

**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**THUYẾT MINH**  
**ĐỀ TÀI NCKH CẤP TRƯỜNG**

**ĐỀ TÀI**

**NGHIÊN CỨU VÀ XÂY DỰNG MẠNG NEURAL TỐI ƯU  
CHO KỸ THUẬT NHẬN DẠNG VĂN BẢN TIẾNG VIỆT**

**Chủ nhiệm đề tài: TH.S TRẦN ĐÌNH VƯƠNG**

**Thành viên tham gia: TH.S NGUYỄN HẠNH PHÚC**

**TH.S LÊ THẾ ANH**

**Hải Phòng, tháng 5/2015**

## MỤC LỤC

MỤC LỤC .....	2
MỞ ĐẦU .....	4
Chương I: TÌM HIỂU MẠNG NEURAL .....	5
1.1 Khái niệm mạng Neural.....	5
1.1.1 Neural sinh học .....	5
1.1.2. Neural nhân tạo.....	6
1.2 Mạng Neural nhân tạo.....	8
1.3 Đặc trưng của mạng neural .....	9
1.3.1. Tính phi tuyến .....	9
1.3.2. Tính chất tương ứng đầu vào đầu ra.....	9
1.3.3. Tính chất thích nghi.....	10
1.3.4. Tính chất đưa ra lời giải có bằng chứng.....	10
1.3.5. Tính chất chấp nhận sai sót.....	10
1.3.6. Khả năng cài đặt VLSI (Very Large Scale Intergrated).....	10
1.3.7. Tính chất đồng dạng trong phân tích và thiết kế.....	10
Chương II PHÂN LOẠI VÀ KIẾN TRÚC MẠNG NEURAL NHÂN TẠO.....	12
2.1 Phân loại mạng Neural nhân tạo .....	12
2.1.1. Phân loại theo kiểu liên kết neural .....	12
2.1.2. Một số loại mạng Neural .....	12
2.1.2.1. Mạng dẫn tiến.....	12
2.1.2.2. Mạng quy hồi.....	14
2.2 Kiến trúc mạng Neural.....	15
Chương III LÝ THUYẾT VỀ HUẤN LUYỆN MẠNG NEURAL.....	17
3.1 Phương pháp học .....	17
3.1.1. Học có giám sát .....	17
3.1.2. Học không giám sát.....	17
3.1.3. Học tăng cường. ....	17
3.2 Thuật toán học .....	18
3.2.1. Thuật toán học của mạng Neural một lớp.....	18

3.2.2. Thuật toán học của mạng Neural nhiều lớp .....	19
3.3 Thu thập dữ liệu cho mạng Neural .....	23
3.4. Biểu diễn tri thức cho mạng Neural.....	24
Chương IV THIẾT KẾ VÀ HUẤN LUYỆN MẠNG NEURAL NHÂN TẠO.....	27
4.1 Thiết kế mạng Neural nhân tạo.....	27
4.2. Lược đồ huấn luyện mạng .....	28
4.3. Thuật toán huấn luyện mạng Neural nhân tạo – Backpropagation.....	29
Chương V KẾT QUẢ HUẤN LUYỆN.....	31
5.1 Thu thập dữ liệu cho việc huấn luyện mạng Neural.....	31
5.2. Xây dựng mạng và huấn luyện .....	40
KẾT LUẬN .....	43
TÀI LIỆU THAM KHẢO .....	44

## MỞ ĐẦU

Từ lâu các nhà khoa học đã nhận thấy những ưu điểm của bộ óc con người và tìm cách bắt chước để thực hiện trên những máy tính, tạo cho nó có khả năng học tập, nhận dạng và phân loại. Vì vậy các nhà khoa học đã nghiên cứu và sáng tạo ra mạng Neural nhân tạo. Nó thực sự được chú ý và nhanh chóng trở thành một hướng nghiên cứu đầy triển vọng đặc biệt là lĩnh vực nhận dạng. Và bài toán nhận dạng ký tự là một bài toán con trong lớp các bài toán nhận dạng, xử lý ảnh.

Hiện nay trên thế giới, các sản phẩm nhận dạng ký tự đã được triển khai tương đối rộng rãi. Tuy nhiên đây là các sản phẩm nhận dạng ký tự tiếng Anh, do đó đối với nhận dạng ký tự tiếng Việt thì chỉ có người Việt Nam mới có thể phát triển được. Ở nước ta trong một vài năm gần đây cũng đã có một số sản phẩm nhận dạng tiếng Việt được triển khai trên thị trường. Nhưng các sản phẩm này được bán trên thị trường dưới dạng đóng kín nên việc để phát triển thành phần mềm tự động cập nhật ảnh là điều không thể. Vì vậy nên tôi đã chọn đề tài ***“Nghiên cứu và xây dựng mạng Neural tối ưu cho kỹ thuật nhận dạng văn bản tiếng Việt”***.

Hệ thống chữ cái Tiếng Việt là được xây dựng dựa trên chữ cái Latinh có thêm chữ ghép và dấu phụ. Do đó việc nhận dạng sẽ gặp khó khăn hơn so với chữ cái Latinh thông thường và cần phải có thuật toán xử lý đem lại độ chính xác cao.

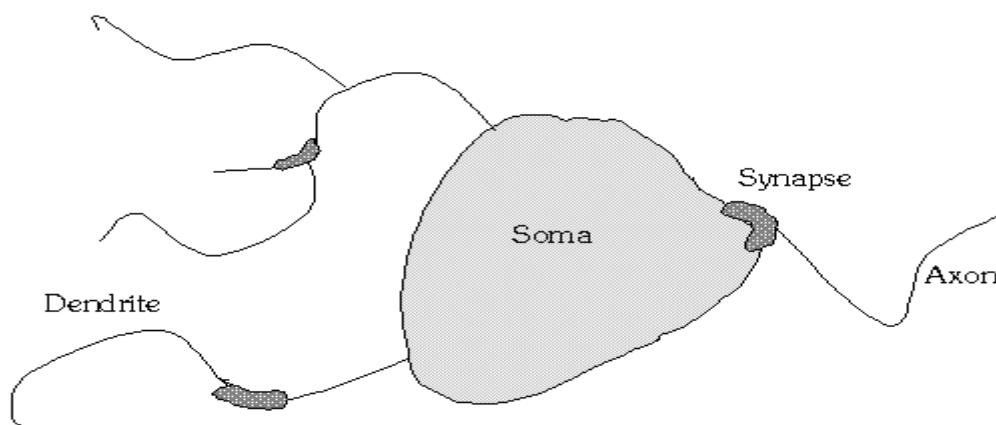
Trong khuôn khổ, thời lượng của luận văn, tôi đưa ra một chương trình mô phỏng mạng Neural nhận dạng 29 chữ cái Tiếng Việt từ A đến Y và 10 chữ số từ 0 đến 9.

## Chương I: TÌM HIỂU MẠNG NEURAL

### 1.1 Khái niệm mạng Neural

#### 1.1.1 Neural sinh học

Một neural được cấu gồm những thành phần chính sau: Dendrite, Soma, Synapse, Axon như hình 1.1.



Hình 1.1: Mô hình neural sinh học

Soma là thân của neural.

Các dendrites là các dây mảnh, dài, gắn liền với soma, chúng truyền dữ liệu (dưới dạng xung điện thế) đến cho soma xử lý. Bên trong soma các dữ liệu đó được tổng hợp lại, có thể xem gần đúng sự tổng hợp ấy như là một phép lấy tổng tất cả các dữ liệu mà neural nhận được.

Một loại dây dẫn tín hiệu khác cũng gắn với soma là các axon. Khác với dendrites, axons có khả năng phát các xung điện thế, chúng là các dây dẫn tín hiệu từ neural đi các nơi khác. Chỉ khi nào điện thế trong soma vượt quá một giá trị ngưỡng nào đó thì axon mới phát một xung điện thế, còn nếu không thì nó ở trạng thái nghỉ.

Axon nối với các dendrites của các neural khác thông qua những mối nối đặc biệt gọi là synapse. Khi điện thế của synapse tăng lên do các xung phát ra từ axon thì synapse sẽ nhả ra một số chất hoá học (neurotransmitters); các chất này mở "cửa" trên dendrites để cho các ions truyền qua. Chính dòng ions này làm thay đổi điện thế trên dendrites, tạo ra các xung dữ liệu lan truyền tới các neural khác.



nhân với trọng số này. Các trọng số liên kết chính là các tham số tự do cơ bản của mạng neuron, có thể thay đổi được nhằm thích nghi với môi trường xung quanh.

Bộ cộng dùng để tính tổng các tín hiệu đầu vào của neural, đã được nhân với các trọng số liên kết tương ứng. phép toán được mô tả ở đây tạo nên một bộ hợp tuyến tính.

Hàm kích hoạt hay còn gọi hàm kích hoạt phi tuyến, chuyển đổi một tổ hợp tuyến tính của tất cả các tín hiệu đầu vào thành tín hiệu đầu ra. Hàm kích hoạt này đảm bảo tính chất phi tuyến cho tính toán mạng neural. Nó được xem như là một hàm giới hạn, nó giới hạn phạm vi biên độ cho phép của tín hiệu đầu ra trong một khoảng giá trị hữu hạn. Mô hình neural trong hình 2.2 còn bao gồm một hệ số điều chỉnh b tác động từ bên ngoài. Hệ số điều chỉnh b có tác dụng tăng lên hoặc giảm đi đầu vào thực của hàm kích hoạt, tùy theo nó dương hay âm.

