CHƯƠNG 1: MỞ ĐẦU

Ngôn điệu chính là cái mang lại cho tiếng nói con người những âm sắc riêng biệt. Ngôn điệu của lời nói liên kết chặt chẽ với ngữ điệu.Ngữ điệu là sự nâng cao hạ thấp của giọng nói trong câu. Tiếng Việt ta là một ngôn ngữ khá phức tạp bao gồm cả ngôn điệu và ngữ điệu. Do đó vấn đề nghiên cứu các phương pháp nhận dạng tiếng nói đã và đang thu hút rất nhiều sự đầu tư và nghiên cứu của nhà khoa học.Tuy nhiên cho đến nay kết quả mang lại vẫn chưa hoàn thiện do tính chất quá phức tạp và không cố định của đối tượng nhận dạng là tiếng nói con người,đặc biệt là tiếng Việt.

Hiện nay có rất nhiều phương pháp nhận dạng tiếng nói. Mô hình Fujisakiđược ứng dụng rộng rãi trong hệ thống của tiếng Nhật, mô hình MFGI (Mixdorff- Fujisaki model of German Intonation) được ứng dụng trong tiếng Đức, mô hình HMM (hidden markov models)…

Trong các mô hình ấy lại áp dụng nhiều phương pháp nhận dạng khác nhau. Mọi phương pháp mang một tính đặc trưng và ưu điểm riêng.

- Phương pháp LPC (linear predictive coding)-mã hóa dự báo tuyến tính: nhược điểm là có một số từ phát âm gần giống nhau thì bị nhầm lẫn nhiều.

- Phương pháp AMDF (average magnitude difference function)- hàm hậu biên độ trung bình: ưu điểm là số ngõ vào ít,kích thước mạng huấn luyện nhỏ,ít phụ thuộc vào cách phát âm nên tỉ lệ đọc sai ít hơn phương pháp LPC, tuy nhiên khuyết điểm là không phân biệt về thanh điệu,khó sử ụng trong trường hợp từ đọc liên tiếp.

- AMDF & LPC :Do ưu và nhược điểm của hai phương pháp LPC và AMDF nên cần sự kết hợp giữa hai phương pháp đó.

- Phương pháp thứ tư MFCC (mel-frequency ceptrums coefficients).

Nhận dạng tiếng nói là một quá trình nhận dạng mẫu,với mục đích là phân lớp thông tin đầu vào là tín hiệu tiếng nói thành một dãy tuần tự các mẫu đã được học trước đó và lưu trữ trong bộ nhớ. Các mẫu là các đơn vị nhận dạng, chúng có thể là các từ hay là các âm vị. Nếu các mẫu này là bất biến và không thay đổi thì công việc nhận dạng tiếng nói trở nên đơn giản bằng cách so sánh dữ liệu tiếng nói cần nhận dạng với các mẫu đã được học và lưu trữ trong bộ nhớ.

CHƯƠNG 2: LÝ THUYẾT ÂM THANH VÀ TIẾNG NÓI

2.1 Nguồn gốc âm thanh:

Âm thanh là do vật thể dao động cơ học mà phát ra. Âm thanh phát ra dưới dạng sóng âm. Sóng âm là sự biến đổi các tính chất của môi trường đàn hồi khi năng lượng âm truyền qua. Âm thanh truyền được đến tai người là do môi trường dẫn âm. Sóng âm có thể truyền được trong chất rắn ,chất lỏng, không khí. Có chất dẫn âm rất kém gọi là chất hút âm như: len,da, chất xốp… Sóng âm không thể truyền trong môi trường chân không.

Khi kích thích dao động âm trong mối trường không khí thì những lớp khí sẽ bị nén và dãn.Trạng thái nén dãn lần lượt được lan truyền từ nguồn âm dưới dạng sóng dọc tới nơi thu âm. Nếu cường độ nguồn âm càng lớn thì âm thanh truyền đi càng xa.

2.2 Các đại lượng đặc trưng cho âm thanh:

a/ Tần số của âm thanh: là số lần dao động của phần tử khí trong một giây. Đơn vị là Hz , kí hiệu : f

b/ Chu kì của âm thanh: là thời gian mà âm thanh đó thực hiện một dao động hoàn toàn. Đơn vị là thời gian, kí hiệu là T.

c/ Tốc độ truyền âm: là tốc độ truyền năng lượng âm từ nguồn tới nơi thu. Đơn vị m/s. Tốc độ truyền âm trong không khí ở nhiệt độ từ 0- 20^0 C thường là 331 – 340 m/s.

d/ Cường độ âm thanh: là năng lượng được sóng âm truyền trong một đơn vị thời gian qua một đơn vị diện tích đặt vuông góc với phương truyền âm.

e/ Thanh áp: là lực tác dụng vào tai người nghe hoặc tại một điểm nào đó của trường âm thanh. Đơn vị : 1pa=1 N/m^2 hoặc 1bar = 1dyn/cm^2 .

f/ Âm sắc: Trong thành phần của âm thanh, ngoài tần số cơ bản còn có các sóng hài ,số lượng sóng hài biểu diễn sắc thài của âm. Âm sắc là một đặc tính của âm nhờ đó mà ta phân biệt được tiếng trầm, bổng khác nhau, phân biệt được tiếng nhạc cụ, tiếng nam nữ ,tiếng người này với người khác.

k/ Âm lượng: là mức độ to nhỏ của nguồn. Đơn vị là W.

2.3 Các tần số của âm thanh:

F_0 gọi là tần số cơ bản của âm thanh. Nam giới f_0 = 150 Hz. Nữ giới : f_0 = 250 Hz.

Giọng nam trầm 80 – 320 Hz
Giọng nam trung 100 – 400 Hz
Giọng nam cao 130 – 480 Hz
Giọng nữ thấp 160 – 600 Hz
Giọng nữ cao 260 – 1200 Hz

Công suất của tiếng nói , khi nói to nhỏ cũng khác nhau.Khi nói thầm công suất 10^{-3} mW, nói bình thường 10 mW, nói to 10^3 mW.

2.4 Cơ chế tạo lập tiếng nói của con người:

Các cơ quan phát âm của con người chủ yếu gồm phổi, khí quản, thanh quản, bộ phận mũi và miệng.Thanh quản có hai nếp gấp gọi là dây thanh âm, dây thanh âm sẽ rung khi luồng không khí đi qua khe thanh môn là khe giữa hai dây thanh âm.Bộ phận miệng là một ống âm không đều.Bộ phận mũi cũng là một ống âm học không đều có diện tích và chiều dài cố định, bắt đầu từ lỗ mũi đến vòm miệng mềm.

Quá trình tạo ra âm phi mũi: vòm miệng mềm ngăn chặn bộ phận mũi và âm thanh phát ra thông qua môi.Đối với quá trình tạo ra âm mũi :vòm miệng mềm hạ xuống và bộ phận mũi liên kết bộ phận miệng, lúc này phía trước của bộ phận miệng khép lại hoàn toàn và âm thanh ra thông qua mũi.Đối với âm thanh nói giọng mũi, âm thanh phát ra cả mũi và môi.

Âm thanh của tiếng nói có thể chia làm ba loại khác nhau:

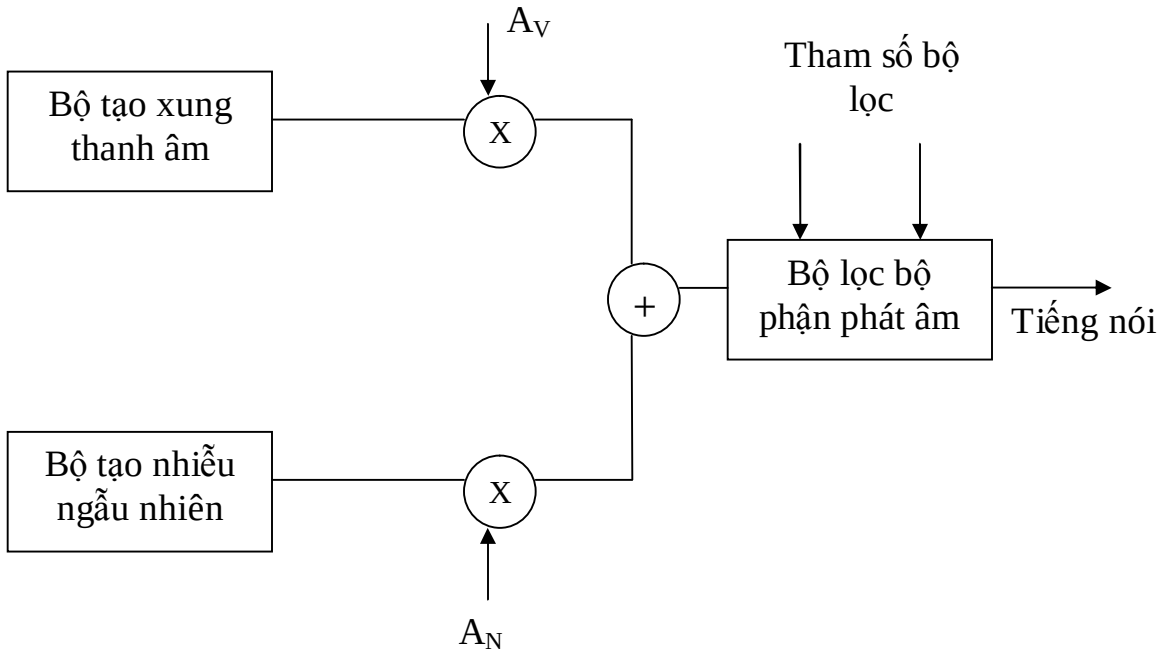
- Âm hữu thanh: giống như âm khi chúng ta nói ‘a’ hay ‘e’,được tạo ra khi dây thanh âm căng lên và rung khi áp suất không khí tăng lên, làm thanh môn mở ra rồi đóng lại khi luồng không khí đi qua.Những dây thanh âm rung tạo ra dạng sóng của luồng không khí có dạng xấp xỉ tam giác.Chu kì cao độ âm thanh của đàn ông trưởng thành thường từ 50Hz đến 250Hz, giá trị trung bình khoảng 120Hz.Đối với phụ nữ trưởng thành, giới hạn trên cao hơn nhiều, có thể lên đến 500Hz.

- Âm vô thanh: được tạo ra khi dây thannh âm không rung.Có hai loại âm vô thanh cơ bản: âm xát và âm hơi.Đối với âm xát như khi ta nói chữ ‘s’, một số điểm trên bộ phận phát âm co lại khi luồng không khí ngang qua nó , hỗn loạn xảy ra tạo nên nhiễu ngẫu nhiên. Đối với âm bật hơi, như khi ta nói chữ ‘h’ , hỗn loạn xảy ra ở gần thanh môn khi dây thannh âm bị giữ nhẹ một phần. Ngoài hai loại âm cơ bản nói trên ,còn có một loại âm trung gian vừa mang tính chất nguyên âm, vừa mang tính chất phụ âm, được gọi là bán nguyên âm hay bán phụ âm. Ví dụ như ‘i’,’u’ trong từ ‘ai ‘ và ‘âu’.

Ø Phụ âm nổ: ví dụ như âm ‘p’, ‘t’,’k’ hay ‘đ’, ‘b’, ‘g’ trong tiếng Việt được tạo ra do loại kích thích khác.

2.5 Mô hình lọc nguồn tạo tiếng nói:

Quá trình tạo tiếng nói là bộ lọc nguồn, trong đó tín hiệu từ nguồn âm thanh (cũng có thể là có chu kì hay nhiễu) được lọc bằng bộ lọc biến thiên theo thời gian có tính chất cộng hưởng tương tự với bộ phận phát âm.Như vậy có thể thu được phổ tần số của tín hiệu tiếng nói bằng cách nhân phổ của nguồn âm thanh với đặc tính tần số của bộ lọc. Hình bên dưới minh họa tiếng nói hữu thanh và vô thanh.Các độ lợi Av và A_N xác định cường độ của nguồn tạo âm hữu thanh và vô thanh.



Mô hình lọc nguồn cho quá trình tạo tiếng nói khá đơn giản nhưng không thể lọc đượcâm xát bằng cách đỉnh cộng hưởng của bộ phận phát âm như âm hữu thanh hay âm bật hơi,vì vậy mô hình lọc nguồn hoàn toàn không chính xác cho âm xát.

2.6 Hệ thống nghe của người:

Quá trình nghe của người như sau: sóng áp suất âm thanh tác động đến tai người, sóng này được chuyển thành chuỗi xung điện, chuỗi nay được truyền tới não bộ thông qua hệ thần kinh,ở não chuỗi được xử lý và giải mã.

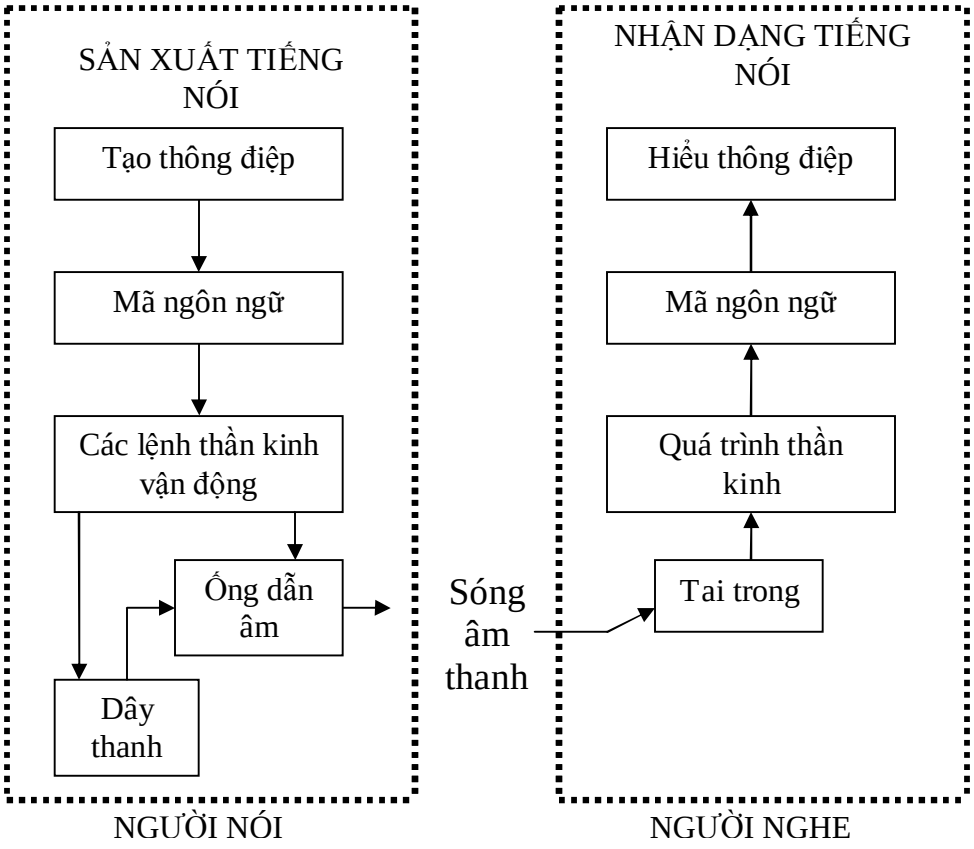
Khi nghe một sóng âm thuần túy tức âm đơn (sóng sine),những điểm khác nhau trên màng đáy sẽ rung động theo tần số của âm đơn đi vào tai.Điểm lệch lớn nhất trên màng đáy phụ thuộc vào tần số của âm đơn. Tần số cao tạo ra điểm lệch lớn nhất ở phía đáy và tần số thấp tạo ra điểm lệch lớn nhất ở phía đỉnh. Như vậy màng đáy làm nhiệm vụ phân tích tần số tín hiệu vào phức tạp thành những tần số khác nhau ở những điểm khác nhau dọc theo chiều dài của nó. Như vậy có thể xem mọi điểm là bộ lọc thông dải và có tần số trung tâm và băng thông xác định.

Ngưỡng nghe của một âm đơn tăng lên khi có sự hiện diện của những âm đơn lân cận khác (âm mặt nạ) và chỉ có bằng tần hẹp xung quanh âm đơn mới tham gia vào hiệu ứng mặt nạ, băng tần này thường gọi là âm tần tới hạn. Giá trị của băng tần tới hạn phụ thuộc vào tần số của âm đơn cần thử.

Tóm lại **quá trình nghe của hệ thính giác là một dãy các bộ lọc băng thông**, có đáp ứng phủ lấp lên nhau và 'băng thông hiệu quả' của chúng xấp xỉ với các giá trị của băng tần tới hạn.

2.7 Quá trình sản xuất tiếng nói và thu nhận tiếng nói của con người:

Sơ đồ biểu diễn quá trình thu nhận tiếng nói của con người



Quá trình sản xuất tiếng nói bắt đầu khi người nói muốn chuyển tải thông điệp của mình cho người nghe thông qua tiếng nói.Tổ chức thần kinh sẽ chịu trách nhiệm chuyển đổi thông điệp sang dạng mã ngôn ngữ.Khi một mã ngôn ngữ được chọn lựa,các lệnh thần kinh vận động điều khiển đồng bộ các khâu vận động nhằm phát ra chuỗi âm thanh.Vậy đầu ra cuối cùng của quá trình là một tín hiệu âm học.

Đối với quá trình thu nhận tiếng nói, người nghe xử lý tín hiệu âm thanh thông qua màng tai trong; nó có khả năng cung cấp một phân tích phổ cho tín hiệu tới. Quá trình thần kinh sẽ chuyển đổi tín hiệu phổ thành các tín hiệu hoạt động với thần kinh thính giác ;có thể coi đây là quá trình lấy ra các đặc trưng.Cuối cùng các tín hiệu được chuyển thành mã ngôn ngữ và hiểu được thông điệp.

2.8 Các âm thanh tiếng nói và các đặc trưng:

2.8.1 Nguyên âm:

Các nguyên âm có tầm rất quan trọng trong nhận dạng tiếng nói; hầu hết các hệ thống nhận dạng dựa trên cơ sở nhận dạng nguyên âm đều có tính năng tốt. Các nguyên âm nói chung là có thời gian tồn tại dài (so với các phụ âm) và dễ xác định phổ.Chính vì thế dễ dàng cho việc nhận dạng tiếng nói, cả đối với con người và máy móc.

