# การจำแนกเสียงพูดโดยใช้นิวรอลเนทเวอร์ค

# SPEECH CODING BY NEURAL NETWORK

โดย

นาย ชัยอนันต์ ปัญญาวุธวรกุล

นาย ธวัชชัย พัฒนเสรี

วัน เดือน ปี 1 ส.ค. 2540
เลขทะเบียน 037108
เลขเรียกหนังสือ ว 98201 ช 51 ก

บริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การจำเสียงพูดโดยใช้นิวรอลเนทเวอร์ค

# SPEECH CODING BY NEURAL NETWORK


โดย

นาย ชัยอนันต์ ปัญญาวุธวรกุล 35104111

นาย ธวัชชัย พัฒนเสรี 35104183

อาจารย์ที่ปรึกษา

อาจารย์ ชินภัทร นันทจิวากรชัย


ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำเสียงพูดโดยใช้นิวรอลเนทเวอร์ค

SPEECH CODING BY NEURAL NETWORK

นาย ชัยอนันต์ ปัญญาวุธวรกุล 35104111

นาย ธวัชชัย พัฒนเสรี 35104183

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การจำเสียงพูดโดยใช้นิวรอลเนทเวอร์ค

ชัยอนันต์ ปัญญาวุธวรกุล
ธวัชชัย พัฒนเสรี
อ.ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา
ปีการศึกษา 2538

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้เป็นการศึกษาเรื่อง SPEECH CODING โดยได้พัฒนาวิธีการวิเคราะห์ข้อมูลจากที่มีผู้ใช้อยู่คือวิธี FOURIER ANALYSIS มาเป็นระบบ NEURAL NETWORK ร่วมกับวิธี LINEAR PREDICTIVE CODING เพื่อให้การคำนวนง่ายขึ้น ความเร็วในการประมวลผลเพิ่มขึ้น และมีความแม่นยำสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# SPEECH CODING BY NEURAL NETWORK

CHAIANUN PUNYAVUTVOURAKUL

TAWATCHAI PATTANASERI

CHINNAPAT NUNTAJIWAKORNCHAI ADV.

1995

ABSTRACT

This thesis is the study from Speech coding by develop in signal processing method. This system use Neural Network and Linear Predictive Coding instead of Fourier Analysis for an easier calculation, less time in processing and higher accuracy.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | |
|---|---|
| **บทที่ 1 บทนำ** | 1 |
| **บทที่ 2 ลักษณะของเสียงพูด** | 2 |
| 2.1 อวัยวะที่ใช้ในการเปล่งเสียง | 2 |
| 2.2 ลำดับการเกิดเสียง | 2 |
| 2.3 หน่วยการจำเสียงพูด | 3 |
| **บทที่ 3 Linear Predictive Coding** | 5 |
| 3.1 การแบ่ง window ของสัญญาณเสียงพูด | 5 |
| 3.2 การเน้นความสำคัญของสัญญาณ | 6 |
| 3.3 การประมาณค่าตัวแปรของ LPC | 7 |
| **บทที่ 4 Neural Network** | 10 |
| 4.1 Introduction to neural network | 10 |
| 4.2 Minimum error lerning | 12 |
| 4.3 Adaline network | 13 |
| 4.4 Backpropagation network | 17 |
| 4.5 Error และการปรับ Connection weight | 20 |
| 4.5.1 การหาค่า Error | 20 |
| 4.5.2 การปรับค่า Connection weight ใน output layer | 21 |
| 4.5.3 การปรับค่า Connection weight ใน middle layer | 21 |
| 4.6 Algorithm ของ backpropagation network | 22 |
| **บทที่ 5 หลักการทำงานของระบบ** | 25 |
| 5.1 ส่วนประกอบของระบบ | 25 |
| 5.2 การทำงานของระบบ | 25 |
| 5.3 รูปแบบภายในของแต่ละส่วน | 26 |
| 5.3.1 Window | 26 |
| 5.3.2 LPC | 26 |
| 5.3.3 Neural network | 27 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 โปรแกรมใช้งาน 27

**บทที่ 6 ผลการทดลองและสรุป** 74

6.1 การทดลอง 74

6.2 ผลการทดลอง 74

6.3 สรุป 75



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป


รูปที่ 2.1 แสดงอวัยวะที่ใช้ในการออกเสียงพูด 3

รูปที่ 3.1 การแบ่งสัญญาณด้วย Window 6

รูปที่ 3.2 Envelope ของเสียงพูด 7

รูปที่ 4.1 Neural Network 10

รูปที่ 4.2 Neurode 11

รูปที่ 4.3 Sigmoid function 12

รูปที่ 4.4 Adaline Network 13

รูปที่ 4.5 Backpropagation Network 18

รูปที่ 5.1 Block diagram 25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่ 6.1 ผลการทดลองของโปรแกรมแบบแรก 74

ตารางที่ 6.2 ผลการทดลองของโปรแกรมแบบที่สาม 75



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

ปัจจุบันได้มีการนำคอมพิวเตอร์มาใช้กันอย่างแพร่หลาย ซึ่งการติดต่อกับคอมพิวเตอร์นั้นผู้ใช้จะต้องพิมพ์คำสั่งลงไปโดยผ่านทางแป้นพิมพ์ จากนั้นได้มีการพัฒนาการติดต่อกับคอมพิวเตอร์โดยผ่านอุปกรณ์อินพุทที่เป็น Pointing device เช่น เมาส์ นอกจากนี้ยังมีการพัฒนาการติดต่อโดยใช้ระบบอื่นๆอีก ทั้งนี้ก็เพื่อให้เกิดความสะดวกในการใช้คอมพิวเตอร์ และในระยะเวลาที่ผ่านมานั้นได้มีการพัฒนาและวิจัยเพื่อที่จะติดต่อกับคอมพิวเตอร์ด้วยเสียง แต่ส่วนใหญ่จะเน้นการใช้งานกับเครื่องใหญ่ เพื่อควบคุมระบบโทรคมนาคม หรือ การพัฒนาเพื่อการรักษาความปลอดภัยทางการทหาร แต่ในปัจจุบันการพัฒนาทางด้านมัลติมิเดียเป็นไปอย่างต่อเนื่อง สามารถบันทึก เสียง ภาพวีดีโอ เพื่อใช้งานร่วมกับคอมพิวเตอร์ได้ และจากความก้าวหน้าทางเทคโนโลยีดังกล่าวนี้เองทำให้ความฝันในการที่จะติดต่อกับคอมพิวเตอร์ PC ธรรมดาด้วยเสียงเป็นจริงขึ้นมา

การติดต่อกับคอมพิวเตอร์ด้วยเสียงนั้น จะติดต่อผ่านทางSound Cardซึ่งเป็นอุปกรณ์บันทึกเสียงเพื่อใช้กับโปรแกรมมัลติมิเดีย ซึ่งการติดต่อกับความพิวเตอร์ด้วยเสียงได้นั้น นอกจากจะต้องอาศัยฮาร์ดแวร์แล้ว ยังต้องอาศัยซอร์ฟแวร์ ซึ่งเรียกว่า Speed Recognition

จุดประสงค์ของปริญญานิพนธ์นี้ก็เพื่อ สร้างซอร์ฟแวร์ ที่ใช้ในการรู้จำเสียงพูดคนซึ่งการทำซอร์ฟแวร์นั้น ต้องศึกษาถึงลักษณะในการพูดโดยทั่วไปก่อน โดยจะพิจารณาถึงอวัยวะ และ ลำดับการทำงานของอวัยวะที่ใช้ในการออกเสียง จากนั้น นำเสียงที่ได้มาคำนวณหาคุณสมบัติของเสียงโดยใช้วิธีการของ LPC (Linear Predictive Cording) เมื่อหาค่าพารามิเตอร์หรือคุณ-สมบัติของเสียงได้แล้ว ก็นำพารามิเตอร์ดังกล่าวมาใช้ในขั้นตอนการรู้จำเสียง ในขั้นตอนนี้เราจะใช้วิธีการของ Neural Network ซึ่งเป็นวิธีที่มีประสิทธิภาพสูงในการเรียนรู้และจดจำเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# บทที่ 2

## ลักษณะของเสียงพูด

โดยทั่วไปแล้วคนเราเปล่งเสียงพูดออกมาตามสำเนียงในระบบภาษาของตน แม้ว่าคนที่อยู่ในสังคมเดียวกัน ใช้ภาษาเดียวกัน แต่เสียงของแต่ละคนที่เปล่งออกมานั้น จะมีลักษณะที่แตกต่างกันแม้คนๆเดียวกันเปล่งเสียงคำเดียวกันสองครั้ง สัญญาณเสียงที่ได้ก็ยังมีความแตกต่างกัน แต่คนเราก็ยังสามารถรู้แยกแยะความหมายของคำที่เกิดจากการพูดของแต่ละคนได้ การจดจำและแยกแยะคำพูดต่างๆจึงดูเป็นเรื่องที่ง่ายในคน แต่สำหรับการที่จะให้คอมพิวเตอร์จำและแยกแยะเสียงพูดได้นั้นเป็นเรื่องที่ยากมาก ขั้นตอนแรกในการทำให้คอมพิวเตอร์จำเสียงพูดได้นั้น จึงต้องมีการศึกษาลักษณะของเสียงพูดเสียก่อน โดยเริ่มจากอวัยวะที่ใช้ในการเปล่งเสียงก่อน

### 2.1 อวัยวะที่ใช้ในการเปล่งเสียง(Articulation)

อวัยวะที่ใช้ในการเปล่งเสียงแบ่งเป็น 3 พวกใหญ่ๆ คือ

1. อวัยวะที่ใช้ในการสร้างลม คือส่วนที่ทำให้เกิดการเคลื่อนไหวของลม

2. อวัยวะส่วนที่เคลื่อนที่ได้ (Active Articulator) หมายถึงอวัยวะที่ติดกับกระดูกคางส่วนล่าง ได้แก่ ริมฝีปากล่างและลิ้น

3. อวัยวะส่วนที่เคลื่อนที่ไม่ได้ (Passiye Articulator) หมายถึงอวัยวะที่ติดกับกระดูกคางส่วนบน ได้แก่ ริมฝีปากบน ฟันบน ปุ่มเหงือก เพดานแข็ง เพดานอ่อน ลิ้นไก่ และอวัยวะที่เป็นช่องว่างได้แก่ ช่องคอ ปาก ช่องจมูก

### 2.2 ลำดับการเกิดเสียง

ลำดับการเกิดเสียงนั้นเราแบ่งออกได้เป็น 3 ขั้นตอนใหญ่ๆ คือ

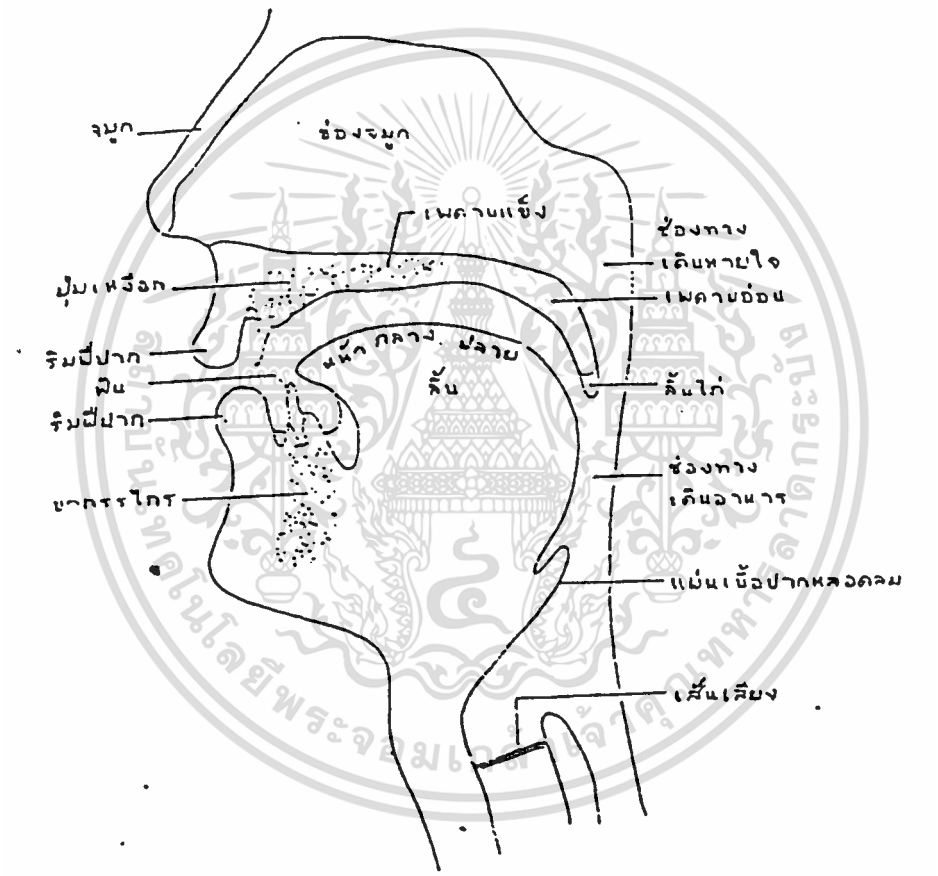
1.จุดเริ่มต้น (Initiation) อวัยวะที่ใช้ในขั้นตอนนี้คือปอด ที่ขั้นตอนนี้ลมจะถูกขับออกจากปอด

2.การดัดแปลงลมที่เส้นเสียง (Phonation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือ อวัยวะที่อยู่ต่อจากปอดขึ้นมาจนถึงกล่องเสียง เป็นขั้นตอนที่ลมจากปอดจะผ่านมาเข้าหลอดลมและกล่องเสียงซึ่งณ ที่กล่องเสียงนี้เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิด ปิดทำให้เกิดเสียงได้ 2 ชนิด คือ ถ้าเส้นเสียงเปิดตลอดเวลาที่ลมผ่าน ลมจะผ่านออกมาได้สะดวก ซึ่งจะทำให้เกิดเสียงชนิดไม่ก้อง แต่ถ้าเส้นเสียงปิดกั้นลมไว้ ลมที่ผ่านออกมาจะเพิ่มแรงดันมากขึ้นจนเส้นเสียงเปิดและปิดสลับกันไป ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดเสียงชนิดก้องและเรียกความถี่ในการเปิด ปิด ของเส้นเสียงว่า ความถี่พื้นฐาน

