

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐỒ ÁN CUỐI KỲ
Nhận diện ký tự với mạng Neural nhân tạo

Giảng viên hướng dẫn: Thầy Nguyễn Thanh Sơn
Bộ môn: Cấu trúc dữ liệu và Giải thuật – IT003.KHTN
Sinh viên: Lê Thu Hà
MSSV: 21520800
Lớp: KHTN2021

TP. Hồ Chí Minh, ngày 01 tháng 06 năm 2022

MỤC LỤC

LỜI NÓI ĐẦU	2
CHƯƠNG 1: TỔNG QUAN VỀ MẠNG NEURAL NHÂN TẠO.....	3
1.1. Khái niệm mạng neural	3
1.1.1. Tìm hiểu về Neural.....	3
1.1.2. Mạng neural nhân tạo.....	5
1.2. Phân loại mạng neural nhân tạo.	5
1.2.1. Phân loại theo kiểu liên kết neural.	5
1.2.2. Một số loại mạng neural.....	6
1.3. Huấn luyện mạng neural.....	7
1.3.1. Phương pháp học.....	7
1.3.2. Thuật toán học của mạng neural nhiều lớp.	8
1.4. Thu thập dữ liệu cho mạng neural.....	11
1.5. Biểu diễn chi thức cho mạng neural.....	12
CHƯƠNG 2: ỨNG DỤNG MẠNG NEURAL NHẬN DẠNG KÍ TỰ	12
2.1. Phát biểu bài toán	12
2.2. Các bước giải quyết bài toán	13
2.2.1. Xây dựng mạng neural.	13
2.2.2. Xử lý dữ liệu (phân tích ảnh).	13
2.2.3. Huấn luyện mạng neural.	18
CHƯƠNG 3: CÀI ĐẶT CHƯƠNG TRÌNH THỬ NGHIỆM	19

LỜI NÓI ĐẦU

Ứng dụng mạng neural trong nhận dạng kí tự.

Nội dung báo cáo gồm những chương sau:

Chương 1: Tổng quan về mạng neural nhân tạo.

Chương 2: Ứng dụng mạng neural nhận dạng kí tự.

Chương 3: Cài đặt chương trình thử nghiệm.

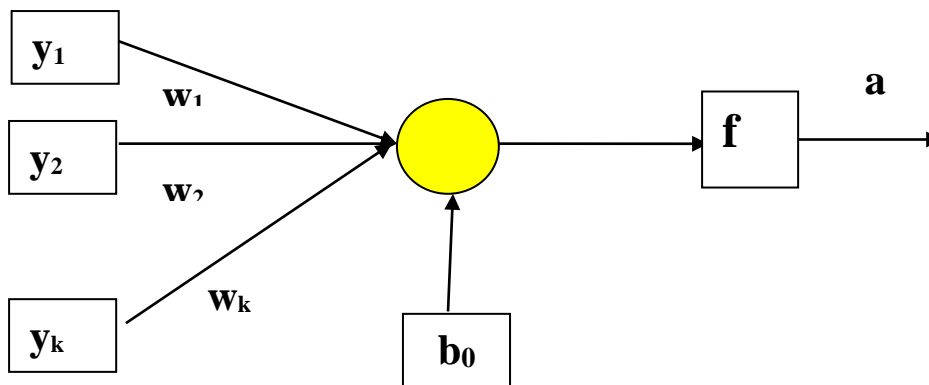
CHƯƠNG 1:

TỔNG QUAN VỀ MẠNG NEURAL NHÂN TẠO

1.1. Khái niệm mạng neural .

1.1.1. Tìm hiểu về Neural.

Neural nhân tạo là một đơn vị tính toán có nhiều đầu vào và một đầu ra, mỗi đầu vào đến từ một liên kết. Đặc trưng của neural là một hàm kích hoạt phi tuyến chuyển đổi 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 của mạng neural.



Đầu vào

Đầu ra $a=f(\sum w'y+b)$

Hình1.1: Mô hình một neural nhân tạo

Một neural được cấu tạo gồm các thành phần chính :

- ✓ Liên kết neural.
- ✓ Bộ cộng .
- ✓ Hàm kích hoạt.

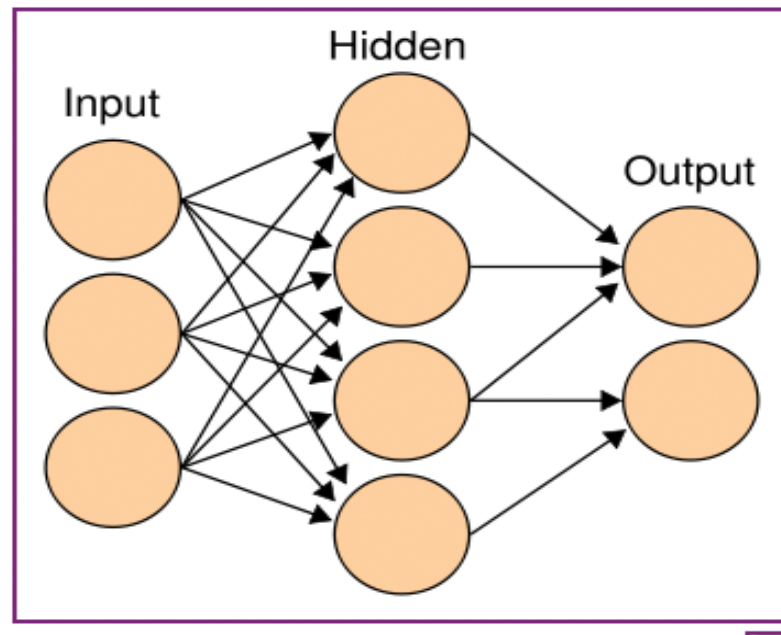
Bảng 1.1 : Một số hàm kích hoạt cơ bản trong mạng neural:

