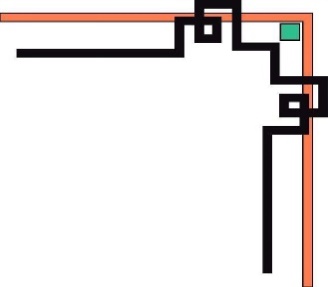
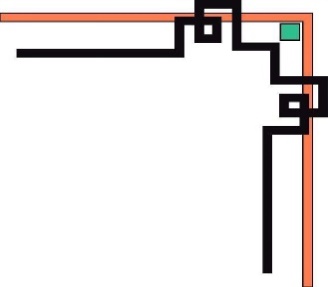
TRƯỜNG ĐẠI HỌC QUY NHƠN  
KHOA CÔNG NGHỆ THÔNG TIN



-------------o0o------------



BÁO CÁO TIỂU LUẬN

HỌC PHẦN HỌC MÁY VÀ ỨNG DỤNG

PHÂN LOẠI BIỂN BÁO GIAO THÔNG BẰNG MẠNG CNN

|  |  |
| --- | --- |
| Sinh viên thực hiện: | Lê Thị Minh Tâm |
| Lớp: | Khoa học máy tính K41 |
| Giảng viên hướng dẫn | TS. Lê Xuân Vinh |

*Quy Nhơn, tháng 12 năm 2021*

MỤC LỤC

[I. Cơ sở lý thuyết 3](#_Toc90187493)

[1. Giới thiệu 3](#_Toc90187494)

[2. Cấu trúc CNN 3](#_Toc90187495)

[3. Những phép toán trong lớp ẩn 4](#_Toc90187496)

[4. Fully connected layer 8](#_Toc90187497)

[II. Thực hành 8](#_Toc90187498)

[1. Phân tích bài toán 8](#_Toc90187499)

[2. Mô hình huấn luyện 9](#_Toc90187500)

[3. Kết quả 10](#_Toc90187501)

[III. Kết luận 10](#_Toc90187502)

[IV. Tài liệu tham khảo 10](#_Toc90187503)

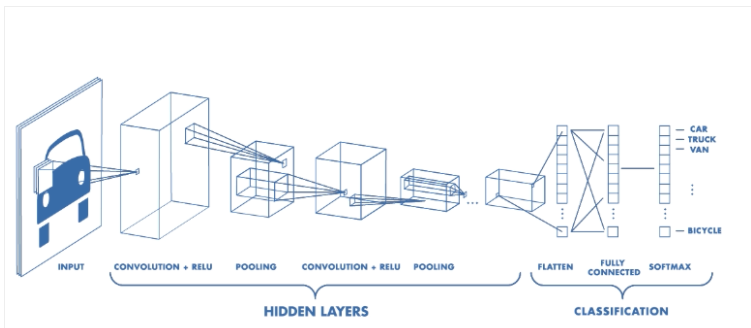
1. Cơ sở lý thuyết
2. Giới thiệu

Convolutional Neural Networks (CNN) là một trong những mô hình deep learning phổ biến nhất và có ảnh hưởng nhiều nhất trong cộng đồng Computer Vision. CNN được dùng trong trong nhiều bài toán như nhận dạng ảnh, phân tích video, ảnh MRI, hoặc cho các bài toán của lĩnh vự xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này.

CNN cũng có lịch sử khá lâu đời. Kiến trúc gốc của mô hình CNN được giới thiệu bởi một nhà khoa học máy tính người Nhật vào năm 1980. Sau đó, năm 1998, Yan LeCun lần đầu huấn luyện mô hình CNN với thuật toán backpropagation cho bài toán nhận dạng chữ viết tay. Tuy nhiên, mãi đến năm 2012, khi một nhà khoa học máy tính người Ukraine Alex Krizhevsky xây dựng mô hình CNN (AlexNet) và sử dụng GPU để tăng tốc quá trình huấn luyện deep nets để đạt được top 1 trong cuộc thi Computer Vision thường niên ImageNet với độ lỗi phân lớp top 5 giảm hơn 10% so với những mô hình truyền thống trước đó, đã tạo nên làn sóng mãnh mẽ về việc sử dụng deep CNN với sự hỗ trợ của GPU để giải quyết càng nhiều các vấn đề trong Computer Vision.

