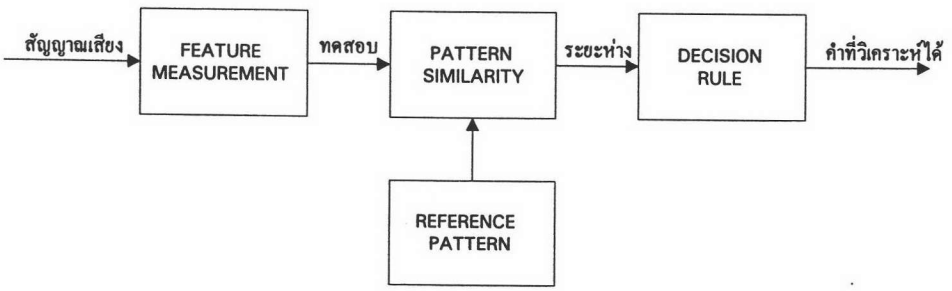
บทที่ 2
หลักการรู้จำเสียงพูด

โมเดลการรู้จำรูปแบบ (pattern recognition) ที่ใช้ในการวิเคราะห์การรู้จำเสียงพูดแบบคำโดด (Isolated word) (Rabiner and Levinson ,1981) แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 โมเดลของการรู้จำรูปแบบ

ซึ่งมี 3 ขั้นตอนในการวิเคราะห์ คือ

1. Feature measurement : ทำหน้าที่สกัดลักษณะเด่นของข้อมูลออกมา และทำการลดจำนวนข้อมูลที่จะถูกประมวลผล
2. Pattern similarity determination : ทำหน้าที่หาค่าความใกล้เคียง ของคำที่เราไม่ทราบ (Unknown word) เทียบกับคำอ้างอิงแต่ละคำ
3. Decision rule :เป็นกฎเกณฑ์ในการเลือกว่าคำที่เราไม่ทราบใกล้เคียงกับคำอ้างอิงคำใดมากที่สุด

สัญญาณเสียงที่จะส่งเข้าไปในขั้นตอน feature measurement ต้องผ่านการพรีโพรเซสซิงก่อน ซึ่งจะทำการเตรียมและปรับข้อมูลให้เหมาะสมกับระบบการรู้จำเสียงพูด การพรีโพรเซสซิงที่ใช้ประกอบด้วย การตัดหัวท้ายคำ (end point detection) และการนอร์แมลไลซ์ (normalization)

2.1 การตัดหัวท้ายคำ (End Point Detection)

การตัดหัวท้ายคำเป็นกระบวนการค้นหาช่วงที่เป็นเสียงพูดจากเสียงที่ได้จากการบันทึก นั่นคือ การแยกส่วนที่เป็นเสียงพูดจากส่วนที่เป็นเสียงพื้นหลัง (background sound) ขั้นตอนนี้เป็นขั้นตอนที่สำคัญ (Rabiner and Levinson,1981) เพราะ

1. ความผิดพลาดในการตัดหัวท้ายคำ จะทำให้ความน่าจะเป็นของความผิดพลาดในการรู้จำเสียงพูดเพิ่มขึ้น

2. การตัดหัวท้ายคำที่ถูกต้อง ช่วยให้การคำนวณทั้งหมดของระบบต่ำสุด

การตัดหัวท้ายคำมีวิธีการหลัก ๆ ดังนี้

1. การตัดหัวท้ายคำโดยใช้แอมพลิจูด (เสาวลักษณ์ อารีย์พงศา,2538) เมื่อสัญญาณมีค่าแอมพลิจูดมากกว่าค่าที่กำหนดไว้เท่ากับจำนวนครั้งที่กำหนด จะให้จุดนั้นเป็นจุดเริ่มต้นของเสียงพูด และทำเช่นเดียวกันในส่วนท้ายของเสียงที่บันทึกมาเพื่อหาจุดสิ้นสุด ข้อดีของวิธีนี้คือ ใช้การคำนวณง่าย ๆ และใช้เวลาในการคำนวณน้อยมาก ข้อเสียของวิธีนี้คือจะตัดคำผิดพลาดเมื่อมีสัญญาณรบกวนที่มีแอมพลิจูดสูงในบริเวณหัวหรือท้ายคำ เช่น เสียงหายใจ

2. การตัดหัวท้ายคำโดยใช้ค่าพลังงาน (Rabiner and Levinson,1981) วิธีนี้ใช้คอนทัวร์ (contour) ของพลังงานในแต่ละส่วนย่อย เพื่อหาจุดที่มีพลังงานมากกว่าระดับที่กำหนดไว้ ติดต่อกันนานกว่าคาบเวลาที่กำหนด จุดเริ่มต้นของเสียงพูดจะอยู่ก่อนจุดที่ตรวจพบด้วยระดับพลังงาน เท่ากับคาบเวลาคงที่ค่าหนึ่ง ข้อดีของวิธีนี้คือ สามารถลดการตัดคำผิดพลาดเมื่อมีสัญญาณรบกวนที่มีแอมพลิจูดสูง ข้อเสียของวิธีนี้คือจุดเริ่มต้นที่คำนวณได้อาจคลาดเคลื่อนจากจุดเริ่มต้นที่แท้จริงของเสียงพูด

