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.

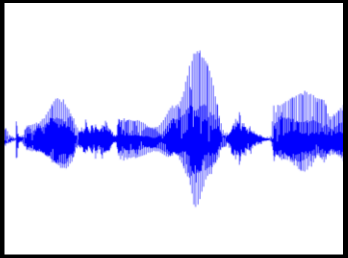
I believe this is our best shot at progress towards real AI.

What do we want computers to do with our data?

Images/video

Label: “Motorcycle”
Suggest tags
Image search
…

Audio



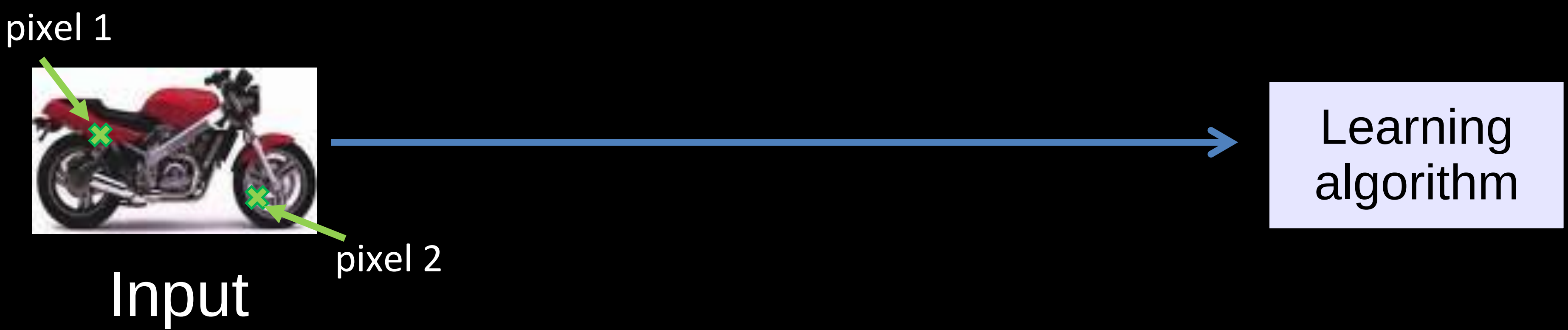
Speech recognition
Speaker identification
Music classification
…

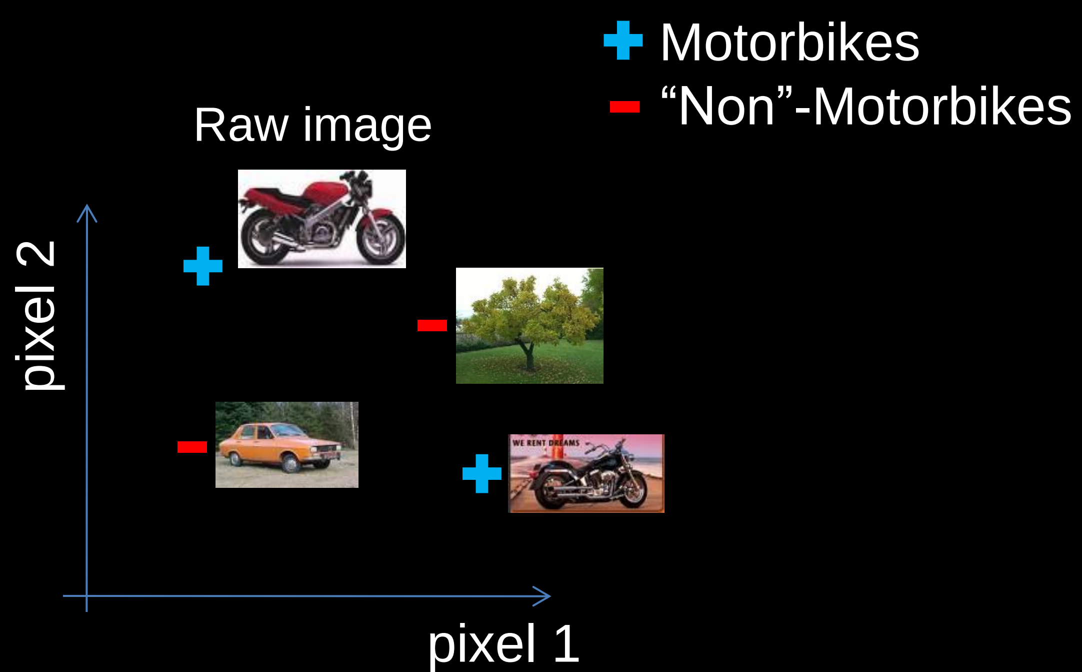
Text

Web search
Anti-spam
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 and feature representations





