Computer Vision II - Recognition:
Convolutional neural network

Michael Yang
Roadmap (4 lectures)

• Object Detection (26.06)

• Image Categorization (03.07)

• Convolutional neural network (10.07)

• Scene Understanding (17.07)

• Poster, Q&A (24.07)
Roadmap (last lecture)

• Image Categorization

• Bag-of-Words (BOW)

• Generative vs. Discriminative Approach

• Spatial Pyramid Matching
# Class-based recognition: Level of Detail

- **Image Categorization**
  - One or more categories per image

- **Object Class Detection**
  - Also find bounding box

- **Part-based Object Detection**
  - Find parts of the object (and in this way the full object)

- **Semantic Segmentation** (see last lecture)
  - Object-class segmentation

---

Frog (branch)

2D bounding box for each frog

Semantic segmentation implies pixel-wise accuracy.
Image Categorization - Steps

1. Training Images
2. Training
3. Training Labels
4. Feature representation
5. Classifier Training
6. Trained Classifier
Image Categorization - Steps

Training

- Training Images
- Feature representation
- Classifier Training
- Trained Classifier

Testing

- Test Image
- Feature representation
- Trained Classifier
- Prediction (APPLE)
• How well does a learned model *generalize* from the data it was trained on to a new test set?
Image Categorization - Bag of Words Approach

Object → Bag of ‘words’
Bag of Words - Overview

learning

- feature detection & representation
- image representation
- category models (and/or) classifiers

recognition

- codewords dictionary
- category decision

Category models (and/or) classifiers
Bag of Words - Image Representation

- Histogram of features assigned to each cluster

K = 174
Classifiers

Training Images

Training

Feature representation

Classifier Training

Training Labels

Trained Classifier
Two approaches

Generative approach: 
*models distributions*

Discriminative function: 
*models decision function*
Generative vs. Discriminative

Generative

• Training
  • Maximize joint likelihood of data and labels
  • Assume (or learn) probability distribution and dependency structure
  • Can impose priors

• Testing
  • \( P(\text{y}=1, \ x) / P(\text{y}=0, \ x) > t \)

• Examples
  • Foreground/background GMM
  • Naïve Bayes classifier
  • Bayesian network

Discriminative

• Training
  • Learn to directly predict the labels from the data
  • Assume form of boundary
  • Margin maximization or parameter regularization

• Testing
  • \( f(\mathbf{x}) > t \); e.g., \( \mathbf{w}^T \mathbf{x} > t \)

• Examples
  • Logistic regression
  • SVM
  • Boosted decision trees
Summary and Discussion

- **Bag of words representation:**
  - Sparse representation of object categories
  - Many Machine learning techniques can be applied (here naïve Bayes and SVM)
  - Robust to occlusion
  - Allows sharing of representation between multiple classes (via codeword dictionary)

- **Problems:**
  - Spatial distribution of visual works is not modelled.

![Images of visual works]
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution

level 0

level 1

level 2
Roadmap (this lecture)

• Shallow vs. deep architectures

• Convolutional neural network (CNN)

• Training CNN

• CNN for X
Reminder: Traditional Recognition Approach

- Features are not learned (hand-crafted)
- Trainable classifier is often generic (e.g. SVM)
Traditional Recognition Approach

• Features are key to recent progress in recognition
• Multitude of hand-designed features currently in use
  • SIFT, HOG, ..............
• Where next? Better classifiers? Or keep building more features?

Felzenszwalb, Girshick, McAllester and Ramanan, PAMI 2007
Yan & Huang
(Winner of PASCAL 2010 classification competition)
What about learning the features?

