LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy, các cô khoa Công nghệ Thông Tin Trường Đại học Dân lập Hải Phòng đã tận tình dạy dỗ, truyền đạt cho chúng em nhiều kiến thức, kinh nghiệm quý báu trong suốt quá trình học trong trường.

Đặc biệt,em xin tỏ lòng biết ơn sâu sắc đến thầy giáo-Tiến sỹ Hồ Văn Canh đã trực tiếp dìu dắt, giúp đỡ em tận tình, chu đáo trong suốt thời gian em hoàn thiện đồ án tốt nghiệp.

Xin chân thành cảm ơn các bạn trong khoa Công Nghệ Thông Tin, trường Đại Học Dân Lập Hải Phòng đã giúp đỡ, động viên tôi rất nhiều trong quá trình thực hiện đề tài.

Em xin chân thành cảm ơn!

Hải Phòng, tháng 06 năm 2010

MỤC LỤC

LỜI MỞ ĐẦU 3
CHƯƠNG I: GIỚI THIỆU CHUNG VỀ MẠNG NEURAL 4
1.1 Tổng quan về mạng neural sinh học 4
1.1.1 Cấu trúc mạng neural sinh học 4
1.1.2 Khả năng của mạng neural sinh học (bộ não) 5
1.1.3 Quá trình học của bộ não 5
1.2 Neural nhân tạo 6
1.2.1 Định nghĩa 6
1.2.2 Mô hình neural 6
1.2.2.1 Neural một đầu vào 7
1.2.2.2 Neural nhiều đầu vào 9
1.3 Mạng neural nhân tạo 10
1.3.1 Định nghĩa 10
1.3.2 Một số chức năng của mạng neural nhân tạo 11
1.3.2.1 Chức năng phân loại mẫu 11
1.3.2.2 Học và tổng quát hóa 11
1.3.3 Lịch sử phát triển của mạng neural nhân tạo 11
1.4 Kiến trúc mạng neural 13
1.4.1 Lớp của các neural 13
1.4.2 Mạng neural nhiều lớp (Multiple Layers of Neurons) 14
1.5 Phân loại mạng neural 16
1.6 Hoạt động của mạng neural nhân tạo 17
1.6.1 Hoạt động của mạng neural 17
1.6.2 Luật học của mạng neural 17
CHƯƠNG II: MẠNG PERCEPTRON ĐA LỚP VỚI LUẬT HỌC LAN TRUYỀN NGƯỢC SAI SỐ 20
2.1 Mạng neural nhiều lớp lan truyền ngược sai số 20
2.1.1 Tổng quan về mạng neural truyền thẳng nhiều lớp 20
2.1.2 Kiến trúc mạng 21
2.1.3 Cơ chế huấn luyện của mạng neural lan truyền ngược sai số 21
2.2 Các nhân tố của quá trình học lan truyền ngược sai số 28
2.2.1 Khởi tạo các trọng số 28
2.2.2 Hằng số học α (Anpha) 29
2.2.3 Tập mẫu học và dự báo 30
2.3 Cấu trúc mạng 31
2.4 Sự hội tụ của thuật toán huấn luyện mạng 32
CHƯƠNG III: KỸ THUẬT NHẬN DẠNG BẢN RÕ TIẾNG ANH 33
3.1 Bài toán 33
3.2 Thuật toán 33
3.2.1 Phần off-line 33
3.2.2 Phần on-line 39
3.2.3 Một số ví dụ 41
CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM 45
4.1 Kết quả đạt được 45
4.2 Mã nguồn của chương trình 46
4.2.1 Thủ tục tính tần số bộ đôi với độ dài k 46
4.2.2 Hàm tính tổng của 2 ma trận 47
4.2.3 Hàm nhận biết ngôn ngữ 47
KẾT LUẬN 48
TÀI LIỆU THAM KHẢO 49

LỜI MỞ ĐẦU

Kỹ thuật nhận dạng đang là một vấn đề rất được quan tâm hiện nay, đặc biệt trong an ninh quốc phòng: như nhận dạng chữ ký, nhận dạng mẫu tóc, nhận dạng hình ảnh, nhận dạng vân lòng bàn tay, nhận dạng chữ viết, nhận dạng ngôn ngữ, nhận dạng sinh trắc học,v.v…

Ngày nay, do sự phát triển nhanh chóng của khoa học công nghệ, đặc biệt là CNTT, ngoài hai kỹ thuật nhận dạng truyền thống là nhận dạng dựa vào các tham số của đối tượng và nhận dạng theo cấu trúc, một hướng mới đang được quan tâm nghiên cứu là nhận dạng dựa vào kỹ thuật mạng neural. Kỹ thuật này bước đầu đang được ứng dụng và đã cho những kết quả quan trọng. Điều này nói lên tính cấp thiết của khoa học về mạng neural trong việc giải quyết nhiều bài toán trong thực tiễn. Khả năng ứng dụng của mạng neural hiện nay không còn nằm trong các phòng thí nghiệm nữa mà đã xuất hiện ứng dụng vào trong các lĩnh vực thương mại.

Xuất phát từ lý do đó nên em mạnh dạn chọn đề tài: Tìm hiểu mạng neural và ứng dụng của nó, làm đồ án tốt nghiệp của mình.

Do đây là một đề tài khó và mới đối với em nên trong quá trình nghiên cứu chắc chắn em sẽ gặp nhiều khó khăn. Do vậy em rất mong được các thầy, cô thông cảm và cho em những chỉ bảo, em xin chân thành cảm ơn!

Hải Phòng, tháng 06 năm 2010

Sinh viên

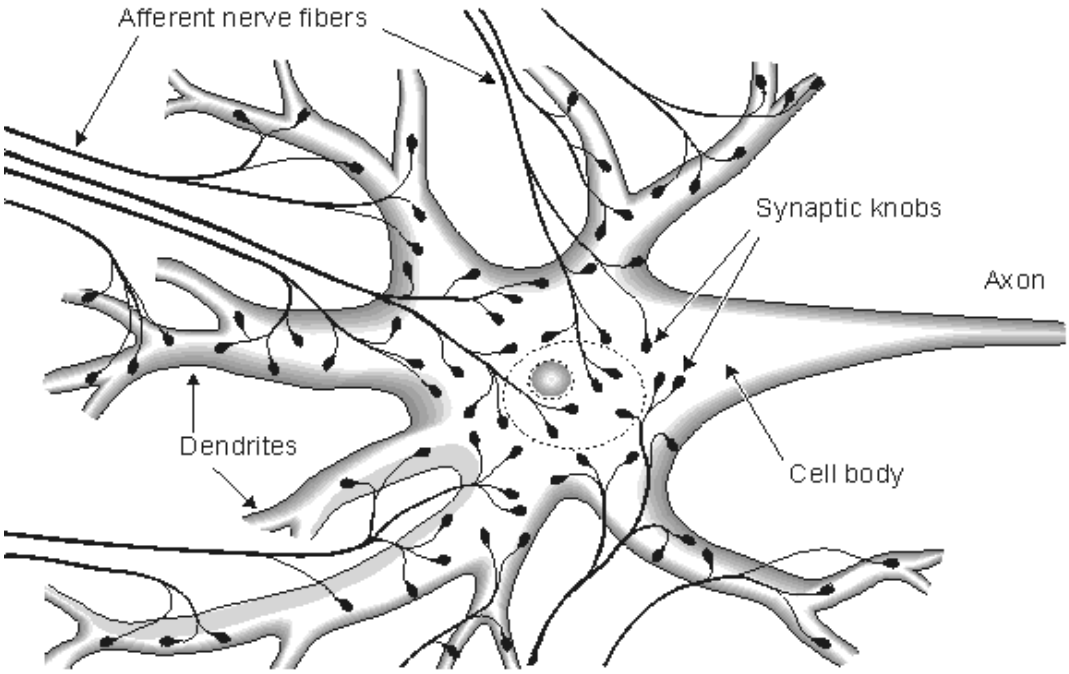
Bùi Duy Quảng

CHƯƠNG I: GIỚI THIỆU CHUNG VỀ MẠNG NEURAL

1.1 Tổng quan về mạng neural sinh học

1.1.1 Cấu trúc mạng neural sinh học

Bộ não người có mạng lưới gồm khoảng 10^{11} tế bào thần kinh (gọi là nơ-ron) liên kết phức tạp với nhau. Mỗi tế bào thần kinh gồm 3 thành phần chính: thân tế bào thần kinh (*cell body* còn gọi là *soma*), hệ thống các dây thần kinh tiếp nhận (*dendrites*) và một sợi trục thần kinh (*axon*).



Hình 1.1 *Mô hình tế bào thần kinh*

Hệ thống dây thần kinh tiếp nhận là một lưới dày đặc các dây thần kinh dạng cây bao bọc xung quanh thân tế bào, chúng dẫn các tín hiệu đến phần thân tế bào. Thân tế bào sẽ tổng hợp các tín hiệu đầu vào này, làm thay đổi điện thế của nó và khi vượt qua một mức ngưỡng thì sẽ cho ra một xung điện trên sợi trục thần kinh ra (*Axon*). Các dây thần kinh *axon* có thể rẽ ra nhiều nhánh để nối đến các dây thần kinh vào hoặc nối trực tiếp với phần thân của các tế bào thần kinh khác thông qua các khớp thần kinh (*synapse*).

Khi một tế bào thần kinh hoạt động, nó được kích thích tạo ra một tín hiệu điện hóa chạy dọc theo sợi *axon* và dẫn đến các khớp thần kinh. Khớp thần kinh được chia làm 2 loại: khớp nối kích thích (*excitalory*) và khớp nối ức chế (*inhibitory*). Tại các khớp thần kinh này xảy ra các quá trình phản ứng và giải phóng các chất hữu cơ tạo nên các tín hiệu điện kích thích tế bào thần kinh. Cường độ tín hiệu mà một tế bào thần kinh nhận được phụ thuộc chủ yếu vào mức độ liên kết của các khớp nối. Những nghiên cứu hoạt động của hệ thần kinh đã chỉ ra rằng quá trình "học" của bộ não chính là việc hình thành hoặc thay đổi mức độ liên kết của các khớp nối.

1.1.2 Khả năng của mạng neural sinh học (bộ não)

- Bộ nhớ được tổ chức theo các bó thông tin và truy nhập theo nội dung (có thể truy xuất thông tin dựa theo các giá trị thuộc tính của đối tượng).

- Bộ não có khả năng tổng quát hóa, có thể truy xuất các tri thức hay các mối liên kết chung của các đối tượng tương ứng với một khái niệm chung nào đó.

- Bộ não có khả năng học.

1.1.3 Quá trình học của bộ não

Khi các xung tín hiệu từ các "dây thần kinh vào" tới các khớp nối, khớp nối sẽ cho tín hiệu đi qua hoặc không kích thích neural tiếp theo. Do vậy hình thành một con đường truyền xung nhất định.

Học là làm sao cho con đường này được lặp lại nhiều lần, nên sức cản của các khớp nối sẽ nhỏ dần, tạo điều kiện cho những lần lặp lại dễ dàng hơn. Có thể nói: Toàn bộ những kiến thức, kinh nghiệm của một người tích lũy được và lưu giữ trong đầu chính là hệ thống sức cản của các khớp nối.

1.2 Neural nhân tạo

1.2.1 Định nghĩa

Neural nhân tạo (*Atificial Neural Networks*) là sự mô phỏng đơn giản của neural sinh học. Mỗi neural nhân tạo thực hiện hai chức năng: chức năng tổng hợp đầu vào và chức năng tạo đầu ra.

Mỗi neural nhân tạo có một số đầu vào và một đầu ra. Mỗi đầu vào được gắn một hệ số nhân gọi là trọng số (*weight*) có ý nghĩa như mức độ liên kết tại khớp nối trong mạng neural sinh học. Trọng số có thể là dương hoặc âm, giống như trong mạng neural sinh học có hai loại khớp nối: khớp nối kích thích và khớp nối ức chế.

Mỗi neural có một giá trị ngưỡng. Chức năng đầu vào chính là tổng có trọng số các tín hiệu vào kết hợp với ngưỡng để tạo ra tín hiệu đầu vào *net input*. Sự kết hợp này được thực hiện bằng một tổng hay theo một số tài liệu gọi là hàm PSP (*Post Synapic Potential function*) - hàm thế sau khớp nối.