Andrew Ng

Machine learning and feature representations

pixel 1

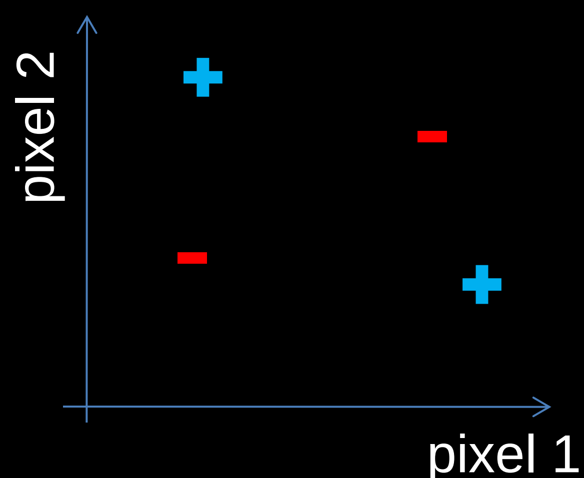
pixel 2

Input

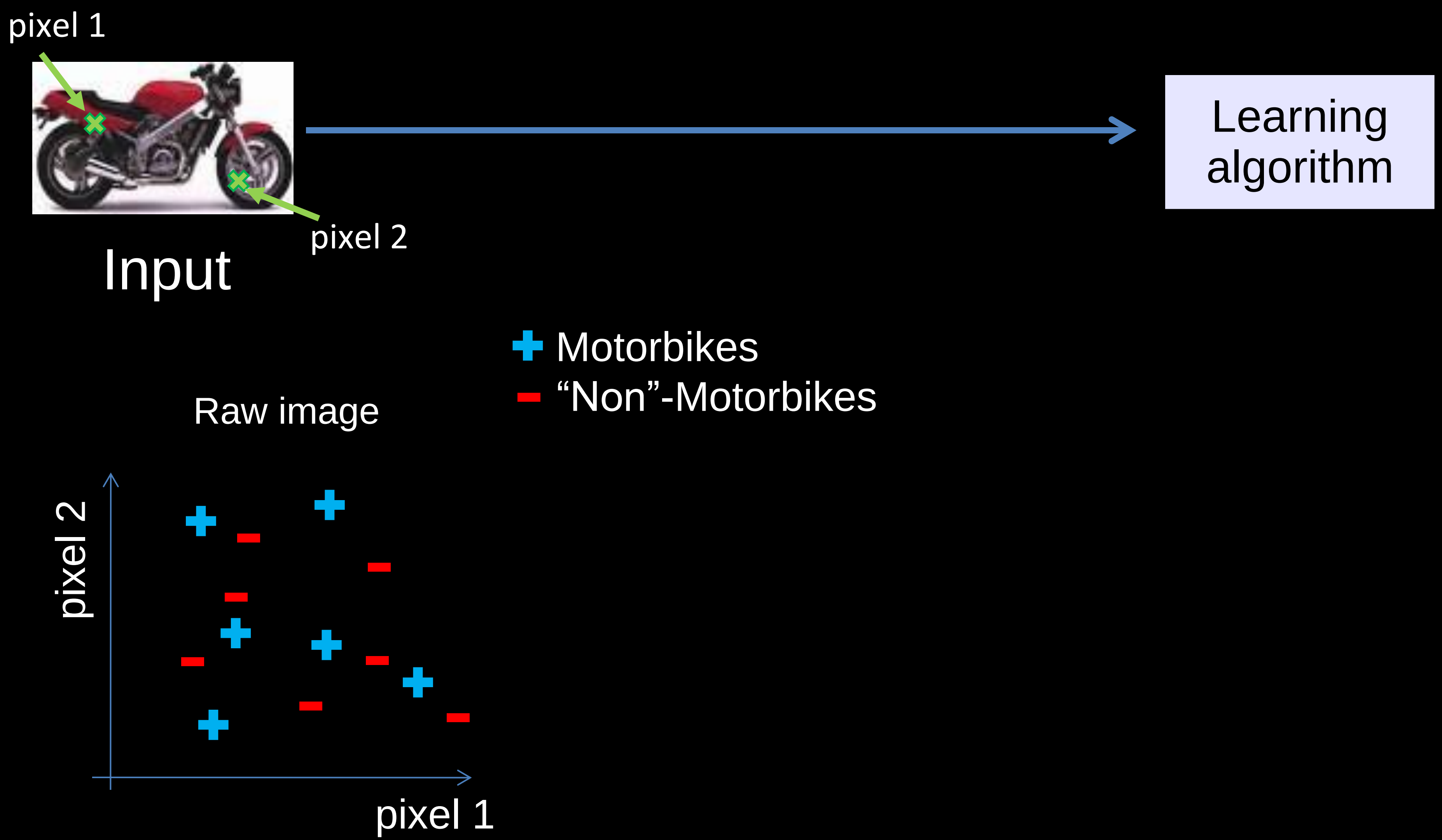
Learning algorithm

+ Motorbikes
– "Non"-Motorbikes

Raw image



Machine learning and feature representations



Andrew Ng

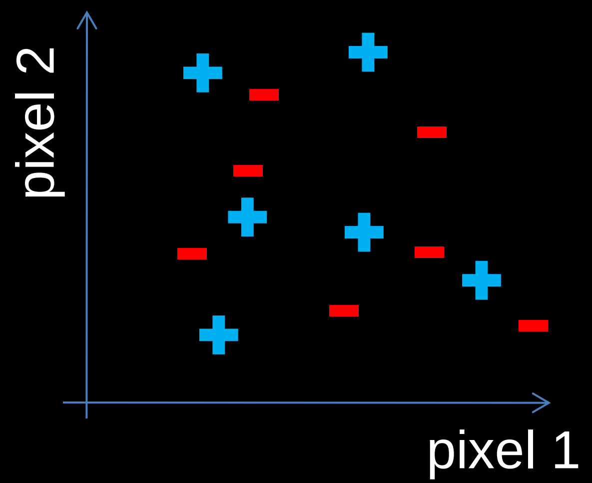
What we want

Input → Feature representation → Learning algorithm

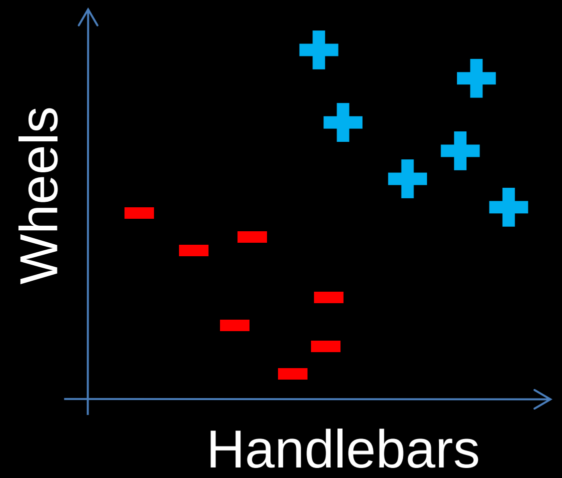
E.g., Does it have Handlebars? Wheels?

