Announcements

• **HW7**: Due next Tuesday
• **Midterm**: grading completed

• **Class roadmap:**
  – Today:
    • A bit more on Deep Learning
    • Summary of neural networks
  – Next:
    • Search
    • Games
    > Artificial Intelligence
Outline

• CNNs with more layers: ResNets
  – Layer problems, residual connections, identity maps
• Data Augmentation & Regularization
  – Expanding the dataset, avoiding overfitting
Last Time: CNNs

Convolutional Neural Networks:

- **Components**: convolutional layers, pooling layers (recall kernels, channels, strides, padding)
- **Architectures**: LeNet, AlexNet, VGG
- **Trend**: bigger, deeper.
LeNet

Image: 28 (height) x 28 (width) x 1 (channel)

Convolution with 5×5 kernel + 2 padding: 28×28×6
  ↓ sigmoid
Pool with 2×2 average kernel + 2 stride: 14×14×6
  ↓ sigmoid
Convolution with 5×5 kernel (no pad): 10×10×16
  ↓ flatten
Dense: 120 fully connected neurons
  ↓ sigmoid
Dense: 84 fully connected neurons
  ↓ sigmoid
Dense: 10 fully connected neurons
  ↓ Output: 1 of 10 classes

AlexNet

Image: 224 (height) x 224 (width) x 3 (channels)

Convolution with 11×11 kernel + 4 stride: 54×54×96
  ↓ ReLu
Pool with 3×3 max. kernel + 2 stride: 26×26×96
  ↓ ReLu
Convolution with 5×5 kernel + 2 pad: 26×26×256
  ↓ ReLu
Pool with 3×3 max. kernel + 2 stride: 12×12×256
  ↓ ReLu
Convolution with 3×3 kernel + 1 pad: 12×12×384
  ↓ ReLu
Convolution with 3×3 kernel + 1 pad: 12×12×384
  ↓ ReLu
Convolution with 3×3 kernel + 1 pad: 12×12×256
  ↓ ReLu
Pool with 3×3 max. kernel + 2 stride: 5×5×256
  ↓ flatten
Dense: 4096 fully connected neurons
  ↓ ReLu, dropout p=0.5
Dense: 4096 fully connected neurons
  ↓ ReLu, dropout p=0.5
Dense: 1000 fully connected neurons
  ↓ Output: 1 of 1000 classes

Credit: Wikipedia
VGG
Evolution of CNNs

ImageNet competition (error rate)

Credit: Stanford CS 231n
Simple Idea: Add More Layers

AlexNet 8 layers, VGG: 19 layers. **Add more layers**... sufficient?

- No! Some problems:
  - i) Vanishing gradients: more layers $\rightarrow$ more likely
  - ii) Instability: can’t even guarantee we learn **identity** map $f(x) = x$

Reflected in training error:

He et al: “Deep Residual Learning for Image Recognition”
ResNet Architecture

**Idea:** Residual (skip) connections help make learning easier

- Right: Example architecture
- Note: residual connections
  - Every two layers for ResNet34
- **Vastly better** performance
  - No additional parameters!
  - Records on many benchmarks

He et al: “Deep Residual Learning for Image Recognition”
More on ResNets

**Idea:** Residual (skip) connections help make learning easier

- Alleviate vanishing gradient issue
- More paths in computation graph: better information flow
Data Concerns

What if we don’t have a lot of data?

• We risk overfitting
• Avoiding overfitting: regularization methods
• Another way: Data Augmentation
Data Augmentation

Augmentation: transform + add new samples to dataset

• Transformations: based on domain
• Idea: build invariances into the model
  – Ex: if all images have same alignment, model learns to use it
• Keep the label the same!
Transformations

Examples of transformations for images

- **Crop** (and zoom)
- **Color** (change contrast/brightness)
- **Rotations** (translate, stretch, shear, etc)

Many more possibilities. Combine as well!

Q: how to deal with this at **test time**?

- A: transform, test, average
Importance of Augmentation

Data augmentation is critical for top performance!

• You should use it!
• **AlexNet**: used (many papers re-used as well)
  – Random crops, rotations, flips. **2048x** expansion!
  – Color augmentation via PCA. **1% error rate reduction**

Krizhevsky et al: “ImageNet Classification with Deep Convolutional Neural Networks”
Summary

• Intro to deeper networks (ResNets)
  – Dealing with problems by adding skip connections
• Data augmentation
Acknowledgements: Inspired by materials by Fei-Fei Li, Ranjay Krishna, Danfei Xu (Stanford CS231n)