Machine Learning and AI via Brain simulations

Andrew Ng
This talk

The idea of “deep learning.” Using brain simulations, hope to:
- Make learning algorithms much better and easier to use.
- Make revolutionary advances in machine learning and AI.

Ideas are not only mine; vision shared with many researchers:

E.g., Samy Bengio, Yoshua Bengio, Tom Dean, Nando de Freitas, Jeff Hawkins, Geoff Hinton, Yann LeCun, Honglak Lee, Tommy Poggio, Dawn Song, Josh Tenenbaum, Kai Yu, Jason Weston, ....

I believe this is our best shot at progress towards real AI.
What do we want computers to do with our data?

Images/video
- Label: “Motorcycle”
- Recognize location, activities, …

Audio
- Speech recognition
- Music classification
- Speaker identification
- …

Text
- Spam classification
- Web search
- Machine translation
- …
Computer vision is hard!
What do we want computers to do with our data?

Images/video
- Label: “Motorcycle”
- Recognize location, activities, …

Audio
- Speech recognition
- Music classification
- Speaker identification
  …

Text
- Spam classification
- Web search
- Machine translation
  …

Machine learning performs well on many of these problems, but is a lot of work. What is it about machine learning that makes it so hard to use?
Machine learning for image classification

“Motorcycle”
Machine learning for image classification

Motorcycles

Not a motorcycle

Testing:
What is this?
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>180</td>
<td>189</td>
<td>190</td>
<td>221</td>
<td>209</td>
<td>205</td>
<td>191</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 and feature representations

Input

Raw image

Motorbikes

“Non”-Motorbikes

Learning algorithm
Machine learning and feature representations

Input

Raw image

Learning algorithm

Motorbikes

“Non”-Motorbikes
Machine learning and feature representations

Input

Raw image

\[ \begin{align*}
\text{Motorbikes} & \quad \text{``Non''-Motorbikes} \\
\end{align*} \]

Learning algorithm
What we want

Input

Raw image

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

Motorbikes
“Non”-Motorbikes

Feature representation

Learning algorithm

Handlebars
Wheels

Features

pixel 2

pixel 1
Computing features in computer vision

But… we don’t have a handlebars detector. So, researchers try to hand-design features to capture various statistical properties of the image.

Find edges at four orientations

Sum up edge strength in each quadrant

Final feature vector
How is computer perception done?

- **Images/video**
  - Image
  - Vision features
  - Recognition
  - “Motorcycle”

- **Audio**
  - Audio
  - Audio features
  - Speaker ID

- **Text**
  - Text
  - Text features
  - Information retrieval, Machine translation, Text classification, etc.

Andrew Ng
Computer vision features

SIFT

GIST

HoG

Shape context

Textons

Spin image
Audio features

Spectrogram

MFCC

Flux

ZCR

Rolloff
Coming up with features is difficult, time-consuming, requires expert domain knowledge.

When working applications of learning, we spend a lot of time tuning the features.
Feature representations

Input

Feature Representation

Learning algorithm
Sensor representation in the brain

Auditory cortex learns to see.

(Same rewiring process also works for touch/somatosensory cortex.)

Seeing with your tongue

Human echolocation (sonar)

[Roe et al., 1992; BrainPort; Welsh & Blasch, 1997]
Other sensory remapping examples


Implanting a 3\textsuperscript{rd} eye.

[Nagel et al., 2005 and Wired Magazine; Constantine-Paton & Law, 2009]
Learning input representations

Rather than hand-engineering features, can we automatically learn a better way to represent images than pixels.
Learning input representations

Automatically learn a feature representation for audio.
Feature learning problem

- Given a 14x14 image patch $x$, can represent it using 196 real numbers.

- Problem: Can we find a better feature vector to represent this?

$$\begin{pmatrix} 255 \\ 98 \\ 93 \\ 87 \\ 89 \\ 91 \\ 48 \\ \vdots \end{pmatrix}$$
How does the brain process images?

Looking to the brain for inspiration:
The first visual processing step in the brain (primary visual cortex, area V1) in looks for “edges” in images.

Neuron #1 of visual cortex (model)

Neuron #2 of visual cortex (model)
Sparse coding (Olshausen & Field, 1996). Originally developed to explain early visual processing in the brain (edge detection).