+ Motorbikes
– “Non”-Motorbikes

Raw image



Features



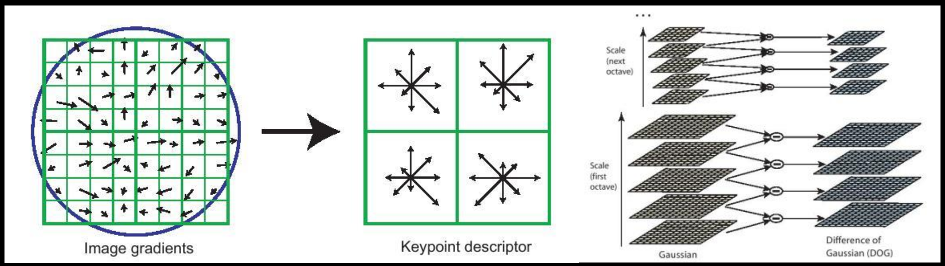
Feature representations

Input

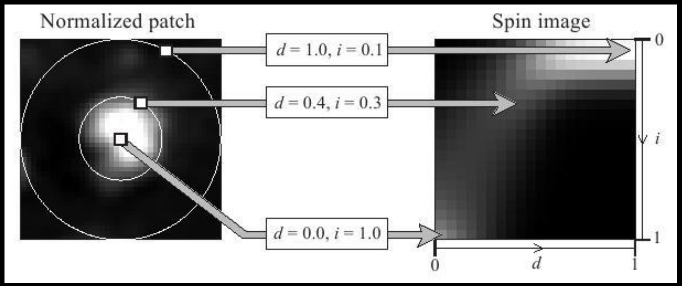
Feature Representation

Learning algorithm

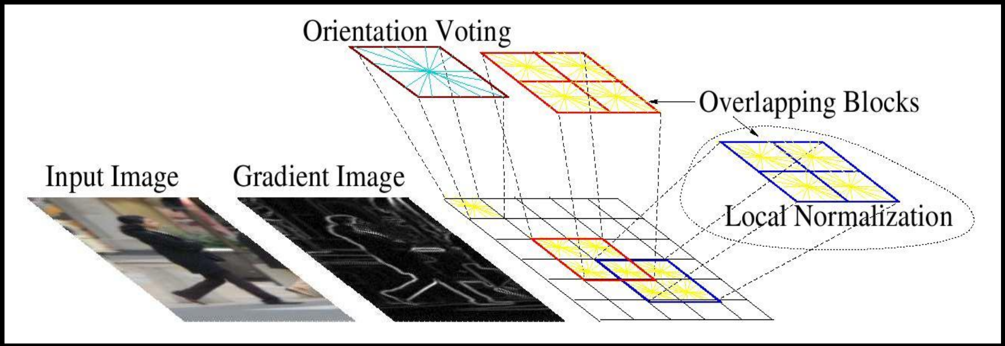
Computer vision features



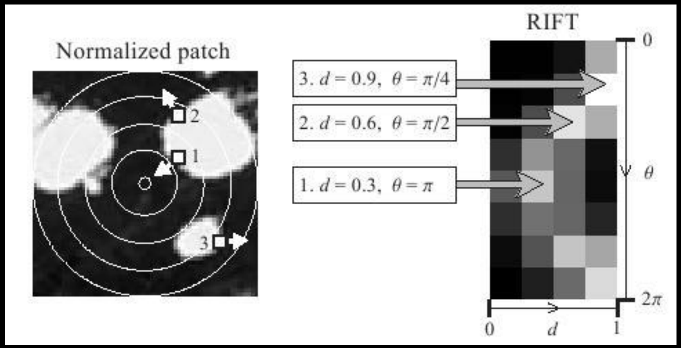
SIFT



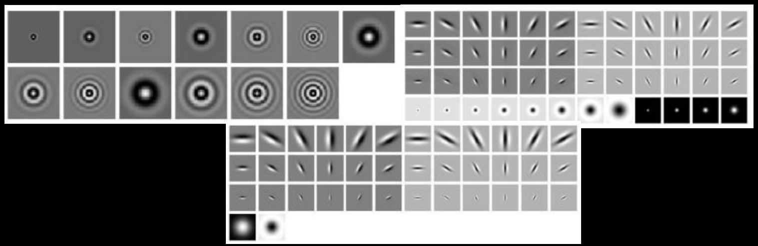
Spin image



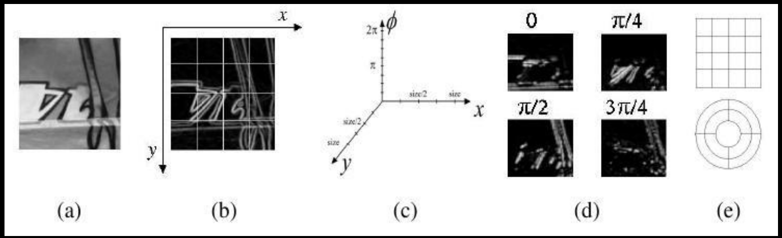
HoG



RIFT



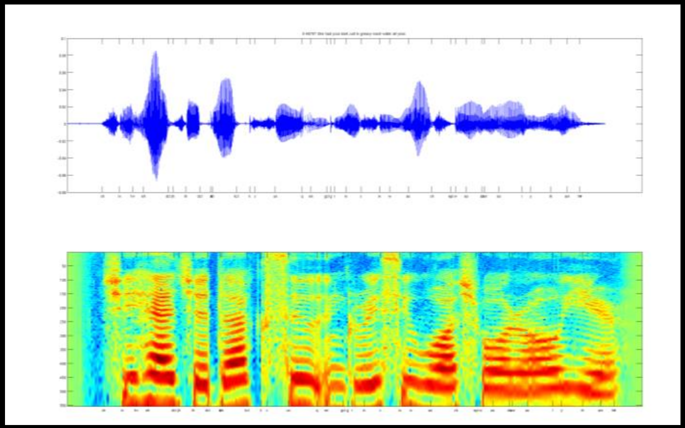
Textons



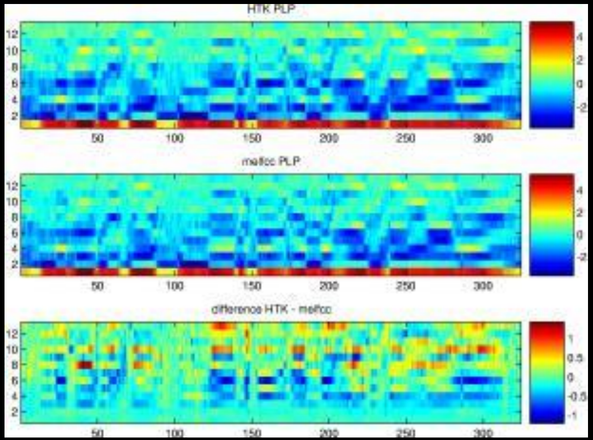
