Unsupervised Feature Learning and Deep Learning

Andrew Ng

Thanks to:

Adam Coates          Quoc Le           Honglak Lee    Andrew Maas   Chris Manning  Jiquan Ngiam
Andrew Saxe          Richard Socher
Develop ideas using...

Computer vision

Audio

Text
Feature representations

Input

Learning algorithm
Feature representations

Input
E.g., SIFT, HoG, etc.
Feature representations

Input → Feature Representation → Learning algorithm
How is computer perception done?

- **Object detection**
  - Image
  - Vision features
  - Detection

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

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

Input → Feature Representation → Learning algorithm
Computer vision features

- SIFT
- Spin image
- HoG
- RIFT
- Textons
- GLOH
Audio features

Spectrogram

MFCC

Flux

ZCR

Rolloff
NLP features

**Parser features**

**NER/SRL**

**Stemming**

**Anaphora**

**POS tagging**

**WordNet features**

His father, Nick Begich, won an election posthumously, only they didn’t know for sure that it was posthumous because his plane just disappeared. It still hasn’t turned up. It’s why locators are now required in all US planes.
Feature representations

Input

Feature Representation

Learning algorithm

Andrew Ng
Sensor representation in the brain

Auditory cortex learns to see.

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

[Andrew Ng]

Seeing with your tongue

Human echolocation (sonar)

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


Implanting a 3rd eye.

[Nagel et al., 2005 and Wired Magazine; Constantine-Paton & Law, 2009]
On two approaches to computer perception

The adult visual (or audio) system is incredibly complicated.

We can try to directly implement what the adult visual (or audio) system is doing. (E.g., implement features that capture different types of invariance, 2d and 3d context, relations between object parts, …).

Or, if there is a more general computational principal/algorithim that underlies most of perception, can we instead try to discover and implement that?
Find a better way to represent images than pixels.
Learning input representations

Find a better way to represent 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?

- \[
\begin{pmatrix}
255 \\
98 \\
93 \\
87 \\
89 \\
91 \\
48 \\
...
\end{pmatrix}
\]
Supervised Learning: Recognizing motorcycles

Motorcycles

Not motorcycles

Testing:
What is this?

[Lee, Raina and Ng, 2006; Raina, Lee, Battle, Packer & Ng, 2007]
Self-taught learning (Feature learning problem)

Motorcycles

Not motorcycles

Unlabeled images

Testing:
What is this?

[Lee, Raina and Ng, 2006; Raina, Lee, Battle, Packer & Ng, 2007]
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 ($\phi_1, ..., \phi_{64}$): “Edges”

Test example

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

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

Compact & easily interpretable
More examples

Represent as: \[a_{15} = 0.6, a_{28} = 0.8, a_{37} = 0.4\].

Represent as: \[a_{5} = 1.3, a_{18} = 0.9, a_{29} = 0.3\].

- Method “invents” edge detection.
- Automatically learns to represent an image in terms of the edges that appear in it. Gives a more succinct, higher-level representation than the raw pixels.
- Quantitatively similar to primary visual cortex (area V1) in brain.
Sparse coding applied to audio

Image shows 20 basis functions learned from unlabeled audio.

[Evan Smith & Mike Lewicki, 2006]
Sparse coding applied to audio

Image shows 20 basis functions learned from unlabeled audio.

[Andrew Ng]

[Evan Smith & Mike Lewicki, 2006]
Sparse coding applied to touch data

Collect touch data using a glove, following distribution of grasps used by animals in the wild.

Example learned representations

[Saxe, Bhand, Mudur, Suresh & Ng, 2011]
Learning feature hierarchies

Higher layer
(Combinations of edges; cf. V2)

“Sparse coding”
(edges; cf. V1)

Input image (pixels)

[Technical details: Sparse autoencoder or sparse version of Hinton’s DBN.]

[Lee, Ranganath & Ng, 2007]
Learning feature hierarchies

Input image

Model V1

Higher layer (Model V2?)

Higher layer (Model V3?)

[Technical details: Sparse autoencoder or sparse version of Hinton’s DBN.]

[Lee, Ranganath & Ng, 2007]
Sparse DBN: Training on face images

[Lee, Grosse, Ranganath & Ng, 2009]
Sparse DBN

Features learned from different object classes.

[Lee, Grosse, Ranganath & Ng, 2009]
Training on multiple objects

Features learned by training on 4 classes (cars, faces, motorbikes, airplanes).

Object specific features

Features shared across object classes

Edges

[Lee, Grosse, Ranganath & Ng, 2009]
Machine learning applications
## Activity recognition (Hollywood 2 benchmark)

<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, Zhou & Ng, 2011]
Sparse coding on audio

\[ x \approx 0.9 \phi_{36} + 0.7 \phi_{42} + 0.2 \phi_{63} \]

[Lee, Pham and Ng, 2009]
Dictionary of bases $\phi_i$ learned for speech

Many bases seem to correspond to phonemes.