3. การตัดหัวท้ายคำโดยใช้ค่าพลังงานและอัตราการตัดค่าศูนย์ (zero-crossing rate)(Furui, 1989) เหมือนกับการตัดหัวท้ายคำโดยใช้ค่าพลังงาน แต่มีการปรับปรุงการหาจุดเริ่มต้นของเสียงพูดโดยใช้อัตราการตัดค่าศูนย์แทนการใช้คาบเวลาคงที่ ทำให้สามารถหาจุดเริ่มต้นได้ถูกต้องมากขึ้น ซึ่งก็ต้องแลกเปลี่ยนด้วยเวลาที่ใช้ในการคำนวณอัตราการตัดค่าศูนย์

ในวิทยานิพนธ์นี้เลือกใช้การตัดหัวท้ายคำโดยใช้ค่าพลังงาน เพราะวิธีนี้สามารถแก้ปัญหาการตัดคำผิดพลาดได้ โดยที่ไม่ใช้เวลาในการคำนวณมากเกินไป ถึงแม้ว่าจุดเริ่มต้นของเสียงพูดที่คำนวณได้อาจคลาดเคลื่อนไปบ้าง แต่ก็สามารถแก้ไขได้โดยใช้การประมาณค่าคาบเวลาที่เหมาะสมกับกลุ่มคำที่ต้องการรู้จำ

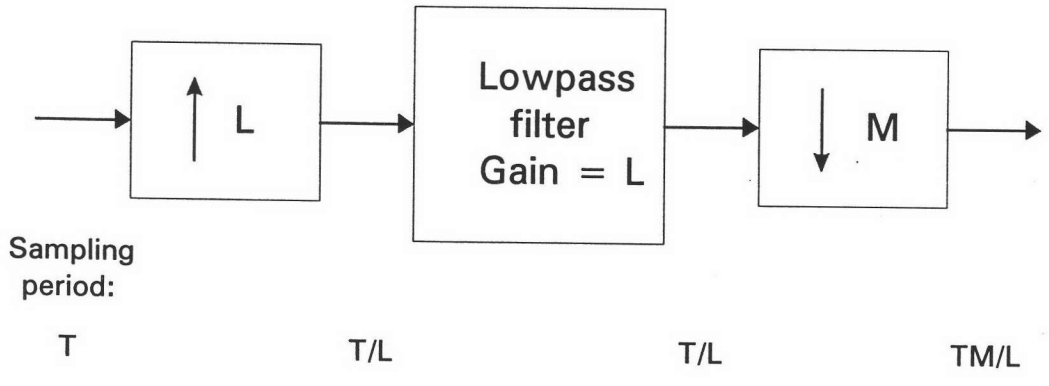
2.2 การนอร์แมลไลซ์ (Normalization)

เป็นกระบวนการที่ทำให้สัญญาณเสียงพูดแต่ละคำมีจำนวนจุดสัญญาณในแกนเวลาเป็นจำนวนเท่ากัน กระบวนการนี้เป็นกระบวนการที่จำเป็นเพราะเสียงพูดแต่ละคำมีความยาวไม่เท่ากัน แต่นิวรอลเน็ตเวิร์กมีจำนวนโหนดในระดับข้อมูลเข้าคงที่ การนอร์แมลไลซ์มีวิธีการหลัก ๆ ดังนี้

1. การประมาณค่าในช่วงเชิงเส้น (linear interpolation) จะทำการประมาณค่าแอมพลิจูดของสัญญาณที่จุดไม่ทราบค่าจากความสัมพันธ์เชิงเส้นของจุดสัญญาณเดิมที่ได้จากการบันทึกเสียงพูด

ที่อยู่ล้อมรอบจุดที่ไม่ทราบค่า ข้อดีของวิธีนี้คือคำนวณได้ง่าย ข้อเสียคือทำให้เกิดการเคลือบแฝง (aliasing) ในสัญญาณที่ประมาณค่า

2. การประมาณค่าโดยใช้การเปลี่ยนอัตราการชักตัวอย่าง (sampling rate) (Oppenheim , 1989) วิธีนี้มีขั้นตอนดังแสดงในรูปที่ 2.2 โดยนำสัญญาณเสียงพูดมาเพิ่มอัตราการชักตัวอย่างขึ้น L เท่า โดยการเพิ่มจุดศูนย์ (zero-packing) ระหว่างแต่ละจุดสัญญาณเดิม จากนั้นนำสัญญาณที่ได้ไปผ่านวงจรกรองแบบผ่านต่ำ แล้วลดอัตราการชักตัวอย่างลง M เท่า โดยที่ M เป็นจำนวนจุดสัญญาณเดิมที่ได้จากการบันทึกและ L เป็นจำนวนจุดสัญญาณที่ต้องการ ข้อดีของวิธีนี้คือป้องกันการเกิดการเคลือบแฝงได้ ข้อเสียของวิธีนี้คือใช้หน่วยความจำและเวลาในการคำนวณมาก



