NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT LIÊN TỤC BẰNG MẠNG NƠ-RON

Lê Tiến Thường, Trần Tiến Đức
Trường Đại học Bách khoa – ĐHQG-HCM
(Bài nhận ngày 02 tháng 5 năm 2002, hoàn chỉnh sửa chữa ngày 18 tháng 6 năm 2002)

TÓM TẮT: *Bài báo này trình bày phương pháp tổ chức nhận dạng tiếng nói tiếng Việt liên tục bằng mạng nơ-ron. Qua phân tích ngữ âm tiếng Việt, ta xác định được các âm vị phụ thuộc cần nhận dạng với bộ từ vựng cho trước, nghĩa là một âm vị được phân lớp khác nhau phụ thuộc vào các âm vị xung quanh nó. Phần trích đặc điểm tiếng nói sử dụng mô hình thông dụng hiện nay là mel-cepstrum, ở đó phổ tiếng nói được biến đổi theo thang tần số mel rồi chuyển log phổ công suất sang miền cepstrum bằng biến đổi cosine rời rạc ngược. Mạng nơ-ron dùng để ước lượng xác suất âm vị ở đầu ra có giá trị giữa 0 và 1. Xác suất âm vị của các frame liên tiếp được bố trí trong một ma trận rồi dùng thuật toán Viterbi để tìm chuỗi âm vị hợp lệ có điểm cao nhất thông qua ma trận trên, đó cũng chính là từ cần nhận dạng. Chương trình thử nghiệm được viết bằng Microsoft Visual C++ 6.0 có kết quả chính xác cao đã khẳng định khả năng ứng dụng của mạng nơ-ron trong nhận dạng tiếng nói tiếng Việt.*

Từ khóa: *nhận dạng tiếng nói, âm vị phụ thuộc, mel-cepstrum, mạng nơ-ron, thuật toán Viterbi.*

I. PHÂN TÍCH NGỮ ÂM TIẾNG VIỆT

Trong các hệ nhận dạng tiếng nói dựa trên phương pháp thống kê như Mô hình Markov ẩn (Hidden Markov Models), Mạng Nơ-ron (Neural Networks), hay lai giữa hai phương pháp trên người ta thường sử dụng mô hình âm vị phụ thuộc [3][5][8]. Xét hai từ: sáu [Saw] và tám [tam], ta nhận thấy âm vị [a] trong từ "sáu" khác với âm vị [a] trong từ "tám". Nửa trái của [a] trong "sáu" bị ảnh hưởng bởi âm vị [S], nửa phải bị ảnh hưởng bởi [w]; còn nửa trái của [a] trong "tám" bị ảnh hưởng bởi [t], nửa phải bị ảnh hưởng bởi [m]; như vậy có bốn âm vị là [S<a], [a>w], [t<a], [a>m]. Người ta gọi âm vị bị ảnh hưởng bởi hai âm vị trước và sau là âm vị phụ thuộc. Quá trình thử nghiệm được tiến hành trên bộ từ vựng 10 chữ số phát âm liên tục gồm: không [XoN], một [mot], hai [haj], ba [bA], bốn [bon], năm [naxm], sáu [Saw], bảy [baj], tám [tam], chín [cin]. Ta phân tích ngữ âm cho trường hợp này. Mỗi âm vị được chia thành một, hai hay ba phần phụ thuộc vào cách phát âm của từ tương ứng và các từ trước sau. Nếu chia âm vị thành một phần thì âm vị đó là âm vị độc lập, không bị ảnh hưởng bởi hai âm vị trước sau. Nếu chia âm vị thành hai phần thì nửa trái của âm vị bị ảnh hưởng bởi âm vị trước, nửa phải bị ảnh hưởng bởi âm vị sau. Nếu chia âm vị thành ba phần thì phần đầu bị ảnh hưởng bởi âm vị trước, phần giữa độc lập, giữ nguyên bản sắc của âm vị đó, phần cuối bị ảnh hưởng bởi âm vị sau. Ngoài ra còn có những âm vị được chia thành một phần nhưng nó lại bị ảnh hưởng bởi âm vị phía sau, gọi là âm vị phụ thuộc phải, thường là âm tắc như [t]. Cụ thể phân tích từ không [XoN] nói liên tục gồm XoN, XoNXon, XoNmot, XoNhaj, XoNbA, XoNbon, XoNnaxm, XoNSaw, XoNbaj, XoNtam, XoNcin. Xét âm vị N, ta có các

