

Lời Mở Đầu

Nhận dạng chữ tay là một lĩnh vực riêng trong nhận dạng chữ viết đã được quan tâm nghiên cứu và ứng dụng từ nhiều năm nay. Về mặt lý thuyết, chưa có phương pháp nào hoàn chỉnh cho bài toán này do tính phức tạp, sự biến dạng của dữ liệu đầu vào.

Nhận dạng chữ viết tay với những mức độ ràng buộc khác nhau về cách viết, kiểu chữ..., phục vụ cho các ứng dụng và xử lý các chứng từ, hóa đơn, phiếu ghi, bản viết tay chương trình...

Nhận dạng chữ viết tay vẫn còn là vấn đề thách thức đối với các nhà nghiên cứu. Bài toán này chưa thể giải quyết trọn vẹn được vì nó hoàn toàn phụ thuộc vào người viết và sự biến đổi quá đa dạng trong cách viết và tình trạng sức khỏe, tinh thần của từng người viết.

Mục tiêu của bài tập nhằm giới thiệu một cách tiếp cận bài toán nhận dạng chữ viết tay với một số ràng buộc, nhằm từng bước đưa vào ứng dụng thực tiễn.

Mặc dù hết sức cố gắng, song do thời gian có hạn và những hạn chế kiến thức nên bài tập có thể còn thiếu sót, mong tiếp tục nhận được sự chỉ bảo của Cô và ý kiến đóng góp của các bạn sinh viên để bài tập được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Chương I : Lý Thuyết Xử Lý Ảnh Và Một Số Thuật Toán Tiền Xử Lý Ảnh

I. Lọc mịn ảnh:

Lọc mịn ảnh là một lọc thông thấp, giá trị của một điểm ảnh là trung bình trọng số của các điểm ảnh lân cận, hay giá trị điểm ảnh là kết quả của quá trình xoắn (convole) của các điểm ảnh lân cận với một nhân. Nhân có kích thước tùy ý 3x3, 5x5, kích thước nhân càng lớn thì càng nhiều điểm lân cận ảnh hưởng vào điểm ảnh kết quả. Ví dụ một số nhân lọc mịn ảnh như sau:

1	1	1
1	1	1
1	1	1

1	1	1
1	2	1
1	1	1

1	1	1
1	4	1
1	1	1

II. Nhị phân ảnh:

Nhị phân ảnh mức xám là tìm giá trị ngưỡng sao cho các điểm ảnh có giá trị lớn hơn ngưỡng được gọi là trắng(nền) và các điểm ảnh có giá trị nhỏ hơn ngưỡng được gọi là đen (đối tượng).

Tiêu chuẩn xác định ngưỡng thường sử dụng nhất là sử dụng sai số bình phương trung bình giữa giá trị mẫu v và mức tái thiết $r(v)$. (ký hiệu MSE)

Theo Otsu , giá trị ngưỡng được xác định như sau :

$$v_1^* = \arg \max \left\{ \frac{[\mu_T \cdot \varpi(v_1) - \mu(v_1)]^2}{\varpi(v_1)[1 - \varpi(v_1)]} \right\}$$

Trong đó :

$$\varpi(v_1) = \int_{v_0}^{v_1} p(v) dv$$

$$\mu(v_1) = \int_{v_0}^{v_1} vp(v) dv$$

Với $p(v)$ ước lượng từ histogram : $\mu_T = \mu(v_2 = v_{\max})$

$$V_0 = V_{\min}$$

v_1 : giá trị cần tìm

III) Tách Liên Thông :

Quét ảnh từ trái sang phải và từ trên xuống dưới, các pixel đen liên thông với nhau và được gán chung một nhãn, nếu gặp liên thông mới thì nhãn mới sẽ được gán :

Để minh họa ta có hình biểu diễn sau :

```

.....
. P P P.
. L ? . .
.....
    
```

Hình a: lân cận của "?" P= dòng trước; L=lân cận trái

```

.....
. * * * . * * * . . . .
. . * * * . * * * . . . .
. * * * . * * * . . . .
. * * * . * * * . . . .
. . . * * * * . . . .
. . . * * * * . . . .
. * * . . . . * * . .
. * * . . . . * * . .
. * * . . . . * * . .
.....
    
```

Hình b : Ảnh Ban Đầu

```

.....
. 1 1 1 1 . . 2 2 2 . . . .
. . 1 1 1 . . 2 2 2 2 . . . .
. 1 1 1 1 . 2 2 2 2 2 . . . .
. . . 1 1 ? * * . . . . . .
. . . * * * * * * . . . .
. * * . . . . * * . . . .
. * * . . . . * * . . . .
. * * . . . . * * . . . .
.....
    
```

Hình c : Tiến trình gán nhãn

```

.....
. 1 1 1 1 . . 2 2 2 . . . .
. . 1 1 1 . . 2 2 2 2 . . . .
. 1 1 1 1 . 2 2 2 2 2 . . . .
. . . 1 1 1 1 1 . . . . . .
. . . 1 1 1 1 1 1 . 3 . . . .
4 4 . . . . . . . 3 3 . . .
. . 4 4 . . . . . . 3 3 . .
. 4 4 . . . . . . . . . .
.....
    
```

Hình d : Sau khi quét đầy đủ

```

.....
. 1 1 1 1 . . 1 1 1 . . . .
. . 1 1 1 . . 1 1 1 1 . . . .
. 1 1 1 1 . 1 1 1 1 1 . . . .
. . . 1 1 1 1 1 . . . . . .
. . . 1 1 1 1 1 1 . 2 . . . .
3 3 . . . . . . . 2 2 . . .
. 3 3 . . . . . . . 2 2 . .
. 3 3 . . . . . . . . . .
.....
    
```

Hình e : Kết quả sau cùng

IV) ~~Chứng~~ ~~Nghiên~~ ~~g~~ : Biến đổi tuyến tính tọa độ điểm ảnh

$$a(x,y) = \tan^{-1} \left(\frac{G_x}{G_y} \right)$$

G_x, G_y là kết quả xoắn điểm ảnh với nhân S_x, S_y .