Về mặt lý thuyết, các cực đại của biểu diễn phổ của tín hiệu nguyên âm chính là các tần số cộng hưởng(formants) tạo nên nguyên âm. Giá trị của các formant đầu tiên (2 hoặc 3 formant đầu tiên) là yếu tố quyết định cho phép chúng ta nhận dạng được nguyên âm. Do nhiều yếu tố biến thiên như sự khác nhau về giới tính, về độ tuổi, tình trạng tinh thần của người nói và nhiều yếu tố ngoại cảnh khác, đối với một nguyên âm xác địnnh các giá trị formant cũng có sự biến thiên nhất định.Tuy nhiên sự khác biệt về các giá trị các formant giữa các nguyên âm khác nhau lớn hơn nhiều; và trong không gian formant chúng ta có thể xác định một cách tương đối các vùng riêng biệt cho từng nguyên âm.

2.8.2 Các âm vị khác:

Nguyên âm đôi thì có sự biến thiên một cách liên tục các formant của biểu diễn phổ theo thời gian. Đối với âm vị loại này,cần phải đặc biệt chú ý đến việc phân đoạn theo thời gian khi nhận dạng.

Các bán nguyên âm như /l/, /r/ và /y/ là tương đối khó trong việc biểu diễn đặc trưng. Các âm thanh này không được coi là nguyên âm nhưng gọi là bán nguyên âm do bản chất tựa nguyên âm của chúng. Các đặc trưng âm học của các âm thanh này chịu ảnh hưởng rất mạnh của ngữ cảnh mà trong đó chúng xuất hiện.

Đối với các âm mũi thì miệng đóng vai trò như một khoảng cộng hưởng có tác dụng bẫy năng lượng âm tại một vài tần số tự nhiên. Các tần số cộng hưởng này của khoang miệng xuất

hiện như các phản cộng hưởng, hay các điểm không của hàm truyền đạt. Ngoài ra, các phụ âm mũi còn được đặc trưng bởi những sự cộng hưởng mạnh hơn về phổ so với các nguyên âm.

Các phụ âm xác vô thanh như /s/, /sh/. Hệ thống tạo ra các phụ âm xác vô thanh bao gồm một nguồn nhiễu tại một điểm thắt mà chia ống dẫn âm thành hai khoang. Âm thanhđược bức xạ tại khoang trước. Khoang sau có tác dụng bẫy năng lượng như trong trường hợp phụ âm mũi, và như vậy là đưa các phản cộng hưởng vào âm thanh đầu ra. Bản chất không tuần hoàn là đặc trưng cơ bản nhất của nguồn kích thích xác vô thanh.

Điểm khác biệt của các âm xác hữu thanh như /v/, /th/ so với các phụ âm xác vô thanh là ở chỗ có hai nguồn kích thích liên quan tới việc tạo ra chúng. Như vậy đặc trưng của phụ âm xác hữu thanh là bao gồm cả hai thành phần kích thích tuần hoàn và nhiễu.

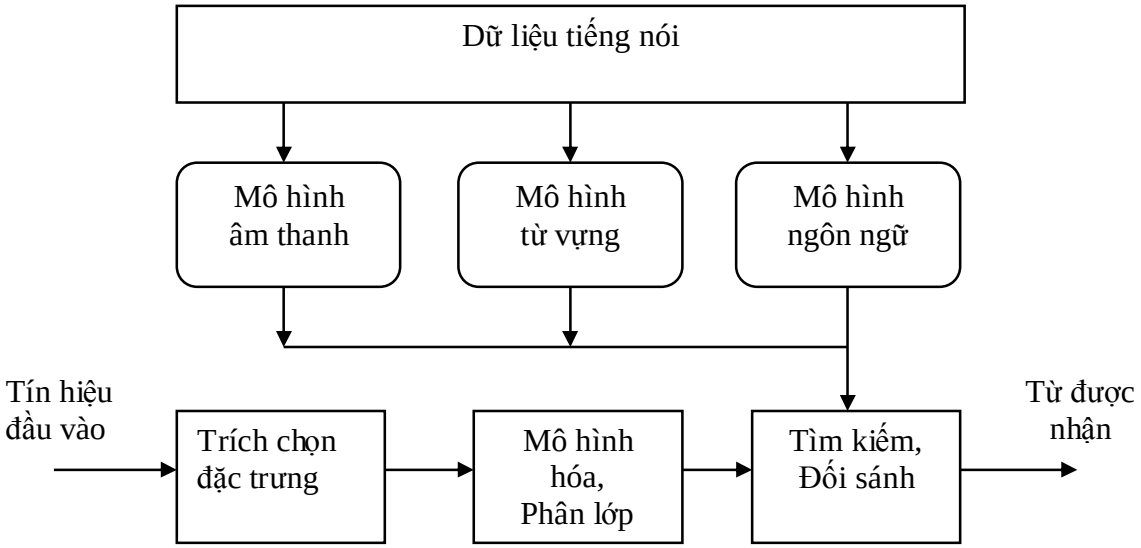
Các âm dừng là các phụ âm /b/, /d/, /g/, /p/, /t/ và /k/ chúng có thời gian tồn tại rất ngắn.Các âm dừng có tính chất động vì thế các thuộc tính của chúng chịu ảnh hưởng rất nhiều bởi nguyên âmđi sau nó.

- hết Chương 2 -

CHƯƠNG 3: LÝ THUYẾT NHẬN DẠNG TIẾNG NÓI

3.1 Tổng quan về nhận dạng tiếng nói

Nhận dạng tiếng nói là một hệ thống tạo khả năng để máy nhận biết ngữ nghĩa của lời nói. Về bản chất, đây là quá trình biến đổi tín hiệu âm thanh thu được của người nói qua Micro, đường dây điện thoại hoặc các thiết bị khác thành một chuỗi các từ. Kết quả của quá trình nhận dạng có thể được ứng dụng trong điều khiển thiết bị, nhập dữ liệu, soạn thảo văn bản bằng lời, quay số điện thoại tự động hoặc đưa tới một quá trình xử lý ngôn ngữ ở mức cao hơn.



Hình 3.1.1: Các phần tử cơ bản của một hệ thống nhận dạng tiếng nói

Các hệ thống nhận dạng tiếng nói có thể được phân loại như sau:

- Nhận dạng từ phát âm rời rạc/liên tục;
- Nhận dạng tiếng nói phụ thuộc người nói/không phụ thuộc người nói;
- Hệ thống nhận dạng từ điển cớ nhỏ (dưới 20 từ)/từ điển cỡ lớn (hàng nghìn từ);
- Nhận dạng tiếng nói trong môi trường có nhiễu thấp/cao;
- Nhận dạng người nói.

Trong hệ nhận dạng tiếng nói với cách phát âm rời rạc có khoảng lặng giữa các từ trong câu. Trong hệ nhận dạng tiếng nói liên tục không đòi hỏi điều này. Tùy thuộc vào quy mô và phương pháp nhận dạng, ta có các mô hình nhận dạng tiếng nói khác nhau. Hình 3.1.1 là mô hình tổng quát của một hệ nhận dạng tiếng nói điển hình .

Tín hiệu tiếng nói sau khi thu nhận được lượng tử hóa sẽ biến đổi thành một tập các vector tham số đặc trưng với các phân đoạn có độ dài trong khoảng 10-30 ms. Các

đặc trưng này được dùng cho đối sánh hoặc tìm kiếm các từ gần nhất với một số ràng buộc về âm học, từ vựng và ngữ pháp. Cơ sở dữ liệu tiếng nói được sử dụng trong quá trình huấn luyện (mô hình hóa/phân lớp) để xác định các tham số hệ thống.

3.2 Các nguyên tắc cơ bản trong nhận dạng tiếng nói

Các nghiên cứu về nhận dạng tiếng nói dựa trên ba nguyên tắc cơ bản:

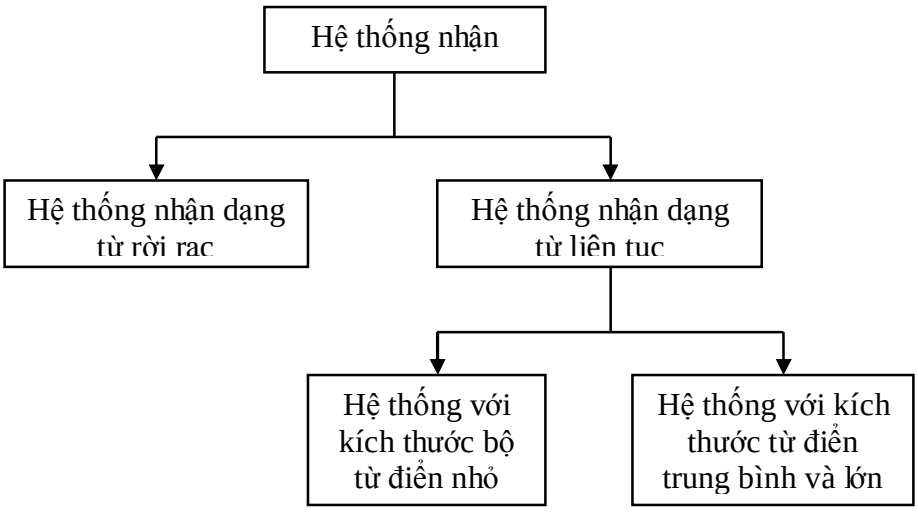
Ø Tín hiệu tiếng nói được biểu diễn chính xác bởi các giá trị phổ trong một khung thời gian ngắn. Nhờ vậy ta có thể trích ra đặc điểm tiếng nói từ những khoảng thời gian ngắn và dùng các đặc điểm này làm dữ liệu nhận dạng tiếng nói.

Ø Nội dung của tiếng nói được biểu diễn dưới dạng chữ viết, là một dãy các kí hiệu ngữ âm. Do đó ý nghĩa của một phát âm được bảo toàn khi chúng ta phiên âm phát âm thành dãy các kí hiệu ngữ âm.

Ø Nhận dạng tiếng nói là một quá trình nhận thức. Ngôn ngữ nói là có nghĩa, do đó thông tin về ngữ nghĩa và suy đoan có giá trị trong quá trình nhận dạng tiếng nói nhất là khi thông tin về âm học là không rõ ràng.

3.3 Các hệ thống nhận dạng tiếng nói:

Các hệ thống nhận dạng tiếng nói có thể được phân chia thành hai loại khác nhau: hệ thống nhận dạng từ rời rạc và hệ thống nhận dạng từ liên tục. Trong hệ thống nhận dạng tiếng nói liên tục, người ta lại phân biệt hệ thống nhận dạng có kích thước từ điển nhỏ và hệ thống nhận dạng với kích thước từ điển trung bình hoặc lớn. Hình 3.3.1 cho ta các lớp hệ thống nhận dạng tiếng nói khác nhau.

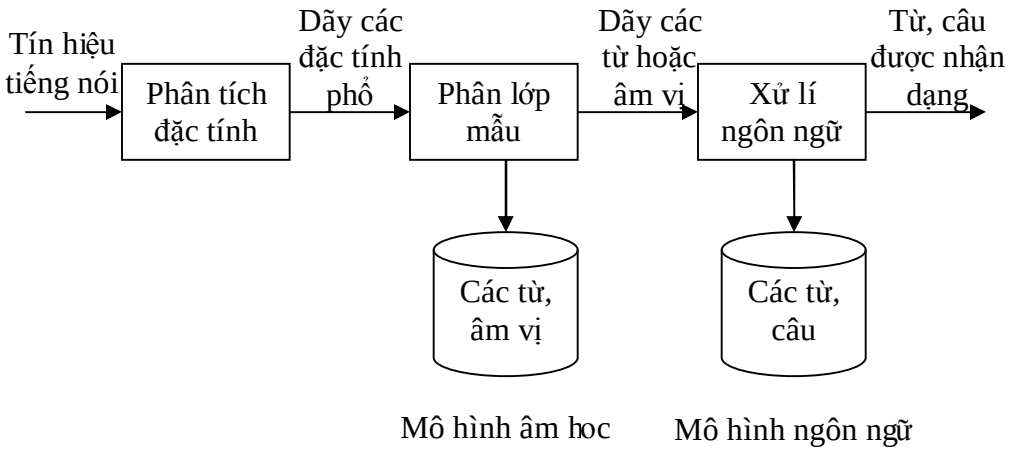


Hình 3.3.1: Các hệ thống nhận dạng tiếng nói

Trong hệ thống nhận dạng từ rời rạc, các phát âm được nhận dạng được giả thiết là chỉ bao gồm một từ hoặc một nhóm từ độc lập. Các từ được nhận dạng mà không phụ thuộc vào ngữ cảnh của nó. Nhận dạng tiếng nói với các từ rời rạc được ứng dụng trong các chương trình dạng câu lệnh-điều khiển (command-control), chẳng hạn như ứng dụng quay số bằng giọng nói trong điện thoại di động. Bài toán nhận dạng tiếng nói các từ rời rạc rõ ràng là dễ hơn rất nhiều so với bài toán nhận dạng tiếng nói liên tục vì ranh giới trái và phải của các từ được coi mặc nhiên là đã được xác định. Tuy nhiên trong thực tế việc tìm ranh giới các từ trong một phát âm liên tục không phải lúc nào cũng là dễ dàng.

3.4 Các quá trình nhận dạng tiếng nói:

Hình 3.4.1 sau đây cho ta thấy các bước cơ bản của một hệ thống nhận dạng tiếng nói, gồm có ba giai đoạn: phân tích đặc tính, phân lớp mẫu và xử lý ngôn ngữ.



Hình 3.4.1: Các quá trình nhận dạng tiếng nói

3.4.1 Phân tích các đặc trưng (tham số) tiếng nói

Quá trình này loại bỏ những thông tin không quan trọng như tiếng ồn của môi trường, nhiễu trên đường truyền, các đặc điểm riêng biệt của người nói...Tiếng nói được phân tích theo các khung thời gian gọi là frame. Kết quả ra của giai đoan này là các vector đặc tính của mỗi khung tín hiệu tiếng nói.

Có 2 cách thông dụng hiện nay thường được áp dụng để phân tích tín hiệu tiếng nói đó là phương pháp **mô phỏng lại quá trình cảm nhận âm thanh của tai người** và phương pháp **mô phỏng lại quá trình tạo âm của cơ quan phát âm**. Cả hai cách này đều đang được áp dụng thành công trong các hệ thống nhận dạng. Tuy nhiên các phương pháp phân tích tiếng nói hiện nay mới chỉ thực hiện được công việc nhỏ so với hệ thống phát âm và nhận thức âm thanh của con người. Sự cải tiến của các phương pháp này sẽ dẫn tới nâng cao năng lực nhận dạng của các hệ thống nhận dạng tiếng nói.

Hai phương pháp trích chọn đặc trưng tiếng nói đang được sử dụng rộng rãi hiện nay trong các hệ thống nhận dạng hiện nay: **MFCC** (melscale frequency cepstral coefficients) và **PLP** (Perceptual Linear Prediction).

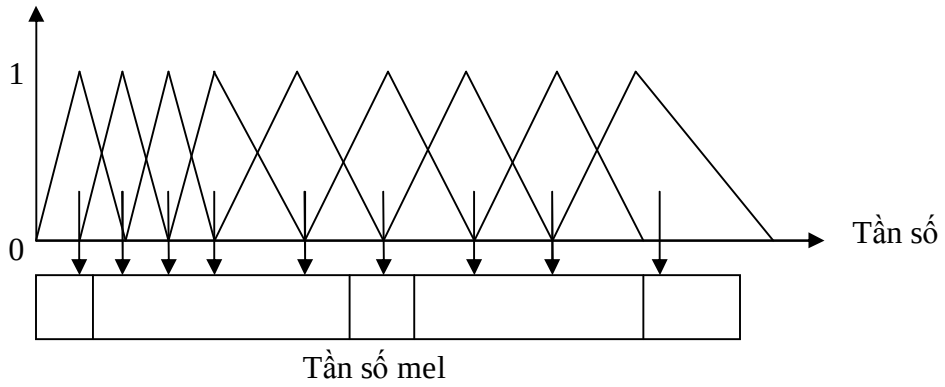
Phân tích cepstral theo thang đo mel MFCC

Phương pháp được xây dựng dựa trên sự cảm nhận của tai người đối với các dải tần số khác nhau. Với các tần số thấp (dưới 1000 Hz), độ cảm nhận của tai người là tuyến tính. Đối với các tần số cao, độ biến thiên tuân theo hàm logarit. Các băng lọc tuyến tính ở tần số thấp và biến thiên theo hàm logarit ở tần số cao được sử dụng để trích chọn các đặc trưng âm học quan trọng của tiếng nói.

Người ta chọn tấn số 1kHz, 40 dB trên ngưỡng nghe là 1000 Mel. Công thức gần đúng biểu diễn quan hệ tần số ở thang mel và thang tuyến tính như sau:

$$mel(f) = 2595*\log_{10}(1+f/700)$$

Một phương pháp để chuyển đổi sang thang mel là sử dụng băng lọc (Hình 3.4.2), trong đó mỗi bộ lọc có đáp ứng tần số dạng tam giác. Số băng lọc sử dụng thường trên 20 băng. Thông thường, người ta chọn tần số từ 0 đến Fs/2 (Fs là tần số lấy mẫu tiếng nói). Nhưng cũng có thể một dải tần giới hạn từ LOFREQ đến HIFREQ sẽ được dùng để lọc đi các tần số không cần thiết cho xử lý. Chẳng hạn, trong xử lý tiếng nói qua đường điện thoại có thể lấy giới hạn dải tần từ LOFREQ=300 đến HIFREQ=3400.



Hình 3.4.2: Các băng lọc tam giác theo thang tần số Mel

Phương pháp mã dự đoán tuyến tính LPC

Mô hình LPC được sử dụng để trích lọc các tham số đặc trưng của tín hiệu tiếng nói. Kết quả của quá trình phân tích tín hiệu thu được một chuỗi gồm các khung tiếng nói. Các khung này được biến đổi nhằm sử dụng cho việc phân tích âm học.