3.การเปลี่ยนแปลงลักษณะเส้นเสียง (Articulation) อวัยวะที่ใช้ในขั้นตอนนี้ก็คือส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปากและช่องว่างใหญ่อีก 2 ช่อง คือ ช่องปากและช่องจมูก ขั้นตอนนี้ลมที่ออกจากกล่องเสียงจะถูกแปลงให้เกิดเสียงลักษณะต่างๆ ทั้งนี้เพราะช่องว่างภายในจะถูกเปลี่ยงแปลงขนาดตลอดเวลาที่เปลี่ยนแปลงเสียง และช่องจมูกจะถูกควบคุมด้วยเพดานอ่อน (Velume) ดังนั้นถ้าช่องว่างเหล่านี้เปลี่ยนขนาดไปก็จะทำให้เกิดความถี่กำทอน (Resonant Frequency)ของเสียงเปลี่ยนแปลงไปด้วย ซึ่งช่องว่างเหล่านี้เปรียบเสมือนท่อที่มีขนาดต่างๆกัน




รูปที่ 2.1 แสดงอวัยวะที่ใช้ในการออกเสียงพูด

**2.3 หน่วยการจำเสียงพูด (Speed recognition Unit)**

หน่วยการจำเสียงพูดมีวิธีการจำแนกตามลักษณะการพูดไว้ 4 แบบคือ

1. ผู้พูดคนเดิม(Speaker Dependent Recognition System)เป็นการที่ผู้พูดคนเดิมพูด เป็นพยางค์เพียงพยางค์เดียว เป็นคำ หรือเป็นประโยค โดยเราต้องสร้างหน่วยเปรียบเทียบที่เป็นเสียงของคนคนเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น

2. ผู้พูดเป็นใครก็ได้ (Spreaker Independent Reconition System) เป็นการที่

ใครก็ได้ที่พูด จะพูดเป็นพยางค์เดียว เป็นคำ หรือเป็นประโยค

3. สอนให้เครื่องรู้จักคำในลักษณะของหน่วยเสียงหรืออย่างมากหน่วยพยางค์เป็นการสอนให้เครื่องเข้าใจในเสียงที่มีการต่อเนื่องมากกว่าหนึ่งพยางค์ ต้นแบบจะถูกเก็บไว้เป็นหน่วยเสียงหรือพยางค์เท่านั้น

4. ผู้พูดสามารถพูดแบบต่อเนื่องเหมือนการพูดปกติ(Globolization) เครื่องจะต้องรู้จักและจำแนกคำที่มีการออกเสียงเชื่อมกัน การพูดแบบนี้ในมนุษย์เป็นเรื่องง่าย แต่สำหรับคอมพิวเตอร์เป็นเรื่องที่ยากมาก เพราะการพูดแบบนี้ไม่สามารถหาเส้นแบ่งเขตระหว่างคำ หรือพยางค์ได้แม่นยำ ดังนั้นการรู้จักโดยการเปรียบเทียบกับต้นแบบจึงไม่ใช่ของง่าย วิธีการนี้ใช้กับเครื่องมินิคอมพิวเตอร์ขึ้นไป

ในปริญญานิพนธ์นี้ เราจะทำการสร้างโปรแกรมที่ใช้ในการจำเสียงพูด ที่เป็นการพูดในลักษณะพยางค์เดียว ซึ่งเป็นการพูดของใครก็ได้ คือเป็นการฝึกให้เครื่องจำเสียงพูดของเลขตั้งแต่ 0 ถึง 9 ซึ่งการฝึกจะต้องมีหน่วยเปรียบเทียบเสียงตัวเลข 0 ถึง 9 ซึ่งจะเป็นเสียงของใครก็ได้ซึ่งคำหนึ่งคำไม่ว่าจะเป็นเสียงใครนั้นจะมีคุณลักษณะที่คล้ายๆกัน ซึ่งเราสามารถที่จะวิเคราะห์ได้ด้วยวิธีการของ LPC (Linear Predictive Coding)ซึ่งจะได้กล่าวถึงในบทต่อไป เมื่อผ่าน LPCแล้วเราจะได้คุณลักษณะหรือพารามิเตอร์ของเสียงนั้น จากนั้นจึงนำพารามิเตอร์ของเสียงนั้นมาเข้ากระบวนการจดจำในกระบวนการนี้จะใช้วิธีการของ Neural Network ซึ่งเป็นวิธีที่ดีวิธีหนึ่งในการที่จะนำมาใช้ในเรื่องการรู้จำเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

## LINEAR PREDICTIVE CODING

Linear Predictive Coding (LPC) เป็นวิธีการหนึ่งที่นิยมใช้กันมากในการประมวลผลคำพูดเนื่องจากว่ามันสามารถใช้งานได้ดีและสามารถรักษาข้อมูลที่สำคัญของสัญญาณเสียงเอาไว้ได้อย่างครบถ้วน ดังนั้นมันจึงสามารถนำมาใช้ในการจำเสียงพูด, สร้างรหัส หรือทำทั้งสองอย่างได้ในระบบเดียวกัน

วิธี LPC นี้มีพื้นฐานเหมือนกับ vocoder อื่นๆ ข้อแตกต่างของ LPC ก็คือ vocal tract filter จะถูกจำลองเป็นแบบ " all pole digital filter " และรวมค่า gain(G) ไว้ใน filter สามารถเขียนเป็นสมการได้ดังนี้

$$H(z)=\frac{G}{1+a_1z^{-1}+\ldots+a_pz^{-p}}=\frac{S(z)}{E(z)}$$

โดยที่ p คือ order ของ model ถ้า s(n) เป็น output ของ model และ e(n) เป็น input จะสามารถเขียนสมการนี้ให้อยู่ในรูป time domain ได้เป็น

$$s(n)=Ge(n)-a_1s(n-1)-\ldots-a_ps(n-p)$$

สมการนี้แสดงให้เห็นถึงเหตุผลที่เรียกวิธีนี้ว่า Linear Predictive Coding เพราะว่าค่าจากการสุ่มสัญญาณทุกค่าถูกคำนวณมาจากผลรวมของการสุ่มสัญญาณก่อนหน้านี้กับค่าที่มาจากการกระตุ้น

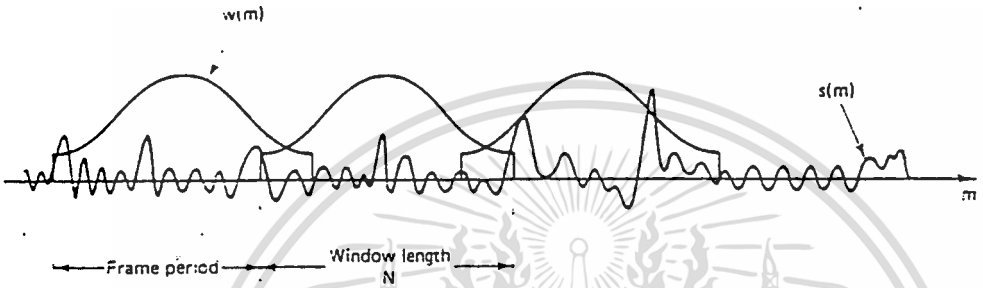
### 3.1 การแบ่ง window ของสัญญาณเสียงพูด

การนำ LPC มาใช้กับสัญญาณเสียงนั้น เราจะต้องแบ่งสัญญาณออกเป็นช่วงๆ เรียกว่า "frame" รูป 3.1 เป็นตัวอย่างของการแบ่งเสียงพูด ซึ่งแบ่งโดยคูณสัญญาณเสียง s(n) ด้วยสัญญาณ window w(n) สัญญาณ w(n) ที่นิยมใช้กันก็คือ Hamming window ซึ่งกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w(n)=0.54-0.46\cos\frac{2n}{N},\quad 0<=n<=-1$$

$$w(n)=0 \qquad \text{นอกเขตที่กำหนด}$$




รูปที่ 3.1 การแบ่งสัญญาณด้วยwindow

โดย N คือความยาวของwindow หน่วยเป็น sample ทั่วๆไปจะใช้ที่ 30ms สมมุติว่าใช้ความถี่ในการสุ่มสัญญาณเสียง 8kHz (8000sample/s) จะได้ N=240sampleและการจะทำ windowให้ได้ผลดีนั้นจะต้องให้แต่ละwindowเหลื่อมกันเล็กน้อย ซึ่งระยะห่างระหว่างwindowเรียกว่า frame period หน่วยเป็น msหรือ sample โดยทั่วไปจะใช้ประมาณ 10-30ms ถ้าเลือกค่าน้อยก็จะได้คุณภาพที่ดีกว่าเพราะจะสามารถจับการเปลี่ยนแปลงของสัญญาณได้ดีขึ้น

## 3.2 การเน้นความสำคัญของสัญญาณ

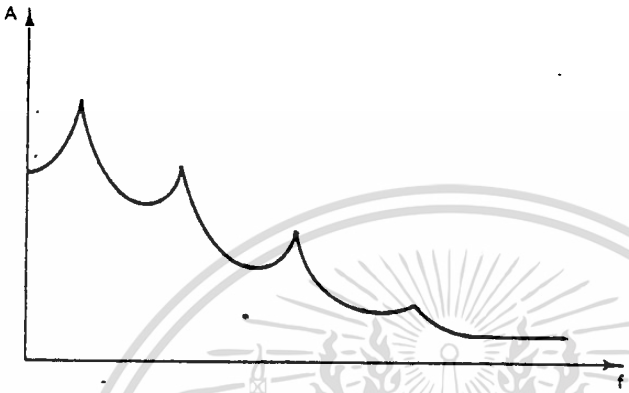
รูป3.2แสดง spectram envelope ของสัญญาณเสียงพูด ซึ่งspectrumจะลดลงที่ความถี่สูงถ้านำไปใช้กับLPC LPCจะทำงานได้ดีที่ช่วงความถี่ต่ำแต่ที่ความถี่สูงจะทำงานได้ไม่ดี เพื่อป้องกันไม่ให้เหตุการณ์นี้เกิดขึ้นเราจะนำสัญญาณเสียงผ่านfilterที่มีtranfer function $1-az^{-1}$เรียกว่า preemphasis filter โดยทั่วไปจะใช้ค่าaประมาณ0.9โดยเฉพาะที่ค่าa=15/16=0.9375 จะได้สัญญาณ preemphasized s'(n) เป็น

$$s'(n)=s(n)-0.9375s(n-1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ประมวลผลแล้ว สัญญาณจะถูกทำให้กลับเป็นอย่างเดิมโดย

$$s(n)=s'(n)+0.9375s(n-1)$$




รูปที่ 3.2 envelopeของเสียงพูด

## 3.3 การประมาณค่าตัวแปรของLPC

การจะหาค่าตัวแปรต่างๆของLPCนั้น เราจะกำหนด

$$s''(n)=-a_1s(n-1)-\ldots-a_ps(n-p)$$

เป็นการประมาณค่าของs(n)จากsampleก่อนหน้านี้ ดังนั้น Error

$$\sum [s(n)-s''(n)]^2$$

จะมีค่าน้อยที่สุดสำหรับทุกๆsample การหาค่า $a_i$จะมาจากการทำให้ผลรวมของกำลังสอง ของค่าErrorมีค่าน้อยที่สุด จากสมการเชิงเส้น

$$a_1r(0)+a_2r(1)+\ldots+a_pr(p-1) =-r(1)$$

$$a_1r(1)+a_2r(0)+\ldots+a_pr(p-2) =-r(2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_1r(p-1)+a_2r(p-2)+\ldots a_pr(0) = -r(p)$$

หรืออยู่ในรูปmatirx

$$R \cdot a = -r$$

โดยที่ $r^T=[r(1)\ r(2)\ldots r(p)]$

$a^T=[a_1\ a_2\ldots\ a_p]$

$$R=\begin{bmatrix} r(0) & r(1) & \ldots & r(p-1) \\ r(1) & r(2) & \ldots & r(p-2) \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ r(p-1) & r(p-2) & \ldots & r(0) \end{bmatrix}$$

จากสมการข้างต้นเราจะกำหนด

$$r(i)=r(-i)=\sum_{n=0}^{N-i-1} s(n)s(n+i)$$

เป็นค่า "autocorrelation" ลำดับที่ i จากสูตรนี้สัญญาณs(n)จะต้องผ่านการwindow แล้ว ดังนั้น autocorrelation ที่ iจะคำนวณโดย เลื่อนสัญญาณไป i sample คูณกับสัญญาณขณะที่ยังไม่ได้เลื่อน แล้วนำผลทั้งหมดมารวมกันดังรูป5.4 วิธีนี้เรียกว่าวิธี autocorrection

ซึ่งมันจะสร้างmatrix R ที่เป็น Toeplitz matrix ซึ่งก็คือmatrixที่ค่าในแนวแทยงของมันจะเป็นค่าตัวแปรตัวเดียวกัน และเพราะว่าmatrixนี้เป็นmatrixแบบ nonsingular และสามารถ invertได้ ดังนั้นเราจะสามารถหาคำตอบของสมการได้อย่างแน่นอน