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
logsig	$a = \frac{1}{1 + e^{-n}}$

1.1.2. 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.

Mô hình mạng neural.



Hình 1.2: 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).
- ✓ Lớp xuất (output).

1.2. Phân loại mạng neural nhân tạo.

1.2.1. Phân loại theo kiểu liên kết neural.

Ta có:

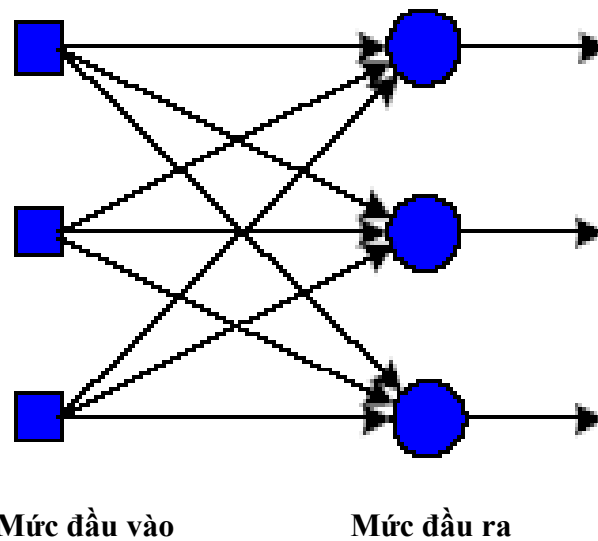
- ✓ Mạng neural dẫn tiến.
- ✓ Mạng neural qui hồi.

1.2.2. Một số loại mạng neural.

1.2.2.1. Mạng dẫn tiến .

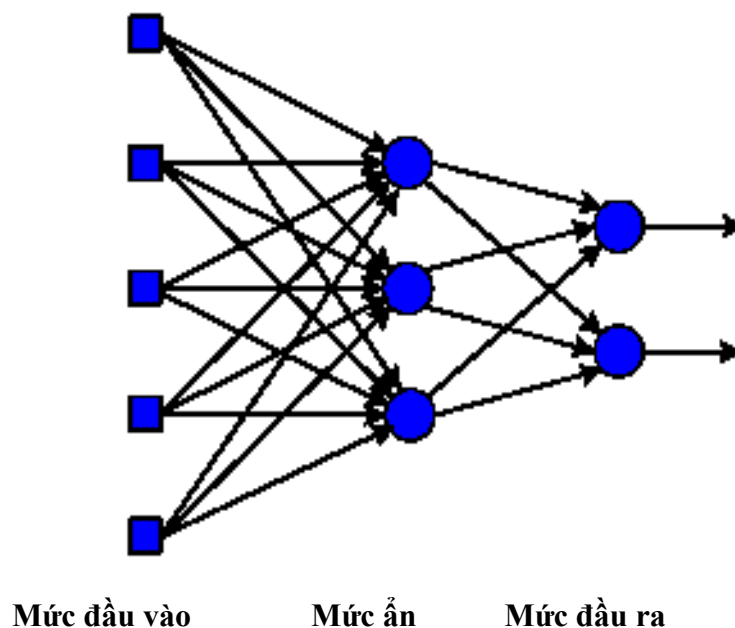
Có thể nói mạng neural dẫn tiến là một kiểu mạng đơn giản trong việc sắp đặt mạng. Trong mạng này thông tin chỉ truyền trên một hướng duy nhất. Từ lớp đầu vào xuyên qua lớp ẩn (nếu có) và kết thúc tại lớp đầu ra. Không có chu trình hoặc vòng trong mạng.

a. Các mạng dẫn tiến đơn mức.



Hình 1.3: Mạng tiến với một mức neural

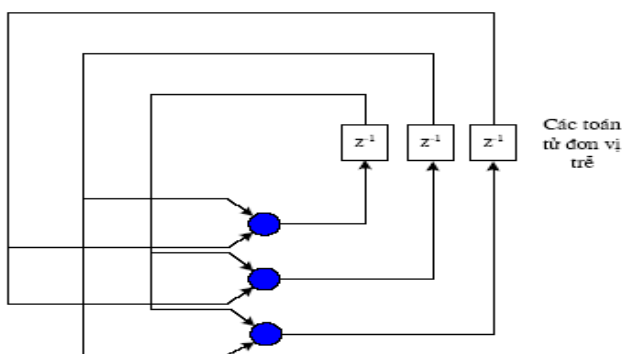
b. Các mạng dẫn tiến đa mức.