- Learn a *feature hierarchy* all the way from pixels to classifier
- Each layer extracts features from the output of previous layer
- Train all layers jointly

![Diagram showing layers of feature extraction](image)
“Shallow” vs. “deep” architectures

Traditional recognition: “Shallow” architecture

Image/Video Pixels \(\rightarrow\) Hand-designed feature extraction \(\rightarrow\) Trainable classifier \(\rightarrow\) Object Class

Deep learning: “Deep” architecture

Image/Video Pixels \(\rightarrow\) Layer 1 \(\rightarrow\) \(\cdots\) \(\rightarrow\) Layer N \(\rightarrow\) Simple classifier \(\rightarrow\) Object Class
Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.
Neural Net Events

founded by Warren McCulloch and Walter Pitts

1943

back propagation by Rumelhart and Hinton

1969

criticism by Minsky in his book “Perceptron”

1986

ImageNet classification over millions of images

2006

deep belief networks by Hinton

2012

Google CAT

2012

Convolutional neural networks
Background: Perceptrons

Input

Weights

\[
\begin{align*}
  x_1 & \quad w_1 \\
  x_2 & \quad w_2 \\
  x_3 & \quad w_3 \\
  \vdots & \quad \vdots \\
  x_d & \quad w_d
\end{align*}
\]

Output: \( \sigma(w \cdot x + b) \)

Sigmoid function:

\[
\sigma(t) = \frac{1}{1 + e^{-t}}
\]
Background: Neural Networks

basic building blocks

\[ z = \sum_{i} x_i w_i + b, \quad y = f(z) \]

where \( f \) is an activation function:

\[ f(z) = \sigma(z) = \frac{1}{1 + \exp(-z)} \]

- sigmoid is bounded between 0 and 1
- monotonically increasing
- differentiation: \( \sigma'(z) = \sigma(z) \cdot \sigma(1 - z) \)
Neural Networks

**representation**

for each neuron in the next layer:

\[
\begin{align*}
  z^{(2)}_i &= \sum_{j=1}^{n} w^{(1)}_{ij} x_j + b^{(1)}_i, \\  a^{(2)}_i &= f(z^{(2)}_i)
\end{align*}
\]

compactly:

\[
\begin{align*}
  z^{(2)} &= W^{(1)} x + b^{(1)}, \\  a^{(2)} &= f(z^{(2)})
\end{align*}
\]

**feed-forward**

for each neuron in the next layer:

\[
\begin{align*}
  z^{(2)}_1 &= w^{(1)}_{11} x_1 + w^{(1)}_{21} x_2 + w^{(1)}_{31} x_3 + b^{(1)}_1, \\  a^{(2)}_1 &= f(z^{(2)}_1)
\end{align*}
\]

compactly:

\[
\begin{align*}
  z^{(3)} &= W^{(2)} x + b^{(2)}, \\  a^{(3)} &= f(z^{(3)})
\end{align*}
\]

globally models a function:

\[
\hat{y} = h_{W,b}(x)
\]

where \( W \) and \( b \) are model parameters
Neural Networks

Supervised Learning

• define a loss function over the true label and the model prediction, such as squared error / cross entropy.

• use gradient descent to optimize the model parameter.

\[
\begin{align*}
    w_{ij}^{(l)} &:= w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} J(W, b; x, y) \\
    b_{i}^{(l)} &:= b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W, b; x, y)
\end{align*}
\]
Note on Back Propagation

\[ y = f(a_3), \quad z^{(3)} = W^{(3)} a^{(2)} + b^{(3)} \]
\[ a^{(3)} = f(z^{(3)}), \quad z^{(2)} = W^{(2)} x + b^{(2)} \]

\[ J = \frac{1}{2} (t - y)^2. \]

\[ \frac{2J}{\partial w}, \quad \frac{2J}{\partial b} \]

G.D. \[ W = W - \alpha \frac{\partial J}{\partial w} \]
\[ b = b - \alpha \frac{\partial J}{\partial b} \]
Inspiration: Neuron cells

- Axon from another cell
- Axonal arborization
- Synapse
- Nucleus
- Cell body or Soma
- Dendrite
- Synapses
Hubel/Wiesel Architecture

- Visual cortex consists of a hierarchy of *simple*, *complex*, and *hyper-complex* cells
Roadmap (this lecture)

- Shallow vs. deep architectures
- Convolutional neural network (CNN)
- Training CNN
- CNN for X
Convolutional Neural Network (CNN/Convnet)