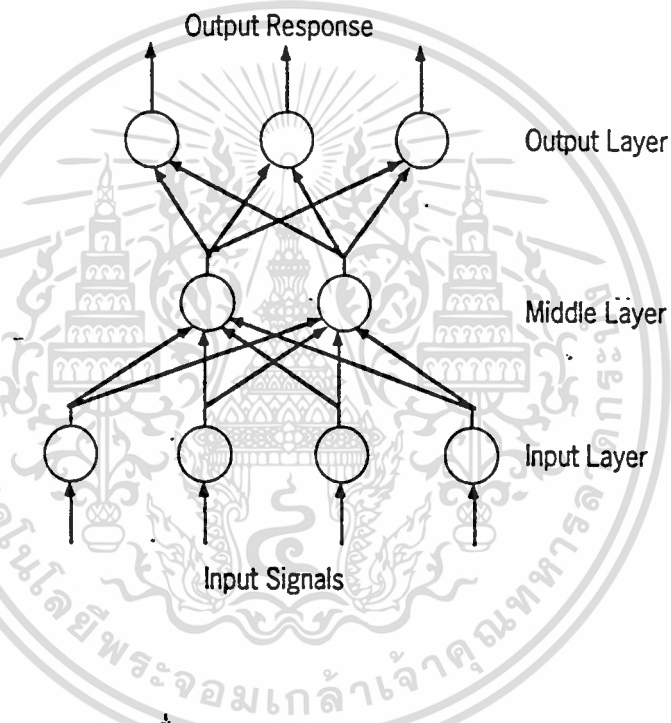
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# NEURAL NETWORK

## 4.1 Introduction To Neural Network

Neural Network คือ วิธีการของระบบการประมวลผลข้อมูลวิธีหนึ่ง ซึ่งระบบNeural Network นี้จะประกอบไปด้วย การติดต่อกันเป็นจำนวนมากของหน่วยประมวลเซลล์เดียวที่เรียกว่า Neurode แต่ละ Neurode นั้นจะติดต่อเชื่อมกันด้วย Connection Weight

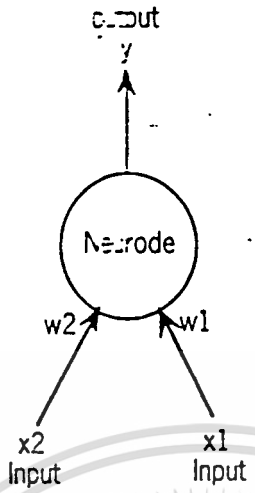



รูปที่ 4.1 neural network

จากรูปที่ 4.1 เป็นรูปโครงสร้างชนิดหนึ่งของ Neural Network ในรูปที่ 4.1นี้เป็น Network ชนิดที่มี 3 Layerโดยจากรูปที่ Input Layer จะเป็นส่วนที่รับสัญญาณ Input จากภายนอกจากนั้นสัญญาณจะถูกส่งผ่าน Connection Weightไปแต่ละ Neurode ในชั้น Middle Layer ที่ Middle Layer นี้แต่ละ NeurodeจะทำการรวมสัญญาณจากInput ที่ถูกถ่วงน้ำหนักแล้วโดยผลรวมของสัญญาณจะเรียกว่า NetInput จากนั้น NetInput ของแต่ละ Neurode ในชั้น Middle Layer จะถูกถ่วงน้ำหนักก่อนจะส่งไปยังแต่ละ Neurodeของ Ourput Layer จากนั้นแต่ละ Neurode ของชั้น Output Layer จะทำการประมวลผลโดยการรวมสัญญาณที่ได้เพื่อคำนวนหาค่า Output ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาแต่และ Neurode ที่ใช้ประมวลผลสัญญาณดังรูปที่ 4.2




รูปที่ 4.2 Neurode

หลักการทางคณิตศาสตร์ที่ใช้อธิบายการประมวลผลของ Neurode แต่ละ Neurode นั้น เราเรียกว่า Transfer Function โดยมีขั้นตอนด้วยกัน 3 ขั้นตอน

1. Neurode จะคำนวนหา Net Input ที่ได้มาจากสัญญาณอินพุท

$$I_i = \sum_{i=1}^{n} (w_{ji})(x_j)$$

$I_i$ คือ Net Input ที่ได้จาก Neurode i

$w_{ji}$ คือ Connection Weight จาก Neurode i ถึง Neurode j

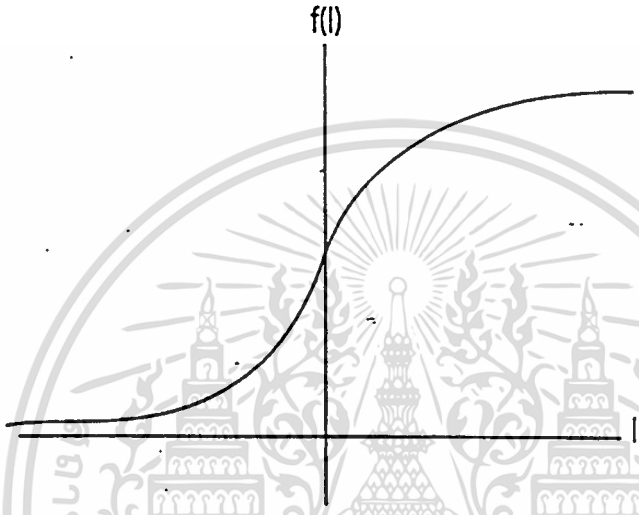
$x_j$ คือ สัญญาณอินพุทของแต่ละ Neurode j

2. เมื่อ Neurode แต่ละ Neurode คำนวนผลรวมของ Net Input จะนำค่า $I_i$ ที่ได้มาทำการเปลี่ยนให้เป็น Activation Level โดยฟังก์ชั่นที่ใช้ในระบบ Neural Network นั้นมีหลายฟังก์ชั่นด้วยกันขึ้นอยู่กับแต่ละรูปแบบวิธีของNeural Network แต่ในที่นี้จะใช้ฟังก์ชั่นแบบ Sigmoid Function ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$f(I) = \frac{1}{1+e^{-I}}$ , e คือค่าเอกซ์โปรเนนเชียล

$\frac{df(I)}{dI} = f(I)(1-f(I))$




รูปที่ 4.3 Sigmoid funtion

3. ขั้นตอนนี้ จะนำค่า f(I) ที่ได้มาเปลี่ยนเป็นค่า Y หรือ OutPut

$Y_i = f(I)$ , $f(I)>T$

$0$ , อื่นๆ

T เป็นค่า Threshold Value

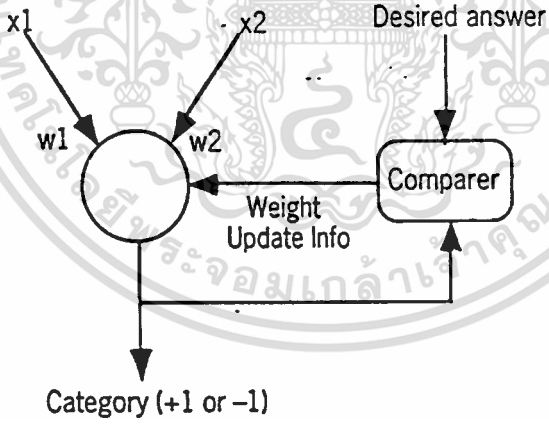
**4.2 Minimum Error Learning**

สมมติว่าต้องการสร้างระบบ Neural Network ที่สามารถแยกแยะจัดInput Pattern ให้เป็น 2 กลุ่มคือ A และ B โดยเราจะทำการฝึกให้ระบบสามารถจำ Input Pattern แต่ละตัวและสามารถแยกแยะได้ว่า Input Patternนั้นอยู่ในกลุ่มใด(A หรือ B) วิธีการอย่างหนึ่งที่ใช้คือ Minimum Error Learning คือ ในช่วงการฝึกให้ระบบจำนั้นระบบ Neural Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะรับค่า Input Pattern ค่าหนึ่งมาซึ่ง Input ค่านี้เราจะต้องกำหนดแยกแยะให้อยู่ในกลุ่มใด กลุ่มหนึ่งก่อน ( A หรือ B ) จากนั้นจะนำค่า Input ค่านี้มาทำการคำนวนหาค่า Outputของระบบ Neural Network หลังจากได้ค่า Output แล้วจะนำค่าที่ได้ไปเปรียบเทียบกับ Output ที่เรากำหนดไว้ก่อนแล้ว (A หรือ B) ค่าที่ได้จากการเปรียบเทียบก็คือค่า Error โดยเราจะนำค่า Error ที่ได้นี้ไปคำนวนค่าการปรับ Connection Weightเมื่อปรับไป 1 ครั้งแล้วก็จะรับ Input Pattern ตัวเดิมมาคำนวนใหม่เพื่อปรับ Connection Weight ต้องทำซ้ำอย่างนี้ไปเรื่อยๆจนกว่าค่า Error ของความแตกต่างของ Outputจากระบบกับ Output ค่าที่เรากำหนดไว้ก่อน (A หรือ B) แตกต่างกันน้อยที่สุดที่เรายอมรับได้ เมื่อได้ค่า Error ที่ยอมรับได้แล้ว ก็แสดงว่าระบบสามารถจะจำค่าหรือแยกแยะค่า Input ค่านั้นได้แล้วว่าอยู่ในกลุ่มใด ก็จะทำการรับ Input ค่าใหม่มาทำการคำนวนเพื่อจำต่อไป และระบบของNeural Network ที่ใช้วิธีการของ Minimum Error ก็คือ Adaline

## 4.3 Adaline Network




รูปที่ 4.4 Adaline Network

จากรูปที่4.4 เป็นรูปที่แสดงNeurode จำนวน 1 Neurode โดย Neurode นี้จะคำนวนค่า Net Input (I) แล้วนำไปเปรียบเทียบกับ Desired Output เพื่อหา Errorแล้วนำไปคำนวนการปรับ Connection Weight ต่อไปเราต้องการแบ่งกลุ่มของ Inputออกเป็น 2 กลุ่ม คือ A กับ B ( Desired Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย $A = 1$

$B = -1$

$$I = \sum_{i=1}^{n} (w_i)(x_i)$$

จากนั้นนำค่า I ที่ได้ไปหาค่าเอาท์พุท(Y) จาก

$I > 0$ ,Y = 1 ( Actual Output(Y))

$I <= 0$ ,Y = -1 ( Actual Output(Y))

เมื่อได้ค่า Output (Y) แล้วนำไปหาค่า Error

Error = Desired Output - Actual Output

จากนั้นนำค่า Error ที่ได้ไปคำนวนการปรับ Connection Weightกรณีของ Adaline จะหาค่า การปรับ Connection weight จาก Delta Rule

$$w = w^{old} + \beta \frac{EX}{x^2}$$

$\beta$ คือ ค่าคงที่การเรียนรู้มีค่าอยู่ระหว่าง 0 ถึง 1

x คือ อินพุท

E คือ ค่า Error มีค่า 2,-2,0

ระบบจะทำการปรับ Connection Weightไปเรื่อยๆจนกว่าจะจำ Input ค่านี้ได้นั่นก็คือในระบบ Adaline ระบบจะจำได้เมื่อ E = 0 เมื่อระบบจำ Input ค่าแรกได้ก็จะรับค่า Input ค่าต่อไปและจะทำการปรับ Connection Weightเพื่อจำ Input ตัวที่ 2 ปรับจนจำได้เมื่อจำได้แล้วต้องกลับไปตรวจสอบ Inputค่าแรกใหม่ว่ายังจำได้หรือไม่ ถ้าจำได้ก็รับ Input 2 ตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีก ถ้ายังจำได้ก็รับ Input3 เพื่อมาทำการปรับค่า Connection Weightไปเรื่อยๆ ถ้าขั้นตอนไหนจำไม่ได้ก็ให้ปรับจนจำได้แล้วกลับไปตรวจสอบใหม่ตั้งแต่ค่า Input ค่าแรก

Pseudocode ของระบบ Adaline

```
for(each pattern)in the Training Set do
  {
     apply the pattern to the adaline
     compute the adaline's net input, I
     if (I>0) out +1;else output -1
     compute error,E,as desired output - actual output
     if(E <> 0) do
          {
          for (this pattern) back thru all previously trained
          pattern, do
          /* if adjust the weights for this pattern,must reconfirm
          that all previous pattern are still ok */
          {
          apply the pattern to adaline
          compute the adaline's net input,I
          if(I>0) output +1;else output -1
          compute error,E,as desired output - actual output
          while (E<>0) do
          /* adjust the weight using delta rule till this pattern
          is correct*/
               {
               compute square of lenght of input vector(L)
               for each element of weight vector,w_i do
               {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            change in w_i = β*E*x_i/L
            w_i = w_i+<change in w_i>
            } /* end for each element of weight vector */

        compute the adaline's net input,I,for this same input
        vector
        (to see if done)
        if(I> 0) output +1;else output -1
        compute error,e,as above
        } /* end while error not equal to zero */
        } /* end for all previously trained pattern */
    } /* end if original error was not zero */
} /* end for all patterns in training set */
```

---

ระบบของ Neural Network นั้นมีอยู่หลายวิธีด้วยกันคือ

1. Adaline
2. Outstar
3. Kohonen
4. Attractor
5. Backpropagation

ซึ่งแต่ละระบบก็มีวิธีการแตกต่างกันแต่ที่กล่าวไปในตอนต้นแล้วนั้นคือ Adaline Network ซึ่งเป็นวิธีที่ยังไม่ดีพอ และไม่เป็นที่นิยมใช้กัน แต่ Adaline ก็เป็นพื้นฐานของอีกวิธีหนึ่งซึ่งเป็นวิธีที่นิยมใช้กันมากนั้นก็คือ Backpropagation Network ซึ่งวิธี Backpropagation Network นี้ก็เป็นวิธีที่นำมาใช้ในปริญญานิพนธ์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 Backpropagation Network

จากที่กล่าวมาแล้วว่า Adaline Network ยังเป็นวิธีที่ไม่ดีพอคือ Adaline นั้นมีอยู่ 2 Layer ด้วยกันคือ Input Layer และ Output Layer ซึ่งถ้าหากดูที่สมการในการหาค่า Net Input (I) คือ

$$I = \sum_{i=1}^{n} (w_i)(x_i)$$

$w_i$ คือ ค่า Connection Weight จาก Neurode อื่นมายัง Neurode i

$x_i$ คือ ค่า Input Pattern ของ Neurode i

จะเห็นได้ว่าเป็นสมการเชิงเส้น นั่นก็คือ Adaline จะแยกแยะ Input Pattern ออกเป็น 2 กลุ่มโดยใช้เส้นตรง ซึ่งทำให้ Adaline เป็นวิธีที่ไม่ยืดหยุ่นพอที่จะนำมาใช้ประโยชน์ได้จริง