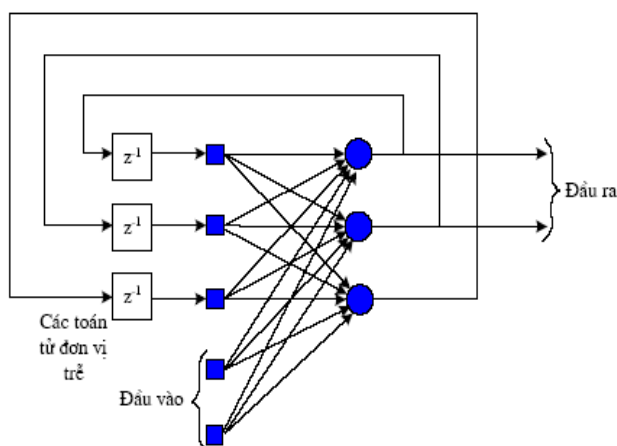
Hình 1.4: Mạng tiến kết nối đầy đủ với một mức ẩn và một mức đầu ra

1.2.2.2. Mạng quy hồi .

Trái với mạng neural dẫn tiến , mạng neural 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 1.5: Mạng hồi quy không có neural ẩn và không có vòng lặp tự phản hồi



Hình 1.6: Mạng hồi quy có các neural ẩn

1.3. Huấn luyện mạng neural.

1.3.1. 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. Có 3 kiểu học chính:

- ✓ Học có giám sát (có mẫu).
- ✓ Học không giám sát.

- ✓ Học tăng cường.

1.3.2. 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 đề:

- ✓ 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 : lớp nhập, lớp ẩn và lớp xuất.

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.

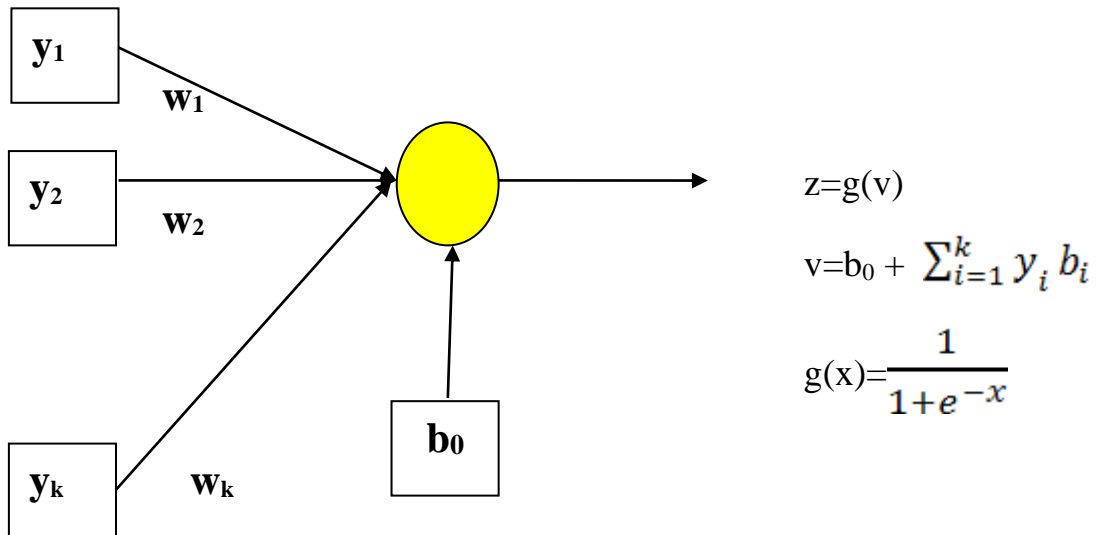
Bước 4: Tính đạo hàm riêng “ δE ” theo trọng số cho lớp xuất. sử dụng GRADIENT của hàm lỗi .

Bước 5: Tính đạo hàm riêng “ δE ” 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:

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

Đạo hàm riêng của hàm lỗi theo trọng số b:

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z} \frac{\partial z}{\partial v} \frac{\partial v}{\partial b} \quad (1.5)$$

Ta có:

$$\begin{aligned} \frac{\partial E}{\partial z} &= (z - t) \\ \frac{\partial z}{\partial v} &= \frac{\partial g}{\partial v} = z(1 - z) \\ \frac{\partial v}{\partial b_1} &= \begin{cases} 1, & i = 0 \\ y_i, & i > 0 \end{cases} \end{aligned}$$