*Bảng 1.1 : Một số hàm kích hoạt cơ bản trong mạng neural:*

Tên hàm	Công thức
hardlim	$a = 0$ với $n < 0$ $a = 1$ với $n \geq 0$
hardlims	$a = -1$ với $n < 0$ $a = 1$ với $n \geq 0$
purelin	$a = n$
Satlin	$a = 0$ với $n < 0$ $a = n$ với $0 \leq n \leq 1$ $a = 1$ với $n > 1$
satlins	$a = -1$ với $n < 0$ $a = n$ với $0 \leq n \leq 1$ $a = 1$ với $n > 1$
tansig	$a = \frac{e^n - e^{-n}}{1 + e^{-n}}$

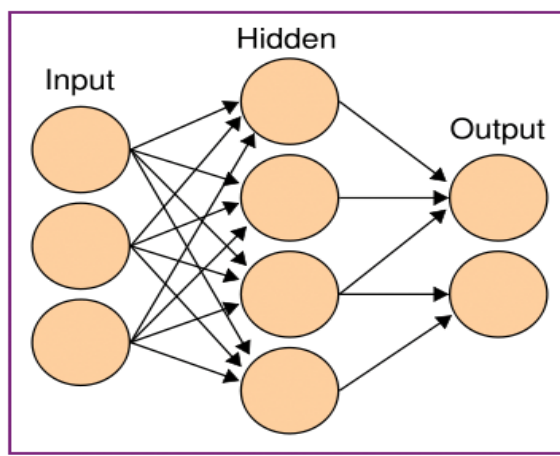
poslin	$a = 0$ với $n < 0$ $a = n$ với $n \geq 0$
compet	$a = 1$ với neural có $n$ lớn nhất $a = 0$ với các neural còn lại
<i>logsig</i>	$a = \frac{1}{1 + e^{-n}}$

## 1.2 Mạng Neural nhân tạo

Là một hệ thống bao gồm nhiều phần tử xử lý đơn giản (hay còn gọi là neural) tựa như neural thần kinh của não người, hoạt động song song và được nối với nhau bởi các liên kết neural. Mỗi liên kết kèm theo một trọng số nào đó, đặc trưng cho tính kích hoạt hoặc ức chế giữa các neural.

Có thể xem các trọng số là phương tiện để lưu trữ thông tin dài hạn trong mạng neural và nhiệm vụ của quá trình huấn luyện của mạng là cập nhật các trọng số khi có thêm thông tin về mẫu học. Hay nói một cách khác, các trọng số đều được điều chỉnh sao cho đáng điệu vào ra của mạng sẽ mô phỏng hoàn toàn phù hợp với môi trường đang xem xét.

Mô hình mạng neural.



Hình 1.3: Sơ đồ đơn giản về một mạng neural nhân tạo



Mô hình mạng neural ở trên gồm 3 lớp: lớp nhập (input), lớp ẩn(hidden) và lớp xuất (output). Mỗi nút trong lớp nhập nhận giá trị của một biến độc lập và chuyển vào mạng.

Dữ liệu từ tất cả các nút trong lớp nhập được tích hợp – ta gọi là tổng trọng số – và chuyển kết quả cho các nút trong lớp ẩn. Gọi là “ẩn” vì các nút trong lớp này chỉ liên lạc với các nút trong lớp nhập và lớp xuất, và chỉ có người thiết kế mạng mới biết lớp này (người sử dụng không biết lớp này).

Các nút trong lớp xuất nhận các tín hiệu tổng trọng hóa từ các nút trong lớp ẩn. Mỗi nút trong lớp xuất tương ứng với một biến phụ thuộc.

### **1.3 Đặc trưng của mạng neural**

#### *1.3.1. Tính phi tuyến*

Một neural có thể tính toán một cách tuyến tính hay phi tuyến. Một mạng neural, cấu thành bởi sự kết nối các neural 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 một thuộc tính rất quan trọng, nhất là khi các cơ chế vật lý sinh ra các tín hiệu đầu vào (ví dụ tín hiệu tiếng nói) vốn là phi tuyến.

#### *1.3.2. Tính chất tương ứng đầu vào đầu ra*

Mặc dù khái niệm “học” hay “huấn luyện” chưa được bàn đến nhưng để hiểu được mối quan hệ đầu vào-đầu ra của mạng neural, chúng ta sẽ đề cập sơ qua về khái niệm này. 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 liên kết của mạng neural bằng việc áp dụng một tập hợp các mẫu tích lũy hay các ví dụ tích lũy. Mỗi một ví dụ bao gồm một tín hiệu đầu vào và một đầu ra mong muốn tương ứng. Mạng neural nhận một ví dụ lấy một cách ngẫu nhiên từ tập hợp nói trên tại đầu vào của nó, và các trọng số liên kết của mạng được biến đổi sao cho có thể cực tiểu hoá sự sai khác giữa đầu ra mong muốn và đầu ra thực sự của mạng theo một tiêu chuẩn thống kê thích hợp. Sự tích lũy của mạng được lặp lại với nhiều ví dụ trong tập hợp cho tới khi mạng đạt tới một trạng thái ổn định mà ở đó không có một sự thay đổi đáng kể nào của các trọng số liên kết. Các ví dụ tích lũy được áp dụng trước có thể được áp dụng lại trong thời gian của

phiên tích lũy nhưng theo một thứ tự khác. Như vậy mạng neural học từ các ví dụ bằng cách xây dựng nên một tương ứng đầu vào-đầu ra cho vấn đề cần giải quyết.

### *1.3.3. Tính chất thích nghi*

Các mạng neural có một khả năng mặc định là biến đổi các trọng số liên kết tùy theo sự thay đổi của môi trường xung quanh. Đặc biệt, một mạng neural đã được tích lũy để hoạt động trong một môi trường xác định có thể được tích lũy lại một cách dễ dàng khi có những thay đổi nhỏ của các điều kiện môi trường hoạt động.

### *1.3.4. Tính chất đưa ra lời giải có bằng chứng*

Trong ngữ cảnh phân loại mẫu, một mạng neural 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.

### *1.3.5. Tính chất chấp nhận sai sót*

Một mạng neural, 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 hoá khi có những điều kiện hoạt động bất lợi. Ví dụ, nếu một neural hay các liên kết kết nối của nó bị hỏng, việc nhận dạng lại một mẫu được lưu trữ sẽ suy giảm về chất lượng.

### *1.3.6. Khả năng cài đặt VLSI (Very Large Scale Intergrated)*

Bản chất song song đồ sộ của một mạng neural làm cho nó rất nhanh trong tính toán đối với một số công việc. Đặc tính này cũng tạo ra cho một mạng neural khả năng phù hợp cho việc cài đặt sử dụng kỹ thuật Very-large-scale-intergrated (VLSI). Kỹ thuật này cho phép xây dựng những mạch cứng tính toán song song quy mô lớn. Chính vì vậy mà ưu điểm nổi bật của VLSI là mang lại những phương tiện hữu hiệu để có thể xử lý được những hành vi có độ phức tạp cao.

### *1.3.7. Tính chất đồng dạng trong phân tích và thiết kế*

Về cơ bản, các mạng neural có tính chất chung như là các bộ xử lý thông tin. Chúng ta nêu ra điều này với cùng ý nghĩa cho tất cả các lĩnh vực có liên quan tới việc ứng dụng mạng neural. Đặc tính này thể hiện ở một số điểm như sau:

Các neural, dưới dạng này hoặc dạng khác, biểu diễn một thành phần chung cho tất cả các mạng neural.

Tính thống nhất này đem lại khả năng chia sẻ các lý thuyết và các thuật toán học trong nhiều ứng dụng khác nhau của mạng neural.

Các mạng tổ hợp (modular) có thể được xây dựng thông qua một sự tích hợp các mô hình khác nhau.

## Chương II PHÂN LOẠI VÀ KIẾN TRÚC MẠNG NEURAL NHÂN TẠO

### 2.1 Phân loại mạng Neural nhân tạo

#### 2.1.1. Phân loại theo kiểu liên kết neural

Ta có mạng neural truyền thẳng và neural mạng qui hồi.

Trong mạng truyền thẳng các neural đi theo một hướng nhất định tạo thành đồ thị không có chu trình, các đỉnh là các neural còn các cạnh là các liên kết giữa chúng.

Các mạng qui hồi cho phép các liên kết neural tạo thành chu trình, các thông tin ra của các neural được truyền lại cho các neural đã góp phần kích hoạt chúng, nên mạng qui hồi còn có khả năng lưu giữ trạng thái trong của nó dưới dạng các ngưỡng kích hoạt ngoài các trọng số liên kết neural.

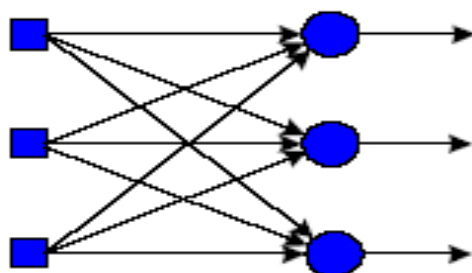
#### 2.1.2. Một số loại mạng Neural

##### 2.1.2.1. Mạng dẫn tiến

Có thể nói mạng neural dẫn tiến là một kiểu mạng đơn giản trong việc sắp đặt mạng. Trong mạng này thông tin chỉ truyền trên một hướng duy nhất. từ lớp đầu vào xuyên qua lớp ẩn (nếu có) và kết thúc tại lớp đầu ra. Không có chu trình hoặc vòng trong mạng.

##### a. Các mạng dẫn tiến đơn mức.

Trong một mạng neural phân mức, các neural được tổ chức dưới dạng các mức. Với dạng đơn giản nhất của mạng phân mức, chúng ta có một mức đầu vào gồm các nút nguồn chiếu trực tiếp tới mức đầu ra gồm các neural.