âm vị phụ thuộc sau: o<N, N>.pau, N>X, N>m, N>h, N>b, N>n, N>S, N>t, N>c. Phân tích cho các từ còn lại, ta được 128 âm vị phụ thuộc trình bày ở **Bảng 1.**

o<N	N>$sil	N>X	N>$nasal	N>h	N>b	N>S	N>t
N>c	t>$sil	t>X	t>$nasal	t>h	t>b	t>S	t>t
t>a	t>c	a<j	j>$sil	j>X	j>$nasal	j>h	j>b
j>S	j>t	j>c	b<A	<A>	A>$sil	A>X	A>$nasal
A>h	A>b	A>S	A>t	A>c	$sil<n	$nasal<n	t<n
j<n	A<n	i<n	o<n	w<n	n>$sil	n>X	n>$nasal
n>h	n>b	n>ax	n>S	n>t	n>c	$sil<m	$nasal<m
t<m	j<m	A<m	ax<m	a<m	w<m	m>$sil	m>X
m>$nasal	m>h	m>b	m>o	m>S	m>t	m>c	a<w
w>$sil	w>X	w>$nasal	w>h	w>b	w>S	w>t	w>c
X<o	$nasal<o	b<o	o>$nasal	o>t	h<a	S<a	b<a
t<a	a>j	a>$nasal	a>w	$nasal<ax	ax>$nasal	c<i	i>$nasal
$sil<X	$nasal<X	t<X	j<X	A<X	w<X	X>o	$sil<h
$nasal<h	t<h	j<h	A<h	w<h	h>a	$sil<b	$nasal<b
t<b	j<b	A<b	w<b	b>A	b>o	b>a	$sil<S
$nasal<S	t<S	j<S	A<S	w<S	S>a	<c>	<.pau>

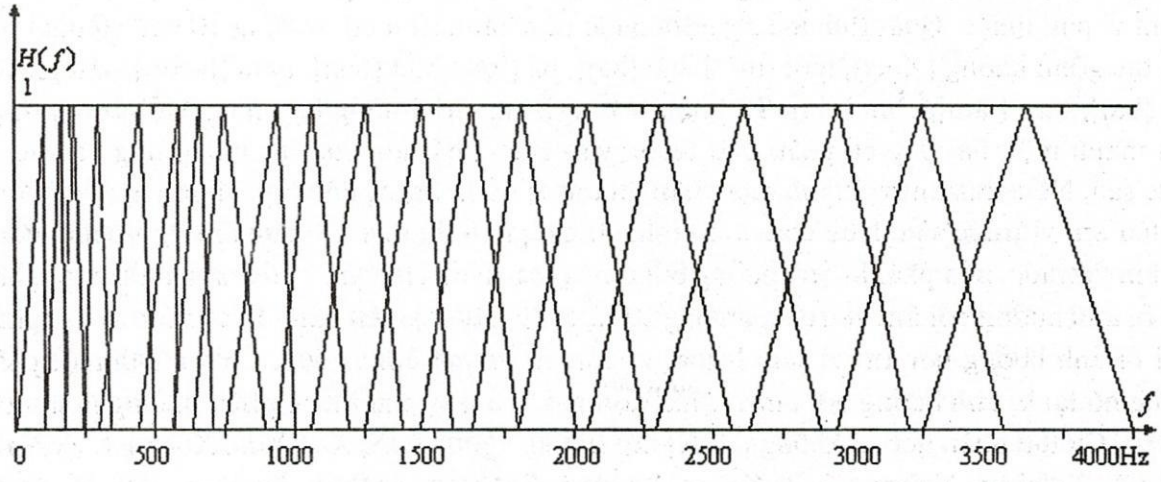
Bảng 1 Các âm vị phụ thuộc

Nhằm giảm bớt số lượng các âm vị phụ thuộc, ta nhóm các âm vị có cấu âm tương tự nhau thành một nhóm. Với bộ từ vựng trên, chúng tôi chỉ sử dụng hai nhóm là $nasal [m n N] và $sil [.pau .garbarge].

II. TRÍCH ĐẶC ĐIỂM TIẾNG NÓI

Trong các lĩnh vực xử lý tiếng nói như nhận dạng, tổng hợp, mã hóa đều cần phải trích đặc điểm tiếng nói. Phần này trình bày phương pháp phổ biến để trích đặc điểm của tiếng nói là phân tích hệ số cepstrum dùng thang tần số mel (Mel Frequency Cepstral Coefficents - MFCC)[5][6]. Theo các nghiên cứu về sự phân tích tần số của tai người thì đáp tuyến này là phi tuyến, nghĩa là độ nhạy của tai người sẽ thay đổi khác nhau với những tần số khác nhau, điều này dẫn đến một đơn vị đo mới là mel, trong đó 1000 Hz là 1000 mel. Mối liên hệ giữa thang mel và thang Hz là:

$$f_{mel}(f) = 2595\log_{10}(1+\frac{f}{700}) \qquad (2.1)$$



Hình 1 Dãy 20 bộ lọc tam giác có khoảng cách 100mel và tần số lấy mẫu 8kHz

Quá trình trích đặc điểm của một frame tiếng nói gồm các bước sau:

