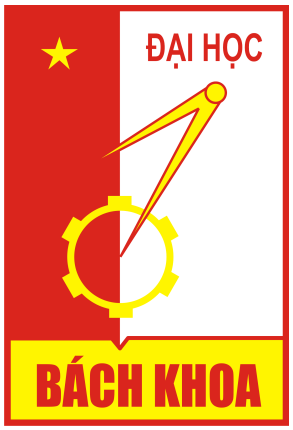
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN

PHÂN LOẠI ẢNH



GIẢNG VIÊN HƯỚNG DẪN:
TS.ĐINH VIẾT SANG

SINH VIÊN THỰC HIỆN:
ĐẶNG MẠNH CƯỜNG - 20130489
PHAN THỊ HỒNG HẠNH - 20131301

14-01-2017

TÓM TẮT

Phân loại ảnh là một trong những bài toán kinh điển trong thị giác máy tính. Mục đích của bài toán là gãn nhãn cho một tập hình ảnh đầu vào. Có rất nhiều phương pháp để giải quyết bài toán này. Trong nội dung bài tập lớn, chúng em sử dụng Deep Learning để giải quyết bài toán.

Mục lục

Danh sách hình vẽ

Chương 1

Giới thiệu chung

1.1 Phát biểu bài toán

Phân loại ảnh là một trong những bài toán trong ngành thị giác máy tính, đã và đang được nghiên cứu rộng rãi. Bài toán sẽ gán nhãn cho tập hình ảnh đầu vào, nhãn có thể là con người, động vật, đồ vật,...
Hiện nay, bài toán đang được áp dụng vào nhiều hệ thống khác nhau:

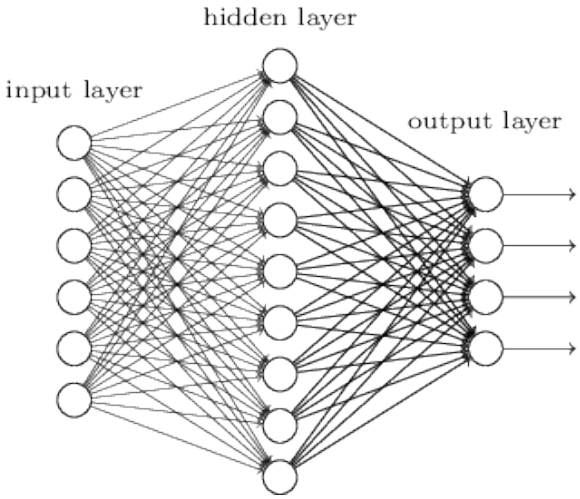
- Hệ thống truy vấn dữ liệu ảnh
- Hệ thống phát hiện mục tiêu và cảnh báo an ninh
- Hệ thống phát hiện làn đường, các đối tượng trên đường đi trong lĩnh vực xe tự hành

1.2 Deep Learning là gì?

Deep learning là một kĩ thuật Machine Learning mạnh mẽ đang được nhiều người trong ngành biết đến và nghiên cứu. Kĩ thuật này nổi trội là do chúng thực hiện được hai việc cùng lúc: biểu diễn thông tin (represent problem/feature engineering) và học (learning). Do đó, kĩ thuật này còn được gọi là representation learning.

Bên cạnh lĩnh vực đã gặt hái được nhiều thành công như xử lý tiếng nói, Deep Learning cũng được áp dụng vào bài toán phân loại ảnh. Với nền tảng là mạng

nơ-ron (Neural Nerwork), Deep Learning hiện đang là một trong những kỹ thuật mạnh mẽ nhất và dữ liệu (Data) là một phần không thể thiếu trong kỹ thuật này. Dưới đây là kiến trúc cơ bản của mạng nơ-ron.



Hình 1.1: Kiến trúc Neral Network

Mạng nơ-ron cơ bản gồm 3 tầng chính:

- Tầng đầu vào (input layer): Tầng biểu diễn thông tin đầu vào, thường được biểu diễn bởi vector đặc trưng cho các thuộc tính của dữ liệu đầu vào
- Tầng ẩn: Biểu diễn, chắt lọc thông tin dữ liệu đầu vào qua các hàm kích hoạt (activation), được liên kết với tầng trước bởi ma trận trọng số.
- Tầng đầu ra (output layer): Biểu diễn thông tin đầu ra, kết quả mà chúng ta mong muốn.