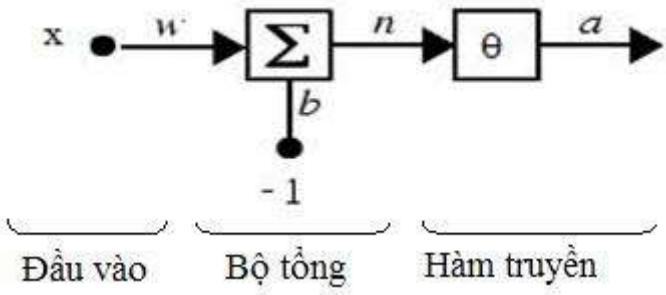
Chức năng tạo đầu ra được thể hiện bằng hàm truyền đạt (*transfer function*). Hàm này sẽ nhận tín hiệu đầu vào *net input* và tạo tín hiệu đầu ra của neural.

1.2.2 Mô hình neural

Mạng neural nhân tạo gồm hai thành phần: Các nút (đơn vị xử lý, neural) và các liên kết giữa chúng được gán một trọng số nào đó đặc trưng cho cường độ liên kết.

Ta ký hiêu: Pi là tín hiệu đầu vào; Xi là tín hiệu đầu ra của neural i. Trạng thái đầu vào của neural i được xác định bởi tổng tuyến tính của các tín hiệu vào có trọng số từ các neural j khác.

1.2.2.1 Neural một đầu vào



Hình 1.2 Mô hình neural một đầu vào

Một neural đơn giản với một đầu vào được diễn tả bởi hình vẽ trên. Đầu vào vô hướng *p* được nhân với trọng số vô hướng *w* thành *wp* là một trong hai số hạng được đưa vào bộ tổng. Một đầu vào khác là 1 được nhân với hệ số bias *b* sau đó được đưa vào bộ tổng. Bộ tổng cho ra *n*, thường được gọi là tín hiệu đầu vào *net input*, *n* được cho qua hàm truyền đạt *f* kết quả được đầu ra *a* của neural. Một số tài liệu gọi hàm *f* là hàm hoạt hóa (*activation function*).

Nếu chúng ta liên hệ mô hình đơn giản này với một neural sinh học thì trọng số *w* tương ứng với độ liên kết (độ mạnh) của khớp nối (*synapse*), đầu vào *p* tương ứng với dây thần kinh tiếp nhận (*dendrite*), còn thân neural (*cell body*) được mô hình bởi bộ tổng và hàm truyền đạt, đầu ra của neural *a* diễn tả tín hiệu ra trên sợi trục neural sinh học (*axon*).

Đầu ra của neural được tính bởi:

$$a=f(wp+b) \qquad (2.1)$$

Ví dụ: với $w=3$, $p=2$ và $b=-1,5$ thì $a=f(3.(2)-1,5)=f(4,5)$

Đầu ra a phụ thuộc vào hàm truyền f được chọn là hàm nào trong từng trường hợp cụ thể.

Hệ số chệch (bias) cũng giống như một trọng số với đầu vào luôn là 1. Neural có thể có hoặc không có hệ số bias (chệch).

Ta thấy rằng w và b là các tham số vô hướng có thể điều chỉnh được của neural. Thông thường dạng hàm truyền được chọn bởi người thiết kế và sau đó các tham số w và b sẽ được điều chỉnh bởi một số luật học để mối quan hệ vào/ra của neural thỏa mã mục đích cụ thể của người thiết kế.

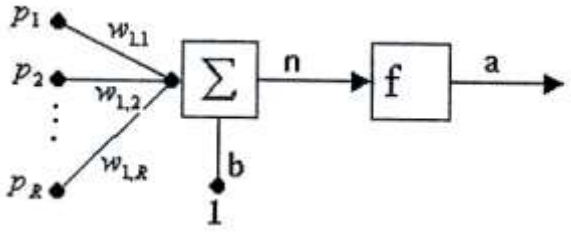
Hàm truyền f có thể là hàm truyền tuyến tính hoặc phi tuyến đối với n. Có rất nhiều dạng hàm truyền được sử dụng.

BẢNG 1.1: CÁC DẠNG HÀM TRUYỀN

Tên hàm	**Công thức**
hardlim	a = 0 với n < 0 a = 1 với n ≥ 0
hardlims	a = -1 với n < 0 a = 1 với n ≥ 0
purelin	a = n
satlin	a = 0 với n < 0 a = n với 0 ≤ n ≤ 1 a = 1 với n > 1
satlins	a = -1 với n < 0 a = n với 0 ≤ n ≤ 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 ≥ 0
compet	a = 1 với neural có n lớn nhất a = 0 với các neural còn lại
logsig	$a = \frac{1}{1 + e^{-n}}$

1.2.2.2 Neural nhiều đầu vào

Thông thường neural có nhiều đầu vào. Một neural với R đầu vào được diễn tả:



Hình 1.3 Mô hình neural nhiều đầu vào

Mỗi đầu vào riêng biệt p_1, p_2, ... p_R đều tương ứng với một trọng số $w_{1,1}$,$w_{1,2}$,... $w_{1,R}$ trong ma trận trọng số W. Ta có:

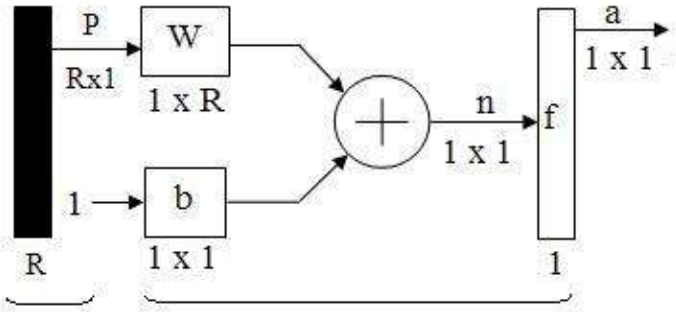
$$n= w_{1,1} .p1 + w_{1,2}. p_2 + ... + w_{1,R} .p_R + b$$

Hay viết dưới dạng ma trận n=Wp + b, trong trường hợp này mà trận W chỉ gồm một hàng.

Véc tơ tín hiều được biểu diễn dưới dạng ma trận như sau:

Đầu ra của neural được tính a= f (Wp + b). Đối với mỗi phần tử của ma trận W, ta quy ước $w_{i,j}$ để chỉ trọng số nối đầu vào thứ j với neural thứ i (trong trường hợp này chỉ có một neural nên i=1).

Mô hình neural nhiều đầu vào trên có thể được ký hiệu vắn tắt như sau:



Hình 1.4 Mô hình vắn tắt neural nhiều đầu vào

Đầu vào a= f (Wp+b)

Nhìn vào mô hình trên ta có thể biết vec-tơ đầu vào p có R phần tử. Ma trận trọng số W có 1 hàng và R cột, hằng số đầu vào 1 được nhân với hệ số bias b. Bộ tổng kết hợp với hệ số bias b và tịch hợp Wp tạo ra tín hiệu đầu vào là số

vô hướng, hàm truyền f biến đổi n thành đầu ra của neural a, trong trường hợp này a là số vô hướng còn trong mạng neural thì a là vec-tơ đầu ra. Từ đây trở đi ta sẽ dùng mô hình vắn tắt như trên để biểu diễn các mạng neural.

1.3 Mạng neural nhân tạo

1.3.1 Định nghĩa

Mạng neural nhân tạo là sự kết hợp giữa các neural nhân tạo với nhau. Mỗi liên kết kèm theo một trọng số nào đó đặc trưng cho đặc tính kích hoạt ức chế giữa các neural. Các neural còn gọi là các nút (node) được sắp xếp trong mạng theo các lớp, bao gồm lớp ra (output player) và các lớp ẩn (hiden layer).
Các đặc điểm của mang neural nhân tạo:

- Mạng được xây dựng bằng các neural liên kết lại với nhau.

- Chức năng của mạng được xác định bởi: cấu trúc mạng, quá trình xử lý bên trong của từng neural, và mức độ liên kết giữa các neural.

- Mức độ liên kết giữa các neural được xác định thông qua quá trình học của mạng (quá trình huấn luyện mạng). Có thể xem các trọng số là các phương tiện để lưu trữ thông tin dài hạn trong mạng neural. Nhiệm vụ của quá trình huấn luyện mạng là cập nhật các trọng số khi có thông tin về các mẫu học.

*** Một số định nghĩa về mạng neural:**

+Mạng neural là một hệ thống gồm nhiều phần tử xử lý hoạt động song song. Chức năng của nó được xác định bởi cấu trúc mạng, độ lớn các liên kết và quá trình xử lý tại mỗi nút hoặc đơn vị tính toán.

+Một mạng neural là một bộ xử lý song song và đồ sộ, có xu hướng tự nhiên là lưu trữ các tri thức dựa trên kinh nghiệm, và tạo ra tri thức mới dựa vào cái đã có.

Nó tương tự với bộ não ở hai khía cạnh:

- Tri thức có được thông qua quá trình học.

- Độ lớn liên kết giữa các neural được dùng như một phương tiện lưu trữ thông tin.

+Hệ thống neural nhân tạo, hay còn gọi là các mạng neural, là một tập hợp các tế bào vật lý, được liên kết với nhau nhằm mục đích thu thập, lưu trữ và sử dụng tri thức, kinh nghiệm một cách tốt nhất.

1.3.2 Một số chức năng của mạng neural nhân tạo

1.3.2.1 Chức năng phân loại mẫu

Phân loại mẫu là sự sắp xếp các mẫu thành các nhóm khác nhau. Mạng neural có thể tạo ra một mẫu ra khi đưa cho nó một mẫu vào, đây là chức năng phân loại mẫu của mạng neural. Mạng neural nhận mẫu vào và tạo một mẫu ở đầu ra đúng với phân loại. Có thể nói mạng neural là một bộ phân loại mẫu. Điểm khác của mạng neural với các bộ phân loại mẫu khác là khả năng học và tổng quát hóa của mạng neural.

1.3.2.2 Học và tổng quát hóa

Đầu tiên là việc học, có thể hiểu việc này là cho mạng neural xem một ít mẫu kèm với đầu ra tương ứng với mẫu đó và mạng neural phải học để phân loại đúng được các mẫu này. Còn khả năng tổng quát hóa là: mạng neural không chỉ nhận biết được các mẫu nó đã được học mà có thể nhận được các mẫu gần với mẫu nó đã được học. Tức là mạng neural có thể suy ra các đặc tính chung của các lớp khác nhau từ các mẫu đã cho. Chức năng này tạo ra một chiến lược tính toán rất phù hợp cho việc giải quyết các vấn đề mang tính "động", tức là thông tin về chúng có rất ít hoặc bị thiếu, không đầy đủ. Điều quan trọng là tìm được mô hình mạng và phương pháp học thích hợp đối với từng bài toán.

Ngoài ra mạng neural còn có khả năng được huấn luyện để trở thành bộ xấp xỉ hàm liên tục bất kỳ.

1.3.3 Lịch sử phát triển của mạng neural nhân tạo

- Cuối thế kỷ 19, đầu thế kỷ 20, một số nghiên cứu về vật lý, tâm lý và hệ thần kinh của các nhà khoa học Herman, Ernst Mach và Ivan Ivalov đã đưa ra các lý thuyết về quá trình học, sự tưởng tượng, sự quyết định... của hệ thần kinh nhưng chưa có sự mô tả toàn học cho hoạt động của mạng neural.

- Năm 1943, mô hình đơn giản mạng neural bằng mạch điện tử lần đầu tiên được đưa ra bởi Warren McCulloch và Walter Pits cùng với sự khẳng định mạng neural nhân tạo về nguyên lý có thể thực hiện được trong phạm vi tính toán các hàm số học và logic. Đây là điểm khởi đầu của lĩnh vực mạng neural.

- Sau đó Donal Hebb đưa ra một cơ chế giải thích cho quá trình học (learning) diễn ra trong các neural sinh học (trong cuốn Organnization of Behaviaor - 1949).

- Cuối thập niên 50, ứng dụng thực tế đầu tiên của mạng neural nhân tạo do Frank Rosenblatt đưa ra. Mạng của ông đưa ra là mạng Perceptron có kết hợp luật học (learning rule) dùng để nhận dạng mẫu (pattern recognition). Cùng thời gian đó, Bernard Widrow và Ted Hoff giới thiệu một thuật toán học (learning algorithm) và sử dụng nó để huấn luyện (training) các mạng neural tiếp hợp tuyến tính (tương tự mạng của Rosenblatt).

- Năm 1969, Minskey và Papert là hai nhà toán học nổi tiếng thời đó đã chỉ ra những hạn chế của mạng Perceptron của Rosenblatt và mạng Widrow-Hoff làm nhiều người nghĩ rằng nghiên cứu về mạng neural sẽ vào ngõ cụt. Hơn nữa vào thời gian này chưa có những máy tính số mạnh để thực nghiệm mạng neural nên các nghiên cứu về mạng nơ-ron bị trì hoãn gần một thập kỷ.