-1	0	1
-2	0	2

-1	0	1
----	---	---

S_x

α : là giá trị trung bình góc nghiêng của các điểm ảnh được xét

Ta có : $\dot{x} = x + tg(\alpha) \cdot y$

$\dot{y} = y$

-1	-2	-1
0	0	0
1	2	1

S_y

V. Chuẩn kích thước:

Chuẩn kích thước ảnh kí tự về một kích thước cố định và phóng sát bốn biên của ảnh.

Phóng ảnh là thực hiện phép biến đổi sau:

$$f_s(x, y) = f\left(\frac{x}{S_x}, \frac{y}{S_y}\right)$$

Với (x, y) là tọa độ điểm ảnh sau khi phóng và s_x, s_y là tỷ lệ phóng theo trục x và y tương ứng, $f_s(x,y)$ là giá trị điểm ảnh kết quả ứng với giá trị tọa độ (x, y) .

VI. Lấp khoảng trống ảnh bằng phép đóng morphology:

Sau khi phóng ảnh, ảnh có thể bị rời rạc, răng cưa biên. Để khắc phục tình trạng này ta dùng phép đóng để lấp các khoảng trống làm đầy ảnh..

Giả sử A, B là hai tập thuộc Z , phép đóng của A đối với B, ký hiệu $A \cdot B$ được định nghĩa:

$$A \cdot B = (A \oplus B) \ominus B$$

Tức phép đóng là phép do thực hiện phép mở rồi thực hiện phép đóng lên kết quả vừa có.

Phép đóng có tác dụng làm đầy những khoảng nhỏ (tùy thuộc vào thành phần cấu trúc B) thường xảy ra trên đường biên.

Thành phần cấu trúc thường được sử dụng là thành phần cấu trúc đối xứng có gốc $(0, 0)$ là ở tâm như hình:

0	1	0
1	1	1
0	1	0

Nhưng do ảnh được quét với độ phân giải 300 dpi, và đối với những chữ có bụng được viết khá nghiêng thì khi thực hiện phóng với thành phần cấu trúc trên, tức thực hiện phép giãn rồi thực hiện phép co, thì phép giãn làm cho phần bụng bị dính lại với nhau do với mỗi hướng ngang và đứng đều được giãn 2 điểm ảnh. Để hạn chế điều này ta sử dụng 2 thành phần cấu trúc không đối xứng và thực hiện phép đóng 2 lần trên 2 thành phần cấu trúc này, vì khi thực hiện phép giãn thì chỉ cần giãn về 1 phía:

0	1	0
1	1	0

0	0	0
---	---	---

0	0	0
0	1	1
0	1	0

Các thành phần cấu trúc không đối xứng

VII) Lấy đường biên và làm trơn đường biên:

Phát hiện biên: Biên của ảnh được thiết lập bằng cách nhân chập ảnh với phần tử có cấu trúc:

0	1	0
1	1	1
0	1	0

Duyệt đường biên :

Đường biên kí tự được duyệt theo cách sau:

Bước 1: quét ảnh đến khi gặp điểm ảnh đen. Gọi là pixel 1

Bước 2: Lặp

Nếu “ điểm ảnh hiện thời là đen ” thì “dò ngược”

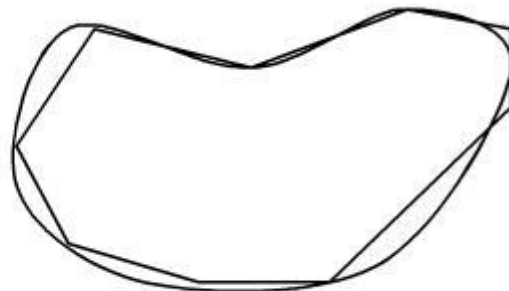
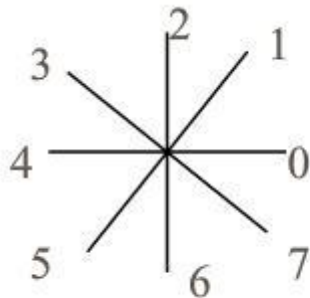
Ngược lại “sang phải”

Đến khi gặp “pixel 1”

Mã hoá hướng điểm biên:

Sau khi duyệt đường biên, ta tiến hành mã hoá hướng các điểm trên đường biên theo 8 hướng sau :

Quy ước :



Làm trơn đường biên :

Duyệt theo đường biên, nếu hai điểm liên tiếp trên đường biên có hiệu số hướng lớn hơn 1 thì hiệu chỉnh để hiệu số hướng bằng 1.

