Machine Learning – Lecture 14

Convolutional Neural Networks II

07.01.2019

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

leibe@vision.rwth-aachen.de
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
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. $1000 \times 1000$ image
    - 100 filters
    - $10 \times 10$ filter size
  - $\Rightarrow$ only 10k parameters

- Result: Response map
  - Size: $1000 \times 1000 \times 100$
  - 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:
Recap: Activation Maps

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

5 × 5 filters

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

- Effect:
  - Make the representation smaller without losing too much information
  - Achieve robustness to translations
  - Pooling happens independently across each slice, preserving the number of slices

![Diagram showing single depth slice and max pool with 2x2 filters and stride 2]

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

• Recap: CNNs

• **CNN Architectures**
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNet

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

• Applications

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


Slide credit: Svetlana Lazebnik
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
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%.

### VGGNet Architectures

<table>
<thead>
<tr>
<th>ConvNet Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
</tr>
<tr>
<td>11 weight layers</td>
</tr>
<tr>
<td>conv3-64</td>
</tr>
<tr>
<td>LRN</td>
</tr>
<tr>
<td>maxpool</td>
</tr>
<tr>
<td>maxpool</td>
</tr>
<tr>
<td>maxpool</td>
</tr>
<tr>
<td>maxpool</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 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: $3 \cdot 3^2 = 27$ instead of $7^2 = 49$.
  - In addition, non-linearities in-between $3 \times 3$ layers for additional discriminativity.

- **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)
## 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/
Newer Developments: Residual Networks

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

GoogleNet, 22 layers (ILSVRC 2014)
Newer Developments: Residual Networks

AlexNet, 8 layers (ILSVRC 2012)  VGG, 19 layers (ILSVRC 2014)  ResNet, 152 layers (ILSVRC 2015)

- 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

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

**152 layers**
ILSRVC Winners

- **2010**: 28.2 - Lin et al
- **2011**: 25.8 - Sanchez & Perronnin
- **2012**: 16.4 - Krizhevsky et al (AlexNet)
- **2013**: 11.7 - Zeiler & Fergus
- **2014**: 7.3 - Simonyan & Zisserman (VGG)
- **2014**: 6.7 - Szegedy et al (GoogLeNet)
- **2015**: 3.6 - He et al (ResNet)
- **2016**: 3 - Shao et al
- **2017**: 2.3 - Hu et al (SENet)
- **Human**: 5.1

- 152 layers
- 152 layers
- 152 layers

- Shallow
- 8 layers
- 8 layers
- 19 layers
- 22 layers

Slide credit: FeiFei Li
Comparing Complexity


Figure credit: Alfredo Canziano, Adam Paszke, Eugenio Culurcello
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

- 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.

B. Leibe
Quirks and Limitations of the Data Set

- Generated from WordNet ontology
  - Some animal categories are overrepresented
  - E.g., 120 subcategories of dog breeds

⇒ 6.7% top-5 error looks all the more impressive
Topics of This Lecture

• Recap: CNNs

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

• 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
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

Slide credit: Svetlana Lazebnik, Rob Fergus
What Does the Network React To?

Input image

Total activation in most active 5\textsuperscript{th} layer feature map

Other activations from the same feature map.

Image source: M. Zeiler, R. Fergus

Slide credit: Svetlana Lazebnik, Rob 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
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
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
Inceptionism: Dreaming ConvNets

https://www.youtube.com/watch?v=IREsx-xWQ0g
Topics of This Lecture

• Recap: CNNs

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

• 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
  \[ \Rightarrow \text{State of the art accuracy already with only 6 training images} \]

Image source: M. Zeiler, R. Fergus
Transfer Learning with CNNs

1. Train on ImageNet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

I.e., swap the Softmax layer at the end
Transfer Learning with CNNs

1. Train on ImageNet

3. If you have medium sized dataset, "finetune" instead: use the old weights as initialization, train the full network or only some of the higher layers.

Retrain bigger portion of the network

Slide credit: Andrej Karpathy
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]
    - 33.4% mAP DPM
  - 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%

Slide credit: Ross Girshick
Faster R-CNN (based on ResNets)

Faster R-CNN (based on ResNets)

Object Detection Performance

![Graph showing improvements in object detection performance over years. The graph compares the mean average precision (mAP) before and after the use of deep convolutional neural networks (convnets). Key points include:

- **Before deep convnets**: A steady performance level.
- **Using deep convnets**: Significant improvements, with Faster RCNN and Fast RCNN showing notable advancements.

The graph illustrates the evolution of object detection techniques, highlighting the impact of deep learning in this domain.**
PASCAL VOC Object Detection Performance

Engines of visual recognition

- HOG, DPM (34) - shallow
- AlexNet (RCNN) (8 layers) - 58
- VGG (RCNN) (16 layers) - 66
- ResNet (Faster RCNN)* - 101 layers, 86

PASCAL VOC 2007 Object Detection mAP (%)
Most Recent Version: Mask R-CNN

Mask R-CNN Results

• Detection + Instance segmentation

• Detection + Pose estimation

Figure credit: K. He, G. Gkioxari, P. Dollar, R. Girshick
YOLO / SSD

- Idea: Directly go from image to detection scores
- Within each grid cell
  - Start from a set of anchor boxes
  - Regress from each of the B anchor boxes to a final box
  - Predict scores for each of C classes (including background)

Slide credit: FeiFei Li
YOLO-v3 Results

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
CNNs vs. FCNs

- **CNN**

- **FCN**

- **Intuition**
  - Think of FCNs as performing a sliding-window classification, producing a heatmap of output scores for each class

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

Image source: Newell et al.
Semantic Segmentation

- Current state-of-the-art
  - Based on an extension of ResNets

[Pohlen, Hermans, Mathias, Leibe, CVPR 2017]
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.

*By using the demo you agree to our terms of service

- 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
References and Further Reading

• ResNets
References: Computer Vision Tasks

- **Object Detection**
References: Computer Vision Tasks

• Semantic Segmentation