รูปที่ 2.2 การนอร์แมลไลซ์โดยใช้การเปลี่ยนอัตราการชักตัวอย่าง

ในวิทยานิพนธ์นี้เลือกใช้วิธีการประมาณค่าในช่วงเชิงเส้นในการนอร์แมลไลซ์สัญญาณเสียงเพราะใช้เวลาในการคำนวณน้อยมาก นอกจากนี้วิธีนี้ยังไม่ต้องใช้หน่วยความจำปริมาณมากในการเก็บข้อมูลในขณะที่วิธีการประมาณค่าโดยใช้การเปลี่ยนอัตราการชักตัวอย่างต้องใช้ในการเก็บข้อมูลขณะเพิ่มอัตราการชักตัวอย่างขึ้น

2.3 การวัดค่าลักษณะสำคัญ (Feature Measurement)

เป็นเทคนิคการลดจำนวนข้อมูล โดยที่ข้อมูลจำนวนมากจะถูกแปลงเป็นชุดของข้อมูลที่มีจำนวนน้อยลง และยังคงแสดงคุณสมบัติสำคัญของรูปคลื่นสัญญาณเสียงได้อย่างถูกต้อง โดยทั่วไปสัญญาณเสียงถูกวิเคราะห์โดยใช้ลักษณะเด่นเชิงสเปกตรัม (spectral feature) เพราะลักษณะเด่นส่วนใหญ่สำหรับการรับรู้เสียงพูดโดยหูของมนุษย์ รวมอยู่ในข้อมูลเชิงสเปกตรัม วิธีการสกัดเอนเวโลปเชิงสเปกตรัม (spectral envelope) แบ่งออกเป็นการวิเคราะห์โดยใช้พารามิเตอร์ (parametric analysis) และการวิเคราะห์โดยไม่ใช้พารามิเตอร์ (nonparametric analysis) การวิเคราะห์โดยใช้พารามิเตอร์จะ

เลือกแบบจำลองที่เหมาะสมกับสัญญาณ และปรับแต่งพารามิเตอร์ลักษณะเด่นที่ใช้แทนแบบจำลองนั้น ในขณะที่การวิเคราะห์โดยไม่ใช้พารามิเตอร์สามารถประยุกต์ใช้กับสัญญาณหลายชนิดได้เพราะการวิเคราะห์วิธีนี้ไม่ได้สร้างแบบจำลองสัญญาณ ถ้าแบบจำลองที่ใช้มีความเหมาะสมกับสัญญาณ การวิเคราะห์โดยใช้พารามิเตอร์จะสามารถแสดงลักษณะเด่นของสัญญาณได้ดีกว่า

การวิเคราะห์โดยไม่ใช้พารามิเตอร์ มีวิธีการหลัก ๆ ดังนี้

1. ชุดวงจรกรองผ่านแถบ (band-pass filter bank) วิธีนี้นำสัญญาณเสียงมาผ่านวงจรกรองผ่านแถบหลายวงจร ที่มีช่วงความถี่ผ่านแตกต่างกัน วงจรกรองผ่านแถบแต่ละวงจรจะให้สัญญาณเอาต์พุตที่สัมพันธ์กับพลังงานของสัญญาณในช่วงความถี่ผ่านของวงจรกรองนั้น วิธีนี้มีข้อดีคือสร้างเป็นฮาร์ดแวร์ได้ง่ายและเหมาะสมสำหรับการประมวลผลเวลาจริง (real-time processing)

2. การวิเคราะห์การตัดค่าศูนย์ (zero-crossing analysis) จะนับจำนวนการเปลี่ยนเครื่องหมายของสัญญาณ ซึ่งเป็นการประมาณค่าความถี่ฟอร์แมนท์ (formant frequency) คือ ความถี่ที่มีพลังงานสูงสุด การวิเคราะห์วิธีนี้มักใช้ร่วมกับวิธีชุดวงจรกรองผ่านแถบ

3. การวิเคราะห์โดยใช้เซ็ปสตรัม (cepstrum) การวิเคราะห์วิธีนี้มีข้อดีคือ สามารถแยกเอนเวโลปเชิงสเปกตรัมและโครงสร้างย่อยเชิงสเปกตรัม (spectral fine structure) ออกจากกันได้ในโดเมนคิวเฟรินซี่ (quefrency domain) ซึ่งเป็นพารามิเตอร์ในโดเมนเวลา แต่มีข้อเสียคือต้องคำนวณผลการแปลงฟูริเยร์แบบเร็ว (fast fourier transform) 2 ครั้ง และคำนวณค่าลอการิทึม ซึ่งต้องใช้เวลาในการคำนวณมาก

