

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ THỊ THU HẰNG

NGHIÊN CỨU VỀ MẠNG NEURAL TÍCH
CHẬP VÀ ỨNG DỤNG CHO BÀI TOÁN
NHẬN DẠNG BIẾN SỐ XE

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

HÀ NỘI, 2016

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ THỊ THU HẰNG

**NGHIÊN CỨU VỀ MẠNG NEURAL TÍCH
CHẬP VÀ ỨNG DỤNG CHO BÀI TOÁN
NHẬN DẠNG BIẾN SỐ XE**

Ngành : Công nghệ thông tin
Chuyên ngành : Kỹ thuật phần mềm
Mã số : 60480103

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Người hướng dẫn khoa học: TS. Nguyễn Văn Vinh

HÀ NỘI, 2016

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn của Tiến sĩ Nguyễn Văn Vinh. Các kết quả đạt được trong luận văn là sản phẩm của riêng cá nhân, không sao chép của người khác. Nội dung của luận văn có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn sách, tạp chí được liệt kê trong danh mục các tài liệu tham khảo.

Tác giả luận văn

Lê Thị Thu Hằng

MỤC LỤC

| | |
|--|----|
| LỜI MỞ ĐẦU | 6 |
| LỜI CẢM ƠN | 7 |
| CHƯƠNG 1: MẠNG NEURAL VÀ MẠNG NEURAL TÍCH CHẬP | 8 |
| 1. Mạng neural và mạng neural lan truyền ngược | 8 |
| 1.1. Giới thiệu về mạng Neural..... | 8 |
| 1.2. Một số kiểu mạng Neural..... | 10 |
| 1.3. Mạng Neural lan truyền ngược MLP..... | 12 |
| 2. Mạng neural tích chập | 21 |
| 2.1. Định nghĩa mạng neural tích chập | 21 |
| 2.2. Convolution (tích chập) | 22 |
| 2.3. Mô hình mạng neural tích chập | 24 |
| 2.4. Xây dựng mạng neural tích chập | 26 |
| CHƯƠNG 2: TỔNG QUAN VỀ NHẬN DẠNG BIỂN SỐ XE..... | 39 |
| 2.1 Khái niệm về hệ thống nhận dạng biển số xe. | 39 |
| 2.1.1 Khái niệm..... | 39 |
| 2.1.2 Lịch sử và phát triển. | 39 |
| 2.1.3. Cách thức hoạt động của hệ thống nhận dạng biển số xe..... | 40 |
| 2.1.4. Phân loại các ứng dụng nhận dạng biển số xe. | 40 |
| 2.1.5. Ứng dụng thực tiễn tại Việt Nam..... | 41 |
| 2.1.6. Phân loại biển số xe. | 42 |
| 2.2. Phương pháp nhận dạng biển số xe từ ảnh chụp của camera. | 44 |
| 2.2.1. Phương pháp chuyển đổi Hough..... | 44 |
| 2.2.2. Phương pháp hình thái học. | 45 |
| 2.3. Phương pháp nhận dạng ký tự trong biển số xe..... | 46 |
| 2.4. Phạm vi nghiên cứu và hướng giải quyết. | 47 |

| | |
|--|----|
| CHƯƠNG 3: ỨNG DỤNG MẠNG NEURAL TÍCH CHẬP TRONG NHẬN DẠNG KÝ TỰ | 49 |
| 3.1. Xây dựng mô hình mạng..... | 49 |
| 3.2. Kết quả nhận dạng ký tự viết tay | 52 |
| 3.3. Kết quả thực nghiệm nhận dạng biển số xe với mạng Neural tích chập..... | 54 |
| 3.4. Kết luận | 55 |
| 3.5. Hướng phát triển của bài toán:..... | 56 |
| DANH MỤC TÀI LIỆU THAM KHẢO | 58 |

LỜI MỞ ĐẦU

Deep Learning là một thuật toán dựa trên một số ý tưởng từ não bộ tới việc tiếp thu nhiều tầng biểu đạt, cả cụ thể lẫn trừu tượng, qua đó làm rõ nghĩa của các loại dữ liệu. **Deep Learning** được ứng dụng trong nhận diện hình ảnh, nhận diện giọng nói, xử lý ngôn ngữ tự nhiên

Hiện nay rất nhiều các bài toán nhận dạng sử dụng deep learning để giải quyết do deep learning có thể giải quyết các bài toán với số lượng lớn, kích thước đầu vào lớn với hiệu năng cũng như độ chính xác vượt trội so với các phương pháp phân lớp truyền thống

Những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Trong luận văn cao học này, em đi vào nghiên cứu về mạng neural cũng như mạng Convolution (tích chập) cũng như ý tưởng của mô hình CNNs trong phân lớp ảnh (Image Classification), và áp dụng trong việc xây dựng hệ thống nhận dạng biển số xe tự động

Nội dung bài báo cáo bao gồm 3 chương.

- Chương 1: Mạng neural và mạng neural tích chập.
- Chương 2: Tổng quan về nhận dạng biển số xe.
- Chương 3: Áp dụng mạng neural tích chập trong nhận dạng ký tự

LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy, các cô khoa Công nghệ Thông tin-Trường Đại học Công nghệ - Đại học quốc gia Hà Nội đã tận tình dạy dỗ, truyền đạt cho em nhiều kiến thức quý báu.

Em xin tỏ lòng biết ơn sâu sắc đến thầy Ts Nguyễn Văn Vinh, người đã tận tình giúp đỡ và truyền đạt nhiều kinh nghiệm để đề tài có thể được thực hiện và hoàn thành.

Xin chân thành cảm ơn các bạn trong khoa Công Nghệ Thông Tin, Trường ĐH Công nghệ đã giúp đỡ, động viên tôi rất nhiều trong quá trình thực hiện đề tài.

Em xin chân thành cảm ơn !

Hà nội, tháng 5 năm 2016

Học viên

Lê Thị Thu Hằng

CHƯƠNG 1: MẠNG NEURAL VÀ MẠNG NEURAL TÍCH CHẬP

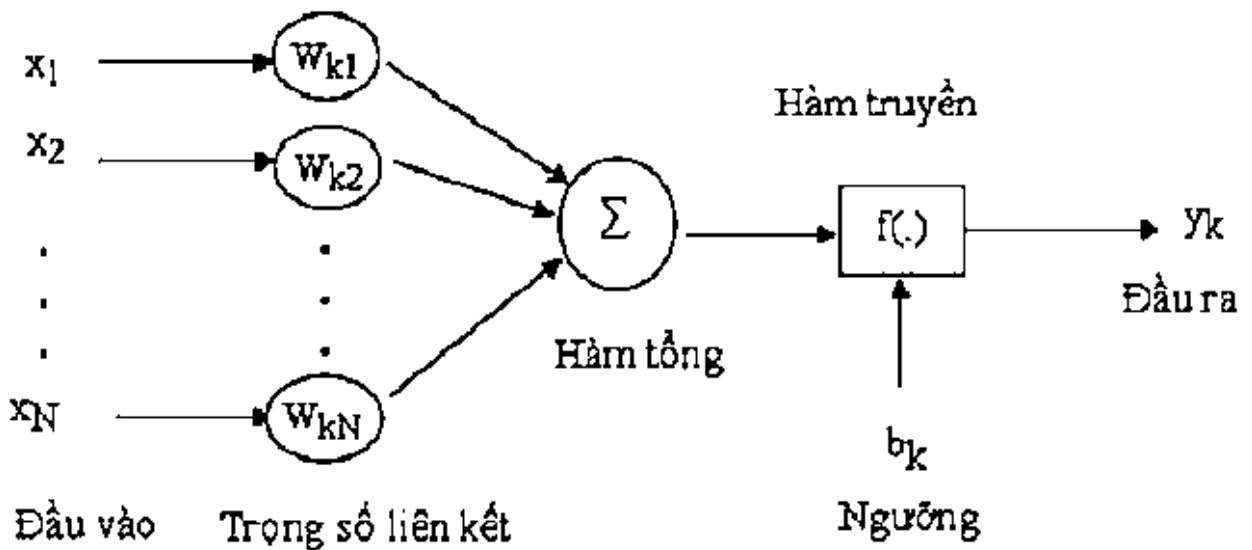
1. Mạng neural và mạng neural lan truyền ngược

1.1. Giới thiệu về mạng Neural

Định nghĩa: Mạng nơron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.

Cấu trúc neural nhân tạo:

Hình 1.1. Cấu tạo một Neural



Các thành phần cơ bản của một nơron nhân tạo bao gồm:

- Tập các đầu vào: Là các tín hiệu vào (input signals) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.
- Tập các liên kết: Mỗi liên kết được thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là w_{kj} . Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên.

nhien ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

- Bộ tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

- Ngưỡng (còn gọi là một độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.

- Hàm truyền (Transfer function): Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

- Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.

Xét về mặt toán học, cấu trúc của một nơron k , được mô tả bằng cặp biểu thức sau:

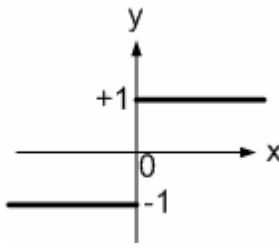
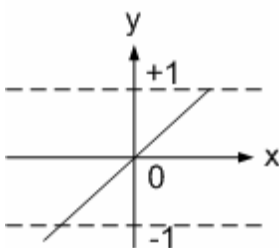
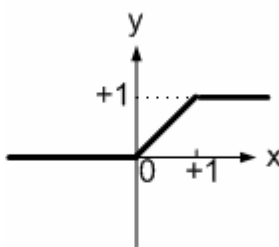
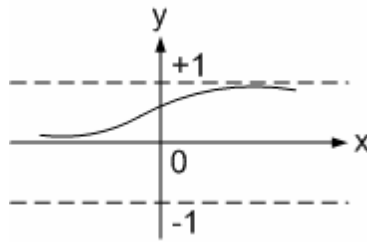
$$u_k = \sum_{j=1}^P w_{kj} x_j$$

và $y_k = f(u_k - b_k)$

Trong đó: x_1, x_2, \dots, x_p : là các tín hiệu vào; $(w_{k1}, w_{k2}, \dots, w_{kp})$ là các trọng số liên kết của nơron thứ k ; u_k là hàm tổng; b_k là một ngưỡng; f là hàm truyền và y_k là tín hiệu đầu ra của nơron.

Như vậy nơron nhân tạo nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả tới hàm truyền), và cho một tín hiệu đầu ra (là kết quả của hàm truyền).

Bảng 1.1. Một số hàm truyền thông dụng

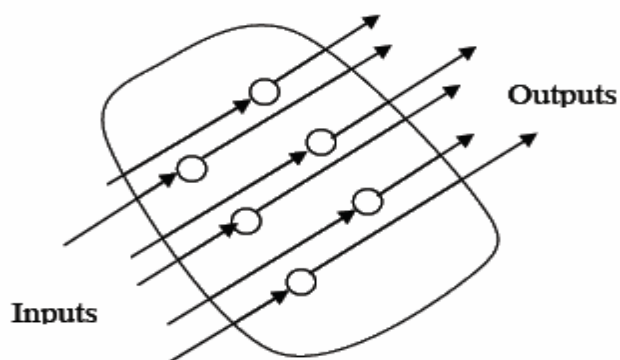
| Hàm truyền | Đồ thị | Định nghĩa |
|-----------------------------------|--|------------|
| Symmetrical Hard Limit (hardlims) |  | |
| Linear (purelin) |  | |
| Saturating Linear (satlin) |  | |
| Log-Sigmoid (logsig) |  | |

1.2. Một số kiểu mạng Neural

Cách thức kết nối các nơon trong mạng xác định kiến trúc (*topology*) của mạng. Các nơon trong mạng có thể kết nối đầy đủ (*fully connected*) tức là mỗi nơon đều được kết nối với tất cả các nơon khác, hoặc kết nối cục bộ (*partially connected*) chẳng hạn chỉ kết nối giữa các nơon trong các tầng khác nhau. Người ta chia ra hai loại kiến trúc mạng chính:

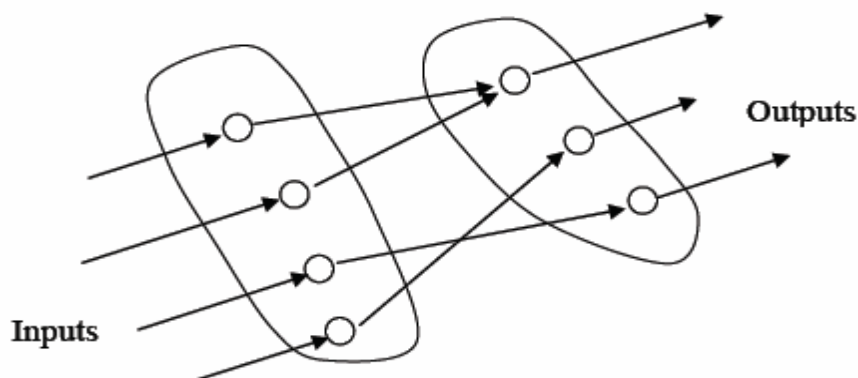
♦ Tự kết hợp (*autoassociative*): là mạng có các nơon đầu vào cũng là các nơon đầu ra. Mạng Hopfield là một kiểu mạng tự kết hợp.

Hình 1.2. Mạng tự kết hợp



♦ Kết hợp khác kiểu (*heteroassociative*): là mạng có tập nơron đầu vào và đầu ra riêng biệt. Perceptron, các mạng Perceptron nhiều tầng (MLP: MultiLayer Perceptron), mạng Kohonen, ... thuộc loại này.

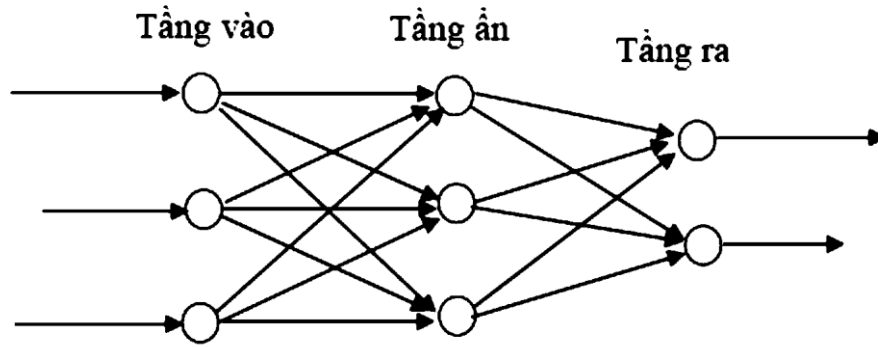
Hình 1.3. Mạng kết hợp khác kiểu



Ngoài ra tùy thuộc vào mạng có các kết nối ngược (*feedback connections*) từ các nơron đầu ra tới các nơron đầu vào hay không, người ta chia ra làm 2 loại kiến trúc mạng.

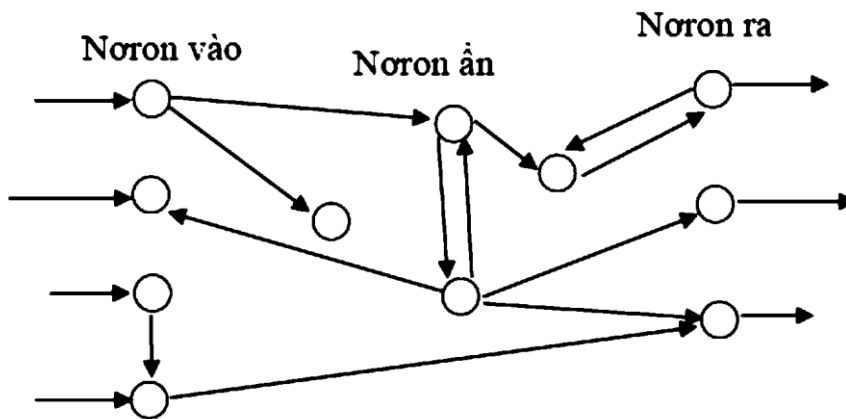
♦ Kiến trúc truyền thẳng (*feedforward architecture*): là kiểu kiến trúc mạng không có các kết nối ngược trở lại từ các nơron đầu ra về các nơron đầu vào; mạng không lưu lại các giá trị output trước và các trạng thái kích hoạt của nơron. Các mạng nơron truyền thẳng cho phép tín hiệu di chuyển theo một đường duy nhất; từ đầu vào tới đầu ra, đầu ra của một tầng bất kỳ sẽ không ảnh hưởng tới tầng đó. Các mạng kiểu Perceptron là mạng truyền thẳng.

Hình 1.4. Mạng truyền thẳng



♦ Kiến trúc phản hồi (*Feedback architecture*): là kiểu kiến trúc mạng có các kết nối từ nơon đầu ra tới nơon đầu vào. Mạng lưu lại các trạng thái trước đó, và trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào mà còn phụ thuộc vào các trạng thái trước đó của mạng. Mạng Hopfield thuộc loại này.

