BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

-----o0o----

TÊN ĐỀ TÀI:

Phân loại chữ số viết tay

|  |  |
| --- | --- |
| GVHD: | Trần Đình Toàn |
| Thành viên: | Nguyễn Hữu Thông-2001210224 |
|  | Nguyễn Hữu Đạt-2001215705 |
|  | Nguyễn Hữu Lý-2001215949 |

Mục lục

[Phần 1. Thuật toán CNN 3](#_Toc167735844)

[1.1. Khái niệm 3](#_Toc167735845)

[1.2. Các lớp cơ bản của thuật toán CNN 4](#_Toc167735846)

[1.2.1. Convolution Layer (Lớp tích chập) 4](#_Toc167735847)

[1.2.2. Stride (Bước nhảy) 5](#_Toc167735848)

[1.2.3. Padding (Đường viền) 5](#_Toc167735849)

[1.2.4. ReLU (Hàm phi tuyến) 5](#_Toc167735850)

[1.2.5. Pooling Layer (Lớp gộp) 6](#_Toc167735851)

[1.3. Cấu trúc mạng của thuật toán CNN 6](#_Toc167735852)

[1.4. Cách chọn tham số khi sử dụng thuật toán CNN 8](#_Toc167735853)

[Phần 2. Phân tích, trực quan hoá dữ liệu 9](#_Toc167735854)

[2.1. Thu thập 9](#_Toc167735855)

[2.2. Tiền xử lí dữ liệu 9](#_Toc167735856)

[2.3. Trực quan hoá dữ liệu 10](#_Toc167735857)

[Phần 3. Dùng thuật toán, kiểm tra và đánh giá model 14](#_Toc167735858)

[3.1. Dùng thuật toán 14](#_Toc167735859)

[3.1.1. Các bước áp dụng thuật toán 14](#_Toc167735860)

[3.2. Kiểm tra và đánh giá mô hình 18](#_Toc167735861)

[Kết luận 20](#_Toc167735862)

**Mục lục hình ảnh**

Hình 1 Thuật toán CNN 3

Hình 2 Các lớp cơ bản của thuật toán CNN 4

Hình 3 Các lớp cơ bản của thuật toán CNN 4

Hình 4 Các lớp cơ bản của thuật toán CNN 5

Hình 5 Các lớp cơ bản của thuật toán CNN 6

Hình 6 Các lớp cơ bản của thuật toán CNN 6

Hình 7 Cấu trúc mạng của thuật toán CNN 7

Hình 8 Chuyển đổi hình ảnh 9

Hình 9 Bảng phân phối dữ liệu 10

Hình 10 Phân bố dữ liệu 11

Hình 11 Minimum validation loss: 0.022137628868222237 11

Hình 12 Maximum validation accuracy: 0.9950000047683716 12

Hình 13 Sơ đồ khối mô hình mạng lenet-5(1998) 14

Hình 14 Tạo mô hình nơ ron tích chập 16

Hình 15 Chia dữ liệu và huấn luyện mô hình 17

Hình 16 Log huấn luyện của mô hình 18

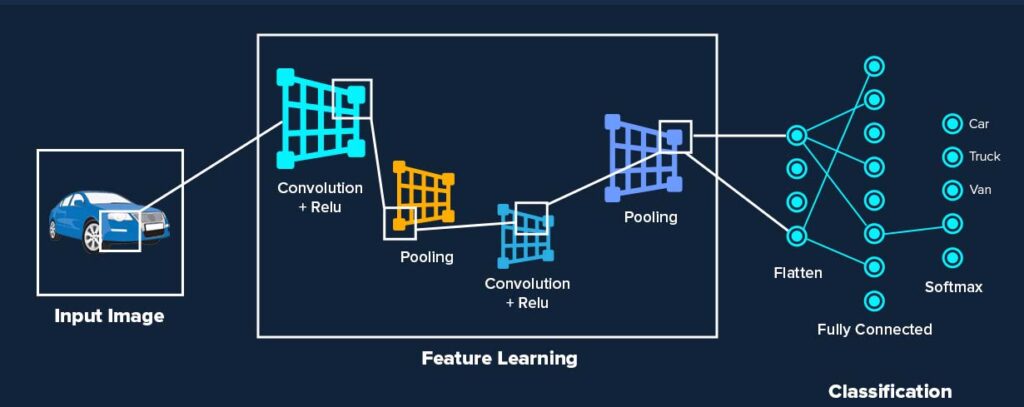
Hình 17 Hiệu suất của mô hình trên tập kiểm tra 19

Hình 18 Dự đoán mẫu 40 hình 19

# Thuật toán CNN

## Khái niệm

Convolutional Neural Network (CNN), hay còn gọi là ConvNet, là một loại mạng nơ-ron tích tụ được sử dụng phổ biến trong lĩnh vực xử lý hình ảnh. Thuật toán CNN là một phần quan trọng của Deep Learning – tập hợp các thuật toán nhằm xây dựng mô hình dữ liệu trừu tượng thông qua việc sử dụng nhiều lớp xử lý cấu trúc phức tạp.



