Deep Learning

Andrew Ng

Thanks to: Adam Coates, Quoc Le, Brody Huval, Andrew Saxe, Andrew Maas, Richard Socher, Tao Wang
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.

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

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 and feature representations

Input

Raw image

Motorbikes

“Non”-Motorbikes

Learning algorithm

pixel 1

pixel 2

pixel 1

pixel 2
Machine learning and feature representations

Input

Raw image

Motorbikes
“Non”-Motorbikes

Learning algorithm
Machine learning and feature representations

Input

Raw image

Motorbikes
“Non”-Motorbikes

Learning algorithm
What we want

Input

Raw image

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

Motorbikes

“Non”-Motorbikes

Feature representation

Learning algorithm

Handlebars

Wheels
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

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

When working applications of learning, we spend a lot of time tuning the features.
The “one learning algorithm” hypothesis

Auditory cortex learns to see

[Andrew Ng]

[Roe et al., 1992]
The “one learning algorithm” hypothesis

Somatosensory cortex learns to see

Somatosensory cortex learns to see

[Metin & Frost, 1989]
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?
Feature Learning via Sparse Coding

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)

More succinct, higher-level, representation.
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.

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

Image shows 20 basis functions learned from unlabeled audio.

[Evans Smith & Mike Lewicki, 2006]
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

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

[Lee, Ranganath & Ng, 2007]
Hierarchical Sparse coding (Sparse DBN): Trained on face images

Training set: Aligned images of faces.

- Pixels
- Edges
- Object parts (combination of edges)
- Object models
State-of-the-art
Unsupervised feature learning
## Images

<table>
<thead>
<tr>
<th>CIFAR Object classification</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Ciresan et al., 2011)</td>
<td>80.5%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>82.0%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NORB Object classification</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Scherer et al., 2010)</td>
<td>94.4%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>95.0%</strong></td>
</tr>
</tbody>
</table>

## Video

<table>
<thead>
<tr>
<th>Hollywood2 Classification</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Laptev et al., 2004)</td>
<td>48%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>53%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KTH</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Wang et al., 2010)</td>
<td>92.1%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>93.9%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YouTube</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Liu et al., 2009)</td>
<td>71.2%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>75.8%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCF</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Wang et al., 2010)</td>
<td>85.6%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>86.5%</strong></td>
</tr>
</tbody>
</table>

## Text/NLP

<table>
<thead>
<tr>
<th>Paraphrase detection</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Das &amp; Smith, 2009)</td>
<td>76.1%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>76.4%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentiment (MR/MPQA data)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior art (Nakagawa et al., 2010)</td>
<td>77.3%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>77.7%</strong></td>
</tr>
</tbody>
</table>

## Multimodal (audio/video)

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

Other unsupervised feature learning records:
- Pedestrian detection (Yann LeCun)
- Speech recognition (Geoff Hinton)
- PASCAL VOC object classification (Kai Yu)
Technical challenge: Scaling up
Large numbers of features is critical. The specific learning algorithm is important, but ones that can scale to many features also have a big advantage.
Scaling up: Discovering object classes

[Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Greg Corrado, Matthieu Devin, Kai Chen, Jeff Dean]
Local Receptive Field networks

Machine #1

Machine #2

Machine #3

Machine #4

Sparse features

Image

Le, et al., *Tiled Convolutional Neural Networks*. NIPS 2010
Asynchronous Parallel SGD

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
Training procedure

What features can we learn if we train a massive model on a massive amount of data. Can we learn a “grandmother cell”?

- Train on 10 million images (YouTube)
- 1000 machines (16,000 cores) for 1 week.
- 1.15 billion parameters
- Test on novel images

Training set (YouTube)  Test set (FITW + ImageNet)
Face neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization
Cat neuron

Top Stimuli from the test set

Average of top stimuli from test set
ImageNet classification

20,000 categories

16,000,000 images

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

Le, et al., Building high-level features using large-scale unsupervised learning. ICML 2012
20,000 is a lot of categories...

... 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
guitarfish, 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
...
Random guess: 0.005%  
State-of-the-art: 9.5%  
Feature learning: ?