Hình 1.5. Mạng phản hồi

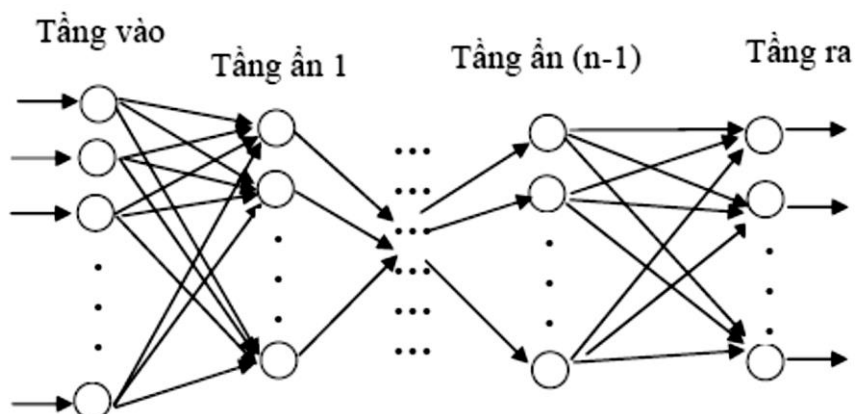


1.3. Mạng Neural lan truyền ngược MLP

a. Kiến trúc mạng MLP

Mô hình mạng nơon được sử dụng rộng rãi nhất là mô hình mạng nhiều tầng truyền thẳng (MLP: Multi Layer Perceptron). Một mạng MLP tổng quát là mạng có n ($n \geq 2$) tầng (thông thường tầng đầu vào không được tính đến): trong đó gồm một tầng đầu ra (tầng thứ n) và $(n-1)$ tầng ẩn.

Hình 1.6. Mạng neural lan truyền ngược MLP



Kiến trúc của một mạng MLP tổng quát có thể mô tả như sau:

- ◆ Đầu vào là các vector (x_1, x_2, \dots, x_p) trong không gian p chiều, đầu ra là các vector (y_1, y_2, \dots, y_q) trong không gian q chiều. Đối với các bài toán phân loại, p chính là kích thước của mẫu đầu vào, q chính là số lớp cần phân loại. Xét ví dụ trong bài toán nhận dạng chữ số: với mỗi mẫu ta lưu tọa độ (x, y) của 8 điểm trên chữ số đó, và nhiệm vụ của mạng là phân loại các mẫu này vào một trong 10 lớp tương ứng với 10 chữ số $0, 1, \dots, 9$. Khi đó p là kích thước mẫu và bằng $8 \times 2 = 16$; q là số lớp và bằng 10.

- ◆ Mỗi nơron thuộc tầng sau liên kết với tất cả các nơron thuộc tầng liền trước nó.

- ◆ Đầu ra của nơron tầng trước là đầu vào của nơron thuộc tầng liền sau nó.

Hoạt động của mạng MLP như sau: tại tầng đầu vào các nơron nhận tín hiệu vào xử lý (tính tổng trọng số, gửi tới hàm truyền) rồi cho ra kết quả (là kết quả của hàm truyền); kết quả này sẽ được truyền tới các nơron thuộc tầng ẩn thứ nhất; các nơron tại đây tiếp nhận như là tín hiệu đầu vào, xử lý và gửi kết quả đến tầng ẩn thứ 2;...; quá trình tiếp tục cho đến khi các nơron thuộc tầng ra cho kết quả.

Một số kết quả đã được chứng minh:

- ◆ Bất kì một hàm Boolean nào cũng có thể biểu diễn được bởi một mạng MLP 2 tầng trong đó các nơron sử dụng hàm truyền sigmoid.

♦ Tất cả các hàm liên tục đều có thể xấp xỉ bởi một mạng MLP 2 tầng sử dụng hàm truyền sigmoid cho các nơron tầng ẩn và hàm truyền tuyến tính cho các nơron tầng ra với sai số nhỏ tùy ý.

♦ Mọi hàm bất kỳ đều có thể xấp xỉ bởi một mạng MLP 3 tầng sử dụng hàm truyền sigmoid cho các nơron tầng ẩn và hàm truyền tuyến tính cho các nơron tầng ra.

b. Huấn luyện mạng MLP:

+ Các phương pháp học:

Khái niệm: Học là quá trình thay đổi hành vi của các vật theo một cách nào đó làm cho chúng có thể thực hiện tốt hơn trong tương lai.

Một mạng nơron được huấn luyện sao cho với một tập các vector đầu vào X , mạng có khả năng tạo ra tập các vector đầu ra mong muốn Y của nó. Tập X được sử dụng cho huấn luyện mạng được gọi là tập huấn luyện (*training set*). Các phần tử x thuộc X được gọi là các mẫu huấn luyện (*training example*). Quá trình huấn luyện bản chất là sự thay đổi các trọng số liên kết của mạng. Trong quá trình này, các trọng số của mạng sẽ hội tụ dần tới các giá trị sao cho với mỗi vector đầu vào x từ tập huấn luyện, mạng sẽ cho ra vector đầu ra y như mong muốn

Có ba phương pháp học phổ biến là học có giám sát (*supervised learning*), học không giám sát (*unsupervised learning*) và học tăng cường (*Reinforcement learning*):

♦ **Học có giám sát:** Là quá trình học có sự tham gia giám sát của một “thầy giáo”. Cũng giống như việc ta dạy một em nhỏ các chữ cái. Ta đưa ra một chữ “a” và bảo với em đó rằng đây là chữ “a”. Việc này được thực hiện trên tất cả các mẫu chữ cái. Sau đó khi kiểm tra ta sẽ đưa ra một chữ cái bất kỳ (có thể viết hơi khác đi) và hỏi em đó đây là chữ gì?

Như vậy với học có giám sát, số lớp cần phân loại đã được biết trước. Nhiệm vụ của thuật toán là phải xác định được một cách thức phân lớp sao cho với mỗi vector đầu vào sẽ được phân loại chính xác vào lớp của nó.

♦ **Học không giám sát:** Là việc học không cần có bất kỳ một sự giám sát nào.

Trong bài toán học không giám sát, tập dữ liệu huấn luyện được cho dưới dạng: $D = \{(x_1, x_2, \dots, x_N)\}$, với (x_1, x_2, \dots, x_N) là vector đặc trưng của mẫu huấn luyện. Nhiệm vụ của thuật toán là phải phân chia tập dữ liệu D thành các nhóm con, mỗi nhóm chứa các vector đầu vào có đặc trưng giống nhau.

Như vậy với học không giám sát, số lớp phân loại chưa được biết trước, và tùy theo tiêu chuẩn đánh giá *độ tương tự* giữa các mẫu mà ta có thể có các lớp phân loại khác nhau.

♦ **Học tăng cường:** đôi khi còn được gọi là học thưởng-phạt (*reward-penalty learning*), là sự tổ hợp của cả hai mô hình trên. Phương pháp này cụ thể như sau: với vector đầu vào, quan sát vector đầu ra do mạng tính được. Nếu kết quả được xem là “tốt” thì mạng sẽ được thưởng theo nghĩa tăng các trọng số kết nối lên; ngược lại mạng sẽ bị phạt, các trọng số kết nối không thích hợp sẽ được giảm xuống. Do đó học tăng cường là học theo nhà phê bình (*critic*), ngược với học có giám sát là học theo thầy giáo (*teacher*).

c. Học có giám sát trong các mạng nơron

Học có giám sát có thể được xem như việc xấp xỉ một ánh xạ: $X \rightarrow Y$, trong đó X là tập các vấn đề và Y là tập các lời giải tương ứng cho vấn đề đó. Các mẫu (x, y) với $x = (x_1, x_2, \dots, x_n) \in X$, $y = (y_1, y_2, \dots, y_m) \in Y$ được cho trước. Học có giám sát trong các mạng nơron thường được thực hiện theo các bước sau:

♦ **B1:** Xây dựng cấu trúc thích hợp cho mạng nơron, chẳng hạn có $(n + 1)$ nơron vào (n nơron cho biến vào và 1 nơron cho ngưỡng x_0), m nơron đầu ra, và khởi tạo các trọng số liên kết của mạng.

♦ **B2:** Đưa một vector x trong tập mẫu huấn luyện X vào mạng

♦ **B3:** Tính vector đầu ra o của mạng

♦ **B4:** So sánh vector đầu ra mong muốn y (là kết quả được cho trong tập huấn luyện) với vector đầu ra o do mạng tạo ra; nếu có thể thì đánh giá lỗi.

♦ **B5:** Hiệu chỉnh các trọng số liên kết theo một cách nào đó sao cho ở lần tiếp theo khi đưa vector x vào mạng, vector đầu ra o sẽ giống với y hơn.

♦ **B6:** Nếu cần, lặp lại các bước từ 2 đến 5 cho tới khi mạng đạt tới trạng thái hội tụ. Việc đánh giá lỗi có thể thực hiện theo nhiều cách, cách dùng nhiều

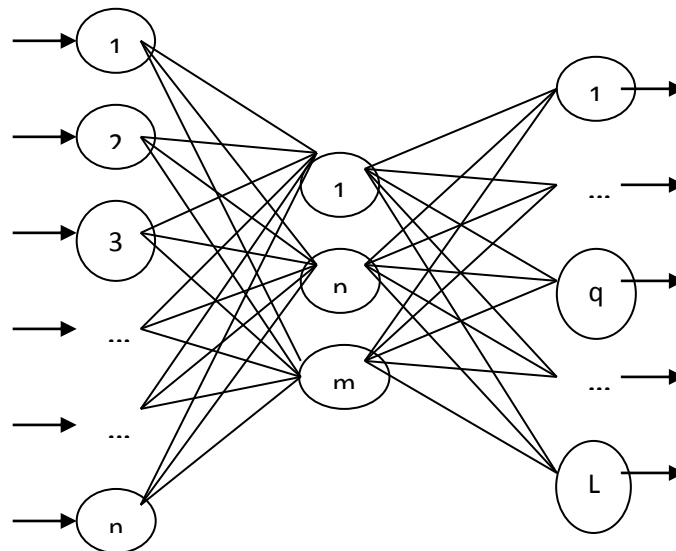
nhất là sử dụng lỗi tức thời: $Err = (\mathbf{o} - \mathbf{y})$, hoặc $Err = |\mathbf{o} - \mathbf{y}|$; lỗi trung bình bình phương (MSE: mean-square error): $Err = (\mathbf{o} - \mathbf{y})^2/2$;

Có hai loại lỗi trong đánh giá một mạng nơron. Thứ nhất, gọi là lỗi rõ ràng (*apparent error*), đánh giá khả năng xấp xỉ các mẫu huấn luyện của một mạng đã được huấn luyện. Thứ hai, gọi là lỗi kiểm tra (*test error*), đánh giá khả năng tổng quát hóa của một mạng đã được huấn luyện, tức khả năng phản ứng với các vector đầu vào mới. Để đánh giá lỗi kiểm tra chúng ta phải biết đầu ra mong muốn cho các mẫu kiểm tra.

Thuật toán tổng quát ở trên cho học có giám sát trong các mạng nơron có nhiều cài đặt khác nhau, sự khác nhau chủ yếu là cách các trọng số liên kết được thay đổi trong suốt thời gian học. Trong đó tiêu biểu nhất là thuật toán lan truyền ngược.

d. Mạng lan truyền ngược <Back Propagation Network>

1. Mô hình mạng: Mạng neural lan truyền ngược có mô hình như sau



Mạng có 3 lớp:

- Lớp vào (Input Layer) – số node vào là số thuộc tính của đối tượng cần phân lớp.
- Lớp ra (Output Layer) – Số node ra là số đặc điểm cần hướng tới của đối tượng (giá trị ra cần hướng đến – học có giám sát).
- Lớp ẩn (Hidden Layer) – Số node ẩn thường là không xác định trước, nó thường là do kinh nghiệm của người thiết kế mạng, nếu số node ẩn quá nhiều mạng sẽ công kênh, quá trình học sẽ chậm, còn nếu số node ẩn quá ít làm mạng học không chính xác.

Các neural ở các lớp trong thường được kết nối đầy đủ với tất cả các neural lớp ngoài, trên mỗi đường kết nối giữa 2 neural ở 2 lớp khác nhau có 1 trọng số mạng (weight). Các trọng số này đóng vai trò là các giá trị ẩn số mà mạng cần phải tìm ra (học) sao cho với các giá trị đầu vào, thông qua mạng ta nhận được kết quả xấp xỉ với đầu ra mong muốn tương ứng của mẫu học.

2. Hoạt động:

Ta sử dụng một số kí hiệu sau:

- x_i : Giá trị đầu vào của neural thứ i
- y_k : Giá trị đầu ra của neural thứ k
- V_{ij} : vector trọng số trên đường kết nối từ neural node vào thứ i tới neural node ẩn thứ j .
- W_{jk} : vector trọng số trên đường kết nối từ neural node ẩn thứ j tới neural node ra thứ k .
- d_k : Giá trị đầu ra mong muốn của neural nút ra thứ k
- η : tốc độ học (Hệ số học) của mạng.
- f : hàm truyền với: $f(x) = 1 / (1 + e^{-x})$ - Ở đây chúng ta sử dụng hàm truyền là hàm Sigmoid, giới hạn giá trị đầu ra trong khoảng $[0-1]$.
- Ta sử dụng thêm 1 số ký hiệu sau:
 - I_i – Input của node Input thứ i
 - O_i – Output của node Input thứ i

- I_{Hi} – Input của node Hidden thứ i
- O_{Hi} – Output của node Hidden thứ i
- I_{Oi} – Input của node Output thứ i
- O_{Oi} – Output của node Output thứ i

Thuật toán lan truyền ngược được mô tả như sau:

Input:

- Mạng feed-forward với n đầu vào, m nút ẩn và L đầu ra.
- Hệ số học η
- Sai số học ε
- Tập dữ liệu huấn luyện $D = \{x_i - \text{là vector đầu vào, } d_k - \text{là vector đầu ra mong muốn}\}$.

Output: Các vector trọng số sau khi đã được huấn luyện.

Thuật toán:

Bước 1: Khởi tạo trọng số V_{ij}, W_{jk} bởi các giá trị ngẫu nhiên nhỏ.

$$V_{ij} = \text{Random}(-1,1), W_{jk} = \text{Random}(-1,1)$$

Bước 2: Lan truyền tiến tính toán đầu ra thực tế y_k

- Tại node vào thứ i (Input):

$$I_{Ii} = x_i, O_{Ii} = I_{Ii}$$

- Tại node ẩn thứ p (Hidden):

$$I_{Hp} = \sum O_{Ii} V_{ip}, O_{Hp} = f(I_{Hp}) = 1 / (1 + e^{-I_{Hp}})$$

- Tại node ra thứ q (Output):

$$I_{Oq} = \sum O_{Hi} W_{iq}, O_{Oq} = f(I_{Oq}) = 1 / (1 + e^{-I_{Oq}})$$

Như vậy giá trị đầu ra thực tế của mạng với bộ trọng số ngẫu nhiên ban đầu là: $y_k = O_{Ok}$. Thực tế ta có thể nhận thấy đầu ra mong muốn d_k và đầu ra thực tế y_k là luôn khác nhau, chúng ta cần phải tính toán độ sai khác này và có phương pháp điều chỉnh

các trọng số mạng qua từng bước học sao cho qua mỗi lần học sự sai khác này giảm xuống, quá trình học sẽ dừng khi sự sai khác này đạt đến 1 ngưỡng giới hạn phù hợp nào đó.

Bước 3: Đánh giá lỗi học - lỗi trung bình bình phương (MSE: mean-square error):

$$E = 1/L * \text{sqrt}(\sum(\mathbf{d}_k - \mathbf{y}_k)^2)$$

Nếu $E \leq \epsilon$ thì dừng học.

Bước 4: Lan truyền ngược điều chỉnh trọng số

- Với mỗi nút q thuộc tầng ra, tính đạo hàm ngược sai số thành phần δ_q theo công thức:

$$\delta_q = (\mathbf{d}_q - \mathbf{y}_q)\mathbf{y}_q(1 - \mathbf{y}_q)$$

- Cập nhật các trọng số từ tầng ẩn tới tầng ra \mathbf{W}_{jk} :

$$\Delta \mathbf{w}_{pq} = \eta \delta_q \mathbf{O}_{Hp}$$

$$\mathbf{w}_{pq(\text{New})} = \mathbf{w}_{pq(\text{Old})} + \Delta \mathbf{w}_{pq}$$

- Với mỗi nút p thuộc tầng ẩn, tính đạo hàm ngược sai số δ_p theo công thức:

$$\delta_p = \mathbf{O}_{Hp}(1 - \mathbf{O}_{Hp})\sum(\mathbf{w}_{pk(\text{old})} \cdot \delta_k), \mathbf{k}=1..L$$

- Cập nhật các trọng số từ tầng vào tới tầng ẩn \mathbf{V}_{ij} :

$$\Delta \mathbf{v}_{ip} = \eta \delta_p \mathbf{O}_{Ii}$$

$$\mathbf{v}_{ip(\text{New})} = \mathbf{v}_{ip(\text{Old})} + \Delta \mathbf{v}_{ip}$$

Lặp lại bước 2 cho tới khi thỏa mãn điều kiện kết thúc ở bước 3. Kết quả thu được bộ trọng số chuẩn \mathbf{V}_{ij} , \mathbf{W}_{jk} sao cho giá trị đầu ra thực tế và giá trị đầu ra mong muốn là gần giống nhau nhất (Trung bình bình phương lỗi nhỏ nhất)

e) Một số vấn đề lưu ý trong xây dựng mạng MLP.