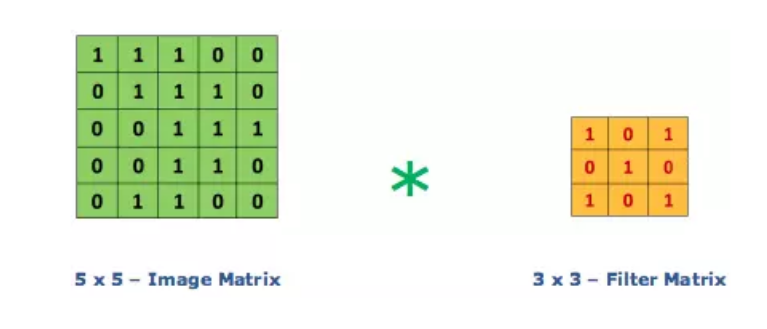
Hình Thuật toán CNN

CNN hoạt động dựa trên dữ liệu ảnh và được sử dụng rộng rãi trong các ứng dụng như nhận dạng hình ảnh và khuôn mặt. Thuật toán CNN hiện nay đã được tích hợp vào các nền tảng mạng phổ biến như Facebook và Google,… Khác với các mạng nơ-ron thông thường, thuật toán CNN nhận đầu vào dưới dạng một mảng hai chiều và xử lý trực tiếp trên hình ảnh, giúp nó trích xuất các đặc trưng quan trọng từ dữ liệu hình ảnh một cách hiệu quả.

Về phần kỹ thuật, thuật toán CNN là mô hình được sử dụng trong quá trình training và kiểm tra dữ liệu hình ảnh. Mỗi hình ảnh đầu vào sẽ đi qua một chuỗi các bước, bao gồm các lớp tích chập với các bộ lọc (Kernel), sau đó là lớp kết nối đầy đủ và cuối cùng áp dụng hàm Softmax để phân loại đối tượng. Kết quả phân loại sẽ là một giá trị xác suất nằm trong khoảng từ 0 đến 1. Dưới đây là mô tả tổng quan về quy trình CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng.

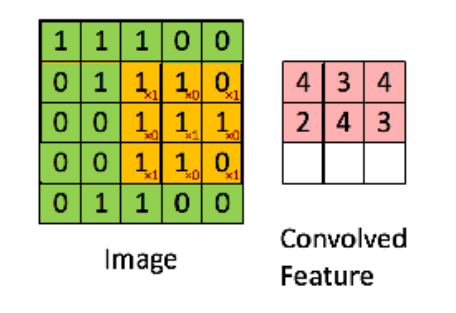
## Các lớp cơ bản của thuật toán CNN

### Convolution Layer (Lớp tích chập)

Tích chập là lớp đầu tiên để lấy ra các đặc điểm từ hình ảnh ban đầu. Nó giữ lại các mối quan hệ giữa các pixel bằng cách xem xét các tính năng của hình ảnh thông qua việc sử dụng các ô vuông nhỏ từ dữ liệu gốc. Đây là một phép tính với hai thành phần đầu vào, là ma trận hình ảnh ban đầu và một bộ lọc hoặc hạt nhân.

Hình 2 Các lớp cơ bản của thuật toán CNN

Lớp Convolution (tích chập) của ma trận hình ảnh 5 x 5 sẽ nhân với bộ lọc 3 x 3 được gọi là Feature Map.

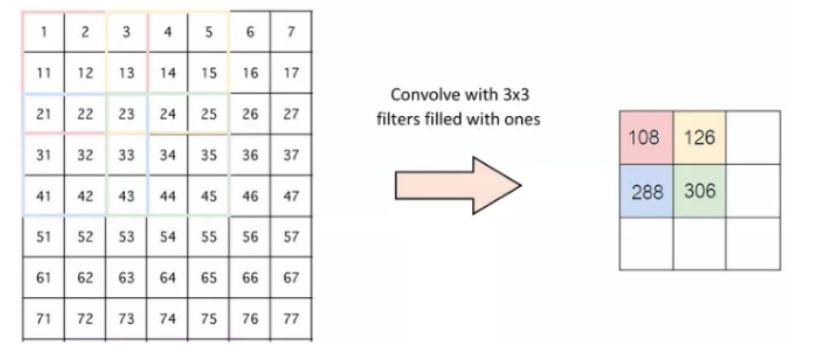


Hình 3 Các lớp cơ bản của thuật toán CNN

Với sự kết hợp giữa một hình ảnh và nhiều bộ lọc thì hình ảnh có thể được làm mờ hoặc sắc nét hay phát hiện cạnh. Hình ảnh tích chập khác nhau khi áp dụng những kernel khác nhau.

