Deep Learning for Computer Vision
Lex Fridman
Computer Vision is Machine Learning

Supervised Learning

Unsupervised Learning

Semi-Supervised Learning

Reinforcement Learning

References: [81]
Images are Numbers

• **Regression**: The output variable takes continuous values

• **Classification**: The output variable takes class labels
  • Underneath it may still produce continuous values such as probability of belonging to a particular class.

References: [89]
Human Vision Seems Easy

**Why:** Data

**Visual perception:** 540 millions years of data

**Bipedal movement:** 230+ million years of data

**Abstract thought:** 100 thousand years of data

“Encoded in the large, highly evolved sensory and motor portions of the human brain is a billion years of experience about the nature of the world and how to survive in it.... Abstract thought, though, is a new trick, perhaps less than 100 thousand years old. We have not yet mastered it. It is not all that intrinsically difficult; it just seems so when we do it.”

Human Vision
Its structure is instructive and inspiring!

*Thalamocortical System Simulation: 8 million cortical neurons + 2 billion synapses:*

References: [118]
Visual Cortex
(Its Structure is Instructive and Inspiring)

Reference: https://www.youtube.com/watch?v=_33K1zTtoow
Computer Vision is Hard

Viewpoint variation

Scale variation

Deformation

Occlusion

Illumination conditions

Background clutter

Intra-class variation

References: [66, 69, 89]
Image Classification Pipeline

References: [81, 89]
Famous Computer Vision Datasets

MNIST: handwritten digits

ImageNet: WordNet hierarchy

CIFAR-10(0): tiny images

Places: natural scenes

References: [90, 91, 92, 93]
Let’s Build an Image Classifier for CIFAR-10

References: [89, 91]
Let’s Build an Image Classifier for CIFAR-10

Accuracy
Random: 10%
Our image-diff (with L1): 38.6%
Our image-diff (with L2): 35.4%

References: [89, 91]
K-Nearest Neighbors: Generalizing the Image-Diff Classifier

the data

NN classifier

5-NN classifier

Tuning (hyper)parameters:

References: [89]
**K-Nearest Neighbors:** Generalizing the Image-Diff Classifier

Accuracy
Random: **10%**
Training and testing on the same data: **35.4%**
7-Nearest Neighbors: ~**30%**
Human: ~**94%**
...
Convolutional Neural Networks: ~**95%**

References: [89, 94]
Reminder: Weighing the Evidence

\[
\text{output} = \begin{cases} 
0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\
1 & \text{if } \sum_j w_j x_j > \text{threshold} 
\end{cases}
\]

References: [78]
**Reminder:** “Learning” is Optimization of a Function

Ground truth for “6”:

\[ y(x) = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0)^T \]

“Loss” function:

\[ C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2 \]

References: [63, 80]
Classify and Image of a Number

Input:

\[
\begin{array}{cccccc}
\text{5} & \text{0} & \text{4} & \text{1} & \text{9} & \text{2}
\end{array}
\]

(28x28)

Network:

References: [80]
Convolutional Neural Networks

Regular neural network (fully connected):

Convolutional neural network:

Each layer takes a 3d volume, produces 3d volume with some smooth function that may or may not have parameters.

References: [95]
Convolutional Neural Networks: Layers

- **INPUT** [32x32x3] will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with three color channels R,G,B.

- **CONV** layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [32x32x12] if we decided to use 12 filters.

- **RELU** layer will apply an elementwise activation function, such as the $\max(0,x)$ thresholding at zero. This leaves the size of the volume unchanged ([32x32x12]).

- **POOL** layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12].

- **FC** (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x10], where each of the 10 numbers correspond to a class score, such as among the 10 categories of CIFAR-10. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

Layers highlighted in blue have learnable parameters.

References: [95]
Dealing with Images: Local Connectivity

Same neuron. Just more focused (narrow “receptive field”).

The parameters on a each filter are spatially “shared”
(if a feature is useful in one place, it’s useful elsewhere)

References: [95]
ConvNets: Spatial Arrangement of Output Volume

- **Depth**: number of filters
- **Stride**: filter step size (when we “slide” it)
- **Padding**: zero-pad the input

References: [95]
References: [95]
References: [95]
### References: [95]
References: [95]
Input Volume (+pad 1) (7x7x3)  
\[ x[:, :, 0] \]

Filter W0 (3x3x3)  
\[ w0[:, :, 0] \]

Filter W1 (3x3x3)  
\[ w1[:, :, 0] \]

Output Volume (3x3x2)  
\[ o[:, :, 0] \]

References: [95]
References: [95]
References: [95]
References: [95]
## Convolution