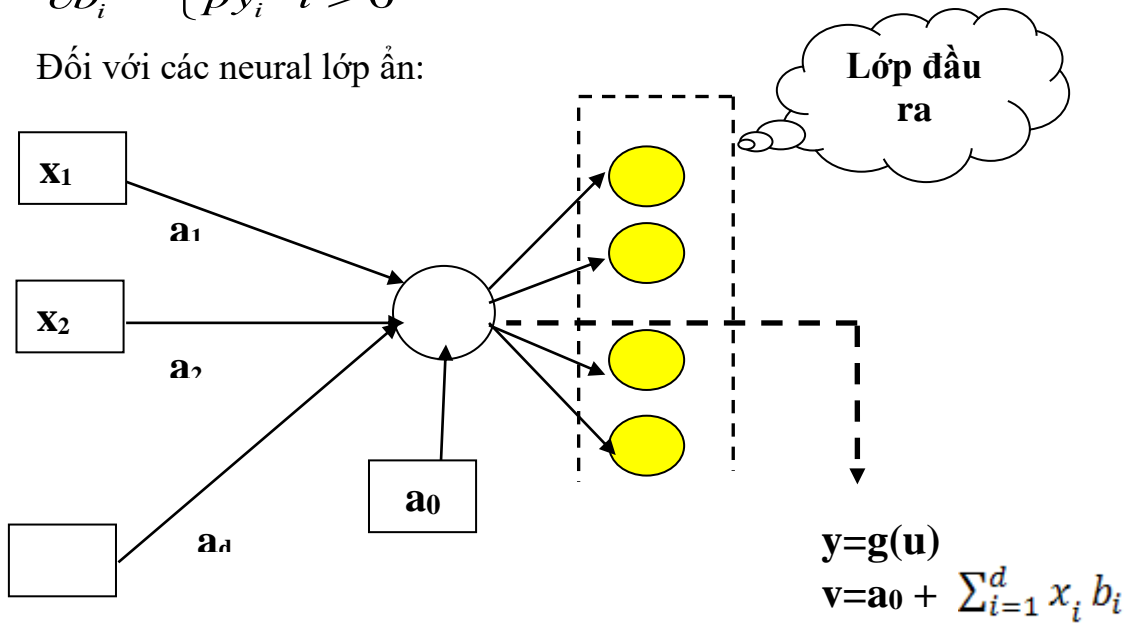
Đặt:

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

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:

$$\begin{aligned}
\frac{\partial E}{\partial y} &= \sum_{i=1}^K \frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial v_i} \frac{\partial v_i}{\partial y} \\
\frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial v_i} &= (z_i - t_i) z_i (1 - z_i) = p_i \\
\frac{\partial v_i}{\partial y} &= b_i \\
\frac{\partial y}{\partial u} &= y(1 - y) \\
\frac{\partial u}{\partial a_j} &= \begin{cases} 1, & j = 0 \\ c_j, & j > 0 \end{cases}
\end{aligned}$$

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:

$$w_i = w_{i-1} - \mu \frac{\partial E}{\partial w_{i-1}} \quad (1.9)$$

1.4. Thu thập dữ liệu cho mạng neural.

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

- Chọn những giá trị huấn luyện có tác dụng.

- 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.

1.5. Biểu diễn chi thức cho 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.

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.

CHƯƠNG 2: ỨNG DỤNG MẠNG NEURAL NHẬN DẠNG KÍ TỰ

2.1. Phát biểu bài toán

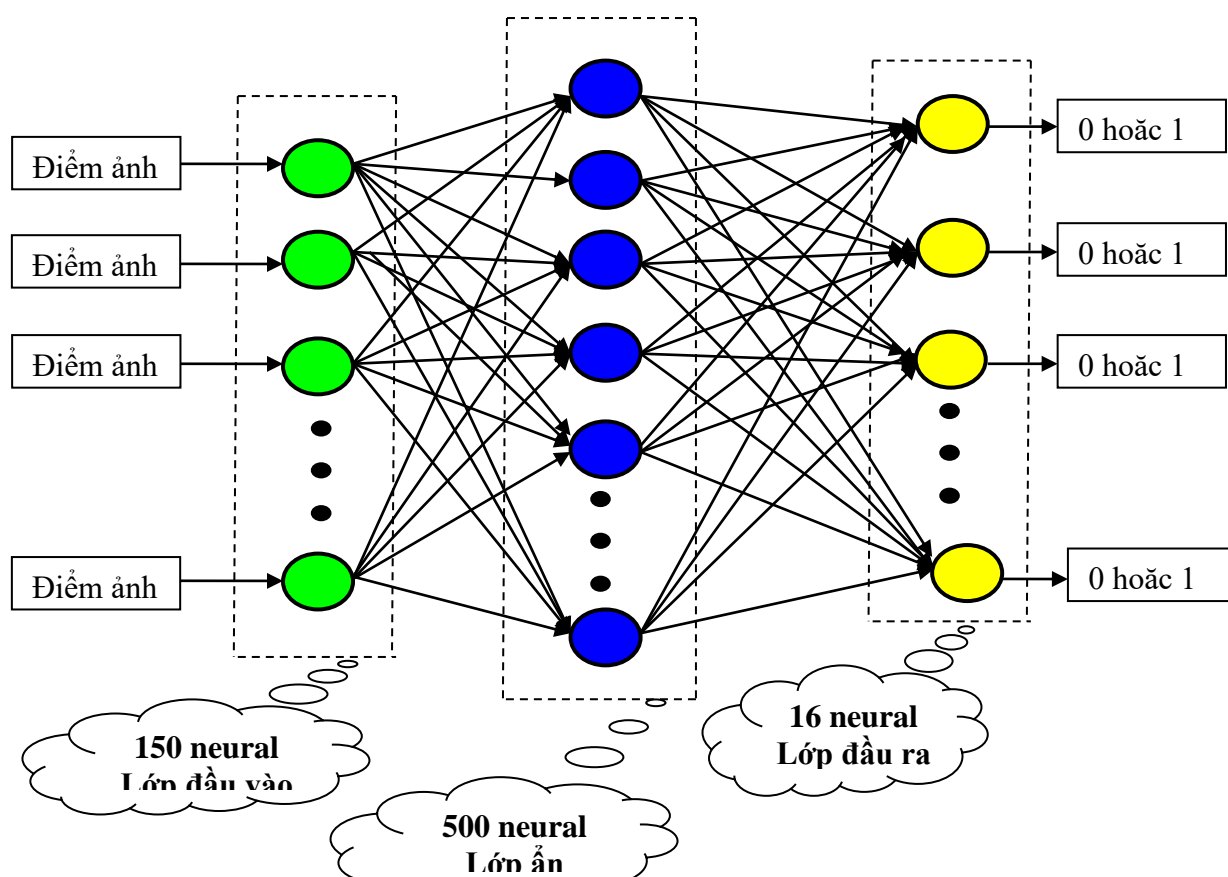
Cho bài toán nhận dạng kí tự với đầu vào là trang văn bản trên định dạng file ảnh (*.bmp, *.jpg, ...). Yêu cầu là xây dựng chương trình nhận dạng kí tự văn bản trên file ảnh. Sử dụng công cụ là mạng nơron và đầu ra là tập hợp các kí tự lưu trữ dưới dạng văn bản.