Xác định kích thước mẫu:

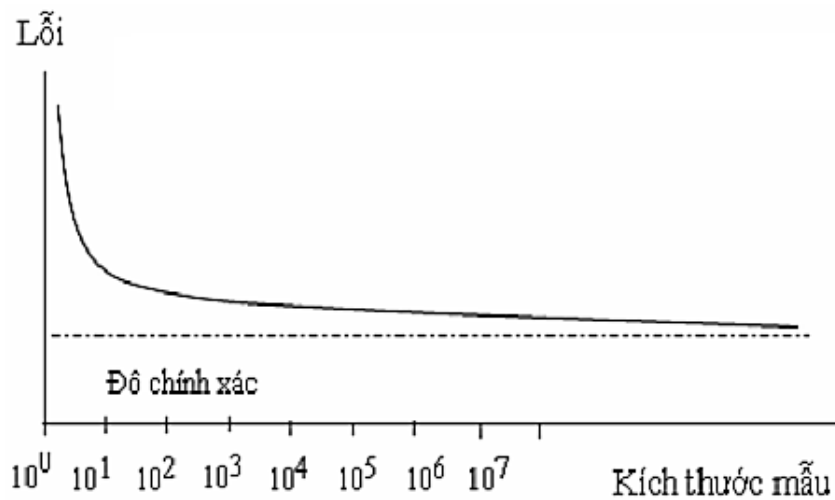
Không có nguyên tắc nào hướng dẫn kích thước mẫu phải là bao nhiêu đối với một bài toán cho trước. Hai yếu tố quan trọng ảnh hưởng đến kích thước mẫu:

- Dạng hàm đích: khi hàm đích càng phức tạp thì kích thước mẫu cần tăng.

- Nhiễu: khi dữ liệu bị nhiễu (thông tin sai hoặc thiếu thông tin) kích thước mẫu cần tăng.

Đối với mạng truyền thẳng, cho hàm đích có độ phức tạp nhất định, kèm một lượng nhiễu nhất định thì độ chính xác của mô hình luôn có một giới hạn nhất định. Nói cách khác độ chính xác của mô hình là hàm theo kích thước tập mẫu.

Hình 1.7. Ảnh hưởng của kích thước mẫu



Xác định số nơron tầng ẩn

Câu hỏi chọn số lượng nơron trong tầng ẩn của một mạng MLP thế nào là khó, nó phụ thuộc vào bài toán cụ thể và vào kinh nghiệm của nhà thiết kế mạng. Có nhiều đề nghị cho việc chọn số lượng nơron tầng ẩn h trong một mạng MLP. Chẳng hạn h phải thỏa mãn $h > (p-1)/(n+2)$, trong đó p là số lượng mẫu huấn luyện và n là số lượng đầu vào của mạng. Càng nhiều nút ẩn trong mạng, thì càng nhiều đặc tính của dữ liệu huấn luyện sẽ được mạng nắm bắt, nhưng thời gian học sẽ càng tăng.

Vấn đề quá khớp

Khái niệm quá khớp: Vấn đề quá khớp xảy ra khi mạng được luyện quá khớp (quá sát) với dữ liệu huấn luyện (kể cả nhiễu), nên nó sẽ trả lời chính xác những gì đã được học, còn những gì không được học thì nó không quan tâm. Như vậy mạng sẽ không có được khả năng tổng quát hóa. Vấn đề quá khớp xảy ra vì mạng có năng lực quá lớn. Có 3 cách để hạn chế bớt năng lực của mạng:

- Hạn chế số nút ẩn

- Ngăn không cho mạng sử dụng các trọng số lớn
- Giới hạn số bước luyện

Khi mạng được luyện, nó chuyển từ các hàm ánh xạ tương đối đơn giản đến các hàm ánh xạ tương đối phức tạp. Nó sẽ đạt được một cấu hình tổng quát hóa tốt nhất tại một điểm nào đó. Sau điểm đó mạng sẽ học để mô hình hóa nhiều, những gì mạng học được sẽ trở thành quá khớp. Nếu ta phát hiện ra thời điểm mạng đạt đến trạng thái tốt nhất này, ta có thể ngừng tiến trình luyện trước khi hiện tượng quá khớp xảy ra. Ta biết rằng, chỉ có thể để đánh giá mức độ tổng quát hóa của mạng bằng cách kiểm tra mạng trên các mẫu nó không được học. Ta thực hiện như sau: chia mẫu thành tập mẫu huấn luyện và tập mẫu kiểm tra. Luyện mạng với tập mẫu huấn luyện nhưng định kỳ dừng lại và đánh giá sai số trên tập mẫu kiểm tra. Khi sai số trên tập mẫu kiểm tra tăng lên thì quá khớp đã bắt đầu và ta dừng tiến trình luyện.

Chú ý rằng, nếu sai số kiểm tra không hề tăng lên, tức là mạng không có đủ số nút ẩn để quá khớp. Khi đó mạng sẽ không có đủ số nút cần thiết để thực hiện tốt nhất. Do vậy nếu hiện tượng quá khớp không hề xảy ra thì ta cần bắt đầu lại nhưng sử dụng nhiều nút ẩn hơn.

2. Mạng neural tích chập

2.1. Định nghĩa mạng neural tích chập

Những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Trong luận văn này, chúng ta sẽ trình bày về Convolution (tích chập) cũng như ý tưởng của mô

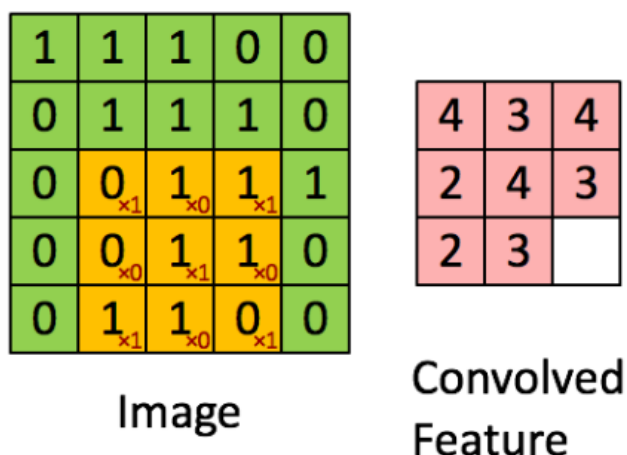
hình CNNs trong phân lớp chữ viết áp dụng trong bài toán nhận dạng biển số xe (Image Classification).

2.2. Convolution (tích chập)

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kỹ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Bạn có thể theo dõi cơ chế của tích chập qua hình minh họa bên dưới.

Hình 1.8. Minh họa tích chập



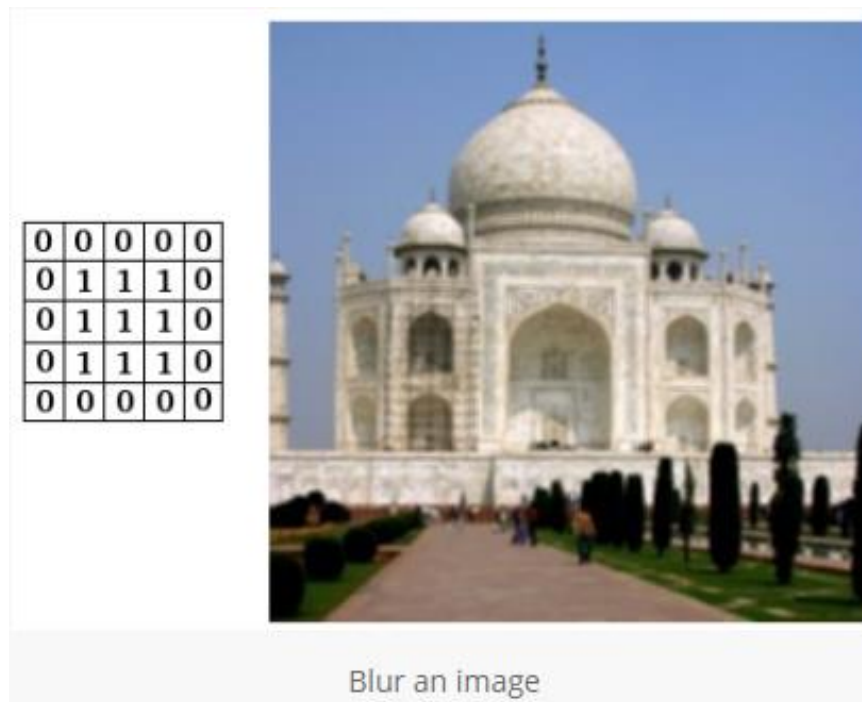
Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Giá trị đầu ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận (convolved feature) sinh ra từ việc trượt ma

trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái. Dưới đây là một vài ví dụ của phép toán tích chập.

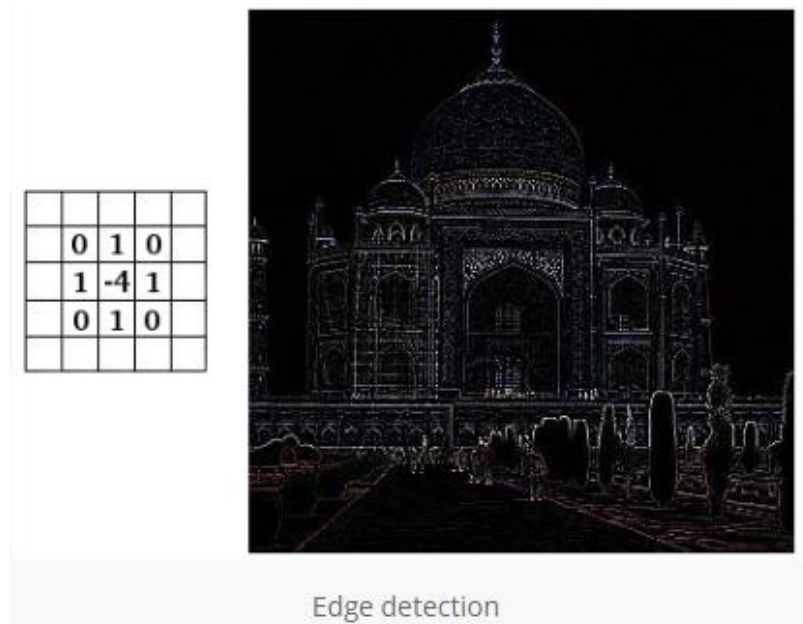
- Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.

Hình 1.9. Ảnh mờ sau khi chập



- Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.

Hình 1.10. Ảnh được phát hiện biên sau khi chập



2.3. Mô hình mạng neural tích chập

Bây giờ, Chúng ta đã biết thế nào là convolution. Vậy CNNs là gì? CNNs chỉ đơn giản gồm một vài layer của convolution kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU hay tanh để tạo ra thông tin trừu tượng hơn (abstract/higher-level) cho các layer tiếp theo.

Trong mô hình Feedforward Neural Network (mạng nơ-ron truyền thẳng), các layer kết nối trực tiếp với nhau thông qua một trọng số w (weighted vector). Các layer này còn được gọi là có kết nối đầy đủ (fully connected layer) hay affine layer.

Trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơ-ron layer trước đó.

Mỗi layer như vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling

layer dùng để chốt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Tuy nhiên, ta sẽ không đi sâu vào khái niệm của các layer này.

Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự *raw pixel* > *edges* > *shapes* > *facial* > *high-level features*. Layer cuối cùng được dùng để phân lớp ảnh.

Hình 1.11. Mô hình mạng neural tích chập

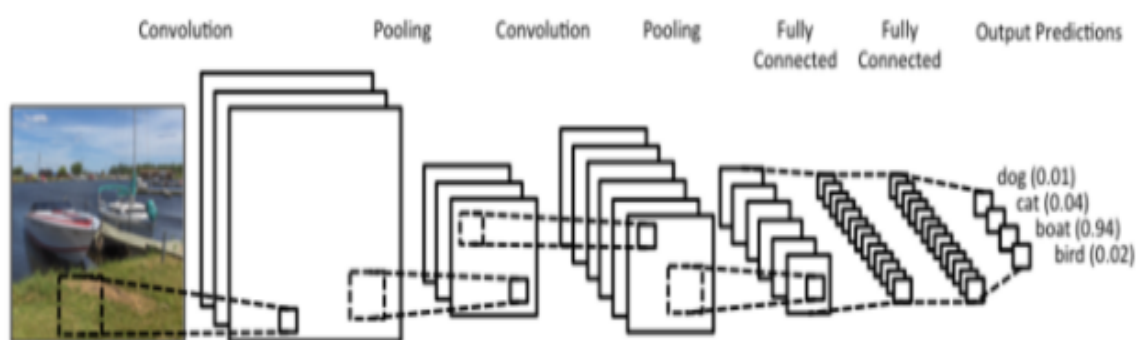


Image Classification with CNN

CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên. Ta phân biệt được một con chó với một con mèo nhờ vào các đặc trưng từ mức độ thấp (có 4 chân, có đuôi) đến mức độ cao (dáng đi, hình thể, màu lông).

2.4. Xây dựng mạng neural tích chập

Phần này sẽ giới thiệu một trong những mạng sâu được sử dụng rộng rãi đó là mạng tích chập sâu (deep convolutional networks). Chúng ta sẽ làm việc cụ thể với mạng tích chập để giải quyết bài toán phân loại chữ số viết tay từ tập dữ liệu MNIST.

Chúng ta sẽ bắt đầu mạng tích chập với việc sử dụng mạng truyền thống để giải quyết bài toán này trong phần trước. Mặc dù nhiều phép toán lặp nhưng chúng ta sẽ xây dựng mạng hiệu quả hơn. Chúng ta sẽ khám phá ra rất nhiều kỹ thuật hiệu quả: Tích chập (convolution), giảm số chiều (pooling), sử dụng GPUs để huấn luyện được nhiều dữ liệu hơn chúng ta đã thực hiện trên mạng cũ, mở rộng giải thuật huấn luyện dữ liệu (để giảm quá khớp – overfitting), sử dụng kỹ thuật dropout để giảm overfitting, việc sử dụng tổng hợp các mạng và các kỹ thuật khác. Kết quả là hệ thống làm việc gần như con người. Trong số 10.000 bức ảnh huấn luyện, hệ thống của chúng ta sẽ phân loại đúng 9.967 bức ảnh.

Phần còn lại của chương sẽ thảo luận về học sâu dưới góc độ tổng quan và chi tiết. Chúng ta sẽ tìm hiểu làm thế nào để các mô hình mạng nơron tích chập có thể ứng dụng để giải quyết các bài toán nhận dạng tiếng nói, xử lý ngôn ngữ tự nhiên và các lĩnh vực khác. Và chúng ta sẽ nghiên cứu về mạng nơron trong tương lai và học sâu (deep learning), từ các ý tưởng như giao diện người sử dụng hướng đích đến vai trò của học sâu trong trí tuệ nhân tạo.

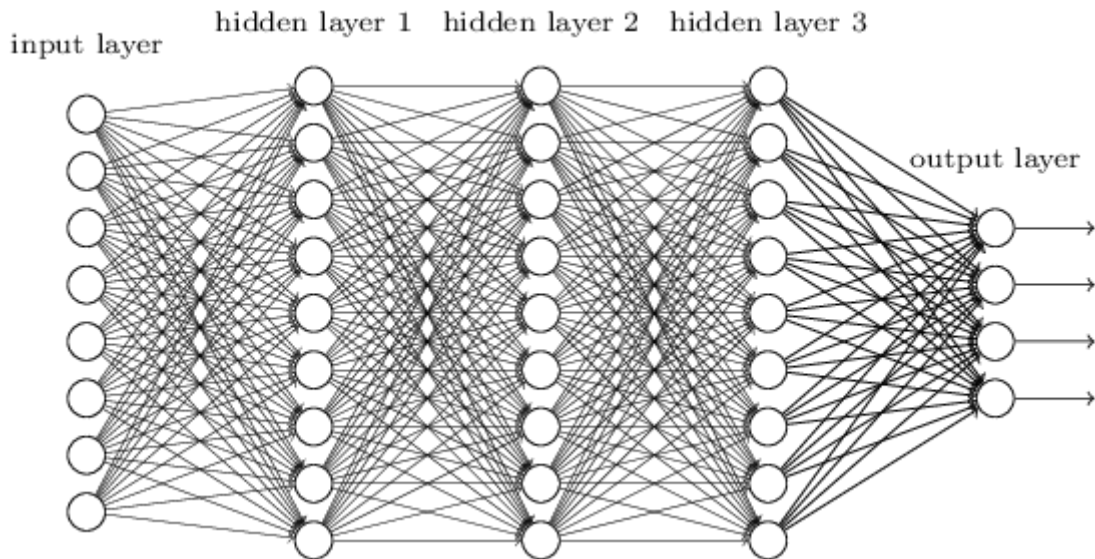
Phần này xây dựng dựa trên các phần trước sử dụng các ý tưởng như: lan truyền ngược (backpropagation), regularization, hàm softmax....

Trong các chương trước, chúng ta đã huấn luyện các mạng nơron nhận dạng chữ số viết tay khá tốt.