- Năm 1972, Teuvo Kohonen và James Anderson độc lập phát triển các mạng neural mới với năng lực nhớ (memory) và khả năng tự tổ chức (self-organizing). Cũng trong giai đoạn này,Stephen Grossberg cũng nghiên cứu tích cực về các mạng tự tổ chức.

- Sang thập kỷ 80, khi ngành công nghiệp máy tính phát triển mạnh mẽ thì những nghiên cứu về mạng neural tăng lên một cách đột ngột. Có hai phát kiến quan trọng nhất là:

+ Sử dụng cơ học thống kê để giải thích hoạt động của mạng hồi qui một lớp (recurrent network), loại mạng được sử dụng như một bộ nhớ kết hợp, được nhà vật lý John Hopfield mô tả.

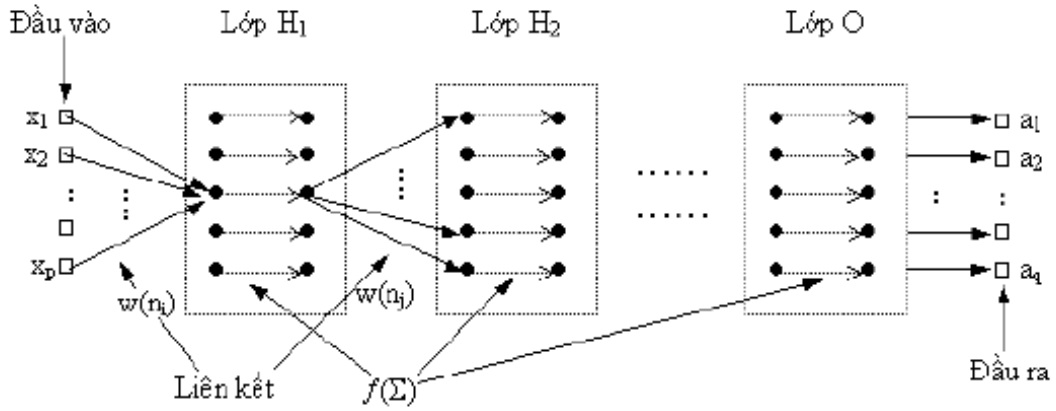
+ Sử dụng thuật toán lan truyền ngược (back-propagation algorithm) để huấn luyện các mạng perceptron đa lớp (mutilayer perceptron network). David

Rumelhalt và James McClrlland là những người trình bày thuật toán lan truyền ngược có ảnh hưởng nhất (1968).

- Ngày nay, lĩnh vực mạng neural được nghiên cứu, phát triển mạnh mẽ và ứng dụng rất nhiều vào trong thực tế.

1.4 Kiến trúc mạng neural

Một neural với rất nhiều đầu vào cũng không đủ để giải quyết các bài toán. Ta cần nhiều neural được tổ chức song song tạo thành "lớp" (layer).

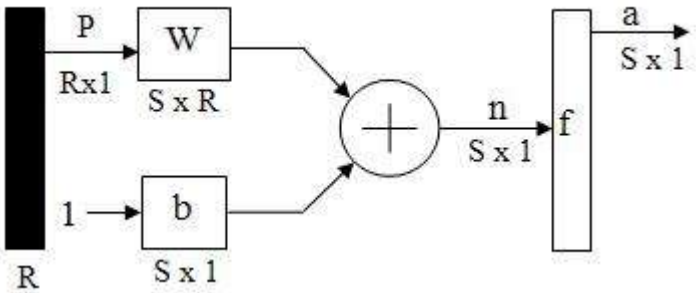


Hình 1.5 Cấu trúc chung của mạng neural

1.4.1 Lớp của các *neural*

Một mạng của lớp S của neural với R đầu vào được biểu diễn bởi hình sau:

a= f (Wp + b)



Hình 1.6 Mô hình mạng neural có 1 lớp S neural

Mỗi một thành phần của R đầu vào được nối với mỗi một neural trong lớp gồm S neural. Trong trường hợp này ma trận trọng số W gồm S hàng và R cột, véctơ đầu ra a gồm S phần tử:

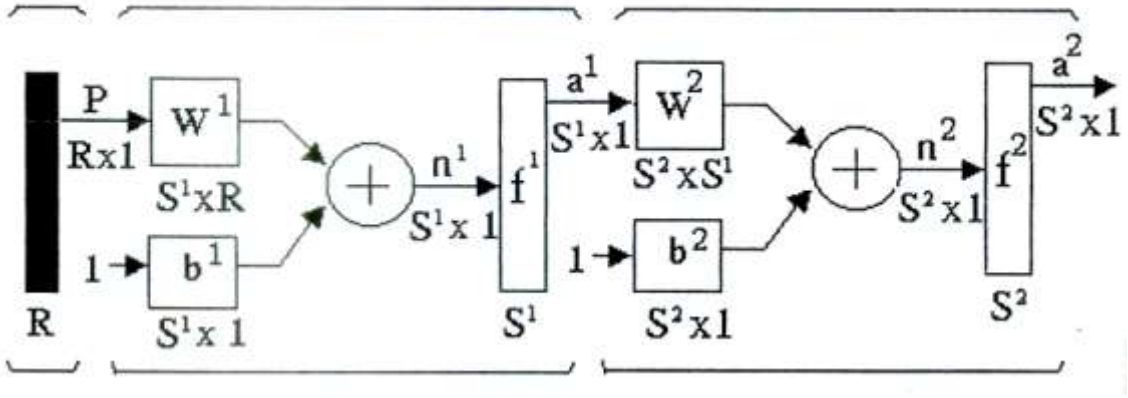
$$a = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_s \end{bmatrix} \qquad W = \begin{bmatrix} w_{1,1} & w_{1,2} \ldots & w_{1,R} \\ w_{2,1} & w_{2,2} \ldots & w_{2,R} \\ \ldots & \ldots & \ldots \\ w_{S,1} & w_{S,2} \ldots & w_{S,R} \end{bmatrix}$$

Lớp neural bao gồm ma trận trọng số, các bộ tổng, véctơ hệ số bias b. Một số tài liệu coi đầu vào là một lớp vào, với ý nghĩa lớp vào gồm các neural chỉ có chức năng nhận tín hiệu vào. Nhưng ở đây ta coi đầu vào là một véctơ các tín hiệu vào chứ không coi là một lớp các neural. Do đó mạng neural trên chỉ có một lớp (là lớp ra của mạng).

Neural thứ i trong lớp có hệ số bias b_1 , bộ tổng hàm truyền f , đầu ra a_i . Kết hợp các nơ-ron trong lớp thì đầu ra là véctơ a. Thông thường số đầu vào là R, các số neural S. Mỗi một neural trong lớp có thể có một hàm truyền riêng không nhất thiết tất cả các neural trong cùng một lớp thì phải có cùng một dạng hàm truyền.

1.4.2 Mạng neural nhiều lớp (Multiple Layers of Neurons)

Ta xét với mạng nhiều lớp. Mỗi lớp có ma trận trọng số W, véctơ bias B, véctơ net input n, và véctơ đầu ra a. Để phân biệt các lớp khác nhau ta dùng thêm chỉ số phụ cho mỗi biến. Do đó W^q để chỉ ma trận trọng số của lớp q, b^q chỉ véctơ bias của lớp q...



Đầu vào Lớp 1 (lớp ẩn) Lớp 2 (lớp ra)

$\mathbf{a^1 = f^1 (W^1 p + b^1)}$ $\mathbf{a^2 = f^2 (W^2 p + b^2)}$

Hình 1.7 Mô hình neural 2 lớp

Theo hình vẽ trên mạng có R đầu vào, có S^1 neural ở lớp thứ nhất, S^2 neural ở lớp thứ hai. Đầu ra của các lớp trước là đầu vào của lớp sau.lớp thứ hai có đầu vào gồm S^1 phần tử trong vectơ ra a^1, có ma trận W^2 với kích thước S^2 x S^1.

Lớp cuối cùng đưa ra kết quả của mạng gọi là lớp ra. Các lớp còn lại gọi là các lớp ẩn. Mạng trên cớ một lớp ẩn (lớp 1) và lớp ra (lớp 2).

Mạng nhiều lớp có khả năng lớn hơn mạng 1 lớp. Ví dụ mạng hai lớp với hàm truyền sigmoid ở lớp ẩn, hàm truyền tuyến tính tại lớp ra thì có thể được huấn luyện để xấp xỉ bất cứ hàm phi tuyến nào. Nhưng mạng một lớp không có khả năng này.

Tùy vào từng bài toán cụ thể mà ta lựa chọn số đầu vào, số neural trên lớp ra của mạng. Ví dụ nếu ta có 4 biến được sử dụng là đầu vào thì sẽ có mạng với 4 đầu vào, nếu có 2 tham số ra thì trên lớp ra của mạng sẽ có 2 neural ra tương ứng với 2 tham số ra đó. Dạng của hàm truyền tại lớp ra cũng phụ thuộc vào đặc tính của biến ra, Chẳng hạn nếu biến ra có giá trị nằm trong khoảng [-1,1] thì hàm truyền hard limit có thể được chọn cho các neural trên lớp ra.

Như vậy, đối với mạng neural một lớp thì kiến trúc mạng được thiết kế dễ dàng tùy thuộc vào bài toán. Nhưng đối với mạng neural một lớp thì kiến trúc mạng được thiết kế dễ dàng tùy thuộc vào bài toán. Nhưng đối với mạng neural

nhiều lớp (có ít nhất 1 lớp ẩn) thì vấn đề tìm ra số lớp ẩn và số neural trên từng lớp ẩn là rất khó. Đây vẫn là lĩnh vực đang được nghiên cứu. Trong thực tế chỉ dùng 1 đến 2 lớp ẩn. Trường hợp dùng 3 hay 4 lớp là rất hiếm.

Đối với mỗi neural có thể có hoặc không có hệ số mẫu *bias b*. Hệ số này tạo thêm cho mạng một biến phụ, do đó mạng có nhiều năng lực hơn so với mạng không có hệ số *bias.* Ví dụ đơn giản neural không có hệ số bias sẽ cho kết quả *net input n* là 0 nếu đầu vào *p* là 0. Điều này không tốt và có thể tránh được nếu neural có hệ số *bias*.

1.5 Phân loại mạng neural

Mạng neural nhân tạo là sự liên kết của các neural nhân tạo. Sự sắp xếp bố trí của các neural và cách thức liên hệ giữa chúng tạo nên kiến trúc mạng neural.

Theo cách sắp xếp neural thì có kiến trúc mạng 1 lớp (*single-layer*) là mạng chỉ có 1 lớp ra và kiến trúc mạng nhiều lớp (*multiple-layer*) là mạng có các lớp ẩn.

Theo cách liên hệ giữa các neural thì kiến trúc mạng truyền thẳng (*feedforward networks*) và kiến trúc mạng hồi quy (*recurrent networks*).

Ngoài ra, còn một loại liên kết theo sự phân bố các neural trong không gian hai chiều trong một lớp, gọi là liên kết bên (lateral conection). Với liên kết này, Kohonen đã tạo ra loại mạng tự tổ chức (*self- onrganizing neural network*).

Có thể phân các loại mạng neural thành hai nhóm chính dựa theo thuật toán học của chúng là loại học có giám sát (*supervised*) và không được giám sát (*unsupervised*).

*** Kiến trúc mạng truyền thẳng**

Kiến trúc mạng truyền thẳng (*feedforward*) là kiến trúc mà liên kết giữa các nơron không tạo thành chu trình. Tín hiệu đi từ các neural lớp vào lần lượt qua các lớp ẩn và cuối cùng đi ra ở neural lớp ra. Kiến trúc này có đáp ứng nhanh và ổn định đối với một tín hiệu đưa vào mạng. Liên kết giữa các lớp có

thể là loại liên kết đầy đủ (*fully connected*) hoặc liên kết một phần (*partly connected*).

1.6 Hoạt động của mạng neural nhân tạo

1.6.1 Hoạt động của mạng neural

Ta thấy rằng các neural trong cùng 1 lớp thì nhận tín hiệu đầu vào cùng một lúc. Do đó, về nguyên tắc chúng có thể xử lý song song. Hoạt động của mạng neural có thể xem như hoạt động của một hệ thống xử lý thông tin được cấu thành từ nhiều phần tử hoạt động song song. Khi mạng neural hoạt động, các thành phần của vectơ tín hiệu vào $\mathbf{p} = (p_1, p_2, ... , p_R)$ được đưa vào mạng, tiếp đó các neural ở lớp ẩn và lớp ra sẽ được kích hoạt dần dần. Sau một quá trình tính toán tại các neural mạng sẽ được kích hoạt hoàn toàn và cho ra vectơ tín hiệu đầu ra $\mathbf{a} = (a_1 , a_2 , ... , a_S)$ tại S neural lớp ra, Ta có thể coi mạng neural như một bảng tra cứu giữa **a** và **p** mà không cần biết hàm quan hệ tường minh của **a** theo **p**.