<table>
<thead>
<tr>
<th>Operation</th>
<th>Filter</th>
<th>Convolved Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>$\begin{bmatrix} 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
<td>![Image]</td>
</tr>
<tr>
<td>Edge detection</td>
<td>$\begin{bmatrix} 1 &amp; 0 &amp; -1 \ 0 &amp; 0 &amp; 0 \ -1 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>![Image]</td>
</tr>
<tr>
<td></td>
<td>$\begin{bmatrix} 0 &amp; 1 &amp; 0 \ 1 &amp; -4 &amp; 1 \ 0 &amp; 1 &amp; 0 \end{bmatrix}$</td>
<td>![Image]</td>
</tr>
<tr>
<td></td>
<td>$\begin{bmatrix} -1 &amp; -1 &amp; -1 \ -1 &amp; 8 &amp; -1 \ -1 &amp; -1 &amp; -1 \end{bmatrix}$</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

References: [124]
Convolution

References: [124]
Convolution: Representation Learning

References: [124]
ConvNets: Pooling

Single depth slice

x

1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4

max pool with 2x2 filters and stride 2

6 8
3 4

224x224x64
pool
112x112x64
downsampling
224
112

References: [95]
Same Architecture, Many Applications

This part might look different for:
- Different image classification domains
- Image captioning with recurrent neural networks
- Image object localization with bounding box
- Image segmentation with fully convolutional networks
- Image segmentation with deconvolution layers
Object Recognition
Case Study: ImageNet

References: [4]
What is ImageNet?

- **ImageNet**: dataset of 14+ million images (21,841 categories)
  - Links to images not images
- Let’s take the high level category of **fruit** as an example:
  - Total 188,000 images of fruit
  - There are 1206 Granny Smith apples:

References: [90]
What is ImageNet?

- **Dataset**: ImageNet: dataset of 14+ million images

- **Competition**: ILSVRC: ImageNet Large Scale Visual Recognition Challenge


References: [90]
ILSVRC Challenge Evaluation for Classification

• Top 5 error rate:
  • You get 5 guesses to get the correct label

• ~20% reduction in accuracy for Top 1 vs Top 5
  • Example: In 2012 AlexNet achieved

• Human annotation is a binary task: “apple” or “not apple”
• **AlexNet (2012):** First CNN (15.4%)
  • 8 layers
  • 61 million parameters

• **ZFNet (2013):** 15.4% to 11.2%
  • 8 layers
  • More filters. Denser stride.

• **VGGNet (2014):** 11.2% to 7.3%
  • Beautifully uniform:
    - 3x3 conv, stride 1, pad 1, 2x2 max pool
  • 16 layers
  • 138 million parameters

• **GoogLeNet (2014):** 11.2% to 6.7%
  • Inception modules
  • 22 layers
  • 5 million parameters
    (throw away fully connected layers)

• **ResNet (2015):** 6.7% to 3.57%
  • More layers = better performance
  • 152 layers

• **CUImage (2016):** 3.57% to 2.99%
  • Ensemble of 6 models

References: [90]
- **AlexNet (2012): First CNN (15.4%)**
  - 8 layers
  - 61 million parameters

- **ZFNet (2013): 15.4% to 11.2%**
  - 8 layers

- **VGGNet (2014): 11.2% to 7.3%**
  - Beautifully uniform: 3x3 conv, stride 1, pad 1, 2x2 max pool
  - 16 layers
  - 138 million parameters

- **GoogLeNet (2014): 11.2% to 6.7%**
  - Inception modules
  - 22 layers
  - 5 million parameters (throw away fully connected layers)

- **ResNet (2015): 6.7% to 3.57%**
  - More layers = better performance
  - 152 layers

- **CUImage (2016): 3.57% to 2.99%**
  - Ensemble of 6 models

References: [4] 
• **AlexNet (2012):** First CNN (15.4%)
  - 8 layers
  - 61 million parameters

• **ZFNet (2013):** 15.4% to 11.2%
  - 8 layers

• **VGGNet (2014):** 11.2% to 7.3%
  - Beautifully uniform: 3x3 conv, stride 1, pad 1, 2x2 max pool
  - 16 layers
  - 138 million parameters

• **GoogLeNet (2014):** 11.2% to 6.7%
  - Inception modules
  - 22 layers
  - 5 million parameters (throw away fully connected layers)

• **ResNet (2015):** 6.7% to 3.57%
  - More layers = better performance
  - 152 layers

• **CUImage (2016):** 3.57% to 2.99%
  - Ensemble of 6 models


References: [128]
• **AlexNet (2012):** First CNN (15.4%)  
  8 layers  
  61 million parameters

• **ZFNet (2013):** 15.4% to 11.2%  
  8 layers  
  More filters. Denser stride.

