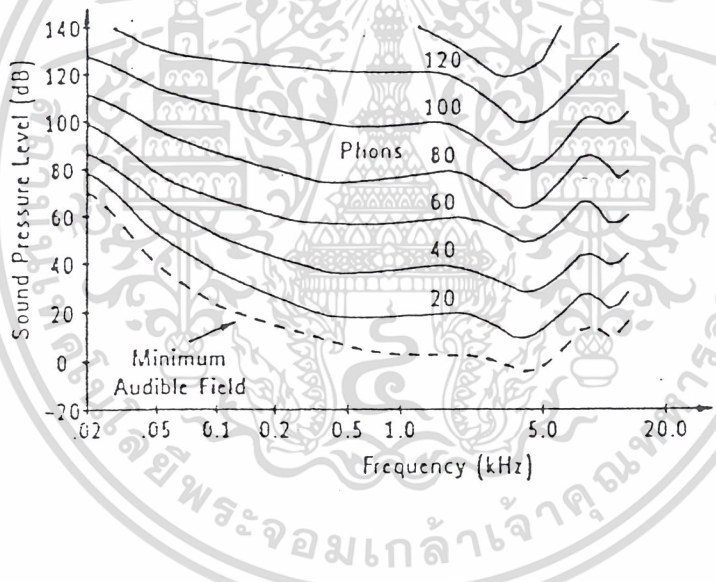
1. ขั้นเริ่มต้น (Initiation) ขั้นตอนนี้เริ่มจากลมถูกขับออกจากปอด เพื่อเข้าสู่ขั้นตอนที่2 ต่อไป
2. ขั้นตอนการดัดแปลงลมที่เส้นเสียง (Phonation) เป็นขั้นตอนที่ลมจากปอดจะผ่านมายังหลอดลม และกล่องเสียง ซึ่งที่กล่องเสียงนี้ เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดปิดทำให้เกิดเสียงได้ 2 ชนิด คือ เสียงก้อง (Voice Sounds) และเสียงไม่ก้อง (Unvoiced Sounds) อวัยวะที่ใช้ในตอนนี้คือ ส่วนที่ต่อจากปอดขึ้นมาที่กล่องเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขั้นตอนเปลี่ยนแปลงลักษณะเสียง (Articulation) ขั้นตอนนี้ลมผ่านออกจากกล่องเสียงจะถูกแปลงให้เกิดเสียงในลักษณะต่างๆอวัยวะที่ใช้ในขั้นตอนนี้ คือ ส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก

2.3 ความเข้าใจ(Preception)

หน้าที่ของการรับเสียงและความสามารถ โดยปกติจะอยู่ในช่วง 16Hz ถึง 16 Hz ในเด็กอาจถึง 20 Hz การรับฟังจะลดสมรรถภาพลงตามอายุ อาจจะถึง 10 Hz การได้ยินความถี่ในช่วงย่านต่ำสัญญาณที่ได้ยินจะเป็นแบบขบวนสัญญาณ (Pulse train) ที่ความถี่ด้านสูงความถี่จะค่อยจางลงจนไม่ได้ยิน ช่วงความเข้มอยู่ในช่วงที่ 1-3 dB ที่ความเข้มต่ำเสียงจะเบาจนไม่ได้ยิน ส่วนที่ความเข้มสูงอาจทำให้ปวดแก้วหู



รูปที่ 2.3 แสดงเส้นแสดงความเท่ากันของระดับความดัง

2.4 ความดัง

การรับรู้ความดังเป็นทั้งหน้าที่ของความถี่ และระดับของเสียง โดยการเปรียบเทียบโทนของเสียงที่ต่างกันของความถี่และแอมปลิจูด

การเข้าใจเสียงพูด ปัญหาพื้นฐานที่กำหนดโดยเสียงพูดคือ สัญญาณเสียงที่เข้ามาที่หูจะแปลงโดยสมองเป็นเสียงพูดได้อย่างไร จากการทดลองหลายๆครั้ง แล้วทำการทดลองก็ได้คำตอบบางส่วน อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็ตามคำตอบที่สมบูรณ์ก็ยังไม่ปรากฏเด่นชัด เราสามารถสรุปความสอดคล้องของบางส่วนที่หาข้อมูลมาได้ดังนี้

ปรากฏว่าสมองจะสร้างความแตกต่างระหว่างเสียงพูดกับสัญญาณที่ไม่ใช่เสียงพูดการทดลองพบว่าเมื่อผู้ฟังได้ยินเสียงที่สังเคราะห์ขึ้นไม่ใช่เสียงพูดแต่ทำคล้ายเสียงพูดจะพบขอบเขตที่มีการเปลี่ยนแปลงกระทันหันข้างใดข้างหนึ่ง จะถูกเข้าใจว่าเป็นเสียงพูด ส่วนอีกข้างจะไม่เป็นเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การวิเคราะห์เสียงพูด (Speech Analysis)

ในการวิเคราะห์เสียงพูดทั้งในโดเมนเวลา (กระทำกับเสียงพูดที่เก็บไว้โดยตรง) และในโดเมนความถี่ (ผ่านการแปลงเป็นความถี่ก่อน) เพื่อหาตัวแทนพารามิเตอร์ในรูปที่เหมาะสมที่สุด สามารถนำไปใช้งานได้ และมีข่าวสารข้อมูลครบถ้วน การวิเคราะห์ในโดเมนเวลา เราต้องการคำนวณเพียงเล็กน้อย ทำให้ถูกจำกัดวัดได้เฉพาะลักษณะที่วัดง่ายเท่านั้น ในขณะที่การวิเคราะห์ในโดเมนความถี่จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพมากกว่า แต่การสุ่มสัญญาณเสียงพูดต้องมีความเที่ยงตรงสูง

3.1 การวิเคราะห์เสียงพูดในช่วงสั้น (Short time speech analysis)

สัญญาณเสียงพูดเป็นสัญญาณที่เปลี่ยนแปลงไปตามเวลาโดยเกิดในลักษณะแบบสุ่ม (Random) แต่ก็ขึ้นกับการควบคุมเสียงของผู้พูดด้วย เพราะเสียงที่เปล่งออกมาในช่วงเวลาหนึ่งนั้น จะขึ้นอยู่กับรูปร่างของท่อกำธรเสียง (Vocal tract) และลักษณะการสั่นของเส้นเสียง (Vocal cord) สัญญาณที่เป็นเสียงพูดจึงเป็นคาบเวลาชั่วขณะ หมายความว่า สัญญาณเสียงพูดมีคาบเวลาคงที่ในระยะเวลาอันสั้น และมีการเปลี่ยนแปลงในระหว่างเวลานั้น ถ้าเราพูดช้ามากๆเสียงที่พูดอาจมีการเปลี่ยนแปลงอยู่ในช่วง 200ms ก็ได้ แต่ในการพูดโดยทั่วไปลักษณะการเปลี่ยนแปลงของเสียงพูดอยู่ในช่วงค่าเฉลี่ยประมาณ 80ms ในการวิเคราะห์เราจะใช้ window วิเคราะห์สัญญาณเป็นช่วงๆ หรือเรียกว่าการวิเคราะห์เฟรม (Frame) ช่วงเวลาของวินโดว์มีค่าไม่แน่นอน ในบางครั้งอาจจะใช้ถึง 100 ms แต่ในบางครั้งเมื่อนำมาวิเคราะห์ สัญญาณเสียงที่เร็วมากและไม่มีช่วงหยุดเลย เราอาจจะต้องใช้เวลาของวินโดว์ค่าถึง 5-10ms

3.1.1 การใช้วินโดว์ (windowing)

รูปแบบของการเฉลี่ย โดยปกติมีวิธีหลายอย่างเพื่อให้ได้เส้นของพารามิเตอร์ที่เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้องทางเลือกทางหนึ่งคือ

1.วินด์โดว์จะต้องมีขนาดเล็กพอๆกับคุณลักษณะของคำพูด

2.วินด์โดว์จะต้องยาวพอที่จะใช้ในการคำนวณหาพารามิเตอร์

3.วินด์โดว์ที่ดีจะต้องไม่สั้นไปจนข้ามบางช่องคำพูด การวิเคราะห์จะทำเป็นคาบๆไปซ้ำกันตลอดที่ได้สัญญาณ

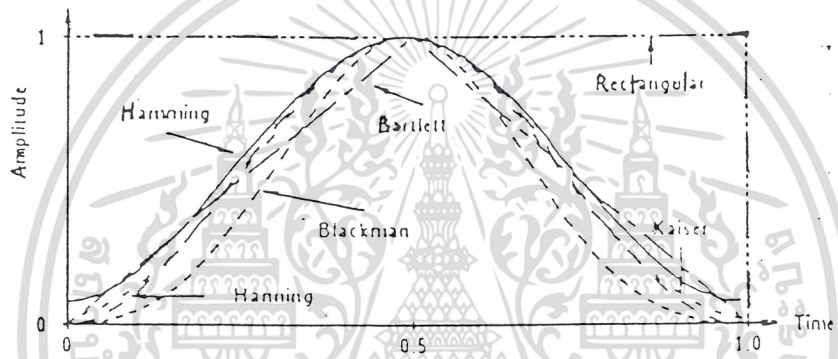
เงื่อนไขท้ายนี้จะขึ้นอยู่กับ frame rate มากกว่าขนาดของวินโดว์โดยปกติ frame rate ประมาณสองเท่าของความถี่เพื่อให้สัญญาณทับกัน 50% การทำบนวินโดว์เป็นการคูณสัญญาณเสียงด้วยวินด์โดว์ที่มีช่วงเวลาจำกัด w(n) ซึ่งกลุ่มของสัญญาณที่สุ่มมาจะถูกให้น้ำหนักโดยวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วินโดว์ที่ใช้ในความจริงเป็นดังรูป 4.1 สมมติว่าเสียงพูดมีค่าคงที่ในช่วง 10 ms เราต้องใช้วินโดว์ในช่วงเวลา 20 ms โดยให้จุดตรงกลางเท่ากับ 10 ms ถ่วงน้ำหนักมากกว่าจุดต้นและจุดปลาย เพราะรูปร่างของวินโดว์มีผลต่อเอาท์พุทต่อพารามิเตอร์ของเสียงพูด ผลของพารามิเตอร์อาจเปลี่ยนแปลงอย่างมากเมื่อใช้ w(n) ต่างกัน

$w(n)=r(n)=1$ เมื่อ $0 \le n \le N-1$ 3.1

$w_H(n) = 0.54-0.46\cos(2\pi n/(N-1))$ เมื่อ $0 \le n \le N-1$ 3.2



รูปที่ 3.1 รูปแบบของวินโดว์ที่มีช่วงเวลาเท่ากับหนึ่ง

การทำให้ขอบของวินโดว์ลาดลง โดยการวิเคราะห์เฟรมในการเลื่อนตลอดแนวความยาวของสัญญาณทั้งหมด ไม่มีผลต่อพารามิเตอร์ของเสียงพูด

3.2 ค่าพารามิเตอร์ในโดเมนความถี่

ค่าพารามิเตอร์ที่สำคัญจำนวนมากถูกพบในโดเมนความถี่ เพราะท่อกำธรเสียงให้กำเนิดสัญญาณที่สามารถวิเคราะห์เป็นความถี่ได้ง่ายกว่าในโดเมนเวลาโดยตรง เสียงพูดของคนคนเดียวในแต่ละครั้ง เมื่อพิจารณาในโดเมนเวลาจะเห็นผลของความแตกต่างของสัญญาณแต่เมื่อพิจารณาในโดเมนความถี่ สัญญาณที่ได้มีความคล้ายคลึงกันมาก

ระบบการได้ยินของมนุษย์จะตอบสนองต่อรูปร่างของสัญญาณเสียง(หรือขนาดที่กระจายอยู่ในโดเมนความถี่) มากกว่าที่เฟสของสัญญาณในโดเมนเวลาด้วยเหตุผลเหล่านี้การวิเคราะห์ความถี่จึงถูกใช้ในการดึงค่าพารามิเตอร์ที่สำคัญจากสัญญาณเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การวิเคราะห์ฟูริเยร์เบื้องต้น (Short time fourier analysis)

การวิเคราะห์ฟูริเยร์ช่วงสั้นเป็นเทคนิกการหาความถี่ที่ใช้กันมานานแล้ว การวิเคราะห์ฟูริเยร์ให้ตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชั่นความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งตลอดเวลา ดังนั้นจึงจำเป็นต้องทำการวิเคราะห์ในช่วงสั้นๆ โดยใช้วินด์โดว์

การแปลงฟูริเยร์ช่วงสั้นมีสมการว่า

$$X_n(e^{-jw}) = \sum_{m=-\infty}^{\infty} x(m)[\exp(-jwm)w(n-m)] \qquad 3.3$$

ในการคำนวนต้องใช้ DFT แทนการแปลงฟูเรียร์แบบต่อเนื่อง โดยใช้ฟังก์ชั่นวินด์โดว์ ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว (N คือ ช่วงเวลาหรือขนาดของวินด์โดว์ที่ใช้ในการแปลง DFT) ข่าวสารต่างๆใน $X_n(e^{-jw})$ จะไม่สูญหายไปจากข้อมูลเดิมถ้าการแปลงนั้นสุ่มมาด้วยความถี่สูงเพียงพอ (คือ ช่วงระยะห่างระหว่าง N) และวินโดว์ w(n) ไม่มีจุดสุ่มที่เป็นศูนย์ตลอดช่วง N ตัวแปร N เป็นตัวแปรที่จะต้องระวังมากเป็นพิเศษในการวิเคราะห์ความถี่ช่วงสั้น ถ้าค่าของ Nต่ำจะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำเฉพาะในช่วงสั้นๆเท่านั้น ในทางตรงกันข้ามถ้า N มีขนาดใหญ่ จะให้ผลของความละเอียดเวลาที่แย่ในโดเมนเวลาแต่จะทำให้โดเมนความถี่มีความละเอียดสูงกว่า

3.2.2 การแปลงฟาสท์ฟูริเยร์ (Fast Fourier Transform)

โดยทั่วไปการแปลงฟูริเยร์ (Discrete Fourier Transform) เป็นการคำนวณที่ใช้เวลาค่อนข้างมาก ลำดับขั้นตอนในการคำนวณ DFT ให้เร็วขึ้น เรียกว่า ฟาสท์ฟูริเยร์ (Fast Fourier Transform หรือ FFT) โดยการแปลง FFT แบ่งได้เป็น 2 ชนิดใหญ่ๆคือ ชนิดลดทอนทางเวลา (Decimation In Time หรือ DIT) และชนิดลดทอนทางความถี่ (Decimation In Frequency หรือ DIF) สำหรับในส่วนนี้จะแสดงเฉพาะชนิดลดทอนทางเวลาซึ่งเกี่ยวกับในโครงงานนี้เท่านั้น