Sự khác biệt giữa mạng neural và hệ thống xử lý thông thường là khả năng thích nghi với dữ liệu vào. Đó là ma trận trọng số và hệ số *bias* của mạng có thể hiệu chỉnh để mạng thích nghi được với bài toán đặt ra. Quá trình hiệu chỉnh các trọng số và hệ số *bias* của mạng gọi là quá trình huấn luyện mạng (*training*) bằng một số luật học.

1.6.2 Luật học của mạng neural

Luật học là một thủ tục để điều chỉnh, thay đổi trọng số và hệ số *bias* của mạng (thủ tục này còn được gọi là thuật toán huấn luyện mạng). Mục tiêu của luật học là huấn luyện mạng để thực hiện một số nhiệm vụ mà ta mong muốn. Có rất nhiều luật học cho mạng neural. Chúng được chia làm 3 loại:

- Luật học có giám sát (*supervised learning*).
- Luật học không giám sát (*unsupervised learing*).
- Luật học tăng cường (*reinforcement learning*).

Trong khuôn khổ đồ án này ta chỉ nghiên cứu luật học có giám sát.

***Luật học có giám sát**

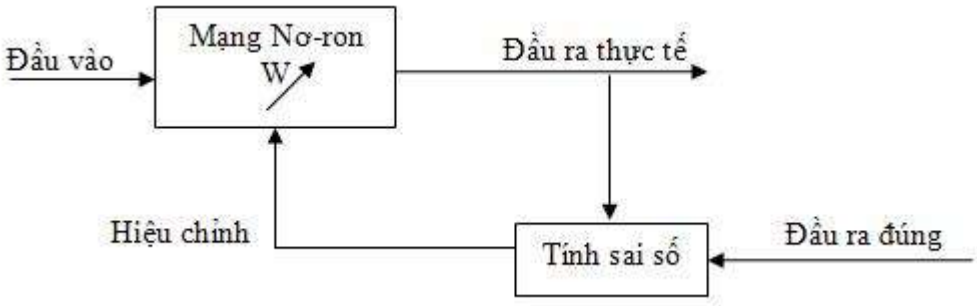
Một thành phần không thể thiếu của phương pháp này là sự có mặt của một người thầy (ở bên ngoài hệ thống). Người thầy này có kiến thức về môi trường thể hiện qua một tập hợp các cặp đầu vào - đầu ra đã được biết trước. Hệ thống học (ở đây là mạng neural) sẽ phải tìm cách thay đổi các tham số bên trong 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.

Trong luật học có giám sát: luật học được cung cấp một tập hợp các mẫu chuẩn (*trainig set*) thê hiện mối quan hệ giữa đầu vào và đầu ra của mạng:

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\}$$

Với p_q là một đầu vào của mạng và t_q tương ứng với đầu ra đích (target) là đầu ra mà trong mạng muốn đáp ứng. Khi đầu vào được đưa vào mạng thì đầu ra thực sự của mạng được so sánh với đầu ra đích. Sai số giữa đầu ra thực của mạng được so sánh với đầu ra đích. Sai số giữa đầu ra thực của mạng và đầu ra đích được sử dụng để điều chỉnh các trọng số và hệ số bias của mạng sao cho di chuyển đầu ra thực của mạng về gần hơn với đầu ra đúng.

Có hai cách sử dụng tập mẫu học: hoặc học lần lượt từng mẫu, hoặc tất cả các mẫu cùng một lúc.



Hình 1.8 Sơ đồ khối mô tả luật học giám sát

Để đánh giá sự sai lệch giữa vectơ đầu ra của mạng và đầu ra đúng người ta dùng hàm sai số (*error function*). Hàm sai số phổ biến nhất là hàm tổng bình

phương sai số (*sum square error function*) tính tổng bình phương các sai số tại đầu ra của các neural lớp ra.

Một khái niệm khác liên quan đến vấn đề đánh giá sai số là mặt sai số (*error surface*). Mỗi một trọng số và hệ số *bias* của mạng tương ứng với một chiều trong không gian, giả sử mạng có tất cả N trọng số và hệ số *bias*, thì chiều thứ nhất N+1 biểu diễn sai số của mạng. Mỗi một bộ trọng số và hệ số *bias,* thì chiều thứ N+1 biểu diễn sai số của mạng. Mỗi một bộ trọng số và hệ số *bias* của mạng sẽ ứng với một điểm của mặt sai số. Mục tiêu của luật học là tìm được bộ trọng số và hệ số *bias* ứng với điểm thấp nhất (điểm cực tiểu) của mặt đa chiều này.

CHƯƠNG II: MẠNG PERCEPTRON ĐA LỚP VỚI LUẬT HỌC LAN TRUYỀN NGƯỢC SAI SỐ

2.1 Mạng neural nhiều lớp lan truyền ngược sai số

2.1.1 Tổng quan về mạng neural truyền thẳng nhiều lớp

Mạng Perception một lớp chỉ có thể phân loại mẫu trong trường hợp không gian mẫu là khả tách tuyến tính (có thể phân chia được bằng các siêu phẳng). Trong trường hợp không gian mẫu không khả tách tuyến tính thì phải dùng mạng Perceptron đa lớp (MLP - *Multilayer Perceptron*). Kiến trúc mạng MLP là kiến trúc truyền thẳng đa lớp (có một hoặc nhiều lớp ẩn), Hàm truyền có thể nhiều dạng không phải chỉ là hàm hardlimit nhưng các neural trong cùng một lớp thì có dùng dạng hàm truyền.

Rosenblat và các tác giả đã mô tả các mạng truyền thẳng nhiều lớp từ cuối năm 50, nhưng họ chủ yếu chỉ nghiên cứu sâu về mạng Perceptron một lớp. Sở dĩ như vậy là do không tìm được cách thay đổi trọng sô liên kết tại các lớp ẩn. Quả thật, ngay cả khi đã biết được sai số tại đầu ra, người ta vẫn chưa hình dung được các sai số đó được phân bố như thế nào tại các neural ẩn. Trong cuốn sách về mạng Perceptron xuất bản năm 1969, Minsky và Papert đã chỉ ra rằng khó có thể tổng quát hóa luật học đối với mạng một lớp sang mạng nhiều lớp. Có hai vấn đề lý giải cho vấn đề này. Thứ nhất, thuật giải học của mạng nhiều lớp có thể không hiệu quả, hoặc không hội tụ về điểm cực trị tổng thể trong không gian véctơ trọng số. Mặt khác, nghiên cứu trong lý thuyết tính toán đã chỉ ra trong trường hợp tồi nhất quá trình học các hàm tổng quát từ mẫu học không phải lúc nào cũng giải quyết được. Các nguyên tắc cơ bản trong luật học đối với mạng nhiều lớp đã được Bryson và Ho đề suất từ năm 1969 nhưng phải tới năm 1980 vấn đề này mới được quan tâm trở lại bởi công trình nghiên cứu của Rumehart năm 1986 và từ đó mạng truyền thẳng nhiều lớp bắt đầu được ứng dụng rộng rãi. Một thống kê cho thấy 90% ứng dụng mạng neural trong công nghệ hóa học sử dụng mô hình này. Tuy nhiên, một số tác giả vẫn sử dụng các mạng này như các

bảng tra, liên kết bộ nhớ, phân lớp và đã thu được kết quả tốt, mặc dù nhiều mạng khác tỏ ra thích hợp hơn cho các nhiệm vụ kể trên.

Thủ tục học tham số của mạng neural truyền thẳng nhiều lớp thường dùng là thủ tục lan truyền ngược sai số. Trong thực tế thủ tục học lan truyền ngược sai số trong mạng neural nhiều lớp đã thông dụng đến mức có rất nhiều tác giả đã đánh đồng mạng neural với mạng neural nhiều lớp lan truyền ngược sai số. Sự hấp dẫn của thủ tục này nằm ở sự rõ ràng, rành mạch của phương trình hiệu chỉnh các trọng số. Các phương trình này được áp dụng cho việc hiệu chỉnh trọng số của từng lớp, bắt đầu từ lớp ra ngược dần lên đến lớp vào. Thủ tục hiệu chỉnh trọng số trong giải thuật lan truyền ngược sai số không giống như quá trình học của các neural sinh học. Thực chất của thủ tục lan truyền ngược sai số là thủ tục dịch chuyển ngược hướng gradient.

2.1.2 Kiến trúc mạng

Mạng Perceptron có kiến trúc mạng truyền thẳng đa lớp: có một hoặc nhiều lớp ẩn. Mỗi lớp có ma trận trọng số W, véctơ bias b, véctơ netinput n và véctơ đầu ra a. Để phân biệt các lớp khác nhau ta dùng thêm chỉ số phụ cho mỗi biến. Do đó, W^q để chỉ ma trận trọng số của lớp q, b^q chỉ véctơ bias của lớp q... Hàm truyền f có thể có nhiều dạng không phải chỉ là hàm sigmoid $f(x) = \frac{1}{1+e^{-n}}$, các Neural trong cùng một lớp thường có cùng dạng hàm truyền.

Theo hình vẽ trên mạng có R đầu vào, có S^1 nơron ở lớp thứ nhất, S^2 nơron ở lớp thứ hai. Đầu ra của lớp trước là đầu vào của lớp sau. Lớp thứ 2 có đầu vào là gồm S^1 phần tử trong vectơ ra a^1, có ma trận w^2 với kích thước S^2 xS^1. Lớp cuối cùng đưa ra kết quả của mạng gọi là lớp ra. Các lớp còn lạ gọi là các lớp ẩn.

2.1.3 Cơ chế huấn luyện của mạng neural lan truyền ngược sai số

Năm 1986, thuật toán huấn luyện cho mạng MLP được đưa ra bởi Rumelhart và McClelland. Nguyên lý của luật học này là việc lan truyền ngược sai số còn gọi là lan truyền ngược độ nhạy (*backpropagating the sensitivites*) từ

lớp ra trở lại các lớp ẩn và đến đầu vào mạng từ đó tìm cách hiệu chỉnh ma trận trọng số và các hệ số bias để tối thiểu hoá sai số trung bình bình phương (*mean square error*).

Các nghiên cứu và thực nghiệm cho thấy rằng: phương pháp học có giám sát với thuật toán lan truyền ngược sai số là phương pháp huấn luyện phổ biến và hiệu quả đối với mạng neural nhiều tầng truyền thẳng MLP áp dụng trong các bài toán phân loại mẫu [3]. Việc huấn luyện mạng với thuật toán lan truyền ngược sai số gồm hai pha ngược chiều nhau: quá trình truyền thẳng (lan truyền xuôi) và quá trình lan truyền ngược.

Với tập mẫu huấn luyện mạng $\{p_1, t_1\}, \{p_2, t_2\}, ..., \{p_Q, t_Q\}$ trong đó là đầu vào ra đích của mạng. Khi mẫu p_q được lan truyền qua mạng và cho ra tín hiệu đầu ra là a_q, thuật toán sẽ điều chỉnh các tham số của mạng để tối thiểu hoá sai số bình phương trung bình.

$$F(x)= E[e^2] = E[(t -a)^2] \quad (1.1)$$

với x là vectơ trọng số và hệ số bias của mạng được viết như sau:

$$x = \begin{bmatrix} W \\ b \end{bmatrix}$$

Khi mạng có nhiều đầu ra thì (1.1) được viết dưới dạng vectơ như sau;

$$F(x)= E[e^Te] = E[(t -a)^T(t-a)] \quad (1.2)$$

đây là trị trung bình của sai số bình phương và thực tế là không tính được mà chỉ có thể xấp xỉ bởi:

$$\hat{F}(x) = \frac{1}{2}(t-a)^{\tau}(t - a)$$

Mục tiêu của huấn luyện mạng là nhằm điều chỉnh W là b sao cho $\hat{F}(x)$ đạt giá trị nhỏ nhất. $\hat{F}(x)$còn được gọi là hàm chất lượng của mạng (*performance index*).

