CS106B Final

8:30a next Monday (Dec 12th) in Cubberly Aud and Dink Aud.

3 hours long, open book, open notes, closed calculation device

All classes up until last Friday. More heavily weighted towards the assignments from Boggle through Trailblazer

Practice Finals go out tonight and on Wednesday night (with solutions posted the following class day)

Review session: Friday December 9, 3:30pm-5:00pm in Bishop Auditorium
Today’s Goal

1. Amazing graph application
2. Learn about one of the great ideas in modern CS
There is something going on in the world of AI
Something big (for us)...
[suspense]
Where is my robot?
Sci-Fi Has Promised Me Robots
House Cleaning Robot
House Cleaning Robot
Robots?

Body

Mind
Robots?

Body

Mind
Early Optimism 1950

1952

1955

Axioms \iff C

ATP System (theorem prover)

Yes (proof/answer)

No

Timeout
“Machines will be capable, within twenty years, of doing any work a man can do.”
–Herbert Simon, 1952
The spirit is willing but the flesh is weak.

(Russian)

The vodka is good but the meat is rotten.

The world is too complex
BRACE YOURSELVES

WINTER IS COMING
Machine Learning: Learn From Experience
Running Example
We call this problem “vision”
Vision is Hard

When a user takes a photo, the app should check whether they're in a national park...

Sure, easy GIS lookup. Gimme a few hours.

...and check whether the photo is of a bird.

I'll need a research team and five years.
Vision is Hard

You see this:

But the camera sees this:

<table>
<thead>
<tr>
<th>194</th>
<th>210</th>
<th>201</th>
<th>212</th>
<th>199</th>
<th>213</th>
<th>215</th>
<th>195</th>
<th>178</th>
<th>158</th>
<th>182</th>
<th>209</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>189</td>
<td>190</td>
<td>221</td>
<td>209</td>
<td>205</td>
<td>211</td>
<td>167</td>
<td>147</td>
<td>115</td>
<td>129</td>
<td>163</td>
</tr>
<tr>
<td>114</td>
<td>126</td>
<td>140</td>
<td>188</td>
<td>176</td>
<td>165</td>
<td>152</td>
<td>140</td>
<td>170</td>
<td>106</td>
<td>78</td>
<td>88</td>
</tr>
<tr>
<td>87</td>
<td>103</td>
<td>115</td>
<td>154</td>
<td>143</td>
<td>142</td>
<td>149</td>
<td>153</td>
<td>173</td>
<td>101</td>
<td>57</td>
<td>57</td>
</tr>
<tr>
<td>102</td>
<td>112</td>
<td>106</td>
<td>131</td>
<td>122</td>
<td>138</td>
<td>152</td>
<td>147</td>
<td>128</td>
<td>84</td>
<td>58</td>
<td>66</td>
</tr>
<tr>
<td>94</td>
<td>95</td>
<td>79</td>
<td>104</td>
<td>105</td>
<td>124</td>
<td>129</td>
<td>113</td>
<td>107</td>
<td>87</td>
<td>69</td>
<td>67</td>
</tr>
<tr>
<td>68</td>
<td>71</td>
<td>69</td>
<td>98</td>
<td>89</td>
<td>92</td>
<td>98</td>
<td>95</td>
<td>89</td>
<td>88</td>
<td>76</td>
<td>67</td>
</tr>
<tr>
<td>41</td>
<td>56</td>
<td>68</td>
<td>99</td>
<td>63</td>
<td>45</td>
<td>60</td>
<td>82</td>
<td>58</td>
<td>76</td>
<td>75</td>
<td>65</td>
</tr>
<tr>
<td>20</td>
<td>43</td>
<td>69</td>
<td>75</td>
<td>56</td>
<td>41</td>
<td>51</td>
<td>73</td>
<td>55</td>
<td>70</td>
<td>63</td>
<td>44</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>57</td>
<td>69</td>
<td>75</td>
<td>75</td>
<td>73</td>
<td>74</td>
<td>53</td>
<td>68</td>
<td>59</td>
<td>37</td>
</tr>
<tr>
<td>72</td>
<td>59</td>
<td>53</td>
<td>66</td>
<td>84</td>
<td>92</td>
<td>84</td>
<td>74</td>
<td>57</td>
<td>72</td>
<td>63</td>
<td>42</td>
</tr>
<tr>
<td>67</td>
<td>61</td>
<td>58</td>
<td>65</td>
<td>75</td>
<td>78</td>
<td>76</td>
<td>73</td>
<td>59</td>
<td>75</td>
<td>69</td>
<td>50</td>
</tr>
</tbody>
</table>
Machine Learning