3.2.2.1 ขั้นตอนวิธีลดทอนทางเวลา (Decimation In Time หรือ DIT)

วิธีนี้เป็นการจัดแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา X(n) ขนาด N จุดออกเป็น 2 ลำดับสัญญาณขนาด N/2 จุดเท่ากัน คือ ลำดับคู่และลำดับคี่ โดยที่ลำดับคู่เกิดจากการเอาลำดับในตำแหน่งคู่มาเรียงกัน ที่เหลือเป็นลำดับคี่ ดังนั้นจะได้

$$x_e(m) = x(2n) \quad ; m=0,1,2,\ldots,(N/2)-1$$
$$x_o(m) = x(2n+1) \quad ; m=0,1,2,\ldots,(N/2)-1 \qquad 3.4$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ WN เท่ากับ exp(-2j/N) จะทำให้การคำนวณ DFT ของลำดับ x(n) ที่ยาว N จุด สามารถเขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N/2-1} x_e(2m) W_N^{2km} + \sum_{m=0}^{N/2-1} x(2m+1) W_N^{(2m+1)k} \qquad 3.5$$

โดยที่

$$W_N^2 = \{\exp[j2\pi / N]^2\} = \exp[j2\pi / N/2] = W_{N/2}$$

ซึ่ง $W_{N/2}$ เป็นค่า W ของลำดับความยาว N/2 จุด สมการ 6.5 สามารถเขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N/2-1} x_e(m) W_{N/2}^{km} + \sum_{m=0}^{N/2-1} x_o(m) W_{N/2}^{km} \qquad 3.6$$

การนำผลการแปลง DFT ขนาด 2 จุดจำนวน N/2 ภาคมารวมกันเพื่อให้เป็นการคำนวณ DFT ขนาด N จุด จะต้องมีหลักเกณฑ์ที่ถูกต้องด้วย จากสมการ6.6 ถ้าเขียนให้อยู่ในช่วง $0 \le k \le N/2-1$ สามารถเขียนใหม่ได้เป็น

$$X(k) = X_e(k) + (W_N^k) X_o(k) \qquad 0 \le k \le N/2-1$$

$$= X_e(k - N/2) + (W_N^k) X_o(k - N/2) \qquad 0 \le k \le N/2-1 \qquad 3.7$$

เทอม W_N^k เรียกว่า ตัวประกอบหมุน (Twiddle Factor) ซึ่งใช้ร่วมกันกับ DFT ขนาด 2 จุดหรือขนาด N/2 จุด ในการนำมาประกอบเป็น DFT ขนาด N จุดได้เหมือนเดิม และจากความสัมพันธ์

$$(W)_N^{k-N/2} = -(W)_N^k$$

จะได้

$$X(k) = X_1(k) + (W)_N^k X_2(k) \qquad ;0 \le k \le N/2-1 \qquad 3.8$$

$$= X_1(k - N/2) + (W)_N^{k-N/2} X_2(k - N/2) \qquad ;N/2 \le k \le N/2-1$$

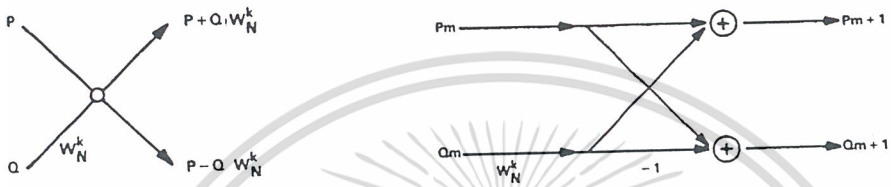
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากสมการนี้สามารถนำไปใช้สร้างหน่วยคำนวณที่ เรียกว่า หน่วยผีเสื้อ (Butterfly Unit) โดยมีข้อมูลเข้าคือ A และ B และข้อมูลออก คือ X และ Y เป็น

$$X = A + (W)_N^k B$$

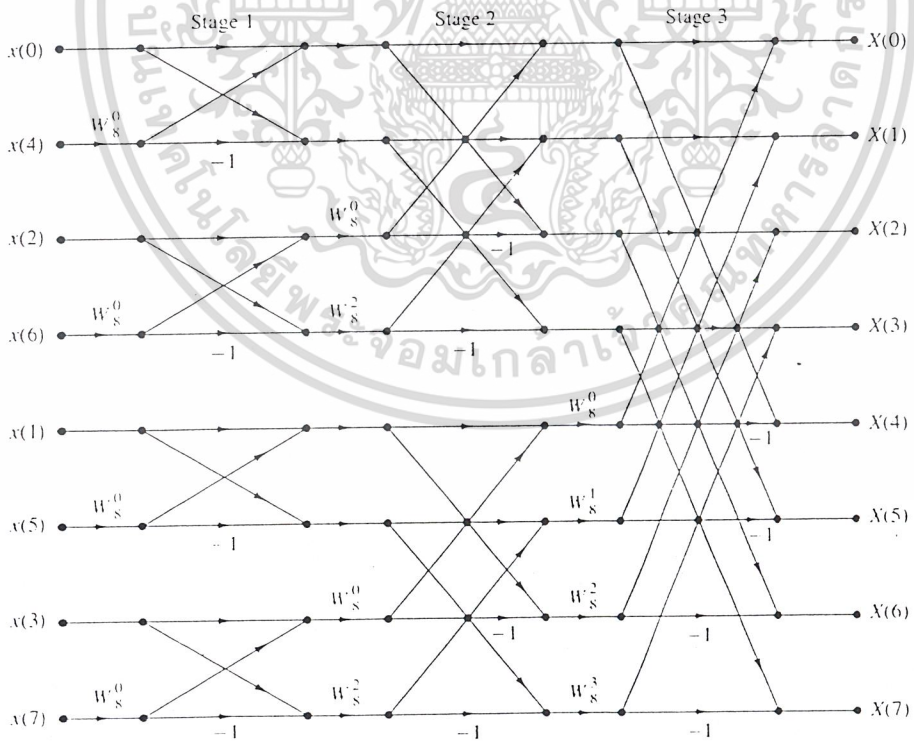
$$X = A + (W)_N^k B \qquad 3.9$$

ซึ่งสามารถเขียนอธิบายแทนด้วยโฟล์ชาร์ตดังแสดงในรูป4.2



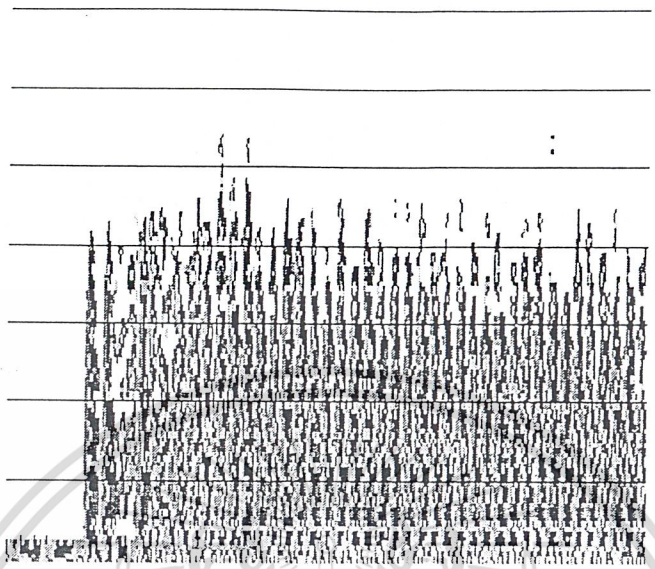
รูปที่3.2 แสดงหน่วยผีเสื้อของการคำนวณตามขั้นตอนวิธีลดทอนทางเวลา

สำหรับตัวอย่างการคำนวณ DFT โดยใช้ FFT แสดงดังรูป 3.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

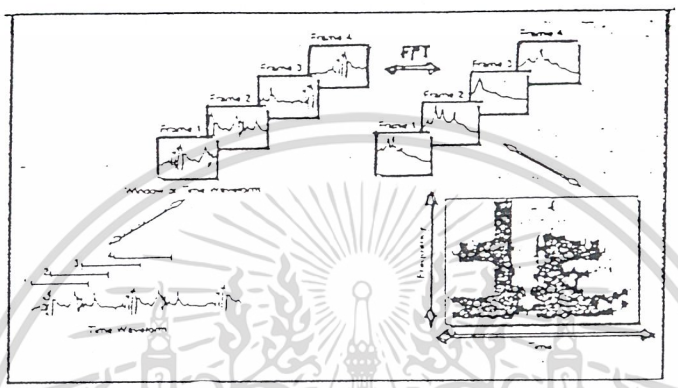
รูปที่3.3 แสดงวิธีการของ FFT แบบลดทอนทางเวลา (DIT) สำหรับข้อมูลขนาดจุด



รูปที่3.4 การแสดงผลสเปกโตรแกรม

เมื่อสัญญาณอินพุตที่แซมปลิ้งเข้ามาจะถูกแบ่งออกเป็นส่วนๆ ที่ซ้อนกันโดยใช้หน้าต่าง (window) ครอบ ในที่นี้ใช้หน้าต่างฮานน์ แล้วคำนวณการแปลงฟูริเยร์ (Discrete-Time Fourier Transform) ของแต่ละส่วนโดยใช้ FFT เพื่อหารายละเอียดเกี่ยวกับความถี่ของสัญญาณนั้นในแต่ละช่วงหน้าต่าง จากนั้นนำผลที่คำนวณได้มาแสดงผลบนกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.5 การคำนวณหาสเปกโตรแกรมของสัญญาณ

ข้อมูลที่นำมาแสดงจะใช้ข้อมูลเพียงครึ่งหนึ่งของจำนวนจุดทั้งหมดที่ใช้ในการคำนวณ FFT แต่มีข้อจำกัดคือ ความกว้างของหน้าต่างเพราะถ้าหน้าต่างมีความกว้างมากจะทำให้สเปกตรัมในแกนความถี่มีความละเอียดมากแต่ขณะเดียวกันก็ให้ความละเอียดต่ำในแกนเวลา และเพื่อไม่ให้มีการสูญสียข้อมูลหน้าต่างจะต้องมีการซ้อนกัน โดยถ้าต้องการความถูกต้องมากต้องให้มีการซ้อนกันมากขึ้น แต่ในทางปฏิบัติความกว้างของหน้าต่างที่ซ้อนกันจะเป็นครึ่งหนึ่งของความกว้างหน้าต่างเพื่อให้สามารถคำนวณ FFT ได้ทัน

3.3สหสัมพันธ์ (Correlation)

ค่าระดับความสัมพันธ์ของข้อมูล 2 ชุด เรียกว่า สัมประสิทธิ์ของสหสัมพันธ์ (Correlative coefficient; r) สามารถหาได้จากเทอมของ Z-Score ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$r_{xy} = \sum_{i=1}^{N} \frac{Z_{x_i} Z_{y_i}}{N} \qquad 3.10$$

เมื่อ r_{xy} คือ สัมประสิทธิ์ของสหสัมพันธ์ของข้อมูล x_i และ y_i

x_i คือ ข้อมูลทางแกนความถี่ของสเปกโตรแกรมเสียงชุดที่ 1 (ทดสอบ)

y_i คือ ข้อมูลในแกนความถี่ของสเปกโตรแกรมเสียงชุดที่ 2 (อ้างอิง)

N คือ จำนวนข้อมูลทั้งหมดที่มี x_i และ y_i เป็นสมาชิก

Z_{x_i} และ Z_{y_i} คือค่า Z-Score เป็นระยะห่างระหว่างค่าข้อมูล

โดย $Z_{x_i} = \frac{x_i - \bar{X}}{S_x}$ 3.11

$$Z_{y_i} = \frac{y_i - \bar{Y}}{S_y} \qquad 3.12$$

$$S_x = \sqrt{\sum \frac{(x_i - \bar{X})^2}{N}} \qquad 3.13$$

$$S_x = \sqrt{\sum \frac{(x_i - \bar{X})^2}{N}} \qquad 3.14$$

จากสมการ 3.10 ค่า r_{xy} จะมีค่าอยู่ระหว่าง -1 ถึง +1 และ

$$0 \leq |r_{xy}| \leq 1$$

พอตีความหมายได้ดังนี้คือ ถ้าค่า $|r_{xy}| = 1$ หมายถึง ข้อมูลทั้ง 2 ชุด มีความสัมพันธ์กันทุกจุดข้อมูล ซึ่งในทางปฏิบัติมีโอกาสน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

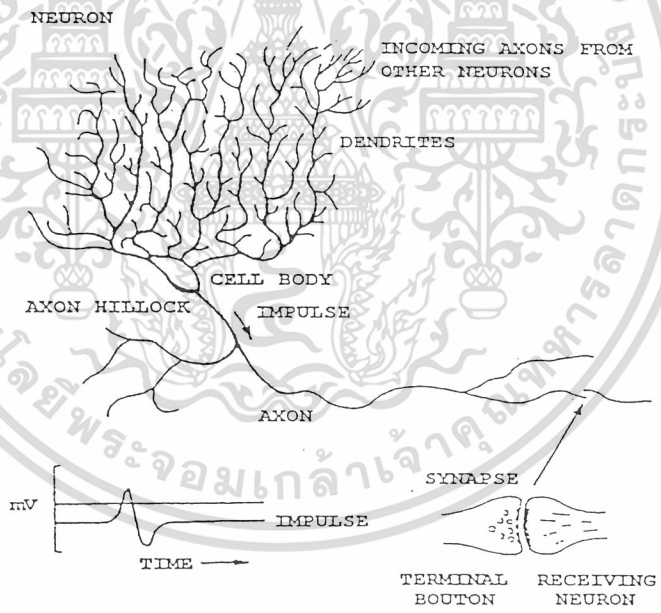
โครงข่ายประสาทเทียม(NEURAL NETWORK)

โครงข่ายประสาท หมายถึงโครงข่ายใยประสาทที่เชื่อมต่อกันระหว่างเซลล์ประสาทจำนวนมากมายมหาศาล บรรจุอยู่ในสมองมนุษย์ มีความสามารถในการประมวลผลสูง สมองชีวภาพเป็นศูนย์กลางควบคุมกิจกรรมการดำเนินชีวิตที่สามารถเรียนรู้และปรับตัวเองได้ตลอดเวลา

การวิจัยเพื่อพัฒนาให้คอมพิวเตอร์มีความสามารถตัดสินใจได้วิธีการหนึ่ง จึงหันมาศึกษาโครงข่ายประสาทเทียม โดยการจำลองลักษณะการทำงานของใยประสาทชีวภาพขึ้นเป็นโมเดลทางคณิตศาสตร์