Giả sử N là tổng số trọng số và hệ số bias của mạng. Ta có thể coi mỗi trọng số và hệ số bias là một biến thì hàm $\hat{F}$ (x) là hàm gồm N biến. Về mặt hình học cơ thể xem $\hat{F}$ (x) là một mặt lỗi mà mỗi điểm của nó tương ứng với một bộ trọng số và hệ số bias. Để tìm điểm thấp nhất trên mặt lỗi ta dung phương pháp giảm dốc nhất (*steepest descent algorithm*) bằng cách lấy đạo hàm riêng của $\hat{F}$ (x) theo từng trọng số và hệ số bias. Hiệu chỉnh số và hệ số bias tại bước lặp thứ kiến trúc thượng tầng +1 theo công thức sau:

$$w^m_{i,j}(k+1) = w^m_{i,j}(K) - \alpha(k)\frac{\partial \hat{F}}{\partial w^m_{i,j}} \quad (1.4)$$

$$b^m_1(k+1) = b^m_1(k) - \alpha(k)\frac{\partial \hat{F}}{\partial b^m_1} \quad (1.5)$$

Với m là chỉ số lớp của mạng, còn α (k) là hê số học (learning rate) tại bước lặp thứ k.

Trong các lớp ẩn, $\hat{F}$.(x) không phải là một hàm hiện mà là hàm giám tiếp của các trọng số, vì vậy ta phải sử dụng đến luật dây chuyền (*chain ruler*) để tính các đạo hàm riêng. Với luật dây truyền, giả thiết hàm f là hàm hiện duy nhất của biến n, khi đó có thể tính đạo hàm của F theo biến số thứ ba là w như sau:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \cdot \frac{dn(w)}{dw} \quad (1.6)$$

Sử dụng 1.6 có thể tính đạo hàm riêng của $\hat{F}$ (x) trong các công thức (1.4) và (1.5). Ta có tín hiệu vào đầu *net input* của neural thứ i trong lớp thứ m như sau:

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \quad (1.7)$$

với s^{m-1} là số neural trên lớp m-1. Đây là một hàm hiện của các trọng số và hệ số bias. Theo đó ta có :

$$\frac{\partial n_j^m}{\partial w_{i,j}^m} = a_j^{m-1} \text{ và } \frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (1.8)$$

Sử dụng luật dây chuyền (1.6) ta có:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} a_j^{m-1} \quad (1.9)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_1^m} x \frac{\partial n_i^m}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \quad (1.10)$$

Trong công thức (1.9) và (1.10) ta đặt : $s_i^m = \frac{\partial \hat{F}}{\partial n_i^m}$ (1.11)

và coi là độ nhạy cảm của hàm $\hat{F}(x)$ đối với sự biến đổi tín hiệu vào *net input* của neural thứ i trên lớp m. Khi đó công thức trọng số bias (1.4) và (1.5) sẽ là:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad (1.12)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (1.13)$$

Ma trận độ nhạy cảm của lớp m có thể viết dưới dạng như sau:

$$s^m = \begin{bmatrix} \frac{\partial F}{\partial n_1^m} \\ \frac{\partial F}{\partial n_2^m} \\ ..... \\ \frac{\partial F}{\partial n_{s^m}^m} \end{bmatrix} \qquad (1.14)$$

Khi đó công thức (1.12) và (1.13) trở thành công thức ma trận như sau:

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^r \quad (1.15)$$

$$B^m(k+1) = B^m(k) - \alpha s^m \quad (1.16)$$

Với W^m là ma trận trong số B^m là ma trận hệ số bias của lớp m. Còn a^{m-1} là ma trận tín hiệu của lớp m-1.

Thuật toán lan truyền ngược được thực hiện theo nguyên tắc độ nhạy cảm của lớp m sẽ được tính toán từ độ nhạy của lớp m+1 đã được tính trước đó. Do vậy, độ nhạy sẽ được lan truyền ngược từ lớp ra trở lại các lớp ẩn.

Điều này được diễn tả như sau:

$$S^m \text{->} \; S^{m-1} -> ... -> S^2 -> s^1$$

Với s^1 là ma trận độ nhạy cảm của lớp ẩn thứ nhất.

Để tìm mối quan hệ độ nhạy giữa lớp m+1 và lớp m ta có:

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial (\sum_k^{s^m} w_{i,k} a_k^m + b_i^{m+1})}{\partial n_j^m} = w_{i.j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \quad (1.17)$$

Ta lại có tín hiệu của neural j của lớp m được tinh theo hàm truyền f của nó với đối số là tín hiệu vào : $a_j^m = f^m(n_j^m)$ (1.18)

Thay vào (1.17) tra có :

$$\frac{\partial n_i^{m+1}}{\partial n_i^m} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^m f^{,m}(n_j^m) \quad (1.19)$$

Theo công thức (1.19) ta phải tính được đạo hàm truyền f theo tín hiệu đầu vào.

Nếu hàm truyền trên lớp m của mạng là hàm sigmoid thì:

$$f'^{m}(n_i^m) = a_i^m(1-a_i^m) \qquad \frac{\partial n^{m+1}}{\partial n^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_{s^{r+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s_{r+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{s^{r+1}}^{m+1}}{\partial n_{s^{r+1}}^m} \end{bmatrix}$$

Để có công thức dạng ma trận, ma trận Jacobian được định nghĩa như sau:

Định nghĩa ma trận đạo hàm truyền của lớp m:

Công thức (1.19) ở dạng ma trận sẽ là :

$$\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1}F'^{m}D(n^{m)} \quad (1.20)$$

Sử dụng luật dây chuyền ta có.

$$F'^{m}(n^m) = \begin{bmatrix} f'^{m}(n_1^m) & 0 & \dots & 0 \\ 0 & f'^{m}(n_2^m) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & f'^{m}(n_{s^m}^m) \end{bmatrix}$$

$$s^m = \frac{\partial F}{n^m} = \left(\frac{\partial n^{m+1}}{\partial n^m}\right)^T \frac{\partial F}{\partial n^{m+1}} = F'^{m}(n^m)(W^{m+1})^T s^{m+1} \quad (1.21)$$

Theo (1.21) thì s^m có thể được tính toán từ S^{m+1}. Công thức này thể hiện lan truyền ngược độ nhạy. Điểm khởi đầu quá trình lan truyền ngược sẽ bắt đầu với việc tính S^m - tức độ nhạy tại lớp ra:

$$S_i^M = \frac{\partial F}{\partial n_i^M} = \frac{\partial \frac{1}{2}\sum_{j=1}^{s^m}(t_j - a_j)^2}{\partial n_i^m} = -(t_i - a_i)\frac{\partial a_I}{\partial n_i^M} \quad (1.22)$$

Vì a_i là tín hiệu của neural i lớp M nên ta có:

$$\frac{\partial a_i}{n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = f'^{M}(n_i^M) \quad (1.23)$$

Thay vào (1.22) ta suy ra:

$$S_i^M = -(t_i - a_i)f'^{M}(n_i^M) \quad (1.24)$$

Dạng ma trận s^M sẽ là:

$$S^M = -F'^M(n^M)(t-a) \quad (1.25)$$

Với a là véc tơ đầu ra của mạng, t là vectơ đích tương ứng với đầu vào.

Với mỗi cặp dữ liệu mẫu $P_{q'}, t_q$ có một hàm sai số $\hat{F}(x)$. Luật học lan truyền ngược truyền tìm cách tối thiểu hoá hàm này để mạng có thể học được mẫu P_q theo nghĩa khi cho vectơ P_q hoặc một vectơ gần với vectơ p_q lan truyền qua mạng thì đáp ứng đầu ra a_q của mạng gần nhất với vectơ t_q.

Tập mẫu học có Q cặp dữ liệu thì sẽ có Q hàm như vậy và ta phải tìm được tập trọng số của mạng đồng thời tối thiểu hoá được tất cả Q hàm đó.

Kết thuc một lần đưa tấtc các mẫu huấn luyện qua mạng được gọi là một lần lặp epoch. Kết quả học được đánh giá bởi hàm trung bình của Q hàm trên gọi là hàm sai số RMS (*root mean square error*).

$$\text{RMS} = \frac{1}{2}\frac{\sum_{s=1}^{0}\sum_{t=1}^{n}(t_{si}-a_{si})^2}{Q}$$

n là số neural trên lớp ra

Rõ ràng khi tập mẫu càng lớn thì việc tìm ra bộ trọng số đồng thời tối thiểu hoá được tất cả các hàn sai số là càng khó.

Do đó có mối liên quan giữa độ phức tạp của tập mẫu và năng lực học của mạng

Tập mẫu càng lớn càng phức tạp thì năng lực học của mạng càng đòi hỏi cao.

Quá trình huấn luyện dừng khi đồng thời giá trị hàm RSM ở trên và giá trị cực đại của hàm sai số $\hat{F}$ (x) ứng với tất cả các mẫu trong tập học giảm đến một giá trị thấp nhận được (nhỏ hơn ngưỡng định ra ban đầu)

* Tóm lại thuật toán huấn luyện mạng MLP lan truyền ngược như sau:

Chuẩn bị tập mẫu đầu vào dưới dạng $\{p_i, t_i\}$ trong đó x_i là đầu vào cho mạng và t_i là giá trị đầu ra mong muốn tương ứng với p_i (còn gọi là giá trị mục tiêu).

Các bước thực hiện

- Bước 1: Khởi tạo ban đầu các giá trị , đặt dữ liệu đầu vào $\{p_i, t_i\}$ lan truyền giá trị qua các neural ở các lớp từ lớp ẩn đầu tiên cho đến lớp cuối cùng.
- Bước 2: Tính giá trị lỗi ở lớp ra, kiểm tra giá trị lỗi có nhỏ hơn ngưỡng không đồng thời kiểm tra đầu ra của mạng và t_i có trùng nhau không nếu đúng, nhẩy tới B6 đồng thời tăng số mẫu nhận dạng đúng trong tập mẫu lên 1. Còn sai thì thực hiện B3.
- Bước 3: Tính toán lỗi và lan truyền ngược các giá trị lỗi từ lớp ra cho đến lớp ẩn đầu tiên
- Bước 4: Tính lại trọng số liên kết từ lớp ẩn đầu tiên cho đến lớp ra
- Bước 5: cộng thêm giá trị lỗi của mẫu p_i vào giá trị tổng lỗi và đồng thời tăng số mẫu chưa nhận dạng đúng lên 1.
- Bước 6: Kiểm tra đã học hết mẫu trong tập chưa nếu sai quay lại B1 nếu đúng làm B7
- Bước 7: kiểm tra số mẫu nhận dạng đúng = số mẫu trong tập thì kết thúc việc học ngược lại thì quay lại B1 bắt đầu học với lần tiếp theo.

Ngoài ra thuật toán dừng khi đã hết số lần học hoặc tỉ lệ lỗi của các mẫu chưa nhận dạng đúng nhỏ hơn ngưỡng cho phép. Trong quá trình truyền thẳng các tham số mạng là cố định, ngược lại trong quá trình lan truyền ngược các tham số này được hiệu chỉnh và cập nhật để mạng có thể nhận dạng các mẫu đã được học một cách chính xác. Có nhiều yếu tố ảnh hưởng đến quá trình huấn luyện mạng, đó là: hàm truyền *f*, hàm giá E, hệ số học η và các tham số khởi tạo. Do đó ta cần lưu ý trong việc lựa chọn các yếu tố này sao cho phù hợp.

Kết thúc một lần đưa tất cả các mẫu huấn luyện qua mạng được gọi là một lần *epoch*. Quá trình huấn luyện lặp cho đến khi sai số ở đầu ra mạng đạt đến một giá trị chấp nhận được (nhỏ hơn ngưỡng sai số ta định ban đầu). Thường thì thủ tục học của mạng lan truyền ngược sai số khá lâu. Nó đòi hỏi hàng nghìn

hoặc hàng chục nghìn epoch mới có thể hội tụ tới lời giải. Nếu các tham số không đúng, thủ tục học có thế không hội tụ.

2.2 Các nhân tố của quá trình học lan truyền ngược sai số

Trong phần này chúng ta sẽ đề cập đến các nhân tố quan trọng ảnh hưởng tới tốc độ hội tụ, tốc độ học, các cực tiểu cục bộ, khả năng dự báo của mạng lan truyền ngược sai số. Các nhân tố đó là:

+ Khởi tạo các trọng số

+ Hằng số học Anpha

+ Hàm giá (sự đánh giá sai số trên lớp ra)

+ Các luật cập nhật

+ Tập mẫu học và sự báo

+ Cấu trúc mạng (số lớp, số neural trên mỗi lớp)

+ Các hàm biến đổi

2.2.1 Khởi tạo các trọng số

Các giá trị của trọng số được khởi tạo ban đầu của mạng lan truyền ngược sai số ảnh hưởng rất lớn đến tới lời giải cuối cùng. Các trọng số này thường được khởi tạo bằng những số ngẫu nhiên nhỏ. Việc khởi tạo tất cả các trọng số bằng nhau sẽ làm cho việc học của mạng trở nên không tốt. Các trọng số khởi tạo ban đầu cũng không được quá lớn vì nếu không hàm sigmoidal hoặc các hàm dạng này sẽ bị bão hòa ngay từ lúc bắt đầu hoặc bằng 0 hoặc bằng 1. Điều này làm cho hệ thống sẽ bị tắc ngay tại một điểm cục bộ hoặc tại mộ vùng bằng phẳng nào đấy gần ngay tại điểm xuất phát.