ปัญหาดังกล่าวแก้ได้โดยใช้ฟังก์ชั่น Sigmoid ซึ่งจะมี f(I)เป็น S-shape

$$f(I) = \frac{1}{1+e^{-I}}$$

I คือ Netput

e คือ ค่าเอกซ์โปรเนนเชียล

ซึ่งวิธี Backpropagationก็ใช้ Sigmoid Function ด้วยโดยที่ Backpropagation Network เป็น Neural Network ที่มีด้วยกัน 3 Layer คือ

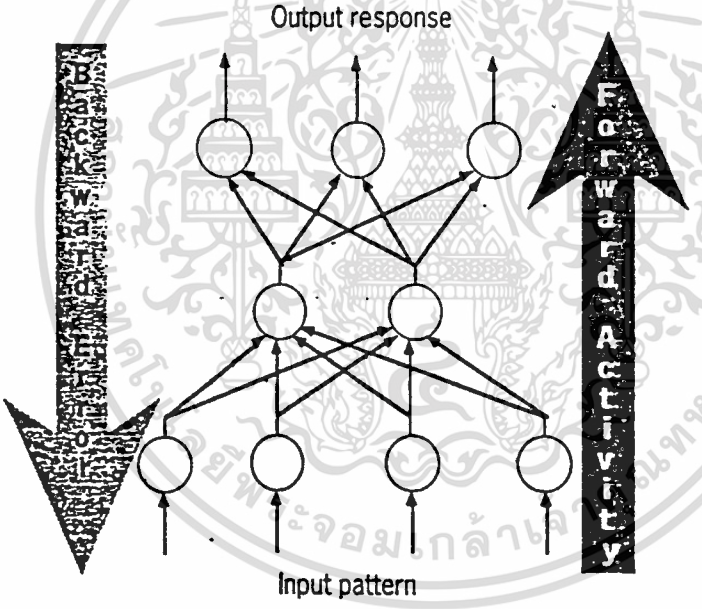
1. Input Layer มีจำนวนของ Neurode ตามจำนวน Input 1ชุด (N) ตามที่เราต้องการ เช่น Input Pattern ของเราอาจจะเป็นข้อมูลชุดหนึ่งซึ่งข้อมูลชุดนี้แทนคุณสมบัติของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียง 1 คำซึ่งข้อมูลคุณสมบัติของเสียง 1 คำนี้มีจำนวน N ค่า เพราะฉะนั้นใน Input Layer จะต้องมีจำนวน Neurode จำนวน N Neurode

2. Middle Layer ปกติจะมีจำนวนของ Neurode น้อยกว่าในชั้นของ output Layer ซึ่งโดยทั่วไปถ้าชั้น output Layer มีจำนวน Neurode = M ชั้น Middle Layerนี้จะมีจำนวน Neurode = M-1

3. Output Layer มีจำนวน Neurode ตามจำนวนของ Output ที่เราต้องการ เช่น เราจะสร้าง Network ที่สามารถแยกแยะเสียงคำได้ 10 คำในชั้น Output อาจมีจำนวนของ Neurode เป็นจำนวน4 Neurode คือค่าOutput ของแต่ละ Neurode จะเป็น 0000,0001, 0010,0011,0100,0101,0110,0111,1000,1001โดยเรากำหนดว่าแต่ละคำจะตรงกับรหัสอะไร หรืออาจจะมี Neurode เป็นจำนวน 10 Neurode เลยก็ได้




รูปที่ 4.5 Backpropagation Network

จากรูป Backpropagation Network จะเห็นได้ว่ามีชั้นของ Middle Layer เพิ่มมาจาก Adaline Network นี่เองที่ทำให้ Backpropagation ต่างจาก Adaline และชั้นของ Middle Layer นี้เองที่ทำให้เกิดปัญหาขึ้นมาคือ ใน Adaline เราหาค่าError ของชั้นเอาท์พุทได้นั้นเพราะเรารู้ Desired Output ของชั้น Output Layer แต่ใน Backpropagation นั้นเรารู้ค่า Desired Output ของชั้น OutputLayerแต่เราไม่รู้ค่า Desired Output ของชั้น Middle Layer วิธีการที่ใช้ใน Backpropagation ก็คือการหาค่า Error ย้อนกลับซึ่งจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้กล่าวถึงต่อไป

ที่กล่าวไปแล้วนั้นว่า Backpropagation Network นั้นคล้ายกับ Adaline Network นั่นก็คือ การหาค่า Net Input(I)ของแต่ละ Neurode ในแต่ละ Layerต่างๆนั้นใช้สูตร

ให้ i แทนแต่ละ Neurode ในชั้น Middle Layer

ให้ k แทนแต่ละ Neurode ในชั้น Input Layer

Net Inputของชั้น Middle Layer คือ $I_i$

ให้ $n_k$ แทนจำนวน Neurode ในชั้น Input Layer

$$I_i = \sum_{k=1}^{n_k} (w_{ki})(x_k)$$

ให้ j แทนแต่ละ Neurode ในชั้น Output Layer

Net Input ของชั้น OutPut Layer คือ $I_j$

ให้ $n_i$ แทนจำนวน Neurode ในชั้น Middle Layer

$$I_j = \sum_{i=1}^{n_i} (w_{ij})(I_i)$$

นำค่า I ที่ได้มาคำนวนโดยฟังก์ชั่น Sigmoid

$$f(I) = \frac{1}{1+e^{-I}}$$

โดย e คือค่าเอกซ์โปรเนนเชียล

โดยที่

$$df(I) = \frac{e^{-I}}{(1+e^{-I})^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037108

$$= \frac{((1+e^{-I})-1)}{(1+e^{-I})^2}$$

$$= \frac{1}{(1+e^{-I})}\left(1-\frac{1}{(1+e^{-I})}\right)$$

$$= f(I)(1-f(I))$$

4.5.1 การหาค่า Error

การที่เราจะทำการปรับ Connection Weight ได้นั้นจะต้องรู้ค่า Error ที่หาได้จาก

Error = Desired OutPut - Actual OutPut

แต่เนื่องจากใน BackPropagation Network นั้นมีชั้น Middle Layer เพิ่มขึ้นมาทำให้เราไม่สามารถหาค่า Error ของชั้น Middle Layer ได้เนื่องจากเราไม่รู้ค่า Desired OutPut ของชั้น Middle Layer ดังนั้นการจะหาค่า Error ของชั้น Middle Layer ได้นั้นต้องใช้สูตรการหาค่า Error ย้อนกลับ

การหาค่า Error ย้อนกลับ หรือ Backpropagation Error หาได้จากสูตร

$$E_j = Y_j(\text{Desired}) - Y_j(\text{Actual})$$

$$E_i = \frac{df(I_i)}{dI}\sum_{j=1}^{n_j}(w_{ij})(E_j)$$

$$Y = f(I) \quad ,f(I)>T$$

$$0 \quad ,f(I)<=T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$n_j$ คือ จำนวน Neurode ในชั้น Output Layer

T คือ ค่า Threshold Value

$E_j$ คือ ค่า Error ของแต่ละ Neurode ในชั้น Output Layer

$E_i$ คือ ค่า Error ของแต่ละ Neurode ในชั้น Middle Layer

$Y_j$ คือ ค่า Outputของแต่ละ Neurode ในชั้น Output Layer

$Y_i$ คือ ค่า Outputของแต่ละ Neutode ในชั้น Middle Layer

$w_{ij}$คือ ค่า Connection Weight จาก Neurode i ไปยัง Neurode j

$I_i$ คือ ค่า Net Input ของชั้น Middle Layer

4.5.2 การปรับ Connection Weight ใน Output Layer

$$w_{ij} = \beta * E_j * f(I_i)$$

หรือ

$$w_{ij} = w_{ij}^{old} + \beta * E_j * f(I_i)$$

4.5.3 การปรับ Connection Weight ใน Middle Layer

$$w_{ki} = \beta * E_i * f(X_k)$$

หรือ

$$w_{ki} = w_{ki}^{old} + \beta * E_i * f(X_k)$$

k คือ Neurode ในชั้น Input Layer

i คือ Neurode ในชั้น Middle Layer

j คือ Neurode ในชั้น output Layer

$n_j$ คือ จำนวน Neurode ในชั้น Output Layer

T คือ ค่า Threshold Value

$E_j$ คือ ค่า Error ของแต่ละ Neurode ในชั้น Output Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$E_i$ คือ ค่า Error ของแต่ละ Neurode ในชั้น Middle Layer