4.1 โครงข่ายประสาทชีวภาพ

ระบบการคิดของมนุษย์ มีโครงสร้างพื้นฐานจากเซลล์สมอง ที่เรียกว่านิวรอล(Neurons)เรียงเป็นชั้นๆ จำนวนมหาศาล ประมาณหมื่นล้าน(10)นิวรอล เชื่อมโยงถึงกันประมาณพันล้านล้านจุด แต่ละนิวรอลจะมีคุณลักษณะต่างกันไป แต่จะมีการรูปแบบทำงานคล้ายๆกันคือ รับค่าเข้ามา ประมวลผล และส่งสัญญาณไฟฟ้าเคมีออกไป



รูป4.1: ตัวอย่างเซลล์ประสาทชีวภาพ

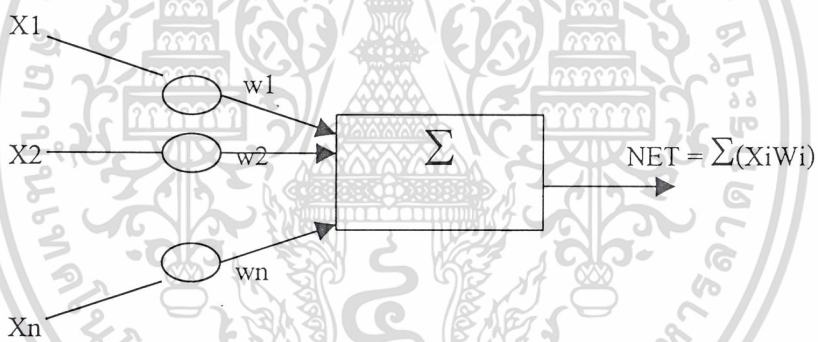
จากภาพแขนงที่ขยายออกไปยังเซลล์อื่นๆเพื่อรับสัญญาณ เรียกว่าเดนไดรท์(Dendrites) จะมีจุดเชื่อมต่อกับเซลล์ประสาทอื่นๆเรียกว่าไซแนปส์ (synapse) และส่วนที่ใช้ส่งสัญญาณออกไปเรียกว่า แอกซอน(Axon) แต่ละนิวรอลจะมีคุณสมบัติในการเพิ่มขยาย หรือลดทอนความเข้มสัญญาณที่เข้ามายังเดนไดรท์ของเซลล์ อาจกระตุ้นหรือยับยั้งตัวเซลล์ก็ได้ เนื่องจากเซลล์ประสาทหนึ่งเซลล์มีเดนไดรท์จำนวนมาก ดังนั้นสัญญาณกระตุ้นเดนไดรท์ที่รับเข้ามาจากเซลล์อื่นๆจะถูกนำมารวมกันที่ตัวเซลล์ประสาท ที่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซลล์ประสาทจะมีค่าเทรสโฮล (Threshold) ค่าหนึ่ง หากผลรวมของสัญญาณไฟฟ้าเคมี (Electrochemical) มีค่ามากกว่าค่าเทรสโฮล เซลล์ประสาทจะส่งสัญญาณค่าหนึ่งผ่านทางแอกซอนไปยังนิวรอลอื่นๆต่อไป การจัดเรียงเป็นชั้น (Layer)และลักษณะการเชื่อมโยงระหว่างนิวรอลในสมองนั้นมีการจัดเรียงที่ซับซ้อน สอดคล้องกับหน้าที่การทำงานเฉพาะส่วน มีการเจริญเติบโต เรียนรู้สัมพันธ์กับสิ่งแวดล้อมตลอดเวลา จึงเป็นการยากที่จะสามารถสร้างโมดูลเลียนแบบการทำงานของสมองชีวภาพได้ทั้งหมด ปัจจุบันยังคงทำได้เพียงการจำลองเลียนแบบการทำงานเฉพาะส่วนบางส่วน ของโครงข่ายประสาทเทียม มาใช้เฉพาะกับงานใดงานหนึ่ง และมีขอบเขตจำกัดการใช้งานด้วย

4.2 โครงข่ายประสาทเทียม (Artificial Neural Network)

การออกแบบสร้างประสาทเทียมมีสมมติฐานจากโครงข่ายประสาทชีวภาพ ซึ่งมีความนำสัญญาณไฟฟ้าเคมีต่างกัน โมดูลโครงข่ายประสาทเทียมจึงต้องมีการถ่วงน้ำหนักก่อนนำไปใช้งานเรียกว่าไซแนปติกเวจท์ ปริมาณข้อมูลจะถูกนำมารวมกัน และตัดสินใจด้วยระดับความสนใจของนิวรอล (Activation Lavel) แล้วส่งเอาท์พุท(Output) ออกไปยังนิวรอลอื่นๆ



รูปที่ 4.2 : ไดอะแกรมของนิวรอล 1 หน่วย

จากภาพ เป็นไดอะแกรมที่จำลองจากแนวความคิดของเซลล์สมองทางชีวภาพ สัญญาณอินพุท คือ X1,X2,...,Xn จะถูกป้อนเข้ามายังนิวรอล เปรียบได้กับสัญญาณไฟฟ้าเคมีเข้ามายังไซแนปส์ของเซลล์ประสาท ค่าอินพุทเหล่านี้จะคูณด้วยค่าถ่วงน้ำหนัก(weight)ที่มีค่าตั้งแต่ 0.0-1.0 ผลรวมของสัญญาณทั้งหมดจะส่งออกมาโดย

$$NET = X1W1+X2W2+ \ldots +XnWn$$

$$= \Sigma(XiWi)$$

จากนั้นจะถูกตัดสินใจโดยหน่วยเซลล์ด้วยฟังก์ชันการตัดสินใจ (Activation Function)และได้ค่าเอาท์พุทออกมา

$$OUT = F(NET)$$

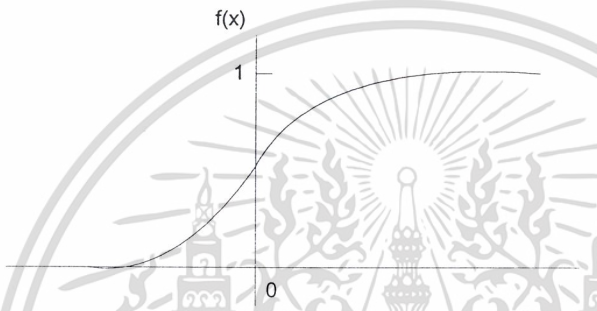
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันตัดสินใจอาจเป็น hard linear function โดย

OUT = 1 ถ้า NET > T

= 0 ถ้า NET มีค่าเป็นกรณีอื่น

T เป็นค่าคงที่เทรสโฮล(Threshold) หรืออาจเป็นฟังก์ชันอื่นๆที่เลียนแบบคุณสมบัติไม่คงที่ของเซลล์ประสาทได้ดีกว่า และใช้เป็นฟังก์ชันให้กับโครงข่ายทั่วไปได้ ฟังก์ชันตัดสินใจที่นิยมใช้กันทั่วไปคือสแควชชิ่ง ฟังก์ชัน(Squashing function หรือซิกมอยด์ ฟังก์ชัน(Sigmoid function) ซึ่งมีรูปร่างคล้ายตัว "S" และมีสมการทางคณิตศาสตร์ดังนี้



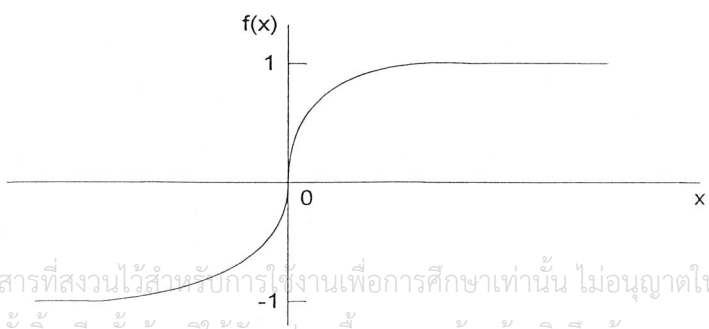
F(x) = 1/[1+exp(-x)]

รูปที่ 4.3 : ซิกมอยด์ ฟังก์ชัน(Sigmoid function)

การใช้ซิกมอยด์ ฟังก์ชัน จะทำให้เทรสโฮลฟังก์ชันมีลักษณะ non linear function ทำให้ได้ค่าอินพุทที่ไวต่อสัญญาณขนาดเล็กๆ และเฉลี่ยต่อสัญญาณแรงๆ (คือสัญญาณไปทางบวกเพียงเล็กน้อยก็จะมีค่าเอาท์พุทใกล้เคียง "1" และสัญญาณเป็นลบค่าต่ำๆ ใกล้เคียง "0" ขณะเดียวกัน สัญญาณบวกขนาดแรงๆก็ยังคงให้ค่าเอาท์พุทใกล้เคียง "1" และสัญญาณลบค่ามากๆก็ยังคงให้เอาท์พุทใกล้เคียงศูนย์ได้ คือมีคุณสมบัติ non linear gainนั่นเอง ซึ่ง คลอสส์เบอร์ก(Grossberg,1973)พบว่าสามารถแก้ปัญหา Noise saturation dilemma ได้ และทำให้นิวรอลสามารถทำงานได้กว้างขวางขึ้น

นอกจากนี้ยังมีฟังก์ชันอื่นๆอีกคือ ไฮเปอร์โบลิก ฟังก์ชัน(Hyperbolic function) ซึ่งมีลักษณะคล้ายกับ logistic function คือ

OUT = tanh(X)



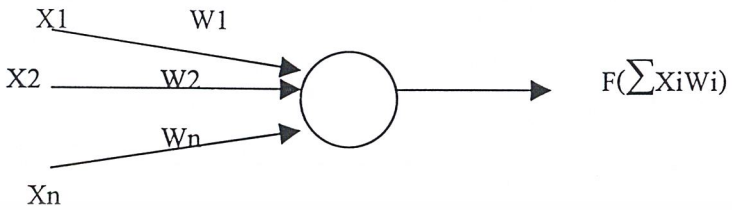
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.4 : hyperbolic tangent function

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Artificial Neural Network)

ซึ่งจะสามารถแก้ปัญหาฟังก์ชันเชิงเส้น(linear function)ได้เท่านั้น



รูป 4.5 : โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Neural Network)

โครงข่ายประสาทแบบชั้นเดียวที่ไม่มีการไหลย้อนกลับของค่าสัญญาณเอาท์พุท จะมีความสามารถในขอบเขตจำกัด คือใช้ตัดสินใจกับปัญหาที่เป็นฟังก์ชันเชิงเส้นเท่านั้น

รูป 4.6 Linear Separable (สามารถแบ่งได้เชิงเส้น)

รูป 4.7 Non-Linear Separable (ไม่สามารถแบ่งได้เชิงเส้น)

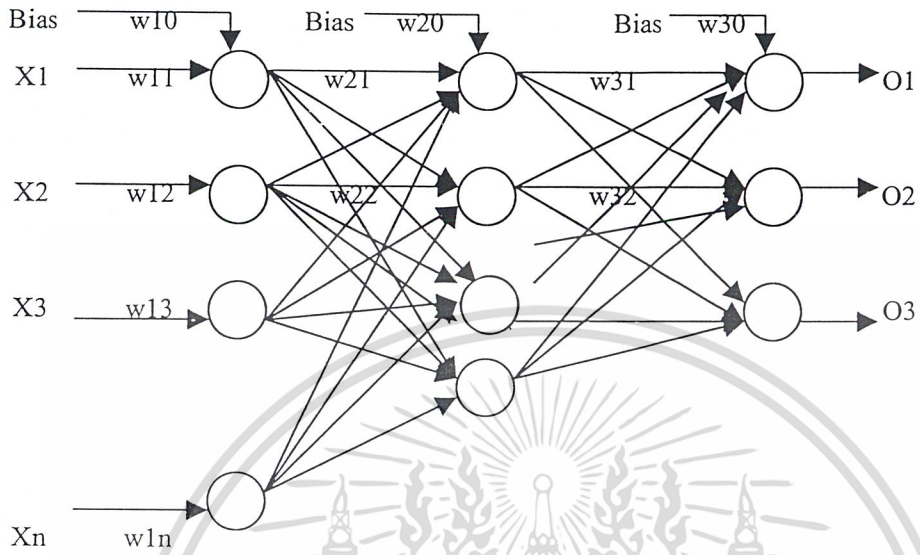
โครงข่ายประสาทเทียมแบบหลายชั้น(Multilayer Artifitial Neural Network)

ซึ่งประกอบด้วยอินพุทเลเยอร์(Input Layer),เอาท์พุทเลเยอร์(Output Layer) และชั้นซ่อน(Hidden Layer) จำนวนตั้งแต่ 1 ชั้นขึ้นไป ทำให้ความสามารถของโครงข่ายดีขึ้น คือสามารถแก้ปัญหาที่ไม่เป็นเชิงเส้นได้ ปัญหาที่มักเกิดขึ้นจากการออกแบบโครงข่ายประสาทแบบหลายชั้นจะเป็นในแง่ของการกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

34069

ขนาดจำนวนหน่วยในแต่ละชั้น จำนวนชั้นซ่อนที่เหมาะสม และลักษณะการเชื่อมของแต่ละหน่วย.แต่ละชั้น



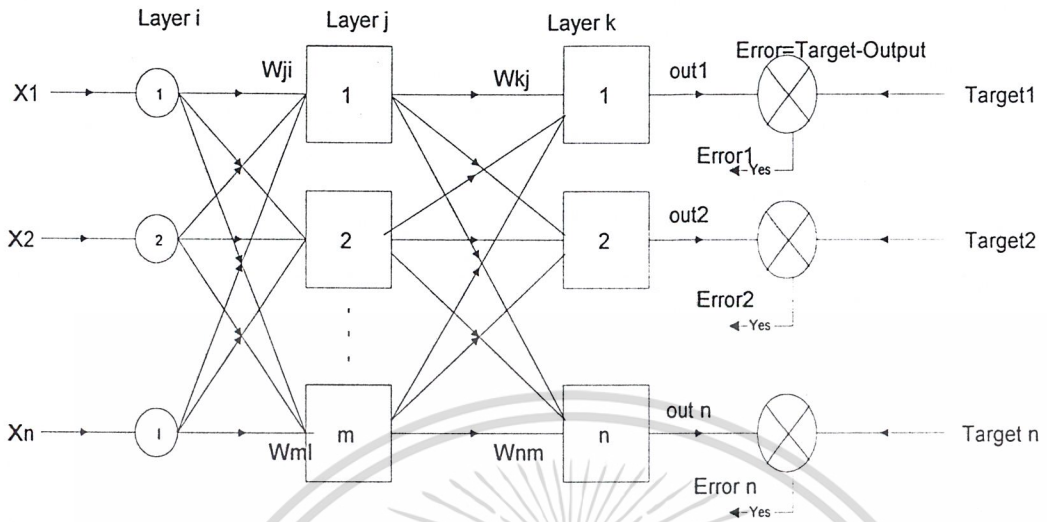
รูป 4.8 โครงข่ายประสาทเทียมแบบหลายชั้น(Multilayer Artifitial Neural Network)

