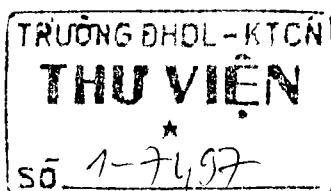


**CÔNG TRÌNH DỰ THI  
GIẢI THƯỞNG “SINH VIÊN NGHIÊN CỨU KHOA HỌC”  
NĂM 2000**

**Tên công trình :**

**NHẬN DẠNG KÝ TỰ TIẾNG VIỆT TRÊN KÊNH  
TRUYỀN DÙNG NEURAL NETWORK**

**Thuộc nhóm ngành : KHOA HỌC KỸ THUẬT 1**



# NHẬN DẠNG KÝ TỰ TIẾNG VIỆT DÙNG NEURAL NETWORK

## TÓM TẮT

Mạng Neural Network đã được soạn thảo từ những yếu tố đơn giản của phép tính được xử lý song song. Những yếu tố này đã được truyền bằng một hệ thống thần kinh sinh vật. Như trong tự nhiên, chức năng của mạng đã được xác định một cách rộng lớn bằng việc kết nối giữa các thành phần. Chúng ta có thể huấn luyện mạng để thực hiện các chức năng đặc thù bằng cách điều chỉnh những giá trị của kết nối hay trọng số giữa các thành phần.

Thông thường, mạng Neural Network đã được điều chỉnh hay huấn luyện vì thế những ngõ vào đặc thù qua những bước để cho ngõ ra đạt được mục đích mong muốn. Những tình huống đó được trình bày như sau. Ở đó, mạng đã được điều chỉnh và làm nền tảng trên việc so sánh giữa ngõ ra so với mục đích, cho đến khi ngõ ra của mạng hợp với mục đích. Một cách điển hình nhiều ngõ vào và đích là một cặp đã được dùng để huấn luyện cho mạng.

Mạng Neural Network đã được huấn luyện để thực hiện những chức năng phức tạp trong những lĩnh vực ứng dụng bao gồm nhận dạng mẫu, nhận diện, phân loại, nhận dạng tiếng nói, hình ảnh và hệ thống điều khiển.

Đề tài này nhằm sử dụng mô hình của neural networks để nhận dạng ký tự tiếng Việt trên kênh truyền trong trường hợp bị nhiễu xâm lấn. Bằng phương pháp chọn lựa những lớp (layers), những hàm số dùng trong mạng tương thích, hệ thống có khả năng nhận dạng hay làm sạch nhiễu khi các ký tự truyền qua kênh truyền có nhiễu. Nội dung bao gồm các phần chính: tìm hiểu các mô hình mạng neural; thiết kế những kiểu mẫu dùng cho việc mã hóa các ký tự tiếng Việt (bao gồm chữ thường, hoa và các dấu); mô phỏng trên máy tính cho hệ thống dùng chương trình Matlab; đánh giá kết quả.

## TỪ KHÓA

Những hàm truyền, mô hình Perceptron, giải thuật trung bình bình phương tối thiểu, giải thuật lan truyền ngược, mô hình học thuật.

## **APPLYING NEURAL NETWORKS TO VIETNAMESE CHARACTER RECOGNITION IN CHANNEL COMMUNICATIONS**

### **ABSTRACT**

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (or weights) between elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown below. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, to train a network.

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

This thesis used model of neural network to recognition character\_vietnamese in the noise channel communication. By method select layers, the functions use in corresponding network, the systems can identify or cancel noise when the letters were transfer in noise channel communication. The content consist of main parts: research models neural network; design pattern use to code characters\_vietnamese (including letter upper, letter lower, mark letter); and simulate on computer for systems used program Matlab; appraisement result.

### **KEY WORDS**

Transfer functions, perceptron, least means squared (LMS), back propagation algorithm.

## 1. ĐẶT VẤN ĐỀ:

Trong những năm gần đây, ngành Điện tử Viễn thông không ngừng phát triển. Nó không chỉ gây ra những chuyển biến thần kỳ trong bản thân ngành Vô Tuyến Điện mà còn trở thành một phương tiện sắc bén, thúc đẩy sự tiến bộ của nhiều ngành khác. Trong xu thế chung của toàn cầu là tích cực đẩy mạnh cuộc cách mạng khoa học kỹ thuật, và có những bước nhảy vọt mới về khoa học kỹ thuật. Cùng với sự tiến bộ của ngành Viễn thông thì việc trao đổi (phát và nhận) thông tin với nhau ở những khoảng cách xa phải đảm bảo được tính chính xác của tin tức, dữ liệu và dữ liệu đó phải đúng với nguyên thủy của nó. Tuy nhiên, theo chúng ta đã biết bất kỳ dạng thông tin nào sau khi đi qua môi trường truyền dẫn đều bị ích nhiều thay đổi so với dạng gốc của chúng. Điều này đặc biệt quan trọng đối với những dữ liệu đòi hỏi độ chính xác cao. Mà dữ liệu ở nơi thu bị sai lệch là do nhiều nguyên nhân chẳng hạn như hiện tượng Fading hay nhiễu gây ra. Có nhiều loại nhiễu ảnh hưởng đến việc tăng giảm tín hiệu như: nhiễu nhiệt, nhiễu gausse, nhiễu trắng,..... Bên cạnh đó, việc thu nhận thông tin bằng tiếng Việt (ký tự có dấu) trên kênh truyền và hiển thị chúng vẫn còn là một đề tài được nhiều ngành khoa học quan tâm và nghiên cứu. Đặc biệt là trong quá trình hội nhập quốc tế về lĩnh vực thông tin viễn thông ở nước ta.

Do những nguyên nhân trên mà tại nơi thu đòi hỏi phải có thiết bị hoặc phần mềm xử lý, triệt nhiễu, hiển thị để có được những thông tin chính xác và giống với dữ liệu gốc nhất. Để có thể thực hiện được điều đó thì con người phải có những sự thay đổi về mặt kỹ thuật để thay thế cho những kỹ thuật đã cũ kỹ, lạc hậu hoặc những phần mềm có thể phối hợp với máy tính để triệt nhiễu.

Chính vì những lý do trên, mà tôi đã chọn đề tài: **“Nhận dạng ký tự Tiếng Việt dùng Neural network”**.

Sự phát triển của các công trình nghiên cứu về tính toán theo mô hình Neural đã được phát triển ban đầu theo sự nhận định của trí óc con người. Mục đích của sự tính toán Neural là để hiểu được những qui luật dùng làm cơ sở cho sự hiểu biết những lời giải của trí óc về các lãnh vực như nhận dạng hình ảnh, ký tự, tiếng nói và áp dụng những qui luật đó vào trong lĩnh vực máy tính.

Một điều cần chú ý ở đây là việc xử lý những thông tin theo kiểu xử lý song song của bộ não chứ không phải tốc độ tính toán của nó.

Mạng Neural là một ngành kỹ thuật mới. Kỹ thuật này cố gắng đạt được thi hành tốt bằng sự kết nối những phần tử tính toán với nhau. Ở đây nhiều giả thiết được xử lý song song, và tốc độ tính toán của máy tính phải rất cao. Thay vì xử lý những lệnh của chương trình theo trình tự trước sau nhưng với mạng Neural thì xử lý song song những giả thiết khác nhau hoặc thậm trí cạnh tranh với nhau. Mạng neural gồm có nhiều phần tử tính toán và chúng được kết nối với nhau, và sự liên hệ đó được biểu diễn bởi trọng lượng của một phần tử đối với các phần tử tính toán khác trong mạng.

Những lợi ích của mạng Neural rất to lớn, đặc biệt là nó cung cấp độ chính xác cao, đồng thời nó cũng chấp nhận một lỗi sai sót cho phép. Tuy nhiên, mạng Neural gồm nhiều nút

mạng và những nút mạng được xử lý đồng thời. Nếu một trong vài nút mạng bị hư hỏng thì mạng vẫn tiếp tục vận hành đúng và kết quả bài toán không bị ảnh hưởng bao nhiêu.