Giả thiết giai đoạn tiền xử lý đã được thực hiện : ảnh đã được phân ngưỡng, xoay ảnh, lọc nhiễu... Đề tài này chỉ đi sâu nghiên cứu quá trình nhận dạng kí tự của mạng neural.

2.2. Các bước giải quyết bài toán

2.2.1. Xây dựng mạng neural.

Chọn mạng đa tầng **Feedforward** với 3 lớp :



Hình 2.1: Sơ đồ mạng neural thiết kế

2.2.2. Xử lý dữ liệu (phân tích ảnh).

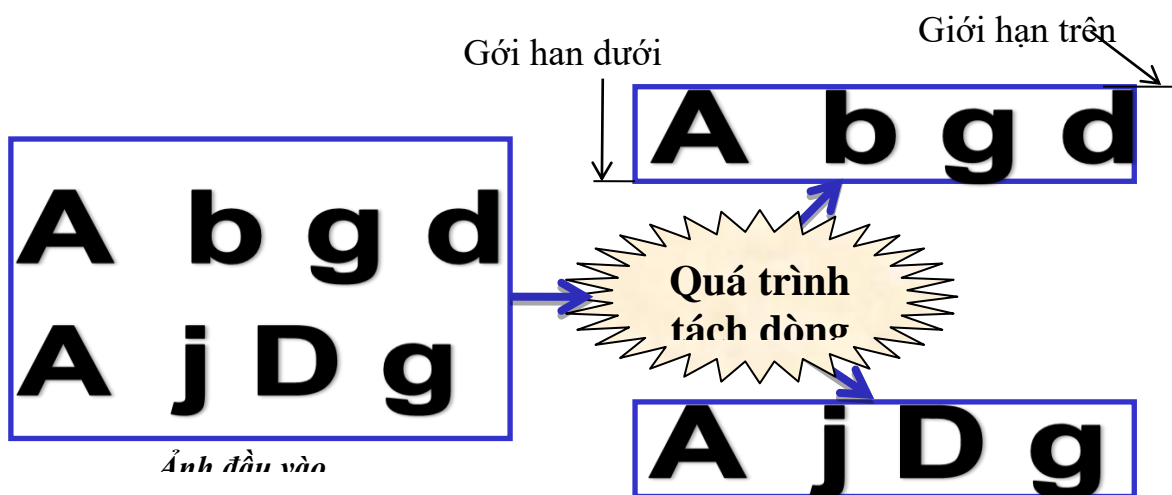
2.2.2.1. Tách dòng kí tự.

Thuật toán:

1. Bắt đầu từ điểm ảnh x, y đầu tiên (0,0), và đặt là dòng 0.
2. Quét đến hết chiều rộng của ảnh trên một giá trị y của ảnh.
3. Nếu là phát hiện điểm đen thì lưu giá trị y là đỉnh của hàng đầu tiên:

➤ Nếu không xét tiếp điểm tiếp theo.

- Nếu không thấy điểm đen nào thì tăng y và khởi động lại x để quét dòng ngang tiếp theo.
 - Bắt đầu từ đỉnh của hàng tìm thấy (0,giới hạn trên).
4. Quét đến hết chiều rộng của ảnh trên một giá trị y của ảnh.
- Nếu không phát hiện điểm đen nào thì đánh dấu giá trị y-1 là giới hạn dưới của hàng. Và tăng số dòng, xét dòng tiếp theo.
 - Nếu vẫn phát hiện thấy điểm đen tăng y và khởi động lại x xét hàng ngang tiếp theo.
5. Bắt đầu từ giới hạn dưới của hàng ngang tìm thấy sau cùng lặp lại từ bước 1 đến bước 4 tìm dòng kế tiếp.
6. Nếu y bằng chiều cao của ảnh thì dừng.