การฝึกสอนให้กับโครงข่าย (Training Algorithm)

เทรนนิ่งอัลกอริทึมถูกจัดเป็น 2 ประเภท คือ แบบควบคุม(Supervise training) และ แบบอิสระ (Unsupervise) โดยเทรนนิ่งแบบควบคุมต้องการข้อมูล 2 ชุดคือ อินพุท กับ ชุดเป้าหมาย เรียกเทรนนิ่งแพร์(Training pairs) เช่น การเรียนรู้แบบแพร่ย้อนกลับ ส่วนการเทรนนิ่งแบบอิสระสร้างขึ้นโดยเปรียบเทียบการทำงานของสมองที่ไม่จำเป็นต้องมีผู้มาคอยคิดค่าเป้าหมายให้ก่อน โครงข่ายจะรับเพียงค่าอินพุทเข้าไปเท่านั้น เทรนนิ่งอัลกอริทึมจะเปลี่ยนแปลงค่าถ่วงน้ำหนักของโครงข่าย เพื่อสร้างเอาท์พุทที่มั่นคง และเมื่อค่าอินพุทมีการเปลี่ยนแปลงไปเพียงเล็กน้อย โครงข่ายจะยังสามารถชี้ได้ว่าเป็นข้อมูลเดิม เนื่องจากเอาท์พุทของโครงข่ายไม่มีการกำหนดมาก่อน ส่วนใหญ่จะถูกแปรรูปให้เข้าใจได้ จึงไม่สามารถใช้ตัดสินใจปัญหาที่มีความยากได้ มักใช้ในงานง่ายๆเช่น การเปรียบเทียบเอกลักษณ์, รูปภาพ,รูปแบบที่สัมพันธ์กันระหว่างอินพุท-เอาท์พุท

ขั้นตอนการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.9 ไดอะแกรมของ Backpropagation Neural Network แบบ 2 ชั้น

การฝึกสอนให้กับโครงข่ายประสาทแบบหลายชั้นใช้การเรียนรู้แบบแพร่ย้อนกลับ โดยโครงข่ายจะรับค่าอินพุท X และค่าเป้าหมาย(Target) ไว้ เมื่อคำนวณค่าเอาท์พุทในโครงข่ายแล้วจะเปรียบเทียบค่าผิดพลาดของเอาท์พุทที่คำนวณได้กับค่าเป้าหมาย

$$\text{Error} = \text{Target-Output}$$

ค่าผิดพลาดนี้จะนำไปใช้ปรับค่าถ่วงน้ำหนักของโครงข่ายเพื่อให้สอดคล้องกับค่าเป้าหมาย การเรียนรู้จะทำซ้ำไปเรื่อยๆจนโครงข่ายสามารถแยกแยะข้อมูลได้

ขั้นตอนการฝึกสอนสามารถสรุปเป็นขั้นๆได้ดังต่อไปนี้

1. สร้างโครงข่ายประสาทที่มีจำนวนอินพุท และเอาท์พุทเท่าจำนวนอินพุท และเอาท์พุทของปัญหา ส่วนจำนวนชั้นซ่อนอาจเริ่มที่ค่ามากพอสมควรก่อน
2. ให้ค่าเริ่มต้นแก่ค่าถ่วงน้ำหนักมุกเส้นในโครงข่าย โดยควรมีค่าในช่วง –1 ถึง 1
3. คำนวณค่า $\sum XiWi = Ui$ สำหรับแต่ละนิวรอล j ของชั้นซ่อน
4. คำนวนค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอล j ของชั้นซ่อน

   $Yj = f(Ui)$

   โดยฟังก์ชันที่นิยมใช้คือ sigmoid function $f(x) = 1/[1+\exp(-x)]$
5. คำนวณค่า $\sum YiWi = Vi$ สำหรับแต่ละนิวรอล k ของชั้นเอาท์พุท
6. คำนวนค่ากระตุ้น (Activation) ของแต่ละนิวรอล k

   $OUT = Zk = f(Vi)$
7. นำค่า Zk มาเปรียบเทียบกับเป้าหมายถ้าค่าผิดพลาดที่คำนวณได้น้อยกว่าระดับที่กำหนดไว้ก็จบการเรียนรู้ มิฉะนั้นก็ทำต่อไป

ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. คำนวณค่าปรับน้ำหนักสำหรับแต่ละค่าถ่วงน้ำหนัก

8.1 สำหรับเส้นเชื่อมชั้นซ่อนกับชั้นเอาท์พุท

$$\Delta W_{jk} = \alpha \delta_k Y_j$$

โดยที่ $\delta_j = (\text{Target } k - Z_k) f'(V_k)$

8.2 สำหรับเส้นเชื่อมชั้นอินพุทกับชั้นซ่อน

$$\Delta W_{ij} = \alpha \delta_j X_i$$

โดยที่ $\delta_j = \sum \delta_k W_{jk} f'(U_j)$

9. ปรับค่าน้ำหนักเส้นเชื่อมจากนิวรอล r ไปนิงรอล s จะได้

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

10. กลับไปทำข้อ 3

ค่า α : อัตราการเรียนรู้(Learning rate) มีค่าในช่วง 0 ถึง 1

การปรับความเร็วในการเรียนรู้อาจใช้การปรับค่า Gain ของ sigmoid function ในส่วนดีกรีของ exponential การปรับอัตราการเรียนรู้ ขนาดจำนวนหน่วยในชั้นซ่อน และการเพิ่มโมเมนตัมแฟกเตอร์เพื่อคำนวณทิศทางการปรับค่าถ่วงน้ำหนัก นอกจากนี้ยังเลือกใช้อัตราการเรียนรู้เป็นฟังก์ชันที่มีความเหมาะสมได้อีกด้วย

ฟังก์ชัน sigmoid เป็นที่นิยมในการใช้เทรนนิ่งเนื่องจากมีความใกล้เคียงกับฟังก์ชันขั้นบันไดและสามารถคำนวณค่าอนุพันธ์ได้ง่ายอีกด้วย โดยอยู่ในรูปนิพจน์ของตัวฟังก์ชันเอง

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผล วิจารณ์ และแนวทางปรับปรุง

<u>จากการทดสอบการทำงานของโปรแกรมโครงข่ายประสาทเทียม (โดยใช้ฟังก์ชันอินพุทที่สร้างขึ้นเอง) พบว่า</u>

1. รูปแบบข้อมูลที่ใช้ในการฝึกสอนที่มาจากส่วน ปรี-โปรเซสซิ่งมีผลสัมพันธ์กับอัตราการเรียนรู้อย่างมาก ถ้าข้อมูลที่ได้จากส่วน ปรี-โปรเซสซิ่งเป็นฟังก์ชันที่มีความแตกต่างกันมากเท่าไร การเรียนรู้ของโครงข่ายจะทำได้เร็วขึ้นเท่านั้น เมื่อทดลองใช้ข้อมูลที่เป็นฟังก์ชันเดียวกัน แต่มีการเลื่อนค่าตามแกน x หรือแกน y การเรียนรู้จะใช้เวลานานมาก และยังมีการตัดสินใจพลาดได้สูงอีกด้วย ดังนั้นการออกแบบส่วนปรี-โปรเซสซิ่งจึงมีความสำคัญต่อโครงข่ายอย่างมาก โดยควรสามารถดึงเอาลักษณะที่แตกต่างกันของแต่ละเสียงออกมาให้ได้มากที่สุด ซึ่งในโครงงานนี้เลือกใช้คุณสมบัติเสียงด้านค่าสหสัมพันธ์ของข้อมูลในโดเมนความถี่ เป็นคุณลักษณะในการตัดสินใจ

2. การใช้โมเมนตัมแฟกเตอร์กำหนดทิศทางการปรับค่าถ่วงน้ำหนักเพิ่มอัตราเร็วในการเรียนรู้ขึ้นอย่างมาก จากการทดลองกับข้อมูลขนาด 20 ค่าอินพุท จำนวน 2 ฟังก์ชัน(2 เป้าหมาย) เมื่อไม่ใช่โมเมนตัมแฟกเตอร์ ทำการเทรนนิ่งไปถึง 25000 ครั้งยังคงมีค่าความผิดพลาดมากกว่า 0.3 และการทดสอบ(Test) โครงข่ายก็พบว่าตัดสินใจผิดพลาดอยู่ เมื่อเพิ่มโมเมนตัน แฟกเตอร์ในการปรับค่าถ่วงน้ำหนักตามสมการ

$$\Delta W_i = \alpha \delta_{i-1} X_i + \Delta W_i(old) * \gamma \text{ (momentum factor)}$$

โครงข่ายใช้เวลาในการสอน (Train) เพียง 100 ครั้ง มีค่าผิดพลาด 0.001 และสามารถตัดสินใจได้ถูกต้องอีกด้วย

3. การเพิ่มขนาดชั้นซ่อน(Hidden Layer) จาก 4 หน่วย จนมีขนาด40 หน่วย (อินพุทเลเยอร์ขนาด 90 หน่วย) พบว่าช่วงแรกๆ การเพิ่มขนาด จะทำให้โครงข่ายตัดสินใจได้ และเร็วขึ้นจนถึงระดับหนึ่ง จากนั้นจะใช้เวลามากขึ้นแทน เพราะขนาดของชั้นซ่อนเกินพอสำหรับตัดสินปัญหาที่ความยากระดับนั้นๆ แล้ว การเพิ่มขนาดจะทำให้ต้องคำนวณปรับค่าจำนวนมากเกินความจำเป็นแทน และปัญหาต่างๆระดับความยากกันจะมีขนาดชั้นซ่อนที่เหมาะสมแตกต่างกัน

<u>เมื่อนำส่วนปรี-โปรเซสซิ่งมาใช้ร่วมกันอย่างเป็นกระบวนการกับส่วนการตัดสินใจ</u>

ขนาดของนิวรอลเน็ทเวิร์คที่เลือกใช้ เมื่อมีการวิเคราะห์คุณสมบัติของข้อมูลโดยผ่านกระบวนการ ปรีโปรเซสซิ่งมาก่อน ทำให้ใช้นิวรอลเน็ทเวิร์คที่มีขนาดเล็กลงได้ ทำให้การประมวลผลเร็วขึ้น

นิวรอลเน็ทเวิร์คที่ใช้ประกอบด้วย 3 ชั้นต่อไปนี้

: ชั้นอินพุทเลเยอร์ (Input Layer) จำนวน 10 หน่วย

: ชั้นซ่อน (Hidden Layer) จำนวน 6 หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: ชั้นเอาท์พุทเลเยอร์ (Output Layer) จำนวน 4 หน่วย เท่ากับจำนวนบิทเอาท์พุทแสดงผล ที่เลือกใช้

ขั้นแรกจะทำการฝึกสอนให้กับโครงข่ายก่อน จนสามารถรู้จำเสียงตัวเลข 0-9ได้ทุกตัวในขอบเขตค่าความผิดพลาดที่กำหนดไว้

ข้อมูลเสียงจะถูกอัดโดยใช้ Sound Blaster Card ซึ่งส่วนหนึ่งจะถูกนำไปใช้เป็น Library file ให้กับโปรแกรมหาค่าสหสัมพันธ์ (Correlation) 50 เสียง และใช้เป็นข้อมูลฝึกสอนให้กับโครงข่าย 10 เสียง และเสียงทดสอบอีก 50 เสียง

ส่วนปรีโปรเซสซิ่งจะประกอบด้วย

1. ส่วนบีบอัด-ขยายแกนเวลา (Normalize Time) จะปรับข้อมูลเสียงที่ผ่านการตัดเสียงรบกวน (Noise)และเสียงเงียบแล้ว ให้มีความยาวเท่ากันในแกนเวลา โดยทำการขยายข้อมูลเมื่อข้อมูลมีขนาดต่ำกว่ากำหนด และบีบอัดเข้าเมื่อมีความยาวมากกว่ากำหนด โดยในโครงงานนี้กำหนดค่าความยาวแกนเวลาอ้างอิงไว้ที่ 4096 ไบท์ และใช้การคำนวณปรับข้อมูลโดยเทียบบัญญัติไตรยางค์ จากนั้นทำการตัดข้อมูลออกเป็นเฟรม กำหนดไว้ที่ 5 เฟรม ข้อมูลแต่ละเฟรมจะเหลื่อม (Overlap) กันอยู่ 256 ไบท์ เพื่อเตรียมสำหรับส่วนต่อไป
2. ส่วนฟาสฟูเรียร์ทรานสฟอร์ม (Fast Fourier Transform) จะแปลงข้อมูลแกนเวลา ไปยังแกนความถี่ เพื่อดึงคุณสมบัติข้อมูลเสียงในแกนความถี่ออกมา
3. ส่วนคำนวณค่าสหสัมพันธ์ (Correlation) เนื่องจากสมมติฐานว่า เสียงเดียวกันจะมีความสัมพันธ์ของข้อมูล(ซึ่งพิจารณาข้อมูลในแกนความถี่)ต่อกันมากที่สุด และสัมพันธ์กับเสียงอื่นๆในลักษณะเดียวกันเสมอ

   -ในโครงงานนี้จะใช้ไฟล์ Library ซึ่งสร้างจากเสียง 50 เสียงเรียงลำดับกันไป จากลำดับเสียง “0” ทั้งหมดก่อน ตามด้วย “1”,”2”,… จนถึง “9” ทุกเสียงตามลำดับ

   -นำข้อมูลเสียงที่ต้องการหาคุณสมบัติไปคำนวณค่าสหสัมพันธ์กับทุกๆเสียงในไฟล์ library แล้วนำผลค่าสหสัมพันธ์กับแต่ละเสียงมาหาค่าเฉลี่ยสหสัมพันธ์ แยกแต่ละชนิดเสียง เพื่อนำมาเป็นข้อมูลอินพุทของโครงข่ายประสาทต่อไป