Input: Images \( x^{(1)}, x^{(2)}, \ldots, x^{(m)} \) (each in \( \mathbb{R}^{n \times n} \))

Learn: Dictionary of bases \( \phi_1, \phi_2, \ldots, \phi_k \) (also \( \mathbb{R}^{n \times n} \)), so that each input \( x \) can be approximately decomposed as:

\[
x \approx \sum_{j=1}^{k} a_j \phi_j
\]

s.t. \( a_j \)'s are mostly zero ("sparse")
Sparse coding illustration

Natural Images

Learned bases $\phi_1, \ldots, \phi_{64}$: “Edges”

Test example

$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$

$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, 0.8, 0, \ldots, 0, 0.3, 0, \ldots, 0, 0.5, 0]$ (feature representation)

Compact & easily interpretable
Sparse coding applied to audio

Image shows 20 basis functions learned from unlabeled audio.
Image shows 20 basis functions learned from unlabeled audio.
Recursive sparse coding: Training on face images

Can we go further?
By recursively applying sparse coding algorithms, get higher-level features.

[Technical details: Sparse autoencoder (Bengio) or Sparse DBN (Hinton).]
Machine learning applications
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hessian + ESURF [Williems et al 2008]</td>
<td>38%</td>
</tr>
<tr>
<td>Harris3D + HOG/HOF [Laptev et al 2003, 2004]</td>
<td>45%</td>
</tr>
<tr>
<td>Cuboids + HOG/HOF  [Dollar et al 2005, Laptev 2004]</td>
<td>46%</td>
</tr>
<tr>
<td>Dense + HOG / HOF [Laptev 2004]</td>
<td>47%</td>
</tr>
<tr>
<td>Cuboids + HOG3D [Klaser 2008, Dollar et al 2005]</td>
<td>46%</td>
</tr>
<tr>
<td><strong>Unsupervised feature learning (our method)</strong></td>
<td><strong>52%</strong></td>
</tr>
</tbody>
</table>

Unsupervised feature learning significantly improves on the previous state-of-the-art. [Le et al., 2011]
Sparse coding on audio

$\approx 0.9 * + 0.7 * + 0.2 *$

Spectrogram
### Phoneme Classification (TIMIT benchmark)

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarkson and Moreno (1999)</td>
<td>77.6%</td>
</tr>
<tr>
<td>Gunawardana et al. (2005)</td>
<td>78.3%</td>
</tr>
<tr>
<td>Sung et al. (2007)</td>
<td>78.5%</td>
</tr>
<tr>
<td>Petrov et al. (2007)</td>
<td>78.6%</td>
</tr>
<tr>
<td>Sha and Saul (2006)</td>
<td>78.9%</td>
</tr>
<tr>
<td>Yu et al. (2006)</td>
<td>79.2%</td>
</tr>
<tr>
<td><strong>Unsupervised feature learning (our method)</strong></td>
<td><strong>80.3%</strong></td>
</tr>
</tbody>
</table>

Unsupervised feature learning significantly improves on the previous state-of-the-art.
State-of-the-art unsupervised feature learning
### Audio

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMIT Phone classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Clarkson et al., 1999)</td>
<td>79.6%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>80.3%</td>
<td></td>
</tr>
<tr>
<td>TIMIT Speaker identification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Reynolds, 1995)</td>
<td>99.7%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>

### Images

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR Object classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Yu and Zhang, 2010)</td>
<td>74.5%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>80.1%</td>
<td></td>
</tr>
<tr>
<td>NORB Object classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Ranzato et al., 2009)</td>
<td>94.4%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>97.0%</td>
<td></td>
</tr>
</tbody>
</table>

### Video

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hollywood2 Classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Laptev et al., 2004)</td>
<td>48%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>53%</td>
<td></td>
</tr>
<tr>
<td>KTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Wang et al., 2010)</td>
<td>92.1%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>93.9%</td>
<td></td>
</tr>
<tr>
<td>YouTube</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Liu et al., 2009)</td>
<td>71.2%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>75.8%</td>
<td></td>
</tr>
<tr>
<td>UCF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Wang et al., 2010)</td>
<td>85.6%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>86.5%</td>
<td></td>
</tr>
</tbody>
</table>

### Multimodal (audio/video)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVLetters Lip reading</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior art (Zhao et al., 2009)</td>
<td>58.9%</td>
<td></td>
</tr>
<tr>
<td>Feature learning</td>
<td>65.8%</td>
<td></td>
</tr>
</tbody>
</table>

Other unsupervised feature learning records:
- Pedestrian detection (Yann LeCun)
- Different phone recognition task (Geoff Hinton)
- PASCAL VOC object classification (Kai Yu)
### Kai Yu’s PASCAL VOC (Object recognition) result (2009)