Chúng ta đã sử dụng mạng nơron mà trong đó các tầng liên kết đầy đủ với nhau. Tức là mỗi nơron trong mạng liên kết với tất cả các nơron trong tầng liền kề.

Hình 1.12. Mô hình mạng perceptron đa tầng



Đặc biệt, đối với mỗi điểm ảnh trong ảnh đầu vào, ta mã hóa cường độ của điểm ảnh là giá trị của nơron tương ứng trong tầng đầu vào.

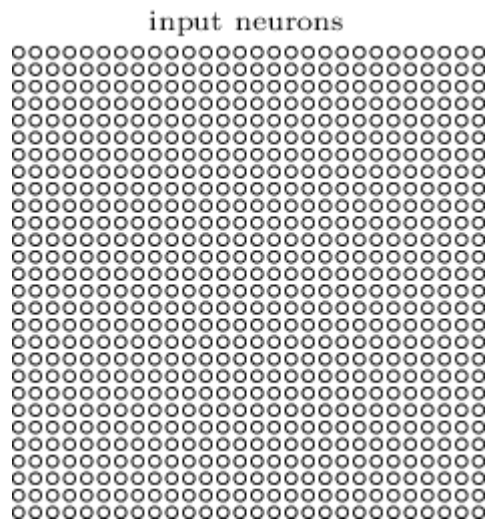
Đối với bức ảnh kích thước 28x28 điểm ảnh mà chúng ta đang sử dụng, mạng có 784 (28x28) nơron đầu vào. Sau đó ta huấn luyện trọng số (weight) và độ lệch (bias) để đầu ra của mạng như ta mong đợi là xác định chính xác ảnh các chữ số '0', '1', '2', ..., '8' hay '9'.

Mạng nơron trước đây của chúng ta làm việc khá tốt: chúng ta đã đạt được độ chính xác trên 98%, sử dụng tập dữ liệu huấn luyện và kiểm thử từ tập dữ liệu chữ viết tay MNIST. Nhưng sau khi xem xét kỹ lại, thì không cần thiết phải sử dụng mạng kết nối đầy đủ để phân loại ảnh. Lý do là kiến trúc mạng như vậy đã không tính đến hay xem xét đến cấu trúc không gian (spatial structure) của ảnh. Ví dụ, nó xử lý các điểm ảnh đầu vào mà còn cách xa nhau và gần nhau trên chính xác vị thế tương tự. Khái niệm như các cấu trúc không gian thay vì phải được suy ra từ dữ liệu huấn luyện. Nhưng điều gì sẽ xảy ra, thay vì bắt đầu với một kiến trúc mạng đơn giản, chúng ta sử dụng một kiến trúc mạng mà cố gắng tận dụng tối đa lợi thế của các cấu trúc không gian? Trong phần này, chúng ta mô tả mạng nơron tích chập.

Những mạng này sử dụng một kiến trúc đặc biệt phù hợp cho bài toán phân loại ảnh. Sử dụng kiến trúc này làm cho mạng tích chập huấn luyện nhanh hơn. Kết quả là giúp chúng ta huấn luyện sâu, mạng nhiều tầng, rất phù hợp cho phân loại ảnh. Ngày nay, mạng tích chập sâu hoặc một số biến thể của nó được sử dụng trong các mạng nơron để nhận dạng ảnh.

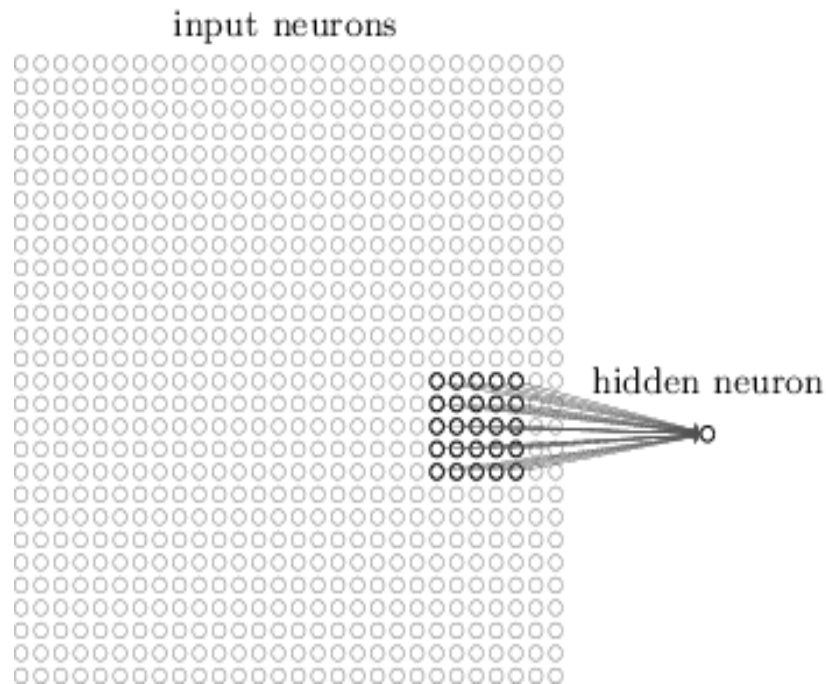
Mạng tích chập sử dụng 3 ý tưởng cơ bản: các trường tiếp nhận cục bộ (local receptive field), trọng số chia sẻ (shared weights) và tổng hợp (pooling). Chúng ta hãy xem xét lần lượt từng ý tưởng.

Trường tiếp nhận cục bộ (**Local receptive fields**): Trong các tầng kết nối đầy đủ được chỉ ra trước đây, đầu vào đã được mô tả là một đường thẳng đứng chứa các nơron. Trong mạng tích chập, ta sẽ thay thế các đầu vào là 28×28 nơron, giá trị tương ứng với 28×28 cường độ điểm ảnh mà chúng ta sử dụng:



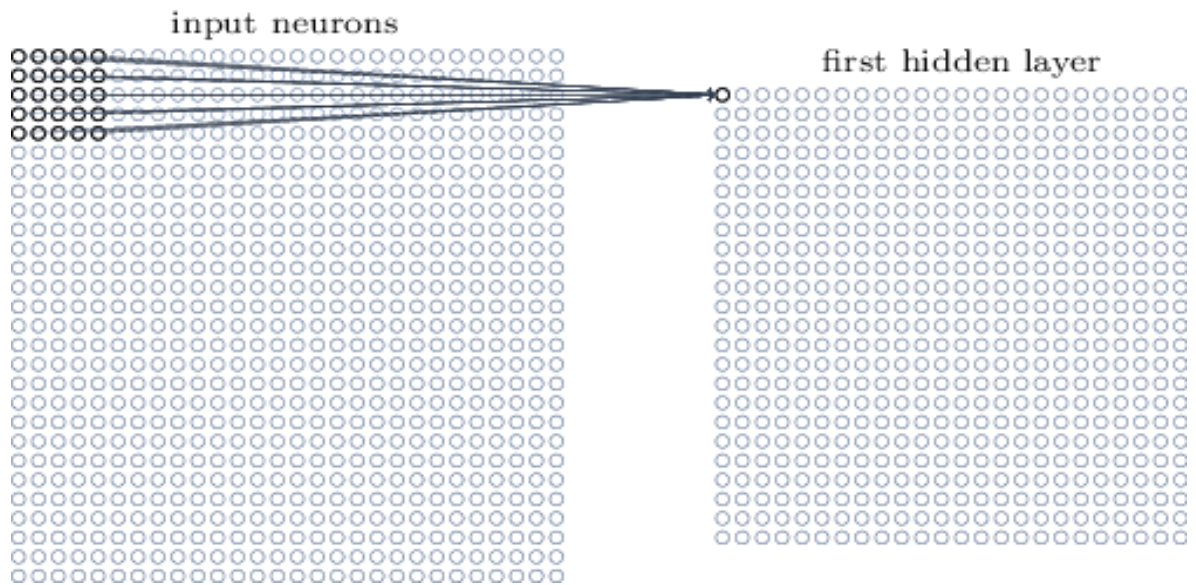
Như thường lệ chúng ta sẽ kết nối các điểm ảnh đầu vào cho các nơron ở tầng ẩn. Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào cho mỗi neuron ẩn. Thay vào đó, chúng ta chỉ kết nối trong phạm vi nhỏ, các vùng cục bộ của bức ảnh.

Để được chính xác hơn, mỗi nơron trong lớp ẩn đầu tiên sẽ được kết nối với một vùng nhỏ của các nơron đầu vào, ví dụ, một vùng 5×5 , tương ứng với 25 điểm ảnh đầu vào. Vì vậy, đối với một nơron ẩn cụ thể, chúng ta có thể có các kết nối như sau:

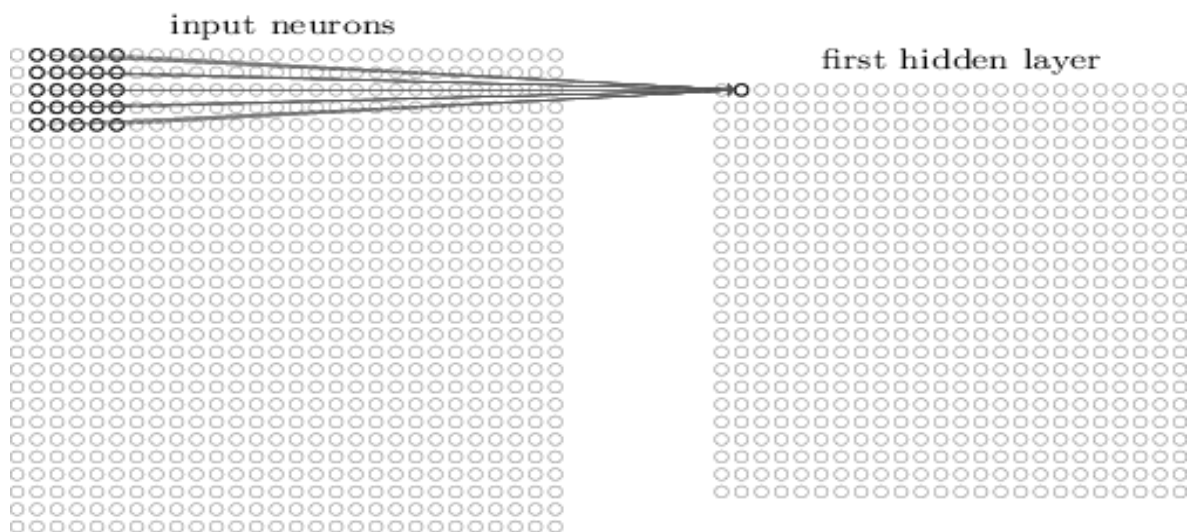


Vùng đó trong bức ảnh đầu vào được gọi là vùng tiếp nhận cục bộ cho nơron ẩn. Đó là một cửa sổ nhỏ trên các điểm ảnh đầu vào. Mỗi kết nối sẽ học một trọng số. Và nơron ẩn cũng sẽ học một độ lệch (overall bias). Bạn có thể hiểu rằng nơron lớp ẩn cụ thể là học để phân tích trường tiếp nhận cục bộ cụ thể của nó.

Sau đó chúng ta trượt trường tiếp nhận cục bộ trên toàn bộ bức ảnh. Đối với mỗi trường tiếp nhận cục bộ, có một nơron ẩn khác trong tầng ẩn đầu tiên. Để minh họa điều này một cách cụ thể, chúng ta hãy bắt đầu với một trường tiếp nhận cục bộ ở góc trên bên trái:



Sau đó, chúng ta trượt trường tiếp nhận cục bộ trên bởi một điểm ảnh bên phải (tức là bằng một nơron), để kết nối với một nơron ẩn thứ hai:



Và như vậy, việc xây dựng các lớp ẩn đầu tiên. Lưu ý rằng nếu chúng ta có một ảnh đầu vào 28×28 và 5×5 trường tiếp nhận cục bộ thì ta sẽ có 24×24 nơron trong lớp ẩn. Có được điều này là do chúng ta chỉ có thể di chuyển các trường tiếp nhận cục bộ ngang qua 23 nơron (hoặc xuống dưới 23 nơron), trước khi chạm với phía bên phải (hoặc dưới) của ảnh đầu vào.

Và như vậy, việc xây dựng các lớp ẩn đầu tiên. Lưu ý rằng nếu chúng ta có một ảnh đầu vào 28×28 và 5×5 trường tiếp nhận cục bộ, sau đó sẽ có 24×24 neuron trong lớp ẩn. Điều này là bởi vì chúng ta chỉ có thể di chuyển các trường tiếp nhận cục bộ 23 neuron ngang qua (hoặc 23 neuron xuống), trước khi chạm với phía bên phải (hoặc dưới) của ảnh đầu vào.

Trọng số và độ lệch (Shared weights and biases) : Mỗi một neuron ẩn có một độ lệch (bias) và 5×5 trọng số liên kết với trường tiếp nhận cục bộ. Những gì chúng ta vẫn chưa đề cập đến là chúng ta sẽ sử dụng các trọng số và độ lệch tương tự cho mỗi neuron ẩn 24×24 . Nói cách khác, đối với những neuron ẩn thứ j, k , đầu ra là:

$$\sigma \left(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l, k+m} \right).$$

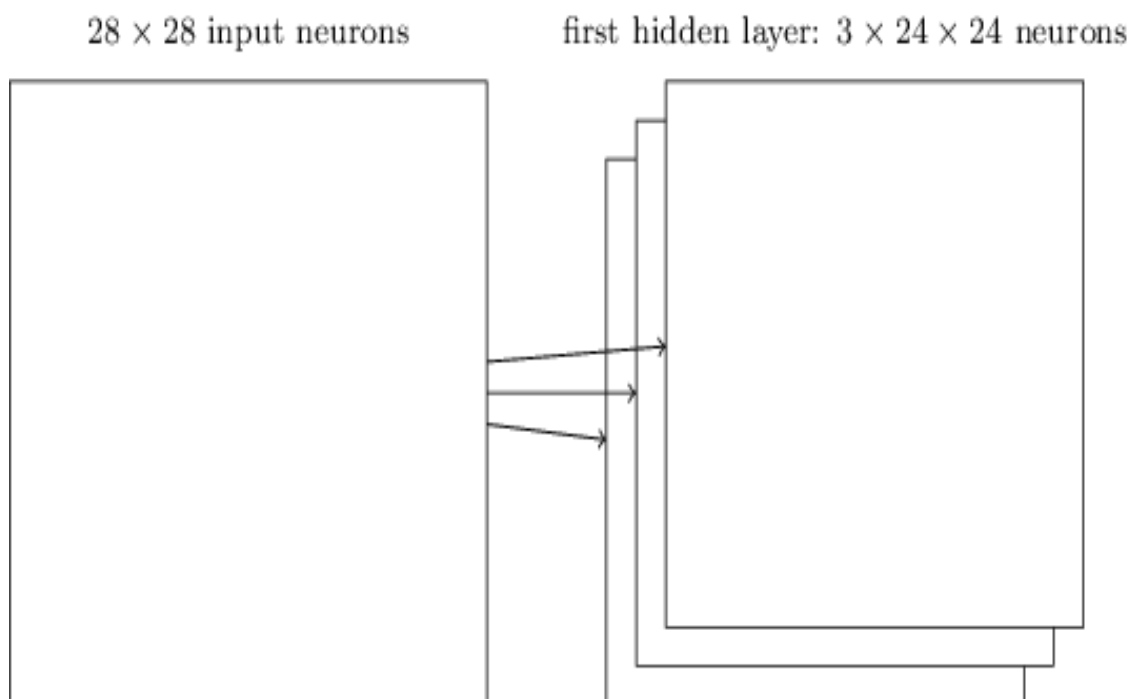
Ở đây, σ là hàm kích hoạt neuron - có lẽ là hàm sigmoid chúng ta sử dụng trong các chương trước. b là giá trị chung cho độ lệch. $w_{l,m}$, m là một mảng 5×5 của trọng số chia sẻ. Và, cuối cùng, chúng ta sử dụng a_{xy} biểu thị giá trị kích hoạt đầu vào tại vị trí x, y .

Chúng ta chưa xác định chính xác khái niệm về đặc trưng. Ta có thể nghĩ rằng của đặc trưng là loại mẫu đầu vào mà làm cho neuron hoạt động: ví dụ, nó có thể là biên của ảnh hoặc có thể là một dạng hình khối khác, ngay tại các vị trí khác nhau của ảnh đầu vào. Tại sao điều này lại có lí, giả sử rằng các trọng số và độ lệch mà các neuron ẩn chọn ra, một biên thẳng đứng (vertical edge) trong trường tiếp nhận cục bộ. Khả năng đó rất hữu ích ở các vị trí khác nhau trong bức ảnh. Và do đó, nó là hữu ích để áp dụng phát hiện các đặc trưng giống nhau trong ảnh. Để đặt nó trong thuật ngữ trừu tượng hơn một chút, mạng chấp được thích nghi với bất biến dịch (translation invariance) của các ảnh: di chuyển ảnh của một con mèo một ít, và nó vẫn là một hình ảnh của một con mèo.