Log công suất ngắn hạn: Công suất ngắn hạn P của một frame tiếng nói là:

$$P = \frac{1}{N}\sum_{n=1}^{N-1}[x(n)]^2 \qquad (2.2)$$

trong đó N là chiều dài frame và log công suất ngắn hạn là $\log(P)$.

Pre-emphasis: Khi phát âm, tiếng nói bị suy hao nên cần phải làm rõ bằng bộ lọc thông cao có phương trình sai phân bậc một là:

$$y(n) = x(n) - ax(n-1) \qquad (2.3)$$

với a là hằng số thường được chọn giữa 0.9 và 1.

Cửa sổ Hamming: Trước khi chuyển một frame tiếng nói sang miền tần số bằng phép biến đổi Fourier thời gian ngắn (Short Time Fourier Transform - STFT), người ta nhân frame đó với hàm cửa sổ, mục đích là tối thiểu hóa sai số do tín hiệu không liên tục ở đầu và cuối mỗi frame. Hàm cửa sổ thông dụng nhất là Hamming:

$$W(n) = \{0.54 - 0.46\cos(\frac{2\pi n}{N-1})\} \text{ nếu } 0 \le n \le N-1 \qquad (2.4)$$

$$= \quad 0 \quad \text{nếu ngược lại}$$

Biến đổi Fourier thời gian ngắn: Biến đổi Fourier thời gian ngắn của một frame là:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} \qquad (2.5)$$

Nếu tính trực tiếp sẽ không hiệu quả vì phải thực hiện một khối lượng tính toán lớn. Để giảm bớt độ phức tạp tính toán, người ta thường dùng biến đổi Fourier nhanh (Fast Fourier Transform - FFT).

Mel-cepstrum: Tín hiệu sau khi chuyển sang miền tần số được đưa đến dãy bộ lọc theo thang mel ở **Hình 1**. Do đáp tuyến tần số của mỗi bộ lọc là tam giác nên ta xác định được $H_i(w)$ là đáp tuyến tần số của bộ lọc thứ i. Gọi ngõ ra của từng bộ lọc là Y_i

$$Y_i = \log(\frac{1}{b_i}\sum_{b_i}|S_i(w)|^2 H_i(w)) \text{ với } i = 1..N_{filter} \qquad (2.6)$$

với $S_i(w)$ là phổ và b_i là số lượng phổ của bộ lọc thứ i. Biến đổi cosine rời rạc ngược để thu được K hệ số cepstrum với K được chọn là 12.

$$c_k = \frac{1}{N_{filter}}\sum_{i=1}^{N_{filter}} Y_i \cos(\frac{k\pi}{N_{filter}}(i-0.5)) \text{ với } k = 1..K \qquad (2.7)$$

Đạo hàm các hệ số cepstrum: Để cải thiện hơn nữa biểu diễn tính chất phổ, người ta lấy thêm đạo hàm các hệ số cepstrum theo thời gian t là:

$$\frac{\partial c_m(t)}{\partial t} = \Delta c_m(t) \approx \mu\sum_{k=-K}^{K} k c_m(t+k) \qquad (2.8)$$

ở đó μ là hằng số chuẩn hóa thích hợp và $(2K+1)$ là số lượng frame cần tính. Điển hình $\mu = 0.5$ và $K = 2$ là giá trị thích hợp nhất để tính đạo hàm cấp một.

Đạo hàm log công suất ngắn hạn: Tương tự như đạo hàm các hệ số cepstrum, đạo hàm log công suất theo t là:

$$\frac{\partial \log P_m(t)}{\partial t} = \Delta \log P_m(t) \approx \mu\sum_{k=-K}^{K} k \log P_m(t+k) \qquad (2.9)$$

có μ và K được chọn giống trên.