## 2. MỤC TIÊU – PHƯƠNG PHÁP:

Mục đích của công trình nhằm nghiên cứu để giải quyết 3 vấn đề lớn đặt ra khi truyền thông tin dạng ký tự tiếng Việt có dấu trên kênh thông tin như sau :

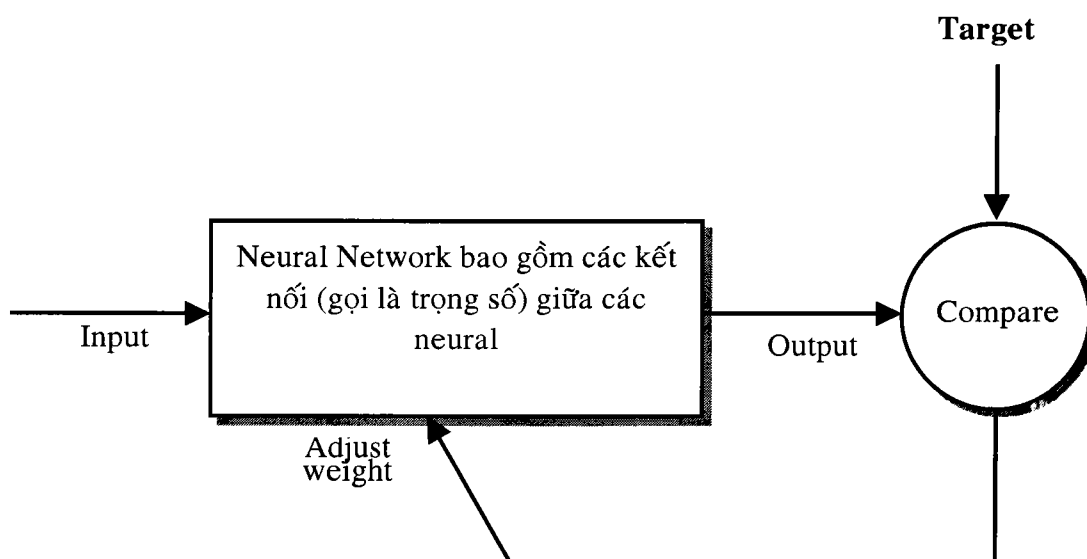
- Thiết kế mạng Neural có khả năng phân loại và nhận dạng ký tự tiếng Việt (ký tự có dấu) trên kênh thông tin.
- Thiết kế bộ phông (Font có dấu) chuẩn tiếng Việt.
- Đề xuất khả năng triệt nhiễu khi ứng dụng mạng Neural trong trường hợp thông tin bị nhiễu trên đường truyền.
- Dùng phần mềm Matlab để mô phỏng hệ thống.

Sau đây, mạng Neural sẽ được xem xét ở khía cạnh xây dựng mạng và kiểm giải thuật hay còn gọi là thuật toán để thực hiện mạng.

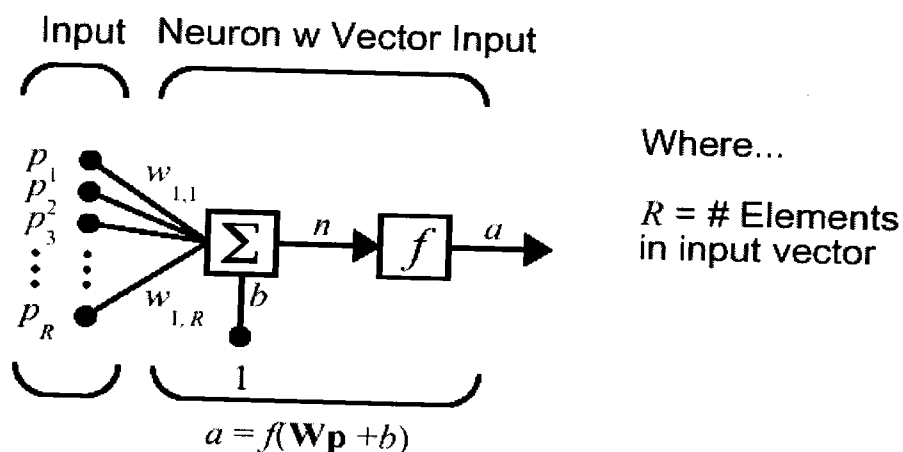
## 3. GIẢI QUYẾT VẤN ĐỀ:

Mạng Neural được diễn tả bởi cơ cấu tổ chức của mạng, những đặc tính của từng nút mạng và những qui luật học tập. Những qui luật cung cấp lúc ban đầu một tập trọng lượng và đồng thời chỉ dẫn cách biến đổi trọng lượng để tăng cường sự liên hệ của những phần tử tính và qua đó tăng hiệu lực thi hành của mạng .

### 3.1. MÔ HÌNH MẠNG NEURAL NETWORK:



### 3.2. MẠNG NEURAL ĐƠN GIẢN



Neural là một đơn vị xử lý thông tin cơ bản cho sự vận hành của Neural, mô hình bao gồm có ba phần chính:

- (1) Một tập “nối kết” được biểu diễn bằng trọng lượng (weight) của nó.
- (2) Một bộ “cộng” để cộng những tín hiệu input, nhân với trọng lượng tương ứng của neural. Phép toán này gọi là phép toán tổ hợp tuyến tính “linear combiner”.
- (3) Một hàm số  $\varphi$  để hạn chế biên độ của biên độ tín hiệu ngõ ra của một Neural. Hàm số đó cũng được gọi là “squashing function”, nghĩa là hạn chế cự ly của biên độ. Đặc biệt biên độ dải chuẩn xuất của neural là  $[0, 1]$  hoặc  $[-1, 1]$ .

Một neural k được diễn tả như sau:

$$U_k = \sum_{i=1}^R W_i \cdot p_i$$

$$Y_k = \varphi ( U_k - \theta_k ).$$

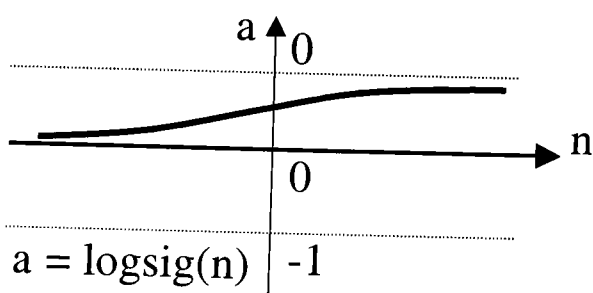
R thành phần vector ngõ vào  $p_1, p_2, p_3, \dots, p_R$

Nhân với trọng số:  $w_1, w_2, w_3, \dots, w_R$

Biểu thức trước khi qua hàm  $f$ :  $n = \mathbf{W} \cdot \mathbf{p} + b$

### 3.3. HÀM TRUYỀN TRONG MẠNG

Có nhiều loại hàm truyền hoạt động trên mạng như: hard limit, linear, ... Nhưng đối với ứng dụng của đề tài thì hàm truyền chủ yếu là hàm Sigmoid, hàm số này thông dụng nhất trong các hàm số hoạt động để xây dựng mạng Neural.



Hàm truyền này dùng để lan truyền ngược trong mạng.