GLOH

Andrew Ng

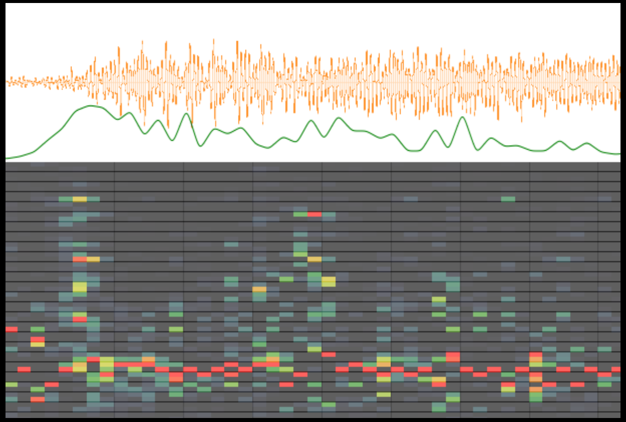
Audio features



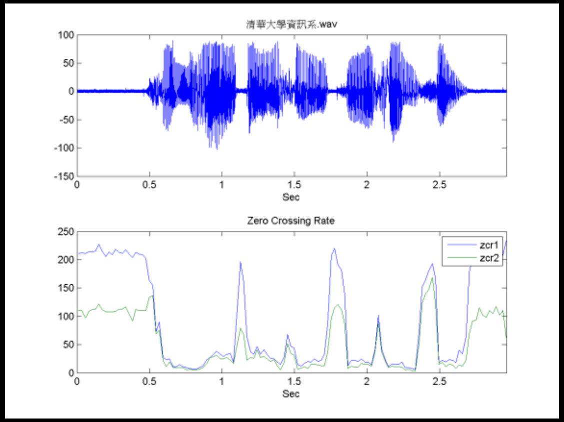
Spectrogram



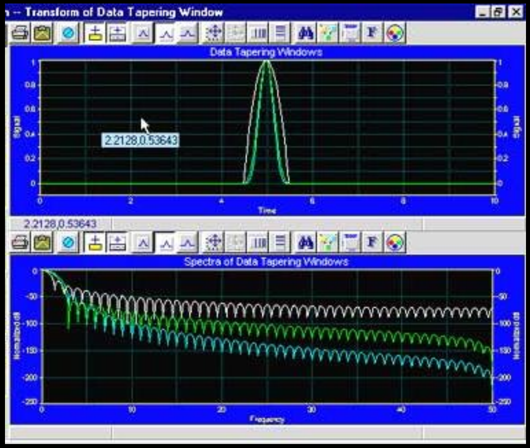
MFCC



Flux

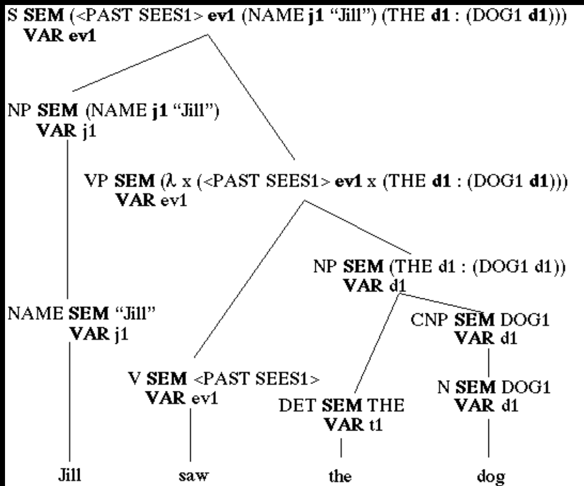


ZCR

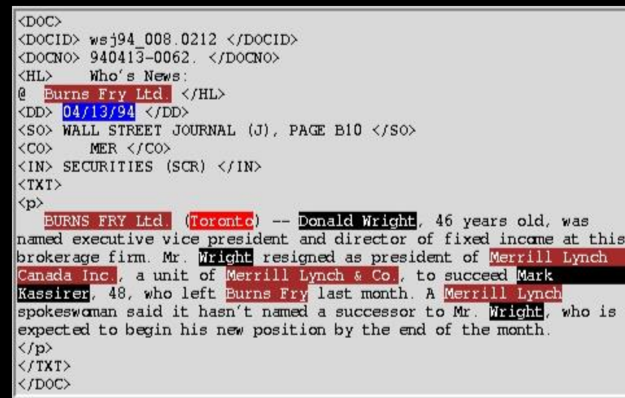


Rolloff

NLP features

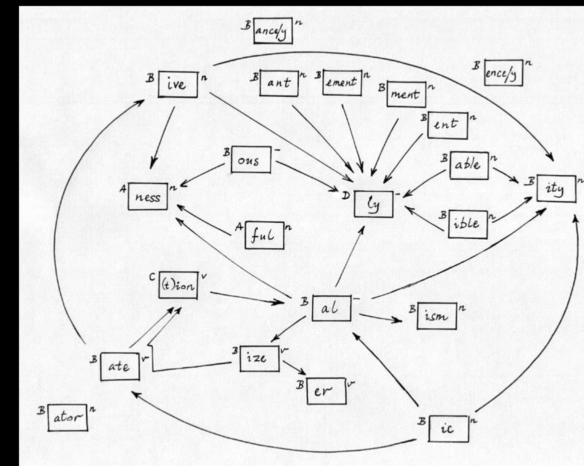


Pars



```
<DOC>
<DOCID> wsj94_008.0212 </DOCID>
<DOCNO> 940413-0062. </DOCNO>
<HL>    Who's News:
@ Burns Fry Ltd. </HL>
<DD> 04/13/94 </DD>
<SO> WALL STREET JOURNAL (J), PAGE B10 </SO>
<CO>    MER </CO>
<IN> SECURITIES (SCR) </IN>
<TXT>
<p>
   BURNS FRY Ltd. (Toronto) -- Donald Wright, 46 years old, was
named executive vice president and director of fixed income at this
brokerage firm. Mr. Wright resigned as president of Merrill Lynch
Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark
Kassirer, 48, who left Burns Fry last month. A Merrill Lynch
spokeswoman said it hasn't named a successor to Mr. Wright, who is
expected to begin his new position by the end of the month.
</p>
</TXT>
</DOC>
```