1. Cấu trúc CNN

CNN bao gồm tập hợp các lớp cơ bản bao gồm: convolution layer + nonlinear layer, pooling layer, fully connected layer. Các lớp này liên kết với nhau theo một thứ tự nhất định. Thông thường, một ảnh sẽ được lan truyền qua tầng convolution layer + nonlinear layer đầu tiên, sau đó các giá trị tính toán được sẽ lan truyền qua pooling layer, bộ ba convolution layer + nonlinear layer + pooling layer có thể được lặp lại nhiều lần trong network. Và sau đó được lan truyền qua tầng fully connected layer và softmax để tính sác xuất ảnh đó chứa vật thế gì.



Hình 1: Cấu trúc mạng CNN

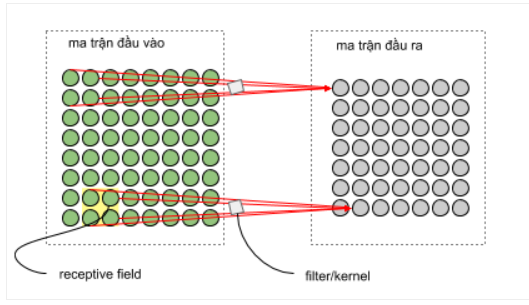
1. Những phép toán trong lớp ẩn

Lớp ẩn là lớp rút trích đặc trưng: trong phần này, mạng sẽ tiến hành tính toán hàng loạt phép tích chập và phép hợp nhất (pooling) để phát hiện các đặc trưng.

1. Convolution layer

Convolution layer là lớp quan trọng nhất và cũng là lớp đầu tiên của của mô hình CNN. Lớp này có chức năng chính là phát hiện các đặc trưng có tính không gian hiệu quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, bộ **filters**, và **receptive field**, **feature map**. Conv layer nhận đầu vào là một ma trận 3 chiều và một bộ filters cần phải học. Bộ filters này sẽ trượt qua từng vị trí trên bức ảnh để tính tích chập (convolution) giữa bộ filter và phần tương ứng trên bức ảnh. Phần tưng ứng này trên bức ảnh gọi là receptive field, tức là vùng mà một neuron có thể nhìn thấy để đưa ra quyết định, và mà trận cho ra bới quá trình này được gọi là feature map.

Với ví dụ ở bên dưới, dữ liệu đầu vào là ma trận có kích thước 8x8x1, một bộ filter có kích thước 2x2x1, feature map có kích thước 7x7x1. Mỗi giá trị ở feature map được tính bằng tổng của tích các phần tử tương ứng của bộ filter 2x2x1 với receptive field trên ảnh. Để tính tất cả các giá trị cho feature map, chúng ta cần trượt filter từ trái sáng phải, từ trên xuống dưới. Do đó, ta có thể thấy rằng phép convolution bảo toàn thứ tự không gian của các điểm ảnh. ví dụ điểm góc trái của dữ liệu đầu vào sẽ tương ứng với một điểm bên góc trái của feature map.



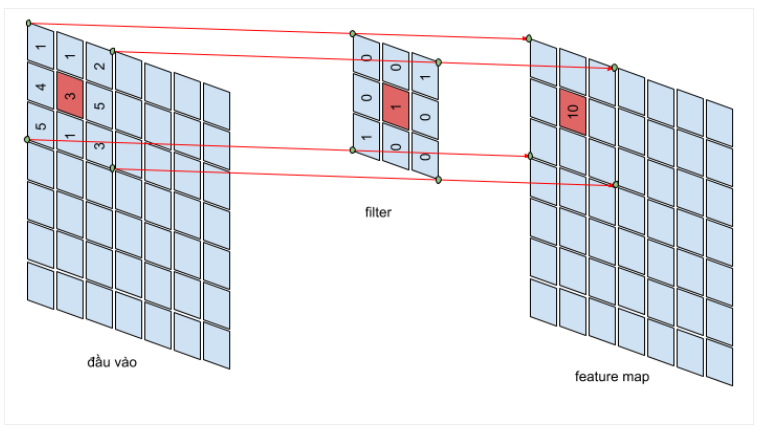
Tầng convolution có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh. Những đặc trưng này bao gồm đặc trưng cơ bản là góc, cạnh, màu sắc, hoặc đặc trưng phức tạp hơn như texture của ảnh. Vì bộ filter quét qua toàn bộ bức ảnh, nên những đặc trưng này có thể nằm ở vị trí bất kì trong bức ảnh, cho dù ảnh bị xoay trái/phải thì những đặc trưng này vẫn bị phát hiện.