จากนั้นทำการฝึกสอนข้อมูลคุณลักษณะเสียงที่ได้จากส่วนปรีโปรเซสซิ่ง โดยกำหนดค่าความผิดพลาดรวมทั้งโครงข่าย(Train Error <Train Error Fix)ไว้ที่ 0.0005 คำนวณค่า Train Error จาก

$$\text{Train Error} = \frac{\sum_{n=1}^{numfile} Erf[n]}{Numfile}$$

ค่า Erf คือค่าความผิดพลาดที่คำนวณจากแต่ละไฟล์อินพุท และค่าเป้าหมายของไฟล์

$$\text{Erf ไฟล์ที่ n} = 0.5 * \left(\sum_{m=1}^{bitM} (Target_m - Output_m) \right)^2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อค่า Train Error ที่คำนวณได้ต่ำกว่ากำหนด ถือว่าโครงข่ายมีระดับความแม่นยำบรรลุเป้าหมายหรือเมื่อจำนวนครั้งที่สอนเกินกว่าจำนวนครั้งที่กำหนด จะหยุดทำการฝึกสอน ทั้งนี้เพื่อป้องกันกรณีที่โครงข่ายไม่สามารถหาค่าถ่วงน้ำหนักสำหรับค่าฝึกสอนชุดนั้นๆได้เลย

ผลการฝึกสอนให้กับโครงข่าย โดยกำหนดจำนวนครั้งที่สอนไว้ 3000 ครั้ง ดังแสดงต่อไปนี้

Please enter Training Time wanted : 3000

Time	Target	Output: bit1*bit2*bit3*bit4*	Error
3000	0000	0.054748*0.049215*0.069258*0.000857*	0.005108
3000	0001	0.042355*0.032319*0.035668*0.965246*	0.002659
3000	0010	0.003890*0.002254*0.965483*0.039465*	0.001385
3000	0011	0.085465*0.000020*0.962105*0.999038*	0.004371
3000	0100	0.000000*0.999999*0.026435*0.015653*	0.000472
3000	0101	0.074935*0.993833*0.002324*0.999600*	0.002829
3000	0110	0.000045*0.948954*0.936330*0.058524*	0.005042
3000	0111	0.000029*0.967090*0.966823*0.950417*	0.002321
3000	1000	0.999921*0.004319*0.000000*0.000835*	0.000010
3000	1001	0.864126*0.025500*0.001207*0.959844*	0.010363

end case1 time=1 stop=1Trainerr=0.003456

**********BETTER!so save weight**********

trainerrsave=0.003456 trainerrfix=0.000500

ผลการฝึกสอนดังกล่าว เมื่อนำมาทดสอบด้วยเสียงทดสอบอีก 50 เสียงที่ผ่านกระบวนปรีโปรเซสซิ่งเพื่อดึงคุณลักษณะแล้ว จะได้ผลการรู้จำดังแสดงต่อไปนี้

ครั้งที่	เสียง "0"	เสียง "1"	เสียง "2"	เสียง "3"	เสียง "4"	เสียง "5"	เสียง "6"	เสียง "7"	เสียง "8"	เสียง "9"
1	0	1	2	3	4	5	6	7	8	9
2	0	9	2	1	4	5	6	7	8	9
3	8	1	2	3	4	5	6	7	8	9
4	0	1	2	3	4	5	6	7	8	1
5	6	1	2	3	4	9	6	4	8	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลอง จะได้อัตรารู้จำเสียง “0” = 60%

เสียง “1” = 80%

เสียง “2” = 100%

เสียง “3” = 80%

เสียง “4” = 100%

เสียง “5” = 80%

เสียง “6” = 100%

เสียง “7” = 80%

เสียง “8”= 100%

เสียง “9” = 80%

หรือพิจารณาว่ามีอัตราการรู้จำรวมเฉลี่ย = 86%

<u>การวิเคราะห์ผล</u>

1. จากผลการทดลองจะเห็นว่าข้อมูลเสียงที่โปรแกรมแยกแยะผิดพลาดไป เช่นเสียง “0” วิเคราะห์ได้เป็นเสียง “8” และ “6” อย่างละครั้ง หากพิจารณาว่าเป็นเพราะเสียงทั้ง 3 เสียงมีคุณสมบัติที่คล้ายคลึงกันจึงทำให้วิเคราะห์ผิด แสดงว่าต้องมีกรณีที่ เสียง “8” หรือ ”6” ถูกวิเคราะห์เป็นเสียง “0” บ้าง

แต่จะเห็นว่าเสียง “8” และ “6” สามารถรู้จำได้อย่างถูกต้องทั้งหมด แสดงว่าความผิดพลาดที่เกิดในการวิเคราะห์เสียง “0” ไม่ได้เกิดจากการมีคุณสมบัติคล้ายคลึงกับเสียง “8” หรือ “6”แต่อย่างใด

และเช่นเดียวกันความผิดพลาดจากการรู้จำเสียงอื่นๆ ก็ไม่น่าจะเกิดจากความคล้ายคลึงของเสียง

หรือพิจารณาว่าการใช้ค่าสหสัมพันธ์ของเสียงทดสอบกับเสียงใน library file ทั้งหมด จะทำให้เป็นการรู้จำรูปแบบระดับความสัมพันธ์ของเสียงนั้นกับตัวเองและเสียงอื่นๆโดยเรียงลำดับเสียงแบบเดียวกัน ดังนั้น โอกาสที่เสียงคนละเสียงจะมีรูปแบบความสัมพันธ์กับตัวเองและเสียงคำอื่นๆได้สอดคล้องกันนั้นจึงเป็นไปได้ยาก โดยเฉพาะในโครงงานนี้ที่กำหนดขอบเขตของเสียง เป็นคำโดด และเป็นคำว่า “0,1,2,…,9” เท่านั้น

2. ความผิดพลาดที่เกิดขึ้นจากการวิเคราะห์เสียงอาจเกิดจากความผิดพลาดจากการตัดสัญญาณเสียงที่ยังไม่ละเอียดพอ และความแปรปรวนของเสียงผู้พูดเองจากอารมณ์ หรือการออกเสียงที่ไม่ชัด

3.จากภาคผนวกที่แสดงค่าสหสัมพันธ์ของเสียงกับ Library file จะเห็นว่าหากมีการแยกแยะเสียงด้วยการใช้ค่าสหสัมพันธ์สูงสุดเพียงอย่างเดียว จะมีความถูกต้องเพียง ประมาณ 60%

และเมื่อใช้โครงข่ายประสาทเทียมรู้จำเพียงอย่างเดียวจะต้องมีขนาดใหญ่มาก การใช้ค่าสหสัมพันธ์ร่วมกับการใช้โครงข่ายประสาทเทียม จึงเป็นการแบ่งเบาภาระซึ่งกันและกัน และช่วยเพิ่มความแม่นยำอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การใช้ค่าสหสัมพันธ์มาเป็นคุณลักษณะในการวิเคราะห์เสียงช่วยให้ลดขนาดของโครงข่ายประสาทลงไปได้อย่างมาก และย่อมส่งผลดีให้เวลาในการประมวลผลลดลง นอกจากนี้ยังพิจารณาได้ว่าช่วยลดงานของโครงข่ายประสาทเทียมลง ทำให้ใช้โครงข่ายขนาดเล็กๆก็เพียงพอแแก่การรู้จำ

<u>แนวทางการปรับปรุง</u>

จากผลการทดลองและการวิเคราะห์สาเหตุของการรู้จำที่ผิดพลาด สามารถอ้างได้ว่ากระบวนการปรีโปรเซสซิ่ง ที่ใช้ หากมีการเก็บข้อมูลอย่างละเอียดขึ้น จะสามารถใช้ดึงคุณลักษณะเสียงในขอบเขตของโครงงานนี้ได้ชัดเจนพอแก่การแยกแยะด้วยโครงข่ายประสาทเทียม

เนื่องจากส่วนการหาค่าสหสัมพันธ์ที่ใช้อยู่ในแกนความถี่ ดังนั้นจึงเป็นการแยกแยะเสียงโดยดึงคุณสมบัติด้าน ขนาด(Amplitude) และความถี่(Frequency) ของสัญญาณ หากเพิ่มเติมการวิเคราะห์คุณสมบัติด้านอื่นของสัญญาณในส่วนปรีโปรเซสซิ่งข้นอีก น่าจะทำให้ความสามารถของโครงข่ายประสาทรู้จำดีขึ้นอีกมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าสหสัมพันธ์ที่ได้จากเสียงทดสอบต่างๆ

ไฟล์ทดสอบ	Library "0"	Library "1"	Library "2"	Library "3"	Library "4"	Library "5"	Library "6"	Library "7"	Library "8"	Library "9"
M0_1	0.7207	0.5106	0.5673	0.3872	0.1481	0.4350	0.7286	0.6925	0.4223	0.5188
M0_2	0.7100	0.4886	0.5586	0.3805	0.1423	0.4272	0.7046	0.6799	0.4132	0.5300
M0_3	0.7111	0.5057	0.5511	0.4043	0.1366	0.5046	0.7254	0.6770	0.4874	0.5909
M0_4	0.4840	0.2764	0.2317	0.2135	0.0993	0.2219	0.3640	0.2925	0.1555	0.2050
M0_5	0.7893	0.5144	0.5403	0.3979	0.1775	0.4879	0.7547	0.6848	0.3990	0.4442
M1_1	0.4840	0.5378	0.3937	0.3441	0.1110	0.3915	0.5435	0.4945	0.3723	0.4182
M1_2	0.5047	0.5693	0.4109	0.3647	0..1395	0.4997	0.5720	0.5251	0.4059	0.4757
M1_3	0.3934	0.5058	0.3254	0.3417	0.2027	0.3633	0.4675	0.4220	0.2840	0.3383
M1_4	0.4316	0.5262	0.3857	0.3723	0.2609	0.3774	0.5098	0.4672	0.2994	0.3696
M1_5	0.4821	0.5724	0.4359	0.3358	0.1745	0.4092	0.5643	0.5323	0.3565	0.4178
M2_1	0.4162	0.3366	0.6185	0.4220	0.0418	0.3635	0.4395	0.4611	0.3394	0.3909
M2_2	0.4552	0.3685	0.6821	0.3363	0.0807	0.3912	0.4913	0.5450	0.3377	0.3989
M2_3	0.5208	0.4398	0.6860	0.3699	0.0928	0.4426	0.5709	0.6020	0.3813	0.4514
M2_4	0.4858	0.3866	0.6942	0.3936	0.0775	0.3927	0.5364	0.5788	0.3447	0.4191
M2_5	0.5711	0.4200	0.6542	0.3497	0.0994	0.4032	0.5896	0.6307	0.3948	0.4490
M3_1	0.3862	0.3972	0.4479	0.6005	0.1141	0.5113	0.4028	0.3782	0.4321	0.4534
M3_2	0.2358	0.2437	0.1915	0.4606	0.1327	0.4456	0.2777	0.1746	0.3069	0.8046
M3_3	0.3886	0.3879	0.4485	0.6114	0.0808	0.4776	0.4023	0.3843	0.4073	0.4553
M3_4	0.4319	0.4175	0.4802	0.5896	0.1450	0.5109	0.4560	0.4348	0.4313	0.4803
M3_5	0.3411	0.3123	0.3035	0.4423	0.0819	0.2936	0.3693	0.3138	0.2106	0.2497
M4_1	0.0977	0.092	0.0295	0.0407	0.3336	0.0881	0.1599	0.0872	0.0353	0.0346
M4_2	0.1778	0.1973	0.1502	0.1592	0.0395	0.1985	0.2551	0.2418	0.0776	0.0752
M4_3	0.1153	0.1666	0.0586	0.1191	0.4143	0.1017	0.1864	0.1315	0.0341	0.0174
M4_4	0.1409	0.1778	0.0441	0.1007	0.4291	0.0985	0.2057	0.1499	0.0411	0.0252
M4_5	0.1719	0.2547	0.1098	0.1346	0.2998	0.2075	0.2322	0.2033	0.1522	0.1559
M5_1	0.4330	0.4250	0.4320	0.4955	0.1475	0.7266	0.4646	0.4395	0.4557	0.5020
M5_2	0.3559	0.3488	0.3325	0.4026	0.1427	0.6656	0.3951	0.3637	0.3871	0.4366
M5_3	0.4011	0.3928	0.3728	0.4532	0.1714	0.7080	0.4344	0.3934	0.4026	0.4305
M5_4	0.4335	0.4204	0.4052	0.4621	0.1376	0.7219	0.4654	0.4740	0.4531	0.5120
M5_5	0.4531	0.4102	0.4510.	0.4256	0.0953.	0.5800	0.4770	0.4850	0.4723	0.5486

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************
                *
NORMALIZE TIME PATH     *

*****************************/
/*DECLATION*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <alloc.h>
#include <ermo.h>
#define frame 1024
#define numframe 5
#define lwant 4096

FILE *f,*fout;
int length;

void GetFile(void)
{
char InFile[40],OutFile[40];
clrscr();
printf("please enter file name to
normalize time- ");
gets(InFile); puts(InFile);
f=fopen(InFile,"r");
length=open(InFile,O_RDONLY);
if (f==NULL) {
   printf("Can't find this file! press
any key back");
   getch();  exit(0);
   }
printf("\nPlease enter out file
name\n");
gets(OutFile); puts(OutFile);
fout=fopen(OutFile,"w");
}

void NormalizeTime(void)
{
long int l,n,m;
int x,begin;
int *data;
float *dat;
float
xy,xx,yy,chk,dif,*val;//,*data/*rate
No change*/
printf("\n\nNormalize now!\n\n");
getch();
l=filelength(length);
if (l==  -1) printf("Can't find the
lenght\n");
if (ermo==EBADF) printf("Error in
handle:bad file number\n");
printf("file length in bytes=%ld\n",l);
getch();
data=(int*)farcalloc(l,sizeof(int));
dat =(float*)farcalloc(l,sizeof(float));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
val =(float*)farcalloc(lwant,sizeof
(float));
if ((data ==NULL)||(val==NULL)){
  clrscr(); printf("allocate
error!!press to exit\n"); getch(); exit
(0);}
for(n=1;n<=l;n++){
  fscanf(f,"%c",&data[n]);
//  fprintf(fout,"%c",data[n]);
  }
for(n=1;n<=l;n++){
  if (data[n]=='d') {if (data
[n+1]=='a'){if (data[n+2]=='t'){
  if (data[n+3]=='a') {
    printf("found HEAD laew na~ ei
ei"); begin=n+7;/*real begin
address*/
    l=l-(n+7);/*compute "l" to got
real length of file*/
    }}}}
  } getch();
printf("begin=%d  so now
lenght=%ld\n",begin,l); getch();
//fprintf(fout,"\n\nDATA BEFORE
COMPRESS\n");
for (n=1;n<=l;n++){ //l: new lenght
= (old)l-begin
  data[n]=data[n+begin];
  dat[n]=((float)data[n]-128)/128;
//  fprintf(fout," %f\n",dat[n]);
  }
//fprintf(fout,"\n\nNOW VALUE
COMPRESSED\n");
dif=(float)lwant/(float)l;//rate over
per one addr of l data
printf("dif=lwant/l=%f\n",dif); getch
();
for (n=1;n<=lwant;n++){
  for (m=0;m<=l;m++){ //find addr
besides keep in parameter x
    chk=(float)n-(float)dif*m;
/*dif*m=normalized addr*/
//    printf("chk=n-
dif*m=%f\n",chk);
    if ((dif>chk)&&(chk>=0)) x=m;
    }
  //  printf("x addr=%d\n",x);
   xx=dif*x; yy=dif*(x+1); //xx:pre
addr seen by lwant,yy:post addr
seen by lwant
  //  printf("xx=pre addr seen by
lwant=%f\n",xx);
  //  printf("yy=post addr seen by
lwant=%f\n",yy);
  xy=(n-xx)/(yy-xx); //yy-xx:addr
addition in l data seen by lwant
                    //addr change
=yy-xx : dat change =dat[x+1]-dat
[x]
                    //so addr
change n-xx: dat change =[(n-xx)/
(xx-yy)]*[dat[x+1]-dat[x]]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//  printf("xy=(n-xx)/(yy-
xx)=%f\n",xy);
   val[n]=(dat[x]+xy*(dat[x+1]-dat
[x]));
//  printf(" val[%ld]=%f",n,val[n]);
//  fprintf(fout," %f",val[n]);
  }
//fprintf(fout,"\n\nTHEN ARRANGE
IN FRAME\n");
int y;
y=1;
for (x=1;x<=numframe;x++){
  for (n=1;n<=frame;n++){
    fprintf(fout," %f",val[y]); y=y+1;
    }
  y=y-256; fprintf(fout,"\n\n"); printf
("finish a frame"); getch();
  }
printf("FINISH leaw ja~\n");
farfree((char*)data);  farfree((float*)
val);
getch();
}