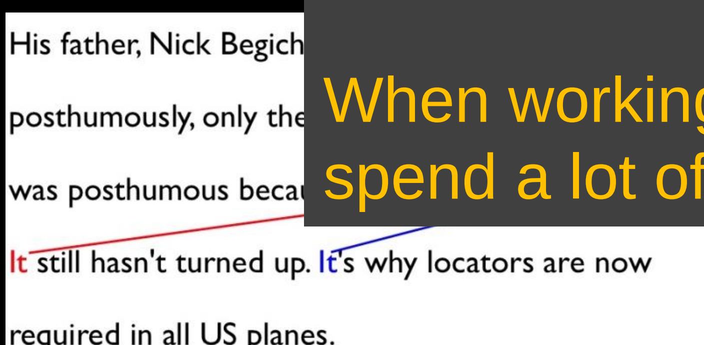
Named entity recognition



Stemming

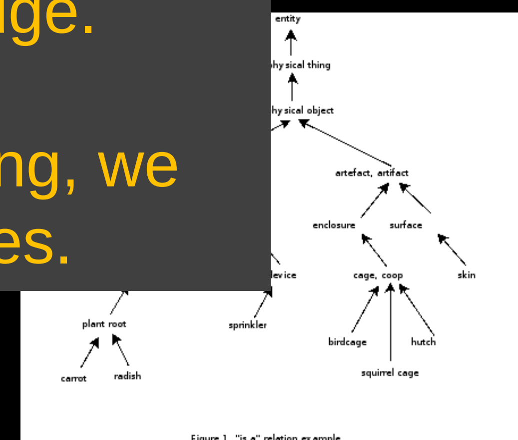
Coming up with features is difficult, time-consuming, requires expert knowledge.

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



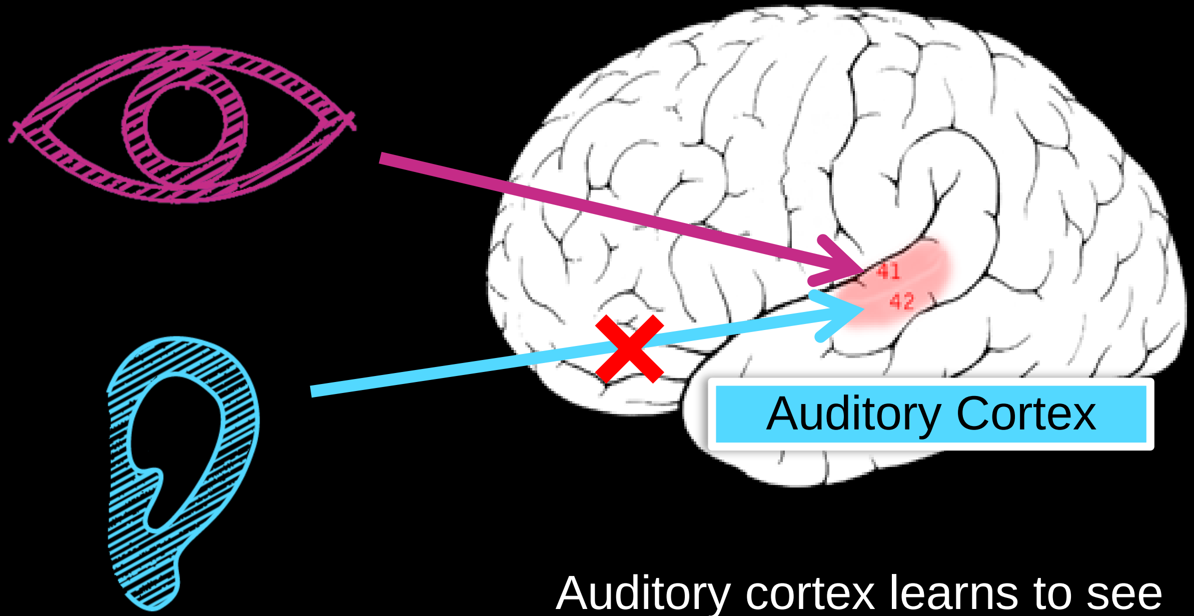
Anaphora

Part of speech



Ontologies (WordNet)