Input

Raw image

Motorbikes

“Non”-Motorbikes

Learning algorithm
Input

Raw image

Motorbikes

“Non”-Motorbikes

Learning algorithm
What we want

Input

Motorbikes

"Non"-Motorbikes

Learning algorithm

E.g., Does it have Handlebars? Wheels?

Raw image

Features

Handlebars

Wheels

Pixel 1

Pixel 2

Features

Handlebars

Wheels
Human Created Features

Find edges at four orientations

Sum up edge strength in each quadrant

Final feature vector

[Andrew Ng]
Human Created Features

Images/video
- Image
- Vision features
- Detection

Audio
- Audio
- Audio features
- Speaker ID
Great idea inspired by biology
Neuron
Some Inputs are More Important
Artificial Neuron
Artificial Neuron

\[ z = \text{input1} \times \text{weight1} + \text{input2} \times \text{weight2} + \text{input3} \times \text{weight3} + \text{input4} \times \text{weight4} \]
input sum = 1 * 2 + 
 1 * 3 + 
 0 * -2 + 
 1 * 1
Sigmoid Function

\[ \frac{1}{1 + e^{-x}} \]
Formally

\[ z = \sum_i x_i w_i \]

\[ y = \frac{1}{1 + e^{-z}} \]
Logistic Regression and Neural Networks

• Single Neuron:

\[ \hat{y} = \frac{1}{1 + e^{-x_0 \theta_1 - x_2 \theta_2 - x_3 \theta_3 - x_4 \theta_4}} \]

• Neural network

\[ \hat{y} = \frac{1}{1 + e^{-x_1 \theta_1 - x_2 \theta_2 - x_3 \theta_3 - x_4 \theta_4}} \]
Biological Basis for Neural Networks

A neuron

Artificial Neuron

Your brain

Neural Network

Actually, it’s probably someone else’s brain

Piech
Java Demo

Artificial Neuron

input1
input2
input3
input4

3.5

2.5
1.0
7.0
9.0

0.97
Digit Recognition Example

Let’s make feature vectors from pictures of numbers

\[
\text{input} = [0, 0, 0, 0, \ldots, 1, 0, 0, 1, \ldots 0, 0, 1, 0] \\
\text{label} = 0
\]

\[
\text{input} = [0, 0, 1, 1, \ldots, 0, 1, 1, 0, \ldots 0, 1, 0, 0] \\
\text{label} = 1
\]
Single Neuron

This means it predicts a 0
Single Neuron

Indicates fully connected

This means it predicts a 0
Single Neuron

This means it predicts a 1
Not So Good

This means it predicts a 1
We Can Put Neurons Together

This means it predicts a 0
We Can Put Neurons Together

Look at a single “hidden” neuron

There is an adjustable parameter for every connection

This means it predicts a 0
Neural Networks are Graphs

Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction
Demonstration

http://scs.ryerson.ca/~aharley/vis/conv/
Forward Pass...
Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction
Forward Pass

Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction
Forward Pass

Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction
Forward Pass

Each node represents a neuron (or a vector of neurons)

Each edge represents the weight of the interaction
Backward Pass...
Backward Pass

The image had a 0 but we predicted a 1
We start by making our missprediction a numerical “loss”.

The image had a 0 but we predicted a 1.
Backward Pass

We start by making our missprediction a numerical “loss”.

The image had a 0 but we predicted a 1.