$w_{ij}$ คือ ค่า Connection Weight จาก Neurode i ไปยัง Neurode j

$w_{ki}$ คือ ค่า Connection Weight จาก Neurode k ไปยัง Neurode i

$I_i$ คือ ค่า Net Input ของชั้น Middle Layer

$X_k$ คือ ค่า Input Pattern ที่เข้ามาในแต่ละ Neurode k ในชั้น Input Layer

$\beta$ คือ ค่า การเรียนรู้มีค่า 0 ถึง 1

## 4.6 Algorithm ของ BackPropagation Network

กำหนดค่า Minimum Error ที่เราสามารถยอมรับได้ {ปกติจะค่าที่น้อยกว่า 10% ของ Desired Output}

```
Repeat
     begin
     Total Error = 0
     for Input Xi แต่ละชุด do
     begin
          for each Neurode in Middle Layer do
              begin
                 หาค่า Ii
                 หาค่า f(Ii)
                 เก็บค่า Ii
              end
          for each Neurode in Output Layer do
              begin
                 หาค่า Ij
                 หาค่า f(Ij)
                 เก็บค่า Ij
              end
          for each Neurode in Output Layer do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin
        หา Ej
        Total Error = Total Error+Ej
    end
  for each Neurode in Middle Layer do
   begin   nj
        หา ∑ wijEj
           j=1
        หา Ei
    end
  for each Neurode in Output Layer do
    begin
       for each Weight from Middle Layer do
         begin
            หา Δwij โดย β*Ej*f(Ii)
            หา wij
         end
     end
   for each Neurode in Middle Layer do
     begin
        for each Weight from Input Layer do
          begin
             หา Δwki โดย β*Ei*f(Xk)
             หา wki
          end
      end
  end {จบการคำนวนของชุด Input Pattern ทุกชุด}
end{จบ Repeat..Until}
Until Total Error < ค่า Error ที่ยอมรับได้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อที่น่าสังเกตคือ Adaline จะรับ Input Pattern 1 Pattern แล้วทำการปรับค่าของ Connection Weight จนจำค่า Input Pattern ค่านั้นได้จึงรับ Input Pattern ค่าต่อไปมาปรับค่า Connection Weight ต่อไป แต่ใน Backpropagation นั้น จะรับ Input Pattern เข้ามาปรับ Connection Weight 1 ครั้งแล้วรับ Input Pattern ตัวต่อๆไปเข้ามาปรับ Connection Weight จนครบทุกๆInput Patternและระหว่างปรับค่า Connection Weight แต่ละ Input Pattern นั้นก็รวมสะสมค่าError ของแต่ละ Input Pattern ทุกๆ ชุดจากแต่ละ Neurode ในชั้น OutPut Layer ไว้ในTotal Errorเมื่อรับ Input Pattern เข้ามาหมดทุกๆ Pattern แล้วก็จะได้ค่า Total Error ค่าหนึ่ง นำค่านี้ไปเปรียบเทียบกับค่า Error น้อยที่สุดที่เรายอมรับได้ ถ้าค่า Total Error นั้นมีค่าน้อยกว่าค่าที่กำหนดไว้ Network ก็จะหยุดถือว่าจำ Input Pattern ทุกตัวได้แล้ว แต่ถ้าค่า Total Errorนั้นมีค่ามากกว่าค่าที่กำหนดไว้Network ก็จะเซตค่า Total Errorเท่ากับ 0 แล้ววนกลับไปรับ Input Pattern ตั้งแต่ตัวแรกจนตัวสุดท้ายใหม่เพื่อรวบรวมค่า Total Error เพื่อมาเปรียบเทียบกับค่าที่กำหนดไว้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

# หลักการทำงานของระบบ

ที่กล่าวมาในบทก่อนหน้านี้เป็นรายละเอียดในส่วนของ Neural Network และ Linaer Predictive Coding ในบทนี้จะเป็นการนำแต่ละส่วนมาประกอบกันและเพิ่มบางส่วนเข้าไปเพื่อให้เหมาะสม

## 5.1 ส่วนประกอบของระบบ

ส่วนประกอบของระบบสามารถแบ่งได้เป็น

1.ส่วนที่เป็น hardware คือคอมพิวเตอร์และ sound card

2.ส่วนที่เป็น software คือ

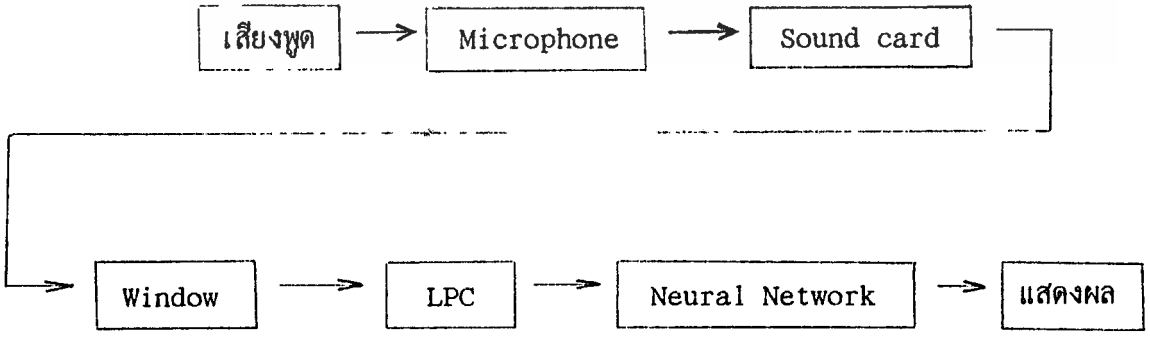
-Windowing

-Linear Predictive Coding

-Neural Network

## 5.2 การทำงานของระบบ

สามารถเขียน block diagram แสดงการทำงานได้ดังนี้




รูปที่ 5.1 block diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถอธิบายการทำงานได้คือ เมื่อเราพูดผ่านทางไมโครโฟน ไมโครโฟนจะเปลี่ยนสัญญาณเสียงให้เป็นสัญญาณไฟฟ้า จากนั้นจะส่งสัญญาณนี้เข้าสู่ sound card ซึ่งจะเปลี่ยนสัญญาณนี้ให้เป็นไฟล์ของคอมพิวเตอร์สกุล WAV นำไฟล์ที่ได้มาทำการแบ่งเป็น window แล้วคำนวนหาค่าพารามิเตอร์ของ LPC สำหรับแต่ละ window แล้วจึงนำพารามิเตอร์เหล่านั้นมาป้อนเข้าสู่ระบบ neural network จากนั้นก็จะแสดงผลออกมา

แต่ก่อนที่จะใช้งานได้นั้นจะต้องมีการเทรนระบบเสียก่อน โดยนำไฟล์เสียงที่เป็นเสียงที่ต้องการให้ระบบจำมาป้อนให้กับระบบพร้อมกับผลลัพธ์ที่ต้องการสำหรับเสียงนั้นๆ และควรจะใช้เสียงจากหลายๆคนมาป้อนเพื่อให้มีความหลากหลาย

## 5.3 รูปแบบภายในของแต่ละส่วน

จากรูปที่ 5.1 แสดงส่วนต่างๆของระบบ ซึ่งส่วนที่เป็นไมโครโฟนและ sound card นั้นเป็นอุปกรณ์ที่มีขายทั่วไปอยู่แล้ว และเป็นมาตรฐาน ดังนั้น ในปริญญานิพนธ์นี้จะทำเฉพาะส่วนของ software เท่านั้น ในหัวข้อนี้จะแสดงถึงการจัดการข้อมูลของแต่ละส่วน

### 5.3.1 window

เนื่องจากว่าแต่ละไฟล์แม้จะเป็นเสียงคำๆเดียวกันแต่ก็จะมีขนาดต่างกัน ดังนั้นจะต้องหาวิธีจัดการในส่วนนี้ ก็จะมีอยู่ 2 วิธีคือให้จำนวน window คงที่แล้วปรับขนาดของ window หรือให้ขนาดของ window คงที่แล้วปรับจำนวน window ในที่นี้จะเลือกใช้วิธีหลังโดยกำหนดให้ขนาดของ window เป็น 330 samplings และframe period เป็น 300 samplings โดยสามารถหาจำนวน window ได้จาก

จำนวน window = (((filesize-330) div 300)+1)

### 5.3.2 LPC

ในการที่จะคำนวนหาค่าพารามิเตอร์ 'a' โดยตรงด้วยการทำ inverse matrix นั้นจะทำได้ยากและเสียเวลา และเราต้องการพารามิเตอร์แค่ 2 ตัวคือ $a_1$ และ $a_2$ เท่านั้น ดังนั้นจะใช้วิธีการของ Durbin แทนโดย

$$E[0]= r[0]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$K_i = \frac{-r[i]+a_1[i-1]r[i-1]+\ldots+a_{i-1}[i-1]r[1]}{E[i-1]} \text{ โดย } i=1,..,p$$

$$a_i[i]= K_i$$

$$a_j[i]= a_j[i-1]+K_i a_{i-j}[i-1] \quad j=1,..,i-1$$

$$E[i]= (1-K_i)E(i-1)$$

$$a_j= a_j[p] \quad j=1,..,p$$

โดยที่ $a_j$ คือสัมประสิทธ์ของ order ที่ p

$K_i$ เรียกว่าสัมประสิทธ์ PARCOR

ซึ่งจะแบ่งค่า $K_1$ และ $K_2$ ออกเป็นกลุ่มทั้งหมด 88 กลุ่ม แล้วนำค่า $K_1$ และ $K_2$ ของแต่ละ window มาพิจารณาว่าอยู่ในกลุ่มใด จากนั้นจะนับว่าในแต่ละกลุ่มมีจำนวนของค่า K อยู่เท่าใดแล้วจึงนำมาป้อนให้กับ neural network

5.3.3 Neural Network

ในระบบนี้จะมี neurode ในชั้น Input layer 88 ตัว ในชั้น Middle layer 9 ตัวและในชั้น Output layer 10 ตัว ซึ่งในชั้น Output layer นี้กำหนดให้ neurode แต่ละตัวแทน Output 1 ค่า เช่นถ้า neurode ตัวที่ 1 มีค่า output มากที่สุดก็แสดงว่า input ที่ป้อนเข้ามานั้นถูกระบบวิเคราะห์ว่าเป็นเสียงเลข 1

## 5.4 โปรแกรมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program back propagation;
uses crt;
type xxxxx=array[0..329] of real;          {lpc in 1 window}
     rr=array[0..2] of double;
     parcor=array[1..2] of double;
     inp=array[1..11,1..8] of integer;    {output from lpc to neural}
     woo=array[1..9,1..10] of extended;   {output layer weight}
     wmm=array[1..88,1..9] of extended;   {middle layer weight}
     inback=array[1..88] of integer;      {input to neural}
     fi_by=file of byte;                  {file.wav}
var
input:inp:
k:parcor;
r:rr;
Sn:xxxxx;
SnO:real;
fi_lpc:fi_by;
filewav,filew:string[15];
fi_in:file of integer:
how,number,size,windows,more,poitt:integer;
i,j,choice,time:integer;
ch:byte;

{variable of backpropagation}
wo:woo;
wm:wmm;
x:inback;
wco:array[1..9,1..10] of extended;    {change weight in output layer}
wcm:array[1..88,1..9] of ewtended;    {change weight in middle layer}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
im:array[1..9] of extended;          {net from middle}
io:array[1..10] of extended;         {net from output}
fim:array[1..9] of extended;         {function exponential of middle}
fio:array[1..10] of extended;        {function exponential of output}
fx:array[1.88] of extended;          {function exponential of input}
eo:array[1..10] of extended;         {error of output layer}
em:array[1..9] of extended;          {error of middle layer}
smaxerr:array[1..60] of extended;    {max error in one pattern}
n:array[1..10] of extended;
mwe:array[1..9] of extended;
ya:array[1..10] of extended;         {actual output}
yd:array[1..10] of extended;         {desired output}
totalerror,n,maxerr:extended;
error,beta,u:real;
turn,a,b:integer;
c,y:char;
file_var:file of extended;
filename:string[10];


function Wn(n:integer):real;     {hamming window}
begin
  Wn:=(0.54-(0.46*(cos(6.28318*n/330))));
end;
procedure lowpass(Sn0:real;var Sn:xxxxx);
var n:integer;
    hamming:real;
    Sn2:real;
    out_Sn:real;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    Sn:=Sn0;
    for n:=0 to 239 do
    begin
      out_Sn:=Sn[n]-(0.9375*Sn2);
      Sn2:=Sn[n];
      hamming:=Wn(n);
      Sn[n]:=out_Sn*hamming;
    end;
end;

procedure make_ri(var Sn:xxxxx);
var i:integer;
    n:integer;
    qq:double;
begin
  for i:=0 to 2 do
    begin
      r[i]:=0;
      for i:=0 to (330-i-1) do
        begin
          qq:=(Sn[n]*Sn[n-i];
          r[i]:=r[i]+qq;
        end;
    end;
end;
if  (0.25  <= k[2]) and (k[2] <  0.5) then
                                begin
                                  input[b,6] :=input[b,6] +1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    end;
if  (0.5<= k[2]) and (k[2] <  0.75)  then
                    begin
                      input[b,7] :=input[b,7]  +1;
                    end;
if  (0.75 <=k[2]) and (k[2] <  1)  then
                    begin
                      input[b,8] :=input[b,8]  +1;
                    end;

end;

procedure count1(k:parcor);
var  b:integer;
begin
  if  (-0.3 >= k[1]) and (k[1] > -0.5)  then
                      begin
                        b:=1; count(k,b);
                      end;
  if  (-0.5>= k[1]) and (k[1] > -0.6) then
                    begin
                      b:=2; count(k,b);
                    end;
  if  (-0.6>= k[1]) and (k[1] > -0.66)  then
                    begin
                      b:=3; count(k,b);
                    end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if  (-0.66>= k[1]) and (k[1] > -0.72)  then
                        begin
                          b:=4; count(k,b);
                        end;
if  (-0.72>= k[1]) and (k[1] > -0.78)  then
                        begin
                          b:=5; count(k,b);
                        end;
if  (-0.78>= k[1]) and (k[1] > -0.83)  then
                        begin
                          b:=6; count(k,b);
                        end;
if  (-0.83>= k[1]) and (k[1] > -0.87)  then
                        begin
                          b:=7; count(k,b);
                        end;
if  (-0.87>= k[1]) and (k[1] > -0.92)  then
                        begin
                          b:=8; count(k,b);
                        end;
if  (-0.92>= k[1]) and (k[1] > -0.95)  then
                        begin
                          b:=9; count(k,b);
                        end;
if  (-0.95>= k[1]) and (k[1] > -0.98)  then
                        begin
                          b:=10; count(k,b);
                        end;
if  (-0.98>= k[1]) and (k[1] > -1 )  then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    begin
                      b:=11; count(k,b);
                    end;
end;

procedure make_win(var Sn0:real; var Sn:xxxxx;
                   var fi_lpc:fi_by);
var
j:integer;
ch:byte;
Sn1:real;

begin
  for j:=0 to 329 do
     begin
       read(fi_lpc,ch);
       Sn[j]:=ch;
     end;
  Sn1:=Sn[329];
  lowpass(Sn0,Sn);
  make_ri(Sn);
  make_k(r);
  count1(k);
  Sn0:=Sn1;
end;

procedure clear(var input:inp);
var
  i,j:integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  for i:=1 to 11 do
     begin
       for i:=1 to 8 do
          begin
            input[i,j]:=0;
          end;
     end;
end;

procedure output(wm:wmm;wo:woo;x:inback);
var
