Course Outline

- Fundamentals
  - Bayes Decision Theory
  - Probability Density Estimation
- Classification Approaches
  - Linear Discriminants
  - Support Vector Machines
  - Ensemble Methods & Boosting
  - Random Forests
- Deep Learning
  - Foundations
  - Convolutional Neural Networks
  - Recurrent Neural Networks

Recap: Convolutional Neural Networks

- Neural network with specialized connectivity structure
  - Stack multiple stages of feature extractors
  - Higher stages compute more global, more invariant features
  - Classification layer at the end

Recap: Intuition of CNNs

- Convolutional net
  - Share the same parameters across different locations
  - Convolutions with learned kernels
- Learn multiple filters
  - E.g. 1000 x 1000 image
    - 100 filters
    - 10 x 10 filter size
    - Only 10k parameters
- Result: Response map
  - Size: 1000 x 1000 x 100
  - Only memory, not params!

Important Conceptual Shift

- Before
  - input layer
  - hidden layer
  - output layer

- Now:
Convolution Layers

• Note: Connectivity is
  - Local in space (5 × 5 inside 32 × 32)
  - But full in depth (all 3 depth channels)

Before: Full connectivity
32 × 32 × 3 weights

Now: Local connectivity
One neuron connects to, e.g.,
5 × 5 × 3 region.
⇒ Only 5 × 5 × 3 shared weights.

All Neural Net activations arranged in 3 dimensions
  - Multiple neurons all looking at the same input region, stacked in depth
  - Form a single [1 × 1 × depth] depth column in output volume.

Replicate this column of hidden neurons across space, with some stride.

Example:
7 × 7 input
assume 3 × 3 connectivity
stride 1
Convolution Layers

- Replicate this column of hidden neurons across space, with some stride.

Example:
7 × 7 input
assume 3 × 3 connectivity
stride 1
⇒ 5 × 5 output

What about stride 2?
⇒ 3 × 3 output

Convolution Layers

- Replicate this column of hidden neurons across space, with some stride.

Example:
7 × 7 input
assume 3 × 3 connectivity
stride 1
⇒ 5 × 5 output

What about stride 2?
⇒ 3 × 3 output

In practice, common to zero-pad the border.
- Preserves the size of the input spatially.
Activation Maps of Convolutional Filters

Each activation map is a depth slice through the output volume.

Effect of Multiple Convolution Layers

Feature visualization of convolutional net trained on ImageNet from [Zeller & Fergus 2015]

Convolutional Networks: Intuition

- Let’s assume the filter is an eye detector
  - How can we make the detection robust to the exact location of the eye?

- Solution:
  - By pooling (e.g., max or avg) filter responses at different spatial locations, we gain robustness to the exact spatial location of features.

Max Pooling

- Effect:
  - Make the representation smaller without losing too much information
  - Achieve robustness to translations

- Note
  - Pooling happens independently across each slice, preserving the number of slices.
### CNNs: Implication for Back-Propagation

- **Convolutional layers**
  - Filter weights are shared between locations
  - Gradients are added for each filter location.

### Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

### CNN Architectures: LeNet (1998)

- Early convolutional architecture
  - 2 Convolutional layers, 2 pooling layers
  - Fully-connected NN layers for classification
  - Successfully used for handwritten digit recognition (MNIST)


### CNN Architectures: AlexNet (2012)

- Similar framework as LeNet, but
  - Bigger model (7 hidden layers, 650k units, 60M parameters)
  - More data (10^6 images instead of 10^3)
  - GPU implementation
  - Better regularization and up-to-date tricks for training (Dropout)


### ImageNet Challenge 2012

- ImageNet
  - ~14M labeled internet images
  - 20k classes
  - Human labels via Amazon Mechanical Turk

- Challenge (ILSVRC)
  - 1.2 million training images
  - 1000 classes
  - Goal: Predict ground-truth class within top-5 responses
  - Currently one of the top benchmarks in Computer Vision

[Deng et al., CVPR’09]

### ILSVRC 2012 Results

- AlexNet almost halved the error rate
  - 16.4% error (top-5) vs. 26.2% for the next best approach
  - A revolution in Computer Vision
  - Acquired by Google in Jan ’13, deployed in Google+ in May ’13
CNN Architectures: VGGNet (2014/15)

**Main ideas**
- Deeper network
- Stacked convolutional layers with smaller filters (+ nonlinearity)
- Detailed evaluation of all components

**Results**
- Improved ILSVRC top-5 error rate to 6.7%.

Comparison: AlexNet vs. VGGNet

- **Receptive fields in the first layer**
  - **AlexNet**: $11 \times 11$, stride 4
  - Zeiler & Fergus: $7 \times 7$, stride 2
  - **VGGNet**: $3 \times 3$, stride 1

- **Why that?**
  - If you stack a $3 \times 3$ on top of another $3 \times 3$ layer, you effectively get a $5 \times 5$ receptive field.
  - With three $3 \times 3$ layers, the receptive field is already $7 \times 7$.
  - But much fewer parameters: $3^3 \times 3 = 27$ instead of $7^2 = 49$.
  - In addition, non-linearities in-between $3 \times 3$ layers for additional discriminativity.