III. MẠNG NƠ-RON

Mạng nơ-ron gồm một tập các phần tử xử lý phi tuyến gọi là các nơ-ron hay nút mạng hoạt động song song. Do khả năng xử lý song song, chịu lỗi, học mẫu và tính toán phi tuyến nên mạng nơ-ron được ứng dụng rộng rãi trong các lĩnh vực như phân lớp, tối ưu, xấp xỉ hàm... [4][8][9]

Giả sử mạng nơ-ron gồm N đầu vào, ký hiệu là $x_1, x_2, ..., x_N$ có các trọng số tương ứng $w_1, w_2, ..., w_N$ kết nối với đầu ra y thì y được tính phi tuyến như sau:

$$y = f(\sum_{i=1}^{N} w_i x_i - \phi) \tag{3.1}$$

trong đó ϕ là ngưỡng cục bộ và f là một hàm phi tuyến. Hàm phi tuyến thường dùng là hàm sigmoid:

$$y = \frac{1}{1+e^{-x}} \tag{3.2}$$

do $y' = y(1-y)$ dễ tính, có đầu ra không âm và nhỏ hơn 1. Để xác định hoàn toàn một mạng nơ-ron, phải xác định các giá trị trọng số và ngưỡng của mỗi phần tử tính toán dựa trên tập mẫu. Thông qua tập mẫu, ta xác định quan hệ giữa tập Q vector đầu vào $x_1, x_2, ..., x_Q$ và tập Q vector đầu ra $y_1, y_2, ..., y_Q$, trong đó ứng với x_1 ta có y_1, x_2 ta có y_2, ..., x_Q ta có y_Q. Qua một số vòng lặp, quá trình học sẽ hội tụ và cho ra một tập các thông số tối ưu.

Cấu hình của mạng perceptron n lớp $(n > 2)$ gồm: lớp thứ nhất là lớp đầu vào, lớp thứ n là lớp đầu ra, và $(n-2)$ lớp ẩn. Mỗi nút của lớp thứ i $(0 < i < n)$ liên kết với mọi nút ở lớp thứ $(i+1)$, và các nút trong cùng lớp không liên kết với nhau. Thuật toán truyền ngược sai số của quá trình học cho phép xác định một mạng perceptron. Nguyên lý của thuật toán là giảm gradient hàm sai số E. Vấn đề hội tụ của mạng hiện chưa có chứng minh lý thuyết nào chỉ ra cấu trúc mạng thích hợp cho một bài toán cụ thể, nên hầu hết đều dùng phương pháp thử-sai. Chia tập mẫu thành hai tập mẫu con: tập mẫu học và tập mẫu kiểm tra. Các mạng có số nút ẩn khác nhau được học với tập mẫu học và được kiểm tra tính tổng quát hóa trên tập mẫu kiểm tra. Theo dõi sai số trên tập mẫu kiểm tra, ta sẽ chọn được số nút ẩn đủ dùng [9].

IV. TÌM KIẾM VITERBI

Sau khi đã có mạng nơ-ron, với tiếng nói bất kỳ nằm trong bộ từ vựng, ta chia thành nhiều frame, mỗi frame khoảng 10ms rồi trích 12 hệ số mel-cepstrum, 12 giá trị đạo hàm của 12 hệ số mel-ceptrum, log công suất và đạo hàm log công suất, nghĩa là mỗi frame trích được 12+12+2=26 hệ số. Do phổ tiếng nói không chỉ phụ thuộc vào thời điểm hiện tại, mà còn phụ thuộc vào sự thay đổi phổ theo thời gian nên ngoài frame hiện tại ta chọn thêm 4 frame nữa ở các thời điểm -60, -30, 30, 60ms so với frame hiện tại. Mỗi frame trích được 26 hệ số, nên 5 frame trích được 26x5=130 hệ số hay vector đặc điểm của mỗi âm vị phụ thuộc có 130 hệ số. Mạng nơ-ron là mạng perceptron ba lớp có 130 nút ở lớp đầu vào tương ứng với vector đặc điểm của một âm vị phụ thuộc, 200 nút ở lớp ẩn và 128 nút đầu ra tương ứng với 128 âm vị phụ thuộc cần phân lớp.

Gọi T là số lượng frame của tiếng nói, $N = 128$ là số lượng âm vị. Với mỗi frame t, hay tại thời điểm t, $1 \le t \le T$, ta có được vector đặc điểm gồm 130 hệ số. Mạng nơ-ron sẽ ước lượng xác suất âm vị cho vector đặc điểm này, chú ý hàm truyền sigmoid không ràng buộc tổng xác suất các âm vị phải bằng 1. Tuy nhiên đối với các ứng dụng như so sánh hay kết nối ước