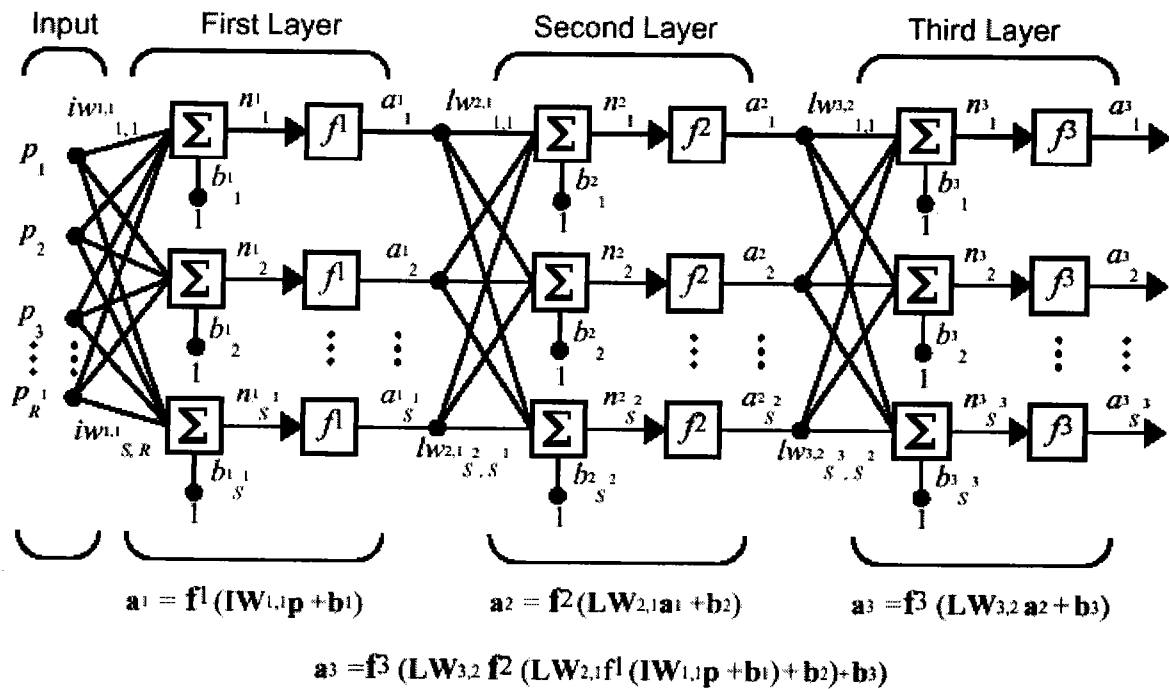
Hàm số sigmoid là hàm số logic:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

a là hằng số dốc của hàm số sigmoid, nếu a tiến đến  $+\infty$  thì  $\varphi(v) = 1$

### 3.4. NHIỀU LỚP CỦA MẠNG NEURAL:

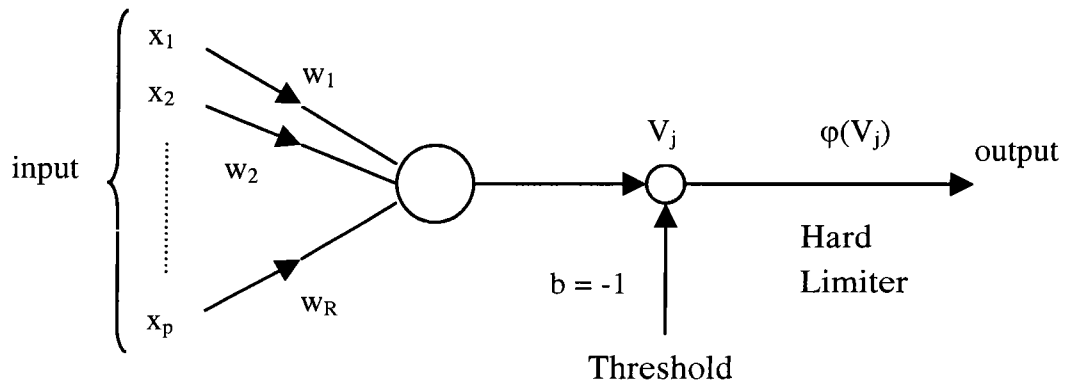
Một mạng có thể có nhiều lớp. Mỗi lớp có ma trận trọng số W, và vector độ lệch b và vector ngõ ra a, để phân biệt giữa các ma trận trọng số và vector ngõ ra ... cho mỗi lớp trong hình của chúng, chúng ta sẽ viết thêm vào số của lớp bên trên để quan tâm sự thay đổi.



### 3.5. MÔ HÌNH PERCEPTRON

Mô hình perceptron là một mạng Neural điển tả dưới hình thức giản dị nhất. Nó chỉ có một neural duy nhất. Mô hình perceptron chỉ được sử dụng cho việc phân loại mẫu (pattern) ra hai loại mà thôi.

Nếu  $w_1, w_2, w_3, \dots, w_p$  chỉ định những trọng lượng của một tầng duy nhất, những tín hiệu input tương ứng là  $x_1, x_2, \dots, x_p$  và b là độ ngưỡng bên ngoài.



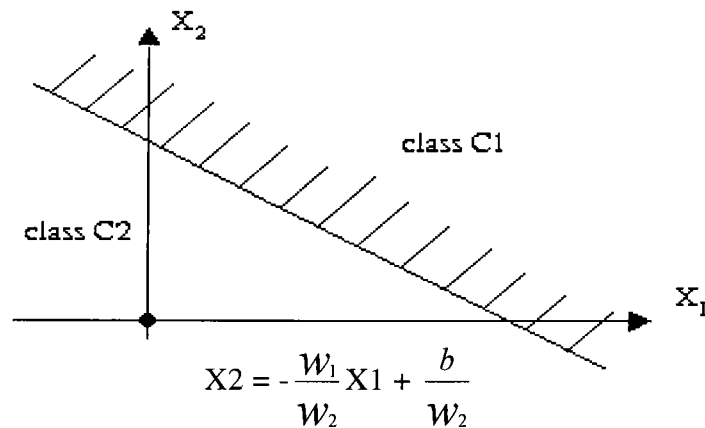
Từ mô hình trên chúng ta kiểm tra output tổ hợp tuyến tính là:

$$V = \sum_{i=1}^R W_i x_i - b$$

Mục đích của mô hình là để phân loại tổ hợp kích thích bên ngoài  $x_1, x_2, \dots, x_p$  ra thành hai loại  $C_1$  hoặc  $C_2$ .  $x_1, x_2, \dots, x_p$  thuộc loại  $C_1$  nếu tín hiệu output của perceptron là  $+1$ , và ngược lại nếu  $x_1, x_2, \dots, x_p$  thuộc loại  $C_2$  nếu tín hiệu output của perceptron là  $-1$ . Trong trường hợp đó chúng ta vẽ biểu đồ những vùng quyết định trong không gian  $p$  chiều dựa trên nền tảng  $p$  biến số input  $x_1, x_2, \dots, x_p$ .

Trong trường hợp của một mô hình cơ bản chúng ta có hai vùng quyết định được phân chia bởi một vùng siêu phẳng (hyperplane) định nghĩa bởi:

$$\sum_{i=1}^R W_i x_i - b = 0$$



Trong trường hợp của biến số  $x_1$  và  $x_2$  có vùng quyết định là một đường thẳng. Một điểm  $(x_1, x_2)$  nằm trên đường phân tách là thuộc  $C_1$  và một điểm  $(x_1, x_2)$  nằm dưới đường phân tách thì thuộc  $C_2$ . Tác dụng của  $b$  là dời đường phân tách ra xa điểm gốc. Trọng lượng  $w_1, w_2, w_3, \dots, w_p$  của perceptron có thể là cố định hoặc thay đổi theo tính toán lặp. Trọng lượng này có thể thay đổi theo giải thuật hội tụ của perceptron.

### 3.5.1. HỌC HỎI CỦA NEURAL

Chúng ta xem xét làm sao để diễn tả khái niệm học hỏi trong mô trình perceptron. Nếu perceptron cho ra một tín hiệu output sai, chúng ta phải giảm bớt xác suất neural cho ra



output sai ấy đừng trở lại, ngược lại nếu neural cho ra output đúng, thì chúng ta không cần làm gì hết. Qui luật chính là cho phép neural học hỏi từ chính sai lầm của nó.

Thí dụ, diễn tả trạng thái đầu của neural, chúng ta cấp cho nó lúc ban đầu tín hiệu input với những trọng lượng ngẫu nhiên. Đó là trạng thái mà neural không biết gì hết. Bây giờ ta cho vào một vật thuộc loại  $C_1$ . Neural cộng tất cả các input rồi nhân với trọng lượng tương ứng và so sánh tổng cộng trị số đó với trị số ngưỡng. Nếu trị số của input lớn hơn trị số ngưỡng, output của neural có trị số 1, nếu không nó cho ra output với trị số 0. Lúc ban đầu, xác suất có kết quả đúng là 50:50, vì input của neural chỉ có một xác suất ngẫu nhiên lớn hơn độ ngưỡng. Chúng ta giả thiết, nó cho ra kết quả đúng, như vậy chúng ta không cần làm gì thêm hết. Nhưng nếu output của neural là 0 khi mà ta cho vào vật thuộc loại  $C_1$ . Như vậy chúng ta phải thay đổi trọng lượng tương đương của input hầu để tổng số của trị số nhập lớn hơn trị số ngưỡng và cho ra kết quả đúng, nghĩa là cho ra trị số 1.