void main(void)
{
GetFile();
NormalizeTime();
fclose(f);
fclose(fout);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Fast Fourier Transfrom

```
#define pi 3.141592654

#include<stdio.h>
#include<math.h>
#include<alloc.h>
#include<stdlib.h>
#include<conio.h>

void bit_reversal(unsigned
int*,int,int);
void WTS(float*,float*,int,int);
void FFT(float *xrr,float *xii,float
*,float *,int,int);
//void max_amp(void),display(void);
//int permute(int);
//double magnitude(int);

int i,k,m,N,n2,sign,power,n;
unsigned int *L;
float *wr,
*wi,*xr,*xi,max,*a,*xrr,*xii,*p;
//char msg[1000000];

char file_name[14];
char name[14];

FILE *workfile;

FILE *stream;

int main()
{

clrscr();

printf("\nEnter name of file containing
data points->");
scanf("%s",file_name);

printf("Enter of data pointsto be read --
>");
scanf("%d",&N);

if((workfile =
fopen(file_name,"rb"))==NULL)
{ fprintf(stderr,"Can not open input
file.\n");
return 1;
}
printf("\nEnter name of file containing
FFT points->");
scanf("%s",name);
if((stream =
fopen(name,"w"))==NULL)
{ fprintf(stderr,"Can not open output
file.\n");
 return 1;
}

//for (i=0;i<6;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//{


fseek(workfile,0,SEEK_SET);
for(n=0;n<5;n++)
{
/* Allocating memmory for FFT arrays
(real and imag.) */
        xr=(float *)malloc(N*sizeof
(float));
        xi=(float *)malloc(N*sizeof
(float));
if (xr == NULL)
{ printf("Can't allocate memmory");
exit(1);
}
if (xi == NULL)
{ printf("Can't allocate memmory");
exit(1);
}
/*

for (i=0; i<=(N-1); i++)
{
        fgets(msg,13L,workfile);

        xr[i] = atof(msg);

}
*/
```

```
for (i=0;i<=(N-1);i++){
  fscanf(workfile,"%f",&xr[i]);
// fprintf(fout," %f",x[n]);
 }

/*
        for(i=0;i<N;i++)
        printf("%f\n",xr[i]);
        getch();
*/

a=(float *)malloc(N*sizeof(float));
if (a == NULL)
{ printf("Can't allocate memmory");
exit(1);
}

for (i=0;i<=(N-1); i++)
a[i] = 0.54-0.46*cos((2*pi*i)/N);

/*apply window to data*/
for(i=0;i<N;i++)
xr[i]=xr[i]*a[i];
//printf("pass window");
/*for(i=0;i<N;i++)
printf("xr[%d]=%f\n",i,xr[i]);
*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
exit(1);
}
if (wi == NULL)
{ printf("Can't allocate memmory");
exit(1);
}

/*Genarating LUT for twiddle factors.
*/
        WTS(wr,wi,N,-1);

/*Taking FFT. */
        FFT(xrr,xii,wr,wi,m,N);
//      printf("FFT output");
p=(float *)malloc(N*sizeof(float));

if (p == NULL)
{ printf("Can't allocate memmory");
exit(1);
}
for(i=0;i<=(N-1);i++)
{
        p[i] = 20*log(sqrt(pow(xrr
[i],2)+pow(xii[i],2)));
        fprintf(stream, "%f ",p[i]);
}

//for(i=0;i<N;i++)
        //printf("xrr[%d]=%e\n",i,xrr[i]);

free(xr);
free(xi);
free(xrr);
free(xii);
free(L);
free(wr);
free(wi);
free(p);
}
fclose(stream);
fclose(workfile);

return 0;
}
void bit_reversal(unsigned int *L,int
m,int N)
/* Routine for generating LUT for bit
reversal.Note:N=(2 to the power of m)
        LUT will reside in L[] */
{
        unsigned int MASK,C,A,j,k,i;

        for (k=0;k<N;k++)
        {
        MASK=1;
        C=0;
        for(i=0,j=m-1;i<m; i++,j--)
        {
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m=(int)(log10((float)N)/log10((float)
2.0));
k=1;
for(i=0;i<m;i++)
        k<<=1;
if(k!=N)
        {
printf("n Length of file has to be
multiples of 2. ");
exit(1);
        }

/*Allocating memmory for bit reversal
LUT. */
L=(unsigned int *)malloc(N*sizeof
(unsigned int));
if (L == NULL)
{ printf("Can't allocate memmory");
exit(1);
}
/*Generate Look-up table for bit
reversal. */
        bit_reversal(L,m,N);
//      printf("after bit reverse");

        /*Setting-up the data in real
and img. arrays. */
xrr=(float *)malloc(N*sizeof(float));
xii=(float *)malloc(N*sizeof(float));
if (xrr == NULL)
{ printf("Can't allocate memmory");
exit(1);
}
if (xii == NULL)
{ printf("Can't allocate memmory");
exit(1);
}

        for(i=0;i<N;i++)
        {
        k=L[i];
        // printf("value k is %d\n",L
[i]);
        xrr[k]=xr[i];
        //      printf("turn %d x
[%d]=%e\n",i,k,xrr[k]);
        xii[k]=0.0;
        }
        //      fclose(fptr);
        /*      for(i=0;i<N;i++)
        printf("%f + %fi \n",xrr[i],xii
[i]);
        getch();
        */
/* Allocateing memmory for twiddle
factors. */
        n2=(N>>1)-1;
        wr=(float *)malloc(n2*sizeof
(float));
        wi=(float *)malloc(n2*sizeof
(float));
if (wr == NULL)
{ printf("Can't allocate memmory");
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
        A=(k&MASK)>>i;
        A<<=j;
        C|=A;
        MASK=MASK<<1;
        }
        L[k]=C;
        }
}

/*****************************
*************************/

void WTS(float *wr, float *wi, int N,
int sign)
/*Generating LUT for twiddle factors.
 Note: sign=-1 for FFT,and
        sign=1 for IFFT */
        {
        int n2,i;
        double theta;

        n2=(N>>1)-1;
 /*Generate look-up tables for twiddle
factor. */
 theta=2.0*pi/((float)N);
  for(i=0;i<n2;i++)
   {
   wr[i]=(float)cos((float)
  ((i+1)*theta));
   wi[i]=(float)sin((float)((i+1)*theta));
   if(sign==(int)(-1)) wi[i]=-wi[i];
   }
  }

/*****************************
*****************************/

 void FFT
 (float *xrr,float *xii,float *wr, float
*wi, int m,int N)
 {
 int ip,k,kk,l,incr,iter,j,i,loop;
 float Tr,Ti;

  /* scale Data */
        for (loop=0;loop<N;loop++)
        { xrr[loop]/=(float)N;
         xii[loop]/=(float)N;
        }

 ip=1;
 kk=(N>>1);
 incr=2;

 for(iter=0; iter<m; iter++)
       {
       for(j=0; j<N; j+=incr)
        {
        i=j+ip;
        Tr=xrr[i];
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น การอ้างอิงเนื้อหาต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Ti=xii[i];
xrr[i]=xrr[j]-Tr;
xii[i]=xii[j]-Ti;

xrr[j]=xrr[j]+Tr;
xii[j]=xii[j]+Ti;
}
if(iter!=0)
 {
 for(k=1; k<ip;k++)
  {
  l=k*kk-1;
 for(j=k; j<N; j+=incr)
  {
  i=j+ip;
  Tr=xrr[i]*wr[l]-xii[i]*wi[l];
  Ti=xrr[i]*wi[l]+xii[i]*wr[l];

  xrr[i]=xrr[j]-Tr;
  xii[i]=xii[j]-Ti;

  xrr[j]=xrr[j]+Tr;
  xii[j]=xii[j]+Ti;
  }
 }
}
kk>>=1;
ip<<=1;
incr<<=1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
val =(float*)farcalloc(lwant,sizeof
(float));
if ((data ==NULL)||(val==NULL)){
   clrscr(); printf("allocate
error!!press to exit\n"); getch(); exit
(0);}
for(n=1;n<=l;n++){
   fscanf(f,"%c",&data[n]);
//   fprintf(fout,"%c",data[n]);
   }
for(n=1;n<=l;n++){
   if (data[n]=='d') {if (data
[n+1]=='a'){if (data[n+2]=='t'){
   if (data[n+3]=='a') {
      printf("found HEAD laew na~ ei
ei"); begin=n+7;/*real begin
address*/
      l=l-(n+7);/*compute "l" to got
real length of file*/
      }}}}
  } getch();
printf("begin=%d  so now
lenght=%ld\n",begin,l); getch();
//fprintf(fout,"\n\nDATA BEFORE
COMPRESS\n");
for (n=1;n<=l;n++){ //l: new lenght
= (old)l-begin
   data[n]=data[n+begin];
   dat[n]=((float)data[n]-128)/128;
//   fprintf(fout," %f\n",dat[n]);
   }
//fprintf(fout,"\n\nNOW VALUE
COMPRESSED\n");
dif=(float)lwant/(float)l;//rate over
per one addr of l data
printf("dif=lwant/l=%f\n",dif); getch
();
for (n=1;n<=lwant;n++){
   for (m=0;m<=l;m++){ //find addr
besides keep in parameter x
      chk=(float)n-(float)dif*m;
/*dif*m=normalized addr*/
//      printf("chk=n-
dif*m=%f\n",chk);
      if ((dif>chk)&&(chk>=0)) x=m;
      }
   //   printf("x addr=%d\n",x);
      xx=dif*x; yy=dif*(x+1); //xx:pre
addr seen by lwant,yy:post addr
seen by lwant
   //   printf("xx=pre addr seen by
lwant=%f\n",xx);
   //   printf("yy=post addr seen by
lwant=%f\n",yy);
   xy=(n-xx)/(yy-xx); //yy-xx:addr
addition in l data seen by lwant
                  //addr change
=yy-xx : dat change =dat[x+1]-dat
[x]
                  //so addr
change n-xx: dat change =[(n-xx)/
(xx-yy)]*[dat[x+1]-dat[x]]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//  printf("xy=(n-xx)/(yy-
xx)=%f\n",xy);
   val[n]=(dat[x]+xy*(dat[x+1]-dat
[x]));
//  printf(" val[%ld]=%f",n,val[n]);
//  fprintf(fout," %f",val[n]);
  }
//fprintf(fout,"\n\nTHEN ARRANGE
IN FRAME\n");
int y;
y=1;
for (x=1;x<=numframe;x++){
  for (n=1;n<=frame;n++){
    fprintf(fout," %f",val[y]); y=y+1;
    }
  y=y-256; fprintf(fout,"\n\n"); printf
("finish a frame"); getch();
  }
printf("FINISH leaw ja~\n");
farfree((char*)data);  farfree((float*)
val);
getch();
}

void main(void)
{
GetFile();
NormalizeTime();
fclose(f);
fclose(fout);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************
*****************************
*
    *                CORRELATION
PROGRAM          *
    *    by assign NN,frame and
data.for speech recognition    *
    *  the sound arrange as follow
style : group 0,group 1,...  *
    *  till the group 9.with define the
num of sound in a set   *

*****************************
*****************************
/

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define NN 1024 //all data in a frame
#define frame 5 //num frame
#define data 5120
#define num_sound 5 //num of sound
arrange in any group
#define kind_sound 10 // sound
0,1,2,3,4,5,6,7,8,9
int n,m;  //paramiter for a sound
int l,k,j;  //paramiter for count all
sound in set
float in_neu[kind_sound]; //for freq
store
float lib_[frame],in_[frame],

lib_stand_devia[frame],in_stand_devia
[frame],
    z_lib[frame],z_in[frame],
    get_in[data+1],get_lib[data+1],
    point_r,frame_r[frame],total_r,r_;
float r_list[kind_sound];//store freg of
r
FILE *lib,*in,*out;
char lib_name[20],in_name
[20],out_name[20];

void corre()
{
 //fseek(lib,0,SEEK_CUR);
/*find lib_[m]*/
  for (n=1;n<=5120;n++){
    fscanf(lib,"%f",&get_lib[n]);
fscanf(in,"%f",&get_in[n]);
   }
  float sum_lib;
  for (m=0;m<=frame-1;m++){
    sum_lib=0;;
    for (n=m*NN+1;n
<=m*NN+NN;n++)
        sum_lib+=get_lib[n];
    lib_[m]=sum_lib/NN;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
/*find in_[m]*/
  float sum_in;
  for (m=0;m<=frame-1;m++){
    sum_in=0;
    for (n=m*NN+1;n
<=m*NN+NN;n++)
        sum_in+=get_in[n];
    in_[m]=sum_in/NN;
    }
/* find lib_stand_devia[m]*/
  float devia,sum_devia;
  for (m=0;m<=frame-1;m++){
    sum_devia=0;
    for (n=m*NN+1;n
<=m*NN+NN;n++){
        devia=pow(get_lib[n]-lib_
[m],2);  sum_devia+=devia;
        }
    sum_devia=sum_devia/NN;
lib_stand_devia[m]=sqrt(sum_devia);
    }
/*find in_stand_devia[m]*/
  for (m=0;m<=frame-1;m++){
    sum_devia=0;
    for (n=m*NN+1;n
<=m*NN+NN;n++){
        devia=pow(get_in[n]-in_
[m],2); sum_devia+=devia;
        }
    sum_devia=sum_devia/NN;
 in_stand_devia[m]=sqrt(sum_devia);
    }
/* find corre()*/
  total_r=0;
  for (m=0;m<=frame-1;m++){
    frame_r[m]=0;
    for (n=m*NN+1;n
<=m*NN+NN;n++){
        z_lib[m]=(get_lib[n]-lib_
[m])/lib_stand_devia[m];
        z_in[m]=(get_in[n]-in_
[m])/in_stand_devia[m];
        point_r=(z_lib[m]*z_in
[m])/NN;  frame_r[m]+=point_r;
        }
    total_r+=frame_r[m]; //printf
("frame_r[%d]=%f\n",m,frame_r[m]);
//printf("total_r+=frame_r=%f\n",total
_r);
    }
  r_=total_r/frame;
  r_=fabs(r_);
//printf("\nr_=r_total/frame=%f\n",r_);
getch();
//  printf("design till here!!"); getch();
}

main()
{
float get;
clrscr();
```