Kích thước bộ filter là một trong những tham số quan trọng nhất của tầng convolution. Kích thước này tỉ lệ thuận với số tham số cần học tại mỗi tầng convolution và là tham số quyết định receptive field của tầng này. Kích thước phổ biến nhất của bộ filter là 3x3.

Kích thước filter nhỏ được ưu tiên lựa chọn thay kích thước lớn vì những lý do sau đây.



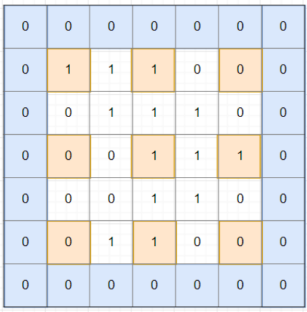
Kích thước filter của tầng convolution hầu hết đều là số lẻ, ví dụ như 3x3 hay 5x5. Với kích thước filter lẻ, các giá trị của feature map sẽ xác định một tâm điểm ở tầng phía trước. Nếu các bạn chọn filter có kích thước 2x2, 4x4 thì chúng ta sẽ gặp khó khăn khi muốn tìm vị trí tương ứng của các giá trị feature map trên không gian ảnh.



Ở những trường hợp đặt biệt như filter có kích thước 1x1, hay có kích thước bằng với ma trận đầu vào, tầng convolution có ý nghĩa rất thú vị. Khi có kích thước 1x1, tầng convolution xem mỗi điểm như một đặc trưng riêng biệt, có chức năng giảm chiều (tăng chiều) khi số lượt feature map ở tầng sau nhỏ hơn (lớn hơn) tầng trước. Filter 1x1 đã được sử dụng trong kiến trúc mạng phổ biến như Inception networks. Trong khi đó, filter với kích thước bằng ảnh đầu vào, tầng convolution có chức năng y hệt fully connected layer.

1. Stride

Stride là bước nhảy bỏ qua một số điểm trên ảnh khi tính tích chập nhằm giảm kích thước của ma trận Y. Ví dụ stride = 2.



Nếu stride = s, padding=1, Kernel = k thì ảnh m x n sau khi convolution có kích thước ((m - k + 2p)/s +1) x ((n - k + 2p)/s + 1)

1. Padding

Khi áp dụng phép convolution thì ma trận đầu vào sẽ nhỏ dần đi, do đó số layer của mô hình CNN sẽ bị giới hạn, và không thể xây dựng deep nets mong muốn. Để giải quyết tình trạng này, ta cần padding ma trận đầu vào để đảm bảo kích thước đầu ra sau mỗi tầng convolution là không đổi. Do đó có thể xây dựng được mô hình với số tầng convolution lớn tùy ý. Một cách đơn giản và phổ biến nhất để padding là sử dụng hàng số 0, ngoài ra cũng có thể sử dụng reflection padding hay là symmetric padding.



1. Nonlinear layer

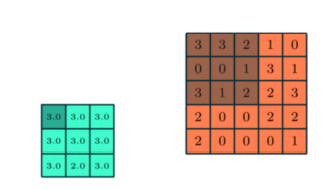
ReLU (Rectified Linear Units, f = max (0, x)) là hàm kích hoạt phổ biến nhất cho CNN tại thời điểm hiện tại, được giới thiệu bởi Geoffrey E. Hinton năm 2010. Trước khi hàm ReLU được áp dụng thì những hàm như sigmoid hay tanh mới là những hàm được sử dụng phổ biến. Hàm ReLU được ưa chuộng vì tính toán đơn giản, giúp hạn chế tình trạng vanishing gradient, và cũng cho kết quả tốt hơn. ReLU cũng như những hàm kích hoạt khác, được đặt ngay sau tầng convolution, ReLU sẽ gán những giá trị âm bằng 0 và giữ nguyên giá trị của đầu vào khi lớn hơn 0.