Một cách tốt nhất để chọn các trọng số ban đầu là thực hiện một cách tự động bằng chương trình. Giá trị khởi động ban đầu cả các trọng số trên lớp thứ

m được chọn ngẫu nhiên trong khoảng [-1/n , 1/n], trong đó n là tổng trọng số có trên lớp m.

2.2.2 Hằng số học α (Anpha)

Một nhân tố ảnh hưởng tới hiệu lực và độ hội tụ của giải thuật học lan truyền ngược sai số là hằng số học α . Không có một giá trị duy nhất tốt phù hợp với tất cả các bài toán khác nhau. Hằng số học được chọn bằng thực nghiệm cho mỗi bài toán ứng dụng cụ thể bằng phương pháp thử và sai. Giá trị α lớn làm tăng tốc quá trình hội tụ. Điều này là không thuận lợi vì thủ tục học sẽ kết thúc rất nhanh tại một cực tiểu cục bộ gần nhất. Nếu giá trị của hằng số học quá nhỏ tốc độ hội tụ của giải thuật lại trở nên rất chậm. Do đó chúng ta cần chọn một giá trị thỏa hiệp giữa tốc độ học và việc ngăn chặn về các cực tiểu cục bộ. Các giá trị của hằng số học nằm trong khoảng 10^{-3} và 10 đã được sử dụng thành công cho rất nhiều bài toán cụ thể. Nói chung giá trị của hằng số học nằm trong khoảng [0.3 , 0.6] được chứng minh bằng thực nghiệm là khá tốt cho việc lựa chọn ban đầu của quá trình tìm kiếm hằng số học thích hợp.

Một vấn đề nảy sinh khác là các giá trị tốt của hằng số học tại thời điểm bắt đầu của quá trình học có thể là không tốt đối với giai đoạn sau của quá trình học. Do đó một phương pháp hiệu quả hơn là sử dụng hằng số học thích nghi. Phương pháp trực giác là kiểm tra xem liệu các trọng số mới đã làm hàm giảm giá chưa. Nếu chưa khi đó quá trình hiệu chỉnh các trọng số đã phóng quá xa, lúc này hằng số học α cần phải được giảm đi. Mặt khác nếu vài vòng lặp liên tiếp làm giảm hàm giá, khi đó chúng ta cần phải thử tăng giá trị của hàm số học. Bằng biểu thức, hằng số học thích nghi cần được cập nhật theo quy tắc sau:

$\Delta\alpha = +a$ nếu ΔE "luôn" nhỏ hơn 0

$\Delta\alpha = -b\alpha$ nếu $\Delta E > 0$

$\Delta\alpha = 0$ trường hợp còn lại

Trong đó ΔE là thay đổi hàm giá, a,b là các hằng số dương. Ý nghĩa của việc "luôn" nhỏ hơn 0 được đánh giá dựa trên k lặp liên tiếp.

2.2.3 Tập mẫu học và dự báo

Chúng ta luôn yêu cầu tập mẫu học phải đủ và đúng. Tuy nhiên không có một thủ tục hay một nguyên tắc chọn mẫu học phù hợp cho mọi trường hợp. Một nguyên tắc rút ra từ kinh nghiệm là tập mẫu học phải bao phủ toàn bộ không gian của tín hiệu vào và sau đó sau quá trình học của véctơ vào - lời giải nên chọn một cách ngẫu nhiên từ tập mẫu học. Một cách chính xác hơn, giả định là không gian tín hiệu vào được phân chia một cách tuyến tính thành M các vùng riêng biệt với ranh giới là các siêu phẳng (*hyperplans*). Gọi P là giới hạn dưới của số mẫu học trong tập mẫu. Nếu chọn P sao cho tỉ số P/M lớn hơn một rất nhiều có thể cho phép mạng trong quá trình học phân biệt được các vùng tách rời ra. Trong một số trường hợp việ tỉ lệ hóa hay chuẩn hóa mẫu học sẽ giúp cho việc học của mạng tốt hơn nhiều. Đối với các mạng neural lan truyền ngược sai số sử dụng hàm biến đổi là sigmoidal khi đó véctơ vào và véctơ ra lời giải cần được tỷ lệ hóa một cách đúng đắn.

Mạng Neural nhiều lớp lan truyền ngược sai số rất tốt cho việc dự báo hay việc sản sinh một lời giải.Một mạng được gọi là dự báo tốt khi nó nội suy các tín hiệu vào chưa biết một cách hợp lý. Nói một cách khác là mạng có khả năng nội suy tốt. Một mạng neural lan truyền ngược sai số với số lượng các trọng số lớn (số tham số học lớn) với một tập mẫu học có một số lượng mẫu học nhất định có thể học rất tốt. Hiện tượng này thường được gọi là quá trình học quá (overtrain or overfiting). Với số lượng trọng số quá nhỏ, mạng neural có thể không học được các mẫu học và cho kết quả rất tồi đối với tập kiểm tra. Để cải thiện khả năng học và dự báo, chúng ta phải làm cho mạng ít nhạy cảm với những thay đổi nhỏ của tín hiệu vào. Nói một cách khác là đối với một thay đổi nhỏ của tín hiệu vào, các thành phần của tín hiệu ra không thay đổi. Điều này có thể được thực hiện bằng việc thêm vào tập mẫu học những thay đổi nhỏ của tín hiệu vào vẫn

giữ nguyên giá trị véctơ lời giải. Điều này có một bất lợi nhỏ là sẽ kéo dài thời gian tính toán trong quá trình học. Một cách khác là biến đổi dạng của hàm giá trong giải thuật lan truyền ngược sai số.Hàm giá mới là tổng của hàm giá được đinh nghĩa trong phương trình 4.4 và một số hạng là hàm jacobien của hàm giá:

$$E_n = \frac{1}{2}\left(\frac{\partial E}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial E}{\partial x_2}\right)^2 + \ldots + \frac{1}{2}\left(\frac{\partial E}{\partial x_m}\right)^2$$

Trong đó x_j là thành phần thứ j của véc-tơ vào, lý do căn bản cho giải pháp trên là tín hiệu vào thay đổi nhỏ, hàm giá mới sẽ không thay đổi. Để tối thiểu hóa hàm giá mới, các tác giả Drucker và LeCum đã xây dựng một mạng lan truyền ngược kép. Mạng lan truyền ngược kép: một lan truyền hai tín hiệu sai số $\partial E / \partial w_j$; và một lan truyền tín hiệu $\partial E_b / \partial w_j$. Kỹ thuật này đã cải thiện rõ rệt khả năng dự báo của mạng neural nhiều lớp lan truyền ngược sai số.

2.3 Cấu trúc mạng

Mạng neural lan truyền ngược sai số như đã nói ở phần cấu trúc luôn có một lớp vào và một lớp ra.Số lớp ẩn có thể thay đổi từ 0 đến vài lớp. Đối với một bài toán cụ thể, số lượng neural trên lớp vào và ra là cố định vì số neural trên lớp vào bằng số biến của véc-tơ vào và số neural trên lớp ra bằng số biến của véc-tơ lời giải.

Như đã đề cập, đại đa số các mạng neural lan truyền ngược sai số đã được công bố chỉ gồm có 1 lớp ẩn, song kích thước của lớp ẩn này (số lượng neural trên lớp ẩn) là một câu hỏi luôn được đặt ra cho các ứng dụng sử dụng mạng neural lan truyền ngược. Các phân tích chính về vấn đề số lượng neural trên lớp ẩn có lẽ sẽ không thực hiện được bởi tính phức tạp và bản chất không tiền định của các thủ tục học. Do đó kích cỡ của lớp ẩn thường được xác định bằng thực nghiệm. Một chỉ dẫn mang tính chung là đối với những mạng neural có kích thước đáng kể (tín hiệu vào có khoảng hàng trăm hoặc hàng nghìn biến), kích thước của lớp ẩn cần thiết ban đầu chỉ là 1 phân số nhỏ của neural trên lớp vào.

Nếu mạng không có khả năng hội tụ về lời giải, chúng ta cần tăng dần số neural trên lớp ẩn. Nếu mạng có khả năng hội tụ về lời giải, chúng ta cần giảm số lượng neural trên lớp ẩn.

2.4 Sự hội tụ của thuật toán huấn luyện mạng

Thuật toán huấn luyện ngược hội tụ đến một giải pháp nào đó mà nó tối thiểu hóa được sai số trung bình bình phương vì cách thức hiệu chỉnh trọng số và hệ số bias của thuật toán là ngược hướng với véc-tơ gradient của hàm sai số trung bình bình phương đối với trọng số.

Ta có thể hình dung phương pháp này như sau: ném 1 quả bóng trên mặt lỗi rồi để nó tự lăn xuống điểm thấp nhất trên mặt lỗi. Tuy nhiên, đối với mạng MLP thì hàm sai số trung bình bình phương thường phức tạp và có nhiều cực trị địa phương nên mặt lỗi sẽ nhấp nhô, có nhiều điểm trũng (các cực trị đựa phương). Do đó, khi quả bóng được thả từ 1 điểm ngẫu nhiên trên mặt lỗi thì nó có thể lăn xuống điểm trũng gần đó nhất và điểm trũng này chưa chắc đã là điểm trũng thấp nhất trên mặt lỗi. Vì thế các phép lặp huấn luyện mạng có thể chỉ đạt đến các cực trị cục bộ của hàm sai số trung bình bình phương mà không đạt đến giá trị cực trị toàn cục. Quá trình huấn luyện sẽ hội tụ như thế nào phụ thuộc vào các điều kiện ban đầu của quá trình huấn luyện. Đặc biệt là việc chọn hệ số học α như thế nào sau từng bước lặp để tăng khả năng hội tụ của mạng.

Như vậy, khi một quá trình huấn luyện theo thuật toán lan truyền ngược hội tụ, ta chưa thể khẳng định rằng nó đã hội tụ đến phương án tối ưu mà mới chỉ đặt ra ngưỡng sai số để đánh giá. Nếu ngưỡng sai số tương đối nhỏ thì có thế dừng quá trình huấn luyện mang.

CHƯƠNG III: KỸ THUẬT NHẬN DẠNG BẢN RÕ TIẾNG ANH CỦA NGÔN NGỮ TỰ NHIÊN

3.1 Bài toán

Cho một mẫu văn bản x nào đó thuộc ngôn ngữ Anh hoặc dãy ngẫu nhiên (chữ cái hoặc chuyển sang mã cơ số 16) nhưng chưa biết x thuộc loại nào trong 2 loại nêu trên. Hãy xác định xem x thuộc ngôn ngữ cụ thể nào.

Ta ký hiệu A_0 là lớp đại diện cho ngôn ngữ không đọc được, A_1 đại diện chop lớp ngôn ngữ tiếng Anh. Vậy bài toán là hãy xác định xem x thuộc lớp đại diện nào trong 2 lớp vừa nêu?

Nội dung thuật toán gồm 2 phần: phần off-line (xây dựng sơ sở dữ liệu để máy học) và phần on-line (phần nhận biết trực tiếp). Sau đây là nội dung cụ thể của 2 phần đó.

3.2 Thuật toán

3.2.1 Phần off-line

Xây dựng ước lượng ma trận các xác suất chuyển trạng thái ứng với các ngôn ngữ tự nhiên tiếng Anh. Trong ước lượng này, chọn độ dài mẫu xấp xỉ 10.000 ký tự, bỏ qua dấu chấm (.), dấu phẩy (,), ... , không phân biệt chữ in hoa chữ in thường. Nó là tổng hợp các loại văn bản thuộc các chuyen ngành khác nhau: chính trị, kinh tế, văn học,địa lý, quân sự, thể thao, ngoại giao, lịch sử, y tế, giáo dục, pháp luật. Mỗi loại được chọn xấp xỉ 1000 ký tự.