### Stride (Bước nhảy)

Stride chính là số pixel luôn thay đổi trên ma trận đầu vào. Khi stride là 1, ta di chuyển kernel 1 pixel, nếu stride là 2 thì di chuyển kernel 2 pixel,…



Hình 4 Các lớp cơ bản của thuật toán CNN

### Padding (Đường viền)

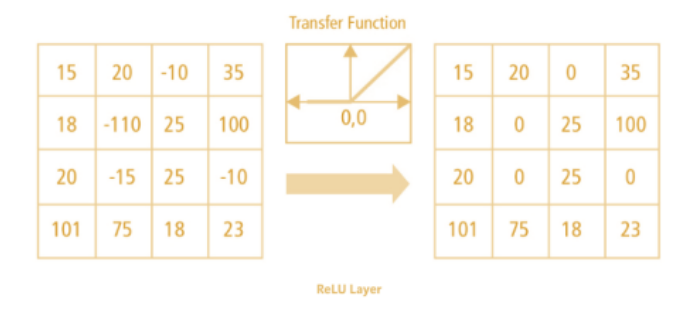
Trong trường hợp kernel không tương xứng với hình ảnh đầu vào thì có thể lựa chọn 1 trong 2 cách sau:

* Chèn thêm số 0 vào 4 đường biên hình ảnh
* Cắt bớt những điểm không phù hợp với kernel

### ReLU (Hàm phi tuyến)

Rectified Linear Unit hay ReLU là một hàm phi tuyến. Đầu ra là: f(x) = max (0,x). ReLU có thể giới thiệu tính phi tuyến trong ConvNet. Dữ liệu mà chúng ta tìm hiểu thường là các giá trị tuyến tính không âm.

Một số hà phi tuyến khác có thể kể đến như: sigmoid, tanh cũng có thể thay thế cho ReLU nhưng ReLU thường được ưu tiên sử dụng bởi hiệu suất tốt.

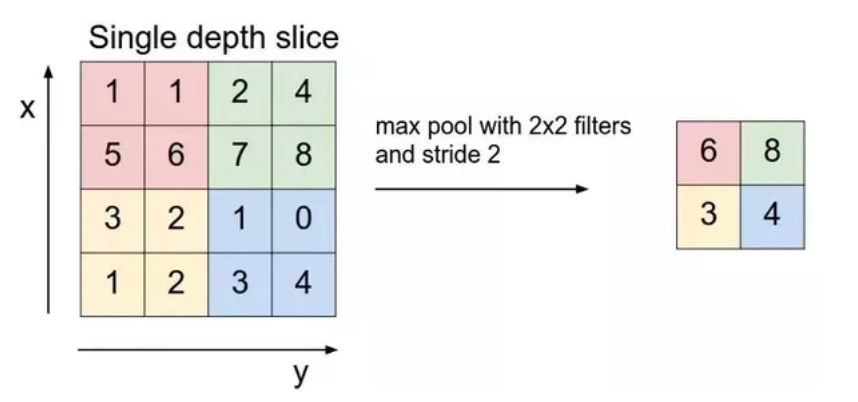


Hình 5 Các lớp cơ bản của thuật toán CNN

### Pooling Layer (Lớp gộp)

Khi hình ảnh quá lớn thì lớp Pooling sẽ giúp giảm bớt số lượng tham số. Không gian Pooling lấy mẫu con giúp giảm kích thước map nhưng vẫn giữ lại được những thông tin quan trọng. Có nhiều loại Pooling khác nhau như: Sum Pooling, Max Pooling và Average Pooling.

Trong đó Max Pooling thường lấy tổng trung bình hoặc lấy phần tử lớn nhất từ ma trận. Tổng tất cả phần tử trong đối tượng map được gọi là Sum Pooling.

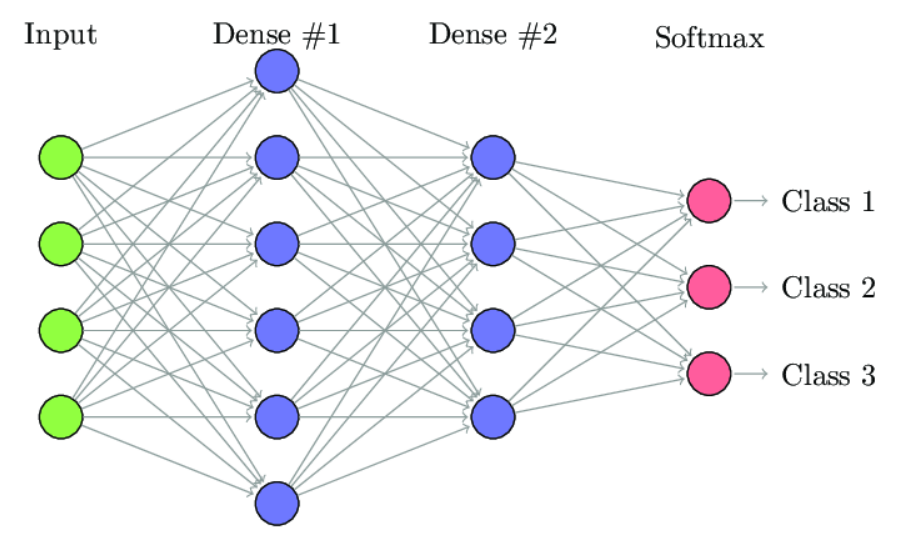


Hình 6 Các lớp cơ bản của thuật toán CNN

## Cấu trúc mạng của thuật toán CNN

Mạng CNN là một trong các lớp Convolution chồng lên nhau. Nó sử dụng các hàm kích hoạt phi tuyến (như ReLU và tanh) để tạo ra thông tin trừu tượng và kích hoạt trọng số trong node. Các lớp này cung cấp tính bất biến và tính kết hợp, nhằm giúp mạng hiểu được đối tượng theo nhiều góc độ khác nhau.

