### 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 ($10^5$ 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%.
Recap: GoogLeNet (2014)
- Ideas:
  - Learn features at multiple scales
  - Modular structure
  - Inception module + copies
  - Auxiliary classification outputs for training the lower layers (deprecated)

Recap: Visualizing CNNs
- Feature visualization of convolutional net trained on ImageNet from [Zeller & Fergus 2013]

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

Spectrum of Depth
- 5 layers: easy
- >10 layers: initialization, Batch Normalization
- >30 layers: skip connections
- >100 layers: identity skip connections
- >1000 layers: deeper
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.

Initialization

- Importance of proper initialization (Recall Lecture 14)
  - Glorot initialization for tanh nonlinearities
  - He initialization for ReLU nonlinearities
  ⇒ For deep networks, this really makes a difference!

Batch Normalization

- Effect of batch normalization
  - Greatly improved speed of convergence

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

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

### Deep Residual Learning

- **Plain net**  
  - Plain net
  - $H(x)$ is any desired mapping
  - $H(x) = F(x)$

- **Residual net**  
  - Residual net
  - $F(x)$ is a residual mapping w.r.t. identity
  - $F(x) = F(x) + x$
  - $H(x) = F(x) + x$

### ImageNet Performance

- Network Design
  - Simple, VGG-style design  
    - (Almost) all 3x3 convolutions  
    - Spatial size /2 ⇒ #filters : 2  
    - Same complexity per layer  
    - Batch normalization
    - ⇒ Simple design, just deep.

- Deep Residual Learning
  - $F(x)$ is a residual mapping w.r.t. identity
  - $F(x) = F(x) + x$

- ImageNet Performance
  - ImageNet Classification top-5 error (%)
    - 152 layers
    - 22 layers
    - 19 layers
    - 11 layers
    - 8 layers
    - Shallow
What Is The Secret Behind ResNets?

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

• He’s original explanation [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 [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


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.
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.

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

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
  - 53.7% mAP

Slide credit: Andrej Karpathy
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.

Faster R-CNN (based on ResNets)


YOLO


Object Detection Performance

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

- **FCN**
  - 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 Image Segmentation

- **Intuition**
  - Use skip connections to preserve higher-resolution information

Semantic Segmentation

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

Other Tasks: Face Identification

- **Learning Similarity Functions**
  - Siamese Network
    - Present the network with triplets of examples
    - Apply triplet loss to learn an embedding \( f(\cdot) \) that groups the positive example closer to the anchor than the negative one.
    - Used with great success in Google's FaceNet face identification.
References and Further Reading

- ResNets

References: Computer Vision Tasks

- Object Detection

- Semantic Segmentation