lượng xác suất của các mạng với nhau đòi hỏi tổng xác suất phải bằng 1. Để đạt được điều này, ta có thể chuẩn hóa hàm truyền đầu ra bằng hàm mũ như softmax [11]. Ký hiệu xác suất tạo ra âm vị thứ j của frame t là $b_j(t)$, trong đó $0 \le b_j(t) \le 1$ và $1 \le j \le N$, toàn bộ T frame sẽ tạo ra ma trận xác suất âm vị $B = \{b_j(t)\}$, mục tiêu đặt ra là tìm trong ma trận B từ tốt nhất. Muốn vậy, ta phải xác định tập các chuỗi âm vị hợp lệ, điều này phụ thuộc vào số lượng từ cần nhận dạng, cách phát âm của từng từ và trật tự kết hợp (văn phạm) của các từ đó. Chẳng hạn, cần nhận dạng hai từ: "không" có phát âm là [XoN] và "một" có phát âm là [mot] nói liên tục dùng mô hình âm vị độc lập, trật tự kết hợp ngẫu nhiên. Ta nhận thấy có ba chuỗi hợp lệ là chuỗi 1 bắt đầu từ âm vị X, chuỗi 2 bắt đầu từ âm vị m, chuỗi 3 bắt đầu từ âm vị <.pau>. Trong quá trình tìm trên ba chuỗi này, ở thời điểm đang xét, ta sẽ chuyển đến âm vị mới nếu xác suất âm vị mới lớn hơn xác suất âm vị hiện tại. Để thuận tiện cho việc trình bày thuật toán Viterbi, ta công thức hóa điều này. Gọi $\pi = \{\pi_j\}$ là ma trận xác suất âm vị ban đầu với π_j là xác suất âm vị j tại thời điểm $t = 1$, $\pi_j = 1$ nếu âm vị j hợp lệ, ngược lại $\pi_j = 0$ nếu âm vị j không hợp lệ. Ký hiệu âm vị ở thời điểm t là q_t; $A_t = \{a_{ij}(t)\}$ là ma trận xác suất chuyển âm vị ở thời điểm t, với $a_{ij}(t)$ là xác suất chuyển âm vị i ở thời điểm t-1 đến âm vị j ở thời điểm t, $a_{ij}(t) = 1$ nếu $j = \arg\max_{1 \le k \le N}[b_k(t)]$. trọng đó chuỗi âm vị $q_{t-1} = i$ và $q_t = k$ hợp lệ, ngược lại $a_{ij}(t) = 0$ nếu chuỗi âm vị $q_{t-1} = i$ và $q_t = k$ không hợp lệ. Như vậy tại mỗi thời điểm t, ta có một ma trận A_t với $a_{ij}(t)$ chỉ có hai giá trị là 0 và 1. Phần **Ví dụ** phía sau sẽ minh họa rõ những điều vừa thảo luận. Áp dụng thuật toán Viterbi [3][5] để tìm chuỗi âm vị tốt nhất $\mathbf{q} = (q_1 q_2 \ldots q_T)$ ứng với ma trận xác suất âm vị B đã cho. Gọi

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i \mid B] \qquad (4.1)$$

nghĩa là $\delta_t(i)$ có xác suất lớn nhất tính từ thời điểm ban đầu đến thời điểm t kết thúc ở âm vị i của chuỗi âm vị trên. Qui nạp

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}(t+1)] b_j(t+1) \qquad (4.2)$$

Muốn xác định chuỗi âm vị, ta sử dụng mảng $\psi_t(j)$ để lưu lại đối số làm cho Phương trình (4.2) cực đại ở từng thời điểm t và j. Thuật toán tìm chuỗi âm vị tốt nhất như sau:

1. Khởi tạo

$$\delta_1(j) = \pi_j b_j(1) \qquad 1 \le j \le N \qquad (4.3a)$$

$$\psi_1(j) = 0 \qquad (4.3b)$$

2. Qui nạp

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}(t)] b_j(t) \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \qquad (4.4a)$$

$$\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}(t)] \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \qquad (4.4b)$$

3. Kết thúc

$$P^* = \max_{1 \le i \le N}[\delta_T(i)] \qquad (4.5a)$$

$$q_T^* = \arg\max_{1 \le i \le N}[\delta_T(i)] \qquad (4.5b)$$

4. Lần ngược con đường (chuỗi âm vị)

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, ..., 1 \qquad (4.6)$$

Bằng cách lấy log các tham số, ta có thể tổ chức thuật toán Viterbi mà không cần phép nhân nào đồng thời tránh tràn do tích quá nhiều đại lượng nhỏ hơn 1.