Mức đầu vào

Mức đầu ra

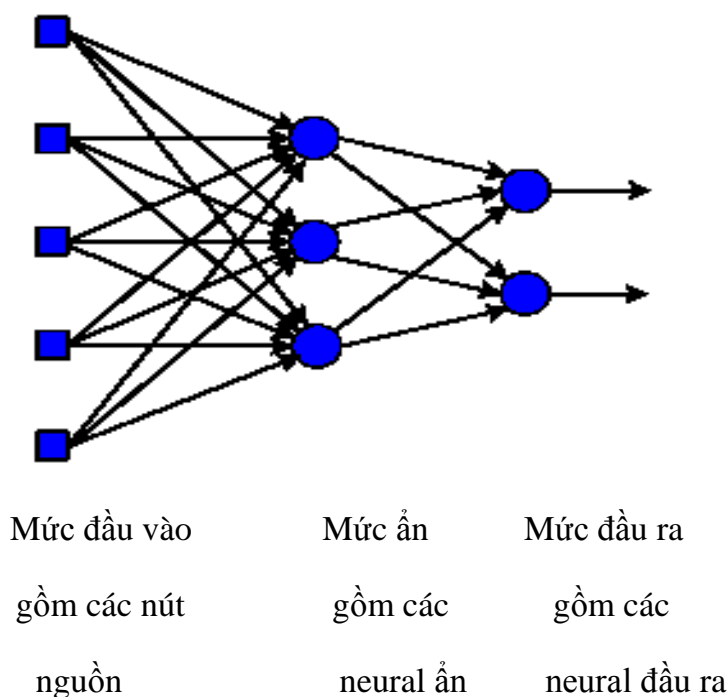
Hình 2.1: Mạng tiến với một mức neural

Như vậy, mạng thực sự là không có chu trình. Nó được minh họa trong hình 2.4 cho trường hợp ba nút đối với cả mức đầu ra và đầu vào. Một mạng như vậy được gọi là một mạng đơn mức. “Đơn mức” tức là chỉ có một mức, chính là mức đầu ra gồm các nút tính toán (các neural). Chúng ta không tính mức đầu vào của các nút nguồn vì không có tính toán nào được thực hiện ở đây.

**b. Các mạng dẫn tiến đa mức.**

Lớp thứ hai của một mạng neural dẫn tiến được phân biệt bởi sự có mặt của một hay nhiều mức ẩn, mà các nút tính toán của chúng được gọi là các neural ẩn hay các đơn vị ẩn (thuật ngữ “ẩn” ở đây mang ý nghĩa là không tiếp xúc với môi trường). Chức năng của các neural ẩn là can thiệp vào giữa đầu vào và đầu ra của mạng một cách hữu hiệu. Bằng việc thêm một vài mức ẩn, mạng có khả năng rút ra được các thông kê bậc cao của tín hiệu đầu vào. Khả năng các neural ẩn rút ra được các thông kê bậc cao đặc biệt có giá trị khi mức đầu vào có kích thước lớn.

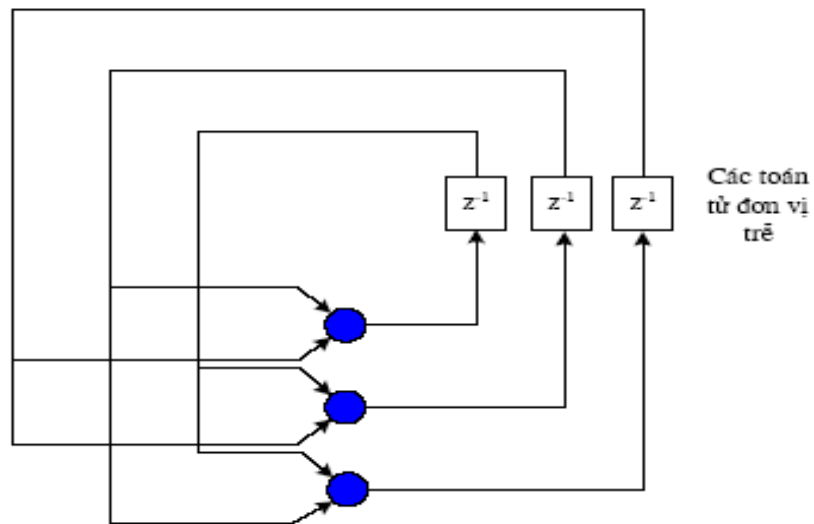
Mạng neural trong hình 2.5 được gọi là kết nối đầy đủ với ý nghĩa là tất cả các nút trong mỗi mức của mạng được nối với tất cả các nút trong mức tiếp sau. Nếu một số kết nối không tồn tại trong mạng, chúng ta nói rằng mạng là kết nối không đầy đủ.



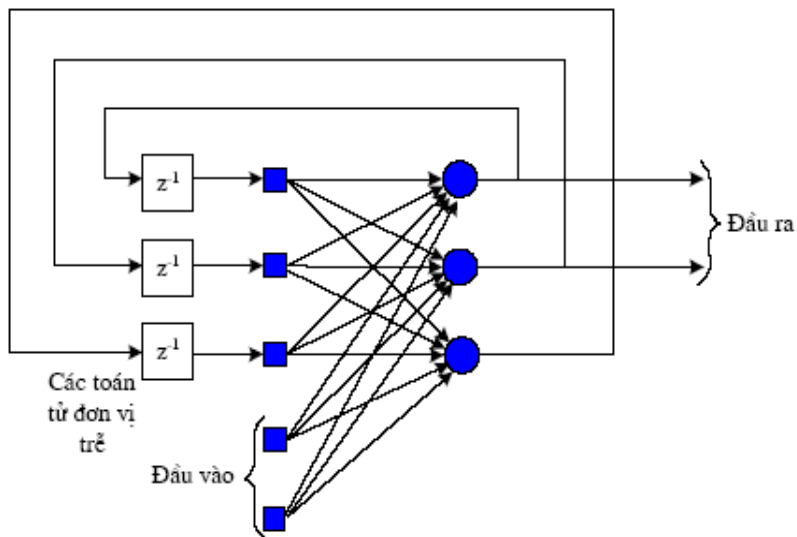
Hình 2.2: Mạng tiến kết nối đầy đủ với một mức ẩn và một mức đầu ra.

2.1.2.2. Mạng quy hồi

Trái với mạng neural dẫn tiến, mạng nơon quy hồi là những mô hình với hai luồng dữ liệu có hướng. Trong khi mạng dẫn tiến truyền dữ liệu theo một đường thẳng thì những mạng neural quy hồi có ít nhất một phản hồi từ những neural xử lý sau quay trở lại các neural xử lý trước đó.



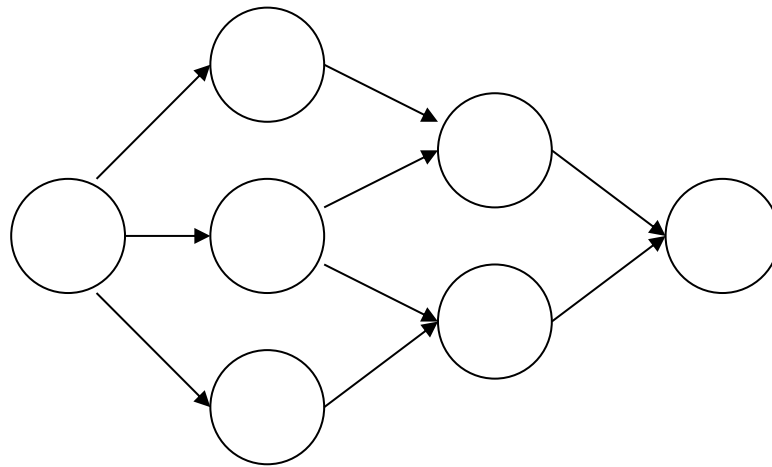
Hình 2.3: Mạng hồi quy không có neural ẩn và không có vòng lặp tự phản hồi



Hình 2.4: Mạng hồi quy có các neural ẩn

## 2.2 Kiến trúc mạng Neural

Về cơ bản ta có thể hiểu mạng neural là một đồ thị có hướng như hình 2.8. Trong đó các đỉnh của đồ thị là các neural và các cạnh của đồ thị là các liên kết giữa các neural.



Hình 2.5: Sơ đồ đồ thị có hướng đơn giản

Vì vậy để xây dựng một mạng neural ta xây dựng một đồ thị có hướng: số đỉnh của đồ thị bằng số neural trong mạng, giá trị của các cạnh chính là trọng số liên kết neural.

### Ví dụ xây dựng một mạng neural đơn giản:

Đầu vào: Cho :

Mạng neural có số lớp (với số lớp > 1).

Mỗi lớp có số neural trong một lớp (số neural  $\geq 1$ ).

Trọng số  $w$  liên kết neural ngẫu nhiên trong khoảng  $(-a, a)$ .

Đầu ra: Mạng neural nhân tạo.

Kiểu dữ liệu: chọn lưu dữ kiểu mảng :

```
int số lớp;
```

```
int số neural[số lớp];
```

```
float w[i][j][k]; với  $2 \leq i \leq \text{số lớp}, 1 \leq j, k \leq \text{số neural}[]$  ,
```

Thuật toán.

Bước 1: Chọn lớp  $i=2$  là lớp bắt đầu

Bước 2: Chọn lớp  $i$  là lớp hiện thời.

Bước 3 : Tại lớp đang xét  $i$  , xét neural thứ  $j$  .

Bước 4: Thực hiện khởi tạo ngẫu nhiên trọng số kết nối với neural  $k$  của lớp  $i-1$  trong khoảng  $(-a,a)$ .

$$w(i,j,k)=\text{random}(-a,a)$$

Bước 5: Nếu  $k \leq \text{số neural}[i-1]$  quay lại “Bước 4”, ngược lại thực hiện “Bước 6”.

Bước 6: Nếu  $j \leq \text{số neural}[i]$  quay lại “Bước 3”, ngược lại thực hiện “Bước 7”.

Bước 7: Nếu  $i \leq \text{số lớp}$  quay lại “Bước 3”. ngược lại kết thúc.



## Chương III LÝ THUYẾT VỀ HUẤN LUYỆN MẠNG NEURAL

### 3.1 Phương pháp học

Mạng neural nhân tạo phỏng theo việc xử lý thông tin của bộ não người, do vậy đặc trưng cơ bản của mạng là có khả năng học, khả năng tái tạo các hình ảnh và dữ liệu khi đã học. Trong trạng thái học thông tin được lan truyền theo hai chiều nhiều lần để học các trọng số. Có 3 kiểu học chính, mỗi kiểu học tương ứng với một nhiệm vụ học trừu tượng. Đó là học có giám sát (có mẫu), học không giám sát và học tăng cường. Thông thường loại kiến trúc mạng nào cũng có thể dùng được cho các nhiệm vụ.