Tính tần số bộ đôi móc xích của dãy X, tức là ta tính số lần xuất hiện các cặp chữ cái Latinh của dãy đó. Giả sử, tần số các chữ cái đó lần lượt được ký hiệu là $X=(x_{ij})$

$$x_{t,t+1}(i,j) = \begin{cases} 1 & \text{Nếu cặp (i,j) xuất hiện tại thời điểm (vị trí) t và t+1} \\ 0 & \text{Trong trường hợp khác} \end{cases}$$

với i, j =a, b, .. , z

$t = \overline{1,N-1}$

Rõ ràng ta có $n_{ij} = \sum_{t=1}^{n-1} X_{t,t+1}(i,j)$ là số lần xuất hiện cặp ký tự (i,j) đứng kề nhau.

Ta có xác suất xuất hiện cặp (i,j) là P_{ij} . Do các ngôn ngữ có tính dừng nên xác suất chuyển P_{ij} không phụ thuộc vào t từ là:

$$x_{t,t+1}(i,j) = \begin{cases} 1 & \text{Với xác suất } P_{ij} \\ 0 & \text{Với xác suất } q_{ij} = 1 - P_{ij} \end{cases}$$

Bây giờ ta tính kỳ vọng toán học của đại lượng ngẫu nhiên n_{ij} , ta có kỳ vọng

$$E[n_{ij}] = E\left[\sum_{t=1}^{n-1} X_{t,t+1}\ \ i,j\ \right] = \sum_{t=1}^{n-1} E.X_{t,t+1}\ \ i,j\ \ = \ n-1\ P_{ij} \qquad \forall\, i,j\ ;\ 1 \le i\ ;\ j \le 26$$

Vì vậy rõ ràng là $\frac{n_{ij}}{n-1}$ là ước lượng không chệch đối với tham số P_{ij} $\forall\, P_{ij}$.

Bây giờ, giả sử cho i cố định, bằng lý luận hoàn toàn tương tự, ta có $\frac{n_{ij}}{n-1}$ là ước lượng không chênh của P_{ij} với i cố định.Trong đó $n_i = \sum_{j=1}^{26} n_{ij}; i = 1,2,...,26$

Tuy nhiên có tồn tại một ước lượng có chênh tốt hơn ước lượng không chệch theo nghĩa sai số trung bình bình phương nhỏ nhất. Ước lượng đó là

$\hat{P}_{ij} = \frac{n_{ij} + c}{n_i + m.c}$ trong đó c là hằng số thường được chọn, c = 0,5 hoặc c = 1/m còn m =26 (số chữ cái của bảng ngôn ngữ). Nếu c = 1/m ta có:

$$\hat{P}_{ij} = \frac{n_{ij} + 0,0385}{n_i + 1} \; ; i, j = 1, 2 , ..., 26$$

Đó là ước lượng có chênh nhưng sai số trung bình bình phương bé nhất của xác suất chuyển P_{ij} của ma trận chuyển P.

Kết quả tính $\hat{P}_{ij}$ được cho ở bảng tương ứng sau:

BẢNG 3.1. ƯỚC LƯỢNG BỘ ĐÔI MÓC XÍCH TIẾNG ANH (A_1)

	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	4	21	29	53	2	11	31	4	33	4	6	65	23	171	2	14		87	77	127	7	25	8	1	9	1
B	13				56				8	2		23			11			15	4	2	13				15	
C	33		7	1	71			36	18		10	9	1		51	3		10		29	11				3	
D	41	16	10	5	66	18	3	9	59		1	4	15	6	16	4		21	18	53	21	5	15		3	
E	88	20	58	126	57	41	20	16	51	1	4	56	57	154	35	7	6	194	153	66	9	26	31	13	5	
F	19	3	5	1	19	21	1	3	30	2		11	1		52			26	8	49	7	3	3		2	
G	21	4	3	2	35	1	3	15	18			6	1	4	21	1	1	23	9	21	9		5		1	
H		1	4		271	5	1	6	58				3	2	44	1		3	12	18	6		5		3	
I	40	7	53	24	25	9	11	3			2	38	26	212	58	2	1	46	82	120	1	3		4		3
J	3				5				1						46						3					
K	2				11				13					2	2				6	2	1		2		1	
L	46	2	5	12	65	7	5	2	44	1	1	54	2	2	26	1	1	2	16	23	9		1		34	
M	52	14	1		67			3	1				7	1	17	9	1	2	12	3	8		1		2	
N	42	10	48	126	63	19	107	12	32	1	6	6	9	7	54	7	1	7	47	132	6	5	14		13	
O	7	12	14	17	5	97	3	5	14			19	2	136	13	3		95	23	42	58	16	28		4	1
P	19	1			37			4	8			15	1		28	9		33	14	7	6					
Q																					17					
R	86	8	16	24	177	4	8	8	78	1	10	6	26	16	60	4		24	38	56	6	11	4		29	
S	66	9	17	9	76	13	1	47	78	3		8	11	12	58	7	6	9	48	121	35	1	28		4	
T	59	23	7	1	79	6	2	332	133	1		14	10	6	79	7		51	50	56	21	3	30		25	
U	12	5	9	6	9	1	6		9		1	20	5	31	2	5		48	40	31		3			1	
V	7				77				29						5										3	
W	38	1	1		39			33	36			4	1	8	15				4	2			1			
X	1		2			1			3						1	5				4			1			
Y	15	5	4	2	7	12	2	6	10			3	7	5	20	8		4	16	31			5			
Z	1				4																					

Từ số liệu của bảng A1

$A_1 = (a_{ij}^{(1)})$ i, j = 1, 2, ... , 26

Trong đó:

$$b_{ij}^{(1)} = \begin{cases} \left[7.\lg \dfrac{14,79}{a_{ij}^{(1)}} \right] & \text{if } a_{ij}^{(1)} > 0 \\ 11 & \text{if } a_{ij}^{(1)} = 0 \end{cases} \qquad i, j = 1 , 2, ... , 26$$

Trong đó lg(.) là logarit cơ số 10

[x] = số nguyên lớn nhất nhưng không bé hơn hoặc bằng x

Hệ số k = 7 này là kết quả thực nghiệm giúp cho việc nhận dạng giữa các lớp được tốt hơn.

Gọi $A = (a_{ij})_{26x26}$ với $a_{ij} = 1/26 \ \forall$ i,j = 1, 2, ... , 26. Ma trận của dãy ngẫu nhiên.

Như vậy, mỗi phần tử $A_0 = (a_{ij}^{(1)})$ của ma trận bộ đôi của dãy ngẫu nhiên là hệ số $14,79 \approx \dfrac{10.000}{26*26}$, lấy 2 chữ số thập phân sau dấu phẩy.

Ví dụ 3.1: [-1,5] = -2 [-1,95] = -2 [-1] = -1

[1,5] = 1 [3] = 3 [0,3] = 0

Ta có bảng như sau:

BẢNG 3.2: ƯỚC LƯỢNG ĐỐI SÁNH CỦA TIẾNG ANH VỚI MẪU NGẪU NHIÊN (B1)

	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	4	-2	-3	-4	6	0	-3	4	-3	4	2	-5	-2	-8	6	0	11	-6	-6	-7	2	-2	1	8	1	8
B	0	11	11	11	-5	11	11	11	1	6	11	-2	11	11	0	11	11	-1	4	6	0	11	11	11	-1	11
C	-3	11	2	8	-5	11	11	-3	-1	11	1	1	8	11	-4	4	11	1	11	-3	0	11	11	11	4	11
D	-4	-1	1	3	-5	-1	4	1	-5	11	8	4	-1	2	-1	4	11	-2	-1	-4	-2	3	-1	11	4	11
E	-6	-1	-5	-7	-5	-4	-1	-1	-4	8	4	-5	-5	-8	-3	2	2	-8	-8	-5	1	-2	-3	0	3	11
F	-1	4	3	8	-1	-2	8	4	-3	6	11	0	8	11	-4	11	11	-2	1	-4	2	4	4	11	6	11
G	-2	4	4	6	-3	8	4	-1	-1	11	11	2	8	4	-2	8	8	-2	1	-2	1	11	3	11	8	11
H	11	8	4	11	-9	3	8	2	-5	11	11	11	4	6	-4	8	11	4	0	-1	2	11	3	11	4	11
I	-4	2	-4	-2	-2	1	0	4	11	11	6	-3	-2	-9	-5	6	8	-4	-6	-7	8	4	11	4	11	4
J	4	11	11	11	3	11	11	11	8	11	11	11	11	11	-4	11	11	11	11	11	4	11	11	11	11	11
K	6	11	11	11	0	11	11	11	0	11	11	11	11	6	6	11	11	11	2	6	8	11	6	11	8	11
L	-4	6	3	0	-5	2	3	6	-4	8	8	-4	6	6	-2	8	8	6	-1	-2	1	11	8	11	-3	11
M	-4	0	8	11	-5	11	11	4	8	11	11	11	2	8	-1	1	8	6	0	4	1	11	8	11	6	11
N	-4	1	-4	-7	-5	-1	-7	0	-3	8	2	2	1	2	-4	2	8	2	-4	-7	2	3	0	11	0	11
O	2	0	0	-1	3	-6	4	3	0	11	11	-1	6	-7	0	4	11	-6	-2	-4	-5	-1	-2	11	4	8
P	-1	8	11	11	-3	11	11	4	1	11	11	-1	8	11	-2	1	11	-3	0	2	2	11	11	11	11	11
Q	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	-1	11	11	11	11	11
R	-6	1	-1	-2	-8	4	1	1	-6	8	1	2	-2	-1	-5	4	11	-2	-3	-5	2	0	4	11	-3	11
S	-5	1	-1	1	-5	0	8	-4	-6	4	11	1	0	0	-5	2	2	1	-4	-7	-3	8	-2	11	4	11
T	-5	-2	2	8	-6	2	6	-10	-7	8	11	0	1	2	-6	2	11	-4	-4	-5	-2	4	-3	11	-2	11
U	0	3	1	2	1	8	2	11	1	11	8	-1	3	-3	6	3	11	-4	-4	-3	11	4	11	11	8	11
V	2	11	11	11	-6	11	11	11	-3	11	11	11	11	11	3	11	11	11	11	11	11	11	11	11	4	11
W	-3	8	8	11	3	11	11	-3	-3	11	11	4	8	1	-1	11	11	11	3	6	11	11	8	11	11	11
X	8	11	6	11	11	8	11	11	4	11	11	11	11	11	8	3	11	11	11	4	11	11	8	11	11	11
Y	-1	3	4	6	2	0	6	2	1	11	11	4	2	3	-1	1	11	4	-1	-3	11	11	3	11	11	11
Z	8	11	11	11	4	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11

3.2.2 Phần on-line

Giả sử X là mẫu nào đó $X = x_1x_2...x_N$ với $x_i \in \{a,b,...,z\}$, i= 1,2,.. N. Vấn đề đặt ra X thuộc ngôn ngữ tiếng Anh hay là một dãy ngẫu nhiên nào đó?

