Convolutional neural networks I

September 28\textsuperscript{th}, 2018

Yong Jae Lee
UC Davis

Many slides from Rob Fergus, Svetlana Lazebnik, Jia-Bin Huang, Derek Hoiem, Adriana Kovashka, Andrej Karpathy
Standard classifiers

Nearest neighbor

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
...

Support Vector Machines

Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting

Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields

McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba
Standard classifiers

Nearest neighbor

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

10^6 examples

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
...

Support Vector Machines

Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting

Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields

McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba
Traditional Image Categorization: Training phase

Training Images

Training

Image Features

Classifier Training

Trained Classifier

Training Labels
Traditional Image Categorization: Testing phase

Training Images

Training

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Test Image

Image Features

Trained Classifier

Prediction

Outdoor
Features have been key.

**Hand-crafted**

SIFT [Loewe IJCV 04]

HOG [Dalal and Triggs CVPR 05]

SPM [Lazebnik et al. CVPR 06]

DPM [Felzenszwalb et al. PAMI 10]

Color Descriptor [Van De Sande et al. PAMI 10]
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

• Layers have (nearly) the same structure

• Train all layers jointly ("end-to-end")
Learning Feature Hierarchy

Goal: **Learn useful higher-level features** from images

Input data

Lee et al., ICML 2009; CACM 2011

Feature representation

1st layer "Edges"

2nd layer "Object parts"

3rd layer "Objects"

Pixels

Lee et al., ICML 2009; CACM 2011

Slide: Rob Fergus
Learning Feature Hierarchy

- Better performance

- Other domains (unclear how to hand engineer):
  - Kinect
  - Video
  - Multi spectral

- Feature computation time
  - Dozens of features needed for good performance
  - Prohibitive for large datasets (10’s sec /image)
“Shallow” vs. “deep” architectures

Traditional recognition: “Shallow” architecture

Image/Video Pixels → Hand-designed feature extraction → Trainable classifier → Object Class

Deep learning: “Deep” architecture

Image/Video Pixels → Layer 1 → … → Layer N → Simple classifier → Object Class
Neural network definition

- **Nonlinear** classifier
- Can approximate any continuous function to arbitrary accuracy given sufficiently many hidden units
Neural network definition

- Activations: \( a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i \)

- Nonlinear activation function \( h \) (e.g. sigmoid, RELU): \( z_j = h(a_j) \)

Figure from Christopher Bishop
Neural network definition

- **Layer 2**
  \[ a_j = \sum_{i=0}^{D} w_{ij}^{(1)} x_i \]
  \[ z_j = h(a_j) \]

- **Layer 3 (final)**
  \[ a_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \]

- **Outputs (e.g. sigmoid/softmax)**
  (binary)
  \[ y_k = \sigma(a_k) = \frac{1}{1 + \exp(-a_k)} \]
  (multiclass)
  \[ y_k = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \]

- **Putting everything together:**
  \[ y_k(x, w) = \sigma \left( \sum_{j=0}^{M} w_{kj}^{(2)} h \left( \sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right) \]
Nonlinear activation functions

**Sigmoid**

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

**tanh**

\[
tanh(x)
\]

**ReLU**

\[
\max(0, x)
\]

**Leaky ReLU**

\[
\max(0.1x, x)
\]

**Maxout**

\[
\max(w_1^T x + b_1, w_2^T x + b_2)
\]

**ELU**

\[
f(x) = \begin{cases} 
  x & \text{if } x > 0 \\
  \alpha (\exp(x) - 1) & \text{if } x \leq 0
\end{cases}
\]
Multilayer networks

- Cascade neurons together
- Output from one layer is the input to the next
- Each layer has its own sets of weights
Feed-forward networks

• Predictions are fed forward through the network to classify
Feed-forward networks

- Predictions are fed forward through the network to classify
Feed-forward networks

- Predictions are fed forward through the network to classify
Feed-forward networks

- Predictions are fed forward through the network to classify
Feed-forward networks

• Predictions are fed forward through the network to classify
Feed-forward networks

• Predictions are fed forward through the network to classify
Deep neural networks

• Lots of hidden layers
• Depth = power (usually)
Convolutional Neural Networks (CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity:
    - Neurons in a layer are only connected to a small region of the layer before it
  - **Share** weight parameters across spatial positions:
    - Learning shift-invariant filter kernels

Image credit: A. Karpathy
LeNet [LeCun et al. 1998]

- Stack multiple stages of feature extractors
- Higher stages compute more global, more invariant features
- Classification layer at the end

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]
ImageNet Challenge 2012

[Deng et al. CVPR 2009]

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

**ImageNet Challenge**: 1.2 million training images, 1000 classes

AlexNet

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

A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
AlexNet for image classification

AlexNet

Fixed input size: 224x224x3

“car”
ImageNet Classification Challenge

![Bar chart showing classification error over years:
- 2010: 0.28
- 2011: 0.26
- 2012: 0.16

AlexNet arrow pointing to the 2012 data point.

Industry Deployment

• Used in Facebook, Google, Microsoft
• Startups
• Image Recognition, Speech Recognition, ....
• Fast at test time

Taigman et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR’14
Beyond classification

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

and many more...
CNNs for Object detection

Fast-RCNN [Girshick et al. ICCV 2015]
Labeling Pixels: Semantic Labels

Fully Convolutional Networks for Semantic Segmentation [Long et al. CVPR 2015]
Labeling Pixels: Edge Detection

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]
CNN for Image Generation

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]
Questions?

See you Monday!