เอกสารนี้ใช้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("************************
******************************
******\n");
printf("*              NOW
CORRELATION PART
*\n");
printf("* This program use with library
file that include 30 sound   *\n");
printf("* file arrange by set from 0-
1,0-2,0-3,1-1,1-2,...,9-3 file  *\n");
printf("* and file input to compute
correlation must complete FFT.
*\n");
printf("* This corre file list by 5 frame
is the most suit form.    *\n");
printf("*
*\n");
printf("************************
******************************
******\n");

printf("Enter library sound file name
:"); scanf("%20s",lib_name);
puts(lib_name);  lib=fopen
(lib_name,"r");
if(lib==NULL){
  printf("Error open library
sound!!.Press to exit"); getch(); exit
(0);
  }
printf("Enter file name to compute
correlation :");
scanf("%20s",in_name);
puts(in_name);  in=fopen
(in_name,"r");
if(in==NULL){
  printf("Error open file!!.Press to
exit"); getch(); exit(0);
  }
printf("enter out file name : "); scanf
("%20s",out_name); puts(out_name);
for (l=1;l<=kind_sound;l++){
//num_sound=2 so have 20 sounds.
  for (k=1;k<=num_sound;k++){
    corre();
    r_list[l]+=r_;// printf("r_list
    [%d]+=r_=%f\n",l,r_list[l]);
    fseek(lib,0,SEEK_CUR);
    }
  }
for (l=1;l<=kind_sound;l++)
  printf("r_list[%d]=%f\n",l,r_list[l]);
for (l=1;l<=kind_sound;l++){
  in_neu[l]=r_list[l]/num_sound;
//
printf("in_neu[l]=r_list/num_sound=%
f\n",in_neu[l]); getch();
  }
out=fopen(out_name,"w");
for (l=1;l<=kind_sound;l++){
  fprintf(out,"%f\n",in_neu[l]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  printf("\nm[%d]=%f",l-1,in_neu[l]);
getch();
  }
fclose(in); fclose(lib);
printf("\n\n  THANKS. NO SERVICE
CHARGE.GOODLUCK\n");
printf("  Press to out of program.");
getch();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************
*****************************
   BACKPROPAGATION
NETWORK FOR SPEECH
RECOGNITION (NUMBER 0-9)

*****************************
*****************************/
/* DECLARATION */
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <alloc.h>

#define BIAS      1
#define sqr(x)    ((x)*(x))
#define FALSE     0
#define TRUE      1
#define NOT       !
#define NUM_LAYERS  3
#define frame_byte  32  //num of data
(byte) in each frame
#define IL        10  //equal as the
sample from each frame
#define OL        4   //equal as bit
numbers of output
#define TrainErrFix 0.0005
/*get data input as continous frame, a
unit learn for one time*/
typedef int     INT;
typedef int     BOOL;
typedef double    DOU;
FILE
*f,*fl,*f0,*f1,*f2,*f3,*f4,*f5,*f6,*f7,*
f8,*f9,*Storef,*ff;
INT   Units[NUM_LAYERS] =
{IL,8,OL};
DOU
*a0,*a1,*a2,*a3,*a4,*a5,*a6,*a7,*a8,
*a9,*aa;
DOU
Erf1,Erf2,Erf3,Erf4,Erf5,Erf6,Erf7,Erf
8,Erf9,Erf10;
double  TrainErr,TrainErrSave;
double  TestError,MM=0;
INT   numfile,T10[5],T9[5],T8[5],T7
[5],T6[5],T5[5],T4[5],T3[5],T2[5],T1
[5];
INT   Time,stop,TrainTime;

typedef struct{          /* A LAYER
OF A NET */
int     Units;         /* number of units
in this layer */
double   *Output;       /* output of
ith unit */
double   *Error;        /* error term of
ith unit */
double   **Weight;      /* connection
weights to ith unit */
double   **WeightSave;  /* save
weight when stop training */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
}LAYER;

typedef struct{          /* A NET */
    LAYER    **Layer;   /* Layer of
this net */
    LAYER    *InputLayer; /* input
layer or LAYER[0] */
    LAYER    *OutputLayer;/* output
layer or LAYER[2] */
    double   Error;     /* Total net
error */
    double   Delta;     /* learning rate
*/
    double   Gain;      /* Gain of
sigmoid */
    double   Alpha;
}NET;
void ChoseOption(NET *Net);
void SimulateNet(NET *Net,DOU
*Output,INT *Target,BOOL
Training);


/* INITIALIZATION */
void GenerateNetwork(NET *Net)
{
int l,i;
Net->Layer=(LAYER**)calloc
(NUM_LAYERS,sizeof(LAYER*));
for (l=0;l<NUM_LAYERS;l++){
  Net->Layer[l]      =(LAYER*)
 malloc(sizeof(LAYER));
  Net->Layer[l]->Units  =Units[l];
  Net->Layer[l]->Output =(DOU*)
calloc(Units[l]+1,sizeof(DOU));
  Net->Layer[l]->Error  =(DOU*)
calloc(Units[l]+1,sizeof(DOU));
  Net->Layer[l]->Weight =(DOU**)
calloc(Units[l]+1,sizeof(DOU*));
  Net->Layer[l]->WeightSave=
(DOU**)calloc(Units[l]+1,sizeof
(DOU*));
  Net->Layer[l]->Output[0] =BIAS;
  if (l!=0){
    for (i=1;i<=Units[l];i++){
    Net->Layer[l]->Weight[i]    =
(DOU*)calloc(Units[l-1]+1,sizeof
(DOU));
      Net->Layer[l]->WeightSave[i] =
(DOU*)calloc(Units[l-1]+1,sizeof
(DOU));
    }
  }
}
Net->InputLayer =Net->Layer[0];
Net->OutputLayer=Net->Layer[2];
Net->Delta     =0.5;
Net->Gain      =1;
Net->Alpha     =0.9;
}

void ReserveInt()  /*reserve input
layer*/
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
a0=(DOU*)calloc(IL,sizeof(DOU));
a1=(DOU*)calloc(IL,sizeof(DOU));
a2=(DOU*)calloc(IL,sizeof(DOU));
a3=(DOU*)calloc(IL,sizeof(DOU));
a4=(DOU*)calloc(IL,sizeof(DOU));
a5=(DOU*)calloc(IL,sizeof(DOU));
a6=(DOU*)calloc(IL,sizeof(DOU));
a7=(DOU*)calloc(IL,sizeof(DOU));
a8=(DOU*)calloc(IL,sizeof(DOU));
a9=(DOU*)calloc(IL,sizeof(DOU));
aa=(DOU*)calloc(IL,sizeof(DOU));
}

void RandomWeights(NET *Net)
{
INT l,i,j;
printf("Begin all weights by
random\n");  getch();
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
    for (j=0;j<=Net->Layer[l-1]->
Units;j++){
          Net->Layer[l]->Weight[i]
[j]=(DOU)(random(9999))/100000;
//        Net->Layer[l]->Weight[i]
[j]=0.1;
          }
    }

  }
}

void FileToWeights(NET *Net)
{
DOU wei;
INT i,j,l;
char StoreFile[40];
printf("Please enter weight file
name\n");
gets(StoreFile); puts(StoreFile);
Storef=fopen(StoreFile,"r");//fopen
("store.txt","r");
if (Storef==NULL) {
  printf("Can't find this weight file!
Exit now!!\n");
  getch(); exit(0);
  }
printf("Begin all weight from file\n");
getch();
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
    for (j=0;j<=Net->Layer[l-1]->
Units;j++){
          fscanf(Storef,"%lf",&wei);
          Net->Layer[l]->Weight[i]
[j]=wei;
          printf("W[%d]
[%d]=%lf",i,j,Net->Layer[l]->Weight
[i][j]);
          /*display Wij*/
          } getch();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
fclose(Storef);
}

void InitialFile()
{
f=fopen("PUPU.txt","w");
}

/* SIMULATE */
void ErrorOpenFile(NET *Net)
{
fprintf(stderr,"\n\n\nNo file here.Press
any key to back!\n");
getch();  ChoseOption(Net);
}

void GetTarget(NET *Net,INT
*Target)
{
INT i,m;
Target[0]=0;
printf("\nEnter Target in 4 bit
binary\n");
for (m=1;m<=4;m++){
  printf("bit %d\n",m);
  scanf("%d",&Target[m]);
  }
printf("Target=");
for(m=1;m<=4;m++)
  printf("%d",Target[m]);
}

int GetTrainTime(NET *Net)
{
printf("\nPlease enter Training Time
wanted ");
scanf("%d",&TrainTime);
return TrainTime;
printf("\nYou want to train %d
time\n",TrainTime);  getch();
}

void GetInput(NET *Net,FILE
*fl,DOU *a)
{
INT i,n;
fseek(fl,0,SEEK_SET);    /*begin data
input at first address*/
for (i=1;i<=Net->InputLayer->
Units;i++){
  fscanf(fl," %lf",&a[i]);
  Net->InputLayer->Output[i]=a
[i];//10000;
//  printf("In[%d]=%lf ",i,a[i]);
  }
//  getch();
}

void PropagateLayer(NET
*Net,LAYER *UpLayer,LAYER
*LoLayer,DOU *a)
{
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
}

void BackpropagateLayer(NET
*Net,LAYER *UpLayer,LAYER
*LoLayer,DOU *a)
{
INT i,j;
DOU Out,Err;
for (i=1;i<=LoLayer->Units;i++){
  Out=LoLayer->Output[i];
  Err=0;
  for (j=1;j<=UpLayer->Units;j++){
//     printf("UpErr[%d]=%lf
",j,UpLayer->Error[j]);
     Err += UpLayer->Weight[j]
[i]*UpLayer->Error[j];
     } //printf("Err=sum
(w*UpErr)=%lf\n",Err);
//  printf("\nOut layer %d=%lf
",i,Out);  getch();
  LoLayer->Error[i]=Net->Gain*Out*
(1-Out)*Err;
//  printf("Error[i]=Err*f'=%lf*%lf(1-
%lf)=%lf\n",Err,Out,Out,LoLayer->
Error[i]);
  }// getch();
}

void BackpropagateNet(NET
*Net,DOU *a)
{
INT l;
//printf("b a c k p r o p a g a t e\n");
for (l=2;l>1;l--){
  BackpropagateLayer(Net,Net->
Layer[l],Net->Layer[l-1],a);
  }
}

void AdjustWeight(NET *Net,DOU
*a)
{
INT l,i,j,Time;
DOU Out,Err,M;
//printf("adjust weight by add
Delta*Err*Out\n"); getch();
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
        for (j=0;j<=Net->Layer[l-1]-
>Units;j++){
        if (l==1) Net->Alpha=0; else
Net->Alpha=0.9;
//        printf("Alpha=%lf",Net->
Alpha);
        Out=Net->Layer[l-1]->
Output[j];
        Err=Net->Layer[l]->Error[i];
//        printf("oldW[%d][%d]=%lf
&&&& ",i,j,Net->Layer[l]->Weight[i]
[j]);
//        Net->Layer[l]->Weight[i][j]
+= Net->Delta*Err*Out;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INT i,j,m;
DOU Sum;
j=0;
for (i=1;i<=UpLayer->Units;i++){
  Sum=0;
  for (j=1;j<=LoLayer->Units;j++){
    Sum += UpLayer->Weight[i]
[j]*LoLayer->Output[j];
//    printf("Weight[%d]
[%d]=%lf*output(%lf)\n",i,j,UpLayer-
>Weight[i][j],LoLayer->Output[j]);
    }
  UpLayer->Output[i] = 1/(1+exp(-
Net->Gain*Sum));
//
printf("Output[%d]=%lf\n",i,UpLayer-
>Output[i]);  getch();
  }
}

void PropagateNet(NET *Net,DOU
*a)
{
INT l;
//clrscr();
//printf("P r o p a g a t e F o r w o r
d\n"); getch();
for (l=1;l<NUM_LAYERS;l++){
  PropagateLayer(Net,Net->Layer
[l],Net->Layer[l-1],a);
  }
}

void GetOutput(NET *Net,DOU
*Output,DOU *a)
{
INT i;
for (i=1;i<=Net->OutputLayer->
Units;i++){
  Output[i-1] = Net->OutputLayer->
Output[i];
  }
}

void ComputeOutputErr(NET
*Net,INT *Target,DOU *a)
{
DOU Err,Out;
INT i,OutBit;
Net->Error=0;
//clrscr();
//printf("C o m p u t e O u t L a y e r E
r r\n"); getch();
for  (i=1;i<=Net->OutputLayer->
Units;i++){
  Out=Net->OutputLayer->Output[i];
  Err=Target[i]-Out; // printf("Err
(OutLayer)=Tar-Out=%lf\n",Err);
  Net->OutputLayer->Error[i] = Net->
Gain*Out*(1-Out)*Err;
//  printf("Error[%d]of out
layer=Err*f'=%lf\n",i,Net->
OutputLayer->Error[i]); getch();
  Net->Error += 0.5*sqr(Err);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
            M=Net->Delta*Err*Out;
            Net->Layer[l]->Weight[i]
[j]+=M+MM*Net->Alpha;
            MM=M;
//          printf("newW add
%lf*%lf*%lf+%lf*%lf=%lf\n",Net->
Delta,Err,Out,MM,Net->Alpha,Net->
Layer[l]->Weight[i][j]);
          }
//        getch();
    }
  }
}

void SimulateNet(NET *Net,DOU
*Output,INT *Target,BOOL
Training,DOU *a)
{
INT i;
PropagateNet(Net,a);
GetOutput(Net,Output,a);
if (Training){
  ComputeOutputErr(Net,Target,a);
  BackpropagateNet(Net,a);
  AdjustWeight(Net,a);
  }
}