Theo mã hướng Freeman, hiệu số hướng của 2 điểm liên tiếp nhau trên đường biên được định nghĩa :

Goi c_i là mã hướng tại điểm biên đang xét p_i , c_{i+1} là mã hướng của điểm kế tiếp trên đường biên p_{i+1}

Đặt $d = c_{i+1} - c_i$ và

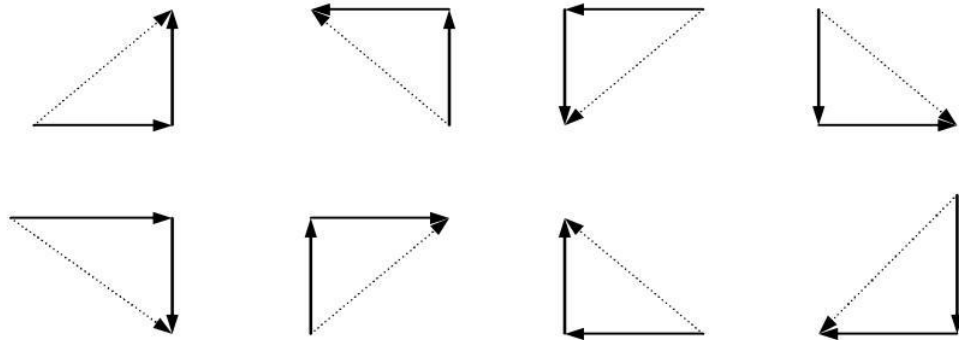
$Dabs = |d|$ nếu $|d| \leq 4$

Và $dabs = 8 - |d|$ nếu $|d| > 4$

Ta có các trường hợp sau:

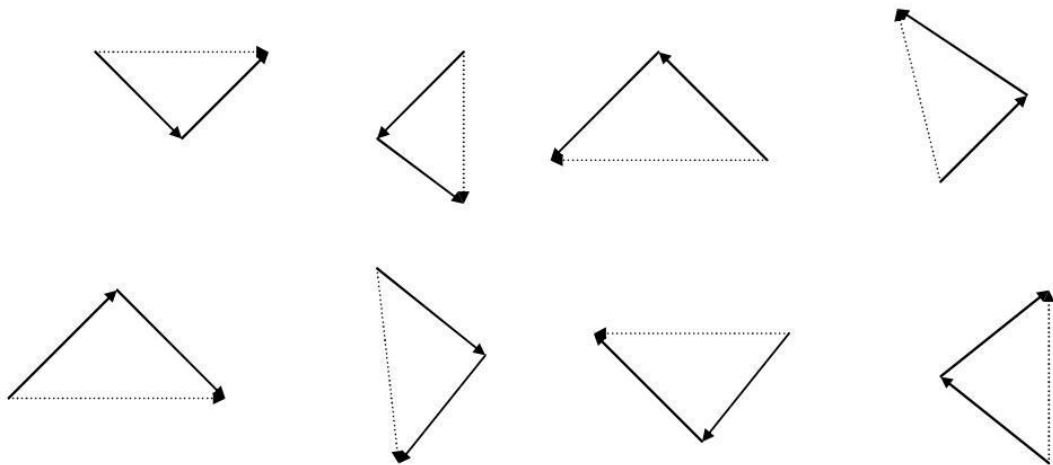
a. $dabs \leq 1$: Điểm biên trơn.

b. $dabs = 2$ và c_i chẵn, c_{i+1} chẵn : bỏ p_{i+1} và thay hướng p_i như sau:



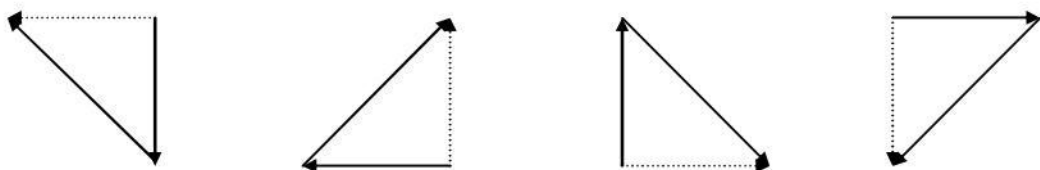
Dabs=2 và c_i chẵn, c_{i+1} chẵn

c. dabs=2 và c_i lẻ, c_{i+1} lẻ : Bỏ p_{i+1} và thay hướng p_i như hình:

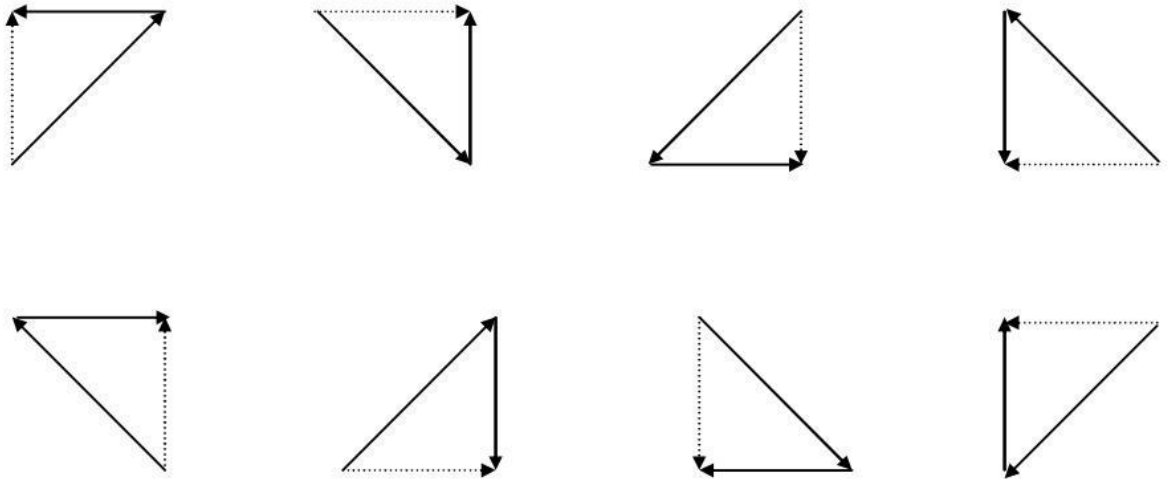


Dabs=2 và c_i lẻ, c_{i+1} lẻ

d. dabs=3, c_i chẵn, c_{i+1} lẻ :



e. dabs=3, c_i lẻ, c_{i+1} chẵn:



Minh hoạ ảnh kí tự sau quá trình tiền xử lý.