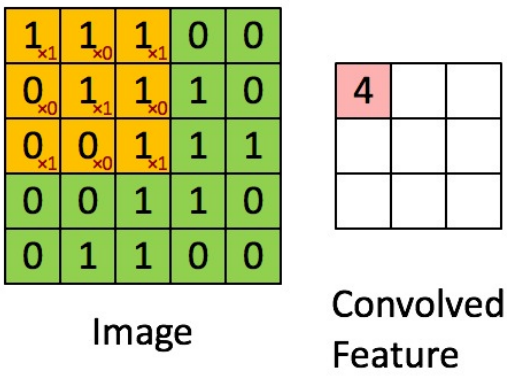
Các nhà khoa học là đã chứng minh rằng trên lý thuyết, với bộ dữ liệu đủ lớn thì chỉ cần một lớp ẩn duy nhất chúng ta có thể giải quyết được tất cả các bài toán. Nhưng trên thực tế để làm được điều này, số lượng nơ-ron ở tầng ẩn sẽ rất lớn, có thể lên tới hàng tỉ nơ-ron và với tốc độ xử lý của các siêu máy tính hiện tại thì chúng ta không thể giải quyết vấn đề với chỉ duy nhất 1 tầng ẩn, thay vào đó mạng nơ-ron sẽ có thêm nhiều lớp ẩn hơn để phù hợp với từng bài toán.

Chương 2

Mô hình

2.1 Convolutional Neural Network là gì?

Đầu tiên, chúng ta sẽ tìm hiểu khái niệm convolution (tích chập). Ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận



Hình 2.1: Tích chập

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh, 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. Ở đây, ta dùng một ma trận filter $3x3$ nhân từng thành phần tương ứng với ma trận ảnh bên trái. Gía trị đầu

ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái.

Convolutional Neural Network 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 cho các layer tiếp theo.

Trong mô hình mạng nơ-ron truyền thống đã trình bày ở chương trước, 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).

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.

2.2 Mô hình áp dụng

Mô hình mà chúng em áp dụng như sau:
[32x32x3]$INPUT$
[32x32x64]$CONV1 + RELU$: 64, 1x5 filter, stride 1, padding SAME
[32x32x64]$CONV2 + RELU$: 64, 5x1 filter, stride 1, pading SAME
[16x16x64]$MAXPOOL1$: 3x3 filter, stride 2
[16x16x64]$NORM1$: Local Response Normalization
[16x16x64]$CONV3 + RELU$: 64, 1x5 filter, stride 1, padding SAME
[16x16x64]$CONV4 + RELU$: 64, 5x1 filter, stride 1, pading SAME
[16x16x64]$NORM2$: Local Response Normalization
[8x8x64]$MAXPOOL2$: 3x3 filter, stride 2
[384] FC1: 384 neurons
[192] FC2: 192 neurons
[10] FC3: 10 neurons

2.2.1 Local Response Normalization

Trong thần kinh học, có một khái niệm gọi là "Ức chế biên". Tế bào thần kinh sẽ bị kích thích bởi các tế bào lân cận, làm gia tăng khả năng nhận thức. Chúng ta sẽ áp dụng cơ chế này vào mô hình học như sau:

- Gọi a^i_{xy} là giá trị tại vị trí (x, y) thuộc kênh i:

$$a^i_{xy} = \frac{a^i_{xy}}{(\gamma + \alpha \sum\limits_{j=max(0,i-r/2}^{min(K-1,i+r/2)} {a^j_{xy}}^2)^\beta}$$

Trong đó:
K là kích thước kênh
r là bán kính lân cận
α, β, γ là các tham số
chọn $\alpha = 0.001, \beta = 0.75, \gamma = 0.1$

2.2.2 Hàm mục tiêu

Cost function:

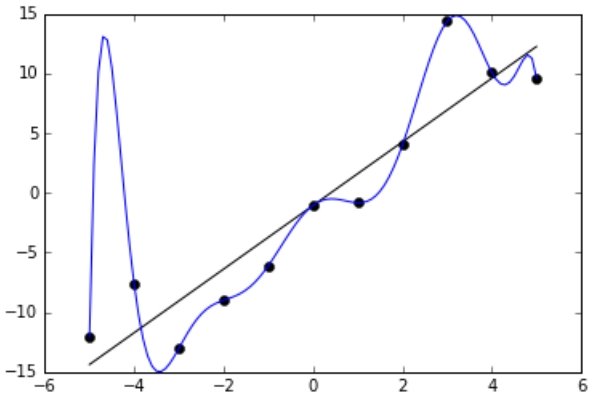
$$C = -\frac{1}{n} \sum_x [ylna + (1-y)ln(1-a)]$$