Nội dung phân tích dự báo tuyến tính là: một mẫu tiếng nói được xấp xỉ bởi tổ hợp tuyến tính của các mẫu trước đó. Thông qua việc tối thiểu hóa tổng bình phương sai số giữa các mẫu hiện tại với các mẫu dự đoán có thể xác định được một tập duy nhất các hệ số dự báo. Các hệ số dự báo này là các trọng số được sử dụng trong tổ hợp tuyến tính.

Với dãy tín hiệu tiếng nói $s(n)$, giá trị dự báo được xác định bởi:

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k)$$

trong đó α_k : là các hệ số đặc trưng cho hệ thống.

Hàm sai số dự báo được tính theo công thức:

$$e(n) = s(n) - \tilde{s}(n)$$

Để cực tiểu hóa lỗi cần tìm tập giá trị { α_k } phù hợp nhất.

Phương pháp PLP

Phương pháp này là sự kết hợp của hai phương pháp đã trình bày ở trên

3.4.2 Phân lớp mẫu:

Ở bước này, hệ thống sẽ gán dãy các vector đặc tính thành dãy các tối ưu đơn vị tiếng nói cơ bản. Có bốn phương pháp hay được áp dụng đó là: đối sánh mẫu, rule-based, mô hình Markov ẩn, mạng Neuron

Nguyên tắc cơ bản của **đối sánh mẫu** đó là cất giữ một số lượng các mẫu tiếng nói, bao gồm các vector đặc tính.Tín hiệu tiếng nói cần nhận dạng được phân tích và các vector đặc tính của chúng sẽ được so sánh với các mẫu đã được cất giữ trước đó. Do tốc độ phát âm là rất khác nhau, kỹ thuật DWT (Dynamic Time Warping)được áp dụng để dãn hoặc co hẹp thời gian trên trục thời gian nhằm giảm sự khác biệt so với các mẫu.

Hệ thống **rule-based** xây dựng một loạt các tiêu chuẩn trên một cây quyết định để xác định xem đơn vị nào của ngôn ngữ nằm trong tín hiệu tiếng nói. Đối với hệ thống nhận dạng tiếng nói lớn, phương pháp này gặp khó khăn trong tổng quát hóa sự đa dạng của tín hiệu tiếng nói. Một vấn đề nữa là với cây quyết định rất khó phục hồi lỗi nếu như một quyết định sai được xác định ngay từ khi bắt đầu phân tích.

Mô hình **Markvo ẩn** được nghiên cứu rộng rãi gần đây như là một công cụ mạnh được áp dụng thành công trong nhận dạng tiếng nói. Đa số các hệ thống nhận dạng tiếng nói đều dùng mô hình Markov ẩn. Chi tiết về mô hình Markov ẩn sẽ được trình bày trong mục ***3.6.2.***

Mạng neuron được áp dụng trong nhận dạng tiếng nói từ những năm 1980 với mong muốn sử dụng khả năng phân lớp mạnh của mạng. Mạng neuron truyền thẳng đa lớp perceptron thường được sử dụng trong nhận dạng tiếng nói. Tuy nhiên mạng neuron có hạn chế về khả năng mô hình hoá sự biến thiên của tiếng nói theo thời gian. Mô hình **mạng Neuron** sẽ được trình bày trong **chương 4**.

3.4.3 Xử lý ngôn ngữ:

Mục đích của mô hình này là tìm ra xác suất của từ trong phát âm theo sau các từ. Một phương pháp đơn giản hay được áp dụng đó là dùng N-gram, với giả thiết rằng từ chỉ phụ thuộc vào n-1 các từ đứng trước nó.

Mô hình ngôn ngữ N-gram cùng một lúc chứa đựng các thông tin về cú pháp, ngữ nghĩa, suy đoán và chúng tập trung vào sự phụ thuộc lân cận của một từ. Các xác suất của mô hình ngôn ngữ có thể được tính toán trực tiếp từ cơ sở dữ liệu mà không cần đến các luật ngôn ngữ như ngữ pháp hình thức của ngôn ngữ.

Về mặt nguyên tắc các xác suất của mô hình ngôn ngữ có thể được tính toán trực tiếp từ số lần xuất hiện của các từ trong cơ sở dữ liệu.

Tuy nhiên vấn đề khó khăn cơ bản của mô hình ngôn ngữ là số lượng các bộ ba là quá lớn. Do đó sẽ có nhiều bộ ba không xuất hiện hoặc xuất hiện rất ít chỉ một hoặc hai lần trong cơ sở dữ liệu.

Mặc dù có khó khăn về tính toán, mô hình ngôn ngữ vẫn chứng minh được là chúng đóng vai trò quan trọng trong các hệ thống nhận dạng. Trong các hệ thống nhận dạng với kích thước lớn hiện nay.

3.5 Các tiếp cận nhận dạng tiếng nói

Về cơ bản có ba tiếp cận nhận dạng tiếng nói chính như sau:

1. Tiếp cận âm thanh-ngữ âm.
2. Tiếp cận nhận dạng mẫu.
3. Tiếp cận trí tuệ nhân tạo.

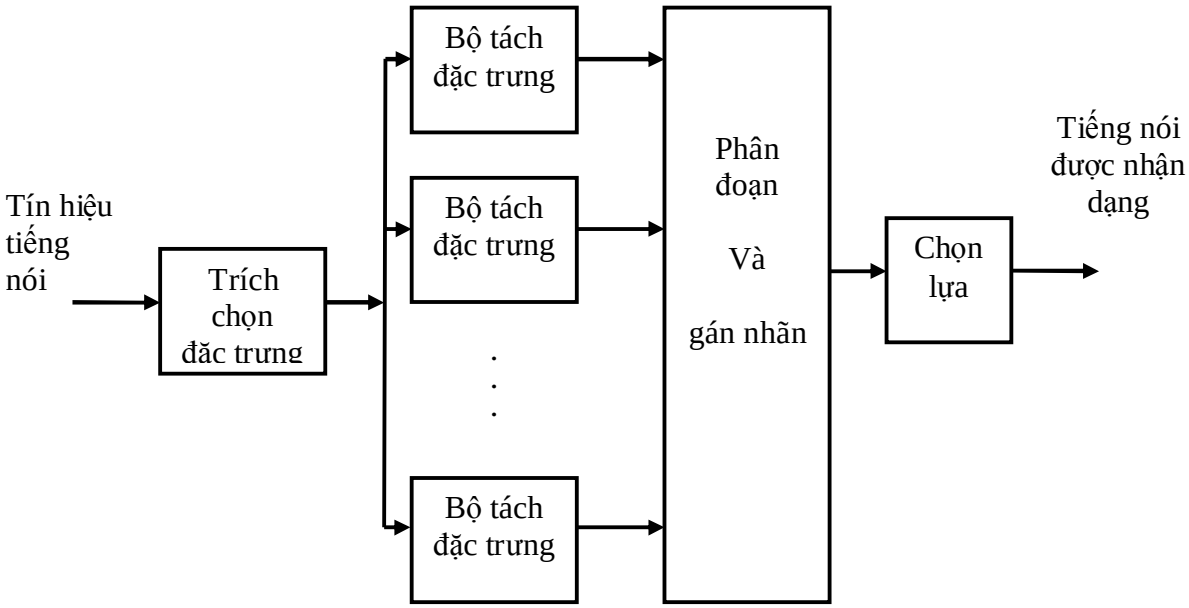
3.5.1 Tiếp cận âm thanh-ngữ âm

Phương pháp này dựa trên lý thuyết về Âm học-Ngữ âm học. Lý thuyết đó cho biết có sự tồn tại của các đơn vị ngữ âm trong ngôn ngữ tiếng nói; các đơn vị ngữ âm này được biểu diễn đặc trưng bởi một tập hợp những thuộc tính thể hiện trong tín hiệu âm thanh hay biểu diễn phổ theo thời gian. Cách tiếp cận này có 2 bước:

- ***Bước 1***: phân đoạn và gán nhãn. Gán một hoặc nhiều nhãn ngữ âm cho mỗi vùng phân đoạn dựa theo các thuộc tính âm học.

- ***Bước 2***: nhận dạng tiếng nói. Cố gắng xác định một từ hợp lệ (hay chuỗi từ hợp lệ) từ một chuỗi các nhãn ngữ âm thu được tử bước 1 dựa trên cơ sở các ràng buộc (về từ vững và cú pháp) của tác vụ cần nhận dạng tiếng nói.

Sơ đồ khối của phương pháp này được biểu diễn ở Hình 3.5.1



Hình 3.5.1: Sơ đồ khối nhận dạng tiếng nói theo Âm học-Ngữ âm học

Nguyên lý hoạt động của phương pháp có thể mô tả như sau:

Trích chọn đặc trưng: Tín hiệu tiếng sau khi số hóa được đưa tới khối trích chọn đặc trưng nhằm xác định các phổ tín hiệu. Các kỹ thuật trích chọn đặc trưng tiếng nói phổ biến là sử dụng băng lọc (filter bank), mã hóa dự đoán tuyến tính (LPC)…

Tách tín hiệu tiếng nói: nhằm biến đổi phổ tín hiệu thành một tập các đặc tính mô tả các tính chất âm học của các đơn vị ngữ âm khác nhau. Các đặc tính đó có thể là: tính chất các âm mũi, âm xát; vị trí các formant; âm hữu thanh, vô thanh; tỷ số mức năng lượng tín hiệu…

Phân đoạn và gán nhãn: Ở bước này hệ thống nhận dạng tiếng xác định các vùng âm thanh ổn định (vùng có đặc tính thay đổi rất ít) và gán cho mỗi vùng này một nhãn phù hợp với đặc tính của đơn vị ngữ âm. Đây là bước quan trọng của hệ nhận dạng tiếng nói theo khuynh hướng Âm học-Ngữ âm học và là bước khó đảm bảo độ tin cậy nhất.

Nhận dạng: Chọn lựa để kết hợp chính xác các khối ngữ âm tạo thành các từ nhận dạng.

Đặc điểm của phương pháp nhận dạng tiếng nói theo hướng tiếp cận Âm học-Ngữ âm học:

• Người thiết kế phải có kiến thức khá sâu rộng về Âm học-Ngữ âm học.
• Phân tích các khối ngữ âm mang tính trực giác, thiếu chính xác.
• Phân loại tiếng nói theo các khối ngữ âm thường không tối ưu do khó sử dụng các công cụ toán học để phân tích.

3.5.2 Tiếp cận nhận dạng mẫu

Về cơ bản đây là một quan điểm sử dụng trực tiếp các mẫu tiếng nói (chính là đoạn tiếng nói cần nhận dạng) mà không cần xác định thật rõ các đặc trưng và cũng không cần phân đoạn tín hiệu.Phương pháp này cũng có 2 bước:

- ***Bước 1***: tích lũy các mẫu tiếng nói: Sử dụng tập mẫu tiếng nói (cơ sở dữ liệu mẫu tiếng nói) để đào tạo các mẫu tiếng nói đặc trưng (mẫu tham chiếu) hoặc các tham số hệ thống.

- ***Bước 2***: nhận dạng mẫu: đối sánh mẫu tiếng nói từ ngoài với các mẫu đặc trưng để ra quyết định.

Trong phương pháp này, nếu cơ sở dữ liệu tiếng nói cho đào tạo có đủ các phiên bản mẫu cần nhận dạng thì quá trình đào tạo có thể xác định chính xác các đặc tính âm học của mẫu (các mẫu ở đây có thể là âm vị, từ, cụm từ…). Hiện nay, một số kỹ thuật nhận dạng mẫu được áp dụng thành công trong nhận dạng tiếng nói là lượng tử hóa vector, so sánh thời gian động (DTW), mô hình Markov ẩn (HMM), **mạng nơron nhân tạo (ANN).** Hệ thống bao gồm các hoạt động sau:

Trích chọn đặc trưng: Tín hiệu tiếng nói được phân tích thành chuỗi các số đo để xác định mẫu nhận dạng. Các số đo đặc trưng là kết quả xử lý của các kỹ thuật phân tích phổ như: lọc thông dải, phân tích mã hóa dự đoán tuyến tính (LPC), biến đổi Fourier rời rạc (DFT).

Huấn luyện mẫu: Nhiều mẫu tiếng nói ứng với các đơn vị âm thanh cùng loại dùng để đào tạo các mẫu hoặc các mô hình đại diện, được gọi là mẫu tham chiếu hay mẫu chuẩn.

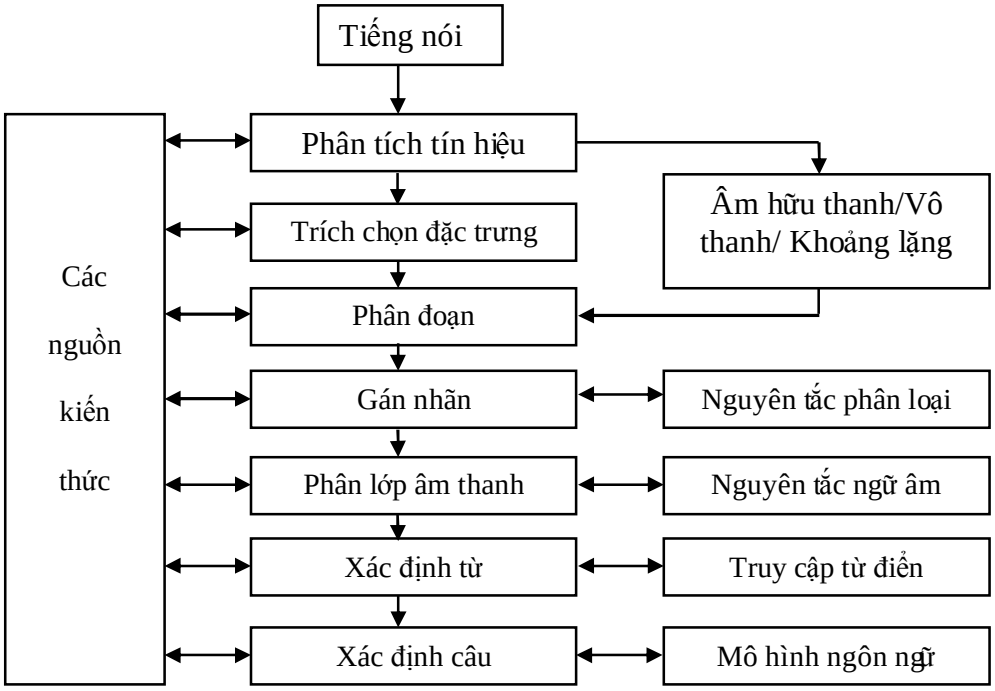
Nhận dạng: Các mẫu tiếng nói được đưa tới khối phân loại mẫu. Khối này đối sánh mẫu đầu vào với các mẫu tham chiếu. Kối nhận dạng căn cứ vào các tiêu chuẩn đánh giá để quyết định mẫu tham chiếu nào giống mẫu đầu vào.

Tiếp cận nhận dạng mẫu thường được lựa chọn cho các ứng dụng nhận dạng tiếng nói bởi các lý do sau:

- Tính dễ sử dụng và dễ hiểu trong thuật toán.
- Tính bất biến và khả năng thích nghi đối với những từ vững, người sử dụng, các tập hợp đặc trưng, các thuật toán so sánh mẫu và các quy tắc quyết định khác nhau.
- Khẳng định tính năng cao trong thực tế.

3.5.3 Tiếp cận trí tuệ nhân tạo:

Tiếp cận trí tuệ nhân tạo là tiếp cận cố gắng "máy móc hóa" chức năng nhận dạng theo cách mà con người áp dụng trí thông minh của mình trong việc quan sát, phân tích và thực hiện những quyết định trên các đặc trưng âm học của tín hiệu. Phương pháp ứng dụng trí tuệ nhân tạo kết hợp các phương pháp trên nhằm tận dụng tối đa các ưu điểm của chúng. Sơ đồ khối của phương pháp trí tuệ nhân tạo theo mô hình từ dưới lên (bottom-up) (Hình 3.5.3).



Hình 3.5.3: Sơ đồ khối hệ nhận dạng tiếng nói theo phương pháp từ dưới lên

Đặc điểm của các hệ thống nhận dạng theo phương pháp này là:

Sử dụng hệ chuyên gia để phân đoạn, gán nhãn ngữ âm. Điều này làm đơn giản hóa hệ thống so với phương pháp nhận dạng ngữ âm.

Sử dụng mạng nơron nhân tạo để học mối quan hệ giữa các ngữ âm, sau đó dùng nó để nhận dạng tiếng nói.

Việc sử dụng hệ chuyên gia nhằm tận dụng kiến thức con người vào hệ nhận dạng:

Kiến thức về âm học: để phân tích phổ và xác định đặc tính âm học của các mẫu tiếng nói.

Kiến thức về từ vựng: sử dụng để kết hợp các khối ngữ âm thành các từ cần nhận dạng.

Kiến thức về cú pháp: nhằm kết hợp các từ thành các câu cần nhận dạng.

Kiến thức về ngữ nghĩa: nhằm xác định tính logic của các câu đã được nhận dạng.

Có nhiều cách khác nhau để tổng hợp các nguồn kiến thức vào bộ nhận dạng tiếng nói.

Phương pháp thông dụng nhất là xử lý "từ dưới lên". Theo cách này, tiến trình xử lý của hệ thống được triển khai tuần tự từ thấp lên cao. Trong Hình 3.5.3, các bước xử lý ở mức thấp (phân tích tín hiệu, tìm đặc tính, phân đoạn, gán nhãn) được triển khai trước khi thực hiện các bước xử lý ở mức cao (phân lớp âm thanh, xác định từ, xác định câu). Mỗi bước xử lý đòi hỏi một hoặc một số nguồn kiến thức nhất định. Ví dụ: bước phân đoạn tiếng nói cần hiểu biết sâu sắc về đặc tính Âm học-Ngữ âm học của các đơn vị ngữ âm; bước xác định từ đòi hỏi kiến thức về từ vựng; bước xác định câu đòi hỏi kiến thức về mô hình ngôn ngữ (nguyên tắc ngữ pháp).

