

Phân loại ảnh sử dụng Convolutional Neural Network

Sinh viên thực hiện
Đặng Mạnh Cường Phan Thị Hồng Hạnh

Giáo viên hướng dẫn
TS.Đinh Viết Sang

Hà Nội, 14-01-2017

1 Nội dung đề tài

2 Mô hình

- Convolutional Neural Network
- Mô hình áp dụng

3 Kết quả thực nghiệm

- Bộ dữ liệu
- Kết quả

- Phân loại tập các hình ảnh thành các tập đối tượng khác nhau
- Với một hình ảnh đầu vào, gán nhãn cho hình ảnh đó
- Sử dụng Convolutional Neural Network

1 Nội dung đề tài

2 Mô hình

- Convolutional Neural Network
- Mô hình áp dụng

3 Kết quả thực nghiệm

- Bộ dữ liệu
- Kết quả

Convolution (Tích chập):

- Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số
- Ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận

Convolutional Neural Network

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

Convolutional Neural Network

1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	1	1	0	0

Image

4	3	

Convolved
Feature

Convolutional Neural Network

1	1	1 _{x1}	0 _{x0}	0 _{x1}
0	1	1 _{x0}	1 _{x1}	0 _{x0}
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1	1	0
0	1	1	0	0

Image

4	3	4

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0 _{x1}	1 _{x0}	1 _{x1}	1	0
0 _{x0}	0 _{x1}	1 _{x0}	1	1
0 _{x1}	0 _{x0}	1 _{x1}	1	0
0	1	1	0	0

Image

4	3	4
2		

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0	1 _{x1}	1 _{x0}	1 _{x1}	0
0	0 _{x0}	1 _{x1}	1 _{x0}	1
0	0 _{x1}	1 _{x0}	1 _{x1}	0
0	1	1	0	0

Image

4	3	4
2	4	

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0	1	1 _{x1}	1 _{x0}	0 _{x1}
0	0	1 _{x0}	1 _{x1}	1 _{x0}
0	0	1 _{x1}	1 _{x0}	0 _{x1}
0	1	1	0	0

Image

4	3	4
2	4	3

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0	1	1	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0 _{x0}	0 _{x1}	1 _{x0}	1	0
0 _{x1}	1 _{x0}	1 _{x1}	0	0

Image

4	3	4
2	4	3
2		

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0	1	1	1	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0 _{x0}	1 _{x1}	1 _{x0}	0
0	1 _{x1}	1 _{x0}	0 _{x1}	0

Image

4	3	4
2	4	3
2	3	

Convolved
Feature

Convolutional Neural Network

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

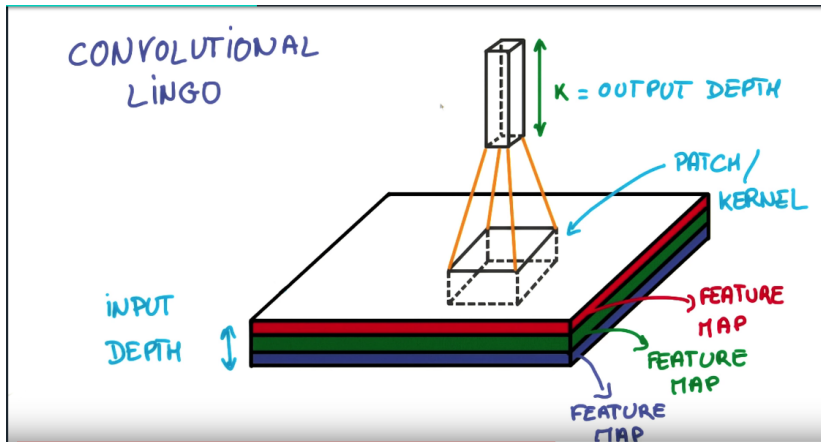
4	3	4
2	4	3
2	3	4

Convolved
Feature

Convolutional Neural Network

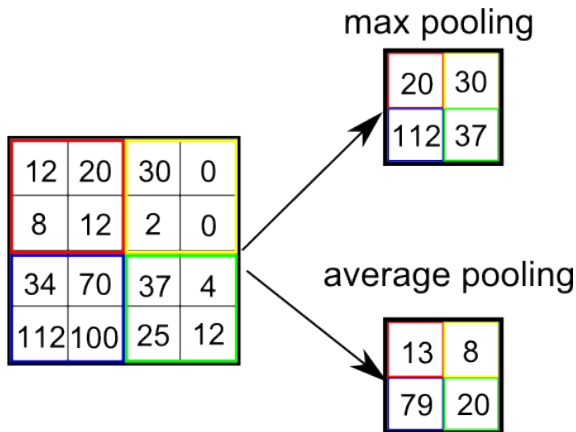
- CNNs chỉ đơn giản gồm một vài layer của convolution kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU hay tanh để tạo ra thông tin trừu tượng hơn cho các layer tiếp theo.
- Các layer liên kết được với nhau thông qua cơ chế convolution
- Ngoài ra còn một số layer khác như maxpooling, norm dùng để chuẩn hóa, chắt lọc thông tin hữu ích.