  i,j:integer;
  m:extended;
begin
   for i:=1 to 9 do
      im[i]:=0;
      for j:=1 to 88 do
        begin
          m:=x[j]*wm[j,i];
          im[i]:=im[i]+m;
        end;
      fim[i]:=1/(1+(exp(-im[i])));
   end;
   for i:=1 to 10 do
      begin
        io[i]:=0;
        for j:=1 to 9 do
          begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้นทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            m:=fim[j]*wo[j,i];
            io[i]:=io[i]+m;
          end;
        fio[i]:=1/(1+(exp(-io[i])));
      end;
end; {end procedure output of backpropagation}

begin {main program}
  writeln('Do you want to train or use ');
  writeln('1] train ');
  writeln('2] use   ');
  write('your choice ');
  readln(choice);
  if choice = 1 then
     begin
       write('how many file.wav you want to make LPC ');
       readln(how);
     end;
  if choice = 2 then
     begin
       how:=1;
     end;
  write('what is the name of complete file ');
  readln(filew);
  assign(fi_in,filew);
  rewrite(fi_in);
  for number:=1 to how do
     begin
       write('what file.wav you want to make LPC ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
readln(filewav);
assign(fi_lpc,filewav);
size:=0;
reset(fi_lpc);
size:=filesize(fi_lpc);
close(fi_lpc);
size:=size-44;
clear(input);
Sn:=0;
window:=(((size-330) div 300)+1);
poitt:=74;
for i:=1 to windows do
  begin
     reset(fi_lpc);
     poitt:=poitt+((i-1)*300);
     for time:=1 to poitt do
       read(fi_lpc,ch);
     poitt:=44;
     make_win(Sn0,Sn,fi_lpc);
     close(fi_lpc);
  end;
for i:=1 to 1f do
  begin
     for j:=1 to 8 do
       begin
          write(fi_in,input[i,j]);
       end;
  end;
end; {end how}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        close(fi_in);

{begin backpropagation}
for i:=1 to 9 do
  begin
     for j:=1 to 10 do
     wco[i,j]:=0;
  end;
for i:=1 to 88 do
  begin
     for j:=1 to 9 do
        wcm[i,j]:=0;
  end;
write('Type filename of initial weight ');
readln(filename);
assign(file_var,filename);
reset(file_var);
for i:=1 to 88 do
  begin
     for j:=1 to 9 do
       read(file_var,wm[i,j]);
  end;
for i:=1 to 9 do
  begin
     for j:=1 to 10 do
       read(file_var,wo[i,j]);
  end;
close(file_var);
if choice=2 then
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin
      reset(fi_in);
      for i:=1 to 88 do
        begin
          read(fi_in,x[i]);
        end;
      output(wm,wo,x);
      for i:=1 to 10 do
        begin
          writeln('output of neural[',i,'] is ',fio[i]);
        end;
      maxerr:=fio[1];
      for i:=1 to 9 do
        begin
          if maxerr > fio[i+1] then maxerr:=maxerr
                               else maxerr:=fio[i+1];
        end;
      for i:=1 to 10 do
        begin
          if maxerr=fio[i] then writeln('This sound is ',(i-1));
        end;
      close(fi_in);
      readln;
    end {end of if use}
  else
  begin {else begin}
      write('Type filename of desired output ');
      readln(filename);
      assign(file_var,filename);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write('Type value of learning constant [0..1] = ');
readln(beta);
write('Type value of momentum [0..1] = ');
readln(u);
turn:=0;
repeat
    for i:=1 to how do
      begin
         smaxerr[i]:=0;
      end; {reset error}
    totalerror:=0;
    reset(fi_in);
    reset(file_var);
    for a:=1 to how do
      begin
        for i:=1 to 88 do
          begin
             read(fi_in,x[i]);
          end;
        for i:=1 to 10 do
          begin
             read(file_var,yd[i]);
          end;
        output(wm,wo,x);
        for i:=1 to 10 do
          ya[i]:=fio[i];
        for i:=1 to 10 do
          eo[i]:=yd[i]-ya[i];
        for i:=1 to 10 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  h[i]:=abs(eo[i]);
maxerr:=h[1];
for i:=1 to 9 do
  begin
     if maxerr > h[i+1] then maxerr:=maxerr;
                        else maxerr:=h[i+1];
  end;
error:=0.1;
smaxerr[a]:=maxerr;
for i:=1 to 9 do
  begin
     mwe[i]:=0;
     for j:=1 to 10 do
      begin
         n:=wo[i,j]*eo[j];
         mwe[i]:=mwe[i]+n;
      end;
     em[i]:=((fim[i]*(1-(fim[i])))*(mwe[i]);
  end;
for i:=1 to 10 do
  begin
     for j:=1 to 9 do
       begin
          wco[i,j]:=((beta*eo[i]*fim[j])+(u*wco[j,i]));
          wo[j,i]:=(wo[j,i]+wco[j,i]);
       end;
  end;
for i:=1 to 88 do
  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            fx[i]:=1/(1+(exp(-x[i]/2)));
          end;
        for i:=1 to 9 do
          begin
            for j:=1 to 88 do
              begin
                wcm[j,i]:=((beta*em[i]*fx[j])+(u*wcm[j,i]));
                wm[j,i]:=(wm[j,i]+wcm[j,i]);
              end;
          end;
      end;
    close(fi_in);
    close(file_var);
    maxerr:=smaxerr[1];
    for i:=1 to (how-1) do
      begin
        if maxerr > smaxerr[i+1] then maxerr:=maxerr
                                 else maxerr:=smaxerr[i+1];
      end;
    if maxerr < error then totalerror:=1;
                      else totalerror:=0;
    writeln('max error  = ',maxerr);
    writeln(;totalerror = ',totalerror);
    turn:=turn+1;
    writeln('turn = ',turn);
until totalerror=1;
write(Type the name of file save weight ');
readln(filename);
assign(file_var,filename);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    rewrite(file_var);
    for i:=1 to 88 do
        begin
          for j:=1 to 9 do
             write(file_var,wm[i,j]);
        end;
    for i:=1 to 9 do
        begin
          for j:=1 to 10 do
             write(file_var,wo[i,j]);
        end;
    close(file_var);
  end;{end else}
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมข้างต้น เป็นโปรแกรมที่จะนำค่า $k_1$ และ $k_2$ ของแต่ละ window มาplot ลงในช่วงต่าง ๆ ที่แบ่งไว้ คือ ค่า $k_1$ 11 ช่วง และ $k_2$ 8 ช่วง ทำให้เกิดตารางทั้งหมด 64 ช่วง จากนั้นก็จะนับจำนวนของค่า k ที่อยู่ในแต่ละช่อง แล้วนำเลขจำนวนนี้ไปป้อนให้กับ Neural Network

โปรแกรมลำดับต่อมา เป็นโปรแกรมที่ใช้วิธีป้อนอินพุทให้กับ Neural Network อีกแบบคือ นำค่า $k_1$ และ $k_2$ ไปป้อนโดยตรง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program Linear_Predictive_coding;

uses crt;

var filename:string[15];
   trainfile,soundfile:file of byte;
   ampli,ampli0:byte;
   w,Lampli:real;
   j,m,n,o,p,I,L,posi,fsize,rest,lengh:integer;
   ofile:file of real;
   R:array[0..6]of real;
   osig:array[0..499]of real;
   E:array[0..6]of real;
   K:array[1..7]of real;
   a:array[0..6,0..6]of real;
   A3,K2:array[1..26]of real;

begin

     clrscr;
     write(' what file do you want to train ? ');
     readln(filename);
     assign(soundfile,filename);
     write(' Enter filename to save in ');
     readln(filename);
     assign(ofile,filename);
     reset(soundfile);
     rewrite(ofile);
     fsize:=filesize(soundfile);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
seek(soundfile,44);
L:=((fsize-44)*2)div 47 ; {size=L+25[0.9L]}
I:=(L-(L div 10))*25+L;
rest:=0;
lengh:=L-1;
if (fsize-44) < I then
   rest:= I-(fsize-44);
for I:=1 to 26 do begin
    if I=1 then begin
       ampli0:=0;
       for n:=0 to (L-1) do begin
          read(soundfile,ampli);
          Lampli:=ampli-0.9375*ampli0;
          ampli0:=ampli;
          w:=0.54-(0.46*cos(6.283*n/L));
          osig[n]:=Lampli*w;
       end; {for n}
    end  {if=1}
    else begin
       posi:=(I-1)*(L-(L div 10))+44;
       seek(soundfile,posi);
       if I=26 then
          if rest > 0 then
               lengh:=(L-1)-rest;
          for n:=0 to lengh do begin
               read(soundfile,ampli);
               Lampli:=ampli-0.9375*ampli0;
               ampli0:=ampli;
               w:=0.54-(0.46*cos(6.283*n/L));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            osig[n]:=Lampli*w;
        end;{for n}
    end;{else}
    for o:=0 to 3 do begin
        R[o]:=0;
        for n:=0 to (L-1-o) do
            R[o]:=R[o]+osig[n]*osig[n+o];
    end;{for o}
E[0]:=R[0];
for m:=1 to 3 do begin
    K[m]:=0;
    p:=m;
    o:=0;
    a[m-1,0]:=1;
    repeat
        K[m]:=K[m]-(R[p]*a[m-1,o])/E[m-1];
        p:=p-1;
        o:=o+1;
    until p=0;
    if m<4 then
        a[m,m]:=K[m];
    j:=1;
    while j<>m do begin
        a[m,j]:=a[m-1,j]+K[m]*a[m-1,m-j];
        j:=j+1;
    end; {while}
    if m<3 then
        E[m]:=(1-sqr(K[m])*E[m-1]);
end;{for}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if K[2]>1 then K[2]:=1;
            A3[I]:=a[3,1]/10;
            K2[I]:=K[2];
            write(ofile,A3[I],K[2]);
            writeln(K[1],'.   ',A3[I],'.   ',K[2]);{readln;}
        end;
        for I:= 1 to 26 do
            write(ofile,K2[I]);
        for I:= 1 to 26 do
            write(ofile,A3[I]);}
        close(soundfile);
        close(ofile);
        write('    work complete !');
        readln;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program backpropagation;

uses crt;

const wsize = 52;
      pattrn = 20;
      nmid=9;

var ip,Emax,Etotal,Iout,Inerror,delW,Finput:real;
    Bet,MOM:real;
    n,A,B,C,mid:byte;
    strt,turn,out:integer;
    wghfile:string[12];
    lpc,wgh:file of real;
    input:array[1..pattrn,1..wsize]of real;
    Wmid,delWprm:array[1..nmid,1..wsize]of real;
    Wout,delWpro:array[1..10,1..nmid]of real;
    Imid,FImid,finalE:array[1..nmid]of real;
    FIout,Error:array[1..10]of real;
    Elast:array[1..pattrn]of real;
    deout:array[1..pattrn,1..10]of real;
    lpcfile:array[1..10]of string[12];
    desire:array[1..10]of byte;

begin
     clrscr;
     writeln;
     write('     how many file do you want to train ? ');
     readln(n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for A := 1 to n do
    for B:= 1 to 10 do
        deout[A,B]:= 0;
for A:= 1 to n do begin
    write('     enter file name : ');
    readln(lpcfile[A]);
    write('     enter desire output : ');
    readln(desire[A]);
    case desire[A] of
       1:deout[A,1]:=1;
       2:deout[A,2]:=1;
       3:deout[A,3]:=1;
       4:deout[A,4]:=1;
       5:deout[A,5]:=1;
       6:deout[A,6]:=1;
       7:deout[A,7]:=1;
       8:deout[A,8]:=1;
       9:deout[A,9]:=1;
       0:deout[A,10]:=1;
    end;
end;

for A := 1 to nmid do
    for B := 1 to wsize do begin
       Wmid[A,B]:=random/100;
       delWprm[A,B]:=0;
    end;
for A := 1 to 10 do
    for B := 1 to nmid do begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Wout[A,B]:=random/100;
        delWpro[A,B]:=0;
    end;
turn:=0;
Emax:=0.1;
Bet:=0.7;
MOM:=0.65;
repeat
      for A := 1 to n do begin
          Elast[A] := 0;
          if turn = 0 then begin
              assign(lpc,lpcfile[A]);
              reset(lpc);
              for C:= 1 to wsize do begin
                  read(lpc,ip);
                  input[A,C]:=1/(1+(exp(-ip)));
              end;
              close(lpc);
          end;{if}

          for mid := 1 to nmid do begin
             Imid[mid]:=0;
             for B := 1 to wsize do
               Imid[mid] := Imid[mid]+(Wmid[mid,B]*input[A,B]);