3.6 Các phương pháp nhận dạng tiếng nói

3.6.1 Mô hình Fujisaki:

Mô hình Fujisaki:

Fujisaki là một mô hình định lượng dùng để mô hình hóa ngữ điệu. Mô hình Fujisaki hướng vào việc mô hình hóa quá trình sinh ra tần số cơ bản F0, giải thích về mặt vật lý học, sinh lý học quá trình sinh ra F0 và các tính chất của quá trình đó. Mô hình được áp dụng chủ yếu trong ứng dụng tổng hợp nhằm xây dựng phần ngữ điệu trong tiếng nói tổng hợp.

Mô hình sinh ra F0 theo 3 công thức sau:

$$\ln F0(t) = \ln Fb + \sum_{i=1}^{I} Ap_i Gp(t - T_{0i}) + \sum_{j=1}^{J} Aa_j \left[Ga(t - T_{1j}) - Ga(t - T_{2j}) \right] \qquad (3.6.1.1)$$

$$Gp(t) = \begin{cases} a^2 t \exp(-a t), & t \geq 0 \\ 0, & t < 0 \end{cases} \qquad (3.6.1.2)$$

$$Gp(t) = \begin{cases} \min[1 - (1 + b t)\exp(-b t), g] & t \geq 0 \\ 0, & t < 0 \end{cases} \qquad (3.6.1.3)$$

Các tham số của mô hình gồm có:

Các hằng số: *Fb* là giá trị khởi đầu của đường tần số cơ bản. *Fb* là giá trị phụ thuộc vào người nói chứ không phụ thuộc vào các mẫu tiếng nói. Giá trị α là tần số góc tự nhiên của lệnh ngữ. Giá trị β là tần số góc tự nhiên của lệnh trọng âm. Giá trị *g* là mức giá trị trần tương ứng với các thành phần trọng âm.

Các đối số: *I* là số lệnh ngữ. *J* là số lệnh trọng âm. Ap_i là cường độ của lệnh ngữ thứ *i*. Aa_j là biên độ của trọng âm thứ *j*. T_{0i} là thời điểm bắt đầu lệnh ngữ thứ *i*. T_{1j} và T_{2j} là thời điểm bắt đầu và kết thúc thanh điệu ở lệnh trọng âm thứ *j*.

Trong mô hình, đường *F0* được xét ở miền *logF0*, mục đích của phép biến đổi này là làm cho giọng nói của nam và nữ giống nhau. Theo (3.6.1.1) các giá trị α*=2.0/s* và β*=20.0/s*, trong một số trường hợp đặc biệt α*=3.0/s*. Tuy nhiên theo quan sát thì α nằm trong khoảng [1.0;3.0], còn β thuộc khỏang [19.5;20.5].

Các tham số $Ap,\alpha,\beta,A_a,T1,T2,Fb$ được gọi là các tham số Fujisaki và phương pháp phân tích bằng tổng hợp bằng đường nét F0 sử dụng mô hình Fujisaki được gọi là phân tích Fujisaki. Các tham số của mô hình có thể được sinh ra tự động bởi nhiều cách khác nhau tùy vào từng ngôn ngữ được phân tích.

Phân tích thanh điệu tiếng Việt bằng mô hình Fujisaki:

Cơ sở dữ liệu: để phân tích đường nét *F0* của thanh điệu tiếng Việt và sự liên cấu âm giữa các thanh điệu liền kề, một tập gồm 72 câu nói, mỗi câu nói gồm 6 âm tiết được xây dựng từ câu gốc " nha mai lắm nhan nhiêu ngô", mỗi âm tiết trong câu gốc sẽ mang các thanh điệu khác nhau để thể hiện nhiều tổ hợp thanh điệu liền kề như:

1) "*Nhà mai lắm nhãn nhiều ngô*"
2) "*Nhà mài lắm nhan nhiêu ngộ*"
3) "*Nha mải lắm nhãn nhiều ngỗ*"
.....

Các câu được phát âm với giọng chuẩn miền Bắc bởi hai người một nam và một nữ. Để đảm bảo tính tự nhiên của lời nói, hai người nói đều được chuẩn bị trước, các câu nói được phát âm nhiều lần và kiểm tra lại để chọn câu nói tự nhiên nhất

Phương pháp phân tích: để phân tích đường nét F0,phân tích các tham số của mô hình Fujisaki. *Fb* được đặt bằng 96Hz cho giọng nam và 210Hz cho giọng nữ. α và β cho cả giọng nam và nữ được lần lượt đặt bằng 2Hz và 25Hz.

Các bước tiến hành phân tích bao gồm:

1) Tính đường nét *F0*
2) Lựa các chọn lệnh ngữ câu nói.
3) Dựa vào thanh điệu của các âm tiết để lựa chọn các lệnh thanh điệu phù hợp.
4) Điều chỉnh các tham số sao cho đường nét *F0* sinh ra sắp xỉ *F0* thực.
5) Tổng hợp lại câu nói với đường nét thanh điệu mới sử dụng phương pháp PSOLA.
6) Cảm nhận bằng tai câu nói tổng hợp, so sánh với câu nói gốc và điều chỉnh lại.

Kết quả phân tích thanh điệu bằng mô hình Fujisaki:

Phân tích cơ sở dữ liệu cho thấy, các thanh ngang, sắc, ngã được biểu diễn bằng một lệnh thanh điệu dương, thanh huyền và hỏi được biểu diễn bằng một lệnh thanh điệu âm, thanh nặng không cần lệnh thanh điệu.

Thanh điệu	**Biểu diễn bằng lệnh thanh điệu**
Ngang	1 lệnh thanh điệu dương ở trước âm tiết
Sắc	1 lệnh thanh điệu dương
Hỏi	1 lệnh thanh điệu âm
Huyền	1 lệnh thanh điệu âm
Ngã	1 lệnh thanh điệu dương
Nặng	Không dùng lệnh thanh điệu

Các câu được phân tích chỉ sử dụng một lệnh ngữ cho cả câu, phù hợp với hiện tượng trong câu nói, người nói thường lên giọng ở đầu câu và hạ giọng ở cuối câu.Tuy nhiên trong tiếng Việt hiện tượng này không rõ rệt như ở các ngôn ngữ khác nên cường độ của lệnh ngữ này không lớn.

Kết luận:

Mô hình về cơ bản không thể áp dụng cho bài toán nhận dạng tiếng nói được. Lí do chủ yếu là mô hình này thực chất tổng hợp đường F0 một cách tuyến tính. Các kết quả phân tích thanh điệu tiếng Việt chứng tỏ rằng có thể áp dụng mô hình fujisaki vào việc mô hình hóa tiếng Việt.Từ đó nâng cao chất lượng của hệ thống tổng hợp tiếng nói và các kết quả phân tích cũng có thể áp dụng kết quả tính toán ngữ âm học vào nhận dạng tiếng nói.

3.6.2 Mô hình Markvo ẩn

a. Quá trình Markov ẩn:

Ta hãy xem xét sự tiến triển theo thời gian của một hệ thống nào đó (có thể là một hệ vật lý hay hệ sinh thái, ...), ký hiệu q_t là vị trí của hệ tại thời điểm *t*. Các vị trí có thể có được của hệ được gọi là không gian trạng thái, ký hiệu là *S= {S1, S2, S3, ...}*. Giả sử ở thời điểm *s* hệ ở trạng thái *Si* , nếu xác suất để hệ ở trạng thái *Sj* ở thời điểm *t* trong tương lai chỉ phụ thuộc vào *s, t, Si, Sj* thì có nghĩa là sự tiến triển của hệ chỉ phụ thuộc vào hiện tại và độc lập với quá khứ. Ta gọi đó là tính Markov và hệ có tính chất này được gọi là quá trình Markov.

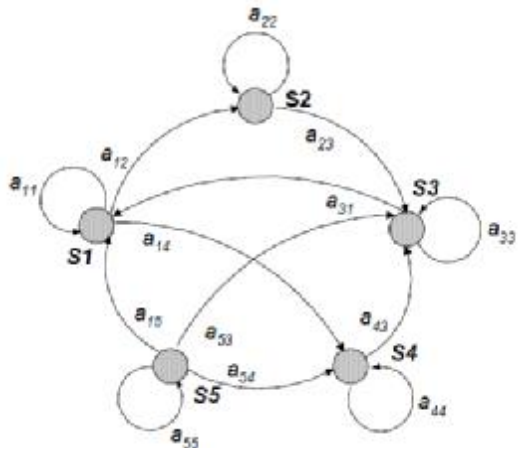
Nếu không gian trạng thái *S* của hệ là đếm được thì ta gọi hệ là xích Markov. Nếu thời gian *t* là rời rạc *t*=0,1,2,... thì ta có xích Markov rời rạc. Ta có thể biểu diễn tính Markov của hệ bằng biểu thức sau :

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, ...) = P(q_t = S_j \mid q_{t-1} = S_i)$$

Đặt $P(s,i,t,j) = P(qt = Sj \mid qs = Si)$ là xác suất để hệ tại thời điểm *s* ở trạng thái *i*, đến thời điểm *t* chuyển sang trạng thái *j*. Ta gọi $P(s,i,t,j)$ là xác suất chuyển của hệ. Nếu xác suất chuyển chỉ phụ thuộc vào $(t-s)$ tức là

$$P(s,i,t,j) = P(s+h,i,t+h,j)$$

thì ta nói hệ là thuần nhất theo thời gian.



Hình 3.6.1 Xích Markov với năm trạng thái S1, S2, ..., S5 và các xác suất chuyển trạng thái.

Tại mỗi thời điểm *t*=0,1,2,... hệ chuyển trạng thái theo xác suất chuyển trạng thái *aij* tương ứng với mỗi trạng thái.

$$\begin{cases} \sum_{j=1}^{N} a_{ij} = 1; i = \overline{1,N} \\ a_{ij} \geq 0; i,j = \overline{1,N} \end{cases}$$

Ngòai ra ta định nghĩa xác suất trạng thái khởi đầu (initial state distribution) $\pi = \{\pi 1, \pi 2, ..., \pi N\}$, trong đó πi là xác suất để trạng thái *i* được chọn tại thời điểm khởi đầu *t*=1.

$$\pi i = P(q1 = Si).$$

$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 \\ \pi_i \geq 0; i = \overline{1,N} \end{cases}$$

Quá trình Markov miêu tả ở trên được gọi là một mô hình Markov quan sát được (observable Markov model). Đầu ra của quá trình là một tập các trạng thái tại các thời điểm rời rạc liên tiếp nhau, trong đó mỗi sự kiện tương ứng với một sự kiện vật lý có thể quan sát được (observation event).

Ví dụ : Ta xét một mô hình Markov ba trạng thái miêu tả thời tiết: *S1, S2, S3*. Trong một ngày thời tiết có thể là một trong ba trạng thái :

S1: mưa
S2: mây
S3: nắng

ma trận xác suất chuyển là

$$A = \{aij\} = \{0.4\ 0.3\ 0.3;\ 0.2\ 0.6\ 0.2;\ 0.1\ 0.1\ 0.8\}$$

Giả thiết là thời tiết tại ngày *t*=1 là nắng. Ta sẽ tìm xác suất để trong 5 ngày liên tiếp có thời tiết như sau : nắng, nắng, mưa, mưa, mây. Tức là ta có một dãy các quan sát (observation) *O= S3, S3, S1, S1, S2*, tương ứng với các thời điểm *t*=1,2,3,4,5

$$\begin{aligned} P(O|\text{Mô hình}) &= P(S3, S3, S1, S1, S2 \mid \text{Mô hình}) \\ &= P(S3).P(S3|S3).P(S1|S3).P(S1|S1).\ P(S2|S1) \\ &= \pi 3.a33.a33.\ .a31.a11.a12 \\ &= 1.\ (0.8).\ (0.8).\ (0.1).(0.4).\ (0.3) \\ &= 768.\ 10^{-4} \end{aligned}$$

b. Mô hình Markov ẩn: (Hidden Markov Model - HMM)

Mô hình Markov mà mỗi một trạng thái tương ứng với một sự kiện quan sát được mở rộng bằng cách các quan sát (observation) tương ứng với các trạng thái là một hàm xác suất của các trạng thái. Mô hình này gọi là mô hình Markov ẩn và đó là một quá trình ngẫu nhiên kép, trong đó có một quá trình ngẫu nhiên không quan sát được. Tập các quan

sát *O* được sinh ra bởi dãy các trạng thái *S1, S2, ..., SN* của mô hình, mà dãy các trạng thái này là không thấy được, đó chính là lý do mô hình được gọi là mô hình Markov ẩn (hidden).

Mô hình Markov ẩn là mô hình thống kê trong đó hệ thống được mô hình hóa được cho là một quá trình Markov với các tham số không biết trước và nhiệm vụ là xác định các tham số ẩn từ các tham số quan sát được, dựa trên sự thừa nhận này. Các tham số của mô hình được rút ra sau đó có thể sử dụng để thực hiện các phân tích kế tiếp, ví dụ cho các ứng dụng nhận dạng mẫu.

Mô hình Markvo ẩn sử dụng kĩ thuật lượng tử hóa vector dùng để lấy trung bình đặc tính của các frame cũng như đánh nhãn các vector.

Mô hình Markvo ẩn được sử dụng rộng rãi trong nhận dạng tiếng nói vì nó có khả năng mô hình hóa thông tin theo thời gian của tín hiệu tiếng nói,trong khi đó mạng nơ-ron đã được chứng minh là một công cụ mạnh mẽ cho việc phân lớp tĩnh do bản thân mang nơ-ron có tính phân biệt một cách tự nhiên.Sự kết hợp giữa mạng nơ-ron với mô hình Markvo ẩn nhằm tăng độ chính xác nhận dạng.

c. Các thành phần của HMM:

Mô hình markvo ẩn gồm một xích Markvo. Mỗi vòng tròn biểu diễn một trạng thái của mô hình và ở thời điểm rời rạc t, tương ứng với một frame tiếng nói, mô hình sẽ ở một trong những trạng thái này và tạo ra một mẫu tiếng nói hay một quan sát. Ở thời điểm t+1 mô hình sẽ di chuyển đến trạng thái mới hay vẫn ở trạng thái cũ và tạo ra một mẫu khác. Lặp lại quá trình này cho đến khi tạo ra toàn bộ các bộ mẫu.

Các thành phần của HMM :

1. N là số trạng thái của mô hình, {1, 2,…,N} là các trạng thái, trạng thái ở thời điểm t là q_t.

2. M là số lượng quan sát phân biệt, ký hiệu tập các quan sát là V ={v_1, v_2,…v_M }. Đối với tiếng nói, M là số lượng vector của code book sau khi lượng tử hóa vector, còn v_i là mã của từng vector.

3. Ma trận xác suất trạng thái vị trí A= { a_{ij} } ở đó a_{ij} là xác suất từ trạng thái i ở thời điểm t đến trạng thái j ở thời điểm t+1

$$a_{ij} = P[q_{t+1} = j | q_t = i] \qquad 1 \le i, j \le N$$

Chú ý rằng $\sum_{j=1}^{N} a_{ij} = 1$ với mọi i,j. Tổng quát từ một trạng thái có thể chuyển đến một trạng thái bất kì, nghĩa là $a_{ij} > 0$ với mọi *i,j*. Tuy nhiên đối với tiếng nói có thể $a_{ij} = 0$ ở cặp i,j nào đó.

4. Ma trận xác suất quan sát B= $\{b_j(k)\}$ ở đó $b_j(k)$ là xác suất tạo ra quan sát v_k khi mô hình đang ở trạng thái j.

$$b_j(k) = P[0_t = v_k | q_t = j], \qquad 1 \le k \le M$$

Chú ý rằng $\sum_{k=1}^{M} b_j(k) = 1$ với mọi j, k.

5. Ma trận xác suất trạng thái ban đầu $\pi = \{\pi_i\}$ ở đó π_i là xác suất mô hình ở trạng thái i tại thời điểm t=0.

$$\pi_i = P[q_t = i], \qquad 1 \le i \le N$$

Chú ý rằng $\sum_{i=1}^{N} p_i = 1$ với mọi j.

Có thể biểu diễn HMM bằng số lượng trạng thái N ,số lượng quan sát M, ba ma trận xác suất A, B, π. Mô hình này được gọi là ẩn vì không thể xác định được các trạng thái tạo ra tương ứng với các quan sát đã cho.Ta kí hiệu HMM là λ = (A, B, π).

d. Đánh giá xác suất:

Muốn tính xác suất của quan sát O = ($o_1, o_2, \ldots, o_T$) tức là tính P(O|λ) ta sử dụng các thuật toán sau:

Thuật toán tiến hay Baum-welch:

Khảo sát biến tiến $\alpha_t(i)$ được định nghĩa như sau:

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i | l)$$

tức là xác suất của miền quan sát $o_1, o_2, \ldots, o_t$ (đến thời điểm t) và trạng thái i ở thời điểm t, ứng với mô hình λ. Ta có thể tính $\alpha_t(i)$ bằng qui nạp như sau:

- ***Bước 1***: Khởi tạo

$$\alpha_t(i) = p_i b_i(o_1) \qquad 1 \le i \le N$$

- ***Bước 2***: Qui nạp

$$a_{t+1}(j) = \left[\sum_{i=1}^{N} a_t(i) a_{ij}\right] b_j(o_{t+1}) \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix}$$

- ***Bước 3***: Kết thúc

$$P(O \mid l) = \sum_{i=1}^{N} a_T(i)$$

Thuật toán lùi:

Tương tự ta định nghĩa biến lùi $\beta_t(i)$ như sau:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid q_t = i, l)$$

tức là xác suất của miền quan sát từ t+1 đến thời điểm T và trạng thái i ở thời điểm t, ứng với mô hình λ.

Ta có thể tính $\beta_t(i)$ bằng qui nạp như sau:

- ***Bước 1***: Khởi tạo:

$$\beta_T(i)=1 \qquad 1 \le i \le N$$

- ***Bước 2***: Qui nạp:

$$b_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) b_{t+1}(j) \qquad \begin{matrix} t = T-1, T-2, \ldots 1 \\ 1 \le i \le N \end{matrix}$$

Thuật toán này chỉ cần N^2T phép tính và dùng cấu trúc lưới.

