Machine Learning – Lecture 15

Convolutional Neural Networks III

10.01.2019

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: CNN Architectures

• Residual Networks
  - Detailed analysis
  - ResNets as ensembles of shallow networks

• Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification
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: 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)

Recap: 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

- **A**
  - 11 weight layers
- **A-LRN**
  - 11 weight layers
- **B**
  - 13 weight layers
- **C**
  - 16 weight layers
- **D**
  - 16 weight layers
- **E**
  - 19 weight layers

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A-LRN</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11 weight layers</td>
<td>11 weight layers</td>
<td>13 weight layers</td>
<td>16 weight layers</td>
<td>16 weight layers</td>
<td>19 weight layers</td>
</tr>
<tr>
<td>conv3-64</td>
<td>conv3-64</td>
<td>LRN</td>
<td>conv3-64</td>
<td>conv3-64</td>
<td>conv3-64</td>
<td>conv3-64</td>
</tr>
<tr>
<td>maxpool</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
<td>conv3-128</td>
</tr>
<tr>
<td>maxpool</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
<td>conv3-256</td>
</tr>
<tr>
<td>maxpool</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td>maxpool</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
<td>conv3-512</td>
</tr>
<tr>
<td>maxpool</td>
<td>FC-4096</td>
<td>FC-4096</td>
<td>FC-1000</td>
<td>soft-max</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mainly used

Image source: Simonyan & Zisserman
Recap: GoogLeNet (2014)

- Ideas:
  - Learn features at multiple scales
  - Modular structure

Image source: Szegedy et al.

(b) Inception module with dimension reductions

Auxiliary classification outputs for training the lower layers (deprecated)
Recap: Visualizing CNNs

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

Slide credit: Yann LeCun
Topics of This Lecture

• Recap: CNN Architectures

• Residual Networks
  - Detailed analysis
  - ResNets as ensembles of shallow networks

• Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification
Recap: Residual Networks

AlexNet, 8 layers (ILSVRC 2012)
11x11 conv, 96, /4, pool/2
5x5 conv, 256, pool/2
3x3 conv, 384
3x3 conv, 384
3x3 conv, 256, pool/2
fc, 4096
fc, 4096
fc, 1000

VGG, 19 layers (ILSVRC 2014)
3x3 conv, 64
3x3 conv, 64, pool/2
3x3 conv, 128
3x3 conv, 128, pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256, pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512, pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512, pool/2
fc, 4096
fc, 4096
fc, 1000

GoogleNet, 22 layers (ILSVRC 2014)

Slide credit: Kaiming He
Recap: Residual Networks

- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers

\[ H(x) = F(x) + x \]
Spectrum of Depth

- 5 layers: easy
- >10 layers: initialization, Batch Normalization
- >30 layers: skip connections
- >100 layers: identity skip connections
- >1000 layers: ?

shallow → deeper

Slide credit: Kaiming He
Spectrum of Depth

- Deeper models are more powerful
  - But training them is harder.
  - Main problem: getting the gradients back to the early layers
  - The deeper the network, the more effort is required for this.

Slide adapted from Kaiming He
Initialization

• Importance of proper initialization (Recall Lecture 12)
  - Glorot initialization for tanh nonlinearities
  - He initialization for ReLU nonlinearities
  ⇒ For deep networks, this really makes a difference!

Slide credit: Kaiming He
Batch Normalization

- Effect of batch normalization
  - Greatly improved speed of convergence
  - Often better accuracy achievable

Image source: Ioffe & Szegedy
Going Deeper

• Checklist
  - Initialization  ok
  - Batch normalization  ok

  - Are we now set?
    - Is learning better networks now as simple as stacking more layers?
Simply Stacking Layers?

- Experiment going deeper
  - Plain nets: stacking 3×3 convolution layers
  - 56-layer net has higher training error than 20-layer net

Slide credit: Kaiming He
Simply Stacking Layers?

- **General observation**
  - Overly deep networks have higher training error
  - A general phenomenon, observed in many training sets

![CIFAR-10 and ImageNet-1000 graphs](graphs.png)

- **CIFAR-10**
  - 56-layer
  - 44-layer
  - 32-layer
  - 20-layer

- **ImageNet-1000**
  - 34-layer
  - 18-layer

Slide credit: Kaiming He
Why Is That???

• A deeper model should not have higher training error!
  - Richer solution space should allow it to find better solutions

• Solution by construction
  - Copy the original layers from a learned shallower model
  - Set the extra layers as identity
  - Such a network should achieve at least the same low training error.