#### 3.1.1. Học có giám sát

Một thành phần không thể thiếu của phương pháp này là sự có mặt của một người thầy (ở bên ngoài hệ thống). Người thầy này có kiến thức về môi trường thể hiện qua một tập hợp các cặp đầu vào - đầu ra đã được biết trước. Hệ thống học (ở đây là mạng neural) sẽ phải tìm cách thay đổi các tham số bên trong của mình (các trọng số và các ngưỡng) để tạo nên một ánh xạ có khả năng ánh xạ các đầu vào thành các đầu ra mong muốn. Sự thay đổi này được tiến hành nhờ việc so sánh giữa đầu ra thực sự và đầu ra mong muốn.

#### 3.1.2. Học không giám sát.

Trong học không có giám sát, ta được cho trước một số dữ liệu  $x$  và hàm chi phí cần được cực tiểu hóa có thể là một hàm bất kỳ của dữ liệu  $x$  và đầu ra của mạng,  $f$  – hàm chi phí được quyết định bởi phát biểu của bài toán. Phần lớn các ứng dụng nằm trong vùng của các bài toán ước lượng như mô hình hóa thống kê, nén, lọc, phân cụm.

#### 3.1.3. Học tăng cường.

Dữ liệu  $x$  thường không được tạo trước mà được tạo ra trong quá trình một agent tương tác với môi trường. Tại mỗi thời điểm  $t$ , agent thực hiện hành động  $y_t$  và môi trường tạo một quan sát  $x_t$  với một chi phí tức thời  $C_t$ , theo một quy trình động nào đó (thường là không được biết). Mục tiêu là một sách lược lựa chọn hành động để

Thuyết minh đề tài NCKH Chương III LÝ THUYẾT VỀ HUẤN LUYỆN MẠNG NEURAL cực tiểu hóa một chi phí dài hạn nào đó, nghĩa là chi phí tích lũy mong đợi. Quy trình hoạt động của môi trường và chi phí dài hạn cho mỗi sách lược thường không được biết, nhưng có thể ước lượng được. Mạng neural nhân tạo thường được dùng trong học tăng cường như một phần của thuật toán toàn cục. Các bài toán thường được giải quyết bằng học tăng cường là các bài toán điều khiển, trò chơi và các nhiệm vụ quyết định tuần tự (sequential decision making) khác.

### 3.2 Thuật toán học

#### 3.2.1. Thuật toán học của mạng Neural một lớp

Xét trường hợp perceptron sử dụng hàm kích hoạt ngưỡng:

$$output = \begin{cases} +1, & \sum_{i=1}^d v_i w_i > t \\ -1, & \sum_{i=1}^d v_i w_i \leq t \end{cases} \quad (1.1)$$

Nếu ta cho  $w_0 = -t$  và  $v_0 = 1$ , ta có thể viết lại

$$output = \begin{cases} +1, & \sum_{i=0}^d v_i w_i > 0 \\ -1, & \sum_{i=0}^d v_i w_i \leq 0 \end{cases} = \text{sign} \left( \sum_{i=0}^d v_i w_i \right) \quad (1.2)$$

Thuật toán:

Thuật toán học của perceptron hay mạng neural một lớp gồm 3 bước chính:

Bước 1: Khởi tạo:

Khởi tạo trọng số  $w$  (0 hoặc ngẫu nhiên).

Chọn tốc độ học  $\mu$ .

Bước 2 : Học

Với mỗi mẫu  $(x,t)$  trong tập học.

Tính  $y=f(x, w)$ .

Nếu  $y \neq t$  thay đổi vector trọng số  $w$  với:

$$w(\text{mới}) = w(\text{cũ}) + \mu(t - y)x;$$

Bước 3: lặp lại bước 2 cho tất cả các mẫu.

Nhận xét:

Phương trình  $w \cdot v = 0$  là chính là siêu phẳng trong không gian  $d$ -chiều, suy ra perceptron có khả năng phân lớp tuyến tính, có khả năng giải bài toán quy hồi tuyến tính.

Hạn chế:

Không thể phân lớp phi tuyến.

Giải pháp: sử dụng mạng neural nhiều tầng MLP

### 3.2.2. Thuật toán học của mạng Neural nhiều lớp

Huấn luyện mạng neural nhiều lớp sử dụng thuật toán Lan truyền ngược gồm hai quá trình: Quá trình truyền tuyến tính và quá trình truyền ngược:

Quá trình truyền tuyến tính: Dữ liệu từ lớp nhập qua lớp ẩn và đến lớp xuất để:

- ✓ Thay đổi giá trị của trọng số liên kết  $W$  của các neural trong mạng biểu diễn được dữ liệu học.
- ✓ Tìm ra sự khác nhau giữa giá trị thật hàm mẫu mà mạng tính được và kết quả dự đoán của mạng gọi là lỗi (học có giám sát).

Quá trình truyền ngược: Giá trị lỗi sẽ được truyền ngược lại sao cho quá trình huấn luyện(học) sẽ tìm ra trọng số  $W_i$  để lỗi nhỏ nhất.

**Thuật toán:**

Xét mạng neural 3 lớp : input, hidden và output.

Hàm kích hoạt các neural : logistic (sigmoid).

$$g(x) = \frac{1}{1 + e^{-x}} \quad (1.3)$$

Bước 1: Khởi tạo trọng số  $w$  ngẫu nhiên nhỏ.

Bước 2 : Chọn lớp đầu vào  $x$  và đính nó vào lớp đầu vào .

Bước 3 : Truyền sự kích hoạt qua trọng số đến khi kích hoạt các neural đầu ra.

Bước 4: Tính đạo hàm riêng “ $\delta E$ ” theo trọng số cho lớp xuất. sử dụng GRADIENT của hàm lỗi .

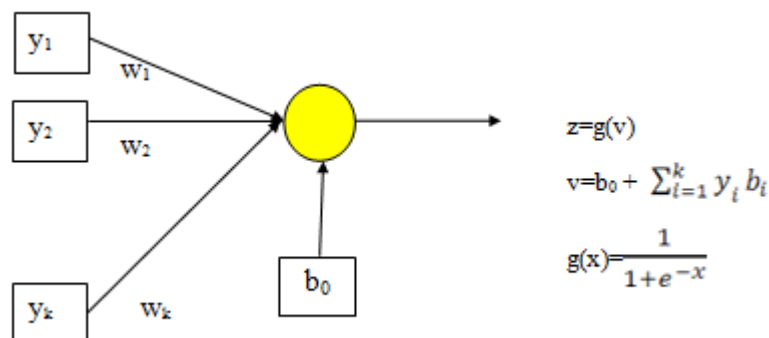
Bước 5: Tính đạo hàm riêng “ $\delta E$ ” theo trọng số cho lớp ẩn. sử dụng GRADIENT của hàm lỗi.

Bước 6: Cập nhật tất cả trọng số theo  $W_i = W_{(i-1)} - \mu(\delta E / \delta W_i)$

Bước 7: quay lại “Bước 2” cho đến “Bước 6” cho tất cả các mẫu.

Trong đó: GRADIENT hàm lỗi:

Đối với các neural lớp đầu ra:



Hàm lỗi:

$$E(z) = \frac{1}{2} (z - 1)^2 \tag{1.4}$$

(t là giá trị thật của mẫu huấn luyện)

Đạo hàm riêng của hàm lỗi theo trọng số b:

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z} \frac{\partial z}{\partial v} \frac{\partial v}{\partial b} \tag{1.5}$$

Ta có:

$$\frac{\partial E}{\partial z} = (z - t)$$

$$\frac{\partial z}{\partial v} = \frac{\partial g}{\partial v} = z(1 - z) \quad \text{Đặt:}$$

$$\frac{\partial v}{\partial b_1} = \begin{cases} 1, & i = 0 \\ y_i, & i > 0 \end{cases}$$

$$p = \frac{\partial E}{\partial z} \frac{\partial z}{\partial v} = (z - t)z(1 - z) \text{ Ta được:}$$

$$\frac{\partial E}{\partial b_i} = \begin{cases} p & i = 0 \\ py_i & i > 0 \end{cases} \quad (1.6)$$

Đối với các neural lớp ẩn:

Đạo hàm riêng của hàm lỗi theo trọng số:

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial a} \quad (1.7)$$

Ta có:

$$\frac{\partial E}{\partial y} = \sum_{i=1}^K \frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial v_i} \frac{\partial v_i}{\partial y}$$

$$\frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial v_i} = (z_i - t_i)z_i(1 - z_i) = p_i$$

$$\frac{\partial v_i}{\partial y} = b_i$$

$$\frac{\partial y}{\partial u} = y(1 - y)$$

$$\frac{\partial u}{\partial a_j} = \begin{cases} 1, & j = 0 \\ c_j, & j > 0 \end{cases}$$

Đặt:

$$q = \frac{\partial E}{\partial y} \frac{\partial y}{\partial u} = \left( \sum_{i=1}^K p_i b_i \right) y(1-y)$$

Ta được:

$$\frac{\partial E}{\partial a_1} = \begin{cases} q & i = 0 \\ qx_i & i > 0 \end{cases} \quad (1.8)$$

Sau khi tính được đạo hàm riêng của hàm lỗi theo từng trọng số, trọng số sẽ được điều chỉnh bằng cách trừ bớt đi 1 lượng bằng tích của đạo hàm riêng và tốc độ học:

$$w_i = w_i - \mu \frac{\partial E}{\partial w_i} \quad (1.9)$$

### Những hạn chế của phương pháp lan truyền ngược:

Ngoài những thành công của giải thuật học lan truyền ngược, vẫn còn có một số khía cạnh làm cho giải thuật trở nên chưa được bảo đảm là mọi lúc đều tốt. Khó khăn chủ yếu là ở quá trình huấn luyện lâu. Có thể do nhịp độ học và động lực không tối ưu. Sự sai sót trong việc huấn luyện nói chung xuất hiện từ hai nguồn: mạng liệt và những cực tiểu địa phương.