Trong đó:
y là kết quả thực
a là kết quả do mô hình dự đoán
Cực tiểu hàm mục tiêu sử dụng Stochastic gradient descent kết hợp thuật toán Lan truyền ngược

2.2.3 Vấn đề Overfitting

Trong ứng dụng thực tế, ta thường sử dụng mạng neuron để mô phỏng những hàm số mà cấu trúc của chúng vẫn chưa được xác định. Khi đó, ta chỉ có thể thu nhập được các bộ mẫu dữ liệu ra (vào) được sinh ra từ hàm số, nhưng lại không thể đặc tả quá trình sinh ra các bộ mẫu đó. Một ví dụ kinh điển đó là quá trình

bộ não con người thu nhận thông tin từ hình ảnh của chữ viết tay, rồi suy luận ra chữ viết. Cơ chế bộ não biểu diễn hình ảnh và suy luận ra thông tin từ đó là một ẩn số đối với khoa học. Tuy nhiên, ta có thể dùng các bức ảnh cùng với nhãn đúng của chúng để huấn luyện mạng neuron mô phỏng xấp xỉ được quá trình xử lý hình ảnh của bộ não. Cho dù cấu trúc giữa bộ não và mạng neuron (có thể) khác nhau, với một thuật toán huấn luyện tốt, chúng sẽ đưa ra kết luận giống nhau với cùng một điểm dữ liệu vào



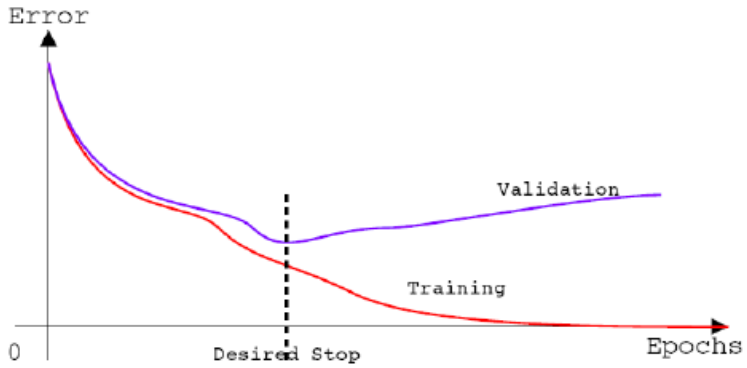
Hình 2.2: Một ví dụ về overfitting

Đối với bài toán dự đoán, vì mục tiêu cuối cùng của ta là mô phỏng một hàm số ẩn, ta không nên cực tiểu hóa hàm mất mát trên tập huấn luyện. Nếu ta làm như vậy sẽ dẫn đến hiện tượng overfitting, tức là mạng neuron sẽ học được một hàm phức tạp để mô phỏng hoàn hảo nhất tập huấn luyện. Tuy nhiên, cũng do cấu trúc phức tạp, hàm này không có tính tổng quát hóa cao, tức là nó rất dễ sai khi gặp một điểm dữ liệu không có trong tập huấn luyện. Khi ấy, mạng neuron giống như một con người chỉ biết học tủ mà không biết cách vận dụng kiến thức để giải quyết những thứ chưa từng gặp phải. Overfitting là một vấn đề nghiêm trọng đối với mạng neuron vì khả năng mô hình hóa của chúng quá cao, dễ dàng học được các hàm phức tạp. Ta sẽ tìm hiểu một số phương pháp thông dụng để chẩn đoán và ngăn ngừa overfitting cho mạng neuron

2.2.3.1 Early Stopping

Để đánh giá độ sai lệch của mạng neuron một cách thực tế, ta không thể dùng giá trị của hàm mất mát trên tập huấn luyện bởi vì giá trị này thường sẽ thấp

hơn độ sai lệch thực tế. Muốn khách quan, ta thường sử dụng một tập Validation (validation set), độc lập với tập huấn luyện và tập test, để tính độ sai lệch trên đó. Quá trình huấn luyện sẽ diễn ra như sau:



Hình 2.3: Early Stopping

- Theo dõi độ sai lệch trên tập huấn luyện và tập validation
- Khi giá trị của độ sai lệch trên tập validation bắt đầu tăng, dừng huấn luyện.
- Cố định mạng neuron với tập tham số mà cho độ sai lệch trên tập validation tốt nhất cho đến trước thời điểm dừng. Dùng tập test để đánh giá mạng neuron này một lần cuối và thông báo kết quả