Trong thực tế, đối với bài toán phân lớp các kí tự MNIST mà chúng ta đang nghiên cứu, bức ảnh được đặt ở trung tâm và chuẩn hóa kích thước. Chính vì vậy mà MNIST có ít bất biến chuyển dịch hơn so với các bức ảnh tìm thấy trong tự nhiên. Tuy nhiên, các đặc trưng có vẻ phù hợp hơn trong các ảnh đầu vào.

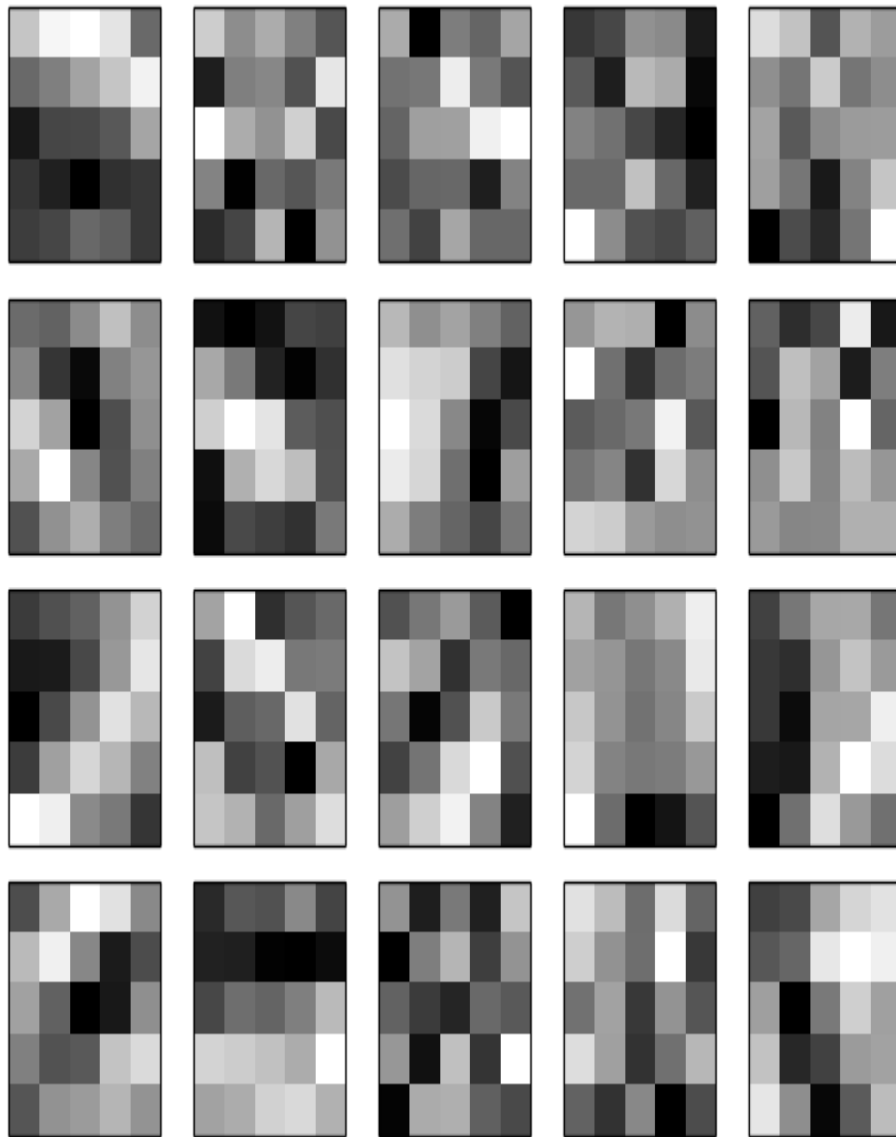
Vì lý do này, chúng ta đôi khi gọi các bản đồ từ các lớp đầu vào cho lớp ẩn là bản đồ đặc trưng (*feature map*). Chúng ta gọi các trọng số xác định các bản đồ đặc trưng là trọng số chia sẻ (*shared weights*). Và chúng ta gọi độ lệch xác định bản đồ đặc trưng là độ lệch chia sẻ (*shared bias*). Các trọng số được chia sẻ và độ lệch thường được gọi là hạt nhân (kernel) hay bộ lọc (filter).

Cấu trúc mạng chúng ta đã vừa mô tả có thể phát hiện một bản đồ đặc trưng. Để nhận dạng ảnh chúng ta cần nhiều hơn một bản đồ đặc trưng. Và do đó, một lớp tích chập hoàn chỉnh bao gồm vài bản đồ đặc trưng:



Trong ví dụ, có 3 bản đồ đặc trưng. Mỗi bản đồ đặc trưng được xác định bởi một tập 5×5 trọng số chia sẻ, và một độ lệch chia sẻ duy nhất. Kết quả là các mạng có thể phát hiện 3 loại đặc trưng khác nhau, với mỗi đặc trưng được phát hiện trên toàn bộ ảnh.

Chúng ta đã chỉ ra 3 bản đồ đặc trưng, để làm cho cho sơ đồ ở trên đơn giản. Tuy nhiên, trong thực tế mạng chập có thể sử dụng nhiều bản đồ đặc trưng hơn. Một trong những mạng chập đầu tiên là LeNet-5, sử dụng 6 bản đồ đặc trưng, mỗi bản đồ được liên kết đến một trường tiếp nhận cục bộ 5×5 , để phát hiện các kí tự MNIST. Vì vậy, các ví dụ minh họa ở trên là thực sự khá gần LeNet-5. Trong ví dụ chúng ta phát triển sau này trong chương này chúng ta sẽ sử dụng lớp tích chập với 20 và 40 bản đồ đặc trưng. Chúng ta hãy xem qua một số bản đồ đặc trưng đã được học.



Trên đây là 20 ảnh tương ứng với 20 bản đồ đặc trưng khác nhau (hay còn gọi là bộ lọc, hay là nhân). Mỗi bản đồ được thể hiện là một hình khối kích thước 5×5 , tương ứng với 5×5 trọng số trong trường tiếp nhận cục bộ. Khối trắng có nghĩa là một trọng số nhỏ hơn, vì vậy các bản đồ đặc trưng đáp ứng ít hơn để

tương ứng với điểm ảnh đầu vào. Khối sẫm màu hơn có nghĩa là trọng số lớn hơn, do đó, các bản đồ đặc trưng đáp ứng nhiều hơn với các điểm ảnh đầu vào tương ứng. Những hình ảnh trên cho thấy các kiểu đặc trưng mà lớp tích chập đáp ứng.

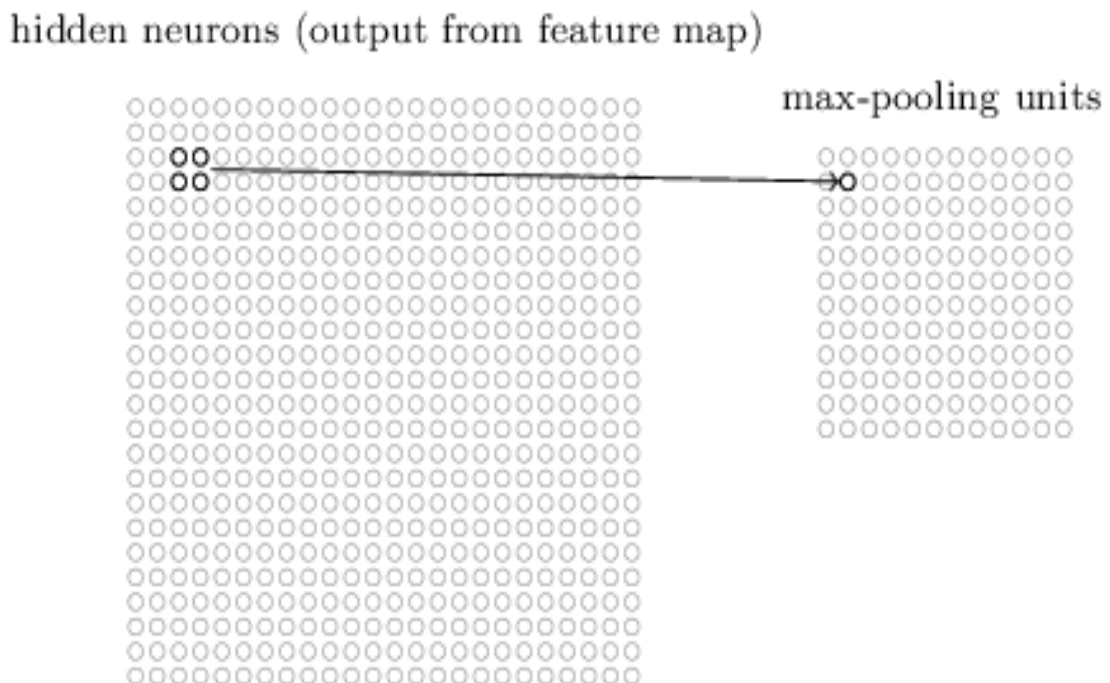
Một ưu điểm quan trọng của trọng số và độ lệch chia sẻ là nó làm giảm đáng kể số lượng các tham số liên quan đến một mạng tích chập. Đối với mỗi bản đồ đặc trưng chúng ta cần $25 = 5 \times 5$ trọng số chia sẻ và một độ lệch chia sẻ duy nhất. Vì vậy, mỗi bản đồ đặc trưng cần 26 tham số. Nếu chúng ta có 20 bản đồ đặc trưng thì cần $20 \times 26 = 520$ tham số để xác định lớp tích chập. Bây giờ chúng ta hãy làm phép so sánh, giả sử chúng ta có lớp đầu tiên kết nối đầy đủ, tức là có $784 = 28 \times 28$ nơron đầu vào, và số nơron lớp ẩn khiêm tốn là 30, như chúng ta sử dụng trong rất nhiều các ví dụ trước đó trong cuốn sách. Như vậy cần 784×30 trọng số, cộng thêm 30 sai lệch (bias), tổng số 23,550 tham số (parameter). Nói cách khác, lớp kết nối đầy đủ (fully – connected layer) sẽ cần số lượng tham số nhiều gấp 40 lần so với lớp tích chập (convolutional layer).

Tất nhiên, chúng ta không thể thực sự làm một so sánh trực tiếp giữa số lượng các tham số, bởi vì hai mô hình này khác nhau. Nhưng về trực giác dường như việc sử dụng bất biến dịch của các lớp tích chập sẽ giảm số lượng các tham số cần thiết mà vẫn đạt được hiệu quả giống như các mô hình kết nối đầy đủ. Mô hình mạng tích chập sẽ cho kết quả huấn luyện nhanh hơn giúp chúng ta xây dựng mạng sâu hơn sử dụng các lớp tích chập.

Cái tên “convolutional” xuất phát là các hoạt động trong phương trình đôi khi được biết đến như convolution. Chính xác hơn một chút, người ta đôi khi viết phương trình như $a_1 = \sigma(b + w * a_0)$, trong đó a_1 là tập kích hoạt đầu ra từ một bản đồ đặc trưng, a_0 là tập hợp các kích hoạt đầu vào, và $*$ được gọi là phép toán chập.

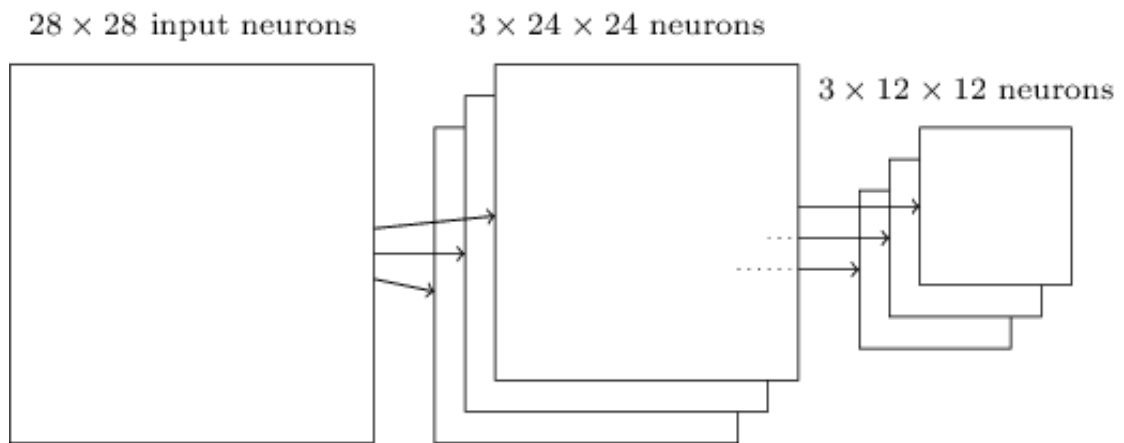
Lớp chứa hay lớp tổng hợp (Pooling layer): Ngoài các lớp tích chập vừa mô tả, mạng nơ-ron tích chập cũng chứa các lớp pooling. Lớp pooling thường được sử dụng ngay sau lớp tích chập. Những gì các lớp pooling làm là đơn giản hóa các thông tin ở đầu ra từ các lớp tích chập.

Ví dụ, mỗi đơn vị trong lớp pooling có thể thu gọn một vùng 2×2 nơ-ron trong lớp trước. Một thủ tục pooling phổ biến là max-pooling. Trong max-pooling, một đơn vị pooling chỉ đơn giản là kết quả đầu ra kích hoạt giá trị lớn nhất trong vùng đầu vào 2×2 , như minh họa trong sơ đồ sau:



Lưu ý rằng bởi vì chúng ta có 24×24 nơ-ron đầu ra từ các lớp tích chập, sau khi pooling chúng ta có 12×12 nơ-ron.

Như đã đề cập ở trên, lớp tích chập thường có nhiều hơn một bản đồ đặc trưng. Chúng ta áp dụng max-pooling cho mỗi bản đồ đặc trưng riêng biệt. Vì vậy, nếu có ba bản đồ đặc trưng, các lớp tích chập và max-pooling sẽ kết hợp như sau:

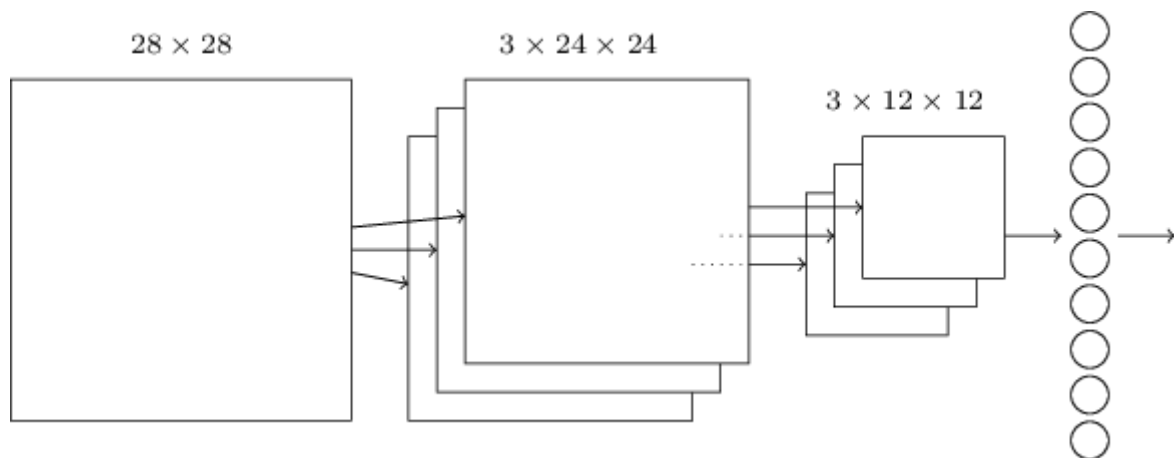


Chúng ta có thể hiểu max-pooling như là một cách cho mạng để hỏi xem một đặc trưng nhất được tìm thấy ở bất cứ đâu trong một khu vực của ảnh. Sau đó nó bỏ đi những thông tin định vị chính xác. Trực giác là một khi một đặc trưng đã được tìm thấy, vị trí chính xác của nó là không quan trọng như vị trí thô của nó so với các đặc trưng khác. Một lợi ích lớn là có rất nhiều tính năng gộp ít hơn (fewer pooled features), và vì vậy điều này sẽ giúp giảm số lượng các tham số cần thiết trong các lớp sau.

Max-pooling không phải là kỹ thuật duy nhất được sử dụng để pooling. Một phương pháp phổ biến khác được gọi là L2 pooling. Ở đây, thay vì lấy giá trị kích hoạt tối đa (maximum activation) của một vùng 2×2 neuron, chúng ta lấy căn bậc hai của tổng các bình phương của kích hoạt trong vùng 2×2 . Trong khi các chi tiết thì khác nhau, nhưng về trực giác thì tương tự như max-pooling: L2 pooling là một cách để cô đọng thông tin từ các lớp tích chập. Trong thực tế, cả hai kỹ thuật đã được sử dụng rộng rãi. Và đôi khi người ta sử dụng các loại pooling khác.