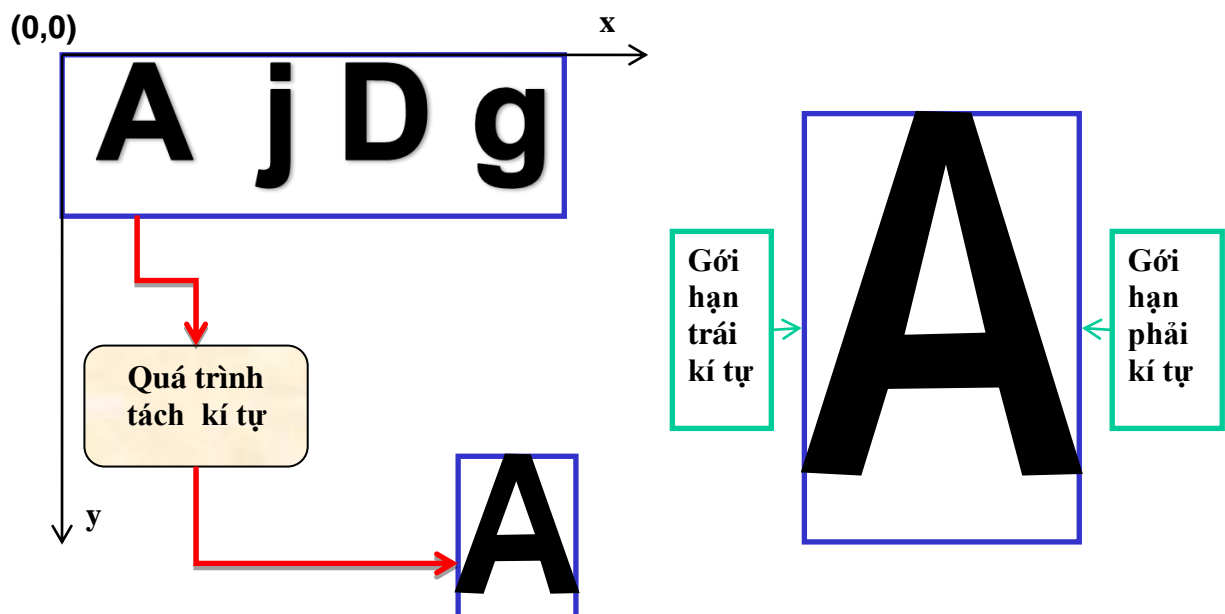
Hình 2.2: Quá trình tách dòng kí tự

2.2.2.2. Tách kí tự.

Thuật toán

1. Bắt đầu từ kí tự đầu tiên của hàng trên cùng với giá trị x đầu tiên.
2. Quét hết chiều rộng với một giá trị y.
 - Nếu phát hiện điểm đen đánh dấu y như là đỉnh của hàng đầu tiên.
 - Nếu không xét điểm tiếp theo.

3. Bắt đầu từ giới hạn trên của kí tự phát hiện được và giá trị x đầu tiên.(0,giới hạn trên kí tự) .
4. Quét đến giới hạn dưới của dòng, giữ nguyên x.
 - Nếu phát hiện điểm đen đánh dấu x là phía trái của kí tự.
 - Nếu không xét điểm tiếp theo.
 - Nếu không thấy điểm đen nào tăng x và khởi động lại y để xét đường thẳng đứng tiếp theo.
5. Bắt đầu từ phía trái của kí tự tìm thấy và đỉnh của dòng hiện thời, (giới hạn trái kí tự, giới hạn trên dòng).
6. Quét hết chiều rộng của ảnh. Trên cùng một giá trị x.
 - Nếu không có điểm đen nào thì đánh dấu x-1 là bên phải của kí tự.
 - Nếu phát hiện điểm đen tăng x và khởi động lại y để xét đường thẳng đứng tiếp theo.