[Lee, Pham and Ng, 2009]
Sparse DBN for audio

Spectrogram

[Lee, Pham and Ng, 2009]
Sparse DBN for audio

[Spectrogram - Lee, Pham and Ng, 2009]
Sparse DBN for audio

[Lee, Pham and Ng, 2009]
## 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.

[Lee, Pham and Ng, 2009]
Technical challenge: Scaling up
Large numbers of features is critical. Algorithms that can scale to many features have a big advantage.

[Coates, Lee and Ng, 2010]
Approaches to scaling up

• Efficient sparse coding algorithms. (Lee et al., NIPS 2006)

• Parallel implementations via Map-Reduce (Chu et al., NIPS 2006)

• GPUs for deep learning. (Raina et al., ICML 2008)

• Tiled Convolutional Networks (Le et al., NIPS 2010)
  – The scaling advantage of convolutional networks, but without hard-coding translation invariance.

• Efficient optimization algorithms (Le et al., ICML 2011)

• Simple, fast feature decoders (Coates et al., AISTATS 2011)
State-of-the-art
Unsupervised feature learning
<table>
<thead>
<tr>
<th>Audio</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMIT Phone classification</td>
<td>80.3%</td>
</tr>
<tr>
<td>Prior art (Clarkson et al., 1999)</td>
<td>79.6%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td>80.3%</td>
</tr>
<tr>
<td>Prior art (Reynolds, 1995)</td>
<td>99.7%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

| Images |
| CIFAR Object classification | Accuracy |
| Prior art (Krizhevsky, 2010) | 78.9% |
| Stanford Feature learning | 81.5% |
| NORB Object classification | Accuracy |
| Prior art (Ranzato et al., 2009) | 94.4% |
| Stanford Feature learning | 97.3% |

| Video |
| Hollywood2 Classification | Accuracy |
| Prior art (Laptev et al., 2004) | 48% |
| Stanford Feature learning | 53% |
| KTH | Accuracy |
| Prior art (Wang et al., 2010) | 92.1% |
| Stanford Feature learning | 93.9% |
| YouTube | Accuracy |
| Prior art (Liu et al., 2009) | 71.2% |
| Stanford Feature learning | 75.8% |
| UCF | Accuracy |
| Prior art (Wang et al., 2010) | 85.6% |
| Stanford Feature learning | 86.5% |

| Multimodal (audio/video) |
| AVLetters Lip reading | Accuracy |
| Prior art (Zhao et al., 2009) | 58.9% |
| Stanford Feature learning | 65.8% |

Other unsupervised feature learning records:
- Pedestrian detection (Yann LeCun)
- Different phone recognition task (Geoff Hinton)
- PASCAL VOC object classification (Kai Yu)
<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.
Learning Recursive Representations
Imagine taking each word, and embedding it in an n-dimensional space. (cf. distributional representations, or Bengio et al., 2003; Collobert & Weston, 2008).

2-d embedding example below, but in practice use ~100-d embeddings.

On Monday, Britain ...

Representation:

\[
\begin{pmatrix}
8 \\
5 \\
2 \\
4 \\
9 \\
2
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 \\
0 \\
0 \\
1 \\
0 \\
0 \\
0
\end{pmatrix}
\]
"Generic" hierarchy on text doesn’t make sense

Node has to represent sentence fragment “cat sat on.” Doesn’t make sense.

Feature representation for words
What we want (illustration)

This node’s job is to represent “on the mat.”
What we want (illustration)

This node’s job is to represent “on the mat.”
The day after my birthday, …

The country of my birth, …
Learning recursive representations

This node’s job is to represent “on the mat.”
Learning recursive representations

This node’s job is to represent “on the mat.”
Learning recursive representations

Basic computational unit: Recursive Neural Network that inputs two children nodes’ feature representations, and outputs the representation of the parent node.

This node’s job is to represent “on the mat.”
The cat on the mat.

[Socher, Manning & Ng]
The cat on the mat.

Parsing a sentence

[Socher, Manning & Ng]
 Parsing a sentence

[Socher, Manning & Ng]
Finding Similar Sentences

- Each sentence has a feature vector representation.
- Pick a sentence (“center sentence”) and list nearest neighbor sentences.
- Often either semantically or syntactically similar. (Digits all mapped to 2.)

<table>
<thead>
<tr>
<th>Similarities</th>
<th>Center Sentence</th>
<th>Nearest Neighbor Sentences (most similar feature vector)</th>
</tr>
</thead>
</table>
| Bad News              | Both took further hits yesterday                                                 | 1. We’re in for a lot of turbulence ...
                                                                         2. BSN currently has 2.2 million common shares outstanding
                                                                         3. This is panic buying
                                                                         4. We have a couple or three tough weeks coming |
| Something said        | I had calls all night long from the States, he said                             | 1. Our intent is to promote the best alternative, he says
                                                                         2. We have sufficient cash flow to handle that, he said
                                                                         3. Currently, average pay for machinists is 22.22 an hour, Boeing said
                                                                         4. Profit from trading for its own account dropped, the securities firm said |
| Gains and good news   | Fujisawa gained 22 to 2,222                                                     | 1. Mochida advanced 22 to 2,222
                                                                         2. Commerzbank gained 2 to 222.2
                                                                         3. Paris loved her at first sight
                                                                         4. Profits improved across Hess’s businesses |
| Unknown words         | Columbia, S.C                                                                     | 1. Greenville, Miss
                                                                         2. UNK, Md |