Ảnh ban đầu

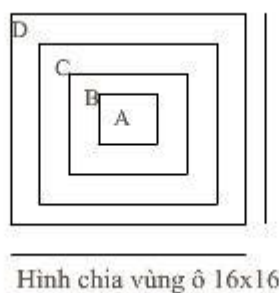
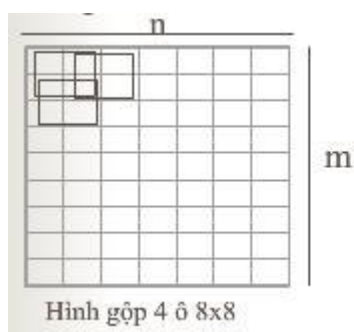


Ảnh qua tiền xử lý

Chương II: Rút đặc trưng của kí tự

I. Chia ô:

Ảnh sau khi tiền xử lý được chia thành các ô vuông nhỏ 8x8. Gom 4 ô kích thước 8x8 thành ô 16x16 phủ lên nhau theo hướng ngang và dọc. Trong mỗi ô 16x16 ta chia làm 4 vùng A,B,C,D :



Vùng A : kích thước 4x4
 Vùng B : kích thước 8x8
 Vùng C : kích thước 12x12
 Vùng D : kích thước 16x16

II) Véc tơ đặc trưng:

Đặt trọng số vùng A,B,C,D tương ứng 4,3,2,1. Gọi x_j là 1 loại đặc trưng, x_j được tính cho 1 ô 16x16 như sau:

$$x_j = 4x_j^{(A)} + 3x_j^{(B)} + 2x_j^{(C)} + x_j^{(D)}$$

Với mỗi ô 16x16 ta rút 4 đặc trưng x_j ($j=1,2,3,4$), x_j tính như trên:

Ảnh kí tự được mô tả:

$$X = (x_1, x_2, x_3, x_4, \dots, x_n)$$

Với $n = k \cdot 4$, k là tổng số

Ô 16x16 xếp chồng lên nhau

x_1 : số điểm biên có hướng 0^0 (hay 180^0)

x_2 : số điểm biên có hướng 45^0 (hay -135^0)

x_3 : số điểm biên có hướng 90^0 (hay -90^0)

x_4 : số điểm biên có hướng 135^0 (hay -45^0)

Chương III : Các Mô Hình Nhận Dạng Và Phân Lớp

I) Một số khái niệm về nhận dạng

1) Một biểu diễn là giá đỡ (cái mang) thông tin, thường biểu diễn dưới dạng sau:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

Mỗi xi biểu diễn kết quả của một phép đo. Tập hợp các biểu diễn xác định X được gọi là không gian biểu diễn. Ví dụ không gian vectơ.

Giải thích một biểu diễn nghĩa là cho một kết quả chẳng hạn một cái tên.

Giả sử: ta có tập hợp các tên là:

$$\Omega = (\omega_1, \omega_2, \dots, \omega_p)$$

Không gian giải thích là một tập thoả mãn các luật, thao tác nào đấy.

Một định danh là một ánh xạ của không gian biểu diễn vào không gian giải thích.

$$\varepsilon : X \rightarrow \Omega$$

$$\varepsilon : (x_1, x_2, \dots, x_n) \rightarrow \Omega$$

Mục đích nhận dạng là thực hiện ánh xạ này và tìm thuật toán để thực hiện trên toàn X. Một thuật toán như vậy gọi là toán tử nhận dạng.

2) Tập mẫu nhận dạng :

Dữ liệu cho bài toán nhận dạng thường được biểu diễn qua tập mẫu học T với

$$T = \{(x, \omega)\} \text{ là tập các cặp (dữ liệu - tên).}$$

3) Độ đồng dạng và dị dạng:

Là hai chỉ số thường dùng để xây dựng trên quan hệ gần thứ tự trên các cặp đặc biệt khoảng cách giữa hai đối tượng là một chỉ số dị dạng thoả mãn 3 tiên đề:

- $p(x, y) \geq 0$, $p(x, x) = 0$
- $p(x, y) = p(y, x)$
- $p(x, z) \leq p(x, y) + p(y, z)$

4) Khoảng cách đối tượng :

Các hàm đặc trưng quan sát có thể dẫn đến một quan hệ gần thứ tự giữa 1 đối tượng X và các khái niệm Ai, nghĩa là với mọi i, j có thể thiết lập một quan hệ :

$$(X, A_i) \leq (X, A_j)$$

Quan hệ này thường được thiết lập nhờ một khoảng cách đối tượng, ký hiệu: $D(X, A)$.