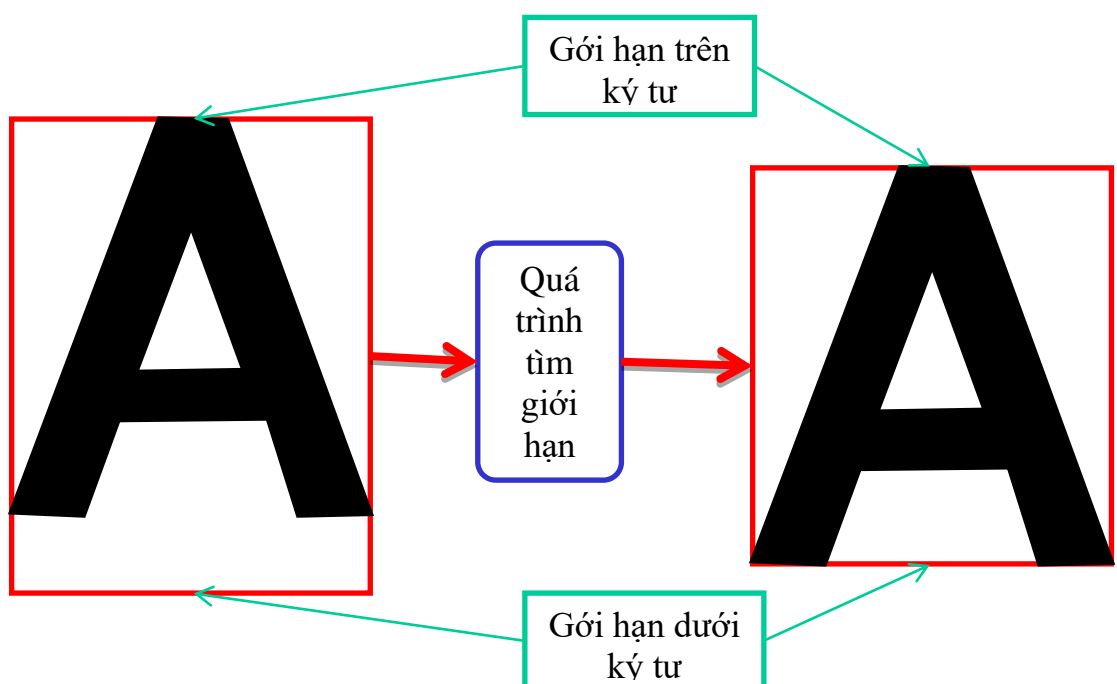


Hình 2.3: Quá trình tách kí tự

2.2.2.3. Tìm giới hạn kí tự.

Thuật toán

1. Bắt đầu từ đỉnh của dòng hiện thời và bên trái của kí tự.
2. Quét đến bên phải của kí tự. cùng một giá trị y.
 - Nếu phát hiện điểm đen thì đánh dấu y và thay đổi lại giới hạn trên.
 - Nếu không xét điểm tiếp theo.
 - Nếu không tìm thấy điểm đen nào tăng y và khởi động lại x, xét đường thẳng ngang tiếp theo.
3. Bắt đầu từ giới hạn dưới của dòng và bên trái của kí tự.
4. Quét tới bên phải của kí tự trên một giá trị y.
 - ✓ Nếu phát hiện điểm đen, đánh dấu y là giới hạn dưới của kí tự.
 - ✓ Nếu không phát hiện điểm đen giảm y và khởi động lại x xét đường thẳng ngang tiếp theo.



Hình 2.4: Quá trình tìm giới hạn kí tự

2.2.2.4. Ánh xạ vào ma trận

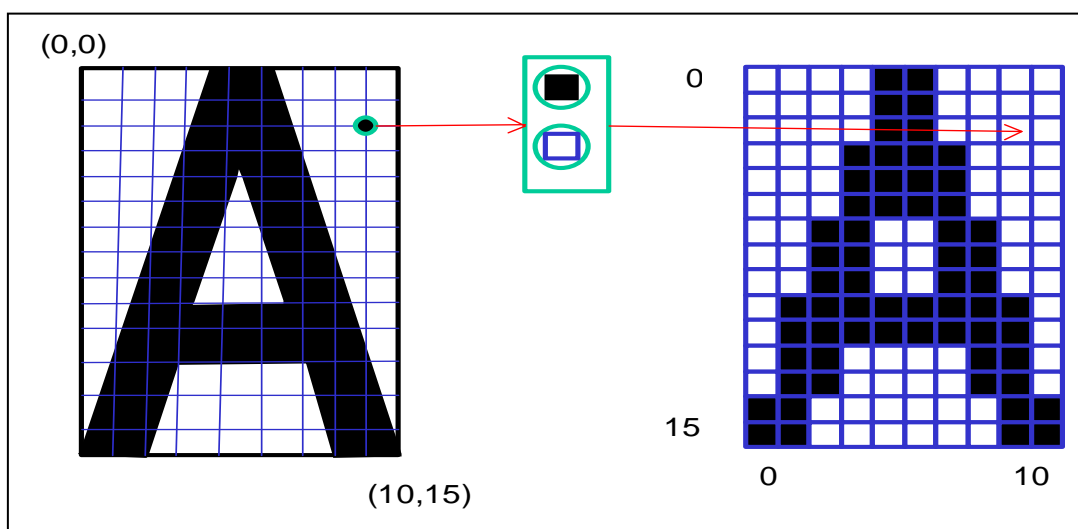
Thuật toán:

1. Đối với chiều rộng:

- Khởi tạo với 10 phần tử tương ứng.
- Ánh xạ điểm đầu $(0,y)$ và điểm cuối (C_rong,y) của ảnh kí tự tương ứng với giá trị đầu $(0,y)$ và giá trị cuối $(10,y)$ của ma trận.
- Chia nhỏ chiều rộng thành 10 giá trị tương ứng.