Convolutional Neural Network



Pooling

Convolutional Neural NetWork



1 Nội dung đề tài

2 Mô hình

- Convolutional Neural Network
- Mô hình áp dụng

3 Kết quả thực nghiệm

- Bộ dữ liệu
- Kết quả

Mô hình áp dụng

[32x32x3] *INPUT*

[32x32x64] *CONV1 + RELU* : 64, 1x5 filter, stride 1, padding SAME

[32x32x64] *CONV2 + RELU* : 64, 5x1 filter, stride 1, padding 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, padding SAME

[16x16x64] *NORM2*: Local Response Normalization

[8x8x64] *MAXPOOL2*: 3x3 filter, stride 2

[384] *FC1*: 384 neurons

[192] *FC2*: 192 neurons

[10] *FC3*: 10 neurons

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. \Rightarrow Áp dụng vào mạng neuron
- Gọi a_{xy}^i là giá trị tại vị trí (x, y) thuộc kênh i :

$$a_{xy}^i = \frac{a_{xy}^i}{(\gamma + \alpha \sum_{j=\max(0, i-r/2)}^{\min(K-1, i+r/2)} a_{xy}^j)^2)^{\beta}}$$

Trong đó:

K là kích thước kênh

r là bán kính lân cận

α, β, γ là các tham số

chọn $\alpha = 0.001, \beta = 0.75, \gamma = 0.1$

Cost function:

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

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

- Là hiện tượng mô hình quá khớp với dữ liệu huấn luyện khiến cho kết quả trên tập test không tốt như mong muốn
- Làm thế nào để giải quyết vấn đề overfitting?
 - L2 regularization
 - Early Stopping
 - Dropout

L2 regularization

Overfitting

Ý tưởng:

- Giới hạn phạm vi học của các tham số

Đặt:

$$T = \lambda \sum_w ||w||^2$$

Cost function:

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

Early Stopping

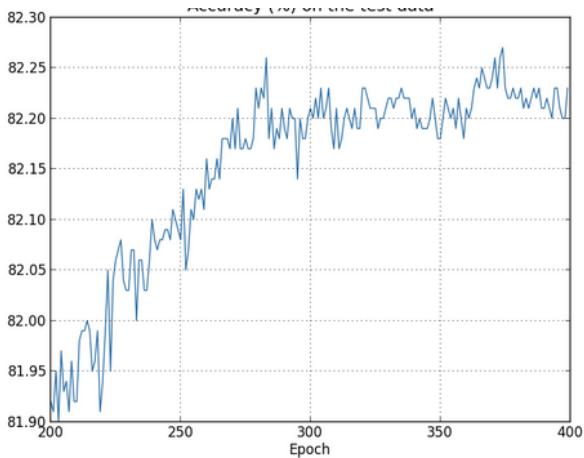
Overfitting

Ý tưởng:

- Xây dựng tập Validation, một phần nhỏ từ tập huấn luyện (1/10 tập huấn luyện)
- Sử dụng tập Validation để phát hiện tượng Overfitting
 - Training trên tập huấn luyện cho đến khi độ chính xác trên tập validation bão hòa \Rightarrow dừng quá trình training

Early Stopping

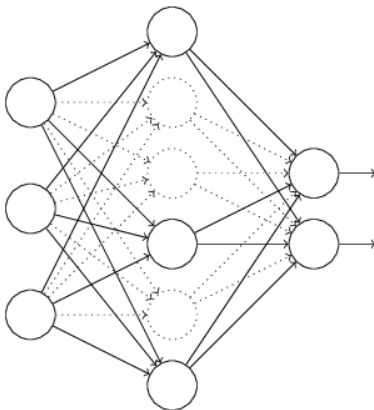
Overfitting



Dropout

Ý tưởng:

- Thay đổi cấu trúc mạng ở mỗi bước training
- Xóa ngẫu nhiên một nửa số neuron ở tầng ẩn



1 Nội dung đề tài

2 Mô hình

- Convolutional Neural Network
- Mô hình áp dụng

3 Kết quả thực nghiệm

- Bộ dữ liệu
- Kết quả

CIFAR10: Gồm 60000 hình ảnh màu kích thước 32×32 trên 10 đối tượng khác nhau

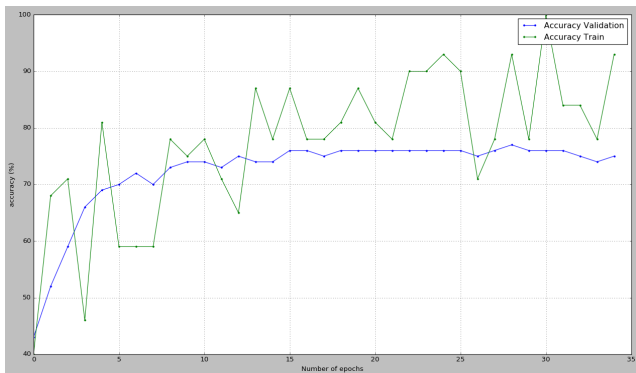
- Tập huấn luyện: gồm 50000 hình ảnh, kèm theo nhãn tương ứng
- Tập test: gồm 10000 hình ảnh, kèm theo nhãn tương ứng

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

Kết quả

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



- Độ chính xác trên tập huấn luyện: 90%
- Độ chính xác trên tập test: 78%
- Thời gian huấn luyện: 6 tiếng

1 Nội dung đề tài

2 Mô hình

- Convolutional Neural Network
- Mô hình áp dụng

3 Kết quả thực nghiệm

- Bộ dữ liệu
- Kết quả

THANK YOU!