**Mạng liệt:** xảy ra khi những trọng số được điều chỉnh tới những giá trị rất lớn. Tổng đầu vào của một đơn vị ẩn hoặc đơn vị đầu ra có thể bởi vậy mà đạt giá trị rất cao (hoặc dương hoặc âm), và qua hàm kích hoạt sigmoid, đơn vị sẽ có một giá trị kích hoạt rất gần 0 hoặc rất gần 1. Giá trị hiệu chỉnh trọng số gần 0, và quá trình huấn luyện có thể đi đến một trạng thái dừng ảo.

**Cực tiểu địa phương:** bề mặt sai số của mạng rất phức tạp đầy những ngọn đồi và những thung lũng. Bởi vì sự hạ độ dốc, mạng có thể bị mắc bẫy tại một cực tiểu địa phương khi có nhiều cực tiểu thấp hơn gần bên cạnh. Những phương pháp thống kê có thể giúp để tránh cái bẫy này, nhưng chúng làm chậm. Một phương án khác là tăng thêm số lượng đơn vị ẩn. Như vậy sẽ làm việc trong không gian sai số nhiều chiều, nên cơ hội gặp bẫy nhỏ hơn. Tuy nhiên việc tăng cũng có giới hạn trên, khi vượt qua giới hạn này, cơ hội mắc bẫy lại tăng lên

### 3.3 Thu thập dữ liệu cho mạng Neural

Một khi ta quyết định giải quyết một vấn đề sử dụng những mạng neural ta cần phải thu thập dữ liệu cho mục tiêu huấn luyện. Tập hợp dữ liệu huấn luyện bao gồm một số trường hợp, mỗi trường hợp chứa những giá trị của đầu vào và đầu ra khác nhau. Những việc đầu tiên cần làm là: những biến nào được sử dụng, bao nhiêu trường hợp cần thu thập.

Sự lựa chọn do trực giác quyết định. Công việc chuyên môn của ta trong lĩnh vực cần giải quyết sẽ cho ta những ý tưởng về các biến đầu vào phù hợp. Trong những mạng neural ta có thể chọn và loại bỏ nhiều biến và mạng neural cũng có thể xác định bằng quá trình thực nghiệm. Trong một bước ta nên tính đến bất kỳ biến nào mà ta nghĩ có ảnh hưởng đến quá trình thiết kế.

Những mạng neural xử lý dữ liệu số trong một tầm giới hạn rõ ràng. Điều này đưa ra một vấn đề, nếu dữ liệu nằm trong một vùng đặc biệt như dữ liệu chưa biết hay không phải dữ liệu số. Thì phải có những phương pháp xử lý thích hợp, chẳng hạn. Dữ liệu số được chia nhỏ thành những khoảng thích hợp cho mạng và những giá trị thiếu có thể thay thế bằng giá trị trung bình hay giá trị thống kê của biến đó thông qua những biến khác đã được huấn luyện.

Xử lý dữ liệu không phải số thì khó hơn. Loại dữ liệu không phải là số thông thường nhất là những biến có giá trị định danh như giới tính(nam, nữ). Biến định danh có thể biểu diễn bằng số học và mạng neural có chức năng hỗ trợ điều này. Tuy nhiên, mạng neural làm việc tốt với những trường hợp biến định danh là một tập nhiều giá trị.

Số trường hợp mẫu dùng để huấn luyện mạng rất khó xác định. Đã có một vài hướng dẫn về mối liên hệ giữa số trường hợp mẫu với kích thước mạng(cách đơn giản nhất là số trường hợp mẫu gấp 10 lần số kết nối trong mạng). Thực ra số trường hợp mẫu cũng có liên quan đến độ phức tạp của hàm mà mạng phải học. Khi số biến tăng lên, số trường hợp mẫu cần huấn luyện cũng tăng phi tuyến, vì thế với số các biến nhỏ(50 hoặc nhỏ hơn) thì lại cần một số lớn các trường hợp mẫu.

Trong hầu hết các vấn đề trong thực tế, số trường hợp mẫu là khoảng hàng trăm hay hàng nghìn mẫu. Đối với những vấn đề rất phức tạp thì cần nhiều hơn, nhưng trường hợp này rất ít. Nếu dữ liệu huấn luyện ít hơn, rõ ràng không đủ thông tin để

Thuyết minh đề tài NCKH Chương III LÝ THUYẾT VỀ HUẤN LUYỆN MẠNG NEURAL huấn luyện mạng, và cách tốt nhất là dùng mạng tuyến tính. Nhiều vấn đề trong thực tế có dữ liệu không đáng tin cậy, một vài dữ liệu bị phá hỏng do nhiễu, hoặc các giá trị không phối hợp được với nhau. Mạng neural có khả năng đặc biệt xử lý dữ liệu bị mất (sử dụng giá trị trung bình hay các những giá trị thống kê khác). Mạng neural cũng chịu được nhiễu, nhưng cũng phải có giới hạn. Nếu thỉnh thoảng có giá trị nằm xa ra khỏi vùng giá trị bình thường thì mạng huấn luyện phải có ngưỡng. Cách tốt nhất đối với trường hợp này là nhận ra và loại bỏ những giá trị nằm xa đó ( có thể hủy trường hợp này hoặc xem giá trị nằm xa này là giá trị bị mất ). Nếu giá trị này khó nhận ra, mạng neural có chức năng huấn luyện chịu được giá trị nằm khỏi vùng này nhưng huấn luyện này thường kém hiệu quả hơn là huấn luyện chuẩn.

Tóm lại cách thu thập dữ liệu có thể nói gọn lại như sau:

1. Chọn những giá trị huấn luyện có tác dụng.
2. Dữ liệu số và biến có giá trị định danh có thể xử lý trực tiếp bằng mạng neural. Chuyển những loại biến khác sang một trong các dạng này.

Cần hàng trăm hoặc hàng ngàn trường hợp mẫu huấn luyện; càng nhiều biến thì càng nhiều mẫu huấn luyện. Mạng neural có khả năng nhận ra những biến hữu dụng để huấn luyện.

### **3.4. Biểu diễn tri thức cho mạng Neural**

Chúng ta có thể đưa ra định nghĩa về tri thức như sau:

Tri thức chính là thông tin được lưu trữ hay các mô hình được con người và máy móc sử dụng để biểu diễn thế giới thực, phán đoán về thế giới và có những đáp ứng phù hợp với thế giới bên ngoài.

Các đặc tính cơ bản của diễn tri thức là:

- ✓ Thông tin gì thực sự được biểu diễn.
- ✓ Làm thế nào thông tin được mã hóa một cách vật lý cho việc sử dụng sau này. Trong các ứng dụng thực tế của các máy tính thông minh, có thể nói rằng một giải pháp tốt phụ thuộc vào một biến tri thức tốt. Điều đó cũng



đúng với các mạng neural, một lớp đặc biệt của các máy thông minh. Tuy nhiên, các dạng biểu diễn có thể từ các đầu vào thành các tham số bên trong của mạng là rất đa dạng, và có khuynh hướng là cho việc tìm ra một giải pháp thích hợp nhằm biểu diễn tri thức bằng phương tiện mạng neural trở nên một sự thách thức về thiết kế.

Ở đây cần nhấn mạnh rằng mạng neural lưu trữ thông tin về thế giới thực bằng chính bản thân cấu trúc của nó kể cả về mặt hình dạng cũng như giá trị tham số bên trong(có thể thay đổi được để nắm bắt môi ). Một nhiệm vụ chính của mạng neural là học một mô hình của thế giới thực để đạt được một số mục đích xác định cần quan tâm. Tri thức của thế giới bao gồm hai loại thông tin sau:

- ✓ Trạng thái thế giới đã biết, được biểu diễn bởi các sự kiện về những cái đã biết; dạng tri thức này được xem như là các thông tin ban đầu.
- ✓ Các quan sát(đo đạc) về thế giới, thu nhận được thông qua các bộ cảm biến được thiết kế để thăm dò môi trường mà trong đó mạng hoạt động. Nói chung, các quan sát này luôn bị nhiễu và sai lệch do nhiều nguyên nhân khác nhau. Các quan sát thu nhận được như vậy cung cấp một quỹ thông tin, mà từ đó lấy ra các ví dụ được dùng để huấn luyện mạng neural.

Do cấu trúc một mạng neural là vô cùng đa dạng, nên để có thể biểu diễn tri thức một cách có hiệu quả, người ta đưa ra bốn quy tắc chung sau:

Quy tắc 1. Các đầu vào tương tự từ các lớp tương tự cần phải luôn tạo ra những biểu diễn tương tự trong mạng, và như vậy nên được phân lớp thuộc về cùng một loại. Trong tiêu chuẩn này, người ta thường sử dụng một số thước đo để xác định độ “tương tự” giữa các đầu vào (ví dụ khoảng cách euclide).

Quy tắc 2. Các phần tử mà có thể phân ra thành các lớp riêng biệt thì nên có những biểu diễn khác nhau đáng kể trong mạng.

Quy tắc 3. Nếu một đặc trưng nào đó đặc biệt quan trọng thì nên có một số lượng lớn neural liên quan đến việc biểu diễn đặc trưng này trong mạng. Số lượng lớn các neural bảo đảm mức độ chính xác cao trong việc thực hiện các quyết định và nâng cao khả năng chịu đựng các neuron hỏng.

Quy tắc 4. Thông tin ban đầu và các tính chất bất biến nên được đưa vào trong thiết kế ban đầu của một mạng neural, và như vậy sẽ giảm bớt gánh nặng cho quá trình học. Quy tắc 4 đặc biệt quan trọng vì nếu chúng ta áp dụng nó một cách thích hợp sẽ dẫn đến khả năng tạo ra các mạng neural với một kiến trúc chuyên biệt. Điều này thực sự được quan tâm do một số nguyên nhân sau:

1. Các mạng neural thị giác và thính giác sinh học được biết là rất chuyên biệt.
2. Một mạng neural với cấu trúc chuyên biệt thường có một số lượng nhỏ các tham số tự do phù hợp cho việc chỉnh lý hơn là một mạng kết nối đầy đủ. Như vậy mạng neural chuyên biệt cần một tập hợp dữ liệu nhỏ hơn cho việc tích lũy; nó học sẽ nhanh hơn, và thường có khả năng tổng quát hoá tốt hơn.
3. Tốc độ chuyển thông tin qua một mạng chuyên biệt là nhanh hơn.
4. Giá của việc xây dựng một mạng chuyên biệt sẽ nhỏ hơn do kích thước nhỏ của nó so với mạng kết nối đầy đủ.