Pooling layer giúp làm cho mạng không nhạy cảm đối với việc dịch chuyển, co giãn và quay hình ảnh. Còn tính kết hợp cục bộ là kết quả của các lớp Convolution liên kết với nhau. Mỗi lớp tiếp theo dựa trên kết quả của lớp Convolution trước đó và đảm bảo kết nối cục bộ hiệu quả nhất. Ngoài ra, có các lớp khác như lớp pooling/subsampling, được sử dụng để lọc thông tin quan trọng và loại bỏ thông tin nhiễu.



Hình 7 Cấu trúc mạng của thuật toán CNN

Trong quá trình huấn luyện, thuật toán CNN sẽ tự động học cách nhận biết các đặc điểm thông qua việc sử dụng các bộ lọc. Quá trình này tương tự việc bộ não con người nhận diện các đối tượng trong thế giới thực.

Thuật toán CNN gồm 3 phần chính, đó là:

* Local receptive field (trường cục bộ): Trường cục bộ có nhiệm vụ chia dữ liệu thành các phần nhỏ để xác định vùng quan trọng và tạo ra các vùng ảnh có giá trị cao nhất.
* Shared weights and bias (trọng số và bias chia sẻ): Trong mạng CNN, trọng số và bias chia sẻ giúp giảm thiểu số lượng tham số cần sử dụng. Mỗi lớp convolution bao gồm nhiều feature map khác nhau, và mỗi feature map có khả năng nhận biết một số đặc điểm trong ảnh.
* Pooling layer (lớp tổng hợp): Lớp tổng hợp là bước cuối cùng trong quá trình tính toán và quét qua các lớp. Nhiệm vụ của nó là đơn giản hóa dữ liệu đầu ra, loại bỏ thông tin không cần thiết và tối ưu hóa kết quả cuối cùng. Điều này giúp đảm bảo rằng người dùng nhận được kết quả đáng tin cậy và phù hợp với yêu cầu của họ.

## Cách chọn tham số khi sử dụng thuật toán CNN

Để điều chỉnh tham số cho CNN một cách tốt nhất, bạn nên xem xét các yếu tố sau:

* Convolution layer (lớp tích chập): Số lượng lớp tích chập ảnh. Thông thường, sử dụng từ 3 đến 5 lớp có thể mang lại kết quả mong muốn. Số lượng lớp nhiều hơn có thể cải thiện hiệu suất, nhưng cần cân nhắc để tránh tốn nhiều tài nguyên.
* Filter size (kích thước filter): Filter thường có kích thước 3×3 hoặc 5×5. Kích thước này có thể được điều chỉnh để phù hợp với bộ dữ liệu cụ thể.
* Pooling size (kích thước pooling): Kích thước pooling thường là 2×2 cho hình ảnh tiêu chuẩn. Đối với hình ảnh lớn hơn, bạn có thể sử dụng kích thước 4×4.
* Số lần train và test: Thực hiện nhiều lần train và test để tối ưu hóa các tham số. Điều này giúp mô hình trở nên “thông minh” và hiệu quả hơn.

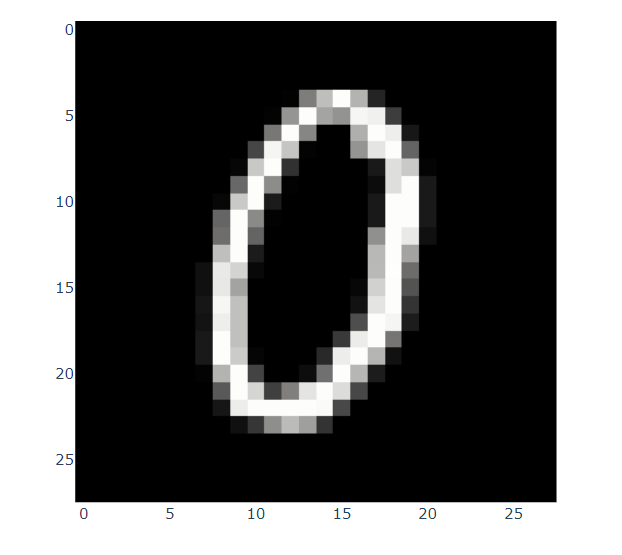
# Phân tích, trực quan hoá dữ liệu

## Thu thập

Sử dụng bộ dữ liệu tiêu chuẩn của kaggle. Thật ngạc nhiên khi mạng lưới thần kinh tích chập có thể phân loại các chữ số viết tay một cách chính xác đến vậy. Trong hạt nhân này, chúng ta chứng kiến một nhóm gồm 15 CNN phân loại các chữ số MNIST của Kaggle sau khi huấn luyện trên 42.000 hình ảnh của Kaggle trong "train.csv" cùng với 25 triệu hình ảnh khác được tạo bằng cách xoay, chia tỷ lệ và dịch chuyển hình ảnh của Kaggle. Học từ 25.042.000 hình ảnh, nhóm CNN này đạt được độ chính xác phân loại 99,75%. Hạt nhân này sử dụng ý tưởng từ các mô hình được xuất bản tốt nhất được tìm thấy trên internet. Các kỹ thuật nâng cao bao gồm tăng cường dữ liệu, các lớp tích chập phi tuyến tính, các lớp tổng hợp có thể học được, kích hoạt ReLU, tập hợp, đóng gói, tốc độ học tập phân rã, bỏ học, chuẩn hóa hàng loạt và tối ưu hóa adam.