ImageNet 2009 (10k categories): Best published result: 17%  
(Sanchez & Perronnin ‘11),  
Our method: **20%**

Using only 1000 categories, our method > 50%

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
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
Unsupervised Feature Learning Summary

• Deep Learning: Lets learn rather than manually design our features.
• Discover the fundamental computational principles that underlie perception.
• Deep learning very successful on vision and audio tasks.
• Other variants for learning recursive representations for text.

Thanks to: Adam Coates, Quoc Le, Brody Huval, Andrew Saxe, Andrew Maas, Richard Socher, Tao Wang
Unsupervised Feature Learning Summary

• Deep Learning: Lets learn rather than manually design our features.
• Discover the fundamental computational principles that underlie perception.
• Deep learning very successful on vision and audio tasks.
• Other variants for learning recursive representations for text.

Thanks to: Adam Coates, Quoc Le, Brody Huval, Andrew Saxe, Andrew Maas, Richard Socher, Tao Wang
Conclusion
Deep Learning Summary

• Deep Learning and Self-Taught learning: Lets learn rather than manually design our features.
• Discover the fundamental computational principles that underlie perception?
• Deep learning very successful on vision and audio tasks.
• Other variants for learning recursive representations for text.
Advanced Topics

Andrew Ng

Stanford University & Google
Analysis of feature learning algorithms

Andrew Coates   Honglak Lee
Supervised Learning

- Choices of learning algorithm:
  - Memory based
  - Winnow
  - Perceptron
  - Naïve Bayes
  - SVM
  - ....

- What matters the most?

“\textquote{It’s not who has the best algorithm that wins. It’s who has the most data.}”

[Banko & Brill, 2001]
Unsupervised Feature Learning

• Many choices in feature learning algorithms;
  – Sparse coding, RBM, autoencoder, etc.
  – Pre-processing steps (whitening)
  – Number of features learned
  – Various hyperparameters.

• What matters the most?
Unsupervised feature learning

Most algorithms learn Gabor-like edge detectors.

Sparse auto-encoder
Unsupervised feature learning

Weights learned with and without whitening.

Sparse auto-encoder
- With whitening
- Without whitening

Sparse RBM
- With whitening
- Without whitening

K-means
- With whitening
- Without whitening

Gaussian mixture model
- With whitening
- Without whitening
Scaling and classification accuracy (CIFAR-10)
Results on CIFAR-10 and NORB (old result)

- K-means achieves state-of-the-art
  - Scalable, fast and almost parameter-free, K-means does surprisingly well.

<table>
<thead>
<tr>
<th>CIFAR-10 Test accuracy</th>
<th>NORB Test accuracy (error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw pixels</td>
<td>Convolutional Neural Networks 93.4% (6.6%)</td>
</tr>
<tr>
<td>RBM with back-propagation</td>
<td>Deep Boltzmann Machines 92.8% (7.2%)</td>
</tr>
<tr>
<td>3-Way Factored RBM (3 layers)</td>
<td>Deep Belief Networks 95.0% (5.0%)</td>
</tr>
<tr>
<td>Mean-covariance RBM (3 layers)</td>
<td>Jarrett et al., 2009 94.4% (5.6%)</td>
</tr>
<tr>
<td>Improved Local Coordinate Coding</td>
<td>Sparse auto-encoder 96.9% (3.1%)</td>
</tr>
<tr>
<td>Convolutional RBM</td>
<td>Sparse RBM 96.2% (3.8%)</td>
</tr>
<tr>
<td>Sparse auto-encoder</td>
<td>K-means (Hard) 96.9% (3.1%)</td>
</tr>
<tr>
<td>Sparse RBM</td>
<td>K-means (Triangle) 97.0% (3.0%)</td>
</tr>
<tr>
<td>K-means (Hard)</td>
<td></td>
</tr>
<tr>
<td>K-means (Triangle, 1600 features)</td>
<td></td>
</tr>
<tr>
<td>K-means (Triangle, 4000 features)</td>
<td></td>
</tr>
</tbody>
</table>
Tiled Convolution Neural Networks

Quoc Le

Jiquan Ngiam
Learning Invariances

• We want to learn invariant features.

• Convolutional networks uses weight tying to:
  – Reduce number of weights that need to be learned. 
    → Allows scaling to larger images/models.
  – Hard code translation invariance. Makes it harder to learn more complex types of invariances.