Thuật toán Viterbi:

Thuật toán Baum-welch không xác định được mô hình đang ở trạng thái nào. Nhằm khắc phục trạng thái "ẩn" này, ta sử dụng thuật toán Viterbi để tìm chuỗi trạng thái đơn tốt nhất $q = (q_1, q_2, \ldots q_T)$ ứng với chuỗi quan sát $O = (o_1, o_2, \ldots, o_T)$ đã cho.Ta cần định nghĩa đại lượng

$$\delta_t(i) = \max P[q_1 q_2 \ldots q_{t-1}, q_t = i, o_1, o_2, \ldots, o_t \mid l]$$

tức là $\delta_t(i)$ có điểm tốt nhất (xác suất lớn nhất) trên con đường đơn, tại thời điểm t ứng với quan sát đã cho và kết thúc ở trạng thái i. Qui nạp ta có:

$$d_{t+1}(j) = [\max_i d_t(i) a_{ij}] b_j(o_{t+1})$$

Muốn xác định chuỗi trạng thái, ta sử dụng mảng $\psi_t(j)$ để lưu lại đối số làm cho phương trình trên cực đại ở từng thời điểm t và trạng thái i.

Thuật toán tìm chuỗi trạng thái tốt nhất được mô tả như sau:

- ***Bước 1***: Khởi tạo:

$$d_1(i) = p_i b_i(o_1) \qquad 1 \le i \le N$$
$$y_1(i) = 0$$

Ø ***Bước 2***: Đệ qui

$$d_t(j) = \max_{1 \le i \le N} \left[d_{t-1}(i) a_{ij} \right] b_j(o_t) \qquad 2 \le t \le T, \quad 1 \le j \le N$$

$$y_t(j) = \arg\max_{1 \le i \le N} \left[d_{t-1}(i) a_{ij} \right] \qquad 2 \le t \le T, \quad 1 \le j \le N$$

Ø ***Bước 3:*** Kết thúc

$$P^* = \max_{1 \le i \le N} \left[d_T(i) \right]$$

$$q_T^* = \arg\max_{1 \le i \le N} \left[d_T(i) \right]$$

Ø ***Bước 4***: Lần ngược con đường (chuỗi trạng thái)

$$q_t^* = y_{t+1}(q_{t+1}^*) \qquad t = t-1, T-2, \ldots, 1$$

e. Ước lượng tham số:

Khó khăn nhất của mô hình Markvo ẩn là tìm ra phương pháp điều chỉnh tham số của mô hình (A, B, π) sao cho thỏa mãn tiêu chuẩn tối ưu nào đó. Không có cách phân tích nào có thể điều chỉnh được tham số của mô hình sao cho đạt được xác suất lớn nhất ứng với quan sát đã cho. Tuy nhiên ta có thể chọn λ = (A, B, π) sao cho xác suất $P(O, l)$ là cực đại địa phương theo phương pháp kì vọng cực đại-expectation maximization (EM).

Ta cần định nghĩa các đại lượng sau:

Ø Biến xác suất hậu nghiệm tức là xác suất ở trạng thái i tại thời điểm t, ứng với quan sát đã cho là O và mô hình λ

$$g_t(i) = P(q_t = i | O, l)$$

$$g_t(i) = \frac{P(O, q_t = i | l)}{P(O | l)}$$

$$g_t(i) = \frac{a_t(i)b_t(i)}{\sum_{i=1}^{N} a_T(i)b_T(i)}$$

Ø Định nghĩa $\xi_t(i,j)$ là xác suất đang ở trạng thái i tại thời điểm t và trạng thái j tại thời điểm t+1, ứng với quan sát đã cho là O và mô hình λ tức là:

$$x_t((i,j) = P(q_t = i, q_{t+1} = j \mid O, l)$$

$$x_t(i,j) = \frac{P((q_t = i, q_{t+1} = j \mid O, l))}{P(O \mid l)}$$

$$= \frac{a_t(i)a_{ij}b_j(o_{t+1})b_{t+1}(j)}{P(O \mid l)}$$

$$= \frac{a_t(i)a_{ij}b_j(o_{t+1})b_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} a_t(i)a_{ij}b_j(o_{t+1})b_{t+1}(j)}$$

Mối liên hệ giữa $\gamma_t(i)$ và $\xi_t(i,j)$ là:

$$g_t(i) = \sum_{j=1}^{N} x_t(i,j).$$

Vậy:

$\sum_{t=1}^{T-1} g_t(i)$ = kỳ vọng số lượng vị trí tại trạng thái i ứng với O.

$\sum_{t=1}^{T-1} x_t(i,j)$ = kỳ vọng số lượng vị trí từ trạng thái i tới trạng thái j ứng với O.

Tập các công thức ước lượng A, B và π như sau:

$\overline{p_i}$ = kỳ vọng tần số (số lần) ở trạng thái i tại thời điểm *t=1*
$= g_1(i)$

$$= \frac{P(O, q_1 = i \mid l)}{P(O \mid l)}$$

$$= \frac{a_1(i)b_1(i)}{\sum_{i=1}^{N} a_T(i)} \qquad (3.6.2.1)$$

$$\bar{a}_{ij} = \frac{\textit{Kỳ vọng số lượng vị trí từ trạng thái i tới trạng thái j}}{\textit{Kỳ vọng số lượng vị trí từ trạng thái i}}$$

$$= \frac{\sum_{i=1}^{T-1} x_t(i,j)}{\sum_{i=1}^{T-1} g_t(i)}$$

$$= \frac{\sum_{t=1}^{T-1} a_t(i)a_{ij}b_j(o_{t+1})b_{t+1}(j)}{\sum_{t=1}^{T-1} a_t(i)b_t(i)} \qquad (3.6.2.2)$$

$$\bar{b}_j(k) = \frac{\textit{Kỳ vọng số lượng ở trạng thái j và quan sát } v_k}{\textit{Kỳ vọng số lượng ở trạng thái j}}$$

$$= \frac{\sum_{t=1}^{T} a_t(j)b_t(j)d(o_t, v_k)}{\sum_{t=1}^{T} a_t(j)b_t(j)} \qquad (3.6.2.3)$$

ở đó ta kí hiệu :

$$d(o_t, v_k) = \begin{cases} 1 & o_t = v_k \\ 0 & \text{Ngược lại} \end{cases}$$

Nếu ta gọi mô hình hiện tại là λ=(A,B,π) rồi dùng mô hình hiện tại này để tính vế phải của các phương trình (3.6.2.1), (3.6.2.2) và (3.6.2.3) ta sẽ có được mô hình $\bar{l} = (\bar{A}, \bar{B}, \bar{p})$Thuật toán này rất phức tạp và có nhiều cực đại địa .Nếu ta gán λ bằng $\bar{l}$ và lặp lại tính toán ước lượng, ta sẽ cải thiện được xác suất của mô hình với quan sát đã cho O, cho tới khi đạt tới trạng thái tới hạn.Kết quả cuối cùng của phương pháp lặp ước lượng là ước lượng ML.Tính chất quan trọng của phương pháp lặp ước lượng là các ràng buộc thống kê cho mô hình markvo ẩn:

$$\sum_{i=1}^{N} \bar{p}_i = 1 \qquad 1 \le i \le N$$

$$\sum_{j=1}^{N} \overline{a_{ij}} = 1 \qquad 1 \le j \le N$$

$$\sum_{k=1}^{M} \overline{b}_j(k) = 1$$

f. Phân loại mô hình Markvo ẩn:

Ta phân loại mô hình markkvo ẩn dựa vào cấu trúc của ma trận vị trí A của xích markvo. Có 2 loại mô hình markvo ẩn:

Ø Mô hình markvo ẩn kết nối đầy đủ, nghĩa là mỗi trạng thái của mô hình có thể đạt tới những trạng thái khác.

Ø Mô hình trái – phải hay mô hình Bakis: mô hình nàyđược sử dụng thông thường trong nhận dạng tiếng nói. Mô hình có tên gọi là trái – phải vì các trạng thái liên kết với mô hình có tính chất là khi thời gian tăng, trạng thái sẽ tăng lên tức là trạng thái tiến dần từ trái sang phải. Điều này phù hợp với cấu trúc tự nhiên của tiếng nói là biến thiên theo thời gian từ trái sang phải.Tính chất cơ bản của mô hình này là các hệ số của ma trận vị trí có tính chất $a_{ij} = 0$ ($j<0$) tức là không cho phép trạng thái sau nhỏ hơn trạng thái hiện tại. Ngoài ra xác suất trạng thái ban đầu có tính chất:

$$p_i = \begin{cases} 0 & i \neq 1 \\ 1 & i = 1 \end{cases}$$

bởi vì trạng thái ban đầu bắt buộc là 1(và kết thúc ở trạng thái N).Và mô hình còn ràng buộc không được chuyển từ trạng thái này đến trạng thái khác quá xa,ràng buộc có dạng:

$$a_{ij} = 0 \qquad j > i + \Delta i$$

Phân loại mô hình Markvo ẩn theo tính chất của hàm phát xạ quan sát, thì có 3 loại mô hình:

Ø *Mô hình HMM rời rạc* : không gian các đặc tính phổ được chia thành một số hữu hạn các vùng bằng phương pháp lượng tử hóa vector VQ.Trọng tâm của mọi vùng được biểu diễn bằng một từ mã mà thực chất là một chỉ số chỉ tới một sách mã.Một khung tín hiệu được biến đổi thành một từ mã bằng cách tìm một vector gần với nó nhất trong sách mã.Nhược điểm của mô hình này là có sai số trong quá trình lượng tử hóa nhất là nếu kích thước của sách mã nhỏ, ngược lại nếu kích thước sách mã lớn thì số lượng tính toán sẽ tăng lên.

Ø *Mô hình HMM liên tục*: khắc phục nhược điểm của mô hình trên.Trong

phương pháp này thì không gian các đặc tính phổ được mô hình hóa bằng các hàm mật độ xác suất ,thông thường là hàm trộn với các hàm Gaussian.Nhược điểm của phương pháp này là mỗi trạng thái đều có các tham số của riêng chúng nên số lượng các tham số là rất lớn và do vậy không thể tránh khỏi các trường hợp không đủ dữ liệu huấn luyện cho các trạng thái.Ngoài ra thời gian tính toán khá lâu.

Ø *Mô hình HMM bán liện tục*: là sự kết hợp của hai mô hình trên.Mô hình này sẽ cải thiện thời gian tính toán của mô hình liên tục.

f/ Tổ chức nhận dạng từ bằng mô hình markvo ẩn:

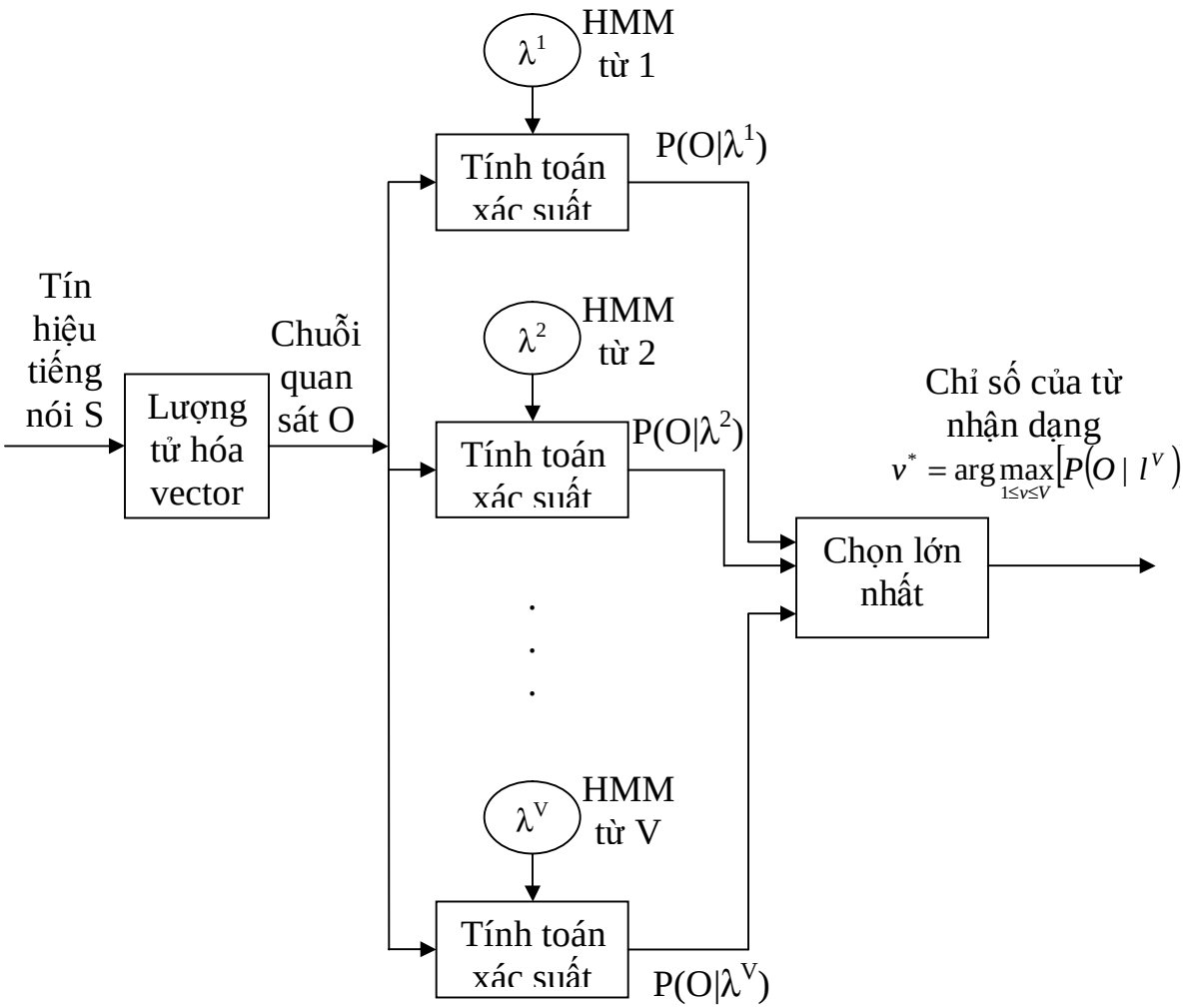
Giả sử ta cần nhận dạng bộ từ vựng có V từ,mỗi từ đều có mô hình marvo riêng và được nói K lần ta thực hiện các bước sau:

Ø ***Bước 1***: với mọi từ v trong bộ từ vựng , ta phải xây dựng mô hình marvo ẩn λ, tức là ta phải ước lượng các tham số của mô hình (A, B ,π) sao cho ML dựa trên tập dữ liệu huấn luyện.

Ø ***Bước 2***: với mỗi từ chưa biết ta xây dựng mô hình nhận dạng như hình bên dưới. Tín hiệu tiếng nói được trích đặc điểm bằng phương pháp mel-cepstrum hay LPC-cepstrum, thông qua bộ lượng tử hóa vector ta có được quan sát O = (o_1, o_2, …, o_T).Tiếp theo ta tính xác suất cho tất cả các mô hình $P(O|\lambda_v)$, $1 \le v \le V$, và chọn từ có xác suất lớn nhất, tức là:

$$v^* = \arg\max_{1 \le v \le V} [P(O| l_v)]$$

Bước tính xác suất thường dùng thuật toán Viterbi và cần $V.N^2.T$ phép tính.Với bộ từ vựng V=100 từ, mô hình 5 trạng thái và T=40 quan sát cho mọi từ chưa biết tổng cộng có 10^3 phép tính. Điều này có thể chấp nhận được cho các máy tính ngày nay.



Hình 3.6.2: Sơ đồ khối hệ nhận dạng từ bằng mô hình markvo ẩn

3.6.3 Mô hình mạng neuron:

Mạng Neuron cũng được ứng dụng trong nhận dạng tiếng nói. Ưu điểm của mạng neuron trong nhận dạng tiếng nói là: thứ nhất về tốc độ huấn luyện cũng như tốc độ nhận dạng tỏ ra vượt trội,có thể mở rộng bộ từ vững. Do đó mạng neuron có tính linh hoạt ,mềm dẻo dễ thích nghi với môi trường.Ta sẽ xem xét chi tiết hơn về mô hình này ở ***chương 4***.

3.7 Những thuận lợi và khó khăn trong nhận dạng tiếng Việt

Một số đặc điểm dễ thấy là tiếng Việt là ngôn ngữ đơn âm, không biến hình (cách đọc cách ghi âm không thay đổi trong bất cứ tình huống ngữ pháp nào). Theo thống kê trong tiếng Việt có khoảng 6000 âm tiết. Nhìn về mặt ghi âm: âm tiết có cấu tạo chung là: phụ âm – vần. Phụ âm là một âm vị và âm vị này liên kết rất lỏng lẻo với phần còn lại của âm tiết. Vần trong tiếng việt lại được cấu tạo từ các âm vị nhỏ hơn, trong đó có một âm vị chính là nguyên âm

Do những đặc điểm như vậy, nhận dạng tiếng nói tiếng Việt có một số thuận lợi:

- Tiếng Việt là ngôn ngữ đơn âm, số lượng âm tiết không quá lớn. Điều này sẽ giúp hệ nhận dạng xác định ranh giới các âm tiết dễ dàng hơn.

- Tiếng Việt là ngôn ngữ không biến hình từ. Âm tiết tiếng Việt ổn định, có cấu trúc rõ ràng. Đặc biệt không có 2 âm tiết nào đọc giống nhau mà viết khác nhau. Điều này sẽ dễ dàng trong việc xây dựng các mô hình âm tiết trong nhận dạng

Ngoài những thuận lợi trên, nhận dạng tiếng nói tiếng Việt cũng gặp rất nhiều khó khăn như sau:

- Tiếng Việt là ngôn ngữ có thanh điệu (6 thanh). Thanh điệu là âm vị siêu đoạn tính, đặc trưng về thanh điệu thể hiện trong tín hiệu tiếng nói không rõ nét như các thành phần khác của âm tiết.
- Cách phát âm tiếng việt thay đổi theo từng vùng địa lý.
- Hệ thống ngữ pháp ngữ nghĩa tiếng Việt rất phức tạp, rất khó để áp dụng vào hệ nhận dạng với mục đích tăng hiệu năng nhận dạng. Hệ thống phiên âm cũng chưa thống nhất.
- Các nghiên cứu nhận dạng cũng chưa nhiều và ít phổ biến.