Nếu input của vật thuộc loại  $C_2$ , chúng ta muốn output của neural là 0. Nghĩa là chúng ta muốn tổng số của trị số input nhỏ hơn trị số ngưỡng. Như vậy mỗi lần chúng ta cho vào vật thuộc loại  $C_2$ , chúng ta làm giảm trị số của trọng lượng để bắt buộc output của neural là 0 ở lần tới.

Hai loại này còn được gọi là khả tách tuyến tính (linearly separable) nếu chúng ta tìm ra một tập hợp  $\omega$ . Mặc khác, nếu chúng ta có một vector trọng lượng  $\omega$  thì hai loại vật  $C_1$  và  $C_2$  là khả tách tuyến tính.

Thủ tục học (learning procedure) mà chúng ta vừa trình bày ở trên có thể được tóm tắt như sau:

### 3.5.2. GIẢI THUẬT HỘI TỤ HỌC CỦA PERCEPTRON:

- (1) Khởi tạo trọng lượng và trị ngưỡng.

Gọi  $W_i(t)$  ( $0 \leq i \leq p$ ), là trọng lượng từ input  $i$  ở thời điểm  $t$ , và  $b$  là trị ngưỡng trong nút output.

Lấy  $W_0 = -b$  gọi là độ lệch và  $x_0 = 1$ .

Khởi tạo  $W_i(0)$  là các giá trị nhỏ ngẫu nhiên và giá trị ngưỡng.

- (2) Trình bày output và input mong muốn.

Trình bày input  $x_1, x_2, \dots, x_p$  và output mong muốn  $d(t)$ .

- (3) Tính output hiện hành

$$Y(t) = \varphi \left( \sum_{i=0}^R W_i(t) \cdot x_i(t) - b \right)$$

- (4) Hiệu chỉnh trọng lượng.

Cập nhật vector trọng lượng

$$W_i(t+1) = W_i(t) + \eta(d(t) - y(t))x_i(t)$$

Với đáp ứng mong muốn  $d(t)$  được định nghĩa như sau

$D(t) = 1$  nếu  $x_i$  thuộc lớp  $C_1$ .

$= -1$  nếu  $x_i$  thuộc lớp  $C_2$

và  $\eta =$  thông số tỉ lệ học, là hằng số dương nhỏ hơn 1.

- (5) Tăng thời gian  $t$  lên một và quay lại bước 2.

### 3.6. GIẢI THUẬT TRUNG BÌNH BÌNH PHƯƠNG TỐI THIỂU (LMS)

**Giải thuật LMS:**

- (1) Khởi tạo

Đặt  $\overline{w_k}(1) = 0$  với  $k = 1, 2, \dots, p$

- (2) Loc

Với thời gian  $t = 1, 2, \dots$ , giá trị tính toán

$$y(t) = \sum_{j=1}^p \overline{w_j}(t) x_j(t)$$

$$e(t) = d(t) - y(t)$$

$$\overline{w_k}(t) = w_k(t) + \eta e(t) x_k(t) \quad \text{với } k = 1, 2, \dots, p$$

**Hướng dẫn cách chọn lựa trị số  $\eta$ :**

Xét hội tụ của giải thuật LMS.

7. Giải thuật LMS được nói là hội tụ theo nghĩa  $\langle \overline{W_k}(t) \rangle \rightarrow W_0$  khi  $t \rightarrow \infty$ .

Và LMS hội tụ theo nghĩa trung bình nếu  $0 < \eta < \frac{2}{\lambda_{\max}}$  với  $\lambda_{\max}$  là trị eigen của

$$R_x (= \langle x(t)x^T(x) \rangle)$$

8. LMS được nói là hội tụ theo nghĩa bình phương trung bình nếu  $\langle e^2(t) \rangle \rightarrow W_0$  khi  $t \rightarrow \infty$ .

Và LMS hội tụ theo nghĩa bình phương trung bình nếu  $0 < \eta < \frac{2}{\text{tr}[R_x]}$  với

$$\text{tr}[R_x] = \sum_{k=1}^p \lambda_k \geq \lambda_{\max}$$

### 3.7. THUẬT GIẢI LAN TRUYỀN NGƯỢC (Back propagation)

Thuật giải huấn luyện lan truyền ngược là một thuật giải gradient lập có mục đích cực tiểu hóa sai số bình phương trung bình giữa output thực sự của một perceptron nhiều tầng và output mong muốn. Ta nhớ lại, tính chất phi tuyến của hàm logsig-moid:

$$\varphi(v(n)) = \frac{1}{1 + \exp(-v(n))}$$

- (1) **Khởi tạo trọng lượng và độ dời:** đặt tất cả trọng lượng và mức ngưỡng của mạng với giá trị ngẫu nhiên bé.
- (2) **Trình bày input và output mong muốn:** giới thiệu một vector input liên tục  $p_0, p_1, p_2, p_3, \dots, p_{R-1}$  và xác định các output mong muốn  $d_0, d_1, d_2, \dots, d_{R-1}$ .
- (3) **Sự tính toán tiến:** đối với mỗi  $[p(n), d(n)]$ , tính thế năng hoạt động và tín hiệu hoạt động của mạng bằng cách tiến về phía trước xuyên qua mạng từng tầng một. Cấp hoạt động bên trong lưới  $v_j^{(l)}(n)$  đối với neural  $j$  trong tầng  $l$  là:

$$v_j^{(l)}(n) = \sum_{i=0}^{N_{l-1}} W_{ji}^{(l)}(n) y_i^{(l-1)}(n)$$

với  $N_{l-1}$  là số nút trong lớp  $l-1$ ,  $y_i^{(l-1)}(n)$  là tín hiệu hàm của neural  $i$  trong lớp trước  $l-1$  tại lần lặp  $n$  và  $w_{ji}^{(l-1)}(n)$  là trọng lượng của neural  $j$  trong lớp  $l$  mà được cấp từ neural  $i$  trong lớp  $l-1$ . Đối với  $i=0$  ta có  $y_0^{(l-1)}(n) = -1$  và

$$w_{j0}^{(l)} = \theta_j^{(l)}(n).$$

$\theta_j^{(l)}(n)$  là ngưỡng được áp dụng vào neural  $j$  trong tầng  $l$ . giả sử rằng, việc sử dụng một hàm phi tuyến, tín hiệu (output) hoạt động của neural  $j$  trong lớp  $l$  là:

$$y_j^{(l)}(n) = \frac{1}{1 + \exp(-v_j^{(l)}(n))}$$

nếu neural  $j$  trong tầng ẩn đầu tiên (nghĩa là  $l=1$ ) đặt  $y_j^{(0)}(n) = p_j(n)$

$p_j(n)$  là thành phần thứ  $j$  của vector input  $x(n)$ . Nếu neural  $j$  ở trong tầng output (nghĩa là  $l=L$ ), ta đặt  $y_j^{(L)}(n) = o_j(n)$ . Vì vậy tính tín hiệu lỗi

$$e_j(n) = d_j(n) - o_j(n)$$

với  $d_j(n)$  là phần tử thứ  $j$  của vector đáp ứng mong muốn  $d(n)$ .

- (4) **Sự tính toán lùi:** tính  $\delta$  (nghĩa là gradient cục bộ) của mạng bằng cách tiến hành lùi, từ lớp này qua lớp khác.

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) o_j(n) [1 - o_j(n)] \text{ cho neural } j \text{ trong một tầng output } L.$$

$$\delta_j^{(l)}(n) = y_j^{(l)}(n) [1 - y_j^{(l)}(n)] \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \text{ cho neural } j \text{ trong lớp ẩn } l.$$

Vì vậy hiệu chỉnh trọng lượng của mạng trong lớp  $l$  theo quy luật delta tổng quát hóa:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha [w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n+1)] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

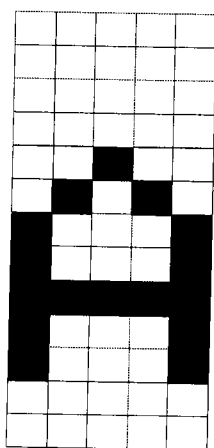
với  $\eta$  là thông số tỉ lệ học tập và  $\alpha$  là moment với  $0 < \alpha < 1$ .