• **VGGNet (2014):** 11.2% to 7.3%  
  Beautifully uniform: 3x3 conv, stride 1, pad 1, 2x2 max pool  
  16 layers  
  138 million parameters

• **GoogLeNet (2014):** 11.2% to 6.7%  
  Inception modules  
  22 layers  
  5 million parameters (throw away fully connected layers)

• **ResNet (2015):** 6.7% to 3.57%  
  More layers = better performance  
  152 layers

• **CUImage (2016):** 3.57% to 2.99%  
  Ensemble of 6 models

---

**References:** [129]

• AlexNet (2012): First CNN (15.4%)
  • 8 layers
  • 61 million parameters

• ZFNet (2013): 15.4% to 11.2%
  • 8 layers
  • More filters. Denser stride.

• VGGNet (2014): 11.2% to 7.3%
  • Beautifully uniform: 3x3 conv, stride 1, pad 1, 2x2 max pool
  • 16 layers
  • 138 million parameters

• GoogLeNet (2014): 11.2% to 6.7%
  • Inception modules
  • 22 layers
  • 5 million parameters (throw away fully connected layers)

• ResNet (2015): 6.7% to 3.57%
  • More layers = better performance
  • 152 layers

• CUImage (2016): 3.57% to 2.99%
  • Ensemble of 6 models

References: [130]

This part might look different for:

- Different image classification domains
- Image captioning with recurrent neural networks
- Image object localization with bounding box
- Image segmentation with fully convolutional networks
- Image segmentation with deconvolution layers
Segmentation

References: [96]
Object Detection

R-CNN: Regions with CNN features

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

References: [97]
Applications: Image Caption Generation

a man sitting on a couch with a dog
a man sitting on a chair with a dog in his lap

1. detect words
   woman, crowd, cat, camera, holding, purple

2. generate sentences
   A purple camera with a woman.
   A woman holding a camera in a crowd.
   A woman holding a cat.

3. re-rank sentences
   #1 A woman holding a camera in a crowd.

References: [43 – Fang et al. 2015]
Applications: Image Question Answering

COCOQA 33827
What is the color of the cat?
Ground truth: black
IMG+BOW: black (0.55)
2-VIS+LSTM: black (0.73)
BOW: gray (0.40)

COCOQA 33827a
What is the color of the couch?
Ground truth: red
IMG+BOW: red (0.65)
2-VIS+LSTM: black (0.44)
BOW: red (0.39)

DAQAR 1522
How many chairs are there?
Ground truth: two
IMG+BOW: four (0.24)
2-VIS+BLSTM: one (0.29)
LSTM: four (0.19)

DAQAR 1520
How many shelves are there?
Ground truth: three
IMG+BOW: three (0.25)
2-VIS+BLSTM: two (0.48)
LSTM: two (0.21)

COCOQA 14855
Where are the ripe bananas sitting?
Ground truth: basket
IMG+BOW: basket (0.97)
2-VIS+BLSTM: basket (0.58)
BOW: bowl (0.48)

COCOQA 14855a
What are in the basket?
Ground truth: bananas
IMG+BOW: bananas (0.98)
2-VIS+BLSTM: bananas (0.68)
BOW: bananas (0.14)

DAQAR 585
What is the object on the chair?
Ground truth: pillow
IMG+BOW: clothes (0.37)
2-VIS+BLSTM: pillow (0.65)
LSTM: clothes (0.40)

DAQAR 585a
Where is the pillow found?
Ground truth: chair
IMG+BOW: bed (0.13)
2-VIS+BLSTM: chair (0.17)
LSTM: cabinet (0.79)


Code: https://github.com/renmengye/imageqa-public

References: [40]
Applications: Video Description Generation

Correct descriptions.

S2VT: A man is doing stunts on his bike.

S2VT: A herd of zebras are walking in a field.

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

S2VT: A man is cutting a piece of a pair of a paper.


Code: [https://vsubhashini.github.io/s2vt.html](https://vsubhashini.github.io/s2vt.html)

References: [41, 42]
Applications: Modeling Attention Steering

References: [35, 36]

Application: Audio Classification

This "dry" audio is what you're currently hearing...

Dry Road

Wet Road
Driving Scene Segmentation

References: [127]
End-to-End Learning of the Driving Task

References: http://cars.mit.edu/deeptesla
# Computer Vision for Intelligent Systems

<table>
<thead>
<tr>
<th>3D Scene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Extraction</td>
</tr>
<tr>
<td>Texture</td>
</tr>
<tr>
<td>Grouping</td>
</tr>
<tr>
<td>Surfaces</td>
</tr>
<tr>
<td>Interpretation</td>
</tr>
<tr>
<td>Objects</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td>Walk, touch, contemplate, smile, evade, read on, pick up, …</td>
</tr>
</tbody>
</table>

References: [120]
Open Problem: Robustness
>99.6% Confidence in the Wrong Answer

References: [67]
Open Problem: Robustness
Fooled by a Little Distortion


References: [68]
Object Category Recognition
Object **Category** Recognition
Object Category Recognition
Object Category Recognition
Object **Cat**egory Recognition

References: [121]
References

All references cited in this presentation are listed in the following Google Sheets file:

https://goo.gl/9Xhp2t