Đặt tất cả chúng lại với nhau (Putting it all together): Bây giờ chúng ta có thể đặt tất cả những ý tưởng lại với nhau để tạo thành một mạng tích chập hoàn chỉnh. Nó tương tự như kiến trúc chúng ta nhìn vào, nhưng có thêm một lớp

10 nơon đầu ra, tương ứng với 10 giá trị có thể cho các số MNIST ('0', '1', '2', v.v...):



Mạng bắt đầu với 28×28 nơon đầu vào, được sử dụng để mã hóa các cường độ điểm ảnh cho ảnh MNIST. Sau đó là một lớp tích chập sử dụng 5×5 trường tiếp nhận cục bộ và 3 bản đồ đặc trưng. Kết quả là một lớp $3 \times 24 \times 24$ nơon lớp ẩn. Bước tiếp theo là một lớp max-pooling, áp dụng cho 2×2 vùng qua 3 bản đồ đặc trưng (feature maps). Kết quả là một lớp $3 \times 12 \times 12$ nơon đặc trưng ở tầng ẩn.

Lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối. Đó là, lớp này nối mọi nơon từ lớp max-pooled tới mọi nơon của tầng ra. Kiến trúc kết nối đầy đủ này cũng giống như chúng ta sử dụng trong các chương trước.

Kiến trúc tích chập này hoàn toàn khác với các kiến trúc được sử dụng trong các chương trước. Nhưng về tổng thể thì tương tự: mạng cấu tạo từ nhiều đơn vị đơn giản, hành vi của nó được xác định bởi trọng số và độ lệch. Và mục tiêu tổng thể là vẫn như nhau: sử dụng dữ liệu huấn luyện để huấn luyện trọng số và độ lệch của mạng vì vậy mạng hiện tốt việc phân loại các chữ số đầu vào.

Đặc biệt, như phần đầu ta đã trình bày, ta sẽ huấn luyện mạng sử dụng gradient descent ngẫu nhiên và lan truyền ngược. Tuy nhiên, chúng ta cần thay

đổi thủ tục lan truyền ngược (backpropagation). Lý do là công thức của lan truyền ngược là cho các mạng với các tầng kết nối đầy đủ. May mắn thay, nó đơn giản để thay đổi công thức lan truyền ngược cho các lớp tích chập và các lớp max-pooling .

CHƯƠNG 2: TỔNG QUAN VỀ NHẬN DẠNG BIỂN SỐ XE

2.1 Khái niệm về hệ thống nhận dạng biển số xe.

2.1.1 Khái niệm

Hệ thống nhận dạng biển số xe là hệ thống có khả năng phân tích hình ảnh và xác định vùng chứa biển số trên xe, thông qua video, thiết bị ghi hình và hình ảnh.

2.1.2 Lịch sử và phát triển.

Năm 1992, công nghệ ALPR (Automatic License Plate Number) hay còn gọi là tự động nhận dạng biển số xe, được phát triển tại Đại học Cambridge ở Vương quốc Anh để ứng phó với chủ nghĩa khủng bố.

Đến năm 1996, công nghệ ALPR đã được hoàn thiện tại mỗi cổng phía tây Vương quốc Anh để đọc tất cả các biển đăng ký xe từ Ireland. Công nghệ ALPR tiếp tục được nghiên cứu và phát triển tại Anh. Kể từ tháng ba năm 2006, hầu hết các con đường, các trung tâm thị trấn, cảng, trạm xăng của London đã được lắp đặt camera chạy phần mềm ALPR.

Trên thế giới hiện nay, bài toán nhận dạng biển số xe được nghiên cứu và phát triển một cách sâu rộng. Nhiều tác giả với các công trình nghiên cứu được công bố với tỉ lệ nhận dạng ngày càng chính xác. Một số bài báo cáo nghiên cứu của các tác giả tiêu biểu trong vài năm trở lại đây như:

Chirag N. Paunwala, 2010 [1] với nội dung: rút trích vùng số xe trong ảnh. Ảnh đầu vào được tiền xử lý bằng cách phương pháp nâng cao chất lượng ảnh, sau đó tìm biên bằng Vertical Edge và xử lý một lần nữa bằng Opening và Closing. Các vùng ứng viên sau đó được kiểm tra bằng thuật toán scan theo dòng để tìm được vùng chứa biển số xe chính xác. Kết quả nhận dạng 750 ảnh trong các điều kiện khác nhau cho tỉ lệ $742/750 = 99.2$.

Choo Kar Soon, 2012 [2] với nội dung: nhận dạng biển số xe tại Malaysia, sử dụng giải thuật Adaboost để training tập dữ liệu gồm gần 100 ảnh biển số. Các ký tự được nhận dạng bằng phương pháp KNN. Kết quả nhận dạng biển số 98% và nhận dạng ký tự 95% trên ảnh tĩnh.

Báo cáo này nghiên cứu cách nhận dạng biển số xe với sự kết hợp của phép biến đổi Hough và giải thuật tìm Contour để cải thiện kết quả phát hiện. Vùng các ứng viên sau đó tiếp tục được scan theo dòng để đếm số đối tượng bị cắt và so sánh với ngưỡng, nhằm tìm ra vùng ứng viên thỏa mãn. Kết quả nhận dạng đạt 98-99%.

Phần mềm nhận dạng biển số xe, đã được ứng dụng thực tế tại các trạm cân, trạm gửi xe, các trụ đèn giao thông để phát hiện xe vi phạm.

2.1.3. Cách thức hoạt động của hệ thống nhận dạng biển số xe.

Hệ thống ALPR (Automatic License Plate Recognition) gồm phần cứng và phần mềm, trong đó phần cứng là camera thu nhận ảnh xe và phần mềm có chức năng nhận dạng biển số xe từ ảnh chụp của camera. Camera thu nhận ảnh được đặt tại một vị trí cố định sao cho có thể quét được hình ảnh xe một cách rõ ràng và chụp lại hình ảnh đối tượng xe có chứa biển số. Ảnh này được đưa vào phần mềm nhận dạng để trích ra chính xác biển số xe có trong ảnh, sau đó một thuật toán OCR (Optical Character Recognition) được sử dụng để lấy từng ký tự và chuyển đổi thành định dạng mà máy tính có thể phân biệt được các chữ và số như dạng text... Cùng với sự phát triển của công nghệ, camera ngày nay đã có thể chụp một cách rõ nét trong điều kiện xe chạy với tốc độ cao như ở các đường cao tốc.

Không có một hệ thống ALPR nào có thể nhận dạng chính xác 100%. Điều đó phụ thuộc vào nhiều yếu tố như thời tiết, độ sáng, góc của camera tới xe,... Một số yếu tố ảnh hưởng đến độ chính xác của hệ thống là:

- Độ phân giải của ảnh kém hoặc ảnh bị mờ.
- Điều kiện ánh sáng yếu, bị phản chiếu hoặc che bóng.
- Các đối tượng có dạng tương tự như biển số xe ở ngoại cảnh.
- Sự khác nhau về cấu trúc biển số xe của mỗi nước

2.1.4. Phân loại các ứng dụng nhận dạng biển số xe.

Có nhiều cách thức khác nhau để phân loại các ứng dụng nhận dạng biển số xe. Một trong những cách đơn giản là phân loại ứng dụng nhận dạng biển số xe thông qua mục đích sử dụng. Có thể chia ứng dụng nhận dạng biển số xe thành hai loại sau:

Loại 1: Giới hạn vùng nhìn

Đầu vào: Ảnh thu trực tiếp từ các thiết bị ghi nhận ảnh kỹ thuật số. Ảnh được ghi nhận thường chỉ giới hạn trong vùng có biển số xe.

Nguyên lý hoạt động: Các phương tiện giao thông phải chạy với một tốc độ đủ chậm để máy ghi nhận hình ảnh có thể thu được ảnh vùng biển số xe.

Ứng dụng: Những ứng dụng nhận dạng biển số xe loại này thường được dùng tại các trạm kiểm soát, các trạm thu phí, các bãi gửi xe tự động, các trạm gác công.

Loại 2: Không giới hạn vùng nhìn

Đầu vào: Ảnh đầu vào thu được từ các thiết bị ghi hình tự động, không phụ thuộc vào góc độ, các đối tượng xung quanh, ảnh không cần bắt buộc chỉ chụp vùng chứa biển số xe, mà có thể ảnh tổng hợp như chứa thêm các đối tượng như người, cây đường phố..., miễn là vùng biển số xe phải đủ rõ để có thể thực hiện nhận dạng được ký tự trong vùng đó.

Nguyên lý hoạt động: Do đặc tính không giới hạn vùng nhìn mà ảnh đầu vào có thể thu được từ một thiết bị ghi hình (camara, máy ảnh...). Và do đó, công việc đầu tiên là dò tìm trong ảnh, để xác định đúng vùng nào là biển số xe. Sau đó, thực hiện tách vùng và nhận dạng. Cuối cùng tùy thuộc vào mục đích sử dụng mà kết quả nhận dạng được truyền đi hay lưu trữ để phục vụ nhu cầu của người dùng cuối.

Ứng dụng: Vì không phụ thuộc vào hình ảnh thu được nên có thể dùng ứng dụng tại nhiều nơi như tại những nơi điều tiết giao thông, tại các vị trí nhạy cảm của giao thông như ngã ba, ngã tư đường giao nhau. Kiểm soát, phát hiện hành vi vi phạm an toàn giao thông.

2.1.5. Ứng dụng thực tiễn tại Việt Nam

Hệ thống nhận dạng biển số xe được xây dựng nhằm mục đích giám sát, kiểm soát các phương tiện. Dưới đây chúng ta đề cập đến một số ứng dụng phổ biến đối với hệ thống nhận dạng biển số xe:

- Thu phí giao thông: Lắp đặt hệ thống “Nhận dạng biển số xe” tại các trạm thu phí nhằm hỗ trợ hoặc tự động hóa công tác thu phí.

- Kiểm soát xe tại các đường biên giới: Mỗi quốc gia đều có những quy định riêng về biển số xe, để phục vụ cho công tác quản lý và phát hiện những phương tiện giao thông (xe) vượt biên giới bất hợp pháp. Việc lắp hệ thống “Nhận dạng biển số xe” tại các trạm kiểm soát sẽ góp phần hỗ trợ công tác kiểm tra và an ninh quốc gia.

- Các trạm gác cổng: Việc lắp đặt hệ thống “Nhận dạng biển số xe” sẽ hỗ trợ hoặc tự động hóa công tác mở cổng cho xe vào. Ngoài ra, hệ thống còn được ứng dụng vào công tác chống trộm xe, các bãi giữ xe tự động, điều tiết giao thông (chẳng hạn như Thành phố Dublin đã ứng dụng công nghệ “Nhận dạng biển số xe tự động” trong việc điều tiết giao thông theo dạng biển số chẵn/lẻ).

2.1.6. Phân loại biển số xe.

2.1.6.1. Quy định về màu sắc và các ký tự trên biển số.

- Biển trắng chữ đen dành cho dân sự.
- Màu trắng 2 chữ, 5 số là biển dành cho người nước ngoài.
- NG là xe ngoại giao.
- NN là xe của các tổ chức, cá nhân nước ngoài: Trong đó 3 số ở giữa là mã quốc gia, 2 số tiếp theo là số thứ tự.

(Ghi chú: Xe số 80 NG xxx-yy là biển cấp cho các đại sứ quán, thêm gạch đờ ở giữa và 2 số cuối là 01 là biển xe của tổng lãnh sự.)

- Biển đỏ chữ trắng là dành cho quân đội.

Bảng 2.1. Quy định biển số cho quân đội.

| | | | |
|----|-------------------------------|----|--------------------------|
| AT | Binh đoàn 12 | AD | QĐ 4, Binh đoàn Cửu Long |
| BB | Bộ Binh | BC | Binh chủng công binh |
| BH | Binh chủng hóa học | BS | Binh đoàn Trường Sơn |
| BT | Binh chủng thông tin liên lạc | BP | Bộ Tư lệnh biên phòng |
| HB | Học viện lục quân | HH | Học viện quân y |

| | | | |
|----|------------------------|-------------|------------------------|
| KA | Quân khu 1 | KB | Quân khu 2 |
| KC | Quân khu 3 | KD | Quân khu 4 |
| KV | Quân khu 5 | KP | Quân khu 7 |
| KK | Quân khu 9 | PP | Các quân y viện |
| QH | Quân chủng hải quân | QK,QP QC | phòng không không quân |
| TC | Tổng cục chính trị | TH | Tổng cục hậu cần |
| TK | Tổng cục CN Quốc phòng | TT | Tổng cục kĩ thuật |
| TM | Bộ tổng tham mưu | VT | Viettel |

2.1.6.2 . Quy định về biển số cho các tỉnh thành.

Bảng 2.2. Quy định biển số cho các tỉnh thành

| | | | | | |
|-------|-------------|-------|------------|----|---------------|
| 11 | Cao Bằng | 43 | Đà Nẵng | 77 | Bình Định |
| 12 | Lạng Sơn | 47 | Đắc Lắc | 78 | Phú Yên |
| 14 | Quảng Ninh | 48 | Đắc Nông | 79 | Khánh Hòa |
| 15,16 | Hải Phòng | 49 | Lâm Đồng | 80 | Các đơn vị TW |
| 17 | Thái Bình | 50-59 | TP.HCM | 81 | Gia Lai |
| 18 | Nam định | 60 | Đồng nai | 82 | Kon tum |
| 19 | Phú thọ | 61 | Bình dương | 83 | Sóc trăng |
| 21 | Yên Bái | 62 | Long an | 84 | Trà vinh |
| 22 | Tuyên quang | 63 | Tiền giang | 85 | Ninh thuận |
| 23 | Hà giang | 64 | Vĩnh long | 86 | Bình thuận |
| 24 | Lào cai | 65 | Cần thơ | 88 | Vĩnh phúc |
| 25 | Lai châu | 66 | Đồng tháp | 89 | Hưng yên |
| 26 | Sơn la | 67 | An giang | 90 | Hà nam |
| 27 | Điện biên | 68 | Kiên giang | 92 | Quảng nam |

| | | | | | |
|-------|-----------|----|------------|----|------------|
| 28 | Hòa bình | 69 | Cà mau | 93 | Bình phước |
| 29-32 | Hà nội | 70 | Tây Ninh | 94 | Bạc Liêu |
| 33 | Hà tây | 71 | Bến tre | 95 | Hậu giang |
| 34 | Hải dương | 72 | BR-VT | 97 | Bắc cạn |
| 35 | Ninh bình | 73 | Quảng bình | 98 | Bắc giang |
| 36 | Thanh hóa | 74 | Quảng trị | 99 | Bắc ninh |
| 37 | Nghệ an | 75 | Huế | | |
| 38 | Hà tĩnh | 76 | Quảng Ngãi | | |

Các xe mang biển A: Xe của Công An - Cảnh Sát tương ứng với các tỉnh.

2.2. Phương pháp nhận dạng biển số xe từ ảnh chụp của camera.

Có nhiều phương pháp để giải quyết vấn đề này nhưng đều quy về các phương pháp chính sau đây:

- Phương pháp dùng chuyển đổi Hough: dựa vào đặc trưng cạnh biên trích được, áp dụng các phương pháp xác định đường thẳng như phép biến đổi Hough để phát hiện các cặp đường thẳng gần song song ghép thành một ảnh biển số. Giao điểm của những đoạn thẳng này chính là vùng bao chứa biển số xe.

- Phương pháp hình thái học: dựa vào đặc trưng hình thái của biển số xe như màu sắc, độ sáng, sự đối xứng... để xác định và trích ra ảnh biển số.

- Phương pháp khớp mẫu: xem biển số là một đối tượng có khung nền riêng và sử dụng các cửa sổ dò để trích từng đối tượng đưa qua mạng nơron (neural network), trí tuệ nhân tạo (artificial intelligence) chẳng hạn để phân loại có phải là vùng biển số hay không.

2.2.1. Phương pháp chuyển đổi Hough.

Dò đặc trưng biên ngang, dọc: làm nổi bật các viền bao của tất cả các đối tượng trong ảnh trong đó có viền bao biển số. Phương pháp sử dụng các bộ lọc gradient để trích được các đặc trưng cạnh biên này. Nghiên cứu này sử dụng bộ lọc Sobel để tiến hành dò. Dùng chuyển đổi Hough tìm các đoạn thẳng ngang dọc trên cơ sở của ảnh nhị phân cạnh thu được từ bước trên.

Tách các đoạn thẳng ngang, dọc có thể là cạnh của biển số.

Trích ứng viên biển số: thành lập các hình chữ nhật là ứng viên cho biển số với tiêu chí cụ thể là các bộ 4 đoạn thẳng thu được sẽ qua đánh giá về kích thước, tỉ lệ chiều rộng trên chiều cao so với một ngưỡng nào đó.

Ưu điểm: độ chính xác cao, không phụ thuộc vào màu sắc của biển số xe.

Nhược điểm: Độ phức tạp tính toán khá cao. Khi ảnh có thêm nhiều đối tượng khác thì khối lượng tính toán tăng lên rất nhiều do mục đích là phải xác định được vùng con nào chứa biển số xe và phụ thuộc rất lớn vào bước trích đặc trưng biên cạnh dẫn đến là các đoạn thẳng ứng viên thu được thường ngắn hơn nhiều so với chiều dọc cũng như chiều ngang của biển số.

2.2.2. Phương pháp hình thái học.

Nhóm tác giả Chirag N. Paunwala, 2012 đại diện cho phương pháp này, với kết quả nhận dạng rất tốt 99.5%.