## Finding Similar Sentences

<table>
<thead>
<tr>
<th>Similarities</th>
<th>Center Sentence</th>
<th>Nearest Neighbor Sentences (most similar feature vector)</th>
</tr>
</thead>
</table>
| Declining to comment = not disclosing| Hess declined to comment                                                        | 1. PaineWebber declined to comment  
2. Phoenix declined to comment  
3. Campeau declined to comment  
4. Coastal wouldn't disclose the terms |
| Large changes in sales or revenue    | Sales grew almost 2 % to 222.2 million from 222.2 million                       | 1. Sales surged 22 % to 222.22 billion yen from 222.22 billion  
2. Revenue fell 2 % to 2.22 billion from 2.22 billion  
3. Sales rose more than 2 % to 22.2 million from 22.2 million  
4. Volume was 222.2 million shares, more than triple recent levels |
| Negation of different types          | There's nothing unusual about business groups pushing for more government spending | 1. We don't think at this point anything needs to be said  
2. It therefore makes no sense for each market to adopt different circuit breakers  
3. You can't say the same with black and white  
4. I don't think anyone left the place UNK UNK |
| People in bad situations             | We were lucky                                                                    | 1. It was chaotic  
2. We were wrong  
3. People had died |

*UNK refers to unknown text.*
Experiments

- No linguistic features. Train only using the structure and words of WSJ training trees, and word embeddings from (Collobert & Weston, 2008).
- Parser evaluation dataset: Wall Street Journal (standard splits for training and development testing).

<table>
<thead>
<tr>
<th>Method</th>
<th>Unlabeled F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy Recursive Neural Network (RNN)</td>
<td>76.55</td>
</tr>
<tr>
<td>Greedy, context-sensitive RNN</td>
<td>83.36</td>
</tr>
<tr>
<td>Greedy, context-sensitive RNN + category classifier</td>
<td>87.05</td>
</tr>
<tr>
<td>Left Corner PCFG, (Manning and Carpenter, '97)</td>
<td>90.64</td>
</tr>
<tr>
<td>CKY, context-sensitive, RNN + category classifier (our work)</td>
<td>92.06</td>
</tr>
<tr>
<td>Current Stanford Parser, (Klein and Manning, '03)</td>
<td>93.98</td>
</tr>
</tbody>
</table>
A small crowd quietly enters the historic church.

Each node in the hierarchy has a “feature vector” representation.
Nearest neighbor examples for image patches

- Each node (e.g., set of merged superpixels) in the hierarchy has a feature vector.
- Select a node (“center patch”) and list nearest neighbor nodes.
- I.e., what image patches/superpixels get mapped to similar features?
Multi-class segmentation (Stanford background dataset)

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel CRF (Gould et al., ICCV 2009)</td>
<td>74.3</td>
</tr>
<tr>
<td>Classifier on superpixel features</td>
<td>75.9</td>
</tr>
<tr>
<td>Region-based energy (Gould et al., ICCV 2009)</td>
<td>76.4</td>
</tr>
<tr>
<td>Local labelling (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>76.9</td>
</tr>
<tr>
<td>Superpixel MRF (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>77.5</td>
</tr>
<tr>
<td>Simultaneous MRF (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>77.5</td>
</tr>
<tr>
<td>Feature learning (our method)</td>
<td>78.1</td>
</tr>
</tbody>
</table>
Multi-class Segmentation MSRC dataset: 21 Classes

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextonBoost (Shotton et al., ECCV 2006)</td>
<td>72.2</td>
</tr>
<tr>
<td>Framework over mean-shift patches (Yang et al., CVPR 2007)</td>
<td>75.1</td>
</tr>
<tr>
<td>Pixel CRF (Gould et al., ICCV 2009)</td>
<td>75.3</td>
</tr>
<tr>
<td>Region-based energy (Gould et al., IJCV 2008)</td>
<td>76.5</td>
</tr>
<tr>
<td><strong>Feature learning (out method)</strong></td>
<td><strong>76.7</strong></td>
</tr>
</tbody>
</table>
Weaknesses & Criticisms
Weaknesses & Criticisms

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

A: Wasn’t there a similar machine learning vs. linguists debate in NLP ~20 years ago….

• Unsupervised feature 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 features are also not really human-understandable (e.g, concatenations/combinations of different features).
Conclusion
Unsupervised feature learning summary

- Unsupervised feature learning.
- Lets learn rather than manually design our features.
- Discover the fundamental computational principles that underlie perception?
- Sparse coding and deep versions very successful on vision and audio tasks. Other variants for learning recursive representations.
- Online tutorial for applying algorithms: http://ufldl.stanford.edu/wiki, or email me.

Thanks to:

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