## Chương IV THIẾT KẾ VÀ HUẤN LUYỆN MẠNG NEURAL NHÂN TẠO

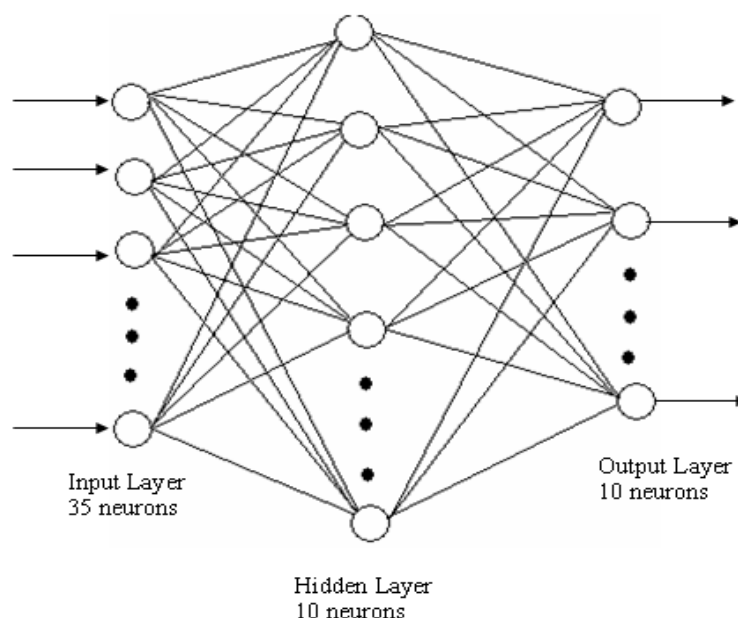
### 4.1 Thiết kế mạng Neural nhân tạo

Trong đề tài này, tôi đã tạo một mạng neuron có 3 tầng. Số phần tử neuron trên mỗi tầng phụ thuộc vào người dùng lựa chọn. Thông thường, có một tầng input, một tầng output và một tầng hidden.

Trên tầng input có 35 phần tử neuron. Có nghĩa là mỗi một ảnh được nhận dạng bởi 35 điểm ảnh đặc trưng. Kích thước của ảnh phụ thuộc vào chiều cao và độ rộng của kí tự. Cuối cùng, tất cả các ảnh cần nhận dạng được trích rút đặc trưng và đưa về kích thước chuẩn: 7x5.

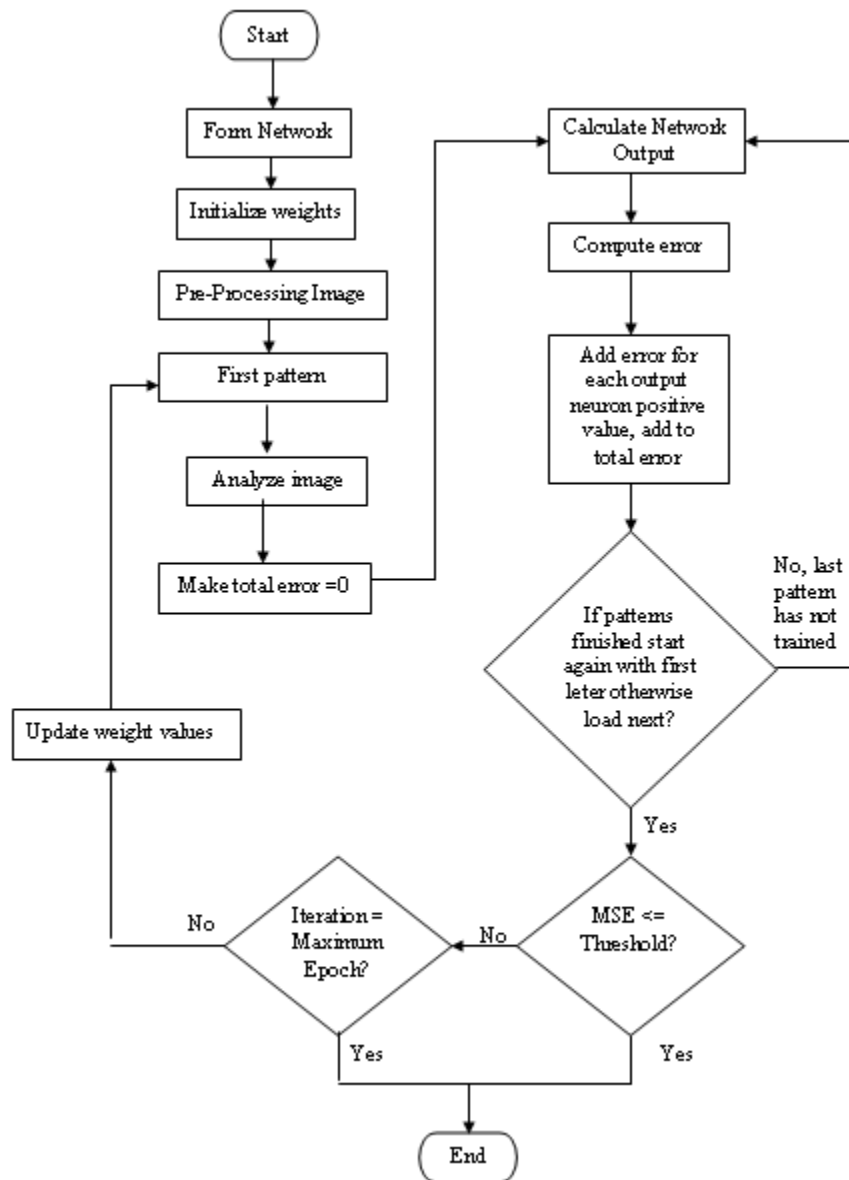
Số neuron trên tầng hidden tùy thuộc vào lựa chọn của người dùng. Nó có thể lấy giá trị từ 10 đến 1000 hoặc nhiều hơn. Nhưng để lấy được kết quả tốt nhất chúng ta phải thực hiện phương pháp thử và sửa sai.

Trên tầng output có 10 phần tử neuron. Mỗi một neuron tương ứng với các số từ 0 đến 9.



Hình 4.1. Mạng Neural nhân tạo

## 4.2. Lược đồ huấn luyện mạng



Hình 4.2. Lược đồ huấn luyện mạng Neural nhân tạo

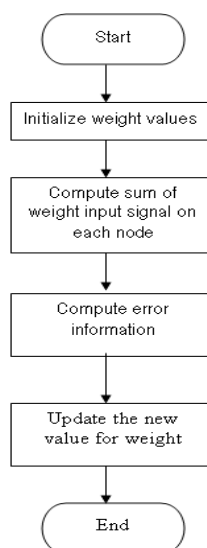
1. Tạo mạng neuron tương ứng với các thông số đầu vào.
2. Giá trị các trọng số được khởi tạo ngẫu nhiên.
3. Sau khi xây dựng được mạng, chúng ta phải xử lý ảnh đầu vào. Ở đề tài này bao gồm: Loại bỏ nhiễu và chuyển đổi sang ảnh nhị phân.
4. Lấy mẫu đầu tiên.
5. Phân tích mẫu đầu vào và phân tách mẫu này thành các mảng tuyến tính.

6. Thiết lập tổng số lỗi =0
7. Huấn luyện mẫu đó.
8. Tính toán lỗi sau khi huấn luyện.
9. Cộng giá trị lỗi cho mỗi neuron rồi cộng vào tổng lỗi.
10. Nếu mẫu kế tiếp sẵn sàng thì load tiếp và lặp lại bước 6.
11. Nếu không còn mẫu nào nữa thì tính toán MSE (Mean Square Error)
12. Nếu  $MSE \leq \text{Threshold}$  thì dừng lại
13. Nếu  $MSE > \text{Threshold}$  thì kiểm tra số lần lặp có thể.
14. Nếu số lần lặp  $< \text{Maximum Epoch}$  thì lặp lại bước 4.
15. Nếu số lần lặp = Maximum thì dừng lại.

### 4.3. Thuật toán huấn luyện mạng Neural nhân tạo – Backpropagation

BackPropagation là một phương pháp phổ biến được dùng để huấn luyện mạng neuron làm thế nào để thực hiện mục đích. Trong đề tài này tôi sử dụng cách huấn luyện có giám sát.

#### Lược đồ thuật toán BackPropagation



Hình 4.3. Lược đồ thuật toán BackPropagation.

- Khởi tạo trọng số.
  - Giá trị của trọng số liên kết giữa các neuron được khởi tạo ngẫu nhiên.
  
- Feed Forward
  - Mỗi neuron trên mỗi tầng được tính tổng trọng số của các tín hiệu vào.
  - Sử dụng hàm activation để tính giá trị ra từ neuron.
  
- Back Propagation của lỗi.
  - Thông tin lỗi được tính
  - Ta sử dụng giá trị này để tính toán thông tin lỗi trên mỗi neuron còn lại.
  
- Cập nhật giá trị trọng số.
  - Chúng ta sử dụng thông tin lỗi trên mỗi node để tính toán và cập nhật lại giá trị mới cho trọng số liên kết giữa các neuron.