Nếu muốn phân lớp hoặc định danh X có thể dùng thông tin này. Giả sử Ci là lớp phân hoạch tương ứng với khái niệm đại diện Ai ; X được gán vào Ci nếu $D(X, A_i)$ là nhỏ nhất.

II) Một số thuật toán phân lớp :

1) Xếp lớp khoảng cách cực tiểu :

Giả thiết là mỗi lớp mẫu được biểu diễn bằng một vector đơn (hoặc trung bình).

$$m_j = \frac{1}{n_j} \sum_{x \in \omega_j} x \quad j=1,2,\dots,M$$

Trong đó n_j là số vector mẫu từ lớp ω_j , M là số lớp cần phân biệt và tổng được xác định từ các vector này, cách xác định lớp của một vector mẫu x chưa biết là chỉ định nó cho lớp đơn điệu

gần nhất. Dùng khoảng cách Euclid để xác định độ gần sẽ giảm được tính toán.

$$D_j(x) = \|x - m_j\|; \quad j=1,2,\dots,M$$

Trong đó $\|a\| = (a^T a)^{1/2}$ là dạng Euclid. Sau đó ta chỉ định x cho lớp w_j nếu $D_j(x)$ là khoảng cách ngắn nhất. Đó là khoảng cách ngắn nhất dùng trong biểu diễn. Ta dễ dàng nhìn thấy nó tương đương với việc đánh giá bằng hàm số

$$d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j=1,2,\dots,M$$

Và chỉ định x cho lớp w_j , nếu $d_j(x)$ cho giá trị số lớn nhất.

2) Thuật toán hàm thế:

Phương pháp nhận dạng theo hàm thế được ứng dụng nhiều trong thực tiễn. Việc sử dụng hàm thế được xuất phát từ nghĩa thế điện trong trường điện từ:

Trong không gian có điện tích q tại A thì xung quanh nó có điện trường theo mọi phía. Tại điểm M của không gian ta có thể gây ra bởi q là:

$$\varphi(M) = a \cdot \frac{q}{r}$$

a : hằng số

q: độ lớn điện tích q

r: khoảng cách từ M tới q

Các dạng hàm thế thường dùng trong thuật toán nhận dạng:

$$\varphi(S, S') = e^{p\alpha(S, S')}$$

$$\varphi(S, S') = \frac{1}{C_1 + C_2 \cdot P^\alpha(S, S')}$$

Ở đây φ , C_1 , C_2 là các hằng số cho trước. $P^\alpha(S, S')$ là khoảng cách S và S' ($\alpha=0, 1, 2..$)

Cách tính thế đối với mỗi lớp:

$$\varphi(S, K_j) = \frac{1}{m_j} \sum_{S_i \in K_j} \varphi(S, S_i)$$

m_j : số mẫu của K_j

S_i : mẫu thuộc K_j

Ta có luật quyết định:

$$S \in K_j \text{ nếu } \varphi(S, K_j) = \arg \max \varphi(S, K)$$

Chú ý :

Việc tính thế đối với mỗi lớp, có thể bổ sung trọng số mẫu $\gamma(S_i)$:

$$\varphi(S, K_j) = \frac{1}{m_j} \sum_{S_i \in K_j} \varphi(S, S_i) \cdot \gamma(S_i)$$

Nhận xét:

Nếu chọn p là hàm khoảng cách Euclid thì giải thuật hàm thế này gần giống với cách xếp lớp theo khoảng cách cực tiểu.

3. Phương pháp LDA (Linear Discriminant Analysis):

Phương pháp LDA cho trường hợp phân biệt 2 lớp, LDA sẽ tìm một phương chiều mà phân biệt tốt nhất các mẫu thuộc hai lớp khác nhau trong tập mẫu. Giả sử ta có một tập gồm n mẫu học

X bao gồm các vector cột d chiều:

$$x_i = (x_i^1, x_i^2, \dots, x_i^d) \quad i = 1 \dots n$$

Trong đó n_1 mẫu thuộc về lớp C1 và nằm trong tập con X_1 , n_2 mẫu thuộc về lớp C2 và nằm trong tập con X_2 .

Giả sử ta có một vector d chiều w , tích vô hướng $y = w^T x$ biểu diễn hình chiếu của vector x lên phương w . Ta sẽ tìm một phương chiếu w nhằm tối ưu hóa độ phân biệt giữa các mẫu thuộc 2 lớp C1 và C2. Điều này tương đương với việc giảm số chiều của vector đặc trưng xuống còn 1 chiều.

Ta gọi m_i , $i=1, 2$ là trị trung bình của các mẫu tương ứng với 2 lớp C1 và C2 .

$$m_i = \frac{1}{n_i} \sum_{x \in X_i} x$$

Và m_i^* tương ứng là trung bình của các mẫu được chiếu lên phương w :

$$m_i^* = \frac{1}{n_i} \sum_{y \in Y_i} y = \frac{1}{n_i} \sum_{x \in X_i} w^T x = w^T m_i$$

Trong đó y là hình chiếu của x lên w . Y_i là tập các hình chiếu của các $x \in X_i$ lên w .