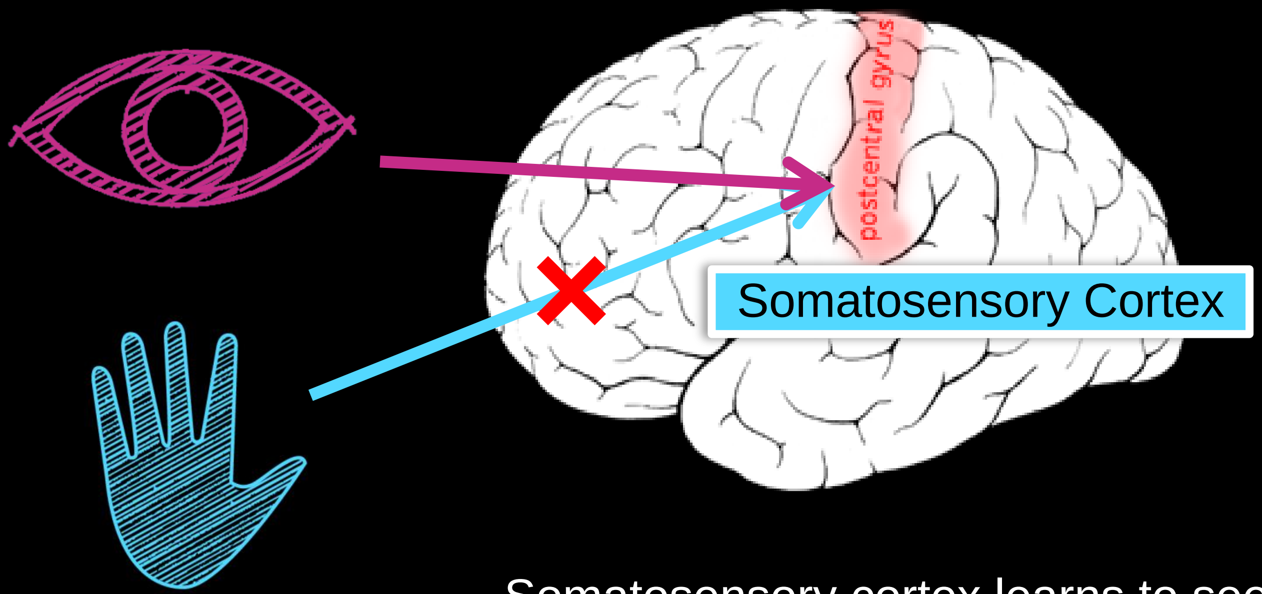
The “one learning algorithm” hypothesis



Auditory cortex learns to see

[Roe et al., 1992]

The "one learning algorithm" hypothesis



Somatosensory cortex learns to see

[Metin & Frost, 1989]