Nhưng khó khăn cơ bản trong nhận dạng tiếng nói đó là tiếng nói biến thiên theo thời gian và có sự khác biệt lớn giữa tiếng nói của những người nói khác nhau, tốc độ nói, ngữ cảnh và môi trường âm học khác nhau.

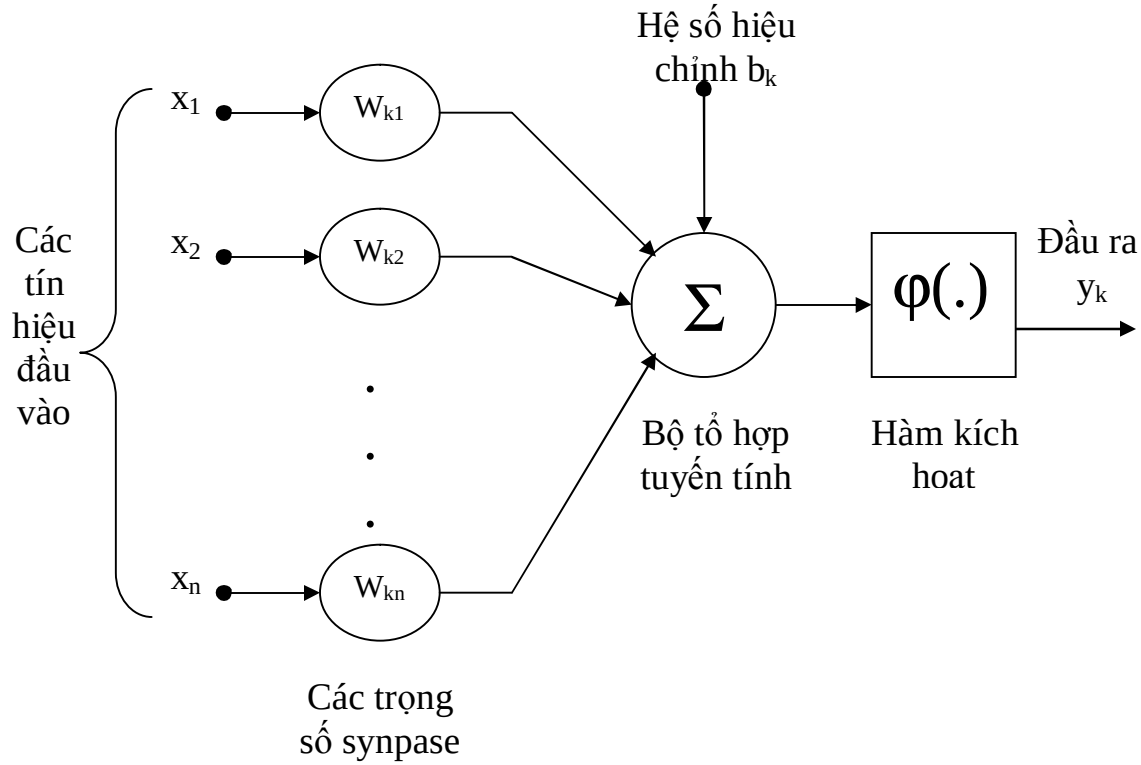
- hết Chương 3 -

CHƯƠNG 4: MẠNG NEURON

4.1 Định nghĩa mạng neuron:

Lý thuyết mạng neuron nhân tạo được xây dựng xuất phát từ thực tiễn là bộ não con người luôn luôn thực hiện các tính toán một cách hoàn toàn khác so với các máy tính số. Có thể coi bộ não là một máy tính hay một hệ thống xử lý thông tin song song, phi tuyến và cực kì phức tạp. Sự mô phỏng bộ não con người của mạng neuron là dựa trên cơ sở một số tính chất đặc thù rút ra từ các nghiên cứu về thần kinh.

Một neuron nhân tạo là một đơn vị tính toán hay đơn vị xử lý thông tin cơ sở cho hoạt động của một mạng neuron. Về cơ bản, mạng neuron là sự kết hợp các thành phần phi tuyến với nhau. Hình 4.1 chỉ ra mô hình của một neuron nhân tạo.



Hình 4.1: Mô hình của mạng neuron

Một mô hình mạng neuron có ba thành phần cơ bản:

1. Một tập hợp các synpase hay các kết nối, mà mỗi một trong chúng được đặc trưng bởi một trọng số riêng của nó.Tức là một tín hiệu x_j tại đầu vào của synpase j nối với neuron k sẽ được nhân với trọng số synpase w_{kj}. Ở đó k là chỉ số của neuron tại

đầu ra của synapse đang xét. Các trọng số synpase có thể nhận cả giá trị âm và giá trị dương.

2. Một bộ cộng để tính tổng các tín hiệu đầu vào của neuron, đã được nhân với các trọng số synpase tương ứng; phép toán được mô tả ở đây tạo nên một tổ hợp tuyến tính.

3. Một hàm kích hoạt (activation function) để giới hạn biên độ đầu ra của neuron. Hàm kích hoạt cũng được xem như là một hàm nén; nó nén (giới hạn) phạm vi biên độ cho phép của tín hiệu đầu ra trong một khoảng giá trị hữu hạn. Hàm kích hoạt có nhiều kiểu như: hàm ngưỡng ,hàm vùng tuyến tính, hàm sigma, hàm tang hyperbol.Trong đó hàm tan sigmoid (4.1.1) hay log sigmoid (4.1.2) hayđược dùng nhất:

$$j\ (x) = tanh(bx) \tag{4.1.1}$$

$$j\ (x) = \frac{1}{1+\exp(-bx)} \tag{4.1.2}$$

Ngoài ra còn có một hệ số hiệu chỉnh b_k tác động từ bên ngoài có tác dụng tăng lên hoặc giảm đi đầu vào thực của hàm kích hoạt, tùy theo nó dương hay âm.

Tín hiệu đầu ra được cho bởi:

$$y_k = j\left(\sum_{i=1}^{N} w_{ki}x_i + b\right)$$

Mạng neuron nhân tạo đang được ứng dụng rộng rãi trong các ngành kĩ thuật như: trong kỹ thuật điều khiển, mạng neuron được ứng dụng để nhận dạng, dự báo và điều khiển các hệ thống động; trong điện tử viễn thông thì ứng dụng để xử lý ảnh, nhận dạng ảnh và truyền thông; trong hệ thống điện thì ứng dụng để nhận dạng, dự báo và điều khiển các trạm biến áp...

4.2 Kiến trúc mạng neuron:

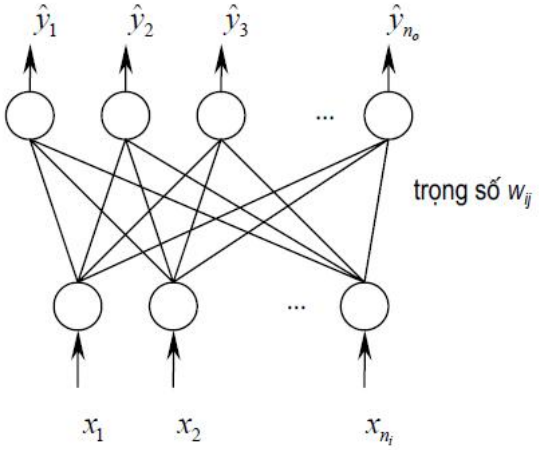
Thông thường thì có ba loại mạng neuron:

4.2.1 Perceptron một hay nhiều lớp:

Perceptron một lớp, chỉ có một lớp neuron, là kiến trúc xuất hiện đầu tiên trong lĩnh vực mạng neuron và không còn sử dụng nữa vì nó quá đơn giản. Perceptron nhiều lớp-multiple layer perceptron (MLP) được sử dụng rộng rãi nhất.

Mạng Perceptron tuyến tính đơn SLP

Mạng SLP (Simple Linear Perceptron) bao gồm một lớp nút vào (input) và lớp nút ra (output). Với mỗi một vector giá trị đầu vào, các giá trị input được đưa vào các nút input, và mạng ANN sẽ cho kết quả tương ứng tại các nút output. Ký hiệu các nút đầu vào *xi* là x_1, x_2 …x_{ni} , trong đó n_i là số lượng nút vào; các nút đầu ra y_i là y_1, y_2…y_{no}, n_0 là số lượng nút ra. Mỗi một nút input x_i liên hệ đều có một nối kết (connection hay synapses) với một nút output y_i. Mỗi nối kết được gán một giá trị, gọi là trọng số (synapses strength), ký hiệu là w_{ij}. Các tín hiệu vào được lan truyền theo các nối kết và được nhân với các trọng số của mỗi nối kết. Tính toán tại lớp vào sẽ được lan truyền sang lớp kế tiếp và do vậy mạng được gọi là lan truyền thẳng (feed-forward).



Hình 4.2.1: Mạng neuron Perceptron đơn

Tại mỗi nút output của mạng, các tín hiệu vào sẽ được nhân với các trọng số và sau đó được cộng lại thành giá trị output như được miêu tả bởi công thức sau:

$$y_i = \sum_{j=1}^{n_i} w_{ij} x_j \qquad (4.2.1.1)$$

Gọi tập dữ liệu mẫu dùng để huấn luyện là (x_k, y_k). Với tập dữ liệu mẫu, mạng ANN với các trọng số, bài toán huấn luyện mạng được đặt ra như là điều chỉnh các trọng số sao cho với mỗi vector giá trị vào x_k , mạng cho một kết quả tương ứng $y^\wedge_k$, gần nhất với kết quả mong muốn theo một tiêu chuẩn nào đó. Lựa chọn thông dụng cho một hàm tiêu chuẩn là hàm bình phương tối thiểu (least square criterion).

$$E = \frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{n_o} (y_{i,k} - \hat{y}_{i,k})^2$$

(4.2.1.2)

trong đó , $y^{\wedge}_{i,k}$ là giá trị output của nút ra thứ *i* tương ứng với vector giá trị vào x_k. $y_{i.k}$ là giá trị mong muốn tương ứng của tập dữ liệu huấn luyện. *K* là số lượng các mẫu trong tập huấn luyện. Giá trị 1/2 trong công thức với mục đích thuận tiện cho tính toán, khi lấy đạo hàm vế phải của (4.2.1.2). Quá trình huấn luyện được thực hiện với mục đích giảm giá trị hàm lỗi *E*. Một phương pháp thông dụng để giảm giá trị hàm lỗi *E* được áp dụng trong hầu hết các mạng là phương pháp giảm gradient.

Phương pháp giảm gradient là một kỹ thuật tối ưu hoá đảm bảo hội tụ về một giá trị cực tiểu cục bộ. Phương pháp được tiến hành theo nhiều vòng lặp, mỗi vòng lặp các giá trị trọng số được điều chỉnh theo hướng ngược với giá trị gradient. Gọi w là giá trị trọng số tại một bước của thuật toán, giá trị trọng số mới được tính toán cho bước tiếp theo:

$$w = w' + \Delta w$$

trong đó Δw biểu diễn sự thay đổi của trọng số, tỷ lệ với giá trị $\nabla_w C$, là giá trị lỗi vector gradient được tính toán theo trọng số *w*

$$w = -\alpha \nabla_w C$$

Các giá trị trọng số được biểu diễn bằng công thức:

$$\Delta w_{ij} = -\alpha \frac{\partial C}{\partial w_{ij}}$$

Trong đó α được gọi là hệ số học (learning rate). Hệ số học quyết định tốc độ hội tụ của mạng. Nếu hệ số học nhỏ tốc độ hội tụ sẽ chậm, ngược lại nếu hệ số học lớn thì tốc độ hội tụ sẽ nhanh hơn. Tuy nhiên nếu hệ số học quá lớn sẽ làm thuật toán khó tiếp cận gần đến điểm cực tiểu. Giá trị tốt nhất của hệ số học phải đảm bảo để mạng hội tụ nhanh, mặt khác đảm bảo để giá trị hàm lỗi *E* là nhỏ nhất.

Trong quá trình huấn luyện, bước lặp đầu tiên sẽ bắt đầu với các trọng số được khởi tạo trước. Sau đó qua các bước lặp mạng sẽ điều chỉnh các trọng số theo hướng giảm gradient để cuối cùng hội tụ tại một điểm cực tiểu địa phương (local minimum). Về mặt lý thuyết, quá trình học có thể bị kẹt (stuck) tại một giá trị cực tiểu địa phương mà không thể tới được giá trị cực tiểu toàn cục (global). Để giải quyết vấn đề này, mạng có thể được huấn luyện vài lần với tập các trọng số được khởi tạo khác nhau.

Quá trình huấn luyện thường được tiến hành với tất cả tập dữ liệu mẫu tại mỗi bước lặp (được gọi là tính toán theo lô, batch mode), quá trình huấn luyện thường mất rất nhiều thời gian. Trong thực tế thay cho tính toán giá trị gradient (4.2.1.2) của toàn bộ dữ liệu mẫu, giá trị gradient được tính toán trực tiếp với mỗi một cặp dữ liệu mẫu (x_k, y_k).

$$E = \frac{1}{2}\sum_{i=1}^{n_o}(y_i - \hat{y}_i)^2 \qquad (4.2.1.3)$$

Các trọng số được cập nhật vẫn theo các công thức như đã trình bầy ở trên. Phương pháp này được gọi là giảm gradient ngẫu nhiên (stochastic gradient descent) và đã được chứng minh là hiệu quả hơn nếu số mẫu huấn luyện là lớn (khoảng vài trăm trở lên).

Từ phương trình (4.2.1.3) lấy đạo hàm riêng theo từng trọng số ta có:

$$\frac{\partial C}{\partial w_{ij}} = \frac{1}{2}\sum_{k=1}^{n_o}\frac{\partial (y_k - \hat{y}_k)^2}{\partial w_{ij}}$$

$$= \frac{1}{2}\frac{\partial (y_i - \hat{y}_i)^2}{\partial w_{ij}}$$

$$= -(y_i - \hat{y}_i)\frac{\partial \hat{y}_i}{\partial w_{ij}} \qquad (4.2.1.4)$$

Từ phương trình (4.2.1.1) ta có

$$\frac{\partial \hat{y}_i}{\partial w_{ij}} = \frac{\partial \sum_{l=1}^{n_i} w_{il} x_l}{\partial w_{ij}}$$

$$= x_i \qquad (4.2.1.5)$$

Từ hai phương trình (4.2.1.4) và (4.2.1.5) ta có

$$\Delta w_{ij} = -\alpha\,(y_i - \hat{y}_i)\,x_j \qquad (4.2.1.6)$$

Phương trình (4.2.1.6) cho thấy sự biến thiên của trọng số của mạng sau khi có một giá trị vào mạng, giá trị này tỷ lệ với hiệu số giữa giỏ trị tại các nút output và giá trị ra mong muốn nhận được.

Ta định nghĩa đại lượng

$$\delta = y_i - \hat{y}_i$$

Khi đó phương trình (4.2.1.6) được viết lại là

$$\Delta w_{ij} = -\alpha\delta\, x_j \qquad (4.2.1.7)$$

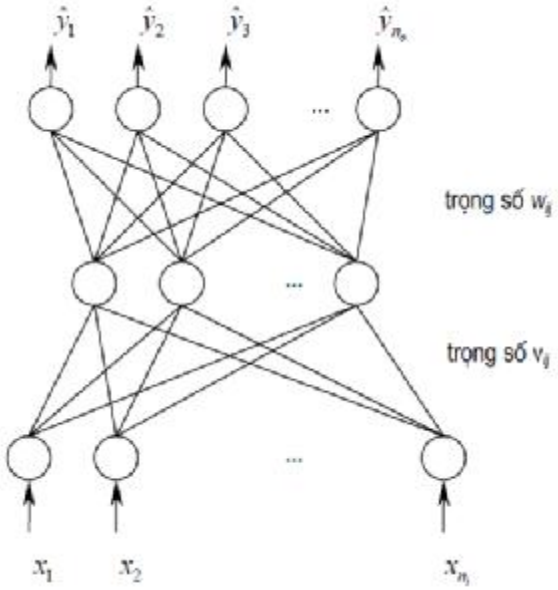
Phương trình (4.7) được gọi là luật delta (delta rule).

Mạng Perceptron đa lớp MLP

Một trong những cấu trúc thông dụng nhất của mạng neuron là mạng Perceptron đa lớp MLP (MultiLayer Perceptron). Mạng MLP gồm có một lớp vào (input), một lớp ra

(output) và một hoặc nhiều lớp ẩn. Mạng MLP cũng có thể được hiểu là mạng Perceptron một lớp được bổ sung thêm một hoặc nhiều lớp ẩn. Một vector đầu vào sẽ được đưa vào lớp vào (input) của mạng và sau đó các tính toán được thực hiện lan truyền thẳng (feed-forward) từ lớp vào input sang các lớp ẩn và kết thúc ở lớp ra output. Hàm kích hoạt kết hợp với các nút ẩn hay các nút output có thể là hàm tuyến tính hay phi tuyến và có thể khác nhau giữa các nút. Hình 4.2.2 miêu tả một ví dụ mạng Perceptron đa lớp.

Quá trình huấn luyện mạng MLP là quá trình học có giám sát, các trọng số giữa các nút của hai lớp kế tiếp được điều chỉnh theo một hàm tiêu chuẩn nào đó (criterion function). Hàm tiểu chuẩn thông dụng hay được dùng giống như mạng Perceptron đơn lớp là hàm tổng bình phương hiệu số giữa các giá trị output và các giá trị mong muốn của các nút ra.



