Advanced Machine Learning
Lecture 16

Convolutional Neural Networks II


Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de
This Lecture: *Advanced Machine Learning*

- **Regression Approaches**
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
  - Gaussian Processes

- **Approximate Inference**
  - Sampling Approaches
  - MCMC

- **Deep Learning**
  - Linear Discriminants
  - Neural Networks
  - Backpropagation & Optimization
  - CNNs, RNNs, ResNets, etc.
Topics of This Lecture

• Recap: CNNs

• CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNets

• Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures

• Applications
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


Slide credit: Svetlana Lazebnik
Recap: Intuition of CNNs

- **Convolutional net**
  - Share the same parameters across different locations
  - Convolutions with learned kernels

- Learn *multiple* filters
  - E.g. 1000×1000 image
  - 100 filters
  - 10×10 filter size
  - ⇒ only 10k parameters

- **Result: Response map**
  - size: 1000×1000×100
  - Only memory, not params!
Recap: Convolution Layers

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

Naming convention:

Slide credit: FeiFei Li, Andrej Karpathy
Recap: Activation Maps

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

5 × 5 filters

Activation maps

Slide adapted from FeiFei Li, Andrej Karpathy
Recap: Pooling Layers

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

- Single depth slice
  - 1 1 2 4
  - 5 6 7 8
  - 3 2 1 0
  - 1 2 3 4

- Max pool with 2x2 filters and stride 2
  - 6 8
  - 3 4

Slide adapted from FeiFei Li, Andrej Karpathy
Topics of This Lecture

• Recap: CNNs

• CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet

• Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures

• Applications
Recap: 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]
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)

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

B. Leibe
CNN Architectures: VGGNet (2014/15)


Image source: Hirokatsu Kataoka
### 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%.

#### ConvNet Configuration

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A-LRN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11 weight layers</td>
<td>11 weight layers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13 weight layers</td>
<td>16 weight layers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 weight layers</td>
<td>19 weight layers</td>
</tr>
</tbody>
</table>

- input (224 x 224 RGB image)

<table>
<thead>
<tr>
<th></th>
<th>conv3-64</th>
<th>conv3-64</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LRN</td>
<td>conv3-64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>conv3-128</th>
<th>conv3-128</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-128</td>
<td>conv3-128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>conv3-256</th>
<th>conv3-256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-256</td>
<td>conv3-256</td>
</tr>
<tr>
<td></td>
<td>conv1-256</td>
<td>conv3-256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>conv3-512</th>
<th>conv3-512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td></td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td></td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td></td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>maxpool</th>
<th>maxpool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-128</td>
<td>conv3-128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>maxpool</th>
<th>maxpool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-256</td>
<td>conv3-256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>maxpool</th>
<th>maxpool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FC-4096</th>
<th>FC-4096</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FC-1000</td>
<td>soft-max</td>
</tr>
</tbody>
</table>

- Mainly used

Image source: Simonyan & Zisserman
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 three \(3 \times 3\) 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: \(3 \cdot 3^2 = 27\) instead of \(7^2 = 49\).
  - In addition, non-linearities in-between \(3 \times 3\) layers for additional discriminativity.
CNN Architectures: GoogLeNet (2014)

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

GoogLeNet Visualization

Inception module + copies

Auxiliary classification outputs for training the lower layers (deprecated)

Convolution Pooling Softmax Other
### Results on ILSVRC

<table>
<thead>
<tr>
<th>Method</th>
<th>top-1 val. error (%)</th>
<th>top-5 val. error (%)</th>
<th>top-5 test error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG (2 nets, multi-crop &amp; dense eval.)</td>
<td>23.7</td>
<td>6.8</td>
<td>6.8</td>
</tr>
<tr>
<td>VGG (1 net, multi-crop &amp; dense eval.)</td>
<td>24.4</td>
<td>7.1</td>
<td>7.0</td>
</tr>
<tr>
<td>VGG (ILSVRC submission, 7 nets, dense eval.)</td>
<td>24.7</td>
<td>7.5</td>
<td>7.3</td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2014) (1 net)</td>
<td>-</td>
<td>-</td>
<td>7.9</td>
</tr>
<tr>
<td>GoogLeNet (Szegedy et al., 2014) (7 nets)</td>
<td>-</td>
<td>-</td>
<td>6.7</td>
</tr>
<tr>
<td>MSRA (He et al., 2014) (11 nets)</td>
<td>-</td>
<td>-</td>
<td>8.1</td>
</tr>
<tr>
<td>MSRA (He et al., 2014) (1 net)</td>
<td>27.9</td>
<td>9.1</td>
<td>9.1</td>
</tr>
<tr>
<td>Clarifai (Russakovsky et al., 2014) (multiple nets)</td>
<td>-</td>
<td>-</td>
<td>11.7</td>
</tr>
<tr>
<td>Clarifai (Russakovsky et al., 2014) (1 net)</td>
<td>-</td>
<td>-</td>
<td>12.5</td>
</tr>
<tr>
<td>Zeiler &amp; Fergus (Zeiler &amp; Fergus, 2013) (6 nets)</td>
<td>36.0</td>
<td>14.7</td>
<td>14.8</td>
</tr>
<tr>
<td>Zeiler &amp; Fergus (Zeiler &amp; Fergus, 2013) (1 net)</td>
<td>37.5</td>
<td>16.0</td>
<td>16.1</td>
</tr>
<tr>
<td>OverFeat (Sermanet et al., 2014) (7 nets)</td>
<td>34.0</td>
<td>13.2</td>
<td>13.6</td>
</tr>
<tr>
<td>OverFeat (Sermanet et al., 2014) (1 net)</td>
<td>35.7</td>
<td>14.2</td>
<td>-</td>
</tr>
<tr>
<td>Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)</td>
<td>38.1</td>
<td>16.4</td>
<td>16.4</td>
</tr>
<tr>
<td>Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)</td>
<td>40.7</td>
<td>18.2</td>
<td>-</td>
</tr>
</tbody>
</table>

