CS519: Deep Learning

1. Introduction

Winter 2017

Fuxin Li

With materials from Pierre Baldi, Geoffrey Hinton, Andrew Ng, Honglak Lee, Aditya Khosla, Joseph Lim
Cutting Edge of Machine Learning: Deep Learning in Neural Networks

Engineering applications:
- Computer vision
- Speech recognition
- Natural Language Understanding
- Robotics
Computer Vision – Image Classification

• Imagenet
  • Over 1 million images, 1000 classes, different sizes, avg 482x415, color
• 16.42% Deep CNN dropout in 2012
• 6.66% 22 layer CNN (GoogLeNet) in 2014
• 3.6% (Microsoft Research Asia) super-human performance in 2015

Benenson, http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html
Speech recognition on Android (2013)

Speech Recognition and Deep Learning

Posted by Vincent Vanhoucke, Research Scientist, Speech Team

The New York Times recently published an article about Google’s large scale deep learning project, which learns to discover patterns in large datasets, including... cats on YouTube!

What’s the point of building a gigantic cat detector you might ask? When you combine large amounts of data, large-scale distributed computing and powerful machine learning algorithms, you can apply the technology to address a large variety of practical problems.

With the launch of the latest Android platform release, Jelly Bean, we’ve taken a significant step towards making that technology useful: when you speak to your Android phone, chances are, you are talking to a neural network trained to recognize your speech.

Using neural networks for speech recognition is nothing new: the first proofs of concept were developed in the late...
Impact on speech recognition

![Graph showing impact on speech recognition over time, with a significant reduction after 2010.]
Deep Learning

Deep Learning Applications

• Engineering:
  • Computer Vision (e.g. image classification, segmentation)
  • Speech Recognition
  • Natural Language Processing (e.g. sentiment analysis, translation)

• Science:
  • Biology (e.g. protein structure prediction, analysis of genomic data)
  • Chemistry (e.g. predicting chemical reactions)
  • Physics (e.g. detecting exotic particles)

• and many more to come
Penetration into mainstream media

A Learning Advance in Artificial Intelligence Rivals Human Abilities

By JOHN MARKOFF  DEC. 10, 2015

Computer researchers reported artificial-intelligence advances on Thursday that surpassed human capabilities for a narrow set of vision-related tasks.
Aha...

The advances reflect the intensifying focus in Silicon Valley and elsewhere on artificial intelligence.

Last month, the Toyota Motor Corporation announced a five-year, billion-dollar investment to create a research center based next to Stanford University to focus on artificial intelligence and robotics.

Also, a formerly obscure academic conference, Neural Information Processing Systems, underway this week in Montreal, has doubled in size since the previous year and has attracted a growing list of brand-name corporate sponsors, including Apple for the first time.

“There is a sellers’ market right now — not enough talent to fill the demand from companies who need them,” said Terrence Sejnowski, the director of the Computational Neurobiology Laboratory at the Salk Institute for Biological Studies in San Diego. “Ph.D. students are getting hired out of graduate schools for salaries that are higher than faculty members who are teaching them.”
Machine learning before Deep Learning
Typical goal of machine learning

**Input: X**
- images/video
- audio
- text

**Output: Y**
- Label: “Motorcycle”
- Suggest tags
- Image search
- ...
- Speech recognition
- Music classification
- Speaker identification
- ...
- Web search
- Anti-spam
- Machine translation
- ...

(Supervised)
Machine learning:

Find $f$, so that $f(X) \approx Y$
e.g.