Hình 4.2.2: Mạng neuron Perceptron đa lớp MLP

Giả thiết rằng mạng MLP gồm có ba lớp, trong đó có một lớp ẩn như miêu tả trong Hình 4.2.2. Gọi hàm kích hoạt đối với các nut ẩn là ρ (*x*) , hàm kích hoạt đối với nút ra là σ (*x*) , ta có trọng số w_{ij} giữa nút ẩn *j* và nút ra *i* được điều chỉnh theo hàm lỗi *E* như sau:

$$E = \frac{1}{2}\sum_{n=1}^{n_o}(y_n - \hat{y}_n)^2$$

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}}$$

Ta có thể tính

$$\frac{\partial E}{\partial w_{ij}} = \frac{1}{2}\sum_{n=1}^{n_o}\frac{\partial (y_n - \hat{y}_n)^2}{\partial w_{ij}}$$

$$-\frac{1}{2}\frac{\partial (y_i - \hat{y}_i)^2}{\partial w_{ij}}$$

$$- -(y_i - \hat{y}_i)\frac{\partial y_i}{\partial w_{ij}} \qquad (4.2.1.8)$$

Trong đó ta có thể tính:

$$\frac{\partial y_i}{\partial w_{ij}} = \frac{\partial \sigma(\overline{y}_i)}{\partial w_{ij}}$$

$$= \frac{\partial \sigma(\overline{y}_i)}{\partial \overline{y}_i}\frac{\partial \overline{y}_i}{\partial w_{ij}}$$

$$= \sigma_i'(\overline{y}_i)\frac{\partial \sum_{l=1}^{n_h} w_{il}\hat{y}_l}{\partial w_{ij}}$$

$$= \sigma_i'(\overline{y}_i)\hat{y}_j \qquad (4.2.1.9)$$

Từ hai phương trình (4.2.1.8) và (4.2.1.9) ta có

$$\Delta w_{ij} = -(y_i - \hat{y}_i)\sigma_i'(y_i)\hat{y}_j$$

Đặt:

$$\varepsilon_i'' = (y_i - \hat{y}_i)\sigma_i'(\overline{y}_i) \qquad (4.2.1.10)$$

Cuối cùng ta có công thức điều chỉnh trọng số tương tự như trường hợp của mạng Perceptron đơn lớp như sau:

$$\Delta w_{ij} = \alpha\varepsilon_i''\hat{y}_j \qquad (4.2.1.11)$$

Bây giờ ta xem xét trường hợp của trọng số v_{jk} giữa nút vào *k* và nút ẩn *j* được điều chỉnh theo hàm lỗi *E*:

$$E - \frac{1}{2}\sum_{n=1}^{n_o}(y_n - \hat{y}_n)^2$$

$$\Delta v_{jk} = \alpha \frac{\partial E}{\partial v_{jk}}$$

Ta có thể tính

$$\frac{\partial E}{\partial v_{jk}} = \frac{1}{2}\sum_{n=1}^{n_o}\frac{\partial (y_n - \hat{y}_n)^2}{\partial v_{jk}}$$

$$= -\sum_{n=1}^{n_o}(y_n - \hat{y}_n)\frac{\partial \hat{y}_n}{\partial v_{jk}} \qquad (4.2.1.12)$$

Ta có thể thấy từ phương trình (4.2.1.12) là sự thay đổi của trọng số v_{jk} liên quan đến toàn bộ các nút ra output của mạng.

$$\frac{\partial v_n}{\partial v_{jk}} = \frac{\partial \rho(\overline{y}_n)}{\partial v_{jk}}$$

$$= \frac{\partial \sigma(\overline{y}_i)}{\partial \overline{y}_i}\frac{\partial \overline{y}_i}{\partial w_{ij}}$$

$$- \sigma_i'(\overline{y}_i)\frac{\partial \sum_{l=1}^{n_h} w_{il}\hat{y}_l}{\partial w_{ij}}$$

$$- \sigma_i'(y_i)\hat{y}_j \qquad (4.2.1.13)$$

Từ hai phương trình (4. 2.1.8) và (4. 2.1.9) ta có

$$\Delta w_{ij} = (y_i \quad \hat{y}_i)\sigma_i'(\overline{y}_i)\hat{y}_j$$

Đặt: $\varepsilon_i^o = (y_i - \hat{y}_i)\sigma_i'(\overline{y}_i)$

Cuối cùng ta có công thức điều chỉnh trọng số tương tự như trường hợp của mạng Perceptron đa lớp như sau:

$$\Delta w_{ij} = \alpha \varepsilon_i^o \hat{y}_j \qquad (4.2.1.14)$$

Bây giờ ta xem xét trường hợp của trọng số v_{jk} giữa nút vào *k* và nút ẩn *j* được điều chỉnh theo hàm lỗi *E*:

$$E = \frac{1}{2}\sum_{n=1}^{n_o}(y_n - \hat{y}_n)^2$$

$$\Delta v_{jk} = -\alpha \frac{\partial E}{\partial v_{jk}} \qquad (4.2.1.15)$$

Ta có thể tính

$$\frac{\partial E}{\partial v_{jk}} = \frac{1}{2}\sum_{n=1}^{n_o}\frac{\partial (y_n - \hat{y}_n)^2}{\partial v_{jk}}$$

$$= -\sum_{n=1}^{n_o}(y_n - \hat{y}_n)\frac{\partial \hat{y}_n}{\partial v_{jk}} \qquad (4.2.1.16)$$

Ta có thể thấy từ phương trình (4.12) là sự thay đổi của trọng số v_{jk} liên quan đến toàn bộ các nút ra output của mạng.

$$\frac{\partial \hat{y}_n}{\partial v_{jk}} = \frac{\partial \rho(\overline{y}_n)}{\partial v_{jk}}$$

$$= \frac{\partial \rho(\overline{y}_n)}{\partial \overline{y}_n}\frac{\partial \overline{y}_n}{\partial v_{jk}}$$

$$= \rho'(\overline{y}_n)\frac{\partial \overline{y}_n}{\partial v_{jk}} \qquad (4.2.1.17)$$

Trong đó:

$$\frac{\partial \overline{y}_n}{\partial v_{jk}} = \frac{\partial \sum_{i=1}^{n_h} w_{ni}\hat{h}_i}{\partial v_{jk}}$$

$$= \sum_{i=1}^{n_h} w_{ni}\frac{\partial \hat{h}_i}{\partial v_{jk}}$$

$$= w_{nj}\frac{\partial \hat{h}_j}{\partial v_{jk}} \qquad (4.2.1.18)$$

Trong đó $h_j^{\wedge}$ là giá trị đầu ra của nút ẩn thứ *j*. Ta tiếp tục tính:

$$\frac{\partial \hat{h}_j}{\partial v_{jk}} = \frac{\partial \rho(\bar{h}_j)}{\partial v_{jk}}$$

$$- \frac{\partial \rho(\bar{h}_j)}{\bar{h}_j} \frac{\partial \bar{h}_j}{\partial v_{jk}}$$

$$= \rho_j'(v_j) \frac{\partial \sum_{m=1}^{n_i} v_{jm} x_m}{\partial v_{jk}}$$

$$= \rho_j'(\bar{h}_j) x_k \qquad (4.2.1.19)$$

Từ các phương trình (4. 2.1.15), (4. 2.1.16), (4. 2.1.17) và (4. 2.1.19) ta có

$$\Delta v_{jk} = \alpha \sum_{n=1}^{n_o} \left[(y_n - \hat{y}_n) \sigma_n'(\bar{y}_n) w_{nj} \right] \rho_j'(\bar{h}_j) x_k$$

$$= \alpha \left\{ \sum_{n=1}^{n_o} \left[w_{nj} (y_n - \hat{y}_n) \sigma_n'(\bar{y}_n) \right] \right\} \rho_j'(\bar{h}_j) x_k$$

$$= \alpha \left(\sum_{n=1}^{n_o} w_{nj} \varepsilon_n^o \right) \rho_j'(\bar{h}_j) x_k \qquad (4.2.1.20)$$

Đặt $\varepsilon_j^h = \left(\sum_{n=1}^{n_o} w_{nj} \varepsilon_n^o \right) \rho_j'(\bar{h}_j)$

ta sẽ có công thức cập nhật trọng số tương tự như công thức (4. 2.1.11):

$$\Delta v_{jk} = \alpha \varepsilon_j^h x_k \qquad (4.2.1.21)$$

Các phương trình (4.2.1.11) và (4.2.1.21) tạo thành một tập phương trình được gọi là qui tắc delta tổng quát (Generalized Delta Rule)

Từ công thức (4.2.1.20) cho ta thấy trong quá trình học, giá trị hàm lỗi là bình phương hiệu số giữa giá trị output của mạng và giá trị mong muốn của tập mẫu được tính toán tạo thành giá trị delta của lớp output, giá trị này được dùng để hiệu chỉnh các trọng số liên kết với lớp output. Sau đó giá trị delta này lan truyền ngược về phía lớp ẩn cho phép tính toán các trọng số liên kết với lớp ẩn theo phương trình

(4.2.1.20). Chính vì vậy quá trình này được gọi là học lan truyền ngược sai số (Error Back Propagation).

4.2.2 Mạng neuron hồi quy (RNN)

Mạng neuron hồi quy (Recurrent neural networks-RNN): nó có một hoặc nhiều con đường quay lùi từ đầu ra trở lại đầu vào. Những vòng lặp quay lui này làm cho mạng ghi nhớ được các sự kiện trước đó.Vì vậy RNN rất hữu ích những lĩnh vực nhận dạng mẫu theo thời gian. Trong đồ án này, mạng hồi quy và mạng tự tổ chức sẽ không được nói đến một cách chi tiết.

4.2.3 Mạng tự tổ chức

Các neuron sắp xếp có thứ tự trong một mảng lớn nhiều chiều. Loại mạng này gần giống với lượng tử hóa vector. Điểm đặc biệt của mạng này là khả năng học mà không cần thầy hoặc không cần giám sát.

4.3 Đặc trưng của mạng neuron:

4.3.1 Tính chất phi tuyến:

Một mạng neuron, cấu thành bởi sự kết nối các neuron phi tuyến thì tự nó sẽ có tính phi tuyến. Hơn nữa, điều đặc biệt là tính phi tuyến này được phân tán trên toàn mạng. Tính phi tuyến là thuộc tính rất quan trọng, nhất là khi các cơ chế vật lý sinh ra các tín hiệu đầu vào vốn là phi tuyến.

4.3.2 Tính chất tương ứng đầu vào – đầu ra:

Tính chất này này liên quan tới vấn đề "học" hay "tích lũy" của mạng neuron. Một mô hình học phổ biến được gọi là học với một người dạy hay học có giám sát liên quan đến việc thay đổi các trọng số synapse của mạng neuron bằng việc áp dụng một tập hợp các mẫu tích lũy hay các ví dụ tích lũy. Mỗi một ví dụ tích lũy bao gồm một tín hiệu đầu vào và một đầu ra mong muốn tương ứng. Mạng neuron nhận một ví dụ lấy một cách ngẫu nhiên từ tập hợp nói trên tại đầu vào của nó, và các trọng số synapse của mạng được biến đổi sao cho có thể cực tiểu hóa sự sai khác giữa đầu ra mong muốn và đầu ra thực sự của mạng theo một tiêu chuẩn thống kê thích hợp.Như vậy mạng neuron học từ các ví dụ bằng cách xây dựng nên một tương ứng đầu vào- đầu ra cho vấn đề cần giải quyết.

4.3.3 Tính chất thích nghi:

Các mạng neuron có một khả năng mặc định là biến đổi các trọng số synapse tùy theo sự thay đổi của môi trường xung quanh. Đặc biệt, một mạng neuron đã tích lũy để hoạt động trong một môi trường xác định có thể tích lũy lại một cách dễ dàng khi có những thay đổi nhỏ của các điều kiện môi trường hoạt động. Khi hoạt động trong môi trường không ổn định, một mạng neuron có thể được thiết kế sao cho có khả năng thay đổi các trọng số synapse của nó theo thời gian thực. Tuy nhiên tính chất này không phải lúc nào cũng đem đến sức mạnh mà nó có thể làm điệu ngược lại.

4.3.4 Tính chất đưa ra lời giải có bằng chứng:

Trong ngữ cảnh phân loại mẫu, một mạng neuron có thể được thiết kế để đưa ra thông tin không chỉ về mẫu được phân loại, mà còn về sự tin cậy của quyết định đã được thực hiện. Thông tin này có thể được sử dụng để loại bỏ các mẫu mơ hồ hay nhập nhằng.

4.3.5 Tính chất chấp nhận sai sót:

Một mạng neuron được cài đặt dưới dạng phần cứng, vốn có khả năng chấp nhận lỗi hay khả năng tính toán thô, với ý nghĩa là tính năng của nó chỉ thoái hóa (chứ không đổ vỡ) khi có những điều kiện hoạt động bất lợi. Để đảm bảo rằng mạng neuron thực sự có khả năng chấp nhận lỗi, có lẽ cần phải thực hiện những đo đạc hiệu chỉnh trong việc thiết kế thuật toán tích lũy mạng neuron.

4.3.6 Tính chất đồng dạng trong phân tích và thiết kế:

Đặc tính này thể hiện một số điểm như sau :

- Các neuron dưới dạng này hoặc dạng khác biểu diễn một thành phần chúng cho tất cả các mạng neuron.
- Tính thống nhất đã đem lại khả năng chia sẽ các lý thuyết và các thuật toán học trong nhiều ứng dụng khác nhau của mạng neuron.
- Các mạng tổ hợp có thể được xây dựng thông qua một sự tích hợp các mô hình khác nhau.

- hết Chương 4 -

CHƯƠNG 5: GIỚI THIỆU HÀM VÀ TOOBOX TRONG MATLAB CẦN ĐỂ XÂY DỰNG HỆ THỐNG NHẬN DẠNG TIẾNG NÓI BẰNG MẠNG NEURON

Như chúng ta đã biết, Matlab (Matrix Laboratory) là một môi trường trợ giúp tính toán và hiển thị rất mạnh được hãng MathWorks phát triển. Mức phát triển của Matlab ngày nay đã chứng tỏ Matlab là một phần mềm có giao diện cực mạnh cùng nhiều lợi thế trong kĩ thuật lập trình để giải quyết những vấn đề đa dạng trong nghiên cứu KHKT.

Ngoài thư viện các hàm tính toán, vào-ra, đồ hoạ… cơ bản, Matlab còn có các toolbox là các thư viện cho từng lĩnh vực cụ thể. Ví dụ có toolbox cho xử lí tín hiệu (Signal Processing), mô phỏng mô hình (SimulLink), logic mờ (Fuzzy Logic), mạng nơron (NNet), … thậm chí cho cả thiết kế máy bay (Aerospace) hay giải phương trình vi phân (PDE)…

Chương này tập trung chủ yếu vào giới thiệu các hàm và toolbox và hàm cần thiết để xây dựng mộ mô hình nhận dạng tiếng nói dung mạng nơron, mà cụ thể là xây dựng mạng MLP ba lớp.

Các hàm xử lý âm thanh:

`[y fs]=wavread(wavfile)`	Đọc tín hiệu âm thanh từ file wav cho bởi xâu wavfile, ***y*** là vector mô tả tín hiệu âm thanh (có giá trị thực từ 0 đến 1), ***fs*** là tần số lấy mẫu (giá trị nguyên)
`wavwrite(y,fs,wavfile)`	Ghi tín hiệu âm thanh từ file wav cho bởi xâu wavfile, ***y*** là vector mô tả tín hiệu âm thanh, ***fs*** là tần số lấy mẫu.
`sound(y)`	Phát âm thanh ra loa, ***y*** là vector mô tả tín hiệu âm thanh.
`y=wavrecord(n, fs)`	Ghi âm (từ micro) với tần số lấy mẫu ***fs*** và ***n*** mẫu. Kết quả là vector ***y***. Đoạn lệnh sau ghi âm trong 2 giây với tần số lấy mẫu 8kHz, rồi ghi vào file: **`y=wavrecord(16000,8000);`** **`wavwrite(y,8000,'temp.wav');`**

VoiceBox toolbox

VoiceBox là một toolbox của Matlab chuyên về xử lí tiếng nói do Mike Brookes phát triển. VoiceBox yêu cầu Matlab phiên bản 5 trở lên. Voicebox có thể tải về từ *http://webscripts.softpedia.com/script/Scientific-Engineering-Ruby/Signal-Processing/Voicebox-34702.html.*

VoiceBox gồm các hàm có thể chia thành một số nhóm chức năng sau:

- Xử lí file âm thanh (đọc, ghi file wav và một số định dạng file âm thanh khác)
- Phân tích phổ tín hiệu
- Phân tích LPC
- Tính toán MFCC, chuyển đổi spectral - cepstral
- Chuyển đổi tần số (mel-scale, midi,...)
- Biến đổi Fourier, Fourier ngược, Fourier thực...
- Tính khoảng cách (sai lệch) giữa các vector và dãy vector.
- Loại trừ nhiễu trong tín hiệu tiếng nói.

Chức năng quan trọng nhất là trích đặc trưng tín hiệu tiếng nói, mà ở đây là 2 loại phổ biến nhất LPC và MFCC.

Hàm tính MFCC của tín hiệu trong VoiceBox là hàm :

```
melcepst(s,fs,w,nc,p,n,inc,fl,fh)
```

Hàm có nhiều tham số, một số tham số quan trọng là:

- ***s*** là vector tín hiệu tiếng nói (có được sau khi dùng hàm hoặc), fs là tần số lấy mẫu (mặc định là 11050).
- ***nc*** là số hệ số MFCC cần tính (tức là số phần tử của vector đặc trưng, mặc định là 12).
- ***p*** là số bộ lọc mel-scale.
- ***w*** là một xâu mô tả các lựa chọn khác: nếu có thì tính thêm log năng lượng, có thì tính thêm đặc trưng delta.

Mặc dù vậy hàm có thể gọi một cách đơn giản là:

```
c=melcepst(s,fs);
```

Lời gọi hàm sinh ra ma trận c, mỗi dòng của ma trận là 12 hệ số MFCC của một frame. Để kèm thêm log năng lượng và dữ liệu delta như trong các hệ nhận dạng khác, ta dùng lệnh:

```
c=melcepst(s,fs,'ed');
```

Khi đó mỗi dòng của c là vector 26 hệ số MFCC của frame tương ứng.

NetLab toolbox

NetLab do Ian T. Nabney phát triển. Chúng tôi sử dụng toolbox NetLab để xây dựng, huấn luyện và thử nghiệm mạng nơron MLP cho hệ thống nhận dạng trong đồ án này. Link tải về: *http://webscripts.softpedia.com/script/Scientific-Engineering-Ruby/Controls-and-Systems-Modeling/Netlab-32705.html*

Lệnh khởi tạo MLP trong NetLab có cú pháp như sau:

```
net = mlp(inode, hnode, onode, func, anpha);
```

Trong đó:

- ***inode, hnode, onode*** lần lượt là số nơron của lớp vào, lớp ẩn và lớp ra.
- ***func*** là kiểu hàm kích hoạt, func có thể có các giá trị **'logistic'**, **'softmax'**…
- ***anpha*** là ngưỡng của giá trị trọng số, thường lấy bằng 0.01.
- ***net*** là mạng MLP do hàm tạo ra.