Ta có thể xem $\|m_1^* - m_2^*\|$ là một độ đo cho tính phân biệt giữa hai tập Y_1 và Y_2 . Tuy nhiên để có được sự phân biệt tốt giữa hai tập khi chiếu lên phương w , ta cần có độ sai khác giữa hai trị trung bình này khá lớn hơn so với độ lệch chuẩn nội tại của mỗi tập (có thể xem như độ rộng của đám mây các mẫu). Thay vì sử dụng phương sai của mỗi tập ta sẽ sử dụng một độ đo khác, gọi là độ rải (scatter) cho các hình chiếu của các mẫu thuộc lớp C_i như sau:

$$S_i^* = \sum_{y \in Y_i} (y - m_i^*)^2$$

Phương pháp LDA sẽ tìm giá trị w để cực đại hóa hàm tiêu chuẩn sau đây:

$$J(w) = \frac{|m_1^* - m_2^*|^2}{(S_1^*) + (S_2^*)}$$

Để thấy $J(w)$ là một hàm theo w ta định nghĩa các ma trận SB và S_w như sau

$$S_w = \sum_{i=1}^2 \sum_{x \in X_i} (x - m_i)(x - m_i)^T$$

$$S_B = (m_1 - m_2)(m_1 - m_2)^T$$

S_w được gọi là ma trận rải nội lớp (within-class scatter matrix)

SB được gọi là ma trận rải liên hợp (between-class scatter matrix)

Ta có:

$$\begin{aligned} S_1^{*2} &= \sum_{x \in X_1} (w^T x - w^T m_1)^2 \\ &= \sum_{x \in X_1} w^T (x - m_1)(x - m_1)^T w \end{aligned}$$

Nên :

$$S_1^{*2} + S_2^{*2} = w^T S_w w$$

Tương tự ta cũng có :

$$(m_1^* - m_2^*)^2 = w^T S_B w$$

Do đó :

$$J(w) = \frac{w^T S_B w}{w^T S_w w}$$

Để xác định w sao cho J(w) cực đại ta cho đạo hàm riêng J(w) theo w bằng 0 kết quả ta sẽ được:

$$S_B w = \lambda S_w w$$

Với λ là trị riêng, giải bài toán tìm trị riêng ta sẽ có:

$$W = S_w^{-1} (m_1 - m_2)$$

Đây là kết quả tìm được của phương pháp LDA đối với trường hợp chỉ có 2 lớp

Sau khi đã tìm được w, mỗi vectơ x cần nhận dạng sẽ được xử lý như sau: lấy x trừ đi trung bình của mẫu học rồi chiếu lên phương w ta được một giá trị vô hướng, tính khoảng cách từ giá trị vô hướng này trên m_i của mỗi lớp này chia cho độ lệch chuẩn σ_i^{*2} ta được một độ đo khoảng cách từ x đến các cụm ứng với mỗi lớp.

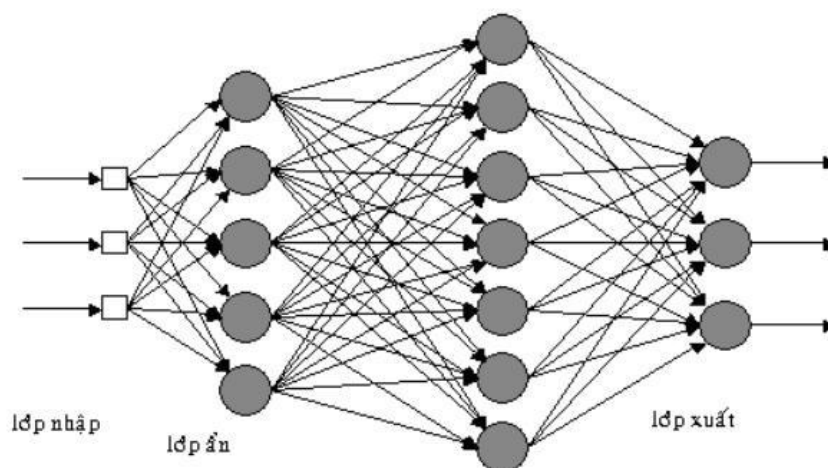
$$d_1 = \frac{(x - m)w - m_1^*}{\sigma_1^*} \quad i=1...2$$

x sẽ được gán vào lớp ứng với cụm gần nhất.

Để phân biệt được n lớp ta xây dựng n bộ phân loại 2 lớp theo phương pháp nêu trên. Mỗi bộ phân loại sẽ phân biệt một lớp với n-1 lớp còn lại. Nếu một vectơ đầu vào được xếp vào cả hai lớp thì ta sẽ sử dụng khoảng cách di nêu trên để quyết định nó thuộc vào lớp nào. Nếu một vectơ không được xếp vào lớp nào thì coi như không nhận dạng được.

4) Phân lớp bằng mạng nơtron:

a) Giới thiệu :



Minh họa mạng nơtron

Mạng nơtron tổng quát có cấu trúc phân lớp, gồm 3 lớp:

Lớp nhập: nhận giá trị bên ngoài vào

Lớp xuất : lớp cuối cùng, sẽ xuất ra kết quả

Lớp ẩn : các lớp còn lại (có thể không có)

Quá trình nhận dạng của mạng là quá trình ánh xạ một mẫu x từ không gian các đặc trưng

vào không gian các lớp.

b. Hoạt động:

Chỉ có thể ở một trong hai trạng thái: trạng thái ánh xạ và trạng thái học.