             FImid[mid] := 1/(1+(exp(-Imid[mid])));
          end; {for mid}

          for out := 1 to 10 do begin
              Iout:=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for B := 1 to nmid do
        Iout := Iout+(Wout[out,B]*FImid[B]);
    FIout[out] := 1/(1+(exp(-Iout)));
end; {for out}

for out := 1 to 10 do begin
    Error[out] := deout[A,out]-FIout[out];
    if Elast[A] < abs(Error[out]) then
        Elast[A] := abs(Error[out]);
end;

for mid := 1 to nmid do begin
    Inerror:=0;
    for B := 1 to 10 do
        Inerror := Inerror+(Wout[B,mid]*Error[B]);
    finalE[mid] := Inerror*FImid[mid]*(1-FImid[mid]);
end; {for mid}

for out := 1 to 10 do
    for B := 1 to nmid do begin
        delW:=(Bet*Error[out]*FImid[B])+
              (MOM*delWpro[out,B]);
        Wout[out,B]:=Wout[out,B]+delW;
        delWpro[out,B]:=delW;
    end;

for mid := 1 to nmid do
    for B := 1 to wsize do begin
        delW:=(Bet*finalE[mid]*input[A,B])+
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    (MOM*delWprm[mid,B]);
                Wmid[mid,B]:=Wmid[mid,B]+delW;
                delWprm[mid,B]:=delW;
            end;

    end; {for A}

    turn:=turn+1;

    Etotal:=Elast[1];
    for B:= 1 to n do
        if Elast[B] > Etotal then
            Etotal := Elast[B];

    if (Etotal < 0.99) then begin
            Bet:=0.7;
            MOM:=0.65;
            end;

    if (0.55 < Etotal) and (Etotal <= 0.75) then begin
            Bet:=0.6;
            MOM:=0.8;
            end;

    if (0.48 < Etotal) and (Etotal <= 0.55) then begin
            Bet:=0.4{0.7};
            MOM:=0.9{0.6};
            end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (turn mod 10) = 1 then begin
           write('     ',turn,' turn passed');
           writeln('   Elast =  ',Etotal);
           end;

    until Etotal < Emax ;

    write('     work complete !  ',turn,' turn');readln;
    write('     Enter file name to save weight : ');
    readln(wghfile);
    assign(wgh,wghfile);
    rewrite(wgh);

    for mid:= 1 to nmid do
        for B:= 1 to wsize do
            write(wgh,Wmid[mid,B]);

    for out:= 1 to 10 do
        for B:= 1 to nmid do
            write(wgh,Wout[out,B]);

    close(wgh);

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if output = 10 then
        output:= 0;
    write('     output is ',output);
    write('     work complete !');readln;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program Usebackpropagation;

uses crt;

const wsize = 52;

var Imid,Iout,outmax,inp:real;
    A,B,mid:byte;
    out,output:integer;
    lpcfile,wghfile:string[15];
    lpc,wgh:file of real;
    Finput:array[1..wsize]of real;
    Wmid:array[1..9,1..wsize]of real;
    Wout:array[1..10,1..9]of real;
    FImid:array[1..9]of real;
    FIout:array[1..10]of real;

begin
     clrscr;
     write('     enter file name : ');
     readln(lpcfile);
     assign(lpc,lpcfile);
     write('     enter weighfile : ');
     readln(wghfile);
     assign(wgh,wghfile);
     reset(wgh);
     for A:= 1 to 9 do
          for B:= 1 to wsize do
               read(wgh,Wmid[A,B]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for A:= 1 to 10 do
     for B:= 1 to 9 do
        read(wgh,Wout[A,B]);

reset(lpc);
for B:= 1 to wsize do begin
     read(lpc,inp);
     Finput[B]:= 1/(1+(exp(-inp)));
end;

for mid := 1 to 9 do begin
    Imid:=0;
    for B := 1 to wsize do
       Imid := Imid+(Wmid[mid,B]*Finput[B]);
    FImid[mid] := 1/(1+exp(-Imid));
end; {for mid}

outmax:=0;
for out := 1 to 10 do begin
     Iout:=0;
     for B := 1 to 9 do
          Iout := Iout+(Wout[out,B]*FImid[B]);
     FIout[out] := 1/(1+exp(-Iout));
     if outmax<FIout[out] then begin
          outmax:=FIout[out];
          output:=out;
     end; {if}
end; {for out}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if output = 10 then
             output:= 0;
        write('     output is ',output);
        write('     work complete !');readln;
    end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะเป็นโปรแกรมอีกชุดหนึ่งซึ่งจะมีการจัดการกับค่า $K_1$ และ $K_2$ ด้วยการนำมาจัดแบ่งออกเป็นช่วงๆเช่นแบบแรก แต่ที่ต่างกันคือจะจัดแบ่ง $K_1$ เป็น 7ช่วง $K_2$ 10ช่วง แต่ละช่วงจะมีค่าที่จะใช้เป็นตัวแทนของช่วง นำค่าตัวแทนของ $K_1$ และ $K_2$ มาคูณกันแล้วใช้ค่านี้เป็นตัวแทนของ window นำไปป้อนให้กับ neural network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program use;
uses crt;
var filename:string[15];
    trainfile,soundfile:file of byte;
    ampli,ampli0,yes:byte;
    w,Lampli:extended;
    j,m,n,o,p,I,L,posi,fsize,rest,lengh:integer;
    ofile:file of extended;
    Ps1,Ps2,AP:extended;
    R:array[0..6]of extended;
    osig:array[0..499]of extended;
    E:array[0..6]of extended;
    K:array[1..4]of extended;
    a:array[0..6,0..6]of extended;
begin
     clrscr;
     yes:=1;
     while yes <> 0 do begin
        write(' what file do you want to train ? ');
        readln(filename);
        assign(soundfile,filename);
        write(' Enter filename to save in ');
        readln(filename);
        assign(ofile,filename);
        reset(soundfile);
        rewrite(ofile);
        fsize:=filesize(soundfile);
        seek(soundfile,44);
        L:=((fsize-44)*2)div 47 ; {size=L+25[0.9L]}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
I:=(L-(L div 10))*25+L;
rest:=0;
lengh:=L-1;
if (fsize-44) < I then
   rest:= I-(fsize-44);
for I:=1 to 26 do begin
if I=1 then begin
    ampli0:=0;
    for n:=0 to (L-1) do begin
        read(soundfile,ampli);
        Lampli:=ampli-0.9375*ampli0;
        ampli0:=ampli;
        w:=0.54-(0.46*cos(6.283*n/L));
        osig[n]:=Lampli*w;
   end; {for n}
end  {if=1}
else begin
     posi:=(I-1)*(L-(L div 10))+44;
     seek(soundfile,posi);
     if I=26 then
        if rest > 0 then
           lengh:=(L-1)-rest;
     for n:=0 to lengh do begin
         read(soundfile,ampli);
         Lampli:=ampli-0.9375*ampli0;
         ampli0:=ampli;
         w:=0.54-(0.46*cos(6.283*n/L));
         osig[n]:=Lampli*w;
         end;{for n}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;{else}
for o:=0 to 3 do begin
    R[o]:=0;
    for n:=0 to (L-1-o) do
        R[o]:=R[o]+osig[n]*osig[n+o];
end;{for o}
E[0]:=R[0];
for m:=1 to 3 do begin
    K[m]:=0;
    p:=m;
    o:=0;
    a[m-1,0]:=1;
    repeat
        K[m]:=K[m]-(R[p]*a[m-1,o])/E[m-1];
        p:=p-1;
        o:=o+1;
        until p=0;
        if m<4 then
            a[m,m]:=K[m];
        j:=1;
        while j<>m do begin
            a[m,j]:=a[m-1,j]+K[m]*a[m-1,m-j];
            j:=j+1;
            end; {while}
        if m<3 then
            E[m]:=(1-sqr(K[m])*E[m-1]);
    end;{for}
    if K[2]>1 then K[2]:=1;
    {write('          ',K[1],'          ',K[2]);readln;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if K[1] < -0.97 then Ps1:=0.1;
            if (K[1] >= -0.97) and (K[1] < -0.93) then Ps1:=0.2;
            if (K[1] >= -0.93) and (K[1] < -0.87) then Ps1:=0.3;
            if (K[1] >= -0.87) and (K[1] < -0.79) then Ps1:=0.4;
            if (K[1] >= -0.79) and (K[1] < -0.69) then Ps1:=0.5;
            if (K[1] >= -0.69) and (K[1] < -0.57) then Ps1:=0.6;
            if K[1] >= -0.57 then Ps1:=0.7;
            if (K[2] <= 1) and (K[2] > 0.1) then Ps2:=0.1;
            if (K[2] <= 0.1) and (K[2] > 0.01) then Ps2:=0.2;
            if (K[2] <= 0.01) and (K[2] > 0.001) then Ps2:=0.3;
            if (K[2] <= 0.001) and (K[2] > 0.0001) then Ps2:=0.4;
            {if (K[2] <= 0.01) and (K[2] > 0.005) then Ps2:=5;
            if (K[2] <= 0.005) and (K[2] > 0.001) then Ps2:=6;}
            if (K[2] <= 0.0001)and (K[2] > 0) then Ps2:=0.7;
            if (K[2] >= -1) and (K[2] < -0.1) then Ps2:=-0.1;
            if (K[2] >= -0.1) and (K[2] < -0.01) then Ps2:=-0.2;
            if (K[2] >= -0.01) and (K[2] < -0.001) then Ps2:=-0.3;
            if (K[2] >= -0.001) and (K[2] < -0.0001) then Ps2:=-0.4;
            {if (K[2] >= -0.01) and (K[2] < -0.005) then Ps2:=-5;
            if (K[2] >= -0.005) and (K[2] < -0.001) then Ps2:=-6;}
            if (K[2] >= -0.0001) and (K[2] <=0) then Ps2:=-0.7;
            AP:=Ps1*Ps2*10;
            if Abs(Ps1)>abs(Ps2) then AP:=AP*10;
            write(ofile,Ap{Ps1,Ps2});
            writeln(Ap,'        ',Ps1,'        ',Ps2);
end;
close(soundfile);
close(ofile);
writeln('      work complete !');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    readkey;
    writeln('      Do you want to train another ? ');
    write('      press 1 for yes and  0 for no  : ');
    readln(Yes);
    end;
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program backpropagation;
uses crt,dos;

const wsize = 26;
      pattrn = 100;
      nmid=9;
var Emax,Etotal,Iout,Inerror,delW,Finput:extended;
    Bet,MOM,addmom,Epr:extended;
    n,A,B,C,D,mid,ChaE:integer;
    ip:extended;
    sh,sm,ss,sfr,eh,em,es,efr,uh,us,um,ufr,stime,etime,utime:word;
    strt,turn,out:integer;
    wghfile:string[12];
    wgh:file of extended;
    lpc,lpc2:file of extended;
    input:array[1..pattrn,1..wsize]of extended;
    Wmid,delWprm:array[1..nmid,1..wsize]of extended;
    Wout,delWpro:array[1..10,1..nmid]of extended;
    Imid,FImid,finalE:array[1..nmid]of extended;
    FIout,Error:array[1..10]of extended;
    Elast:array[1..pattrn]of extended;
    deout:array[1..pattrn,1..10]of extended;
    desire:byte;
    lpcfile,lpcfile2:string[15];
begin
      clrscr;
      writeln;
      write('      how many file do you want to train ?  ');
      readln(n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if n < 21 then addmom:=0.2;
if n < 41 then addmom:=0.1;
if n < 61 then addmom:=0.05;
if n > 60 then addmom:=0;
for A := 1 to n do
    for B:= 1 to 10 do
        deout[A,B]:= 0;
for A:= 1 to n do begin
    write('     enter file name : ');
    readln(lpcfile);
    assign(lpc,lpcfile);
    reset(lpc);
    for C:= 1 to wsize do begin
      read(lpc,ip);
      input[A,C]:=1/(1+(exp(-ip)));
      end;
    close(lpc);
end;
for A:=1 to n do begin
    desire:=(A-1) mod 10;
    case desire of
       1:deout[A,1]:=1;
       2:deout[A,2]:=1;
       3:deout[A,3]:=1;
       4:deout[A,4]:=1;
       5:deout[A,5]:=1;
       6:deout[A,6]:=1;
       7:deout[A,7]:=1;
       8:deout[A,8]:=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            9:deout[A,9]:=1;
            0:deout[A,10]:=1;
        end;
    end;

    for A := 1 to nmid do
        for B := 1 to wsize do begin
            Wmid[A,B]:=random/100;
            delWprm[A,B]:=0;
        end;
    for A := 1 to 10 do
        for B := 1 to nmid do begin
            Wout[A,B]:=random/100;
            delWpro[A,B]:=0;
        end;
    turn:=0;
    Emax:=0.1;
    Bet:=0.7;
    MOM:=0.65;
    ChaE:=0;
    Epr:=0;
    gettime(sh.sm.ss,sfr);
    repeat
      for A := 1 to n do begin
        Elast[A] := 0;
        {if turn = 0 then begin
           assign(lpc,lpcfile[A]);
           reset(lpc);