การวิเคราะห์โดยใช้พารามิเตอร์ มีวิธีการหลัก ๆ ดังนี้

1. การวิเคราะห์โดยการสังเคราะห์ (analysis-by-synthesis) วิธีนี้สามารถสร้างแบบจำลองที่ถูกต้องแม่นยำได้ โดยการใช้พารามิเตอร์หลายค่าเช่น ค่าความถี่ฟอร์แมนท์,ความกว้างแถบ (bandwidth), เอนเวโลปเชิงสเปกตรัมและอื่น ๆ แต่มีข้อเสียคือต้องใช้เวลาคำนวณในการวนซ้ำมาก เพราะค่าพารามิเตอร์หลายค่ามีผลกระทบต่อกัน

2. การประมาณพันธะเชิงเส้น (linear predictive coding) เป็นการสร้างแบบจำลองของสเปกตรัมอย่างง่ายโดยใช้โพล (all-pole spectrum modeling) พารามิเตอร์สามารถประมาณค่าได้จากค่าความแปรปรวนร่วมหรือค่าอัตสหสัมพันธ์ โดยไม่ใช้การวนซ้ำ วิธีนี้มีข้อดีคือสามารถแทนสัญญาณเสียงได้อย่างมีประสิทธิภาพโดยใช้พารามิเตอร์จำนวนน้อยและใช้การคำนวณที่ค่อนข้างง่าย

2.3.1 การประมาณพันธะเชิงเส้น (Linear Predictive Coding)

ในวิทยานิพนธ์นี้ใช้การประมาณพันธะเชิงเส้น (Linear Predictive Coding) ซึ่งเป็นวิธีที่ใช้กันแพร่หลายในการหา feature ของเสียง การประมาณพันธะเชิงเส้น (Linear Predictive Coding) เป็นขบวนการทางคณิตศาสตร์ที่ใช้ในการหาเอกลักษณ์ของระบบ โดยพิจารณาว่าเสียงเกิด

จากผลรวมเชิงเส้น (linear combination) ของสัญญาณที่ทราบค่าแล้ว โดยใช้วิธีกำลังสองน้อยที่สุด (least-squares method) ในการเลือกค่าพารามิเตอร์ของระบบ หลักการประมาณพันธะเชิงเส้นมีวิธีการหลัก 2 วิธีคือวิธีการหาค่าความแปรปรวนร่วมและวิธีอัตสหสัมพันธ์ เนื่องจากวิธีอัตสหสัมพันธ์ใช้การคำนวณน้อยกว่าวิธีความแปรปรวนร่วม และมีความแน่นอนด้านเสถียรภาพ (Furui,1989) ดังนั้นวิทยานิพนธ์นี้จึงเลือกใช้การประมาณพันธะเชิงเส้นโดยวิธีอัตสหสัมพันธ์ การประมาณพันธะเชิงเส้นสามารถแสดงคุณสมบัติได้ใกล้เคียงกับพื้นฐานโมเดลการกำเนิดเสียงของมนุษย์ (Rabiner and Levinson , 1981)

2.3.1.1 หลักการประมาณพันธะเชิงเส้น

สมมุติให้แบบจำลองการสร้างสัญญาณเสียง ประกอบด้วย แหล่งกำเนิดสัญญาณกระตุ้น $U(z)$ ซึ่งจะป้อนสัญญาณเข้าสู่วงจรกรองจัดรูปสัญญาณ $H(z)$ ทำให้ได้สัญญาณเสียง $S(z)=U(z)H(z)$ วงจรกรอง $H(z)$ สามารถจำลองได้โดยใช้ค่าสัมประสิทธิ์ดังนี้

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} \tag{2.1}$$

ถ้าป้อนสัญญาณ $s(n)$ เข้าสู่วงจรกรอง predictor (ส่วนกลับของวงจรกรองจัดรูปสัญญาณ $H(z)$)

$$A(z) = 1-\sum_{k=1}^{p} a_k z^{-k} \tag{2.2}$$

สัญญาณออก $e(n)$ เรียกว่าสัญญาณค่าความผิดพลาด แสดงได้โดย

$$e(n) = s(n)-\sum_{k=1}^{p} a_k s(n-k) \tag{2.3}$$

2.3.1.2 การประมาณพันธะเชิงเส้นโดยวิธีอัตสหสัมพันธ์

การประมาณพันธะเชิงเส้นโดยวิธีอัตสหสัมพันธ์ จะทำการคูณสัญญาณด้วยฟังก์ชันหน้าต่าง (window function) เพื่อจำกัดให้สัญญาณ $x(n)=w(n)s(n)$ มีค่าอยู่ในช่วงเวลาจำกัดเท่ากับ N ข้อมูลชักตัวอย่าง (sampled data) ดังนั้น $x(n)=0$ เมื่อ $n<0$ และ $n\geq N$ แล้วคำนวณค่าสัมประสิทธิ์ LPC เพื่อใช้เป็นตัวแทนของสัญญาณในแต่ละช่วงเวลานี้ กำหนดให้ E เป็นพลังงานของความผิดพลาด

$$E = \sum_{-\infty}^{\infty} e^2(n) = \sum_{-\infty}^{\infty} [x(n) - \sum_{k=1}^{p} a_k x(n-k)]^2 \qquad (2.4)$$