ML → “motorcycle”
e.g.
Why is this 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>150</td>
<td>189</td>
<td>180</td>
<td>221</td>
<td>269</td>
<td>205</td>
<td>191</td>
<td>167</td>
<td>147</td>
<td>115</td>
<td>129</td>
<td>165</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>55</td>
</tr>
<tr>
<td>102</td>
<td>112</td>
<td>105</td>
<td>135</td>
<td>125</td>
<td>136</td>
<td>152</td>
<td>147</td>
<td>123</td>
<td>96</td>
<td>66</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>97</td>
<td>69</td>
<td>67</td>
</tr>
<tr>
<td>60</td>
<td>71</td>
<td>69</td>
<td>30</td>
<td>89</td>
<td>92</td>
<td>90</td>
<td>55</td>
<td>69</td>
<td>76</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td>41</td>
<td>55</td>
<td>66</td>
<td>99</td>
<td>63</td>
<td>45</td>
<td>60</td>
<td>60</td>
<td>52</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>55</td>
<td>41</td>
<td>51</td>
<td>73</td>
<td>35</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>73</td>
<td>74</td>
<td>52</td>
<td>68</td>
<td>59</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>72</td>
<td>89</td>
<td>33</td>
<td>66</td>
<td>84</td>
<td>92</td>
<td>84</td>
<td>74</td>
<td>67</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>56</td>
<td>76</td>
<td>73</td>
<td>59</td>
<td>75</td>
<td>69</td>
<td>80</td>
</tr>
</tbody>
</table>
Raw representation

Input

Raw image

Motorbikes
“Non”-Motorbikes

Learning algorithm
Raw representation

Input

Raw image

Learning algorithm

Motorbikes

“Non”-Motorbikes

+ Motorbikes
- “Non”-Motorbikes
Raw representation

Input

Raw image

+ Motorbikes
- “Non”-Motorbikes

Learning algorithm
What we want

Input

Raw image

Features

Motorbikes
“Non”-Motorbikes

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

Feature representation

Learning algorithm
Some feature representations

- SIFT
- Spin image
- HoG
- RIFT
- Textons
- GLOH
Some feature representations

Coming up with features is often difficult, time-consuming, and requires expert knowledge.
Deep Learning: Let’s learn the representation!

object models

object parts (combination of edges)

edges

pixels
Neural Networks

• Neuron:

\[ y = g \left( b + \sum_{i} x_i w_i \right) \]

• Many stacked neurons!
Historical Remarks

The high and low tides of neural networks
1950s – 1960s The Perceptron

- The Perceptron was introduced in 1957 by Frank Rosenblatt.

Perceptron:

\[ \sum w_i x_i \cdot f(t) \]

Activation functions:

- Step Function
- Sign Function
- Sigmoid Function

Learning:

\[ y^{(t)} = f \left( \sum w_i^{(t)} x_i^{(t)} \right) \]

Update:

\[ \Delta w_i^{(t)} = \varepsilon (d^{(t)} - y^{(t)}) x_i^{(t)} \]

\[ w_i^{(t+1)} = w_i^{(t)} + \Delta w_i^{(t)} \]
1970s -- Hiatus

- Perceptrons. Minsky and Papert. 1969
  - Revealed the fundamental difficulty in linear perceptron models
  - Stopped research on this topic for more than 10 years

Back-propagate error signal to get derivatives for learning

Compare outputs with correct answer to get error signal
1990s: Universal approximators

  • Success in handwritten digits
  • Boltzmann machines
  • Network of all sorts
  • Complex mathematical techniques

• Kernel methods (1992 – 2010):
  • (Cortes, Vapnik 1995), (Vapnik 1995), (Vapnik 1998)
  • Fixed basis function
  • First paper is forced to publish under “Support Vector Networks”
Recognizing Handwritten Digits

- MNIST database
  - 60,000 training, 10,000 testing
  - Large enough for digits
  - Battlefield of the 90s

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear classifier (perceptron)</td>
<td>12.0</td>
</tr>
<tr>
<td>K-nearest-neighbors</td>
<td>5.0</td>
</tr>
<tr>
<td>Boosting</td>
<td>1.26</td>
</tr>
<tr>
<td>SVM</td>
<td>1.4</td>
</tr>
<tr>
<td>Neural Network</td>
<td>1.6</td>
</tr>
<tr>
<td>Convolutional Neural Networks</td>
<td>0.95</td>
</tr>
<tr>
<td>With automatic distortions + ensemble + many tricks</td>
<td>0.23</td>
</tr>
</tbody>
</table>
What’s wrong with backpropagation?