- VGGNet and GoogLeNet perform at similar level
  - Comparison: human performance ~5% [Karpathy]

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
Newest Development: Residual Networks

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

GoogleNet, 22 layers (ILSVRC 2014)
Newest Development: 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...

\[ H(x) = F(x) + x \]
ImageNet Performance

ImageNet Classification top-5 error (%)

- ILSVRC'15 ResNet: 3.57%
- ILSVRC'14 GoogleNet: 6.7%
- ILSVRC'14 VGG: 7.3%
- ILSVRC'13: 11.7%
- ILSVRC'12 AlexNet: 16.4%
- ILSVRC'11: 25.8%
- ILSVRC'10: 28.2%

152 layers

22 layers, 19 layers, 8 layers, 8 layers, shallow
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.

B. Leibe
More Finegrained Classes

**PASCAL**
- birds
  - bird
- cats
  - cat
- dogs
  - dog

**ILSVRC**
- flamingo
- cock
- ruffed grouse
- quail
- partridge
- Egyptian cat
- Persian cat
- Siamese cat
- tabby
- lynx
- dalmatian
- keeshond
- miniature schnauzer
- standard schnauzer
- giant schnauzer

Image source: O. Russakovsky et al.
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

Image source: M. Zeiler, R. Fergus
Visualizing CNNs


Slide credit: Richard Turner

Image source: M. Zeiler, R. Fergus
Visualizing CNNs

Layer 3

Image source: M. Zeiler, R. Fergus
Visualizing CNNs

Layer 4

Layer 5

Image source: M. Zeiler, R. Fergus
What Does the Network React To?

• Occlusion Experiment
  - Mask part of the image with an occluding square.
  - Monitor the output
What Does the Network React To?

Input image

$p(\text{True class})$

Most probable class

Image source: M. Zeiler, R. 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.

Slide credit: Svetlana Lazebnik, Rob Fergus

Image source: M. Zeiler, R. Fergus
What Does the Network React To?

Input image

$p(\text{True class})$

Most probable class

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.

Image source: M. Zeiler, R. Fergus
What Does the Network React To?

Input image

p(True class)

Most probable class

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.

Image source: M. Zeiler, R. Fergus

Slide credit: Svetlana Lazebnik, Rob Fergus
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.

$\Rightarrow$ Network hallucinates characteristics of the learned class.

[Link to Inceptionism blog post](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
Inceptionism: Dreaming ConvNets
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

Image source: M. Zeiler, R. Fergus
Other Tasks: Detection

R-CNN: Regions with CNN features

1. Input image
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]
  - DPM 33.4% mAP
  - R-CNN: 53.7% mAP

Faster R-CNN (based on ResNets)


B. Leibe
Faster R-CNN (based on ResNets)


B. Leibe
Other Tasks: Semantic Segmentation

[Farabet et al. ICML 2012, PAMI 2013]
Semantic Segmentation

- More recent results
  - Based on an extension of ResNets

[Pohlen, Hermans, Mathias, Leibe, arXiv 2016]
Other Tasks: Face Verification

Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR 2014

Slide credit: Svetlana Lazebnik
Commercial Recognition Services

- E.g., clarifai

Try it out with your own media

Upload an image or video file under 100mb or give us a direct link to a file on the web.

- Be careful when taking test images from Google Search
  - Chances are they may have been seen in the training set...
Commercial Recognition Services
References and Further Reading

- **LeNet**

- **AlexNet**

- **VGGNet**

- **GoogLeNet**
References and Further Reading

- **ResNet**
Effect of Multiple Convolution Layers

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Slide credit: Yann LeCun