- 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

Convolutional Neural Network (CNN/Convnet)

- Feed-forward feature extraction:
  1. Convolve input with learned filters
  2. Non-linearity
  3. Spatial pooling
  4. Normalization

- Supervised training of convolutional filters by back-propagating classification error
1. Convolution

- Dependencies are local
- Translation invariance
- Few parameters (filter weights)
- Stride can be greater than 1 (faster, less memory)
2. Non-Linearity

- Per-element (independent)
- Options:
  - Tanh
  - Sigmoid: \(1/(1+\exp(-x))\)
  - Rectified linear unit (ReLU)
    - Simplifies backpropagation
    - Makes learning faster
3. Spatial Pooling

- Sum or max
- Non-overlapping / overlapping regions
- Role of pooling:
  - Invariance to small transformations
  - Larger receptive fields (see more of input)
4. Normalization

- Within or across feature maps
- Before or after spatial pooling
Compare: SIFT Descriptor

Image Pixels → Apply oriented filters

Spatial pool (Sum)

Normalize to unit length

Feature Vector

Lowe [IJCV 2004]
Success Convnet

- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

- Simpler recognition benchmarks
  - CIFAR-10 (9.3% error [Wan et al. 2013])
  - Traffic sign recognition
    - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

- But until 2012, less good at more complex datasets
  - Caltech-101/256 (few training examples)
ImageNet Challenge 2012

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk
- Challenge: 1.2 million training images, 1000 classes

[Deng et al. CVPR 2009]

Demo

ImageNet Challenge 2012

- Similar framework to LeCun’98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data ($10^6$ vs. $10^3$ images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week
  - Better regularization for training (DropOut)

ImageNet Challenge 2012

- Krizhevsky et al. -- 16.4% error
- Next best (non-convnet) - 26.2% error
ImageNet Classification 2013 Results


![Bar Chart](chart.png)

Test error (top-5)

<table>
<thead>
<tr>
<th>Team</th>
<th>Test Error (top-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarifai (extra data)</td>
<td>0.1175</td>
</tr>
<tr>
<td>NUS</td>
<td>0.135</td>
</tr>
<tr>
<td>Andrew Howard</td>
<td>0.1525</td>
</tr>
<tr>
<td>UvA-Euvision</td>
<td>0.17</td>
</tr>
<tr>
<td>Adobe</td>
<td></td>
</tr>
<tr>
<td>CognitiveVision</td>
<td></td>
</tr>
</tbody>
</table>
Roadmap (this lecture)

- Shallow vs. deep architectures

- Convolutional neural network (CNN)

- Training CNN

- CNN for X
3 Minutes break
Training CNN

• Backpropagation + stochastic gradient descent
  • *Neural Networks: Tricks of the Trade*

• Dropout
• Data augmentation

• Initialization
  • Transfer learning
Main Idea: approximately combining exponentially many different neural network architectures efficiently

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 (val)</th>
<th>Top-5 (val)</th>
<th>Top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM on Fisher Vectors of Dense SIFT and Color Statistics</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT</td>
<td>-</td>
<td>-</td>
<td>27.3</td>
</tr>
<tr>
<td>Conv Net + dropout (Krizhevsky et al., 2012)</td>
<td>40.7</td>
<td>18.2</td>
<td>26.2</td>
</tr>
<tr>
<td>Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)</td>
<td>38.1</td>
<td>16.4</td>
<td>16.4</td>
</tr>
</tbody>
</table>

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]
Data Augmentation (Jittering)

- Create *virtual* training samples
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion

Deep Image [Wu et al. 2015]
Parametric Rectified Linear Unit

![Graphs showing the difference between linear and parametric rectified linear units.](image)

<table>
<thead>
<tr>
<th></th>
<th>team</th>
<th>top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VGG [25]</td>
<td>7.32</td>
</tr>
<tr>
<td></td>
<td>GoogLeNet [29]</td>
<td>6.66</td>
</tr>
<tr>
<td>post-competition</td>
<td>VGG [25] (arXiv v5)</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>Baidu [32]</td>
<td>5.98</td>
</tr>
<tr>
<td></td>
<td>MSRA, PReLU-nets</td>
<td>4.94</td>
</tr>
</tbody>
</table>

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [He et al. 2015]
Transfer Learning

- Improvement of learning in a new task through the **transfer of knowledge** from a related task that has already been learned.