Relu cũng có một số vấn đề tiềm ẩn như: không có đạo hàm tại điểm 0, giá trị của hàm ReLU có thể lớn đến vô cùng và nếu chúng ta không khởi tạo trọng số cẩn thận, hoặc khởi tạo learning rate quá lớn thì những neuron ở tầng này sẽ rơi vào trạng thái chết, tức là luôn có giá trị < 0.

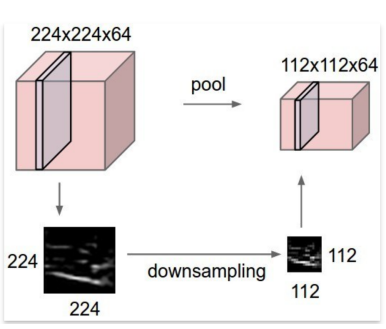
1. Pooling

Pooling là thao tác để giảm kích thước ảnh nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model.

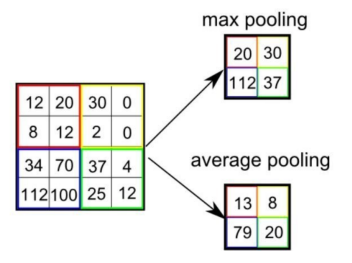
Cách làm rất đơn giản. Quy định trước kích thước pooling là K\*K. Ta trượt khung K\*K này lên ma trận ảnh sau khi convolution lấy max hoặc average các giá trị trong khung đó.



Thường dùng pooling với size =2 x 2, stride=2, padding=0. Khi đó kích thước dữ liệu giảm đi một nửa. Depth thì giữ nguyên do pooling xử lý riêng trên từng mảng 2 chiều.



Một số loại pooling layer phổ biến như là max-pooling, average pooling, với chức năng chính là giảm chiều của tầng trước đó. Với một pooling có kích thước 2x2 ta cần phải trược filter 2x2 này trên những vùng ảnh có kích thước tương tự rồi sau đó tính max, hay average cho vùng ảnh đó.



1. Fully connected layer

Tầng cuối cùng của mô hình CNN trong bài toán phân loại ảnh là tầng fully connected layer. Tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán. Ví dụ, trong bài toán phân loại số viết tay MNIST có 10 lớp tương ứng 10 số từ 0-1, tầng fully connected layer sẽ chuyển ma trận đặc trưng của tầng trước thành vector có 10 chiều thể hiện xác suất của 10 lớp tương ứng.

Cuối cùng, quá trình huấn luyện mô hình CNN cho bài toán phân loại ảnh cũng tương tự như huấn luyện các mô hình khác. Chúng ta cần có hàm mất mát để tính sai số giữa giá trị dự đoán của mô hình và nhãn chính xác, cũng như sử dụng thuật toán backpropagation cho quá trình cập nhật trọng số.