• Reason: Optimization difficulties
  - Solvers cannot find the solution when going deeper…

Slide credit: Kaiming He
Deep Residual Learning

- Plain net

\[ x \]
\[ \text{weight layer} \]
\[ \text{relu} \]
\[ \text{weight layer} \]
\[ \text{relu} \]
\[ H(x) \]

- \( H(x) \) is any desired mapping
- Hope the 2 weight layers fit \( H(x) \)

Slide credit: Kaiming He
Deep Residual Learning

• **Residual net**

\[ \begin{align*}
H(x) &= F(x) + x \\
F(x) &= \text{relu}(\text{weight layer}) \\
H(x) &= \text{relu}(\text{weight layer}) \\
\end{align*} \]

- \( H(x) \) is any desired mapping
- Hope the 2 weight layers fit \( H(x) \)
- Hope the 2 weight layers fit \( F(x) \)
  - Let \( H(x) = F(x) + x \)

Slide credit: Kaiming He
Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity

- If identity were optimal, it is easy to set weights as 0
- If optimal mapping is closer to identity, it is easier to find small fluctuations
- Further advantage: direct path for the gradient to flow to the previous stages

$H(x) = F(x) + x$

Slide credit: Kaiming He
Network Design

- Simple, VGG-style design
  - (Almost) all $3 \times 3$ convolutions
  - Spatial size /2 $\Rightarrow$ #filters $\cdot$ 2
    (same complexity per layer)
  - Batch normalization
    $\Rightarrow$ Simple design, just deep.
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

Slide credit: Kaiming He
PASCAL VOC Object Detection Performance

Engines of visual recognition

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

PASCAL VOC 2007 Object Detection mAP (%)
Topics of This Lecture

• Recap: CNN Architectures

• Residual Networks
  - Detailed analysis
  - ResNets as ensembles of shallow networks

• Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification
What Is The Secret Behind ResNets?

• Empirically, they perform very well, but why is that?

• He’s original explanation \([\text{He, 2016}]\)
  - ResNets allow gradients to pass through the skip connections in unchanged form.
  - This makes it possible to effectively train deeper networks.
  ⇒ Secret of success: *depth is good*

• More recent explanation \([\text{Veit, 2016}]\)
  - ResNets actually do not use deep network paths.
  - Instead, they effectively implement an ensemble of shallow network paths.
  ⇒ Secret of success: *ensembles are good*

**Idea of the Analysis**

• Unraveling ResNets
  - ResNets can be viewed as a collection of shorter paths through different subsets of the layers.
  - Deleting a layer corresponds to removing only some of those paths.

**Ordinary feedforward network**

**Residual network**

**Unraveled view**

Effect of deleting layer $f_2$

Image source: Veit et al., 2016
Effect of Deleting Layers at Test Time

- Experiments on ImageNet classification
  - When deleting a layer in VGG-Net, it breaks down completely.
  - In ResNets, deleting a single layer has almost no effect (except for the pooling layers)
  - Deleting an increasing number of layers increases the error smoothly

⇒ *Paths in a ResNet do not strongly depend on each other.*
Which Paths Are Important?

- How much does each of the paths contribute?
  - Distribution of path lengths follows a Binomial distribution
  - Sample individual paths and measure their gradient magnitude
  - Effectively, only shallow paths with 5-17 modules are used!
  - This corresponds to only 0.45% of the available paths here.

Image source: Veit et al., 2016
Summary

• The effective paths in ResNets are relatively shallow
  ➢ Effectively only 5-17 active modules

• This explains the resilience to deletion
  ➢ Deleting any single layer only affects a subset of paths (and the shorter ones less than the longer ones).

• New interpretation of ResNets
  ➢ ResNets work by creating an ensemble of relatively shallow paths
  ➢ Making ResNets deeper increases the size of this ensemble
  ➢ Excluding longer paths from training does not negatively affect the results.

Image source: Veit et al., 2016
Topics of This Lecture

• Recap: CNN Architectures

• Residual Networks
  - Detailed analysis
  - ResNets as ensembles of shallow networks

• Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification
The Learned Features are Generic

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

More 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.

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

Faster R-CNN (based on ResNets)

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
Object Detection Performance

PASCAL VOC

mean Average Precision (mAP)

Before deep convnets

Using deep convnets

year


Slide credit: Ross Girshick

B. Leibe
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
CNNs vs. FCNs

- CNN

- FCN

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

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
Learning Similarity Functions

- **Siamese Network**
  - Present the two stimuli to two identical copies of a network (with shared parameters)
  - Train them to output similar values if the inputs are (semantically) similar.

- **Used for many matching tasks**
  - Face identification
  - Stereo estimation
  - Optical flow
  - …
Extension: Triplet Loss Networks

• Learning a discriminative embedding
  ➢ Present the network with triplets of examples

    Negative  Anchor  Positive
    ![Image of a person](image.png)

    ![Image of a person](image.png)
    ![Image of a person](image.png)

  ➢ Apply triplet loss to learn an embedding \( f(\cdot) \) that groups the positive example closer to the anchor than the negative one.

\[
\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2
\]

⇒ Used with great success in Google’s FaceNet face identification
References and Further Reading

• ResNets
References: Computer Vision Tasks

• Object Detection

References: Computer Vision Tasks

• Semantic Segmentation