* **Trạng thái ánh xạ** : Như đã nói, ở trạng thái ánh xạ, mỗi vectơ đầu vào x sẽ được ánh xạ thành một vectơ kết quả z . Quá trình này được thực hiện như sau:

Đầu tiên vectơ mẫu x sẽ được đưa vào lớp nhập. Mỗi neuron trong lớp nhập sẽ mang giá trị của một thành phần của x . Các nút nhập sẽ không tính toán gì cả mà gửi trực tiếp giá trị của nó đến các neuron ở lớp tiếp theo. Tại mỗi neuron của các lớp tiếp theo, một thao tác giống nhau sẽ được thực hiện. Đầu tiên nó sẽ tính tổng trọng hóa của tất cả các giá trị được gửi tới. Sau đó một hàm truyền sẽ được áp dụng trên tổng trọng hóa này để cho giá trị xuất của nút này. Hàm truyền có tác dụng nén giá trị của tổng trọng hóa vào một miền giới hạn nào đó. Giá trị này được truyền cho các neuron ở lớp kế tiếp. Cứ thế thông tin được lan truyền cho đến lớp xuất của mạng.

* **Trạng thái học** : Xét mạng MLP có một lớp ẩn với thuật toán lan truyền ngược.

Thuật toán lan truyền ngược là thuật toán hữu hiệu cho quá trình học của MLP. Thuật toán này sẽ cập nhật trọng số dựa trên một hàm lỗi E giữa kết xuất của mạng với giá trị đích.

Mục đích của việc học có giám sát bằng MLP là cực tiểu hóa hàm lỗi này. Kỹ thuật cơ bản để cực tiểu hóa hàm lỗi là phương pháp giảm gradient. Mặc dù phương pháp này có thể dẫn đến một cực tiểu cục bộ, nhưng nó được áp dụng rộng rãi vì tính đơn giản của nó. Thực tế cũng cho thấy trong hầu hết trường hợp phương pháp giảm gradient đều cho kết quả chấp nhận được.

Quá trình học của mạng MLP theo thuật toán lan truyền ngược sẽ lặp đi lặp lại các thao tác sau:

- Lan truyền tiến : tính kết xuất y của mạng với một mẫu x .
- Lan truyền ngược : tính sai số giữa kết xuất y và giá trị đích t và lan truyền ngược sai số này lại để cập nhật trọng số cho mạng.

Quá trình học sẽ dừng khi mạng đã đạt được một độ lỗi nhỏ nhất định.

Để thay đổi trọng số của mạng nhằm cực tiểu ta có thể áp dụng phương pháp giảm Gradient theo các bước sau :

- + Chọn ngẫu nhiên một điểm x_0 trong không gian trọng số.
- + Tính độ dốc của hàm lỗi tại x_0 .
- + Di chuyển điểm x_0 theo hướng dốc nhất của hàm lỗi.

Quá trình tính độ dốc và di chuyển điểm x_0 được lặp đi lặp lại cho đến khi x_0 tiến tới giá trị làm cho hàm lỗi cực tiểu.

Ta có công thức cập nhật trọng số theo phương pháp giảm gradient:

$$w_{ij}(t+1) = w_{ij}(t) - \varepsilon \frac{\partial E}{\partial w_{ij}}(t) \quad t: \text{số lần cập nhật trọng số hiện tại}$$

ε : hệ số học

w : trọng số bất kì trong mạng

E : Hàm lỗi

*) Quy tắc tính đạo hàm lỗi :

+ Ta có hàm lỗi sai số trung bình bình phương được sử dụng là:

$$E = \frac{1}{2} \frac{\sum_{n=1}^N \sum_{k=1}^K (z_{kn} - t_{kn})^2}{N.K}$$

Với

(X_k, Z_k) : tập mẫu học

T_k : ánh xạ của X_k qua mạng

K : số mẫu học

N : số chiều của vector output

M : số chiều của vector input

+ Trọng số nút xuất :

$$\frac{\partial E}{\partial b_j} = \begin{cases} p & j = 0 \\ py_j & j \leq J \end{cases}$$

Trong đó :

$$p = (z-t)z(1-z)$$

$$y_j = \frac{\partial v}{\partial b_j}$$

E: Hàm lỗi

+ Trọng số nút ẩn:

$$\frac{\partial E}{\partial a_i} = \begin{cases} q & i = 0 \\ qx_i & i > 0 \end{cases}$$

Trong đó :

$$q = \left(\sum_{k=1}^K p_k b_k \right) y(1-y)$$

$$\frac{\partial u}{\partial a_i} = x_i$$

+ Hàm truyền được sử dụng là hàm logistic:

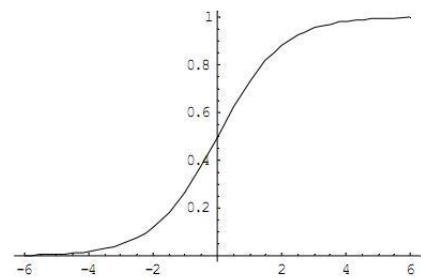
$$f(x) = \frac{1}{1 + e^{-ax}}$$

Mục đích ánh xạ mẫu trong
Quá trình lan truyền tiến

Trong đó :

$$(X_k, Z_k) = (x_{k1}, \dots, x_{kM}; z_{k1}, \dots, z_{kN})$$

$$T_k = \text{ánh xạ } (X_k) = (t_{k1}, \dots, t_{kN})$$



Đồ thị hàm logistic

Chương IV : Minh Họa Chương Trình Nhận Dạng Kí Tự Viết Tay

I. Giới thiệu :

Chương trình nhận dạng kí tự viết tay bước đầu thử nghiệm xây dựng bộ nhận dạng cho 2 lớp kí tự :

Lớp kí tự chữ cái:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,

a, b, d, e, f, g, h, i, j, k, l, m, n, q, r, s, t, v, x, y.