<table>
<thead>
<tr>
<th>Class</th>
<th>Feature Learning</th>
<th>Best of Other Teams</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aeroplane</td>
<td>88.1</td>
<td>86.6</td>
<td>1.5</td>
</tr>
<tr>
<td>Bicycle</td>
<td>68.6</td>
<td>63.9</td>
<td>4.7</td>
</tr>
<tr>
<td>Bird</td>
<td>68.1</td>
<td>66.7</td>
<td>1.4</td>
</tr>
<tr>
<td>Boat</td>
<td>72.9</td>
<td>67.3</td>
<td>5.6</td>
</tr>
<tr>
<td>Bottle</td>
<td>44.2</td>
<td>43.7</td>
<td>0.5</td>
</tr>
<tr>
<td>Bus</td>
<td>79.5</td>
<td>74.1</td>
<td>5.4</td>
</tr>
<tr>
<td>Car</td>
<td>72.5</td>
<td>64.7</td>
<td>7.8</td>
</tr>
<tr>
<td>Cat</td>
<td>70.8</td>
<td>64.2</td>
<td>6.6</td>
</tr>
<tr>
<td>Chair</td>
<td>59.5</td>
<td>57.4</td>
<td>2.1</td>
</tr>
<tr>
<td>Cow</td>
<td>53.6</td>
<td>46.2</td>
<td>7.4</td>
</tr>
<tr>
<td>Diningtable</td>
<td>57.5</td>
<td>54.7</td>
<td>2.8</td>
</tr>
<tr>
<td>Dog</td>
<td>59.3</td>
<td>53.5</td>
<td>5.8</td>
</tr>
<tr>
<td>Horse</td>
<td>73.1</td>
<td>68.1</td>
<td>5.0</td>
</tr>
<tr>
<td>Motorbike</td>
<td>72.3</td>
<td>70.6</td>
<td>1.7</td>
</tr>
<tr>
<td>Person</td>
<td>85.3</td>
<td>85.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Pottedplant</td>
<td>36.6</td>
<td>39.1</td>
<td>-2.5</td>
</tr>
<tr>
<td>Sheep</td>
<td>56.9</td>
<td>48.2</td>
<td>8.7</td>
</tr>
<tr>
<td>Sofa</td>
<td>57.9</td>
<td>50.0</td>
<td>7.9</td>
</tr>
<tr>
<td>Train</td>
<td>86.0</td>
<td>83.4</td>
<td>2.6</td>
</tr>
<tr>
<td>Tvmonitor</td>
<td>68.0</td>
<td>68.6</td>
<td>-0.6</td>
</tr>
</tbody>
</table>

- Sparse coding to learn features.
- Unsupervised feature learning beat all the other approaches by a significant margin.

[Courtesy of Kai Yu]
Weaknesses & Criticisms
Weaknesses & Criticisms

• You’re learning everything. It’s better to encode prior knowledge about structure of images (or audio, or text).

A: There was a similar linguists vs. machine learning/IR debate in NLP ~20 years ago….

• Deep learning cannot currently do X, where X is:
  - Go beyond Gabor (1 layer) features.
  - Work on temporal data (video).
  - Learn hierarchical representations (compositional semantics).
  - Get state-of-the-art in activity recognition.
  - Get state-of-the-art on image classification.
  - Get state-of-the-art on object detection.
  - Learn variable-size representations.

A: Many of these were true, but not anymore (were not fundamental weaknesses). There’s still work to be done though!

• We don’t understand the learned features.

A: True. Though many vision/audio/text features are also not really human-understandable (e.g, concatenations/combinations of different features). There’re also techniques to bound the parts we don’t understand.
Technical challenge: Scaling up
Current models simulate $10^3 - 10^6$ neurons.

The brain has $10^{11}$ neurons.

Larger models:
- Simulate cortical neuronal properties more accurately.
- Work much better on machine learning tasks.
Machine learning application (CIFAR-10)

Accuracy vs. model size (#features).
Summary
• Lets learn rather than manually design our features.
• Sparse coding and Deep Learning is best method currently for many tasks.
• Discover the fundamental computational principles that underlie perception, make progress on AI.
• Key challenge: Scalability.
• If interested in (i) collaborating on these topics, or (ii) learning more about deep learning, contact me (email ang@cs.stanford.edu). Online tutorials available to Computer forum members.

Thanks to:

Adam Coates  Quoc Le  Honglak Lee  Andrew Saxe  Andrew Maas  Chris Manning  Jiquan Ngiam  Richard Socher  Will Zou