Ví dụ: Để minh họa quá trình tìm kiếm bằng thuật toán Viterbi, xét bài toán nhận dạng hai từ: "không" [XoN] và "một" [mot] nói liên tục gồm 6 âm vị độc lập theo thứ tự âm vị 1 là X, âm vị 2 là N, âm vị 3 là m, âm vị 4 là o, âm vị 5 là t, âm vị 6 là <.pau>. Mạng nơ-ron có 6 nút ở lớp đầu ra. Cho tiếng nói bất kỳ $T = 10$ frame, thông qua mạng nơ-ron ta được ma trận xác suất âm vị B là:

X	0.1	0.1	0.2	0.1	0.3	0.1	0.1	0.1	0.1	0.1
N	0.2	0.3	0.3	0.4	0.2	0.2	0.2	0.2	0.1	0.1
m	0.2	0.7	0.8	0.8	0.6	0.2	0.1	0.1	0.1	0.1
o	0.3	0.2	0.1	0.1	0.9	0.8	0.5	0.4	0.2	0.1
t	0.4	0.3	0.2	0.1	0.2	0.2	0.7	0.8	0.4	0.3
<.pau>	0.8	0.9	0.3	0.2	0.1	0.1	0.1	0.1	0.8	0.9

Bước 1: Khởi tạo

$$\delta_1(j) = \pi_j b_j(1) \qquad 1 \le j \le N$$

$$\psi_1(j) = 0$$

Tại $t = 1$, âm vị bắt đầu luôn luôn là X, m, <.pau> nên $\pi = [1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]^T$, ký hiệu chỉ số trên T là ma trận chuyển vị. Như vậy có tất cả 3 chuỗi hợp lệ: chuỗi 1 bắt đầu ở âm vị X, chuỗi 2 ở âm vị m, chuỗi 3 ở âm vị <.pau>

$\delta_1(1)$ = (1)(0.1)=0.1

$\delta_1(2)$ = (0)(0.2)=0

$\delta_1(3)$ = (1)(0.2)=0.2

$\delta_1(4)$ = (0)(0.3)=0

$\delta_1(5)$ = (0)(0.4)=0

$\delta_1(6)$ = (1)(0.8)=0.8

$\psi_1(1)$ = 0, $\psi_1(2)$=0, $\psi_1(3)$=0, $\psi_1(4)$=0, $\psi_1(5)$=0, $\psi_1(6)$=0

Bước 2: Qui nạp

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i)a_{ij}(t)]b_j(t) \qquad 2 \le t \le T, \quad 1 \le j \le N$$

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

* Đối với chuỗi 1, do tại $t = 1$ là âm vị X nên tại $t = 2$ âm vị hợp lệ là X, o. Bởi vì $b_o(2) = 0.2$ lớn hơn $b_X(2) = 0.1$ nên $a_{14}(2) = 1$, các $a_{1j}(2)$ còn lại đều bằng 0.

Đối với chuỗi 2, do tại $t = 1$ là âm vị m nên tại $t = 2$ âm vị hợp lệ là m, o. Bởi vì $b_m(2) = 0.7$ lớn hơn $b_o(2) = 0.2$ nên $a_{33}(2) = 1$, các $a_{3j}(2)$ còn lại đều bằng 0.

Đối với chuỗi 3, do tại $t = 1$ là âm vị <.pau> nên tại $t = 2$ âm vị hợp lệ là <.pau>, X, m. Bởi vì $b_{<.pau>}(2) = 0.9$ lớn hơn $b_X(2) = 0.1$ và $b_m(2) = 0.7$ nên $a_{66}(2) = 1$, các $a_{6j}(2)$ còn lại đều bằng 0.

Do không tồn tại những chuỗi khác nên các $a_{2j}(2)$, $a_{4j}(2)$, $a_{5j}(2)$ đều bằng 0.

$\delta_2(1)$ = max[(0.1)(0) (0)(0) (0.2)(0) (0)(0) (0)(0) (0.8)(0)](0.1) = 0

$\delta_2(2)$ = max[(0.1)(0) (0)(0) (0.2)(0) (0)(0) (0)(0) (0.8)(0)](0.3) = 0

$\delta_2(3)$ = max[(0.1)(0) (0)(0) (0.2)(1) (0)(0) (0)(0) (0.8)(0)](0.7) = 0.14

$\delta_2(4)$ = max[(0.1)(1) (0)(0) (0.2)(0) (0)(0) (0)(0) (0.8)(0)](0.2) = 0.02

$\delta_2(5)$ = max[(0.1)(0) (0)(0) (0.2)(0) (0)(0) (0)(0) (0.8)(0)](0.3) = 0

$\delta_2(6)$ = max[(0.1)(0) (0)(0) (0.2)(0) (0)(0) (0)(0) (0.8)(1)](0.9) = 0.72

$\psi_2(1)$ = 0, $\psi_2(2)$ = 0, $\psi_2(3)$ =3, $\psi_2(4)$ =1, $\psi_2(5)$ =0, $\psi_2(6)$ =6

* Đối với chuỗi 1, do tại $t = 2$ là âm vị o nên tại $t = 3$ âm vị hợp lệ là o, N. Bởi vì $b_N(3) = 0.3$ lớn hơn $b_o(3) = 0.1$ nên $a_{42}(3) = 1$, các $a_{4j}(3)$ còn lại đều bằng 0.