void SaveWeight(NET *Net)
{
INT i,j,l;
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
    for (j=0;j<=Net->Layer[l-1]->
Units;j++)
      Net->Layer[l]->WeightSave
[i][j]=Net->Layer[l]->Weight[i][j];
    }
  }
}

void RecallWeight(NET *Net)
{
INT i,j,l;
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
    for (j=0;j<=Net->Layer[l-1]->
Units;j++)
      Net->Layer[l]->Weight[i]
[j]=Net->Layer[l]->WeightSave[i][j];
    }
  }
}

void StoreWeightFile(NET *Net)
{
INT i,j,l;
Storef=fopen("be.txt","w");//fopen
(StoreFileName,"w");
for (l=1;l<NUM_LAYERS;l++){
  for (i=1;i<=Net->Layer[l]->
Units;i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for (j=0;j<=Net->Layer[l-1]->
Units;j++){
        fprintf(Storef," %lf",Net->
Layer[l]->Weight[i][j]);
        printf("W[%d][%d]=%lf
",i,j,Net->Layer[l]->Weight[i][j]);
        } getch();
    }
  }
fclose(Storef);
printf("\n already store\n"); getch();
}

void ManageWeight(NET *Net,DOU
*Output,INT *Target,BOOL
Training,DOU *a,int TrainTime)
{
INT m;
DOU Error;
//if (TrainTime-Time==1){
  printf("\n%d  ",Time);
  for (m=1;m<=4;m++)
    printf("%d",Target[m]);
  printf("   ");
  for (m=1;m<=4;m++)
    printf("%lf*",Net->OutputLayer->
Output[m]);
  printf("   %lf",Net->Error);
  getch();
//  }
}

void TrainNet(NET* Net,INT
*Target)
{
DOU Output[OL];
INT i,m;
Time=0;  stop=1;
TrainErr=0;
TrainErrSave=HUGE_VAL;
ReserveInt();
printf("trainerrsave=%lf\n",TrainErrSa
ve);
char tf10[20],tf9[20],tf8[20],tf7[20],tf6
[20],tf5[20],tf4[20],tf3[20]
  ,tf2[20],tf1[20];
printf("How many train file\n");
scanf("%d",&numfile);
printf("%d\n",numfile);
switch(numfile){
  case 10: printf("\nEnter 10th train
file\n"); scanf("%40s",tf10);
        puts(tf10); f9=fopen
(tf10,"r");
        if (f9==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T10);

  case 9 : printf("\nEnter 9th train
file\n");  scanf("%40s",tf9);
        puts(tf9); f8=fopen(tf9,"r");
        if (f8==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T9);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  case 8 : printf("\nEnter 8th train
file\n");  scanf("%40s",tf8);
            puts(tf8); f7=fopen(tf8,"r");
            if (f7==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T8);

  case 7 : printf("\nEnter 7th train
file\n"); scanf("%40s",tf7);
            puts(tf7); f6=fopen(tf7,"r");
            if (f6==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T7);

  case 6 : printf("\nEnter 6th train
file\n"); scanf("%40s",tf6);
            puts(tf6); f5=fopen(tf6,"r");
            if (f5==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T6);

  case 5 : printf("\nEnter 5th train
file\n"); scanf("%40s",tf5);
            puts(tf5); f4=fopen(tf5,"r");
            if (f4==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T5);

  case 4 : printf("\nEnter 4th train
file\n"); scanf("%40s",tf4);
            puts(tf4); f3=fopen(tf4,"r");
            if (f3==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T4);

  case 3 : printf("\nEnter 3rd train
file\n"); scanf("%40s",tf3);
            puts(tf3); f2=fopen(tf3,"r");
            if (f2==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T3);

  case 2 : printf("\nEnter 2nd train
file\n"); scanf("%40s",tf2);
            puts(tf2); f1=fopen(tf2,"r");
            if (f1==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T2);

  case 1 : printf("\nEnter 1st train
file\n"); scanf("%40s",tf1);
            puts(tf1); f0=fopen(tf1,"r");
            if (f0==NULL)
ErrorOpenFile(Net); GetTarget
(Net,T1); break;

  default: clrscr(); printf("Error! Press
to back\n"); getch();
            ChoseOption(Net);
  }
TrainTime=GetTrainTime(Net);
do{
switch(numfile){
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 10: GetInput(Net,f9,a9);
SimulateNet(Net,Output,T10,TRUE,a
9);
            Erf10=Net->Error;
ManageWeight(Net,Output,T10,TRUE
,a9,TrainTime);

    case 9 : GetInput(Net,f8,a8);
SimulateNet(Net,Output,T9,TRUE,a8)
;
            Erf9=Net->Error;
ManageWeight(Net,Output,T9,TRUE,
a8,TrainTime);

    case 8 : GetInput(Net,f7,a7);
SimulateNet(Net,Output,T8,TRUE,a7)
;
            Erf8=Net->Error;
ManageWeight(Net,Output,T8,TRUE,
a7,TrainTime);

    case 7 : GetInput(Net,f6,a6);
SimulateNet(Net,Output,T7,TRUE,a6)
;
            Erf7=Net->Error;
ManageWeight(Net,Output,T7,TRUE,
a6,TrainTime);

    case 6 : GetInput(Net,f5,a5);
SimulateNet(Net,Output,T6,TRUE,a5)
;
            Erf6=Net->Error;
ManageWeight(Net,Output,T6,TRUE,
a5,TrainTime);

    case 5 : GetInput(Net,f4,a4);
SimulateNet(Net,Output,T5,TRUE,a4)
;
            Erf5=Net->Error;
ManageWeight(Net,Output,T5,TRUE,
a4,TrainTime);

    case 4 : GetInput(Net,f3,a3);
SimulateNet(Net,Output,T4,TRUE,a3)
;
            Erf4=Net->Error;
ManageWeight(Net,Output,T4,TRUE,
a3,TrainTime);

    case 3 : GetInput(Net,f2,a2);
SimulateNet(Net,Output,T3,TRUE,a2)
;
            Erf3=Net->Error;
ManageWeight(Net,Output,T3,TRUE,
a2,TrainTime);

    case 2 : GetInput(Net,f1,a1);
SimulateNet(Net,Output,T2,TRUE,a1)
;
            Erf2=Net->Error;
ManageWeight(Net,Output,T2,TRUE,
a1,TrainTime);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 1 : GetInput(Net,f0,a0);
SimulateNet(Net,Output,T1,TRUE,a0)
;
            Erf1=Net->Error;
ManageWeight(Net,Output,T1,TRUE,
a0,TrainTime);
            Time=Time+1;
            printf("\nend case1
time=%d   stop=%d",Time,stop);
getch();
            break;

    default: clrscr(); printf
("Error!!Press to back!\n");
            getch(); exit(0);
    }

TrainErr=(Erf1+Erf2+Erf3+Erf4+Erf5
+Erf6+Erf7+Erf8+Erf9+Erf10)/10;
  printf("Trainerr=%lf\n",TrainErr);
  if (TrainErr<TrainErrSave){
// printf("    *****BETTER SO
SAVE WEIGHT\n");

printf("*****************BETTER!s
o save
weight*****************\n");
  SaveWeight(Net);
  TrainErrSave=TrainErr;
  }
  if (Time>=TrainTime) stop=0;
  printf("trainerrsave=%lf
trainerrfix=%lf\n",TrainErrSave,Train
ErrFix); getch();
  if (TrainErrSave<=TrainErrFix)
stop=0;
  printf("----------------------------------
----------------------------------------");
  }while (stop);
if (TrainErr>TrainErrFix)
RecallWeight(Net);
StoreWeightFile(Net);
printf("\n\n  FINISH TRAINING...
train %d time\n",Time);
printf("THE BEST RESULT...
O/P_ERROR=%lf\n",TrainErrSave);
printf("press any key to continue\n");
getch();
}

void Display(NET *Net)
{
INT n,disp,count;

printf("RESULT NUMBER IN
DECADE\n");
count=0;
for (n=1;n<=4;n++){
  printf("\nbit %d = %lf\n",n,Net->
OutputLayer->Output[n]);
  if ((Net->OutputLayer->Output[n])>
0.5) disp=1; else disp=0;
  switch(n){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 1 : printf("n=%d\n",n); count=
(disp*8)+count; break;
    case 2 : printf("n=%d\n",n); count=
(disp*4)+count; break;
    case 3 : printf("n=%d\n",n); count=
(disp*2)+count; break;
    case 4 : printf("n=%d\n",n); count=
(disp*1)+count; break;
    }
  }
printf("\n
all=%d\n\n\n\n\n\n\n\n\n",count);  getch
();
switch(count){
  case 0 : printf("      0000\n");
           printf("     00  00\n");
           printf("     00  00\n");
           printf("     00  00\n");
           printf("     00  00\n");
           printf("     00  00\n");
           printf("     00  00\n");
           printf("     0000\n");
           break;
  case 1 : printf("       11\n");
           printf("      111\n");
           printf("     1111\n");
           printf("       11\n");
           printf("       11\n");
           printf("       11\n");
           printf("       11\n");
           printf("       11\n");
           break;
  case 2 : printf("      2222\n");
           printf("     22  22\n");
           printf("     22  22\n");
           printf("       22\n");
           printf("      22\n");
           printf("     22\n");
           printf("     22\n");
           printf("     222222\n");
           break;
  case 3 : printf("      3333\n");
           printf("     33  33\n");
           printf("       33\n");
           printf("      333\n");
           printf("       33\n");
           printf("       33\n");
           printf("     33  33\n");
           printf("      3333\n");
           break;
  case 4 : printf("      444\n");
           printf("     4444\n");
           printf("    44 44\n");
           printf("   44  44\n");
           printf("  444  44\n");
           printf("  44444444\n");
           printf("     44\n");
           printf("     44\n");
           break;
  case 5 : printf("     555555\n");
           printf("     55\n");
           printf("     55\n");
           printf("     55555\n");
           printf("       55\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SimulateNet(Net,Output,Target,FALS
E,aa);
Display(Net);
}

void ChoseOption(NET *Net)
{
INT Target[5];
INT op;
clrscr();
printf("Please chose option\n");
printf("Press 1 for Train Network
option\n");
printf("Press 2 for Test Network
option\n");
printf("Press 3 for Exit the program");
op=getch()-48;
switch(op){
  case 1 : clrscr(); printf("Now
Training\n");
          TrainNet(Net,Target);
ChoseOption(Net);
  case 2 : clrscr(); printf("Now Test
Network\n");
          TestNet(Net,Target);
ChoseOption(Net);
  case 3 : exit(0); break;
  default: clrscr(); printf("\n\n Press
wrong!! Try again");
          getch(); ChoseOption(Net);
  }
}

void EndFile(NET* Net)
{
int l,i;
free((void**)(Net->Layer));
for (l=0;l<NUM_LAYERS;l++){
  free((void*)Net->Layer[l]);
  free((void*)Net->Layer[l]->Output);
  free((void*)Net->Layer[l]->Error);
  free((void**)Net->Layer[l]->
Weight);
  free((void**)Net->Layer[l]->
WeightSave);
  for (i=0;i<=Units[l];i++){
    free((void**)Net->Layer[l]->
Weight[i]);
    free((void**)Net->Layer[l]->
WeightSave[i]);
  }
  }
free((void*)a0); free((void*)a1); free
((void*)a2); free((void*)a3);
free((void*)a4); free((void*)a5); free
((void*)a6); free((void*)a7);
free((void*)a8); free((void*)a9); free
((void*)aa);
fclose(f); fclose(fl); fclose(f0); fclose
(f1); fclose(f2); fclose(f3);
fclose(f4); fclose(f5); fclose(f6); fclose
(f7); fclose(f8); fclose(f9);
fclose(ff);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* MAIN */
void main()
{
INT    ww;
NET    Net;
BOOL   Stop;              /* running
"do-while" loop when Stop=TRUE */
double  MinTestError;
GenerateNetwork(&Net);
clrscr();
printf("Want to    1.random weight or
2.weight from file\n");
ww=getch()-48;
switch(ww){
  case 1 : RandomWeights(&Net);
break;
  case 2 : FileToWeights(&Net);
break;
  default: clrscr; printf("Press Wrong!
Try again!!\n");
            getch(); ChoseOption
(&Net);
  }
InitialFile();
Stop=FALSE;
ChoseOption(&Net);
EndFile(&Net);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงงานครั้งนี้ได้รับความช่วยเหลือจากบุคคลหลายฝ่าย โดยเฉพาะอย่างยิ่ง ดร สุรพันธุ์ เอื้อไพบูลย์ ที่ให้คำแนะนำเกี่ยวกับโครงงานและแหล่งข้อมูลในการค้นคว้า ทำให้ผู้จัดทำมีวิสัยทัศน์ที่กว้างขึ้น ขอขอบพระคุณอาจารย์ที่ปรึกษา คือ อาจารย์ มานิตย์ เกียรติกำจายขจร ซึ่งชี้แนะเกี่ยวกับภาษาในการเขียนโปรแกรม ขอบคุณเพื่อนๆภาควิชาอิเล็กทรอนิกส์ที่คอยช่วยเหลือสนับสนุน สุดท้าย ขอกราบขอบพระคุณ บิดา มารดา ที่คอยเป็นกำลังใจและเป็นแรงใจของผู้จัดทำเสมอมา

(น.ส. ปวิณา เตโช)

(น.ส. ปิยะวรรณ ปิยะพรมดี)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. NECTEC,สาร NECTEC ฉบับเดือน กรกฎาคม-กันยายน 2538 ,หน้า 19-23
2. NECTEC,สาร NECTEC ฉบับเดือน ตุลาคม-ธันวาคม 2538, หน้า 19-23
3. NECTEC,สาร NECTEC ฉบับเดือน มกราคม-กุมภาพันธ์ 2539 ,หน้า 13-27
4. นาย เสรี ปานซาง,วิทยานิพนธ์ เรื่อง "การรู้จำเสียงพูดคำไทยแบบไม่ขึ้นกับผู้พูดด้วยนิวรอลเน็ตเวอร์ค" โดย นาย เสรี ปานซาง ,นักศึกษาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง พ.ศ. 2540
5. IEEE,วารสาร IEEE SIGNAL PROCESSING MAGAZINE ฉบับเดือน มกราคม 1993 หน้า 8-21
6. ศ.ดร.วัลลภ สุระกำพลธร,"การประมวลผลสัญญาณเชิงเลข" สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 116หน้า, ตุลาคม 2533
7. McGraw-HILL,"Image Processing", SID-AHMED,99 p.,1994
8. Wiley,"Digital Signal Processing",Defatta Lucas

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้