- (5) **Lặp:** lặp lại việc tính toán bằng cách trình bày những thời kỳ của các thí dụ huấn luyện mới cho mạng tới khi các thông số tự do của mạng ổn định giá trị của chúng và sai số bình phương trung bình  $E_{av}$  được tính trên toàn bộ tập huấn luyện có giá trị nhỏ chấp nhận được hoặc cực tiểu.

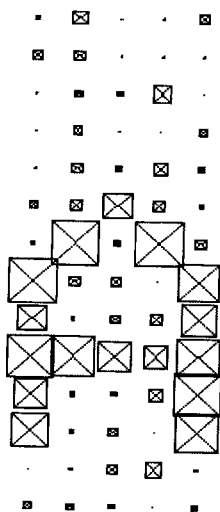
### 3.8. ỨNG DỤNG MẠNG NHẬN DẠNG KÝ TỰ TIẾNG VIỆT

Mạng được thiết kế và huấn luyện để nhận biết 203 ký tự alphabet. Hình ảnh hệ thống đã được số hóa mỗi ký tự trong hệ thống thấy được là những biến. kết quả là mỗi ký tự thì được đại diện bởi một lưới 5-14 giá trị boolean.

Ví dụ sau đây là chữ A:



Tuy nhiên hệ thống này không được hoàn hảo và chịu sự ảnh hưởng của nhiễu do đó chữ A sẽ trở thành:



Hai trăm lẻ ba (203) ký tự với mỗi ký tự có 70 thành phần vector ngõ vào thì được định nghĩa trong hàm **bangchu**. Vector mục tiêu thì cũng được định nghĩa trong bài này với

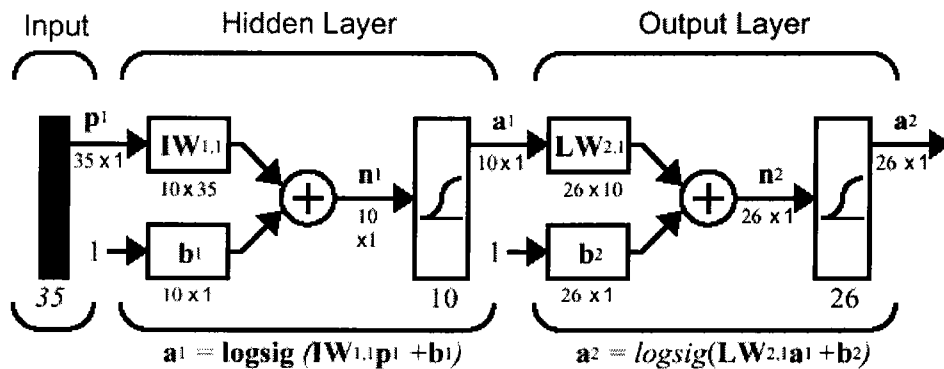
những biến được gọi là đích (target). Mỗi vector mục đích là 203 vector thành phần như **1** với vị trí của ký tự mà nó đại diện. Và **0** ở thành phần thứ 2 cho đến 203.

Mạng Neural: mạng sẽ nhận được 70 giá trị boolean như 70 ngõ vào. Nó được yêu cầu để xác nhận ký tự bằng cách đáp ứng với 203 vector thành phần ngõ ra. 203 vector thành phần ngõ ra của mỗi ký tự đại diện. Để hoạt động đúng trong mạng sẽ đáp ứng với 1 trong những vị trí đã được đại diện trong mạng. Tất cả những giá trị khác ở ngõ ra sẽ bằng không.

Thêm vào đó, mạng sẽ có thể dùng để xử lý nhiễu. Trong thực tế mạng sẽ không nhận được vector boolean hoàn toàn như ngõ vào. Một cách đặc biệt, mạng sẽ có thể làm đối với một vài lỗi khi việc phân loại vector với nhiễu bằng 0 và sự sai lệch nhỏ hơn hoặc bằng 0,2 (giá trị 0.2 ở đây là tùy thuộc vào yêu cầu đặt ra trước trong khi thiết kế mạng).

### 3.8.1. CẤU TRÚC CỦA MẠNG:

Mạng Neural cần 70 ngõ vào và 203 Neural ở lớp ngõ ra của nó để xác định ký tự mạng có hai lớp mạng. Hàm truyền **log-sigmoid** được chọn lựa bởi vì ngõ ra của nó trong khoảng từ 0 đến 1 nó cũng hoàn thiện cho cho việc học giá trị boolean ở ngõ ra.



Lớp ẩn (đầu tiên) có 100 neural. Số này được chọn lựa bởi sự phỏng đoán và kinh nghiệm. Nếu mạng có sự học sai, thì neural có thể được tăng thêm ở lớp này.

Mạng được huấn luyện để ngõ ra bằng một ở vị trí đúng của nó và các ngõ ra khác bằng không. Tuy nhiên nhiễu sẽ ảnh hưởng để ngõ ra sẽ không tạo hoàn toàn là giá trị 1 hay 0. Sau khi mạng đã được huấn luyện ngõ ra sẽ chuyển qua hàm cạnh tranh **compet**. Điều này làm cho chắc chắn rằng ngõ ra đúng với ký tự nhất, kết quả của việc xử lý này là ngõ ra sẽ thật với người dùng.

### 3.8.2. CÀI ĐẶT CÁC THÔNG SỐ:

Tạo mạng hai lớp:

$S1 = 100$ ;  $S2 = 203$ ;

`net = newff(minmax(P),[ S1 S2],{'logsig' 'logsig'},'traingdx');`

### 3.8.3. HUẤN LUYỆN:

Để tạo một mạng mà có thể xử lý vector nhiều ngõ vào. Nó thì tốt nhất để train cho cả vector nhiều và lý tưởng. Để làm điều này thì đầu tiên mạng sẽ được train trên vector lý tưởng cho đến khi nó có tổng bình phương lỗi thấp.

Sau đó mạng được train trên 10 bộ vector lý tưởng và nhiễu. Mạng đã được train trên hai bảng sao của nhiễu ký tự alphabet tiếng Việt được dùng để duy trì mạng có khả năng phân loại vector lý tưởng ở ngõ vào.

Một cách đáng tiếc, sau khi việc training được mô tả ở trên mạng có thể được học để phân loại một vài vector nhiễu khó như vector nhiễu tự do bị tiêu hao độ chính xác. Do đó mạng một lần nữa được train chỉ vector lý tưởng. Điều này đảm bảo rằng mạng sẽ hoàn toàn đáp ứng khi được trình bày với một ký tự lý tưởng. Tất cả việc training được làm với hàm **traingdx**

Việc training không có nhiễu: mạng sẽ được bắt đầu train mà không có nhiễu cho lớn nhất là 5000 epoch hay tổng bình phương nhiễu của mạng rớt dưới 0.01.

```
P = alphabet; % gồm 203 ký tự tiếng Việt.
T = targets;
net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 20;
net.trainParam.epochs = 5000;
net.trainParam.mc = 0.95;
[net,tr] = train(net,P,T);
```

Việc training với nhiễu: để đạt được mạng không nhạy cảm với nhiễu. Chúng ta đã train với hai bảng sao lý tưởng và hai bảng sao nhiễu trong bảng alphabet. Những vector mục đích bao gồm 4 bảng sao của các vector ở đích. Những cvector nhiễu bằng 0.1 và 0.2 đã thêm vào chúng. Neural linh động này được học để xác định chính xác ký tự nhiễu như thế nào, trong khi đang yêu cầu mà nó vẫn có thể đáp ứng tốt cho vector lý tưởng.