2. Đối với chiều cao:

- Khởi tạo với 15 phần tử tương ứng.
- Ánh xạ điểm đầu $(x,0)$ và điểm cuối (x,C_cao) của ảnh kí tự tương ứng với giá trị đầu $(x,0)$ và giá trị cuối $(x,15)$ của ma trận.
- Chia nhỏ chiều cao thành 15 giá trị tương ứng.



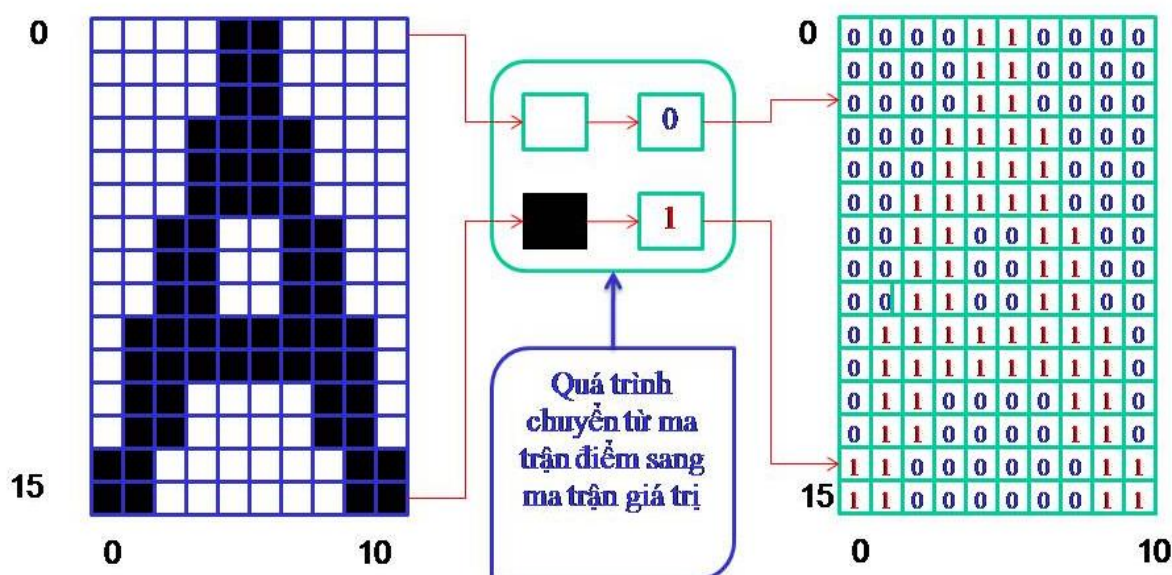
Hình 2.5: Quá trình chia lưới kí tự

Để đưa giá trị vào mạng neural ta cần chuyển ma trận điểm ảnh sang ma trận giá trị.

Thuật toán:

1. Bắt đầu từ phần tử (0,0).
2. Tăng x giữ nguyên giá trị y cho tới khi bằng chiều rộng của ma trận.
 - Ánh xạ mỗi phần tử tới một phần tử của mảng tuyến tính.
 - ✓ Nếu là điểm đen thì nhận giá trị bằng 1.
 - ✓ Ngược lại nhận giá trị bằng 0.
 - Nếu x = chiều rộng thì khởi động lại x và tăng y.

Lặp lại cho tới khi $(x,y)=(C_Rong, C_Cao)$.



Hình 2.6: Quá trình ánh xạ từ ma trận điểm sang ma trận giá trị.

2.2.3. Nhận dạng ảnh kí tự.

Thuật toán:

1. Nạp file ảnh.
2. Phân tích ảnh cho các dòng kí tự.
3. Với mỗi dòng tách các kí tự liên tiếp.

- Phân tích và xử lý ảnh kí tự cho việc ánh xạ vào một vector đầu vào.
- Đưa giá trị vecto đầu vào cho mạng neural và tính toán giá trị đầu ra.
- Chuyển đổi mã Unicode đầu ra từ nhị phân tới kí tự tương ứng và trả ra dưới dạng textbox.

CHƯƠNG 3: CÀI ĐẶT CHƯƠNG TRÌNH

3.1. Môi trường thử nghiệm.

Phần cứng : Một máy tính Lenovo yoga460.

Card đồ họa Nvidia GeForce 940M 2GB

Ram 8 GB .

Link github: <https://github.com/21520800/character-recognizing>

TÀI LIỆU THAM KHẢO

- [1.] *MathNeuralNetworks*. Ben Krose, faculty of Mathematics and computer science, university of Amsterdam. And Patrick van der smagt, institute of robotics and system dynamics German aerospace Research establishment.
- [2.] *Artificial Neural Networks and Information theory*, colin Fyfe, department of computing and information system, the university of Paisley.
- [3.] *A Growth Algorithm for Neural Networks Decision Trees*. Mostefa golea and Mario Marchand, Department of physics, university of Ottawa Canada.
- [4.] *Artificial neural network* From Wikipedia, the free encyclopedia.
- [5.] *Neural network* From Wikipedia, the free encyclopedia.
- [6.] <http://en.wikipedia.org/wiki/Non-linear>
- [7.] <http://www.codeproject.com/>