- Weight initialization for CNN

---

Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks [Oquab et al. CVPR 2014]
Transfer Learning

CNN Features off-the-shelf: an Astounding Baseline for Recognition
[Razavian et al. 2014]
Roadmap (this lecture)

• Shallow vs. deep architectures

• Convolutional neural network (CNN)

• Training CNN

• CNN for X
Beyond classification

• Detection
• Segmentation
• Regression
• Pose estimation
• Matching patches
• Synthesis

and many more...
R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL

R-CNN [Girshick et al. CVPR 2014]
https://github.com/rbgirshick/fast-rcnn
Labeling Pixels: Semantic Labels

Fully Convolutional Networks for Semantic Segmentation [Long et al. CVPR 2015]
DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection

[Bertasius et al. CVPR 2015]
CNN for Regression

DeepPose [Toshev and Szegedy CVPR 2014]
CNN as a Similarity Measure for Matching

Stereo matching [Zbontar and LeCun CVPR 2015]
Compare patch [Zagoruyko and Komodakis 2015]

FaceNet [Schroff et al. 2015]

FlowNet [Fischer et al 2015]
Match ground and aerial images [Lin et al. CVPR 2015]
Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]
Chair Morphing

Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]
CNN for Image Restoration/Enhancement

Super-resolution
[Dong et al. ECCV 2014]

Non-blind deconvolution
[Xu et al. NIPS 2014]

Non-uniform blur estimation
[Sun et al. CVPR 2015]
CNN Reconstruction

Reconstruction from different layers

Multiple reconstructions

Understanding deep image representations by inverting them
[Mahendran and Vedaldi CVPR 2015]
Breaking CNNs

Intriguing properties of neural networks [Szegedy ICLR 2014]

Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).
What is going on?

\[ x + 0.007 \times \frac{\partial E}{\partial x} = x + \alpha \frac{\partial E}{\partial x} \]

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]
http://karpathy.github.io/2015/03/30/breaking-convnets/
What is going on?

- Gradient descent training: modify the weights to reduce classifier error
  
  \[ w \leftarrow w - \alpha \frac{\partial E}{\partial w} \]

- Adversarial examples: modify the image to increase classifier error
  
  \[ x \leftarrow x + \alpha \frac{\partial E}{\partial x} \]

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]

http://karpathy.github.io/2015/03/30/breaking-convnets/
Fooling a linear classifier

- Perceptron weight update: add a small multiple of the example to the weight vector:

  \[ w \leftarrow w + \alpha x \]

- To fool a linear classifier, add a small multiple of the weight vector to the training example:

  \[ x \leftarrow x + \alpha w \]

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]

http://karpathy.github.io/2015/03/30/breaking-convnets/
Fooling a linear classifier

http://karpathy.github.io/2015/03/30/breaking-convnets/
Breaking CNNs

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Video
Images that both CNN and Human can recognize

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]
Direct Encoding

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]
Roadmap (this lecture)

- Shallow vs. deep architectures
- Convolutional neural network (CNN)
- Training CNN
- CNN for X
Tools

• Caffe
• cuda-convnet2
• Torch
• MatConvNet
• Pylearn2
Resources

• http://deeplearning.net/

• https://github.com/ChristosChristofidis/awesome-deep-learning

• http://cs231n.stanford.edu/syllabus.html
• The Deep Learning Saga
   https://www.youtube.com/watch?v=mlXzufEk-2E

• Deep Neural Networks are Easily Fooled
  https://www.youtube.com/watch?v=M2IebCN9Ht4

• Visualizing and Understanding Deep Neural Networks
  https://www.youtube.com/watch?v=ghEmQSxT6tw