Lớp kí tự số :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

II) Thực hiện chương trình:

Ta có quy trình xử lí như sau:

Ảnh đầu vào → lọc ảnh → nhị phân hóa → tách các liên thông chữ → chỉnh nghiêng → chuẩn hóa kích thước → tìm biên → rút đặc trưng trên đường biên → qua bộ phân lớp → quyết định lớp của ảnh nhận dạng → xuất kết quả theo định dạng trật tự kí tự trên hàng.

Như vậy

- Trong quá trình tiền xử lí, ảnh của kí tự được chuẩn hóa về kích thước chuẩn được chọn là 80x56

- Sau khi xác định biên và mã hóa đường biên, vectơ đặc trưng của kí tự được xác định như lý thuyết đã nói ở phần xác định đặc trưng của kí tự, từ đó ta có :

$$X=(x_1, x_2, \dots, x_n) \quad \text{với } n=216$$

- Mạng nơron được thiết kế để nhận dạng là mạng 2 lớp : 1 lớp vào và 1 lớp ra. Với giải thuật lan truyền ngược và giảm Gradient, chọn $\varepsilon = 0.08$.

- Vectơ $X=(x_1, x_2, \dots, x_n)$ trong quá trình huấn luyện và nhận dạng được chuẩn hóa sao cho $\|X\|=1$, nghĩa là các thành phần của vectơ X được tính lại như sau:

$$x_i = \frac{x_i}{\|X\|} \quad i = 1..n$$

Thử nghiệm trên 2 mạng :

- Mạng thứ nhất được thử nghiệm trên 2366 mẫu học đối với lớp kí tự chữ.

- Mạng thứ hai thử nghiệm trên 1000 mẫu học đối với lớp kí tự số.

- Mạng thứ nhất luyện sau 30.000 học kì.

- Mạng thứ hai luyện sau 10.000 học kì.

Kết quả:

Sau thời gian học mạng thứ nhất và mạng thứ hai phân biệt gần hoàn toàn các mẫu đã học, đặc biệt khả năng tổng quát của mạng rất cao.

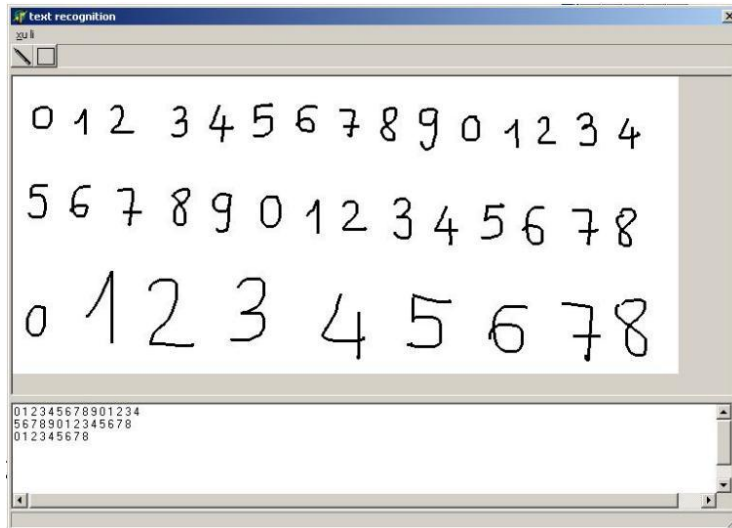
III) Minh Họa 1 Số Kết Quả:

+ Nhận dạng kí tự số :

Ảnh viết bằng mouse:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
 5 6 7 8 9 0 1 2 3 4 5 6 7 8
 0 1 2 3 4 5 6 7 8

Kết Quả :

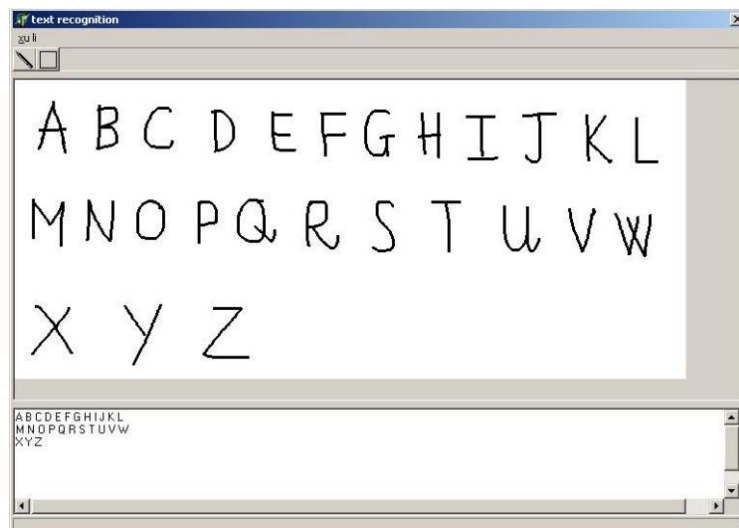


+ Nhận dạng kí tự chữ:

Ảnh viết bằng mouse :

A B C D E F G H I J K L
 M N O P Q R S T U V W
 X Y Z

Kết quả :



Như vậy, chúng ta đã tìm hiểu sơ qua về cơ chế nhận dạng kí tự viết tay (bao gồm chữ và số).

Trên đây là toàn bộ nội dung bài tập nhóm mà nhóm chúng em đã làm xong. Do kiến thức còn hạn chế nên không thể tránh khỏi thiếu sót. Mong cô giáo hướng dẫn thêm để chúng em hoàn thành tốt bài tập nhóm này.