ค่าสัมประสิทธิ์ a_k ที่ทำให้ E มีค่าน้อยที่สุดสามารถหาได้โดยการแก้สมการ $\frac{\partial E}{\partial a_k} = 0$ เมื่อ $k = 1,2,3,\ldots,p$ ซึ่งจะได้สมการเชิงเส้น p สมการดังนี้

$$\sum_{n=-\infty}^{\infty} x(n-i)x(n) = \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} x(n-i)x(n-k) \text{ เมื่อ } i = 1,2,3,\ldots,p \qquad (2.5)$$

เนื่องจากพจน์ด้านซ้ายมือของสมการคือค่าอัตสหสัมพันธ์ $R(i)$ ของ $x(n)$ และ $x(n)$ มีค่าในช่วงเวลาจำกัด จะได้

$$\sum_{k=1}^{p} a_k R(i-k) = R(i) \qquad ; \quad 1 \le i \le p \qquad (2.6)$$

โดยที่ค่าอัตสหสัมพันธ์ $R(i)$ คำนวณได้จาก

$$R(i) = \sum_{n=i}^{N-1} x(n)x(n-i) \qquad (2.7)$$

จากสมการ (2.6) เราสามารถเขียนอยู่ในรูปของเมตริกซ์ได้ดังนี้

$$\begin{bmatrix} R_0 & R_1 & \ldots & R_{1-p} \\ R_1 & R_0 & \ldots & R_{2-p} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ R_{p-1} & R_{p-2} & \ldots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ . \\ . \\ . \\ R_P \end{bmatrix} \qquad (2.8)$$

เมื่อแก้สมการเชิงเส้น p สมการนี้จะได้ค่าสัมประสิทธิ์ a_k ที่ใช้ในการประเมินค่าสัญญาณจำนวน p ค่า เนื่องจากเมตริกซ์ของค่าอัตสหสัมพันธ์ อยู่ในรูปของเมตริกซ์ Toeplitz กล่าวคือมีลักษณะสมมาตรและทุก ๆ สมาชิกในแนวทแยงมุมมีค่าเท่ากัน ทำให้การแก้สมการเพื่อหาค่าสัมประสิทธิ์โดยวิธีอัตสหสัมพันธ์สามารถคำนวณได้ง่ายกว่าวิธีความแปรปรวนร่วม ในการหาค่าสัมประสิทธิ์

a_k วิทยานิพนธ์นี้ใช้วิธีของ Levinson-Durbin (Douglas O'Shaughnessy ,1988) ซึ่งเป็นวิธีที่มีประสิทธิภาพในการแก้สมการที่อยู่ในรูปเมตริกซ์ Toeplitz โดยทำการคำนวณลำดับของสมการสำหรับ $m = 1,2,\ldots,p$

$$
\begin{aligned}
k_m &= \frac{R(m) - \sum_{i=1}^{m-1} a_{m-1}(i)R(m-i)}{E_{m-1}} \\
a_m(m) &= k_m \\
a_m(i) &= a_{m-1}(i) - k_m a_{m-1}(m-i) \qquad ; \quad 1 \le i < m \\
E_m &= (1-k_m^2)E_{m-1}
\end{aligned} \tag{2.9}
$$

โดยที่ $E_0 = R(0)$ และ $a_0 = 0$ ในแต่ละรอบของ m ค่าสัมประสิทธิ์ $a_m(i)$ เมื่อ $i = 1,2,\ldots,m$ แทนการประมาณพันธะเชิงเส้นอันดับ (order) m ที่เหมาะที่สุด

2.4 การหาความคล้ายคลึงกันของรูปแบบ (Pattern Similarity Determination)