Đối với chuỗi 2, do tại $t = 2$ là âm vị m nên tại $t = 3$ âm vị hợp lệ là m, o. Bởi vì $b_m(3) = 0.8$ lớn hơn $b_o(3) = 0.1$ nên $a_{33}(3) = 1$, các $a_{3j}(3)$ còn lại đều bằng 0.

Đối với chuỗi 3, do tại $t = 2$ là âm vị <.pau> nên tại $t = 3$ âm vị hợp lệ là <.pau>, X, m. Bởi vì $b_m(3) = 0.8$ lớn hơn $b_X(3) = 0.2$ và $b_{<.pau>}(3) = 0.3$ nên $a_{63}(3) = 1$, các $a_{6j}(3)$ còn lại đều bằng 0.

Do không tồn tại những chuỗi khác nên các $a_{1j}(3)$, $a_{2j}(3)$, $a_{5j}(3)$ đều bằng 0.

Ta nhận thấy, tại $t = 3$, chuỗi 2 có $a_{33}(3) = 1$ và chuỗi 3 có $a_{63}(3) = 1$, nghĩa là chuỗi 2 và chuỗi 3 nhập chung lại. Thuật toán Viterbi sẽ chọn chuỗi có điểm tích lũy cao nhất tại thời điểm này, đó là chuỗi 3. Lúc này chỉ còn hai chuỗi là 1 và 3.

$\delta_3(1)$ = max[(0)(0) (0)(0) (0.14)(0) (0.02)(0) (0)(0) (0.72)(0)](0.2) = 0

$\delta_3(2)$ = max[(0)(0) (0)(0) (0.14)(0) (0.02)(1) (0)(0) (0.72)(0)](0.3) = 0.006

$\delta_3(3)$ = max[(0)(0) (0)(0) (0.14)(1) (0.02)(0) (0)(0) (0.72)(1)](0.8) = 0.576

$\delta_3(4)$ = max[(0)(0) (0)(0) (0.14)(0) (0.02)(0) (0)(0) (0.72)(0)](0.1) = 0

$\delta_3(5)$ = max[(0)(0) (0)(0) (0.14)(0) (0.02)(0) (0)(0) (0.72)(0)](0.2) = 0

$\delta_3(6)$ = max[(0)(0) (0)(0) (0.14)(0) (0.02)(0) (0)(0) (0.72)(0)](0.3) = 0

$\psi_3(1)$ = 0, $\psi_3(2)$ =4, $\psi_3(3)$ =6, $\psi_3(4)$ =0, $\psi_3(5)$ =0, $\psi_3(6)$ =0

Lập luận tương tự cho các thời điểm t còn lại. Lưu ý do tại $t = 7$ âm vị là t và nói liên tục nên âm vị hợp lệ tại $t = 8$ là t, <.pau>, X, m. Cuối cùng lần ngược chuỗi, ta sẽ được chuỗi âm vị tốt nhất (có xác suất lớn nhất) là 6633445566 hay <.pau><.pau>mmoott<.pau><.pau>. Đó là từ "một".

Nếu sử dụng mô hình âm vị phụ thuộc thì số lượng âm vị sẽ nhiều hơn, còn thuật toán Viterbi vẫn không thay đổi. Cũng ví dụ trên, ta có 20 âm vị phụ thuộc là <.pau>, $sil<X, N<X, t<X, X>o, X<o, o>N, o<N, N>$sil, N>X, N>m, $sil<m, N<m, t<m, m>o, m<o, o>t, t>$sil, t>X, t>m. Tại $t = 1$, âm vị bắt đầu luôn luôn là <.pau>, $sil<X, N<X, t<X, $sil<m, N<m, t<m. Tại $t = 2$, nếu trước đó là các âm vị $sil<X, N<X, t<X, thì âm vị bây giờ chỉ có thể là chính các âm vị đó và X>o.