• Goal: Preserve computational scaling advantage of convolutional nets, but learn more complex invariances.
Fully Connected Topographic ICA

Doesn’t scale to large images.
Fully Connected Topographic ICA

Doesn’t scale to large images.
Local Receptive Fields

Pooling Units (Sqrt)

Simple Units (Square)

Input
Convolution Neural Networks (Weight Tying)

Pooling Units (Sqrt)

Simple Units (Square)

Input
Local pooling can capture complex invariances (not just translation); but total number of parameters is small.
Tiled Networks (Partial Weight Tying)

Pooling Units (Sqrt)

Tile Size \((k) = 2\)

Simple Units (Square)

Input
Tiled Networks (Partial Weight Tying)

Number of Maps ($l$) = 3

Tile Size ($k$) = 2

Pooling Units (Sqrt)

Simple Units (Square)

Input
Tiled Networks (Partial Weight Tying)

- **Input**
- **Simple Units** (Square)
- **Pooling Units** (Sqrt)
- **Tile Size** \( (k) = 2 \)
- **Number of Maps** \( (l) = 3 \)
- **Local Orthogonalization**

- Number of Maps: 3
### NORB and CIFAR-10 results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>NORB Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Tiled CNNs [this work]</td>
<td>96.1%</td>
</tr>
<tr>
<td>CNNs [Huang &amp; LeCun, 2006]</td>
<td>94.1%</td>
</tr>
<tr>
<td>3D Deep Belief Networks [Nair &amp; Hinton, 2009]</td>
<td>93.5%</td>
</tr>
<tr>
<td>Deep Boltzmann Machines [Salakhutdinov &amp; Hinton, 2009]</td>
<td>92.8%</td>
</tr>
<tr>
<td>TICA [Hyvarinen et al., 2001]</td>
<td>89.6%</td>
</tr>
<tr>
<td>SVMs</td>
<td>88.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>CIFAR-10 Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved LCC [Yu et al., 2010]</td>
<td>74.5%</td>
</tr>
<tr>
<td><strong>Deep Tiled CNNs [this work]</strong></td>
<td><strong>73.1%</strong></td>
</tr>
<tr>
<td>LCC [Yu et al., 2010]</td>
<td>72.3%</td>
</tr>
<tr>
<td>mcRBMs [Ranzato &amp; Hinton, 2010]</td>
<td>71.0%</td>
</tr>
<tr>
<td>Best of all RBMs [Krizhevsky, 2009]</td>
<td>64.8%</td>
</tr>
<tr>
<td>TICA [Hyvarinen et al., 2001]</td>
<td>56.1%</td>
</tr>
</tbody>
</table>
Scaling up: Discovering object classes

[Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Greg Corrado, Matthieu Devin, Kai Chen, Jeff Dean]
Training procedure

What features can we learn if we train a massive model on a massive amount of data. Can we learn a “grandmother cell”?

- Train on 10 million images (YouTube)
- 1000 machines (16,000 cores) for 1 week.
- 1.15 billion parameters
- Test on novel images
Face neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization

[Image of top stimuli from the test set]

[Image of optimal stimulus by numerical optimization]
Invariance properties

- **Horizontal shift**: 3D rotation angle, 90°
- **Vertical shift**: +15 pixels
- **Scale factor**: 1.6x

- **Feature response** for each transformation type is shown, with green lines representing the best feature and red lines representing the threshold.
Top Stimuli from the test set

Optimal stimulus by numerical optimization
Visualization

Top Stimuli from the test set

Optimal stimulus by numerical optimization
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/audio/etc. features also suffer from this (e.g, concatenations/combinations of different features).
Summary/Big ideas
Probabilistic vs. non-probabilistic models
Two main settings in which good results obtained. Has been confusing to outsiders.

- Lots of labeled data. “Train the heck out of the network.”

- Small amount of labeled data. (Lots of unlabeled data.) Unsupervised Feature Learning/Self-Taught learning.
Summary

• Large scale brain simulations as revisiting of the big “AI dream.”

• “Deep learning” has had two big ideas:
  – Learning multiple layers of representation
  – Learning features from unlabeled data

• Scalability is important.

• Detailed tutorial: http://deeplearning.stanford.edu/wiki