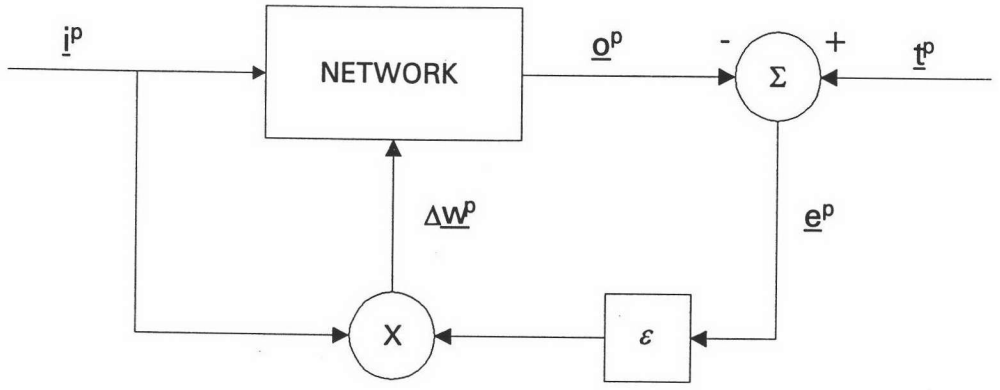
เป็นการหาค่าความใกล้เคียงของคำที่เราไม่ทราบเทียบกับคำอ้างอิงแต่ละคำ วิธีที่ใช้ในขั้นตอนนี้มีหลายวิธีเช่น DTW, HMM, นิวรอลเน็ตเวิร์ก DTW ใช้เทคนิคการปรับยืดขยายหรือหดรูปคลื่นสัญญาณตามแกนเวลาแบบไดนามิค วิธีนี้ถูกนำมาใช้ในการรู้จำเสียงสระภาษาไทยโดย ธีระ ภัทราพรนันท์ (2538) วิธี HMM ถูกนำมาใช้ในการรู้จำเสียงตัวเลขภาษาไทยโดย เสาวลักษณ์ อารีย์พงศา (2538) ขีดจำกัดของวิธี DTW และวิธี HMM คือ เมื่อเพิ่มจำนวนคำที่ต้องการรู้จำมากขึ้นทำให้เสียเวลาในการทดสอบมากขึ้น เพราะต้องทดสอบกับแบบอ้างอิงของเสียงทุกคำ ส่วนวิธีนิวรอลเน็ตเวิร์กยังคงใช้เวลาในการทดสอบเท่าเดิม เพราะนิวรอลเน็ตเวิร์กเก็บความรู้เกี่ยวกับลักษณะของเสียงทุก ๆ คำรวมอยู่ในน้ำหนักการเชื่อมต่อ ไม่ได้แยกเก็บเป็นแบบอ้างอิงสำหรับแต่ละคำ อย่างไรก็ตามถ้าเพิ่มจำนวนคำขึ้นมาก ๆ ต้องเพิ่มขนาดของนิวรอลเน็ตเวิร์กขึ้นด้วย เพื่อให้นิวรอลเน็ตเวิร์กมีความสามารถเพียงพอในการรู้จำเสียง

2.4.1 นิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กแบ่งแยกตามลักษณะของการเรียนรู้ได้เป็น 2 ชนิดคือ unsupervised learning และ supervised learning ในวิทยานิพนธ์นี้เลือกใช้นิวรอลเน็ตเวิร์กแบบ multi-layer perceptron ซึ่งอยู่ในประเภท supervised learning

2.4.1.1 ขั้นตอนการฝึก (training) นิวรอลเน็ตเวิร์ก

Multi-layer perceptron neural network ใช้การฝึก (training) แบบ error backpropagation หรือ generalized delta rule ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 โครงสร้างของการฝึก

โดยที่ $\underline{i}^p$ แทนค่าเวกเตอร์ input pattern ลำดับที่ p

$\underline{o}^p$ แทนค่าเวกเตอร์ output pattern ลำดับที่ p ที่ได้จากเน็ตเวิร์ก

$\underline{\omega}^p$ แทนค่าเวกเตอร์ network weights เมื่อใส่ค่าอินพุตลำดับที่ p เข้าสู่เน็ตเวิร์ก

$\underline{t}^p$ แทนค่าเวกเตอร์ output pattern ลำดับที่ p ที่ต้องการ

ε แทนค่า learning rate

$\underline{e}^p$ แทนค่าเวกเตอร์ความผิดพลาดของเอาต์พุตลำดับที่ p

นิวรอลเน็ตเวิร์กจะเรียนรู้จาก แต่ละตัวอย่างคู่ข้อมูลอินพุตเอาต์พุต $(\underline{i}^p, \underline{t}^p)$ ที่อยู่ในชุดฝึก (training set) ซึ่งมีขั้นตอนพื้นฐานดังนี้

-ป้อนค่าเวกเตอร์อินพุต (input vector) ให้กับระดับข้อมูลเข้า (input layer) ของนิวรอลเน็ตเวิร์ก

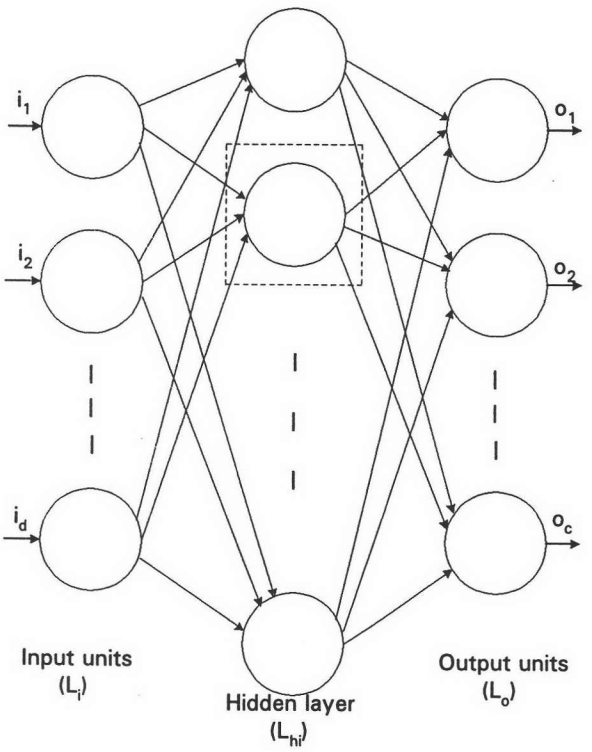
-'Feed forward' หรือแพร่กระจายค่าอินพุต (input) เพื่อหาค่าเอาต์พุต (output) ของทุกโหนด