V. THỬ NGHIỆM VÀ KẾT LUẬN

Từ những điều đã trình bày, ta xây dựng hệ nhận dạng tiếng nói có tần số lấy mẫu 8kHz. Quá trình này gồm các bước sau:

1. Chia tiếng nói thành thành nhiều frame, mỗi frame chừng 10ms.
2. Trích vector đặc điểm của mỗi frame gồm phổ frame đó và một số frame lân cận.
3. Ước lượng xác suất âm vị cho vector đặc điểm của mỗi frame dùng mạng nơ-ron.
4. Căn cứ vào ma trận xác suất âm vị, cách phát âm từng từ và luật văn phạm của các từ cần nhận dạng rồi áp dụng thuật toán Viterbi để xác định từ phù hợp nhất.

Chương trình thử nghiệm được viết bằng *Microsoft Visual C++ 6.0* dựa trên CSLU Speech Toolkit: http://cslu.cse.ogi.edu phụ thuộc người nói cho kết quả nhanh, có độ chính xác từ là 99% và độ chính xác câu là 97% đã khẳng định khả năng ứng dụng của mạng nơ-ron trong lãnh vực rất phức tạp là nhận dạng tiếng nói.

VI. HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi chỉ quan tâm đến phương pháp nhận dạng tiếng nói tiếng Việt liên tục theo mô hình âm vị phụ thuộc mà chưa đề cập đến thanh điệu tiếng Việt là đường nét của tần số cơ bản. Các nghiên cứu gần đây[1][2] cho thấy phép biến đổi wavelet đã được ứng dụng thành công trong nhận dạng tiếng Việt. Vì vậy, hướng phát triển trong tương lai là thay thế phân tích MFCC dựa trên biến đổi Fourier và cosine bằng biến đổi wavelet để cải thiện độ chính xác khi nhận dạng.

CONTINUOUS VIETNAMESE SPEECH RECOGNITION USING NEURAL NETWORKS

Le Tien Thuong, Tran Tien Duc
University of Technology – VNU-HCM

ABSTRACT: This paper describes the method for creating neural networks based continuous Vietnamese speech recognizer. By Vietnamese phonetic analyzing, we can determine the context-dependent phonemes from a given vocabulary, which means that one phoneme is classified differently denpending on the phonemes that surround it. The feature extraction has used the current popular model as the mel-ceptrum, where the short-time spectrum is warped according to the mel scale, then direct transformation of the log power spectrum to the cepstral domain using an inverse discrete cosine transform. The neural netwoks has been applied to estimate context-dependent phoneme probabilities that outputs value in the range 0 to 1. The phoneme probabilities for the successive frames are arranged in a matrix. We then use the Viterbi algorithm to find the legal string of phonmes throught the matrix gives us highest score, that is also the target word. The experiments were programmed in Microsoft Visual C++ 6.0. The high accurate results confirmed applicable of the neural networks for Vietnamese speech recognition.

Keywords: *speech recognition, context dependent phoneme, mel-cepstrum, neural networks, Viterbi algorithm.*

TÀI LIỆU THAM KHẢO

[1] Thuong Le-Tien, "A study on the continuous wavelet transform for the Vietnamese speech processing", *Proceedings of the 97 international conference on natural information and intelligent information systems*, Vol. 2, New Zealand, 1997.

[2] T. Le-Tien *et al*, "Recognizing formants and pitch periods for Vietnamese speech based on the local modulus maxima in the wavelet domain", *Tạp chí Phát triển khoa học và công nghệ*, Đại học Quốc gia Tp. HCM, Số 1&2, Tập 4, 2001.

[3] Frederick Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, 1998.

[4] Yoshua Benglo, *Neural Networks for Speech and Sequence Recognition*, International Thomson Computer Press, 1995.

[5] L. Rabiner, B.-H. Juang, *Fundamentals of speech recognition*, PTR Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1993.

[6] John R. Deller, Jr., John G. Proakis, John H. L. Hansen, *Discrete-time processing of speech signals*, Macmillan, 1993.

[7] F. J. Owens, *Signal processing of speech*, Macmillan, London, 1993.

[8] John-Paul Hosom *et al*, "Speech Recognition Using Neural Networks", *Center for Spoken Language Understanding ở Oregon Graduate Institute of Science and Techonology*, 1999.

[9] Nguyễn Đình Thúc, *Mạng Nơ-ron - Phương pháp và Ứng dụng*, NXB Giáo dục, 2000.

[10] Nguyễn Thành Phúc, "Phương pháp nhận dạng lời Việt dùng mạng nơ-ron", *Tạp chí Khoa học & Công nghệ các trường đại học kỹ thuật*, Số 19+20, 1999.

[11] Steve Renals, Nelson Morgan, "Connectionist Probability Estimation in HMM Speech Recognition", *Internationnal Computer Science Institute*, Berkeley University, 1992