For each edge weight we calculate:

$$\frac{\partial \text{Loss}}{\partial \text{EdgeWeight}}$$
Gradient of output layer params

\[
\frac{\partial LL(\theta)}{\partial \theta_i(\hat{y})} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i(\hat{y})}
\]

\[
\hat{y} = \sigma \left( \sum_{j=0}^{m_n} h_j \theta_j(\hat{y}) \right)
\]

\[
\frac{\partial \hat{y}}{\partial \theta_i(\hat{y})} = \sigma \left( \sum_{j=0}^{m_n} h_j \theta_j(\hat{y}) \right) \left[ 1 - \sigma \left( \sum_{j=0}^{m_n} h_j \theta_j(\hat{y}) \right) \right] \cdot \frac{\partial}{\partial \theta_i(\hat{y})} \sum_{j=0}^{m_n} h_j \theta_j(\hat{y})
\]

\[
= \hat{y}[1 - \hat{y}] \cdot \frac{\partial}{\partial \theta_i(\hat{y})} \sum_{j=0}^{m_n} h_j \theta_j(\hat{y})
\]

\[
= \hat{y}[1 - \hat{y}] \cdot h_i
\]

That looks scarier than it is
Chain Rule Down the Graph

\[
\frac{d}{do} = \frac{d}{dp} \times \frac{d}{dp}
\]
Generalized Backprop: One of the greatest graph algorithms
model.update(data)
Works for any number of layers

Weight between two neurons
```cpp
int main() {
    // learn from data
    NeuralNet neuralNet = train();

    // test your model
    double accuracy = test();
    cout << "Train accuracy: " << accuracy;
}
```
/**
  * Function: Train
  * ----------------
  * Start with a random neural network, then based
  * on data, learn useful weights.
  */

NeuralNet train() {
  // at start neuralNet has "random" weights
  NeuralNet neuralNet = buildGraph();
  Vector<Image> trainData = loadData();

  // repeat many times..
  for(int i = 0; i < ITERS; i++) {
    // for each image, update weights
    for(Image img : trainData) {
      Vector<double> grad = neuralNet.backprop(img);
      neuralNet.update(grad * LEARNING_RATE);
    }
  }

  return neuralNet;
}
/**
 * Function: Test
 * ----------------
 * Given a trained neural network, evaluate how well it is able to predict labels.
 */
double test(NeuralNet neuralNet) {
    Vector<Image> testData = loadData();
    double correct = 0;

    // for each image make a prediction
    for(Image img : testData) {
        int prediction = predict(neuralNet, img);
        if(isCorrect(img, prediction)) {
            correct++;
        }
    }

    return correct / testData.size();
}
/**
 * Function: BuildGraph
 * --------------
 * Make and connect the nodes that constitute the
 * neural network.
 */
NeuralNet buildGraph() {
    NeuralNet graph;
    NeuralNode * l1 = new ConvNode(INPUT_SIZE, H1);
    NeuralNode * l2 = new NeuralNode(H1, H2);
    NeuralNode * l3 = new NeuralNode(H2, OUTPUT_SIZE);
    NeuralNode * loss = new LossNode();
    l1->addChild(l2);
    l2->addChild(l3);
    l3->addChild(loss);
    graph.add(l1);
    return graph;
}
Let's Train!

http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html
Neural Networks are Turing Complete

- Some datasets are not linearly separable

- These are classifiers learned by neural networks

http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html
Like lego pieces
GoogLeNet Brain

1 Trillion Artificial Neurons
GoogLeNet Brain Graph

Multiple, Multi class output

22 layers deep
The Face Neuron

Top stimuli from the test set

Optimal stimulus by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
Top stimuli from the test set

Optimal stimulus by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
Hire the smartest people in the world
Invent cat detector
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
22,000 categories

14,000,000 images

Hand-engineered features (SIFT, HOG, LBP),
Spatial pyramid, SparseCoding/Compression

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
22,000 is a lot!

... smoothhound, smoothhound shark, Mustelus mustelus American smooth dogfish, Mustelus canis Florida smoothhound, Mustelus norrisi whitetip shark, reef whitetip shark, Triaenodon obesus Atlantic spiny dogfish, Squalus acanthias Pacific spiny dogfish, Squalus suckleyi hammerhead, hammerhead shark smooth hammerhead, Sphyrna zygaena smalleye hammerhead, Sphyrna tudes shovelhead, bonnethead, bonnet shark, Sphyrna tiburo angel shark, angelfish, Squatina squatina, monkfish electric ray, crampfish, numbfish, torpedo smalltooth sawfish, Pristis pectinatus guitarfish

roughtail stingray, Dasyatis centroura butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari cownose ray, cow-nosed ray, Rhinoptera bonasus manta, manta ray, devilfish