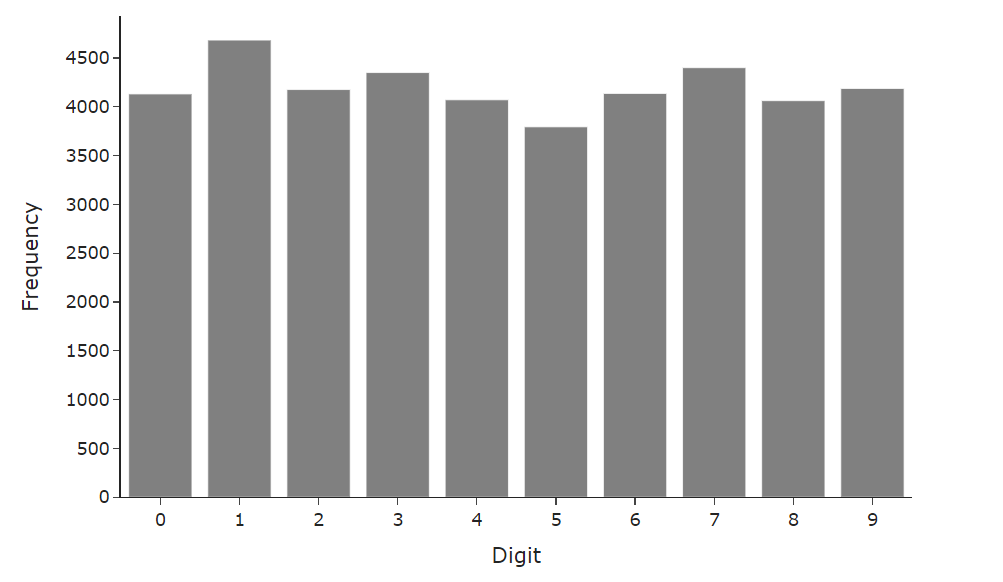
## Tiền xử lí dữ liệu

Chuyển đổi hình ảnh sang dạng thang xám (grayscale) và chuẩn hóa kích thước của hình ảnh (28x28 pixel)