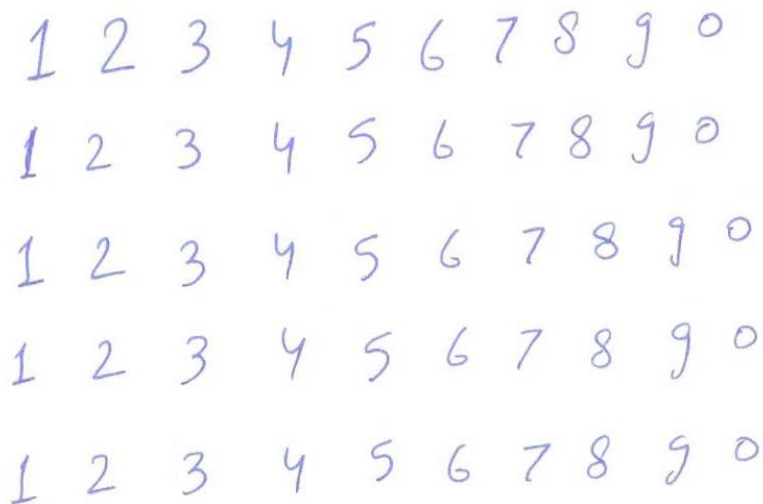
## Chương V KẾT QUẢ HUẤN LUYỆN

### 5.1 Thu thập dữ liệu cho việc huấn luyện mạng Neural

Dưới đây là một số định dạng mẫu của việc thu thập mẫu chữ viết tay:



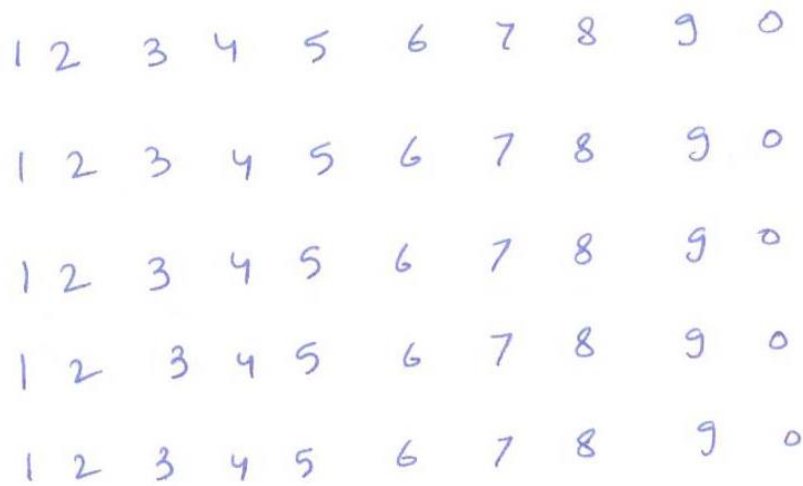
Hình 5.1. Mẫu được viết bởi Abhinav



Hình 5.2. Mẫu được viết bởi Abhishek

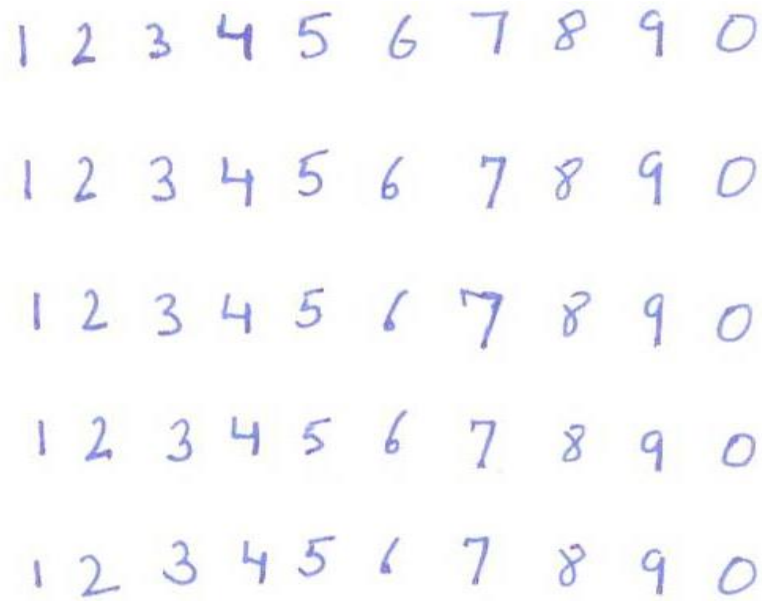


Hình 5.3. Mẫu được viết bởi Amit



Hình 5.4. Mẫu được viết bởi Amit

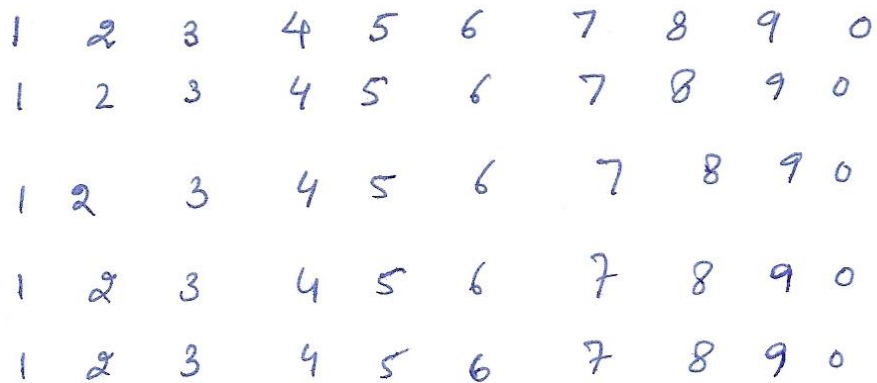




Hình 5.5. Mẫu được viết bởi Anubhav



Hình 5.6. Mẫu được viết bởi Barun



Hình 5.7. Mẫu được viết bởi Kapala\_Ma'am1

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

Hình 5.8. Mẫu được viết bởi Kapala\_Ma'am2

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

Hình 5.9. Mẫu được viết bởi KimAnh

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

Hình 5.10. Mẫu được viết bởi Linh

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

Hình 5.11. Mẫu được viết bởi Manish

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

Hình 5.12. Mẫu được viết bởi Mayank

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

Hình 5.13. Mẫu được viết bởi Nargdra

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

Hình 5.14. Mẫu được viết bởi Naveen

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

Hình 5.15. Mẫu được viết bởi Ram Manghar

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

Hình 5.16. Mẫu được viết bởi Sapneswar

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

Hình 5.17. Mẫu được viết bởi Sunil

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

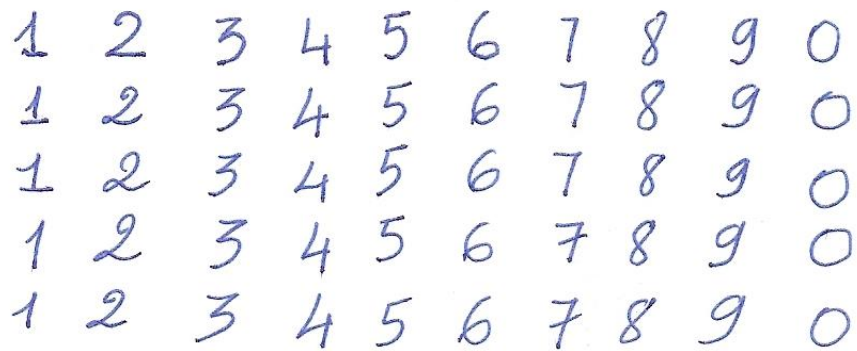
Hình 5.18. Mẫu được viết bởi Sunita\_Maam

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

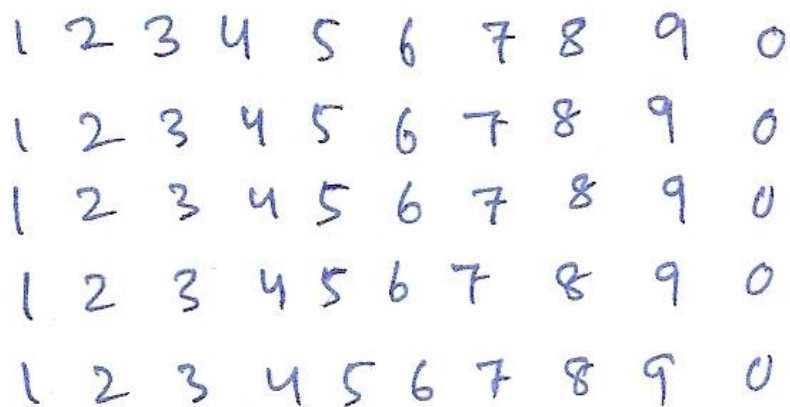
Hình 5.19. Mẫu được viết bởi Sunita\_Maam



Hình 5.20. Mẫu được viết bởi Swati



Hình 5.21. Mẫu được viết bởi Tung




Hình 5.22. Mẫu được viết bởi Tushar\_Sir1



Handwritten sample of numbers 1 through 0, repeated five times in a row. The numbers are written in a simple, slightly irregular cursive style.

Hình 5.23. Mẫu được viết bởi Tushar\_Sir2



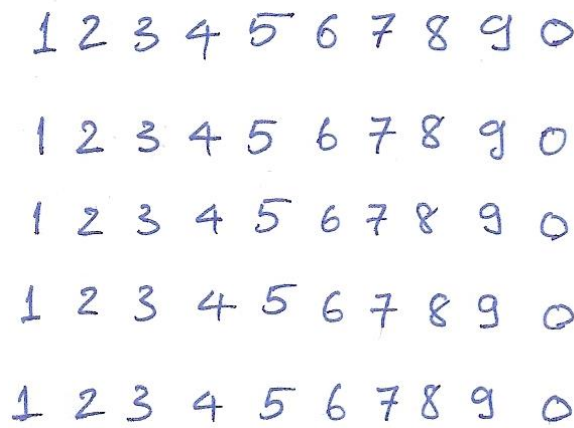
Handwritten sample of numbers 1 through 0, repeated five times in a row. The numbers are written in a simple, slightly irregular cursive style.

Hình 5.24. Mẫu được viết bởi Vineet



Handwritten sample of numbers 1 through 0, repeated five times in a row. The numbers are written in a simple, slightly irregular cursive style.