-เปรียบเทียบค่าเอาต์พุต $\underline{o}^p$ ในระดับข้อมูลออก (output layer) กับค่าเอาต์พุตที่ต้องการ $\underline{t}^p$

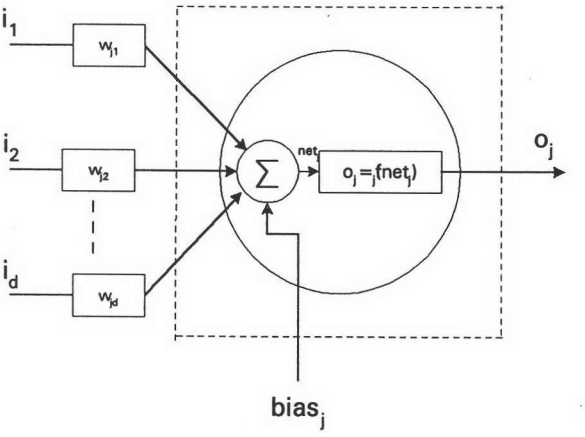
-คำนวณและแพร่กระจายค่าความผิดพลาด ในทิศทางย้อนกลับ (เริ่มจากระดับข้อมูลออก (output layer)) ตลอดทั้งเน็ตเวิร์ก

-ลดค่าความผิดพลาดที่แต่ละระดับ (layer) โดยการปรับค่าน้ำหนัก (weight) ที่เชื่อมต่อระหว่างโหนดของแต่ละระดับ

การฝึกนี้จะถูกทำซ้ำไปเรื่อย ๆ จนกว่าค่าความผิดพลาดจะอยู่ในระดับที่ยอมรับได้ จึงจะยุติการฝึก ค่าน้ำหนักการเชื่อมต่อ (connection weight) ของนิวรอลเน็ตเวิร์กที่ผ่านการฝึกแล้ว เปรียบเทียบได้กับความรู้ที่ได้รับจากการฝึกจากตัวอย่างคู่ข้อมูลอินพุตเอาต์พุต ดังนั้นถ้ามีตัวอย่างคู่ข้อมูลอินพุตเอาต์พุตที่หลากหลาย จะทำให้นิวรอลเน็ตเวิร์กมีความรู้เพียงพอที่จะใช้ในการเปรียบเทียบเสียง



รูปที่ 2.4 โครงสร้างของ multi-layer perceptron neural network



รูปที่ 2.5 รายละเอียดของโหนดในนิวรอลเน็ตเวิร์ก

รูปที่ 2.4 แสดงโครงสร้างของ multi-layer perceptron neural network ซึ่งประกอบด้วย ระดับข้อมูลเข้า (input layer) ,ระดับซ่อนตัว (hidden layer) ซึ่งอาจมีมากกว่า 1 ระดับ และระดับข้อมูลออก (output layer) ระดับข้อมูลเข้า (input layer) มีจำนวนโหนดเท่ากับ d โหนด ระดับข้อมูลออก (output layer) มีจำนวนโหนดเท่ากับ c โหนด ที่แต่ละโหนดในชั้นใด ๆ จะมีค่าน้ำหนักการเชื่อมต่อที่เชื่อมต่อไปยังโหนดที่อยู่ในชั้นถัดไปที่อยู่ติดกันเท่านั้น รูปที่ 2.5 แสดงรายละเอียดของโหนดในนิวรอลเน็ตเวิร์กโดยที่ ค่า net input ที่โหนด j แสดงได้ดังนี้

$$net_j^p = \sum_i \omega_{ji} \tilde{o}_i^p + bias_j \qquad (2.10)$$

เมื่อ $\tilde{o}_i^p = o_i^p$ ถ้าอินพุตเป็นค่าเอาต์พุตของโหนดในระดับ (layer) ที่อยู่ข้างหน้า (เมื่อ j เป็นโหนดในระดับซ่อนตัว (hidden layer) และระดับข้อมูลออก (output layer))

$= i_i^p$ ถ้าอินพุตเป็นค่าข้อมูลอินพุตที่ป้อนเข้าสู่เน็ตเวิร์ก

ω_{ji} เป็นค่าน้ำหนักการเชื่อมต่อที่เชื่อมต่อจากโหนด i ไปยังโหนด j ที่อยู่ในระดับถัดไป

$bias_j$ เป็นค่าที่ใช้ปรับให้ net_j มีค่าไม่เท่ากับศูนย์ ในกรณีที่ $\tilde{o}_i^p$ ทุกโหนดมีค่าเป็นศูนย์หมด