Ta tiến hành như sau:

Step 1: Tính tần số bộ đôi móc xích của dãy X, tức là ta tính số lần xuất hiện các cặp chữ cái Latinh của dãy đó. Giả sử, tần số các chữ cái đó lần lượt được ký hiệu là $F = (f_{ij})$ với i,j = a,b,...,z

Trong đó:

$$f_{t,t+1}(i,j) = \begin{cases} 1 & \text{Nếu cặp (i,j) xuất hiện tại thời điểm (vị trí) t và t+1} \\ 0 & \text{Trong trường hợp khác với t} = \overline{1,N-1} \end{cases}$$

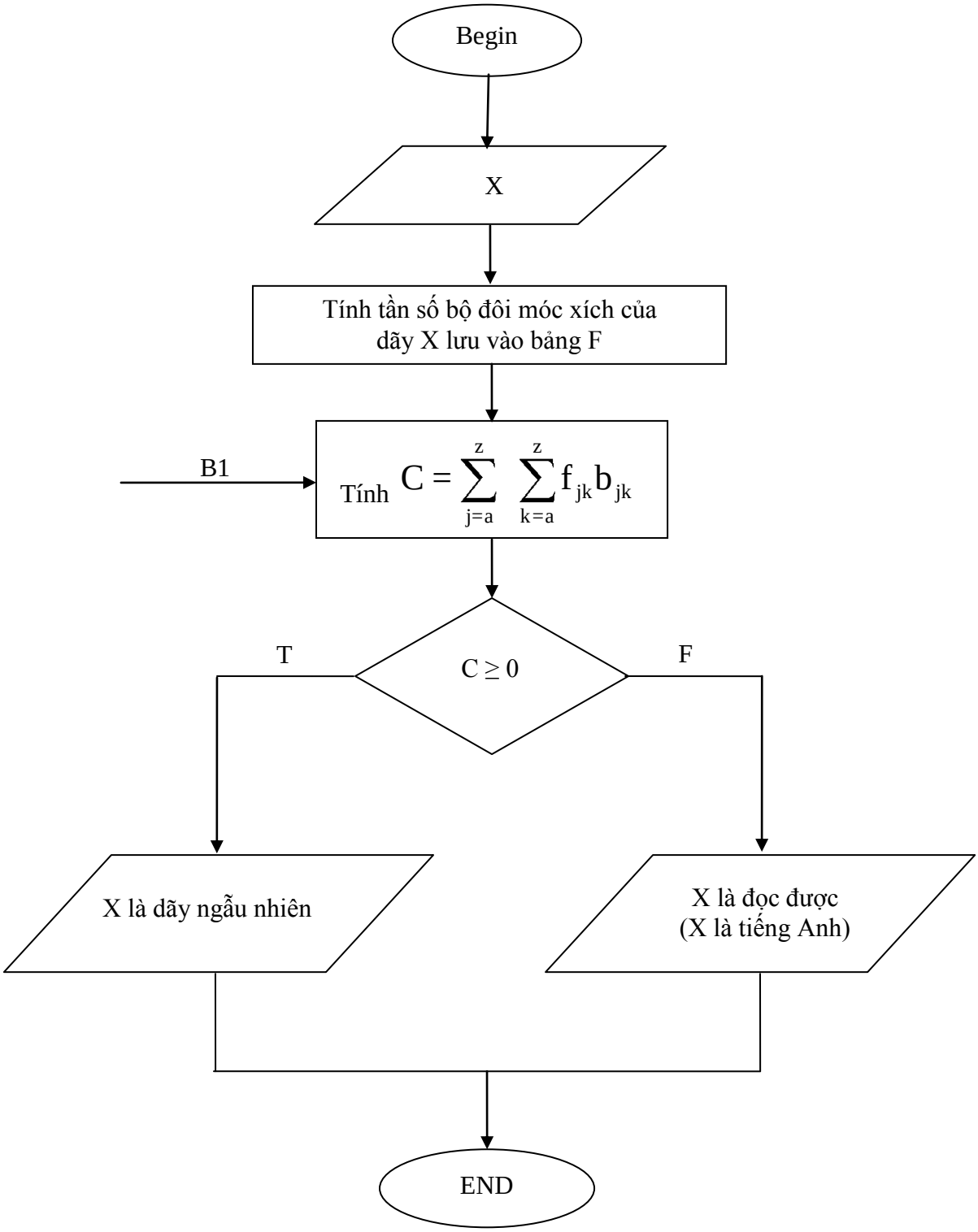
và $f_{ij} = \sum_{t=1}^{N-1} f_{t,t+1}(i,j)$ i,j = a,b,... ,z

Step 2: Tính Tr(F.B) bằng công thức:

$$C = \sum_{j=a}^{z} \sum_{k=a}^{z} f_{jk} b_{jk}$$

Nếu $C \geq 0$ thì thuật toán dừng và kết luận X thuộc lớp dãy ngẫu nhiên.

Nếu $C < 0$ thì thuật toán kết thúc và thông báo X thuộc bản rõ tiếng Anh.



Hình 3.1 Sơ đồ khối của thuật toán

3.2.3 Một số ví dụ

Ví dụ 1: Ta kiểm tra mẫu văn bản:

Cho X = phooi irsia ectoi ueeso oeefp hfspa psoat ltet trpb vtqiu igdsn eknrh e.

Vậy với thuật toán thì nó nhận ra thế nào. Quá trình thực hiện như sau:

Step 1: Tính tần số bộ đôi móc xích, được bảng sau (ký hiệu bảng F)

	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					1											1				1						
B																						1				
C																				1						
D																			1							
E			1		2	1					1								1	1						
F																1			1							
G				1																						
H					1	1									1											
I	1						1		1									1			2					
J																										
K														1												
L					1																					
M																										
N					1													1								
O	1				1				2						2											
P	1	1						2											1							
Q									1																	
R								1								1			1							
S									1					1	2	1										
T												1			1		1	1		2						
U					1				1																	
V																				1						
W																										
X																										
Y																										
Z																										

Sử dụng bảng B1: tính $C = \sum_{j=a}^{z} \sum_{k=a}^{z} f_{jk} b_{jk}$

Dòng	Các phép tính tương ứng trên dòng của bảng B1 và F
A	1.6 + 1.0 + 1.(-7) +
B	1.11 +
C	1.(-3) +
D	1.(-1) +
E	1.(-5) + 2.(-5) + 1.(-4) + 1.(-8) + 1.(-5) +
F	1.11 + 1.1 +
G	1.6 +
H	1.(-9) + 1.3 + 1.(-4) +
I	1.(-4) + 1.11 + 1.(-4) +
K	1.6 +
L	1.(-5) +
N	1.(-5) + 1.2 +
O	1.2 + 1.3 + 2.0 + 2.0
P	1.(-1) + 1.8 + 2.4 + 1.0 +
Q	1.11 +
R	1.1 + 1.4 + 1.(-3) +
S	1.(-6) + 1.0 + 2.(-5) + 1.2
T	1.(-6) + 1.11 + 1.(-4) + 2.(-5) +
U	1.1 + 1.1 +
V	1.11 = 26

Vậy C =26 > 0. Suy ra X là văn bản không hiểu được.

Ví dụ 2: Cho văn bản X = Edit windows are where you type in and edit your Turbo Pascal code. You can also do the following in an Edit window.

Tính tần số bộ đôi móc xích, được bảng sau (ký hiệu bảng F)

	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												2		3				1	1							
B															1											
C	2														1											
D					2				3						3											
E				3		1			1									1					1		2	
F															1											
G									1																	
H					2																					
I														5						3						
J																										
K																										
L			1									1			1				1							
M																										
N	3			3	1		1																			
O				2								1				1				1	3		3			
P	1				1																					
Q																										
R		1			2															1						
S	1		1												1											
T									1												1		2		2	
U			1															2		1						
V																										
W									1	3									1							
X																										
Y																3	1									
Z																										

Sử dụng bảng B1: tính $C = \sum_{j=a}^{z} \sum_{k=a}^{z} f_{jk} b_{jk}$

Dòng	Các phép tính tương ứng trên dòng của bảng B1 và F
A	2.(-5) + 3.(-8) + 1.(-6) + 1.(-6) +
B	1.0+
C	2.(-3) + 1.(-4) +
D	2.(-5) + 3.(-5) + 3.(-1) +
E	3.(-7) + 1.(-4) + 1.(-4) + 1.(-8) + 1.(-3) + 2.3 +
F	1.(-4) +
G	1.(-1) +
H	2.(-9) +
I	5.(-9) + 3.(-7) +
L	1.3 + 1.(-4) + 1.(-2) + 1.(-1) +
N	3.(-4) + 3.(-7) + 1.(-2) + 1.(-1) +
O	2.(-1) + 1.(-1) + 1.4 + 1.(-4) + 3.(-5) + 3.(-2) +
P	1.(-1) + 1.(-3) +
R	1.(-1) + 2.(-8) + 1.(-5) +
S	1.(-5) + 1.(-1) + 1.(-5) +
T	1.(-10) + 1.(-2) + 2.(-3) + 2.(-2) +
U	1.(1) + 2(-4) + 1.(-3) +
W	1.(-3) + 31.(-3) + 1.4 +
Y	3(-1) + 1.1 = -357

Vậy C = -357 < 0. Suy ra X là văn bản Tiếng Anh.

CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM

4.1 Kết quả đạt được

Một chương trình phần mềm trên ngôn ngữ C++ đã được xây dựng nhằm thử nghiệm các phép kiểm định đã nêu trên.

Các mẫu thử được lựa chọn là tài liệu Tiếng Anh thuộc các lĩnh vực: Chính trị, kinh tế, văn học, địa lý, quân sự, thể thao, ngoại giao, lịch sử, y tế, giáo dục, pháp luật; với độ dài khác nhau

Độ chính xác trong các trường hợp lấy độ dài kiểm tra khác nhau

Độ dài (ký tự)	Tiếng Anh			Dãy ngẫu nhiên		
	T.Số	Đúng	Tỷ lệ	T.Số	Đúng	Tỷ Lệ
50	150	148	98,67%	15	14	93%
60	150	149	99,33%	15	15	100%
70	150	150	100%	15	15	100%
80	150	150	100%	15	15	100%

4.2 Mã nguồn của chương trình

4.2.1 Thủ tục tính tần số bộ đôi với độ dài k

```
int TTSBD(int A[26][26], char *Fname, int k)
{
  FILE *F; char i,j;
  for(i=0;i<26;i++)
        for(j=0;j<26;j++)
              A[i][j]=0;
   if((F=fopen(Fname,"rt"))==NULL) return 0;
   int n=0;
  while (!feof(F)&&n<k)
        {
         j=fgetc(F);
         if(('A'<=j)&&(j<='Z'))
              {
                (A[i-65][j-65])++;
                i=j;n++;
              }
         if(('a'<=j)&&(j<='z'))
              {
                j=j-32;    (A[i-65][j-65])++;
                i=j; n++;
              }
        }
        fclose(F);
        return 1;
}
```

4.2.2 Hàm tính tổng của 2 ma trận

```
int Mul_A(int A[26][26], int B[26][26])
{
        int k=0;
        for(int i=0;i<26;i++)
            for(int j=0;j<26;j++)
                k+=A[i][j]*B[i][j];
        return k;
}
```

4.2.3 Hàm nhận biết ngôn ngữ

```
void thuattoan(int A[26][26], int B[26][26])
{
        get_A(A,"Data\\ANHMAX.txt");// Lay gia tri bang B1 vao mang A
        TTSBD(B,nameFile,k) //Tinh tan so bo doi moc xich cua mau trong File
        if(Mul_A(A,B)<0) cout<<nameFile<<"ban ro Tieng Anh"<<endl;
        else if(Mul_A(A,B)>0) cout<<nameFile<<"la day ngau nhien"<<endl;
        else cout<<"Nhap them ky tu vao ban mau"<<nameFile<<"va quay lai tinh tiep";
}
```

KẾT LUẬN

Như vậy, trong báo cáo luận văn của mình, em tập trung tìm hiểu mấy vấn đề cơ bản sau đây:

- Tìm hiểu mạng Neural nhân tạo.
- Tìm hiểu sâu về mạng Perceptron đa lớp với luật học lan truyền ngược sai số.
- Tìm hiểu về kỹ thuật nhận dạng bản rõ tiếng Anh của ngôn ngữ tự nhiên.
- Cài đặt và lập trình thử nghiệm trên máy PC bằng ngôn ngữ C++.

Mặc dù em đã có nhiều cố gắng nhằm hoàn thành tốt nhất bản luận văn của mình, song để nắm vững lý thuyết về mạng Neural, đòi hỏi sinh viên phải có một trình độ nhất định nào đó về kiến thức toán học. Như vậy về lý thuyết em trình bày chưa được sâu. Còn về thực hành, em cố gắng lập chương trình máy tính phân biệt (nhận dạng) một mẫu tiếng Anh với một dãy giả ngẫu nhiên.

Sắp tới, nếu có điều kiện, em tiếp tục nghiên cứu giải bài toán nhận dạng ngôn ngữ tự nhiên thuộc lớp các ngôn ngữ Latinh. Cuối cùng, một lần nữa, em xin chân thành cảm ơn các thầy, cô thuộc khoa CNTT trường Đại học dân lập Hải Phòng đã truyền đạt những kiến thức, nhờ đó em đã hoàn thành được luận văn tốt nghiệp của mình.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo tiếng Việt

[1.] Đinh Mạnh Tường - Trí tuệ nhân tạo

Nhà xuất bản khoa học và kỹ thuật - Hà Nội 2002.

[2.] Hoàng Minh Tuấn - Hồ Văn Canh

Nghiên cứu phương pháp thám mã khối trên máy tính song song (Luận văn tiến sỹ) - Dùng hệ điều hành Linux (4/2008).

Học viện kỹ thuật quân sự, Bộ Quốc Phòng.

[3.] Hoàng Kiếm - Nguyễn Ngọc Kỷ... Nhận dạng: Các phương pháp và...

Nhà xuất bản thống kê 7/1992.

Tài liệu tham khảo tiếng Anh

[4.] Robert J. Schallkoff - Artificial Neural Netwoks

The McGraw - Hill Companies, Inc 1997.

[5.] DASPA. Neural Network Study, MA: MIT Lincoln Laboratory, 1988.