Hình 8 Chuyển đổi hình ảnh

Bảng phân phối dữ liệu

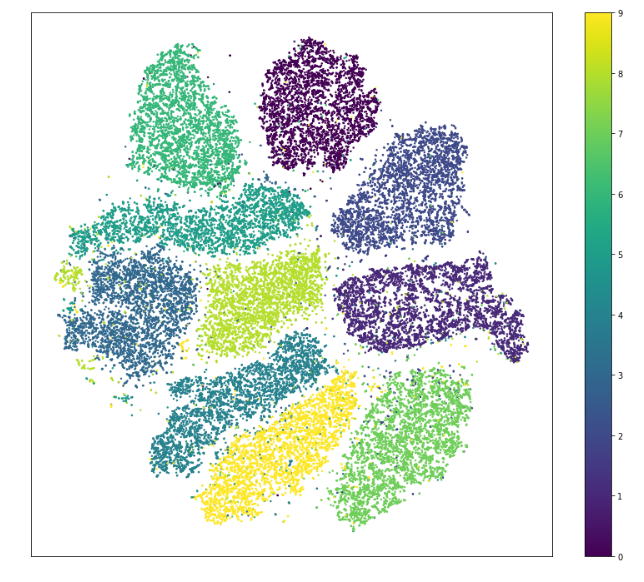


Hình 9 Bảng phân phối dữ liệu

## Trực quan hoá dữ liệu

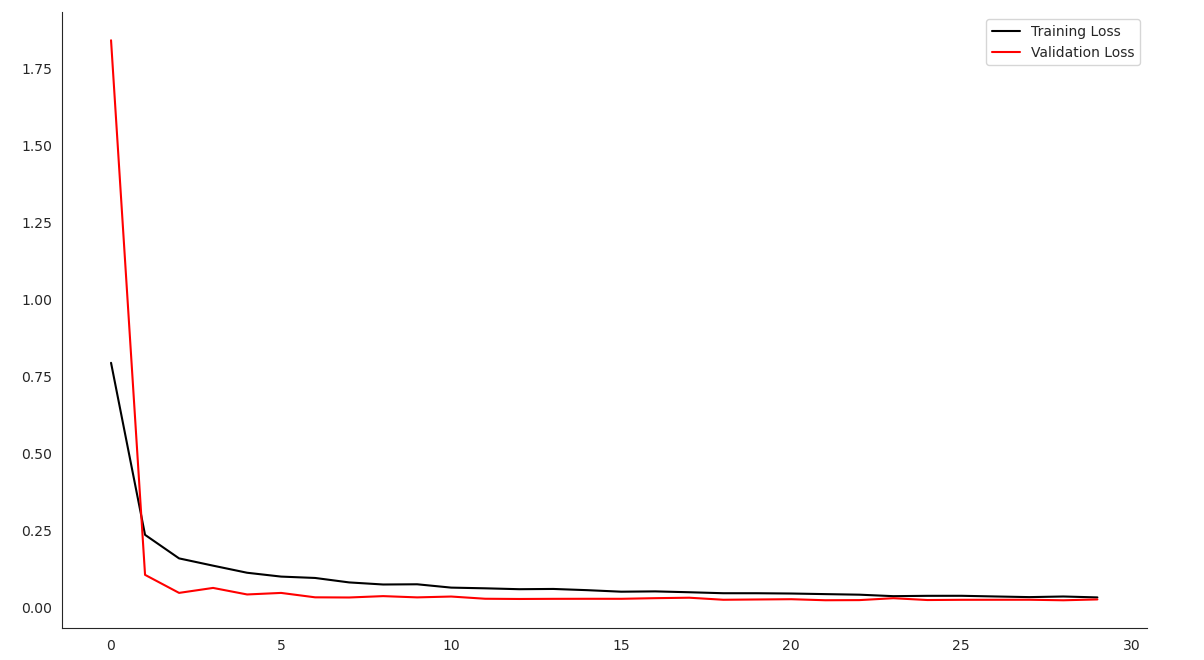
Bộ dữ liệu MNIST chứa 42.000 hình ảnh đào tạo về các chữ số viết tay từ 0 đến 9 (10 lớp khác nhau) và 28.000 hình ảnh để thử nghiệm mà không cần mã định danh nhãn (để gửi). Những hình ảnh này là giá trị pixel thang độ xám và được biểu diễn dưới dạng ma trận 28 × 28.

TSNE - nhúng hàng xóm ngẫu nhiên phân phối t. Đây là thuật toán giảm kích thước được thiết kế để giữ cấu trúc cục bộ trong tập dữ liệu nhiều chiều nhưng ít quan tâm đến cấu trúc toàn cục. Ở đây, chúng tôi sử dụng nó để chuyển từ kích thước 784 pixels của hình ảnh sang hai chiều. Điều này làm cho việc vẽ đồ thị dễ dàng hơn. Thang màu là nhãn MNIST ban đầu và người ta có thể thấy rằng sự tách biệt của các nhãn là rõ ràng.



Hình 10 Phân bố dữ liệu

Trong quá trình huấn luyện mô hình, validation loss thường được theo dõi để đảm bảo mô hình không bị overfitting (học quá khớp) trên tập huấn luyện. Giá trị "Minimum Validation Loss" là giá trị validation loss thấp nhất mà mô hình đạt được trong suốt quá trình huấn luyện.

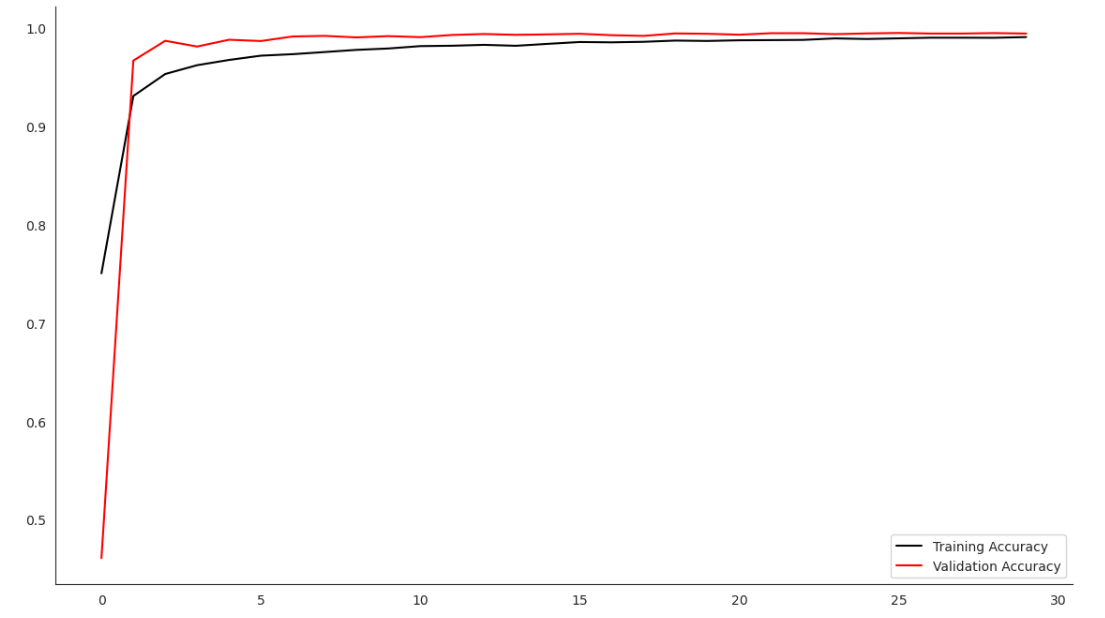


Hình 11 Minimum validation loss: 0.022137628868222237

Giá trị "0.022137628868222237" có nghĩa là mô hình đạt được validation loss tối thiểu là khoảng 0.022, tức là chỉ mắc khoảng 2.2% lỗi trên tập validation. Đây là một kết quả rất tốt, cho thấy mô hình có khả năng dự đoán chính xác cao trên dữ liệu mới.