2.2.3.2 L2 regularization

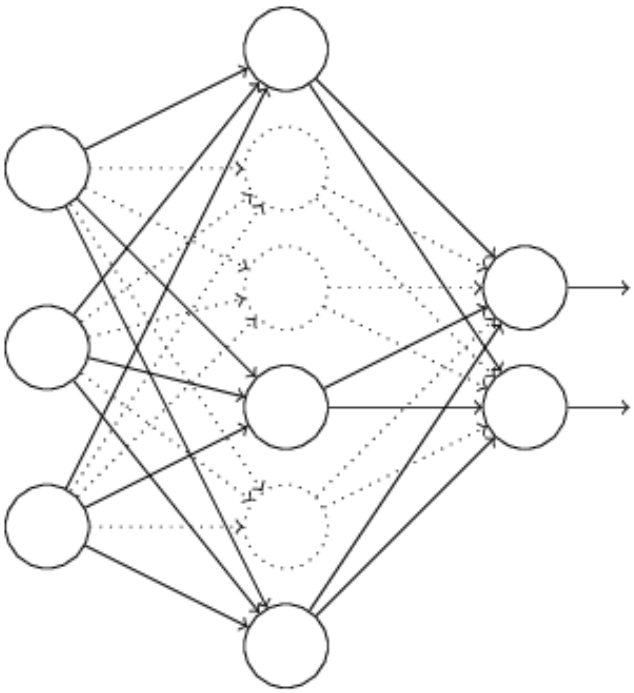
Một cách khác để ngăn ngừa overfitting đó là giới hạn không gian hàm số mà mạng neuron có thể biểu diễn. Điều này được tiến hành bằng việc thu hẹp lại biên độ giá trị của tập tham số. Cụ thể hơn, thay vì huấn luyện mạng chỉ bằng hàm mục tiêu ở trên, ta thêm vào một đại lượng thể hiện "độ lớn" của tập tham số

$$C = -\frac{1}{n}\sum_{x}[ylna + (1-y)ln(1-a)] + \lambda\sum_{w}||w||^2$$

với λ là tham số điều khiển, w là các trọng số.

2.2.3.3 Dropout

Thay vì chỉ huấn luyện trên một mô hình mạng duy nhất. Dropout cho phép dữ liệu được huấn luyện với các kiến trúc mạng khác nhau. Khi được học trên các kiến trúc mạng khác nhau, mỗi kiến trúc có thể sẽ cho ta những kết quả khác nhau, từ đó ta có thể lựa chọn kết quả phù hợp nhất. Với nhiều kiến trúc khác nhau, mô hình sẽ học tổng quát hơn, từ đó có thể ngăn chặn được overfitting.



Hình 2.4: Ví dụ về kĩ thuật dropout

- Ở mỗi bước huấn luyện, ta xóa ngẫu nhiên một nửa số neuron ở tẳng ẩn
- Từ đó, ta sẽ thu được kết quả học từ nhiều mạng neuron khác nhau