Atlantic manta, Manta birostris devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis little skate, Raja erinacea

...
0.005% Random guess
1.5% Pre Neural Networks
? GoogLeNet

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
0.005% Random guess 1.5% Pre Neural Networks 43.9% GoogLeNet

Szegedy et al, Going Deeper With Convolutions, CVPR 2015
Vision has Social Implications

- Apoptotic
- Viable tumor region
- Necrosis

![Images of tissue samples](image)

![Comparison of Hand engineered Features and RICA](image)

Neural network
Much of perception in the brain can be explained with a single learning algorithm.
One Algorithm Hypothesis

Auditory cortex learns to see

[Roe et al., 1992]

[Andrew Ng]
One Algorithm Hypothesis

Somatosensory cortex learns to see

[Metin & Frost, 1989]

[Andrew Ng]
Piech Told Vision Was 30 Years Out
Almost perfect...
Huge Progress
Deep Reinforcement Learning

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html
Neural Network Decision Making

Predicts “utility” of each action

In this example: 4 actions

Any input (e.g., pixels)
Deep Mind Atari Games

https://www.youtube.com/watch?v=V1eYniJ0Rnk

Score compared to best human
When Do I Get a Robo Tutor?
The diagram represents a visualization of exercises in a math course. The exercises are labeled with various types, such as solving for x-intercept, solving for y-intercept, graphing linear equations, square roots, and slope of a line. The exercises are marked as either correct (grey) or incorrect (white). The page title is "Story of Riley."
Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
Story of Riley

Exercise index

Exercise Type:  
- Solving for x-intercept  
- Solving for y-intercept  
- Graphing linear equations  
- Square roots  
- Slope of a line

Answer:
- Correct  
- Incorrect

1  
10  
20  
30  
40  
50
Story of Riley

Exercise index

1 10 20 30 40 50

Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
What does Riley know?

Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
What should Riley do next?

Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
Story of Riley

Exercise Type:
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:
- Correct
- Incorrect
Given $n$ historical answers:

$x_1, x_2, \ldots, x_n$

$x_{n+1}$

Answer is a tuple:

$x_i = \{q_i, a_i\}$

Predict the next one

Knowledge Tracing
Recurrent Neural Network

$h_0 \rightarrow h_1 \quad x_1$
Recurrent Neural Network
Recurrent Neural Network
Recurrent Neural Network

$y_1 \leftrightarrow h_0 \rightarrow h_1 \rightarrow h_2 \rightarrow h_3 \rightarrow \ldots \rightarrow h_T \leftrightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_T$
Deep Knowledge Tracing

One hot encoding of question id and correct \((q_i, a_i)\)
Deep Knowledge Tracing

Prediction of all question responses

$h_0 \rightarrow h_1 \rightarrow h_2 \rightarrow h_3 \rightarrow \cdots \rightarrow h_T$

$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_T$

$y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_T$
Recurrent Neural Network

Neurons which represent a student’s changing knowledge
Recurrent Neural Network

A representation of knowledge at time 2
**Prediction Results**

Huge improvement in ability to predict for real students

RNN vs LSTM didn’t matter too much

Benchmark AUC

0.6 0.7 0.8 0.9

Khan

Huge improvement in ability to predict for real students

0.6 0.7 0.8 0.9

Khan AUC

0.6 0.7 0.8 0.9

Marginal  BKT  DKT
Learn Concept Relationships

Scatter plots

Pythagorean theorem

Systems of Equations

Exponents

Lines

Angles

Functions

Line graphs

Fractions
Learns Concept Relationships
Learns Concept Relationships

Finding the x-intercept
Piech Learns Concept Relationships

Finding the y-intercept
Learns Concept Relationships

Slope of a line
Graphing Linear Equations
Learns Concept Relationships

Graphing Systems of Equations
Path to Neural Networks

- CS106B
- CS103
- CS109
- CS221
- CS224N
- CS229

Neural Nets
The future of artificial intelligence is uncertain
You tell me.
Open questions:
Natural Language Processing?
One Shot Learning?
General AI?
How will these ideas help improve the quality of human life?
Take CS109
Please come talk to me if you want to hear more!