           for C:= 1 to wsize do begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            read(lpc,ip);
            input[A,C]:=1/(1+(exp(-ip)));
        end;
        close(lpc);
    end;{if}

    for mid := 1 to nmid do begin
        Imid[mid]:=0;
        for B := 1 to wsize do
            Imid[mid] := Imid[mid]+(Wmid[mid,B]*input[A,B]);
        FImid[mid] := 1/(1+(exp(-Imid[mid])));
    end; {for mid}

    for out := 1 to 10 do begin
        Iout:=0;
        for B := 1 to nmid do
            Iout := Iout+(Wout[out,B]*FImid[B]);
        FIout[out] := 1/(1+(exp(-Iout)));
    end; {for out}

    for out := 1 to 10 do begin
        Error[out] := deout[A,out]-FIout[out];
        if Elast[A] < abs(Error[out]) then
            Elast[A] := abs(Error[out]);
    end;

    for mid := 1 to nmid do begin
        Inerror:=0;
        for B := 1 to 10 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Inerror := Inerror+(Wout[B,mid]*Error[B]);
            finalE[mid] := Inerror*FImid[mid]*(1-FImid[mid]);
        end; {for mid}

        for out := 1 to 10 do
            for B := 1 to nmid do begin
              {if abs(Error[out]) > 0.0001 then begin}
                delW:=(Bet*Error[out]*FImid[B])+
                        (MOM*delWpro[out,B]);
                Wout[out,B]:=Wout[out,B]+delW;
                delWpro[out,B]:=delW;
              {end;}
            end;

        for mid := 1 to nmid do
            for B := 1 to wsize do begin
                delW:=(Bet*finalE[mid]*input[A,B])+
                      (MOM*delWprm[mid,B]);
                Wmid[mid,B]:=Wmid[mid,B]+delW;
                delWprm[mid,B]:=delW;
            end;

end; {for A}

turn:=turn+1;

Etotal:=Elast[1];
for B:= 1 to n do
    if Elast[B] > Etotal then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Etotal := Elast[B];

 if Etotal > 0.99999 then begin
        MOM:=0;
        Bet:=0.8;
        end;

 if (Etotal < 0.99) then begin
        Bet:=0.5;
        MOM:=0.3;
        end;

if (0.55 < Etotal) and (Etotal <= 0.75) then begin
        Bet:=0.45;
        MOM:=0.25;
        end;

if (0.48 < Etotal) and (Etotal <= 0.55) then begin
        Bet:=0.4{0.7};
        MOM:=0.2{0.6};
        end;

if (0.4 < Etotal) and (Etotal <= 0.48) then begin
       Bet:=0.35;
       MOM:=0.2;
       end;

if (Epr-Etotal) < 0.002 then begin
      ChaE:= ChaE+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            end
            else ChaE:=0;
         if ChaE >= 100 then begin
            MOM:=0.9;
            Bet:=0.3;
            end;
         Epr:=Etotal;
         MOM:=MOM+addmom;

         if (turn mod 2) = 1 then begin
            write('      '.turn,' turn passed');
            writeln('   Elast =  ',Etotal);
         end;

until Etotal < Emax ;
gettime(eh,em,es,efr);
stime:=(sh*3600)+(sm*60)+ss;
etime:=(eh*3600)+(em*60)+es;
utime:=etime-stime;
uh:=utime div 3600;
um:=(utime mod 3600) div 60;
us:=(utime mod 60);
writeln('     work complete !  ',turn,' turn');
write('     spending time : ',uh,':',um,':',us);
readln;
write('     Enter file name to save weight : ');
readln(wghfile);
assign(wgh,wghfile);
rewrite(wgh);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for mid:= 1 to nmid do
        for B:= 1 to wsize do
            write(wgh,Wmid[mid,B]);

    for out:= 1 to 10 do
        for B:= 1 to nmid do
            write(wgh,Wout[out,B]);

    close(wgh);
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program usebackprop;
uses crt;
const wsize = 26;
var Imid,Iout,outmax:extended;
    A,B,mid,yes:byte;
    inp:extended;
    out,out2,output:integer;
    lpcfile,wghfile:string[15];
    wgh:file of extended;
    lpc:file of extended;
    Finput:array[1..wsize]of extended;
    Wmid:array[1..9,1..wsize]of extended;
    Wout:array[1..10,1..9]of extended;
    FImid:array[1..9]of extended;
    FIout:array[1..10]of extended;
begin
     clrscr;
     yes:=1;
     while yes <> 0 do begin
     write('     enter file name : ');
     readln(lpcfile);
     assign(lpc,lpcfile);
     write('     enter weighfile : ');
     readln(wghfile);
     assign(wgh,wghfile);
     reset(wgh);
     for A:= 1 to 9 do
         for B:= 1 to wsize do
             read(wgh,Wmid[A,B]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for A:= 1 to 10 do
    for B:= 1 to 9 do
        read(wgh,Wout[A,B]);
close(wgh);
reset(lpc);
for B:= 1 to wsize do begin
    read(lpc,inp);
    Finput[B]:=1/(1+(exp(-inp)));
    end;
close(lpc);
for mid := 1 to 9 do begin
    Imid:=0;
    for B := 1 to wsize do
        Imid := Imid+(Wmid[mid,B]*Finput[B]);
    FImid[mid] := 1/(1+exp(-Imid));
end; {for mid}
outmax:=0;
out2:=0;
for out := 1 to 10 do begin
    Iout:=0;
    for B := 1 to 9 do
        Iout := Iout+(Wout[out,B]*FImid[B]);
    FIout[out] := 1/(1+exp(-Iout));
    if outmax<FIout[out] then begin
       outmax:=FIout[out];
       output:=out;
    end; {if}
end; {for out}
for out:=1 to 10 do begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if ((outmax-FIout[out])<0.15) and (outmax<>FIout[out]) then
       begin
            out2:=1;
       end;
    end;
    if output = 10 then
       output:= 0;
    write('     output is ',output);
    if out2=1 then write('    but not sure  ');
    writeln('     work complete !');
    writeln('     Do you want to use another ? ');
    write('     1 for yes   0 for no  :');
    readln(yes);
    end;{while}
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6
# ผลการทดลองและสรุป

## 6.1 การทดลอง

ก่อนที่เราจะใช้งานระบบ เราจะต้องtrainให้ระบบจำเสียงให้ได้ โดยจะเลือกเสียงของคำที่เป็นคำเดี่ยว โดยจะใช้เสียง 0 ถึง 9 จำนวน 6 ชุดมา train ให้กับระบบแล้วใช้อีก 7ชุด เป็น test pattern ซึ่งทั้งหมดนี้เป็นเสียงของคนคนเดียว

## 6.2 ผลการทดลอง

การทดสอบโปรแกรมแบบแรกได้ผลดังนี้

ตารางที่ 6.1 ผลการทดลองของโปรแกรมแบบแรก

| เสียง | จำนวนที่ทดสอบ | ถูกต้อง | คิดเป็น % |
|---|---|---|---|
| 0 | 7 | 5 | 71.42 |
| 1 | 7 | 5 | 71.42 |
| 2 | 7 | 5 | 71.42 |
| 3 | 7 | 4 | 57.14 |
| 4 | 7 | 7 | 100.00 |
| 5 | 7 | 6 | 85.71 |
| 6 | 7 | 7 | 100.00 |
| 7 | 7 | 4 | 57.14 |
| 8 | 7 | 6 | 85.71 |
| 9 | 7 | 4 | 57.14 |
| รวม | 70 | 53 | 75.71 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแบบที่สองไม่สามารถจำ training pattern ได้ ส่วนโปรแกรมแบบที่สามนั้นได้ผลการทดลองดังนี้

ตารางที่ 6.2 ผลการทดลองของโปรแกรมแบบที่สาม

| เสียง | จำนวนที่ทดสอบ | ถูกต้อง | คิดเป็น % |
|---|---|---|---|
| 0 | 7 | 5 | 71.42 |
| 1 | 7 | 3 | 42.86 |
| 2 | 7 | 0 | 0.00 |
| 3 | 7 | 6 | 85.71 |
| 4 | 7 | 6 | 85.71 |
| 5 | 7 | 2 | 28.57 |
| 6 | 7 | 6 | 85.71 |
| 7 | 7 | 2 | 28.57 |
| 8 | 7 | 4 | 57.14 |
| 9 | 7 | 4 | 57.14 |
| รวม | 70 | 38 | 54.29 |

## 6.3 สรุป

จากผลการทดลองจะเห็นได้ว่า การจัดรูปแบบ Input ที่จะป้อนเข้าสู่ neural network มีผลต่อประสิทธิภาพมาก การนำค่า K มาแบ่งเป็นช่วงช่วยให้ระบบสามารถจำ pattern ได้ดีกว่าการนำค่า K ป้อนเข้าไปโดยตรง และการแบ่งช่องออกเป็น 88 ช่อง($K_1$ 11ช่วง,$K_2$ 8ช่วง) แล้วนับจำนวนของค่า K ในแต่ละช่องก็จะให้ประสิทธิภาพสูงกว่าการนำค่าตัวแทนของแต่ช่วงป้อนเป็น Inputให้กับ neural network

สำหรับแนวทางในการพัฒนาต่อไปนั้นสามารถทำได้หลายทาง เช่น การจัดรูปแบบของ Input ใหม่ การเลือกระบบ neural network หรือการใช้ parameter อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก.

## แสดงตัวอย่างค่า K ของเสียง 0 ถึง 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียง 0

| $K_1$ | $K_2$ |
|---|---|
| -0.993 | 0.0071 |
| -0.963 | 0.0599 |
| -0.893 | 0.2094 |
| -0.981 | 0.0437 |
| -0.957 | 0.0068 |
| -0.949 | -0.0012 |
| -0.958 | -0.0008 |
| -0.960 | 0.0077 |
| -0.952 | 0.0156 |
| -0.950 | 0.0234 |

เสียง 1

| $K_1$ | $K_2$ |
|---|---|
| -0.994 | 0.0063 |
| -0.994 | 0.0066 |
| -0.993 | 0.0052 |
| -0.990 | 0.0065 |
| -0.989 | 0.0034 |
| -0.921 | 0.0443 |
| -0.919 | -0.0039 |
| -0.920 | 0.0018 |
| -0.915 | 0.0175 |
| -0.910 | 0.0236 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียง 2

| $K_1$ | $K_2$ |
|---|---|
| -0.945 | -0.0575 |
| -0.878 | -0.1398 |
| -0.863 | -0.1715 |
| -0.894 | -0.1414 |
| -0.897 | -0.1445 |
| -0.902 | -0.1301 |
| -0.899 | -0.1244 |
| -0.912 | -0.1059 |
| -0.908 | -0.1203 |
| -0.906 | -0.1274 |

เสียง 3

| $K_1$ | $K_2$ |
|---|---|
| -0.989 | 0.0031 |
| -0.896 | -0.1178 |
| -0.839 | -0.2668 |
| -0.814 | -0.3678 |
| -0.808 | -0.3766 |
| -0.819 | -0.3566 |
| -0.842 | -0.2732 |
| -0.854 | -0.2322 |
| -0.868 | -0.2033 |
| -0.864 | -0.2258 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียง 4

| $K_1$ | $K_2$ |
|---|---|
| -0.992 | 0.0103 |
| -0.988 | 0.0193 |
| -0.987 | 0.0199 |
| -0.978 | 0.0355 |
| -0.987 | 0.0184 |
| -0.972 | 0.0033 |
| -0.890 | -0.0265 |
| -0.842 | -0.0637 |
| -0.839 | 0.0369 |
| -0.815 | 0.0435 |

เสียง 5

| $K_1$ | $K_2$ |
|---|---|
| -0.952 | -0.0412 |
| -0.926 | -0.0624 |
| -0.875 | -0.1622 |
| -0.821 | -0.2930 |
| -0.809 | -0.3502 |
| -0.810 | -0.3783 |
| -0.801 | -0.3714 |
| -0.801 | -0.3778 |
| -0.821 | -0.3915 |
| -0.820 | -0.4001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เสียง 6**

| $K_1$ | $K_2$ |
|---|---|
| -0.993 | 0.0060 |
| -0.993 | 0.0066 |
| -0.986 | -0.0098 |
| -0.960 | -0.0444 |
| -0.965 | -0.0169 |
| -0.935 | -0.0665 |
| -0.965 | -0.0457 |
| -0.922 | -0.0228 |
| -0.950 | -0.0485 |
| -0.927 | -0.0425 |

**เสียง 7**

| $K_1$ | $K_2$ |
|---|---|
| -0.988 | 0.0134 |
| -0.980 | 0.0237 |
| -0.985 | 0.0033 |
| -0.759 | -0.0271 |
| -0.844 | -0.0978 |
| -0.721 | -0.0866 |
| -0.800 | -0.0398 |
| -0.767 | -0.0891 |
| -0.791 | -0.0949 |
| -0.819 | -0.0971 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียง 8

| $K_1$ | $K_2$ |
|---|---|
| -0.842 | -0.3264 |
| -0.833 | -0.3278 |
| -0.818 | -0.3423 |
| -0.802 | -0.3328 |
| -0.801 | -0.3099 |
| -0.811 | -0.3411 |
| -0.793 | -0.2993 |
| -0.786 | -0.2611 |
| -0.784 | -0.2913 |
| -0.786 | -0.2711 |

เสียง 9

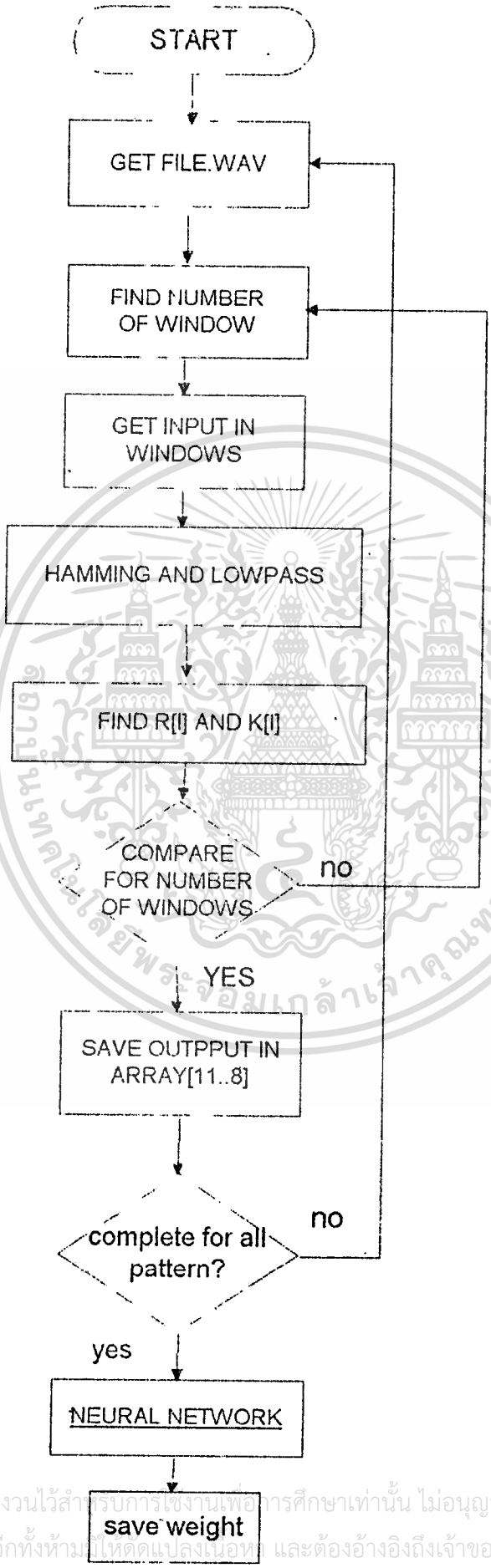
| $K_1$ | $K_2$ |
|---|---|
| -0.949 | -0.0438 |
| -0.840 | -0.2444 |
| -0.821 | -0.3135 |
| -0.811 | -0.3666 |
| -0.813 | -0.4003 |
| -0.793 | -0.4988 |
| -0.776 | -0.5766 |
| -0.785 | -0.5369 |
| -0.803 | -0.4574 |
| -0.810 | -0.4321 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

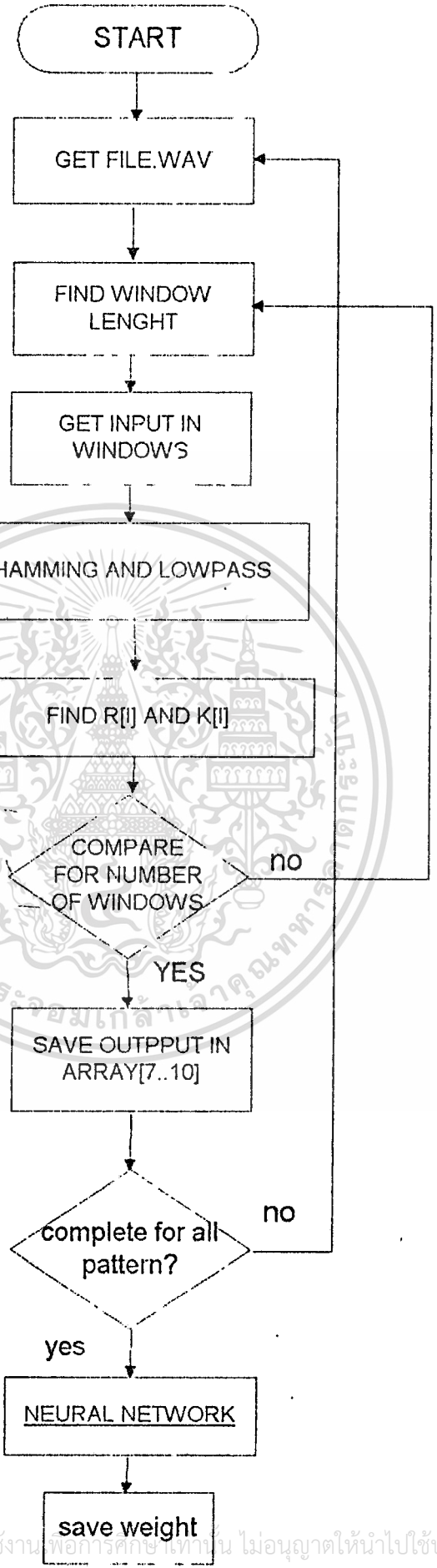
# ภาคผนวก ข.

Flow chart แสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา

# กิตติกรรมประกาศ

ที่รครงงานนี้สำเร็จได้ด้วยดี ก็เนื่องมาจากความช่วยเหลือของอาจารย์ชินภัทรที่คอยแนะนำ และให้คำปรึกษาเป็นอย่างดี อาจารย์เทอดศักดิ์ที่ยินยอมให้ใช้ห้องคอมพิวเตอร์จนเกือบถึงเช้า และเพื่อนๆที่เอื้อเฟื้อที่นอนให้ ขอบคุณทุกๆคนมากครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เอกสารอ้างอิง


1. ยงยุทธ์ เทพเรียน, ทรวง คุปติวิทยากุล, "การรู้จำเสียงพูด" ,คณะวิศวกรรมศาสตร์ ปริญญานิพนธ์ปีการศึกษา 2537

2. บุญเลิศ เอี่ยมทัศนา, "เรียนรู้ภาษาปาสคาลด้วยเทอร์โบปาสคาล 4.0-5.0",ซีเอ็ด, 273 หน้า,2532

3. Panos E. Papamichalis, "Pratical approches to speech coding", Prentice Hall (1987)

4. Maureen Caudull and Chares Butler. " Understanding Neural Networks :Computer Exploration, Volume 1", A Bradford Book The MIT Press (1993)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้