Convolutional Neural Networks

Amin Mir
Mohammad Javadi
Application

- Image recognition
- Completely dominated the machine vision space
- One of the hottest topics in AI today
- Tricky to understand
Why not Regular Neural Nets

• They don't scale well to full images.

• In CIFAR-10
  ◦ Images of size $32 \times 32 \times 3 \Rightarrow 3072$ weights per neuron

• Larger images
  ◦ $200 \times 200 \times 3 \Rightarrow 120,000$ weights
Why not Regular Neural Nets

• Input consists of images

• Neurons in layers arranged in three dimension:
  ○ Width, height, depth
Why not Regular Neural Nets
Historical Overview

- CNN’s are inspired by organization of animals visual cortex
- In 1998, Yann LeCun et al. presented first CNN
- Between 10 thousands of images, it gave only 82 case errors
ImageNet Challenge

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
- As of 2016, over ten million of images have been hand-annotated
- Every year error rates fell to a few percent (25%, 16% …)
ConvNet Architecture

- Convolution Layer
- ReLU Layer
- Pooling Layer
- Fully-Connected Layer
- Softmax
Convolution Layer

- Neurons are not fully-connected
- Compute dot product
Convolution Layer
Convolution Layer

- Several filters
Convolution Layer

- Weights $\rightarrow$ Learnable Filter
Convolution Layer

- Slide the filter over the width and height
- Like Convolution Operation
- Produces a 2-dimensional activation map
Convolution Layer

- Example
Convolution Layer
Convolution Layer
ReLU

Rectified linear unit

Applies an element-wise activation function

\[ y = \max(0, X) \]
Pooling Layer

• Performs a downsampling operation

• Progressively reduces the spatial size of activation maps
  ◦ Shrinks the number of parameters & computation
  ◦ Control overfitting

• Max pool with filters of size 2 and stride of 2
  ◦ reduces the spatial extent by half
Pooling Layer

The diagram illustrates the process of pooling in a neural network. It shows how a 224x224x64 input is downsized to 112x112x64 through the process of pooling. Each depth slice is processed through a max pooling operation with 2x2 filters and a stride of 2, resulting in a reduced dimensionality of 112x112.
Fully-Connected Layer

- Fully-Connected
- No parameter sharing
- Using ReLU activation function instead of Sigmoid is common
Fully-Connected Layer
Softmax

- Normalized exponential function
- Generalization of the logistic function

\[ \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K. \]

this is called the “logit”
CNN Architecture

• Stack Conv/ReLU

• Periodically use Pool layers
CNN in Practice
VGGNet (Simonyan and Zisserman 2014)

- 3x3 filters
- zero-padding of 1
- stride of 2
- 2x2 MAX POOLING with stride of 2
- 7.3% top five error
### VGGNet

**INPUT**: [224x224x3]  memory: 224*224*3=150K  params: 0  (not counting biases)

**CONV3-64**: [224x224x64]  memory: 224*224*64=3.2M  params: (3*3*3)*64 = 1,728

**CONV3-64**: [224x224x64]  memory: 224*224*64=3.2M  params: (3*3*64)*64 = 36,864

**POOL2**: [112x112x64]  memory: 112*112*64=800K  params: 0

**CONV3-128**: [112x112x128]  memory: 112*112*128=1.6M  params: (3*3*64)*128 = 73,728

**CONV3-128**: [112x112x128]  memory: 112*112*128=1.6M  params: (3*3*128)*128 = 147,456

**POOL2**: [56x56x128]  memory: 56*56*128=400K  params: 0

**CONV3-256**: [56x56x256]  memory: 56*56*256=800K  params: (3*3*128)*256 = 294,912

**CONV3-256**: [56x56x256]  memory: 56*56*256=800K  params: (3*3*256)*256 = 589,824

**CONV3-256**: [56x56x256]  memory: 56*56*256=800K  params: (3*3*256)*256 = 589,824

**POOL2**: [28x28x256]  memory: 28*28*256=200K  params: 0

**CONV3-512**: [28x28x512]  memory: 28*28*512=400K  params: (3*3*256)*512 = 1,179,648

**CONV3-512**: [28x28x512]  memory: 28*28*512=400K  params: (3*3*512)*512 = 2,359,296

**CONV3-512**: [28x28x512]  memory: 28*28*512=400K  params: (3*3*512)*512 = 2,359,296

**POOL2**: [14x14x512]  memory: 14*14*512=100K  params: 0

**CONV3-512**: [14x14x512]  memory: 14*14*512=100K  params: (3*3*512)*512 = 2,359,296

**CONV3-512**: [14x14x512]  memory: 14*14*512=100K  params: (3*3*512)*512 = 2,359,296

**CONV3-512**: [14x14x512]  memory: 14*14*512=100K  params: (3*3*512)*512 = 2,359,296

**POOL2**: [7x7x512]  memory: 7*7*512=25K  params: 0

**FC**: [1x1x4096]  memory: 4096  params: 7*7*512*4096 = 102,760,448

**FC**: [1x1x4096]  memory: 4096  params: 4096*4096 = 16,777,216

**FC**: [1x1x1000]  memory: 1000  params: 4096*1000 = 4,096,000
LeNet

- 5 x 5 filter with stride of 1
- 2x2 MAX POOLING with stride of 2
Other Examples

• GoogleNet
• MSRA (Microsoft Research Asia)
• SqueezeNet
• And …
Toolbox and frameworks

- Caffe
- Tensorflow
- CNTK (Microsoft)
- Theano
- and …
Showtime

- http://demo.caffe.berkeleyvision.org/
DenseCap

• Fei-Fei Li

• Andrej Karpathy

• Justin Johnson

• Dense Captioning

• a Convolutional Network

• a dense localization layer

• Recurrent Neural Network language
DenseCap

“We introduce the dense captioning task, which requires a computer vision system to both localize and describe salient regions in images in natural language.”
bus parked on the street. a city street scene. front windshield of a bus. man walking on sidewalk. a silver car parked on the street. a city scene. a green traffic light. a building in the background. the bus has a number. a large building. a brick building. red brick building with windows. a blue sign with a white arrow. white lines on the road.

a plate of food. food on a plate. a blue cup on a table. a plate of food. a blue bowl with red sauce. a bowl of soup. a cup of coffee. a bowl of chocolate. a glass of water. a plate of food. a silver metal container. a small bowl of sauce. table with food on it. a slice of orange. a table with food on it. a slice of meat. yellow and white cheese.

green jacket. a white horse. a man on a horse. two people riding horses. man wearing a green jacket. the helmet is black. brown horse with white mane. white van parked on the street. a paved sidewalk. green and yellow jacket. a helmet on the head. white horse with white face.
Other Researches and Applications

- FaceApp
- JibJab
- Soccer Activity Recognition
- and ...
Thanks everyone!
Any Question??!!