ค่าเอาต์พุต o_j ของโหนด j สามารถคำนวณได้จาก net_j ดังนี้

$$o_j^p = f_j(net_j) \qquad (2.11)$$

โดยที่ฟังก์ชันกระตุ้น (activation function) f_j เป็นฟังก์ชันเพิ่มและเป็นฟังก์ชันที่สามารถหาอนุพันธ์ได้ (differentiable) ในวิทยานิพนธ์นี้เลือกใช้ฟังก์ชัน sigmoid

$$f_j(net_j) = \frac{1}{1+e^{-net_j}} \tag{2.12}$$

2.4.1.2 การปรับค่าน้ำหนักการเชื่อมต่อ

เวกเตอร์ค่าความผิดพลาดของเอาต์พุต สำหรับตัวอย่างคู่ข้อมูลอินพุตเอาต์พุตที่ p กำหนดโดย

$$\underline{e}^p = \underline{t}^p - \underline{o}^p \tag{2.13}$$

E_p แทนค่าความผิดพลาดของเอาต์พุต สำหรับตัวอย่างคู่ข้อมูลอินพุตเอาต์พุตที่ p หาได้จาก

$$E_p = \frac{1}{2}\Sigma_j (t_j^p - o_j^p)^2 \tag{2.14}$$

หลักการของการปรับค่าน้ำหนักการเชื่อมต่อใน backpropagation training เริ่มต้นจากการคำนวณพื้นผิวของค่าความผิดพลาด E และคำนวณค่าเกรเดียนต์ (gradient) ของ E เทียบกับค่าน้ำหนักการเชื่อมต่อ $\partial E / \partial \omega_{ji}$ การปรับค่าน้ำหนักการเชื่อมต่อ $\Delta\omega_{ji}$ จะปรับค่าเป็นสัดส่วนกับ $-\partial E / \partial \omega_{ji}$ เพื่อให้การปรับน้ำหนักการเชื่อมต่อเป็นไปในทิศทางที่ลดค่าผิดพลาดลง สมการสำหรับการปรับค่าน้ำหนักการเชื่อมต่อแสดงได้ดังนี้

$$\Delta^p \omega_{ji} = \varepsilon \delta_j^p \tilde{o}_i^p \tag{2.15}$$

เมื่อ $\tilde{o}_i^p$ มีค่าตามที่แสดงในสมการ 2.10

ε คือค่า learning rate ซึ่งเป็นค่าคงที่และมีค่าเป็นบวก

สำหรับค่า δ_j^p คือค่าความไว (sensitivity) ของค่าความผิดพลาดเทียบกับค่า net input ที่โหนด j

$$\delta_j^p = -\frac{\partial E_p}{\partial net_j^p} \tag{2.16}$$

สำหรับโหนดในระดับข้อมูลออก (output layer) $\delta_j^p = (t_j^p - o_j^p) f_j'(net_j^p)$

สำหรับโหนดใน internal layer $\delta_j^p = f_j'(net_j^p)\sum_n \delta_n^p \omega_{nj}$ เมื่อ δ_n^p เป็นค่าความไว (sensitivity) ในชั้นถัดออกไป

อนุพันธ์ของฟังก์ชันกระตุ้นชนิด sigmoid อยู่ในรูปที่คำนวณได้ง่าย ซึ่งนับเป็นข้อดีของฟังก์ชันชนิดนี้ กำหนดโดย

$$f_j'(net_j^p) = o_j^p(1 - o_j^p) \tag{2.17}$$

2.5 กฎเกณฑ์การตัดสินใจ (Decision Rule)

การใช้กฎเกณฑ์ใดตัดสินว่าคำที่เราไม่ทราบ (Unknown word) คือเสียงใด เราจะต้องคำนึงถึงวิธีการที่ใช้ในขั้นตอน Pattern similarity determination เพราะต้องมีความสอดคล้องกัน เนื่องจาก multi-layer perceptron neural network ที่ผ่านการฝึกแล้ว จะให้ค่าเอาต์พุตที่คล้ายคลึงกับค่าเอาต์พุตของตัวอย่างคู่ข้อมูลอินพุตเอาต์พุตที่มีค่าอินพุตของข้อมูลฝึกคล้ายกับค่าอินพุตที่ป้อนเข้ามา ดังนั้นเกณฑ์การตัดสินใจที่เลือกใช้คือการเลือกเสียงที่ตรงกับโหนดเอาต์พุตที่มีค่าเอาต์พุตสูงสุด อีกเหตุผลหนึ่งที่ใช้สนับสนุนเกณฑ์การตัดสินใจนี้คือ การทำงานของนิวรอลเน็ตเวิร์กขณะฝึกมีการปรับน้ำหนักการเชื่อมต่อในทิศทางที่ลดความผิดพลาดให้เหลือน้อยที่สุด ดังนั้นนิวรอลเน็ตเวิร์กที่ผ่านการฝึกแล้วจะให้ค่าเอาต์พุตที่มีความผิดพลาดน้อยที่สุดบนพื้นฐานความรู้ที่นิวรอลเน็ตเวิร์กได้รับจากการฝึก โหนดเอาต์พุตที่มีค่าเอาต์พุตสูงสุดคำนวณได้จาก

$$nodeoutput = i \quad \text{เมื่อ } o_i = Max(o_1, o_2, \ldots, o_c) \tag{2.18}$$

โดยที่ c คือจำนวนกลุ่มของเสียงที่ต้องการรู้จำ