• It requires a lot of labeled training data
• The learning time does not scale well
• It is theoretically **the same** as kernel methods
  • Both are “universal approximators”
• It can get stuck in poor local optima
  • Kernel methods give **globally optimal** solution
• It overfits, especially with many hidden layers
  • Kernel methods have proven approaches to control overfitting
Caltech-101: Long-time computer vision struggles without enough data

- Caltech-101 dataset
  - Around 10,000 images
  - Certainly not enough!

~80% is widely considered to be the limit on this dataset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM with Pyramid Matching Kernel (2005)</td>
<td>58.2%</td>
</tr>
<tr>
<td>Spatial Pyramid Matching (2006)</td>
<td>64.6%</td>
</tr>
<tr>
<td>SVM-KNN (2006)</td>
<td>66.2%</td>
</tr>
<tr>
<td>Sparse Coding + Pyramid Matching (2009)</td>
<td>73.2%</td>
</tr>
<tr>
<td>SVM Regression w object proposals (2010)</td>
<td>81.9%</td>
</tr>
<tr>
<td>Group-Sensitive MKL (2009)</td>
<td>84.3%</td>
</tr>
<tr>
<td>Deep Learning (pretrained on Imagenet) (2014)</td>
<td>91.4%</td>
</tr>
</tbody>
</table>
2010s: Deep representation learning

• Comeback: Make it deep!
  • Learn many, many layers simultaneously
  • How does this happen?
  • Max-pooling (Weng, Ahuja, Huang 1992)
  • Stochastic gradient descent (Hinton 2002)
  • ReLU nonlinearity (Nair and Hinton 2010), (Krizhevsky, Sutskever, Hinton 2012)
    • Better understanding of subgradients
  • Dropout (Hinton et al. 2012)
  • WAY more labeled data
    • Amazon Mechanical Turk (https://www.mturk.com/mturk/welcome)
    • 1 million+ labeled data
  • A lot better computing power
    • GPU processing
Convolutions: Utilize Spatial Locality

Sobel filter

\[
\begin{matrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1
\end{matrix}
\]

\[
\begin{matrix}
+1 & +2 & +1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{matrix}
\]

Convolution

\[
\begin{pmatrix}
F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\
F_{21} & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\
F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} \\
F_{51} & F_{52} & F_{53} & F_{54} & F_{55} & F_{56} \\
F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{pmatrix}
\] \times

\[
\begin{pmatrix}
H_{11} & H_{12} & H_{13} & H_{14} & H_{15} & H_{16} \\
H_{21} & H_{22} & H_{23} & H_{24} & H_{25} & H_{26} \\
H_{31} & H_{32} & H_{33} & H_{34} & H_{35} & H_{36} \\
H_{41} & H_{42} & H_{43} & H_{44} & H_{45} & H_{46} \\
H_{51} & H_{52} & H_{53} & H_{54} & H_{55} & H_{56} \\
H_{61} & H_{62} & H_{63} & H_{64} & H_{65} & H_{66}
\end{pmatrix}
\]

= Convolution

Image transformation
Convolutional Neural Networks

**Learning filters:**

- CNN makes sense because *locality* is important for visual processing
A Convolutional Neural Network Model

Every filter is learned!
Images that respond to various filters

Layer 1

Layer 2

Layer 5

Zeiler and Fergus 2014
Recurrent Neural Network

- Temporal stability: *history always repeats itself*
  - Parameter sharing across time
What is the hidden assumption in your problem?

• Image Understanding: Spatial locality
• Temporal Models: Temporal (partial) stationarity
• How about your problem?
References