Nói chung, việc đạt được validation loss thấp như vậy là một dấu hiệu tích cực, cho thấy mô hình đã được huấn luyện tốt và có khả năng generalize (khái quát hóa) tốt trên dữ liệu mới. Đây là một chỉ số quan trọng để đánh giá hiệu suất của mô hình máy học.

Maximum Validation Accuracy: Trong quá trình huấn luyện mô hình, accuracy trên tập validation thường được theo dõi. Giá trị "Maximum Validation Accuracy" là giá trị accuracy cao nhất mà mô hình đạt được trên tập validation trong suốt quá trình huấn luyện.



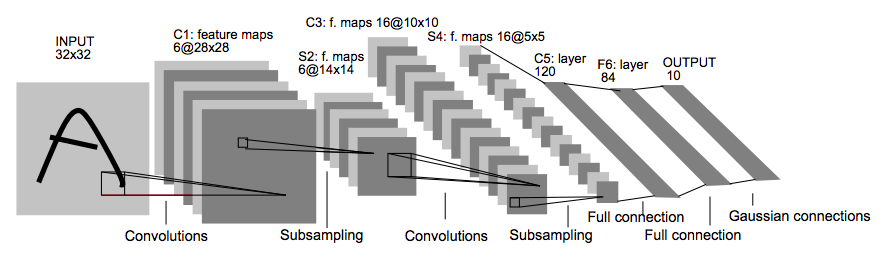
Hình 12 Maximum validation accuracy: 0.9950000047683716

Giá trị "0.9950000047683716" có nghĩa là mô hình đạt được accuracy tối đa trên tập validation là khoảng 99.5%. Điều này cho thấy mô hình có khả năng dự đoán chính xác 99.5% các mẫu dữ liệu trong tập validation.

Một accuracy như vậy là rất cao, cho thấy mô hình đã được huấn luyện rất tốt và có khả năng phân loại các mẫu dữ liệu mới với độ chính xác gần như hoàn hảo. Đây là một kết quả rất tích cực, thường chỉ ra rằng mô hình đã được tối ưu hóa và có khả năng generalize tốt trên dữ liệu mới.

# Dùng thuật toán, kiểm tra và đánh giá model

## Dùng thuật toán

các điểm nổi bật về kiến trúc:

Hình 13 Sơ đồ khối mô hình mạng lenet-5(1998)

Các CNN trong kernel này tuân theo thiết kế của LeNet-5 (hình minh họa ở trên) với các cải tiến sau:

* Hai bộ lọc 3x3 được xếp chồng thay thế cho bộ lọc 5x5 đơn. Chúng trở thành các tích chập 5x5 không tuyến tính
* Tích chập với bước 2 thay thế cho các lớp pooling. Chúng trở thành các lớp pooling có thể học được.
* Hàm kích hoạt ReLU thay thế sigmoid.
* Thêm chuẩn hóa theo batch
* Thêm Dropout Thêm nhiều map đặc trưng (kênh)
* Sử dụng một tập hợp gồm 15 CNN với phương pháp bagging

### Các bước áp dụng thuật toán

1. Tăng dữ liệu (Data Augmentation):

* Sử dụng các kỹ thuật tăng dữ liệu như xoay, dịch chuyển, phóng to, thu nhỏ, làm méo ảnh để tạo ra 25 triệu ảnh mới từ dữ liệu MNIST gốc.
* Việc tăng dữ liệu giúp mô hình học được các đặc trưng bền vững hơn và tránh tình trạng quá khớp (overfitting) khi huấn luyện.

1. Kiến trúc mạng nơ-ron tích chập (Convolutional Neural Network): Mô hình sử dụng một kiến trúc mạng nơ-ron tích chập (CNN) với các lớp sau:

* Các lớp tích chập (Convolutional layers): Học các đặc trưng cục bộ từ ảnh đầu vào.
* Các lớp pooling: Giảm kích thước của đặc trưng, tăng tính không gian bất biến.
* Các lớp kết nối đầy đủ (Fully connected layers): Kết hợp các đặc trưng để thực hiện phân loại.
* Kiến trúc CNN phù hợp với bài toán phân loại ảnh vì nó có thể học được các đặc trưng cục bộ và không gian bất biến.

1. Huấn luyện mô hình:

Mô hình được huấn luyện trên bộ dữ liệu 25 triệu ảnh sử dụng thuật toán SGD (Stochastic Gradient Descent) với các siêu tham số như tốc độ học, kích thước batch, ... Quá trình huấn luyện được thực hiện trên các GPU để tăng tốc độ tính toán.

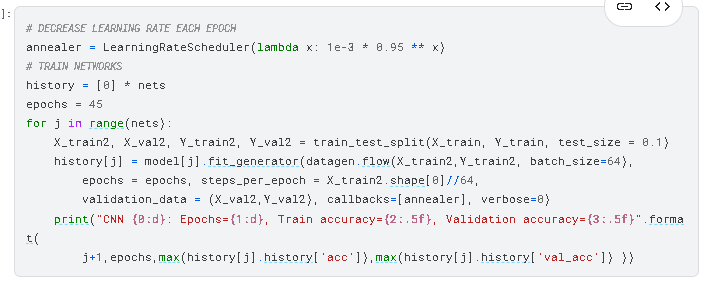
tạo 15 mô hình mạng nơ-ron tích chập (Convolutional Neural Network - CNN) sử dụng thư viện Keras.