Mạng MLP sau khi điều kiện khởi tạo có thể huấn luyện với một bộ dữ liệu huấn luyện cho trước. Lệnh huấn luyện MLP trong NetLab có cú pháp như sau:

```
[net, error] = mlptrain(net, x, t, its)
```

Trong đó:

- ***x, t*** là bộ dữ liệu huấn luyện. ***x*** là các vector đầu vào, ***t*** là các vector đầu ra cần đạt đến (target).
- ***its*** là số vòng huấn luyện (số lần thực hiện thuật toán lan truyền ngược lỗi).
- ***net*** là mạng nơron.
- ***error*** là tổng sai số của lần huấn luyện cuối cùng.

Sau khi huấn luyện ta có thể dùng mạng MLP để tính đầu ra ứng với các đầu vào bất kì Lệnh tính đầu ra y của MLP ứng với đầu vào x như sau:

```
y = mlpfwd(net, x)
```

Trong đó:

- ***x*** là một hay nhiều vector đầu vào
- ***y*** là các vector đầu ra tương ứng.

- hết Chương 5 -

CHƯƠNG 6: XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG NHẬN DẠNG TIẾNG NÓI BẰNG MẠNG NEURON MLP

Nhận dạng tiếng nói là một lĩnh vực tuy không mới nhưng vô cùng phức tạp. Nhận dạng tiếng nói được thế giới bắt đầu nghiên cứu cách đây hơn 50 năm, tuy nhiên những kết quả thực tế đạt được vô cùng khiêm tốn. Còn phải rất lâu nữa con người mới đạt đến việc xây dựng một hệ thống hiểu được tiếng nói như con người. Trong phạm vi chỉ là một đồ án môn học, chúng tôi chỉ xây dựng một **chương trình nhỏ nhận dạng mười chữ số tiếng Việt** bằng những công cụ có sẵn của Matlab. Chúng tôi cũng rất muốn xây dựng một hệ thống nhận dạng tiếng Việt với bộ từ điển lớn hơn, có thể ứng dụng được vào thực tế. Tuy nhiên do chỉ mới tiếp xúc ở lĩnh vực này nên khả năng, kiến thức của chúng tôi còn rất hạn chế, cộng vào đó là những khó khăn về thời gian, phương tiện…nên chúng tôi chỉ có thể xây dựng một hệ thống nhận dạng nhỏ. Trong tương lai nếu có điều kiện tiếp xúc và nghiên cứu sâu hơn về lĩnh vực này, chúng tôi mong muốn phát triển đồ án này lên để có thể ứng dụng trong thực tế.

6.1 Các bước xây dựng

Hệ thống nhận dạng mười chữ số tiếng Việt được xây dựng với các đặc trưng như sau:

- Phương pháp: nhận dạng từ đơn (isolate word recognition).
- Input: file wav, mỗi file chỉ chứa một từ. Hoặc ghi âm trực tiếp.
- Output: chữ số được nhận dạng trong file đầu vào.
- Bộ từ vựng: 11 từ đơn âm các chữ số tiếng Việt (“không”, “một”, “hai”... “mười”).

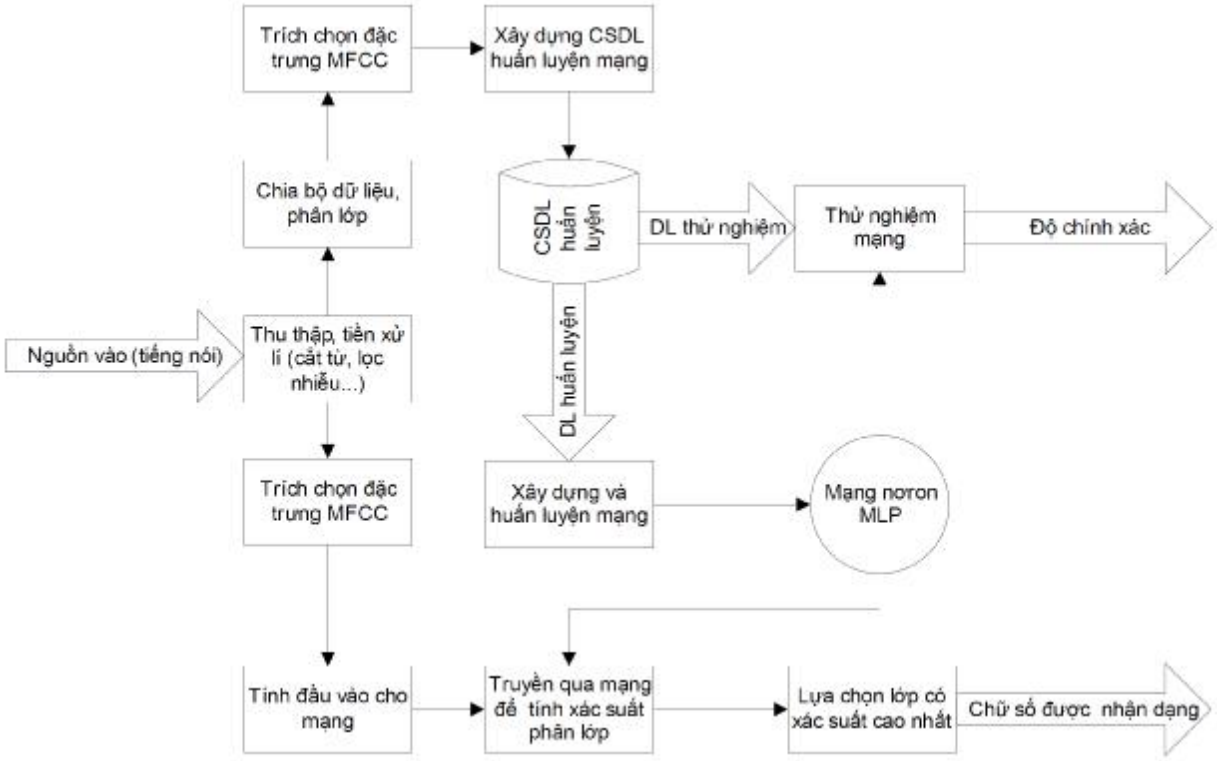
Sơ đồ khối hệ thống nhận dạng tiếng nói các chữ số tiếng Việt bằng mạng nơron MLP trên môi trường Matlab được mô tả trong hình 6.1. Chức năng của từng khối được mô tả như sau:

Thu thập và tiền xử lí: tín hiệu tiếng nói ở giai đoạn huấn luyện được thực hiện bằng phương pháp thủ công: sử dụng phần mềm ghi âm, lọc nhiễu và cắt thành các từ riêng rẽ, mỗi từ ghi vào một file (tên file ghi từ tương ứng).

Bộ dữ liệu do chúng tôi tự xây dựng gồm:

- File wav 16 bit 8kHz, mỗi file là phát âm của một từ.

- Từ là các chữ số tiếng Việt từ 1 đến 10 . (Mặc dù "mười" không phải là chữ số nhưng vẫn cần trong hệ nhận dạng chữ số vì có các số phát âm là "mười một", "mười hai"...).



Hình 6.1: Sơ đồ khối hệ thống nhận dạng tiếng nói các chữ số tiếng Việt bằng mạng nơron MLP trên môi trường Matlab

Việc thu thập và tiền xử lí (cắt các vùng không chứa tín hiệu tiếng nói) được thực hiện bởi các lệnh sau:

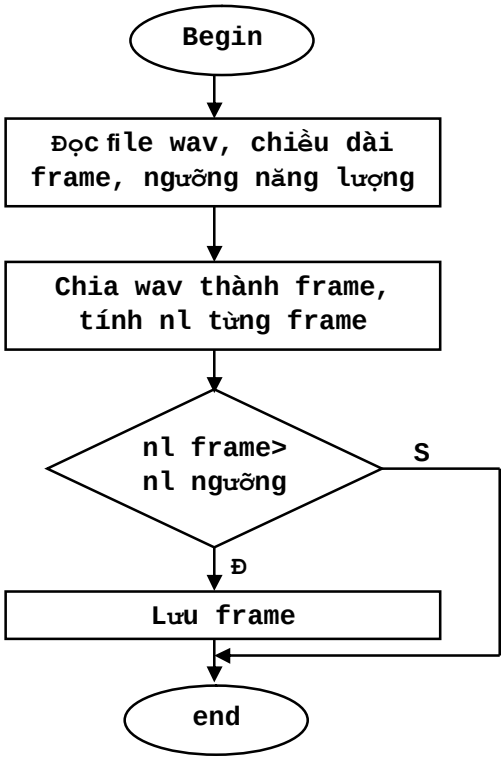
```
x = wavrecord(10000,8000); %tần số lấy mẫu 8kHz, ghi âm chừng hơn 1s
x = x';      %chuyển x thành ma trận dòng
y = endcut(x, 64, 1.5E-3); %cắt khoản lặng
```

Hàm endcut dùng cắt các khoảng lặng không chứa tín hiệu âm, sơ đồ giải thuật miêu tả trong hình 6.2. Các lệnh miêu tả như sau:

```
function y = endcut(x, n, es)
% cat khoang lang ra khoi x.
% n là độ dài frame, es là ngưỡng năng lượng.
x = x - mean(x);  %dk: x đã được chuẩn hóa
if nargin < 3
      es = 2E-3;  %mặc định là 2e-3
end;
```

```
if nargin < 2
      n = 128;    %mặc định là 128 mẫu
end;

y=[]; i=1;
while i<=length(x)-n
      t=x(i:i+n-1);
      e=mean(t.^2);
      if (e>es)
            y = [y t];
      end;
      i=i+n;
end;
```



Hình 6.2: Giải thuật cắt khoản lặng trong file wav

Mỗi file âm thanhđược trích chọn đặc trưng MFCC thành một dãy các vector MFCC bằng hàm wave2mfcc:

```
function mfcc = wave2mfcc(wav, fs, p);

if nargin < 3 % mặc định lấy vector MFCC 8pt
```

```
      p = 8;
end;

if nargin < 2 % mặc định tần số lấy mẫu = 8kHz
      fs = 8000;
end;

if isstr(wav) % nếu wav là tên file thì đọc
[wav fs] = wavread(wav);
end;

% chuẩn hoá để max(wav)=1.
mx = max(wav);
wav = wav ./ mx;

% tính vector MFCC p phần tử, gồm cả năng lượng
mfcc = melcepst(wav,fs,'e',p-1);
```

Vì các file wav có độ dài ngắn khác nhau nên dãy các vector đặc trưng MFCC tương ứng cũng không có cùng số phần tử. Nhưng đầu vào của mạng nơron MLP lại phải cố định. Do đó chúng tôi thực hiện một biện pháp đơn giản là *chia dãy đặc trưng thành 5 phần đều nhau, tính trung bình của từng phần được 5 vector rồi ghép lại thành một vector. Kết quả đầu vào của mạng nơron là một vector 8x5=40 thành phần.*

```
function x = VecAvr(ft,k);
% trích ft thành k phần chia trung bình.
n = length(ft); m = floor(n/k); x=[];
i=0;t=1;

while i<k
      i=i+1;
      f=sum(ft(t:t+m-1,:))./m;
      x=[x f];
      t=t+m;
end;
```

Đối với bộ từ vựng cũng thực hiện tương tự các bước trên, ta có được bộ dữ liệu dùng để huấn luyện mạng neural.

Đến lúc này, việc chuẩn bị các dữ liệu đầu vào cho mạng neural đã xong, chúng ta sẽ đi xây dựng mạng neural MLP 3 lớp dùng để nhận dạng. Ta dùng lệnh mlp để xây dựng:

```
net = mlp(inode, hnode, onode, func, anpha);
      %inode, hnode, onode: so neural vào, ẩn, ra
      %func: hàm kích hoạt
      %alpha: ngưỡng của giá trị trọng số.
```

Chúng ta huấn luyện mạng bằng lệnh :

```
[net err]=mlptrain(net, xtrain, target, loop);
            %net: mang mlp; xtrain: dữ liệu đầu vào
            %taget: dữ liệu ra cần đạt đến
            %err: độ sai khác giữa xtrain và target
```

Dữ liệu ra target được xây dựng rất đơn giản, nó là một vactor 11 phần tử. Vector này có đặc điểm là phần tử thứ i tương ứng với số i cần nhận dạng bằng 1, các phần tử còn lại bằng 0. Ví dụ target dùng để huấn luyện cho phát âm "số một" thì sẽ có dạng [1 0 0 0 0 0 0 0 0 0 0], target dùng để huấn luyện cho phát âm "số hai" thì sẽ có dạng [0 1 0 0 0 0 0 0 0 0 0].... target dùng để huấn luyện cho phát âm "số không" thì sẽ có dạng [0 0 0 0 0 0 0 0 0 0 1].

Nhận dạng thông qua hàm:

```
ytest = mlpfwd(net, xtest);
            %net: mang mlp; xtest: dữ liệu cần nhận dạng
            %ytest: dữ liệu ra
```

Xtest sẽ là vector đặc trưng MFCC gồm 40 phần tử đã tính ở trên, ytest là một vector 11 phần tử. Nếu quá trình huấn luyện tốt, phần tử thứ i của ytest tương ứng với chữ số i cần nhận dạng sẽ có giá trị lớn nhất.

6.2 Chương trình nhận dạng phát âm mười chữ số tiếng Việt

Giao diện chính của chương trình như sau:

Hình 6.2.1: Giao diện chính của chương trình

Trước tiên ta phải tạo một mạng neural MLP 3 lớp với đầy đủ các thông số: số neural lớp vào, số neural lớp ẩn, số eural lớp ra, hàm kích hoạt, ngưỡng giá trị trọng số. Giao diện của chương trình tạo neural như hình 6.2.2. Mạng MLP sau khi tạo được mô tả như sau:

```
net =

       type: 'mlp'
        nin: 40
    nhidden: 100
       nout: 11
       nwts: 5211
      outfn: 'logistic'
      alpha: 0.0100
         w1: [40x100 double]
         b1: [1x100 double]
         w2: [100x11 double]
         b2: [0.1124 -0.1461 -0.0608 0.0194 -0.0348 0.0377
0.0656 -0.0049 0.0018 0.0889 -0.1626]
```

Sau đó, để mạng có thể nhận dạng được phát âm các chữ số tiếng Việt ta cần phải huấn luyện cho mạng. Dữ liệu dùng để huấn luyện là file .wav đã được thu âm sẵn. Trong chương trình huấn luyện này bao gồm cả việc tiền xử lý như: cắt các khoản lặng trong file, trích đặc trưng MFCC để đưa vào mạng... Giao diện chương trình huấn luyện như hình 6.2.3.

Hình 6.2.2: Tạo mạng neural MLP 3 lớp

Hình 6.2.3: Chương trình huấn luyện mạng MLP

Sau khi huấn luyện, chúng ta có thể nhận dạng các phát âm chữ số tiếng Việt. Chúng ta có thể nhận dạng trực tiếp từ micro hay từ file.wav. Nhìn chung, khi nhận dạng từ file .wav cho kết quả chính xác hơn nhận dạng trực tiếp, đó là do các giải thuật tiền xử lý chưa được tốt. Hình 6.2.4 mô tả chương trình nhận dạng từ file wav, hình 6.2.5 mô tả chương trình nhận dạng trực tiếp từ micro.

Hình 6.2.4: Chương trình nhận dạng từ file

Hình 6.2.5: Chương trình nhận dạng trực tiếp từ micro

- hết Chương 6 -

Nội dung của đồ án:

PHẦN I: LÝ THUYẾT

Chương 1: Mở đầu

Chương 2: Lý thuyết âm thanh và tiếng nói
- 2.1 Nguồn gốc âm thanh
- 2.2 Các đại lượng đặc trưng cho âm thanh
- 2.3 Các tần số cơ bản của tiếng nói
- 2.4 Cơ chế tạo lập tiếng nói con người.
- 2.5 Mô hình lọc nguồn tạo tiếng nói.
- 2.6 Hệ thống nghe của người.
- 2.7 Quá trình sản xuất và thu nhận tiếng nói của con người.
- 2.8 Các âm thanh tiếng nói và các đặc trưng
  - 2.8.1 Nguyên âm
  - 2.8.2 Các âm vị khác

Chương 3: Lý thuyết nhận dạng tiếng nói
- 3.1 Tổng quan về nhận dạng tiếng nói
- 3.2 Các nguyên tắc cơ bản trong nhận dạng tiếng nói.
- 3.3 Các hệ thống nhận dạng tiếng nói.
- 3.4 Các quá trình nhận dạng tiếng nói.
  - 3.4.1 Phân tích các đặc tính tiếng nói
  - 3.4.2 Phân lớp mẫu
  - 3.4.3 Xử lý ngôn ngữ
- 3.5 Các tiếp cận nhận dạng tiếng nói
  - 3.5.1 Tiếp cận âm thanh-ngữ âm
  - 3.5.2 Tiếp cận nhận dạng mẫu
  - 3.5.3 Tiếp cận trí tuệ nhân tạo
- 3.6 Các phương pháp nhận dạng tiếng nói.
  - 3.6.1 Mô hình Fujisaki
  - 3.6.2 Mô hình Markvo ẩn
  - 3.6.3 Mô hình mạng neuron
- 3.7 Những thuận lợi và khó khăn trong nhận dạng tiếng Việt

Chương 4: Mạng neuron và ứng dụng trong nhận dạng tiếng nói
- 4.1 Định nghĩa mạng neuron
- 4.2 Kiến trúc mạng neuron
- 4.3 Đặc trưng của mạng neuron

PHẦN II: MÔ PHỎNG NHẬN DẠNG TIẾNG NÓI BẰNG MẠNG NEURON TIẾN ĐA LỚP BẰNG MATLAB

Chương 5: Giới thiệu các hàm và toolbox trong matlab
`

Chương 6: Xây dựng chương trình nhận dạng phát âm mười chữ số tiếng Việt bằng các toobox của matlab

PHẦN III: KẾT LUẬN VÀ TÀI LIỆU THAM KHẢO