Nội dung của phương pháp: Dựa vào đặc trưng quan trọng là biển số xe máy có độ sáng (tức mức xám khi chuyển bức ảnh về dạng xám) là tương đối khác so với các vùng khác trong bức ảnh, cũng như sự phân bố mức xám là khá đồng đều trên biển số và vì vậy khi được nhị phân hoá, vùng biển số là một đối tượng có đặc thù hình thái, có thể phân biệt được với các vùng khác. Như vậy các bước thực hiện là:

- Xác định ngưỡng xám. Thực chất là không có phương pháp nào chọn cho đúng ngưỡng xám để thực hiện. Thay vào đó, ngưỡng xám sẽ được quét trong một khoảng nào đó. Thông qua lược đồ xám ta nhận thấy vùng biển số thường sẽ có độ sáng tương đối lớn (từ 130-200) vì vậy ta sẽ xác định ngưỡng xám cần chọn sẽ thuộc vùng này nhờ đó ta sẽ giảm được thời gian lập tìm ngưỡng xám.

- Nhị phân hoá ảnh xám đầu vào với ngưỡng xám đã xác định.
- Lọc các nhiễu gây ảnh hưởng xấu tới đối tượng biển số.
- Gắn nhãn cho các đối tượng trong ảnh nhị phân thu được.
- Trích ra các đối tượng ứng viên biển số theo tiêu chí cụ thể của biển số xe về chiều cao, chiều rộng, tỉ lệ các cạnh, diện tích, trọng tâm, số điểm

cắt...

Hình 2.1 Ảnh xám và lược đồ xám của ảnh



2.3. Phương pháp nhận dạng ký tự trong biển số xe.

Phương pháp phổ biến nhất để nhận dạng ký tự là sử dụng mạng nơron (hoặc SVM, K-NN,...), tức là huấn luyện cho máy tính để nhận dạng các ký tự. Tuy nhiên do số lượng ký tự trên biển số là không nhiều nên để đảm bảo tốc độ xử lý, chúng ta cũng có thể sử dụng phương pháp Hình thái học để giải quyết khâu này bởi vì các ký tự đều có những đặc điểm hình thái đặc biệt có thể phân biệt với nhau chẳng hạn như “0” có lỗ trống ở giữa, “8” có 2 lỗ trống hay “X” đối xứng 2 trục dọc và ngang... Khâu này được thực hiện trên cơ sở xây dựng cây nhị phân tối ưu của các đặc điểm hình thái nên đảm bảo tính khoa học và tính chính xác cao. Thuật toán cơ bản của bước này như sau:

- Quan sát chọn ra các đặc tính phân biệt ký tự để xây dựng ma trận đặc tính.
- Xây dựng cây nhị phân tối ưu từ ma trận đặc tính và tập ký tự thu được.
- Quan sát cây nhị phân, kiểm tra số đặc tính như vậy đã đủ để nhận dạng chưa, thiếu (dư) thì phải bổ sung (bỏ đi) và quay lại bước đầu tiên.
- Tiến hành nhận dạng các ký tự trên cơ sở cây nhị phân tối ưu tìm được.

2.4. Phạm vi nghiên cứu và hướng giải quyết.

Trong đồ án này, công việc cần phải giải quyết vấn đề phát hiện vùng chứa biển số xe và nhận dạng ký tự trong biển số. Với bài toán phát hiện vùng chứa biển số cách tiếp cận theo phương pháp hình thái học để phát hiện vùng biển số cho các biển đăng ký xe của Việt Nam. Bài toán nhận dạng ký tự sẽ sử dụng mô hình mạng Neural tích chập để tiến hành nhận dạng.

Một số đặc điểm để nhận dạng biển số xe tại Việt Nam.

a) Tiêu chuẩn về kích thước (theo quy định của Bộ Giao Thông Vận Tải)

Ở mỗi nước thường có tiêu chuẩn về kích thước nhất định. Đối với nước ta, biển số xe qui định khá đồng đều cho mỗi loại xe, tỷ lệ chiều dài, rộng cho mỗi loại xe là như nhau. Quy định về kích thước như sau:

- Biển ô tô
 - Chiều cao: 110 mm. Chiều rộng: 470 mm (biển dài).
 - Chiều cao: 200 mm. Chiều rộng: 280 mm (biển vuông).
- Biển xe máy:
 - Chiều cao: 140 mm. Chiều rộng: 190 mm.

Như vậy, tỉ lệ Chiều cao / Chiều rộng là:

- $0.18 < \text{Chiều cao}/\text{Chiều rộng} < 0.3$ (biển số có 1 hàng).
- $0.6 < \text{Chiều cao}/\text{Chiều rộng} < 0.85$ (biển số xe có 2 hàng).

Từ các đặc điểm này, ta có thể xác định được vùng nào có khả năng là biển số theo ràng buộc về kích thước.

b) Tiêu chuẩn về ký tự.

Theo đo đạc trên biển số thực tế, mỗi ký tự thường có tỷ lệ kích thước về chiều rộng, chiều cao tương ứng với chiều dài và rộng của biển số xe. Ví dụ, chiều cao của mỗi ký tự luôn nhỏ hơn 85% chiều cao của biển số xe và luôn lớn hơn 60% chiều cao của biển xe đối với biển số xe có một hàng, với biển số xe có hai hàng

thì chiều cao mỗi ký tự không quá 50% chiều cao của biển số xe. Chiều rộng của ký tự thường không lớn hơn 20% chiều cao của mỗi ký tự. Mỗi ký tự của biển số xe được xem như là một vùng liên thông con hay contour con (bao đóng). Do đó, chúng ta có thể đếm số contours con thỏa mãn tính chất đó là ký tự. Ở nước ta chỉ có số ký tự trên mỗi biển số xe nằm trong khoảng 6 đến 9 ký tự.

Từ những phân tích trên, có thể tóm tắt phương pháp thực hiện của chương trình nhận dạng như sau:

Bước 1: Ảnh đầu vào là ảnh màu BGR, tiền xử lý bằng các thuật toán xử lý ảnh.

Bước 2: Tìm các contour trên ảnh xe. (Mỗi contour là 1 vùng bao kín, do vùng biển số là 1 vùng bao kín nên sẽ tương ứng với một contour)

Bước 3: Lọc các contour theo các tiêu chí như kích thước, góc, tỉ lệ, số ký tự,...

Bước 4: Xử lý kết quả đầu ra để lấy vùng biển số

Bước 5: Tách ký tự trên vùng biển số tìm được

Bước 6: Đưa tập ký tự đã tách vào mạng Neural để nhận dạng

Bước 7: Hiện thị kết quả lên giao diện chương trình

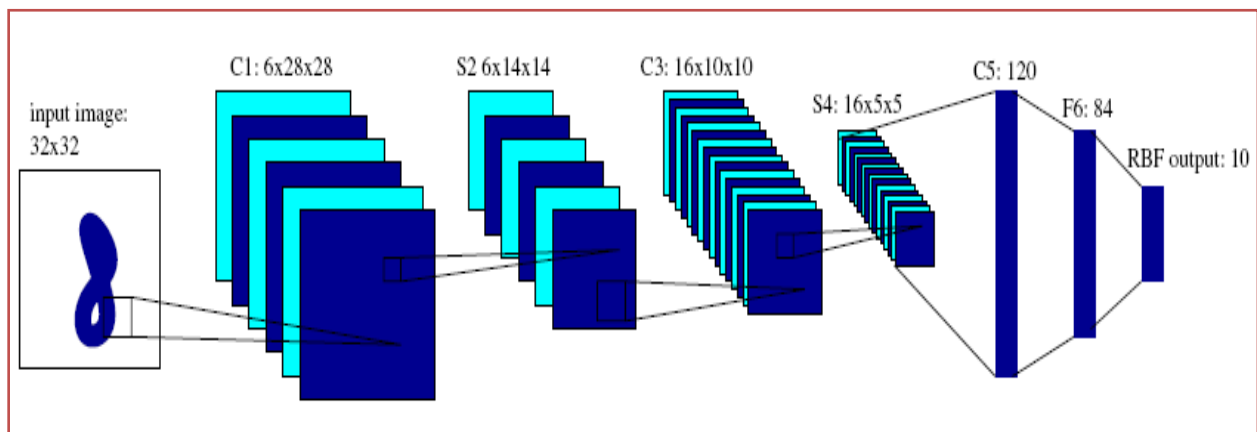
CHƯƠNG 3: ỨNG DỤNG MẠNG NEURAL TÍCH CHẬP TRONG NHẬN DẠNG KÝ TỰ

3.1. Xây dựng mô hình mạng

Sau khi đã tách được các ký tự dưới dạng ảnh từ biển số xe, bước tiếp theo chúng ta cần nhận dạng các ký tự từ ảnh chuyển về text. Để nhận dạng được các ký tự có rất nhiều phương pháp nhận dạng, có thể là KNN, SVM, mạng neural lan truyền ngược,... Ở đây chúng ta sử dụng mạng neural tích chập trong nhận dạng ký tự số nguyên do mạng neural tích chập có độ chính xác cao, và hiệu năng tốc độ xử lý tính toán nhanh hơn các mạng trước đó.

Mô hình mạng neural tích chập được xây dựng để nhận dạng các ký tự như sau:

Hình 3.1. Mô hình mạng neural tích chập trong nhận dạng ký tự viết tay



Ảnh đầu vào là 1 bức ảnh thô kích thước 32x32 pixel. Chúng ta sử dụng 6 ma trận chập kích thước 5x5 cho ra 6 ma trận ảnh đặc trưng sau khi chập lần 1 đó là các ma trận ánh xạ đặc trưng ở tầng chập C1, mỗi ma trận ánh xạ đặc trưng này có kích thước 28x28. Tức là ảnh gốc ban đầu được phân tích theo 6 chiều đặc trưng khác nhau với ma trận chập 5x5

Do kích thước các ảnh đặc trưng ở tầng chập C1 có kích thước 28x28 còn lớn, cho nên bước tiếp theo chúng ta thực hiện phép giảm số chiều ở ma trận đặc trưng (down Sampling – hay Pooling – hay subsampling) với hệ số tỷ lệ là 2 sử dụng hàm max:

Ví dụ 2 ô cạnh nhau trong ma trận đặc trưng có giá trị 3,5 tạo thành 1 ô trong ma trận đặc trưng subsampling là : $\max(3,5)=5$

Như vậy với 6 ma trận đặc trưng kích thước 28x28 ở tầng chập C1 ta tạo được 6 ma trận kích thước 14x14 ở tầng subsampling (S2)

Tiếp tục sử dụng 16 ma trận chập kích thước 5x5 chập với các ma trận ở tầng S2 ta được 16 ma trận ánh xạ đặc trưng kích thước 10x10 ở tầng chập C3

Do kích thước các ảnh đặc trưng ở tầng chập C3 có kích thước 14x14 còn lớn, cho nên bước tiếp theo chúng ta thực hiện phép giảm số chiều ở ma trận đặc trưng (down Sampling – hay Pooling – hay subsampling) với hệ số tỷ lệ là 2 sử dụng hàm max. Kết quả với 16 ma trận đặc trưng kích thước 10x10 ở tầng chập C3 ta tạo được 16 ma trận kích thước 5x5 ở tầng subsampling (S4)

Tiếp tục sử dụng 120 ma trận chập kích thước 5x5 chập với các ma trận ở tầng S4 ta được 120 ma trận ánh xạ đặc trưng kích thước 1x1 ở tầng chập C5

Do các đặc trưng ở tầng chập C5 là các điểm đặc trưng 1x1, cho nên ta không thực hiện phép toán subsampling nữa.

Tiếp theo ta sử dụng phép toán max để giảm kích thước ở tầng chập C5 do tầng C5 có tới 120 node đặc trưng, ta dùng hàm max giảm xuống còn 84 node ở tầng F6.

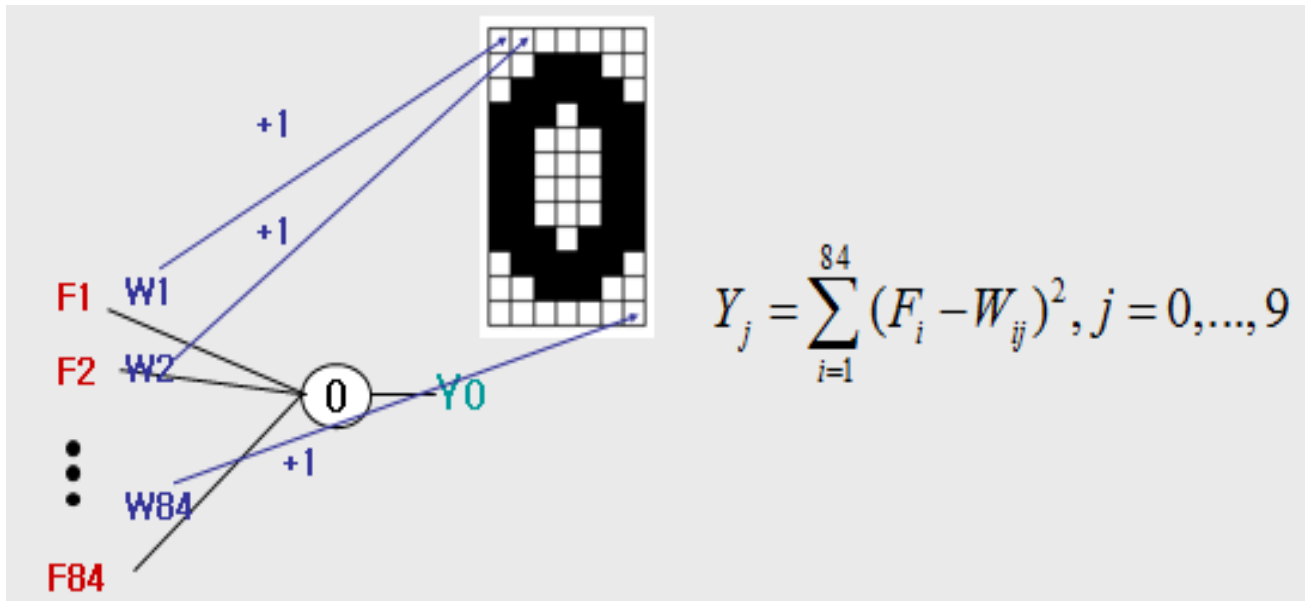
Lưu ý rằng đối với mỗi khối ở các tầng từ đầu tới F6 ta sử dụng hàm kích hoạt Sigmoid dạng :

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

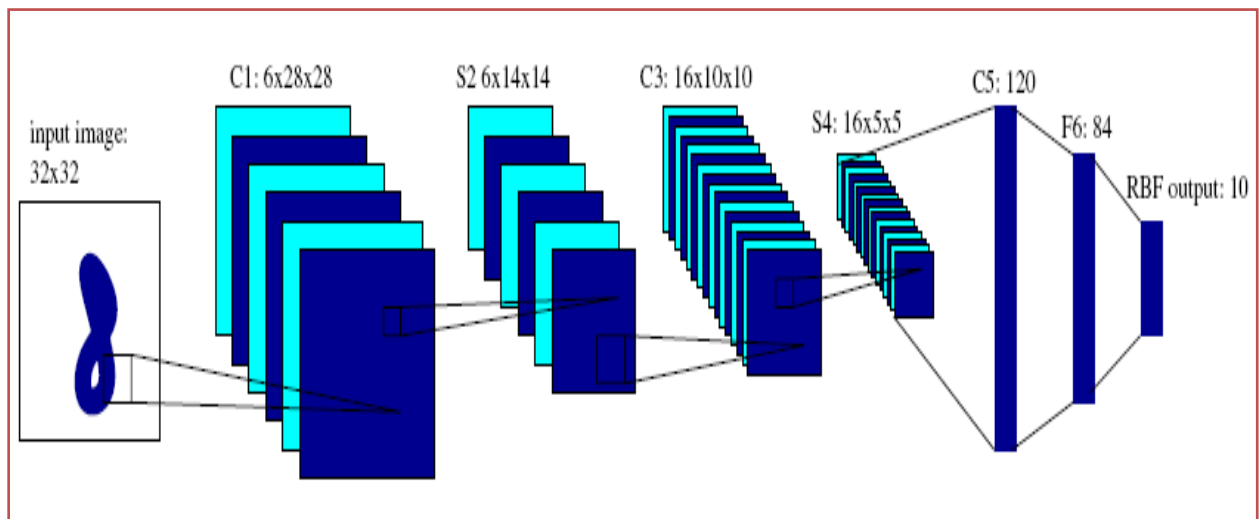
để tính toán giá trị ra của các node mạng, tức là $y' = f(A*I)$ trong đó A là ảnh chập, I là ma trận chập, và y' là giá trị 1 node trên ma trận ánh xạ đặc trưng

Với 84 node ở tầng F6, ở đây sử dụng mô hình mạng neural truyền thẳng với các kết nối Fully Connection, với 10 outputs được thiết kế như sau:

Hình 3.2. Minh họa Fully Connection

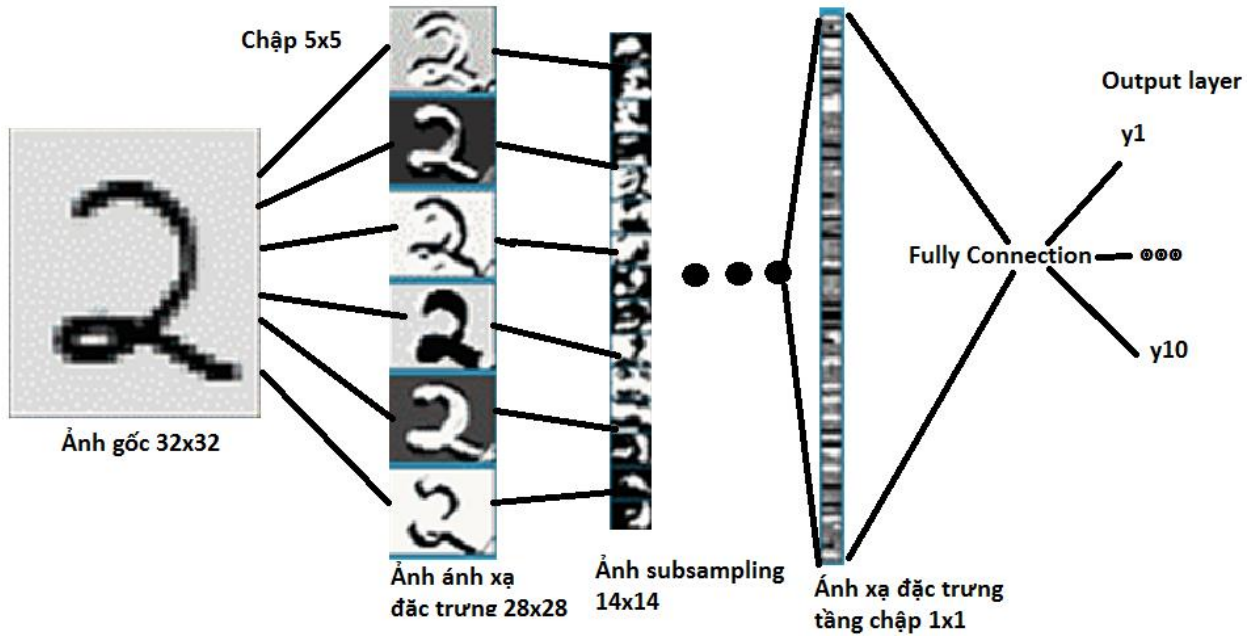


Với mỗi giá trị ra được tính như công thức trên



Ta có thể hình dung mô hình nhận dạng chữ viết được thực hiện như sau:

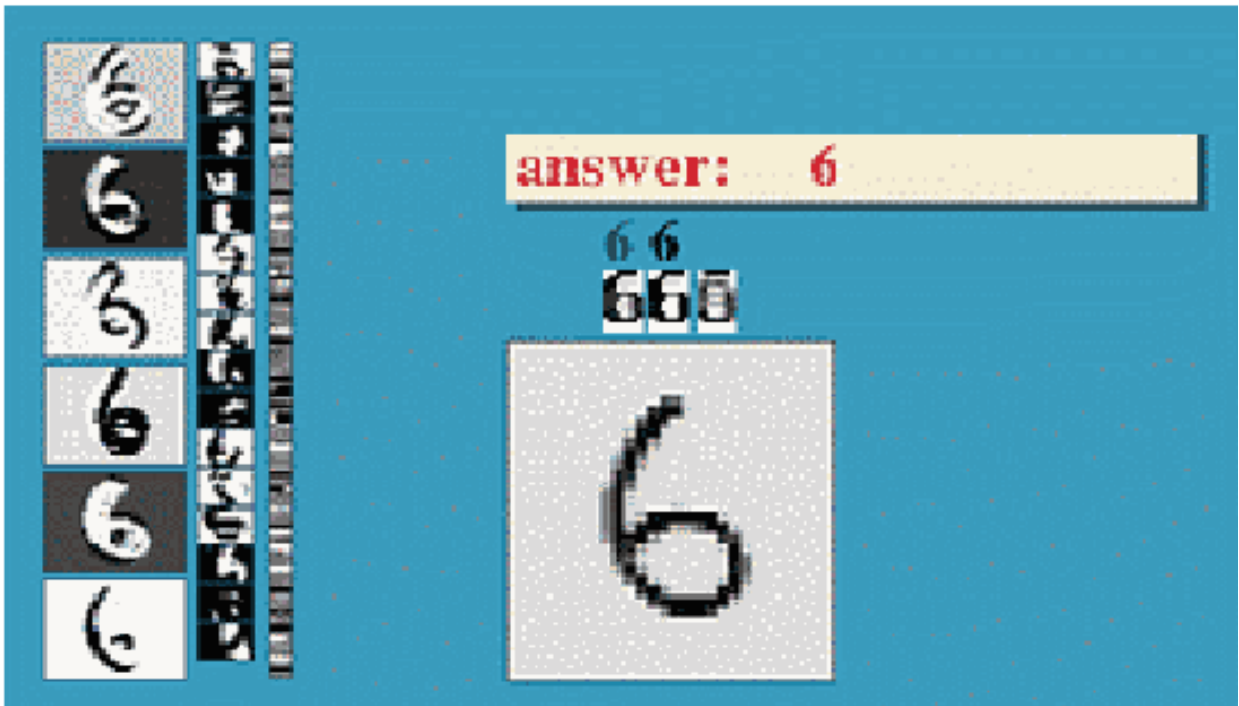
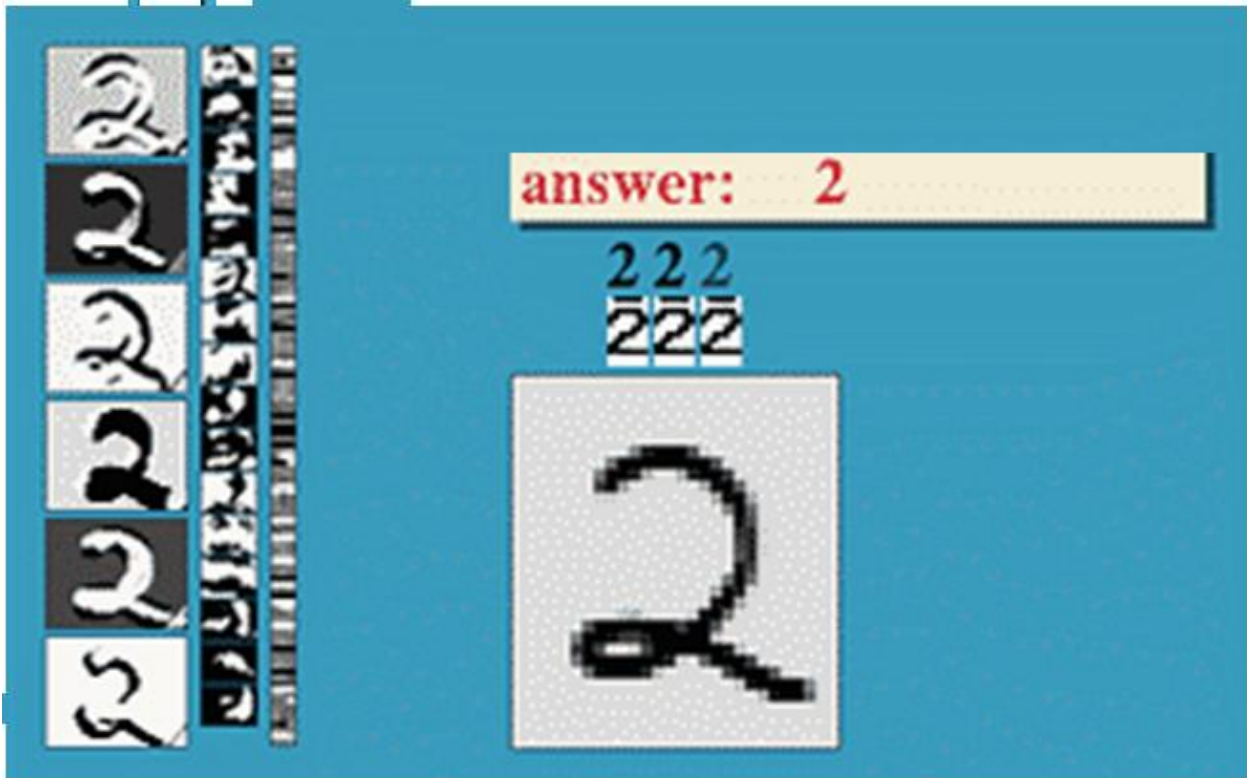
Hình 3.3. Minh họa các bước tích chập



3.2. Kết quả nhận dạng ký tự viết tay

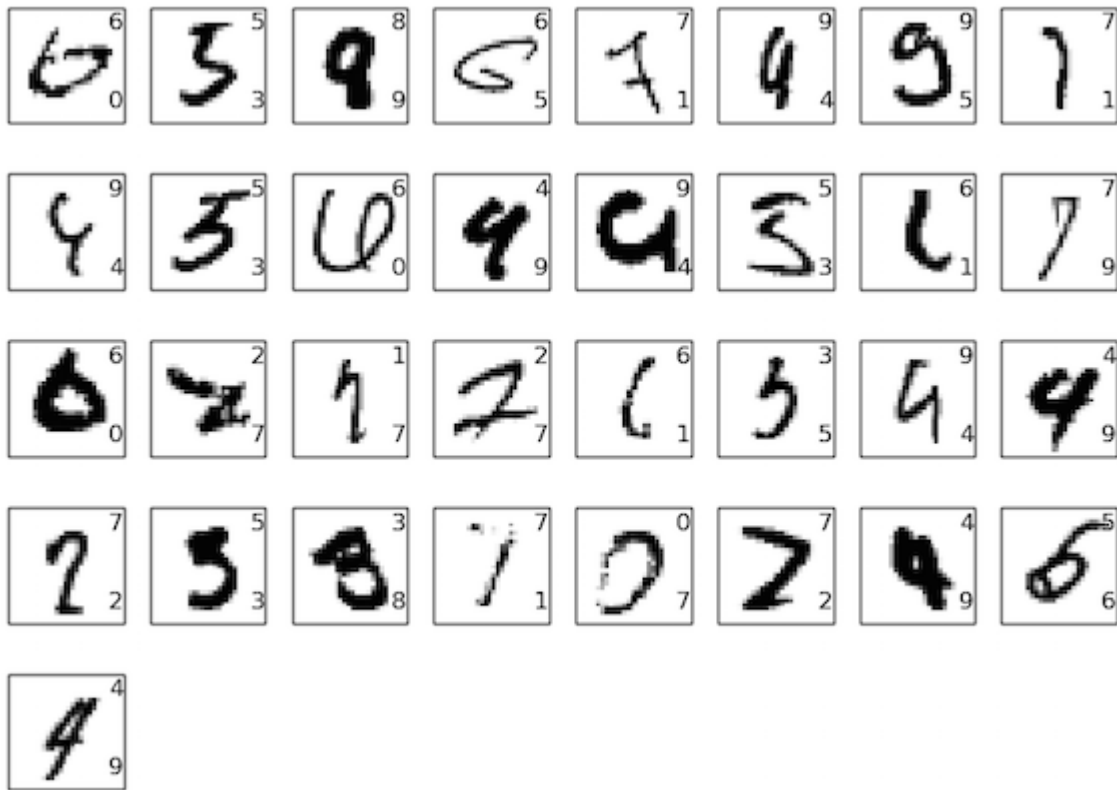
Ta xây dựng phần mềm nhận dạng biển số xe dựa trên mạng neural tích chập, với phần nhận dạng ký tự được trình bày ở trên, ta có kết quả thực nghiệm nhận dạng chữ viết tay với phần mềm mạng neural tích chập:

Hình 3.4. Kết quả thực nghiệm CNN



Quá trình thử nghiệm với 1000 chữ viết tay với phần mềm ta thấy rằng kết quả chính xác với 99.6%, một số kết quả sai do chúng ta viết không thể phân biệt được, ví dụ như với bộ chữ viết sau:

Hình 3.6. Một số mẫu chữ lỗi



Với phần mềm nhận dạng chữ viết tay kết quả thu được gần như có độ chính xác tuyệt đối, ta áp dụng nhận dạng chữ viết tay trong việc nhận dạng biển số xe

3.3. Kết quả thực nghiệm nhận dạng biển số xe với mạng Neural tích chập

Chạy ứng dụng nhận dạng biển số, đưa lần lượt 376 ảnh dữ liệu về xe để kiểm tra kết quả phát hiện biển số, tách ký tự và nhận dạng ký tự của chương trình, ghi nhận kết quả.

Tỉ lệ biển nhận dạng đúng vùng biển số: 372/376 ~ 98%



Biển bị bóng, phản chiếu

Biển số bị tối, thiếu ánh sáng

Biển mờ, mù, ố vàng

Hình 4.2. Một số biển không phát hiện được biển số

Tỉ lệ biển tách đúng và đầy đủ ký tự: 330/335 ~ 98%



Hình 3.3. Một số biển không tách đúng ký tự Chữ số bị dính với các vật bên ngoài như đinh ốc, ký tự bị mờ nét, mất nét, loang lổ,...

Kết quả nhận dạng chung của ứng dụng từ khâu phát hiện biển số, đến tách ký tự và nhận dạng ký tự đạt xấp xỉ 65% với dữ liệu có nhiều ảnh không đạt tiêu chuẩn như bị bóng mờ, quá mờ, nhòe,... Với ảnh biển số sạch sẽ, rõ ràng, không chứa các phụ kiện gắn trên biển, tỉ lệ nhận dạng có thể đạt tới hơn 70%.

3.4. Kết luận

Demo nhận dạng với phương pháp trình bày trong đề án này đã đạt được một số điểm như:

- Nắm được các vấn đề cơ bản của ảnh số và xử lý ảnh số
- Sử dụng tương đối tốt thư viện EmguCV cho C# để xử lý ảnh.

- Nắm được các đặc điểm của một bài toán nhận dạng nói chung và nhận dạng biển số xe nói riêng
- Tìm hiểu được một số phương pháp nhận dạng biển số xe, nắm được ưu nhược điểm của từng phương pháp.
- Demo và Test thử thành công trên các bộ dữ liệu mẫu với tỉ lệ nhận dạng chính xác 99%.

Ưu điểm: thuật toán cài đặt nhanh, tìm vùng biển số và cách ly ký tự với tỉ lệ thành công cao (ở những biển số thông thường), tìm được ở những ảnh tự nhiên, vùng biển số bị nghiêng.

Bên cạnh đó, còn tồn tại một số điểm hạn chế chưa giải quyết được như:

- Tỉ lệ nhận dạng còn phụ thuộc nhiều vào điều kiện ánh sáng, phản chiếu hoặc che bóng.
- Với những biển số có đường viền phức tạp thì mức độ nhận dạng không cao. Chỉ giới hạn ảnh chụp trong góc 40 độ để nhận dạng được tốt, nếu góc lớn hơn thì khả năng nhận dạng được giảm.
- Ảnh xe trong bóng tối hay ban đêm không có đèn flash thì khả năng nhận dạng thấp do thiếu sáng.

3.5. Hướng phát triển của bài toán:

- Nâng cao hiệu quả chương trình, tách ly các kí tự trong biển số trong các trường hợp biển số bị nhiễu nhiều, mất mát thông tin do nhiễu từ điều kiện môi trường, tìm vùng biển số trong ảnh có độ tương phản giữa biển số và nền thấp. Đặc biệt là biển xe có nền màu đỏ chữ trắng.

- Phát triển chương trình thành module phần cứng. Có khả năng tương thích với các thiết bị quan sát như camera.

- Nghiên cứu theo hướng một ứng dụng cụ thể như : giám sát phương tiện giao thông, xử lý vi phạm giao thông, quản lý xe tại các bãi giữ xe, các kho vật tư....

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <http://neuralnetworksanddeeplearning.com/chap6.html>
- [2] <http://neuralnetworksanddeeplearning.com/chap2.html>
- [3].Y. LeCun and Y. Bengio.“Convolutional networks for images, speech, and time-series.” In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
- [4].Fabien Lauer, ChingY. Suen, Gérard Bloch,”A trainable feature extractor for handwritten digit recognition“,Elsevier, october 2006.
- [5].Patrice Y. Simard, Dave Steinkraus, John Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," International Conference on Document Analysis and Recognition (ICDAR), IEEE
- [6] Chirag N. Paunwala & Suprava Patnaik. “A Novel Multiple License Plate Extraction Techniquefor Complex Background in Indian Traffic Conditions”, Sarvajanic College of Engineering and Technology, 2010
- [7] Choo Kar Soon, Kueh Chiung Lin, Chung Ying Jeng and Shahrel A. Suandi , “Malaysian Car Number Plate Detection and Recognition System”, 2012
- [8] Tran Duc Duan, Duong Anh Duc, Tran Le Hong Du, “Combining Hough Transform and Contour Algorithm for detecting Vehicles License-Plates”, University of Natural Sciences, 2004
- [9] Nobuyuki Otsu, "A threshold selection method from gray-level histograms", 1979
- [10] Suman K. Mitra. “Recognition of Car License Plate using Morphology”, hirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India
- [11] Các tài liệu về EmguCV tại www.emgucv.com OPenCV tại www.opencv.com