Chương 3

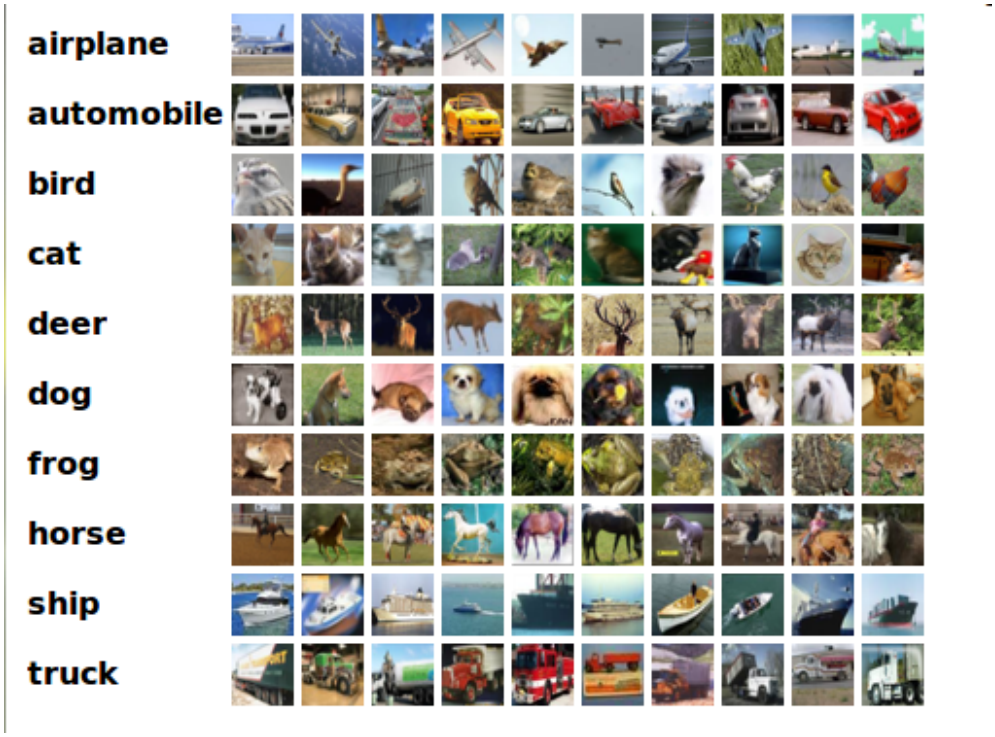
Kết quả thực nghiệm

Thực nghiệm trên 2 bộ dữ liệu Binary-class và multi-class

3.1 Dữ liệu thực nghiệm

Bộ dữ liệu CIFAR 10:

- Bộ dữ liệu gồm 60000 hình ảnh màu kích thước 32x32 trên 10 lớp đối tượng khác nhau, 6000 ảnh trên 1 đối tượng
- Chia thành 2 tập
  - Tập huấn luyện: gồm 50000 hình ảnh kèm theo nhãn
  - Tập test: gồm 10000 hình ảnh kèm theo nhãn



Hình 3.1: Ví dụ về bộ dữ liệu

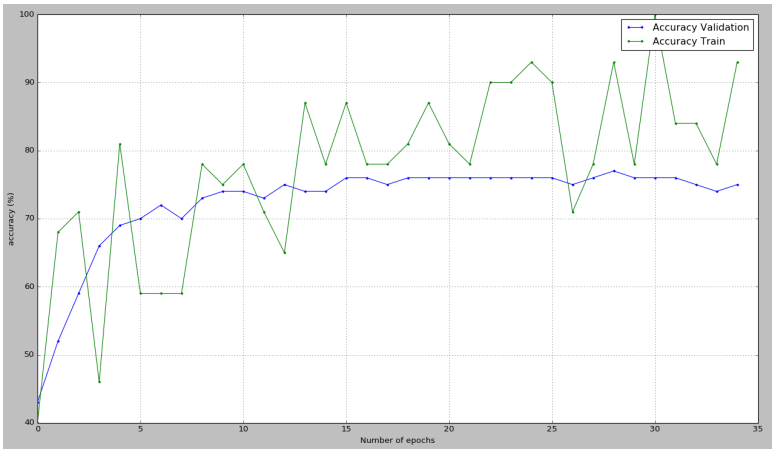
3.2 Môi trường thực nghiệm

- Hệ điều hành ubuntu 14.04-64 bit
- Intel core i7 - 3540M, CPU 3.0Ghz x 4
- Ngôn ngữ lập trình: python
- Thư viện hỗ trợ: Tensorflow

3.3 Kết quả

Đồ thị thể hiện độ chính xác trên tập training và tập validation trong quá trình học

Kết quả tại bước huấn luyện có độ chính xác trên tập validation là lớn nhất là:



Hình 3.2: Độ chính xác trong quá trình học

- Độ chính xác trên tập huấn luyện: 90%
- Độ chính xác trên tập test: 78%

Chương 4

Kết luận và hướng phát triển

Trong nội dung bài tập lớn, chúng em đã xây dựng mô hình khá cơ bản, dựa trên mạng AlexNet để giải quyết bài toán phân loại ảnh.

Tuy nhiên, kết quả chưa thực sự tốt, chúng em sẽ cố gắng cải tiến mô hình để đạt kết quả tốt hơn. Ngoài ra, còn một số mô hình như mạng Resnet, Googlenet,.. Trong tương lai chúng em sẽ thực nghiệm với những mô hình trên để so sánh với mô hình của chúng em.

Tài liệu tham khảo

Michael Nielsen, Neural Networks and Deeplearning. *neuralnetworksanddeeplearning.com*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, (pp. 1097-1105).

Ong Xuan Hong. Convolutional Neural Network. *https://ongxuanhong.wordpress.com/*.