GoogLeNet Visualization

- **Inception module + copies**
- **Auxiliary classification outputs for training the lower layers (deprecated)**


**Main ideas**
- “Inception” module as modular component
- Learns filters at several scales within each module

Results on ILSVRC

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG (142, multi crop &amp; deeper arch)</td>
<td>25.5</td>
<td>6.4</td>
<td>6.8</td>
</tr>
<tr>
<td>VGG (142, multi crop &amp; 32x32 arch)</td>
<td>24.4</td>
<td>6.2</td>
<td>6.6</td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2014) (Inception)</td>
<td>24.7</td>
<td>6.3</td>
<td>6.8</td>
</tr>
<tr>
<td>GoogLeNet (Simonyan et al., 2015) (1 x 1)</td>
<td>24.7</td>
<td>6.3</td>
<td>6.8</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (multi crop)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2015) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>VGG (Simonyan et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Inception (Szegedy et al., 2014) (1 x 1)</td>
<td>27.5</td>
<td>7.1</td>
<td>7.3</td>
</tr>
</tbody>
</table>
Newer Developments: Residual Networks

- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers
  - We’ll analyze this mechanism in more detail later…

ImageNet Performance

Understanding the ILSVRC Challenge

- Imagine the scope of the problem!
  - 1000 categories
  - 1.2M training images
  - 50k validation images

- This means...
  - Speaking out the list of category names at 1 word/s... takes 15mins.
  - Watching a slideshow of the validation images at 2s/image... takes a full day (24h+).
  - Watching a slideshow of the training images at 2s/image... takes a full month.

More Finegrained Classes
Quirks and Limitations of the Data Set

- Generated from WordNet ontology
  - Some animal categories are overrepresented
  - E.g., 120 subcategories of dog breeds
  - \( \Rightarrow \) 6.7% top-5 error looks all the more impressive

Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogleNet
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

Visualizing CNNs

- DeconvNet
- ConvNet

Visualizing CNNs

- Reconstruction of image patches from that unit (indicates aspect of patches which unit is sensitive to)
- Top 9 image patches that cause maximal activation in layer 2 unit

Visualizing CNNs

- Layer 3

Visualizing CNNs

- Layer 4
- Layer 5
What Does the Network React To?

- Occlusion Experiment
  - Mask part of the image with an occluding square.
  - Monitor the output

Image source: M. Zeiler, R. Fergus
Slide credit: Svetlana Lazebnik, Rob Fergus
What Does the Network React To?

Input image

Total activation in most active 5th layer feature map

Other activations from the same feature map.

Inceptionism: Dreaming ConvNets

• Idea
  - Start with a random noise image.
  - Enhance the input image such as to enforce a particular response (e.g., banana).
  - Combine with prior constraint that image should have similar statistics as natural images.
  - Network hallucinates characteristics of the learned class.

http://googleresearch.blogspot.de/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism: Dreaming ConvNets

• Results

http://googleresearch.blogspot.de/2015/07/deepdream-code-example-for-visualizing.html

https://www.youtube.com/watch?v=IREsx-xW09g

Topics of This Lecture

• Recap: CNNs
• CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
• Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
• Applications

The Learned Features are Generic

• Experiment: feature transfer
  - Train network on ImageNet
  - Chop off last layer and train classification layer on CalTech256
  - State of the art accuracy already with only 6 training images

state of the art level (pre-CNN)
### Other Tasks: Detection

**R-CNN: Regions with CNN features**

1. Input
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

- Results on PASCAL VOC Detection benchmark
  - Pre-CNN state of the art: 35.1% mAP [Uijlings et al., 2013]
  - R-CNN: 53.7% mAP


---

### Most Recent Version: Faster R-CNN

- One network, four losses
  - Remove dependence on external region proposal algorithm.
  - Instead, infer region proposals from same CNN.
  - Feature sharing
  - Joint training
  - Object detection in a single pass becomes possible.

- mAP improved to >70%

---

**Faster R-CNN (based on ResNets)**


---

**YOLO**


---

### Semantic Image Segmentation

- Perform pixel-wise prediction task
  - Usually done using Fully Convolutional Networks (FCNs)
    - All operations formulated as convolutions
    - Advantage: can process arbitrarily sized images

Image source: Long, Shelhamer, Darrell
Semantic Image Segmentation

- Encoder-Decoder Architecture
  - Problem: FCN output has low resolution
  - Solution: perform upsampling to get back to desired resolution
  - Use skip connections to preserve higher-resolution information

Semantic Segmentation

- More recent results
  - Based on an extension of ResNets

Other Tasks: Face Verification

Other Tasks: Face Verification

Commercial Recognition Services

- E.g., clarifai

Commercial Recognition Services

- Be careful when taking test images from Google Search
  - Chances are they may have been seen in the training set...

References and Further Reading

- LeNet

- AlexNet

- VGGNet
  - K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

- GoogLeNet
• ResNet

• ReLu

• Initialization
  X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, AISTATS 2010.

• Batch Normalization

• Dropout