1. Thực hành

Phân loại biển báo giao thông sử dụng mô hình CNN

1. Phân tích bài toán

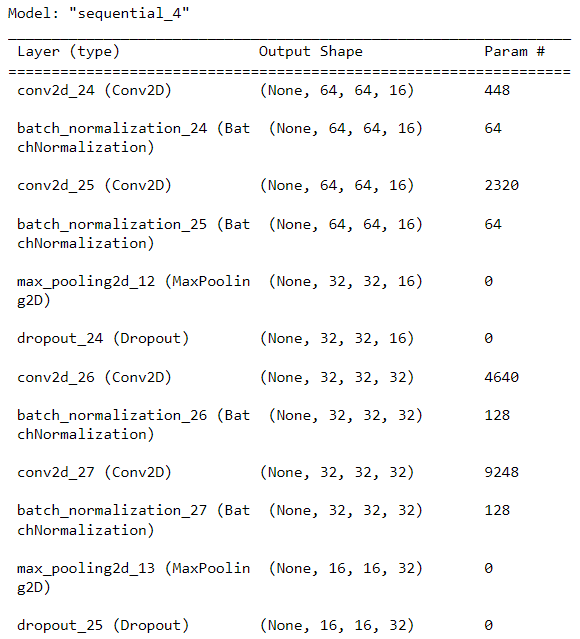
Input: 1 ảnh biển báo giao thông

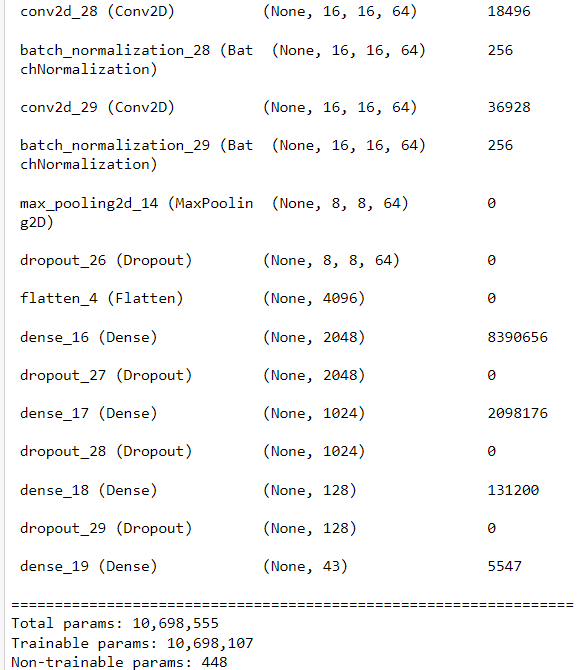
Output: Ảnh đó là biển báo gì

Data:

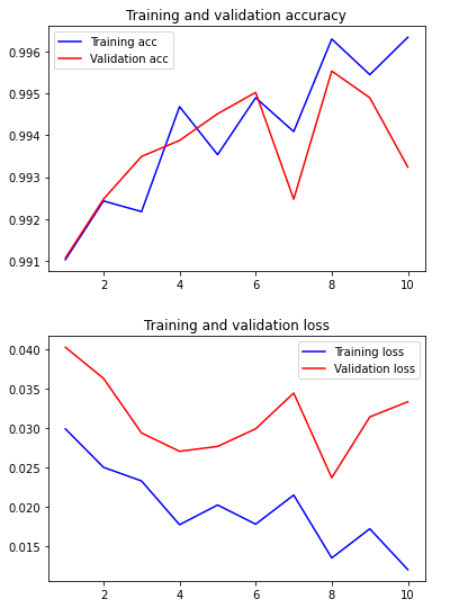
* Sử dụng bộ dữ liệu biển báo giao thông nổi tiếng German Traffic Sign.
* Bộ dữ liệu này gồm khoảng gần 40k ảnh chia thành 43 folder là 43 loại biển báo khác nhau. Mỗi folder sẽ có 1 file CSV chứa thông tin các ảnh trong thư mục. Chúng ta chú ý vài thông tin sau:
* ROI: Là vùng ảnh chứa biển báo với các thông tin X1, Y1, X2, Y2.
* ClassID: Là nhãn của biển số (từ 1 đến 43).

1. Mô hình huấn luyện





1. Kết quả



1. Kết luận

Trong quá trình tìm hiểu và xây dựng mô hình CNN để phân loại biển báo giao thông tôi đã đạt được một số kết quả nhất định như sau:

* Nghiên cứu và tìm hiểu, trình bày một cách khái quát nhất về mạng CNN.
* Nắm được các thành phần cấu thành mạng CNN.
* Xây dựng thành công chương trình phân loại biển báo giao thông.

1. Tài liệu tham khảo

[1] Mehta, Smit, Chirag Paunwala, and Bhaumik Vaidya. "CNN based traffic sign classification using adam optimizer." *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019.