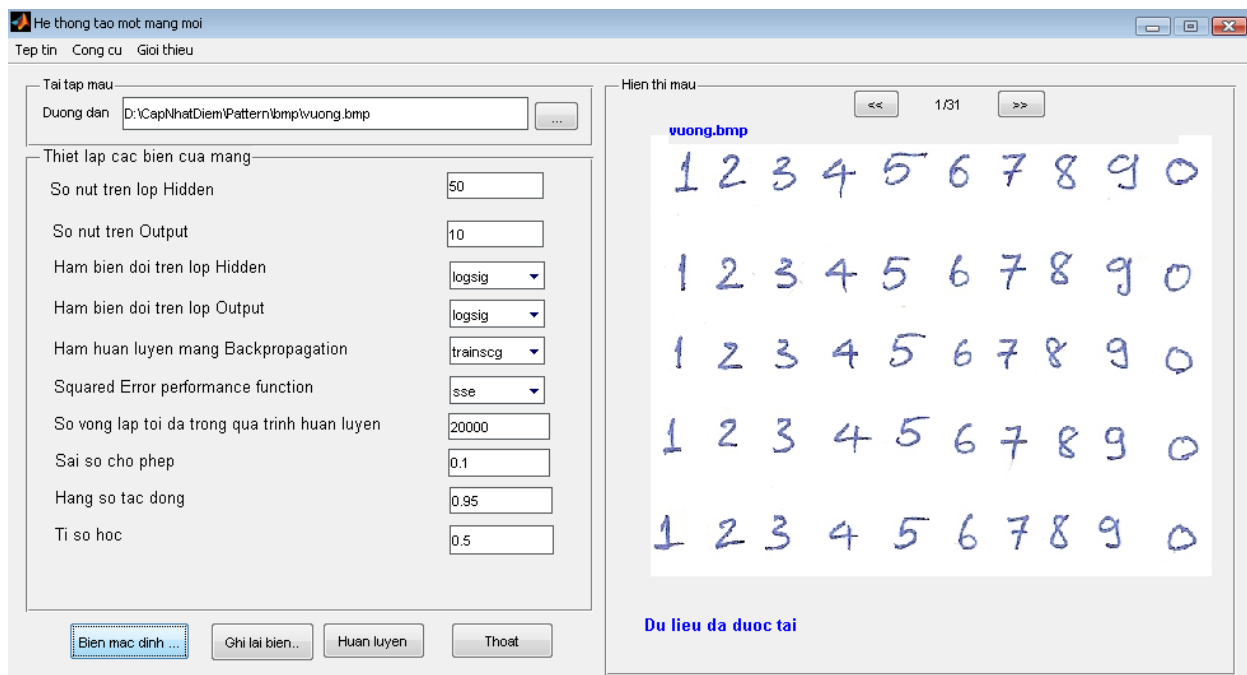
Hình 5.25. Mẫu được viết bởi Vishal



Hình 5.26. Mẫu được viết bởi Vương

## 5.2. Xây dựng mạng và huấn luyện

Sau khi đã thu thập mẫu, ta xây dựng mạng nơron và huấn luyện mạng dựa trên mẫu đã thu thập được. Việc huấn luyện mạng đã được tôi thiết kế thành một module riêng trong chương trình cho phép người dùng tạo các kiểu mạng nơron theo kiến trúc khác nhau.



Hình 5.27. Form cho phép người dùng tạo vào huấn luyện một mạng mới

Dưới đây là kết quả của việc huấn luyện mạng với một vài tham số khác nhau:



No	Số mẫu	Noron trên lớp ẩn	Hàm Activation		Hàm huấn luyện	Momentum Constant	Lỗi chấp nhận	Hệ số học	Số vòng- (times)	Hoàn thành
			Lớp ẩn	Lớp ra						
1	130	10	logsig	logsig	traingdx	0.95	0.1	0.01	4-(1s)	No
2	130	20	logsig	Logsig	Traingdx	0.95	0.1	0.01	2-(1s)	No
3	130	30	logsig	Logsig	Traingdx	0.95	0.1	0.01	2-(1s)	No
4	130	10	tansig	Logsig	Traingdx	0.95	0.1	0.01	6-(1s)	No
5	130	20	tansig	Logsig	Traingdx	0.95	0.1	0.01	9-(1s)	No
6	130	30	tansig	Logsig	Traingdx	0.95	0.1	0.01	6-(1s)	No
7	130	10	logsig	Logsig	Trainscg	0.95	0.1	0.01	683-(20s)	No
8	130	20	logsig	Logsig	Trainscg	0.95	0.1	0.75	420-(17s)	No
9	130	20	logsig	Logsig	Trainscg	0.95	0.1	0.01	278-(10s)	No
10	130	20	logsig	Logsig	Trainrp	0.95	0.1	0.01	648-(19s)	No
11	130	20	logsig	Logsig	Trainscg	0.95	0.1	0.01	301-(15s)	No
12	130	100	logsig	Logsig	Trainscg	0.95	0.1	0.01	176-(35s)	Yes
13	130	200	logsig	Logsig	Trainscg	0.95	0.1	0.01	234-(90s)	Yes
14	130	500	logsig	Logsig	Trainscg	0.95	0.1	0.01	248-(235s)	Yes
15	130	100	logsig	Logsig	Trainscg	0.95	0.001	0.01	264-(45s)	Yes
16	130	100	logsig	Logsig	Trainscg	0.75	0.1	0.71	283-(55s)	Yes
17	130	1000	logsig	Logsig	Trainscg	0.85	0.001	0.05	310-(635s)	Yes
18	130	50	tansig	tansig	Trainscg	0.45	0.1	0.05	5931-(600s)	No
19	130	90	logsig	logsig	Traingdx	0.75	0.01	0.75	1-(1s)	No
20	130	90	logsig	logsig	Trainscg	0.75	0.01	0.75	221-(41s)	Yes
21	130	150	logsig	logsig	Trainscg	0.75	0.01	0.75	220-(65s)	Yes
22	130	150	logsig	logsig	Trainscg	0.75	0.0001	0.75	316-(92s)	Yes
23	130	70	logsig	logsig	Trainscg	0.55	0.01	0.95	197-(28s)	Yes
24	130	70	logsig	logsig	Trainscg	0.95	0.01	0.95	201-(33s)	Yes
25	130	70	logsig	logsig	Trainscg	0.05	0.01	0.95	195-(27s)	Yes
26	130	70	logsig	logsig	Trainscg	0.55	0.01	0.25	184-(29s)	Yes

27	130	70	logsig	logsig	Trainscg	0.55	0.01	0.45	268-(39s)	Yes
28	130	70	logsig	logsig	Trainscg	0.55	0.01	0.75	224-(32s)	Yes
29	130	100	logsig	logsig	Trainscg	0.55	0.01	0.75	284-(58s)	Yes
30	130	100	logsig	logsig	Trainscg	0.55	0.01	0.65	284-(57s)	Yes

Dựa vào một số kết quả trong bảng trên, trong quá trình huấn luyện mạng ta nên sử dụng hàm logsig cho mỗi nút trên các tầng khác nhau và sử dụng hàm Traingscg thì việc huấn luyện mạng sẽ dễ dàng thành công hơn.

## KẾT LUẬN

Dựa vào các kết quả huấn luyện mạng nêu trên, có rút ra một số kết luận như sau:

1. Huấn luyện mạng neural có thành công hay không phụ thuộc nhiều vào các giá trị khởi tạo ban đầu. Nếu lựa chọn được giá trị tối ưu thì việc huấn luyện sẽ nhanh đạt được kết quả.
2. Chất lượng của quá trình huấn luyện phụ thuộc nhiều vào chất lượng của tập mẫu. Nếu tập mẫu càng nhiều, đầy đủ, thì chất lượng nhận về tri thức của mạng càng cao, tuy nhiên thời gian huấn luyện sẽ lâu hơn.
3. Thời gian huấn luyện phụ thuộc nhiều vào các lựa chọn tham số đầu vào tại các nút ẩn, nút ra, nút vào.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

- [1] Hoàng Kiếm, Nguyễn Hồng Sơn, Đào Minh Sơn, “Ứng dụng mạng nơron nhân tạo trong hệ thống xử lý biểu mẫu tự động”, Kỷ yếu hội nghị kỷ niệm 25 năm thành lập Viện Công nghệ Thông tin, 2001, tr. 560-567.
- [2] Bùi Minh Trí, “Quy hoạch toán học”, Nhà xuất bản Khoa học và kỹ thuật, Hà nội, 2006.
- [3] Lê Hoài Bắc, Lê Hoàng Thái, “Neural Network & Genetic Algorithm in Application to Handwritten Character Recognition”, Tạp chí Tin học và Điều khiển học, Tập 17, số 4, 2001, tr. 57-65.
- [4] Nguyễn Thị Thanh Tân, Ngô Quốc Tạo, “Một cấu trúc mạng nơ ron thích hợp cho việc nhận dạng chữ số viết tay”, Kỷ yếu hội thảo FAIR03, NXB KH&KT Hà Nội, 2004, tr. 200-210.
- [5] Nguyễn Thị Thanh Tân, Lương Chi Mai, “Phương pháp nhận dạng từ viết tay dựa trên mô hình mạng nơ ron kết hợp với thông kê từ vựng”, Tạp chí Tin học và Điều khiển học, Tập 22, số 2, 2006, tr. 141-154.
- [6]. Nguyễn Thị Thanh Tân, Lương Chi Mai, “Phương pháp nhận dạng từ viết tay dựa trên mô hình mạng nơ ron kết hợp với thông kê từ vựng”, *Tạp chí Tin học và Điều khiển học*, Tập 22, số 2, 2006, tr. 141-154.

### Tiếng Anh

- [7] J. Platt, N. Cristianini and J. Shawe-Taylor, “Large Margin DAGs for Multiclass Classification”, In Advances in Neural Information Processing Systems, volume 2, pp. 547-553, 2000.
- [8] Nguyen, D.D., Ho, T.B., A Bottom-up Method for Simplifying Support Vector Solutions, IEEE Transactions on Neural Networks, Vol.17, No. 3, 2006, pp. 792-796.
- [9] Sergios Theodoridis and Konstantinos Koutroumbas, “Pattern Recognition”, Academic Press, 2006.

- [10] Robert A. Dunne, “A Statistical Approach to Neural Networks for Pattern”, N. Y.: John Wiley & Sons, 2007.
- [11] S. W. Lee, Y. J. Kim, “Multiresolutional Recognition of Handwritten Numerals with Wavelet Transform and Multilayer Cluster Neural Network”, in Proc. 3rd Int. Conf. Document Analysis and Recognition, pp.1010-1014, Montreal, Canada, 1995.