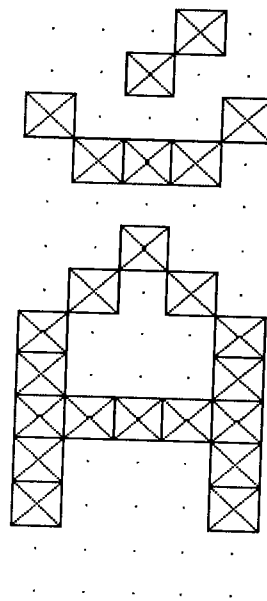
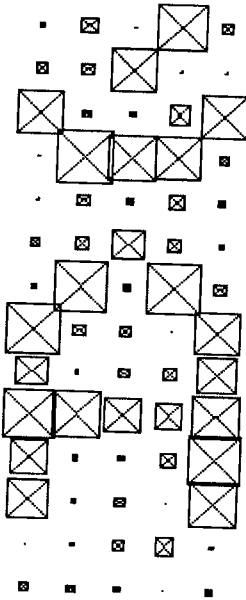
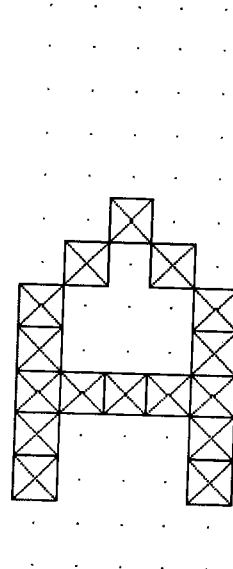
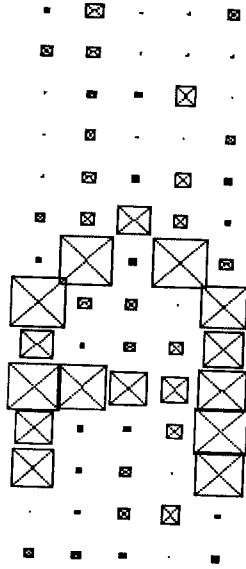
Để train với nhiễu có số epoch lớn nhất giảm đến 300 và lỗi cuối cùng giảm đến 0.006 phản ánh rằng lỗi cao hơn đã mong đợi cho nhiễu vector hơn. Bao gồm một vài nhiễu đã được trình bày.

```
netn = net;
netn.trainParam.goal = 0.006;
netn.trainParam.epochs = 300;
T = [targets targets targets targets];
for pass = 1:10
P = [alphabet, alphabet, ...
(alphabet + randn(R,Q)*0.1), ...
(alphabet + randn(R,Q)*0.2)];
[netn,tr] = train(netn,P,T);
end
```

Training không nhiễu một lần nữa: một lần nữa mạng đã được training với nhiễu, nó làm thông minh để train nó không nhiễu hơn một lần nữa để chắc chắn rằng luôn phân loại một cách đúng. Đó đó một lần nữa mạng được train với mã đúng ở phần “training without noise”.

Để mô phỏng và hiển thị ký tự tiếng Việt ta dùng hàm **baopluchar**, hàm **sim**, hàm **compet**.

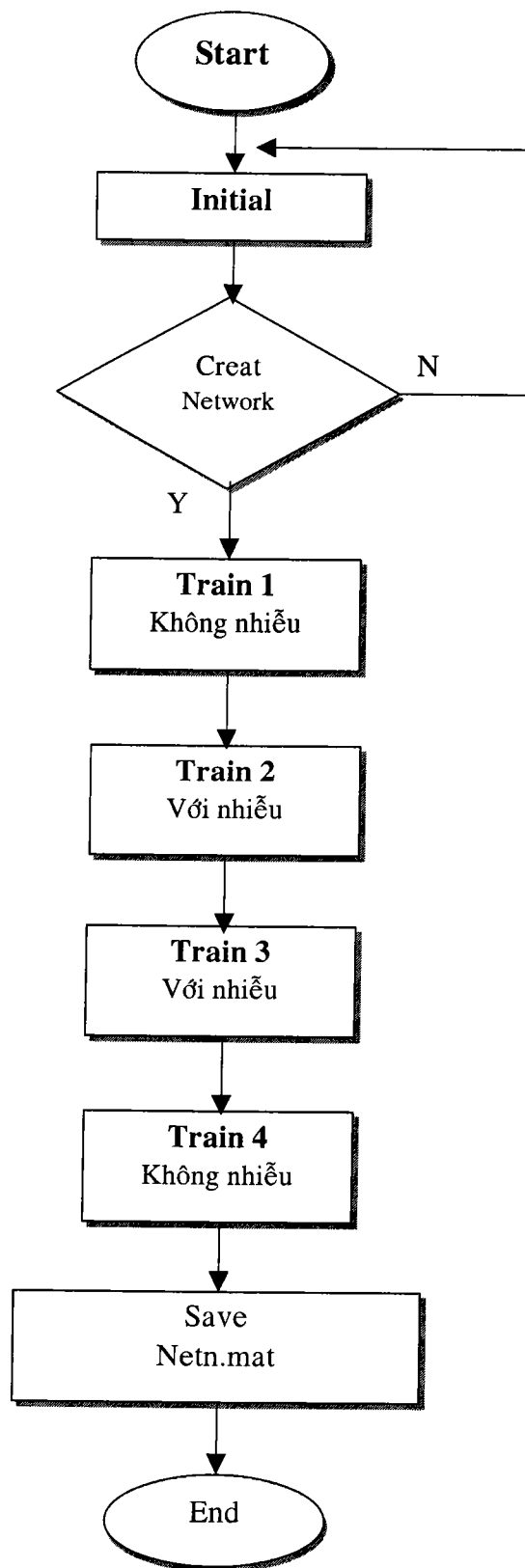
Hình sau trình bày ký tự nhiễu và ký tự nhận được mạng sau khi qua mạng Neural:



### 3.9. CHƯƠNG TRÌNH MÔ PHỎNG

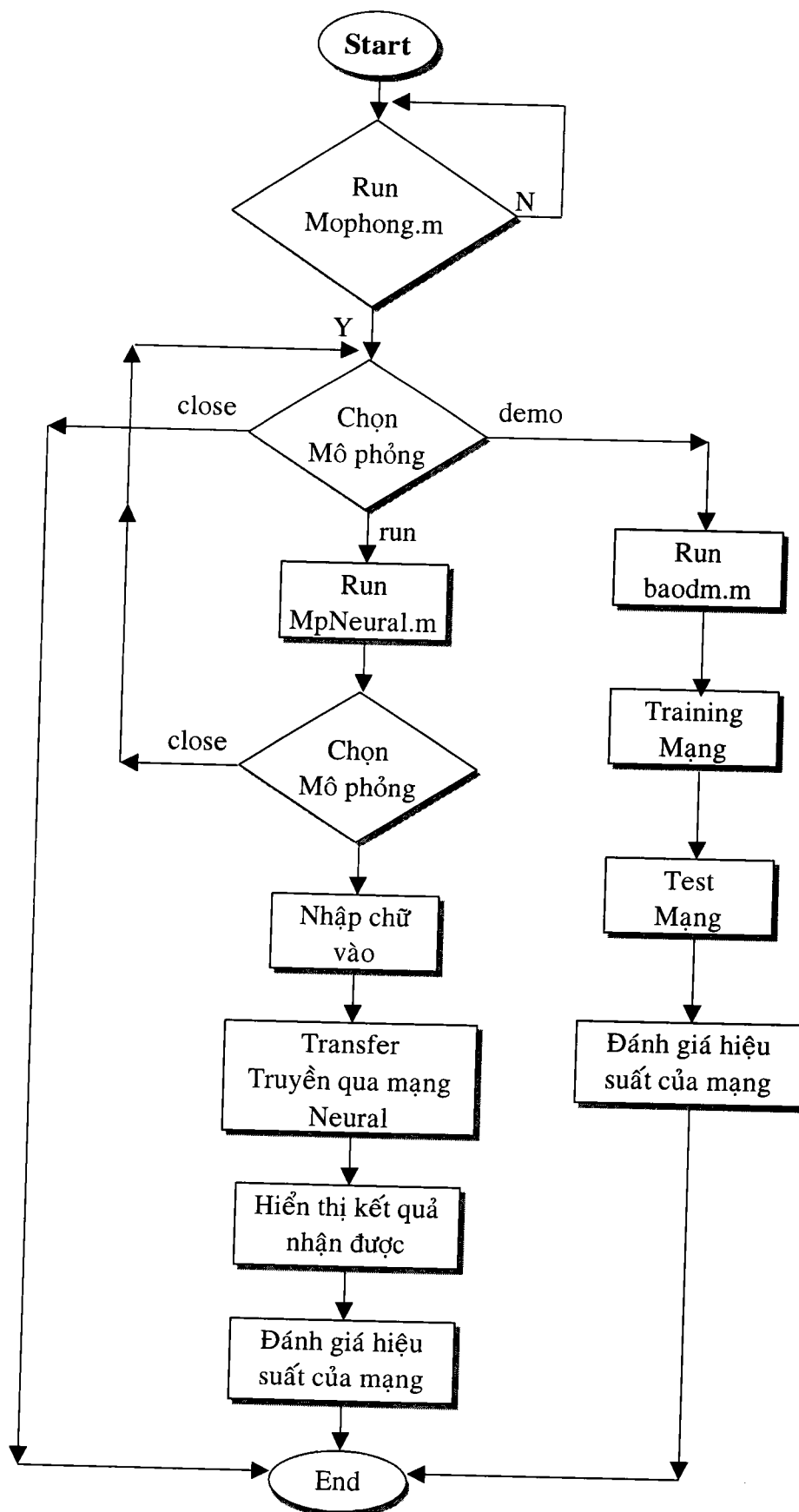
#### 3.9.1. LƯU ĐỒ

Ta có lưu đồ chương trình tạo mạng Neural như sau:

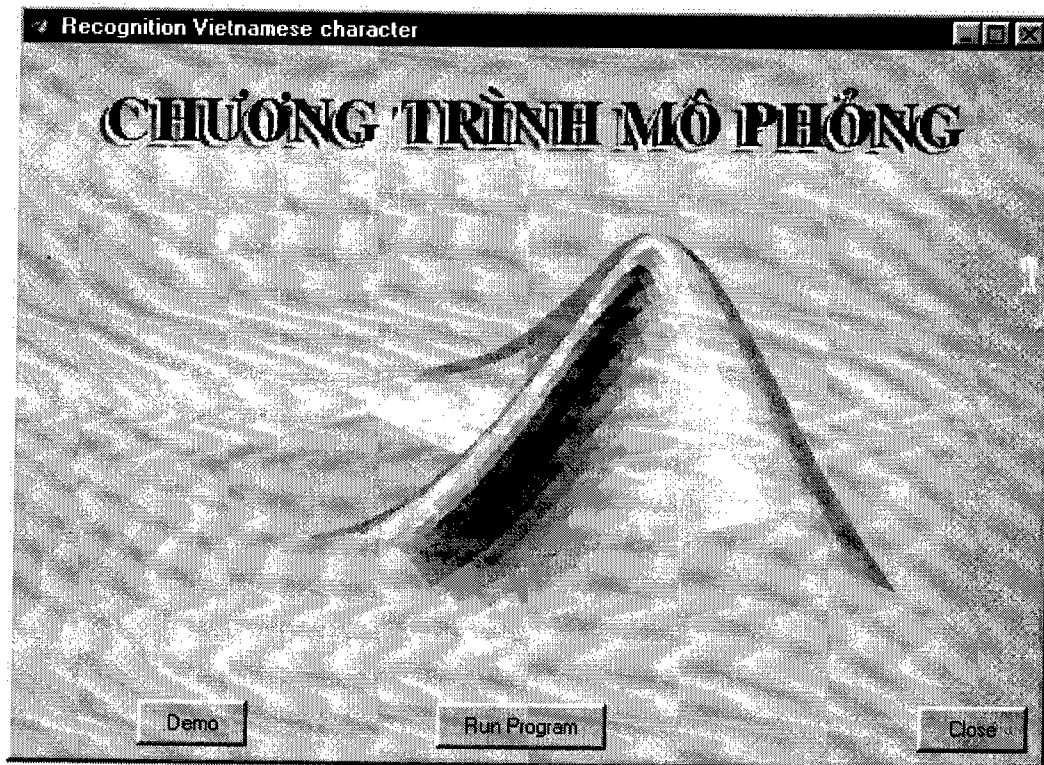
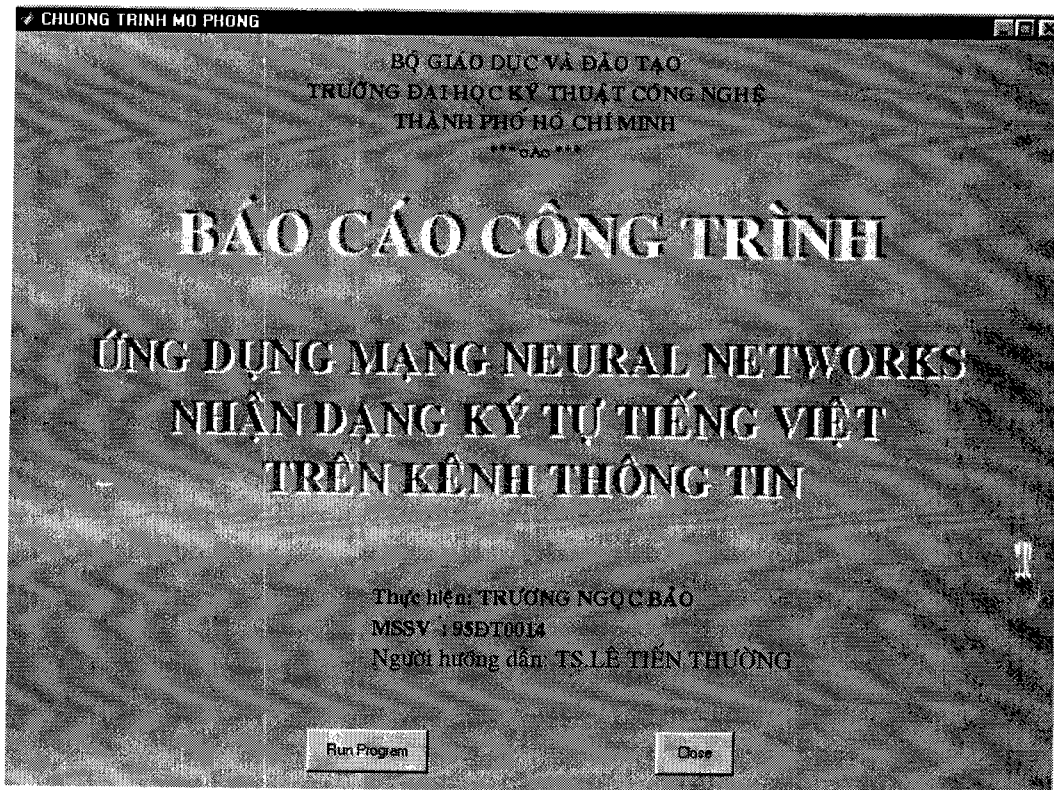


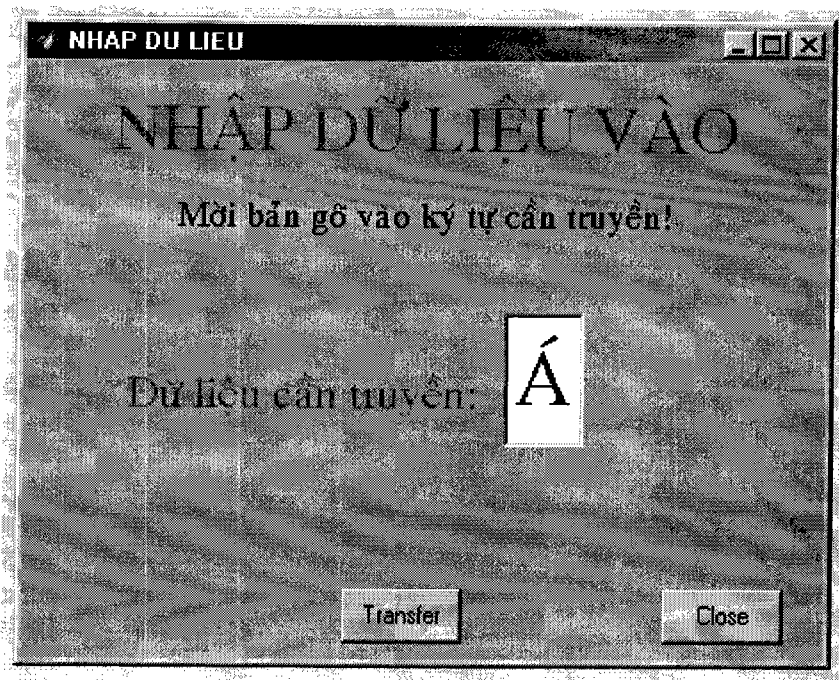


Ta có lưu đồ và màn hình chương trình Giao diện mô phỏng chính như sau:

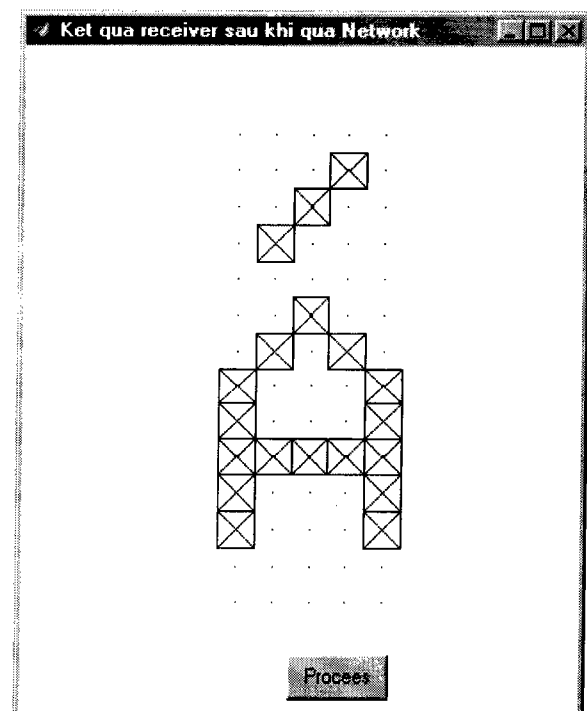
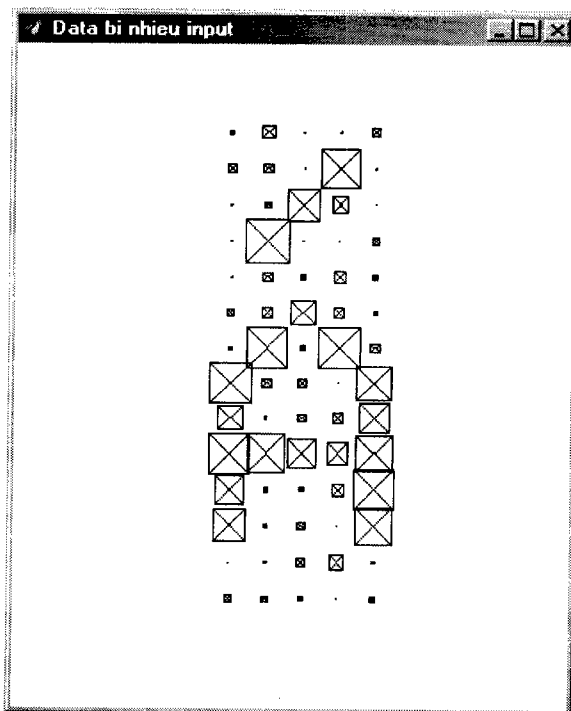


## 3.9.2. CHƯƠNG TRÌNH MÔ PHỎNG



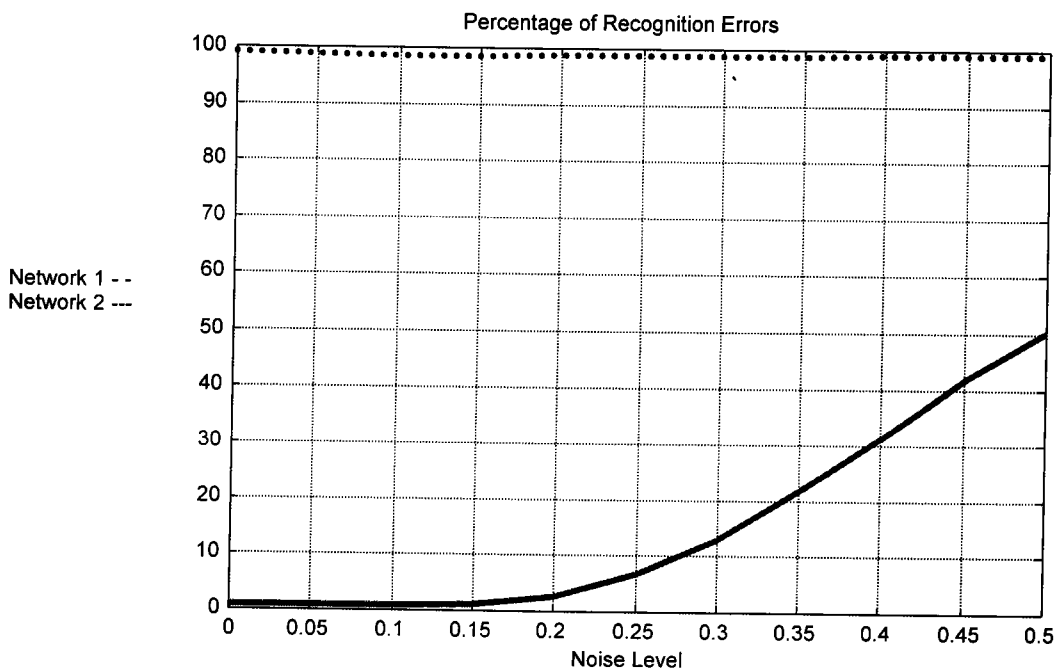


### 3.9.3. KẾT QUẢ NHẬN ĐƯỢC



### 3.10. ĐÁNH GIÁ MẠNG NEURAL

Độ tin cậy của mạng Neural Network trong hệ thống nhận dạng ký tự thì đã được kiểm tra bằng cách kiểm tra mạng với hàng trăm vector ngõ vào với những lượng nhiễu khác nhau. Điều này đã được mô phỏng trong chương trình. Chúng sẽ kiểm tra mạng với những mức độ nhiễu khác nhau và sau đó vẽ đồ thị theo tỉ lệ phần trăm của nhiễu trên tín hiệu, hay còn gọi là tỉ lệ S/N. nhiễu với mức độ từ 0 đến giá trị tiêu chuẩn thì được chia nhỏ thành từ 0 đến 0.5 và được cộng vào vector ngõ vào. Tại mỗi vạch nhiễu trên đồ thị, mức độ phần trăm khác nhau của nhiễu tại mỗi ký tự thì được làm và ngõ ra của mạng đã được tính toán. Ngõ ra sau đó được cho qua hàm truyền so sánh vì thế chỉ có một trong 203 ngõ ra, đại diện cho các ký tự trong bảng Alphabet, là có giá trị bằng 1 còn các ngõ ra còn lại có giá trị 0.



### 4. KẾT LUẬN VÀ ĐỀ NGHỊ:

Trong công trình này không có ý bao hàm hầu hết các tiến bộ trong mạng Neural đã được thực hiện trong vòng 10 năm gần đây trong lĩnh vực nhận dạng mẫu. Đề tài đã trình bày những khái niệm căn bản về mạng neural nhân tạo, những nguyên tắc thiết kế mạng và những nguyên tắc học tập. Ở đây, mạng được xây dựng dựa trên những hiểu biết của chúng ta về những hệ thống thần kinh hay nói cách khác là về những cách xử lý song song. Những lợi ích của mạng rất to lớn. Nó được áp dụng trong nhiều lĩnh vực khác nhau như nhận dạng tiếng nói, xử lý hình ảnh, điều khiển,... Chiều hướng thay đổi từ việc chỉ nhận dạng mẫu sang hiểu biết giữa ngữ cảnh và ngôn ngữ đan chéo nhau hơn là chỉ nhận dạng mẫu, vấn đề không chỉ nhận dạng mẫu mà còn hơn thế nữa là hiểu ngôn ngữ.