Hình Tạo mô hình nơ ron tích chập

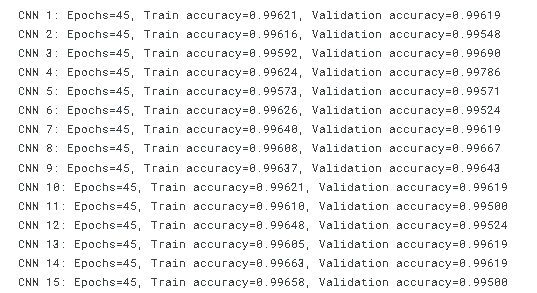
1. Đầu tiên, script khởi tạo một danh sách 15 mô hình Keras Sequential rỗng, sẽ được sử dụng để tạo các mô hình CNN.
2. Với mỗi một trong 15 mô hình, script thêm các lớp sau:

* Conv2D: Đây là lớp tích chập 2D với 32 bộ lọc, kích thước lõi nhân 3x3 và hàm kích hoạt ReLU. Kích thước đầu vào là (28, 28, 1), có thể là kích thước của ảnh đầu vào.
* BatchNormalization: Lớp chuẩn hóa theo batch, giúp ổn định và cải thiện hiệu suất của mô hình.
* Các lớp Conv2D, BatchNormalization, Conv2D, BatchNormalization, Conv2D, BatchNormalization và Dropout tiếp theo với các cấu hình khác nhau về số lượng bộ lọc, kích thước lõi nhân, stride và tỷ lệ dropout.
* Flatten: Lớp làm phẳng các feature map 3D thành vector 1D.
* Dropout: Thêm một lớp dropout khác với tỷ lệ 0.4.
* Dense: Lớp kết nối đầy đủ với 10 nút và hàm kích hoạt softmax, dùng cho bài toán phân loại 10 lớp.

1. Cuối cùng, script biên dịch mỗi một trong 15 mô hình với optimizer Adam, hàm mất mát entropy chéo và metric độ chính xác.

huấn luyện 15 mô hình CNN sử dụng tập dữ liệu được chia thành tập huấn luyện và tập kiểm tra, với tốc độ học giảm dần theo số lượng epoch tăng lên. Kết quả huấn luyện của mỗi mô hình được lưu vào danh sách history và in ra thông tin về độ chính xác trên tập huấn luyện và tập kiểm tra.

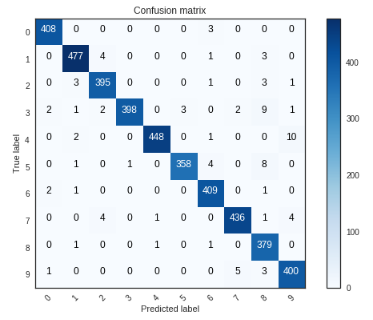
Hình Chia dữ liệu và huấn luyện mô hình



Hình Log huấn luyện của mô hình

## Kiểm tra và đánh giá mô hình

Kết quả dự đoán của mô hình CNN. Điều này giúp trực quan hóa và đánh giá được hiệu suất của mô hình.



Hình Hiệu suất của mô hình trên tập kiểm tra

Hình Dự đoán mẫu 40 hình

# Kết luận

Trong bài báo cáo này, chúng em đã trình bày một mô hình CNN để dự đoán chữ số viết tay từ tập dữ liệu MNIST. Mô hình được thiết kế gồm các lớp convolution, pooling và fully connected, sử dụng các kỹ thuật regularization như Dropout để tránh overfitting. Qua quá trình huấn luyện và đánh giá, mô hình đạt được độ chính xác lên đến 99,5% trên tập kiểm tra, cho thấy khả năng phân loại chữ số viết tay của mô hình CNN rất tốt. Ưu điểm nổi bật của mô hình CNN là khả năng tự động học các đặc trưng từ dữ liệu đầu vào mà không cần phải thiết kế các tính năng thủ công. Mô hình có thể học và phát hiện các mẫu cục bộ như vết nét, cạnh góc... từ ảnh input, từ đó phân loại chính xác chữ số. Kết hợp với việc sử dụng các kỹ thuật regularization hiệu quả, mô hình CNN đã đạt được hiệu suất cao trên tập kiểm tra, chứng tỏ tính hiệu quả và robustness của kiến trúc này trong bài toán dự đoán chữ số viết tay. Tuy nhiên, một hạn chế của mô hình là độ phức tạp tính toán tương đối cao do cấu trúc sâu và số lượng tham số lớn. Điều này có thể ảnh hưởng đến thời gian huấn luyện và triển khai mô hình trên những môi trường tài nguyên hạn chế. Vì vậy, việc tối ưu hóa mô hình hoặc thiết kế các kiến trúc CNN hiệu quả hơn vẫn là hướng nghiên cứu đáng quan tâm trong tương lai.