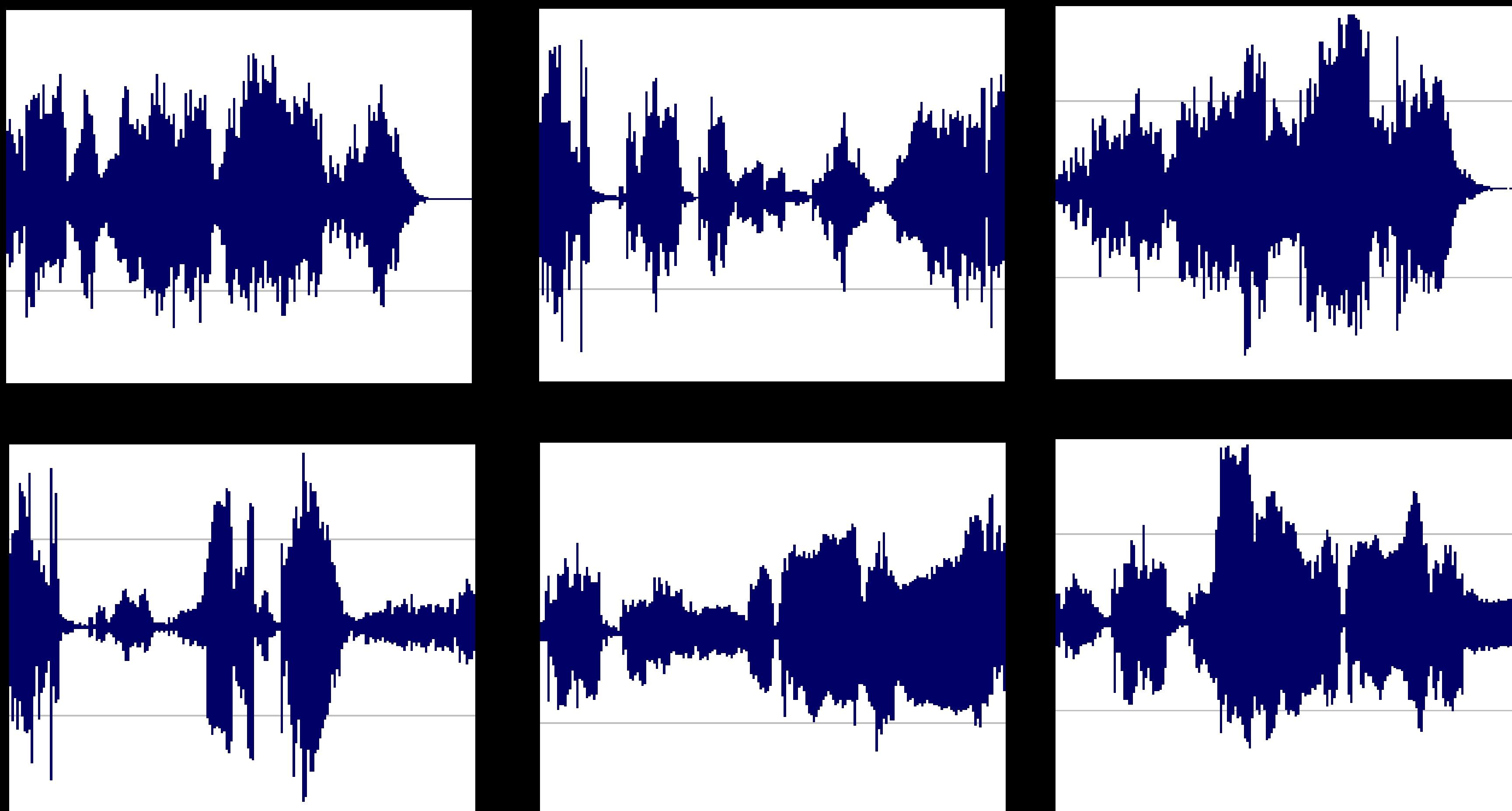
Learning input representations






Find a better way to represent images than pixels.

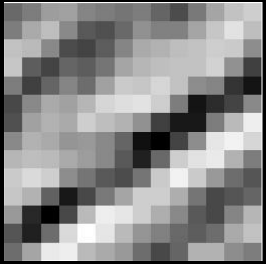
Learning input representations



Find a better way to represent audio.

Feature learning problem

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



$$\begin{bmatrix} 255 \\ 98 \\ 93 \\ 87 \\ 89 \\ 91 \\ 48 \\ \cdots \end{bmatrix}$$

- Problem: Can we find a learn 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 $R^{n \times n}$)

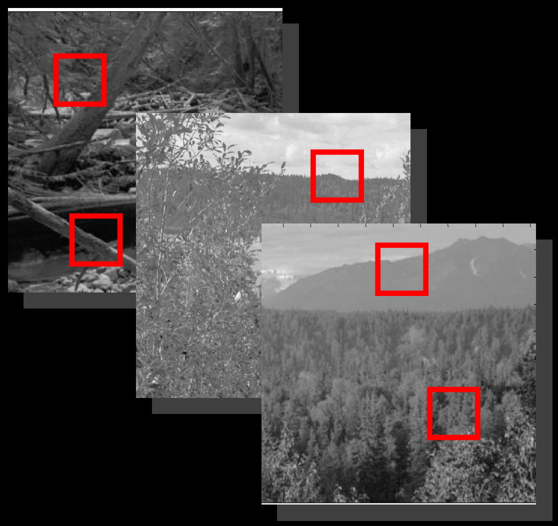
Learn: Dictionary of bases $\phi_1, \phi_2, \ldots, \phi_k$ (also $R^{n \times n}$), so that each input x can be approximately decomposed as:

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

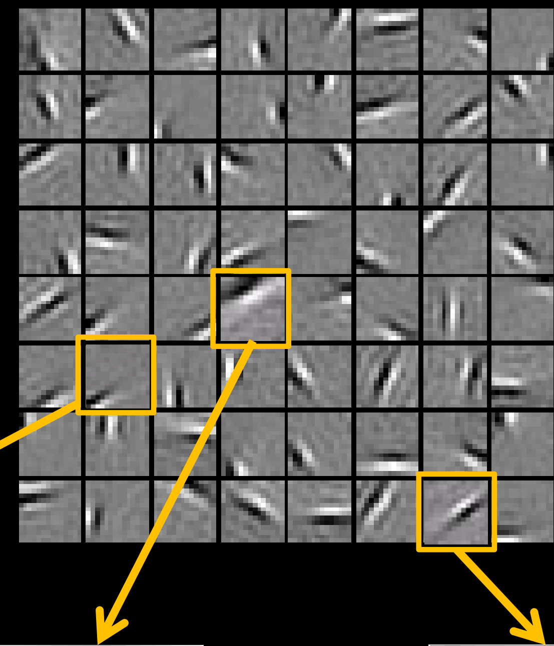
s.t. a_j's are mostly zero ("sparse")

Sparse coding illustration

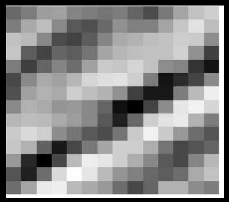
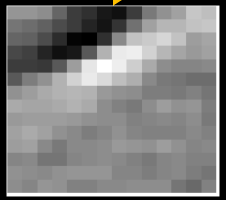
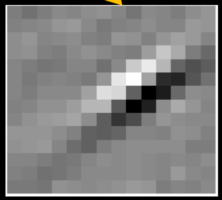
Natural Images



Learned bases ($\phi_1, \ldots, \phi_{64}$): "Edges"



Test example

 $\approx 0.8 *$  $+ 0.3 *$  $+ 0.5 *$ 

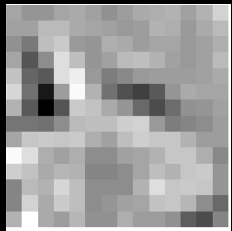
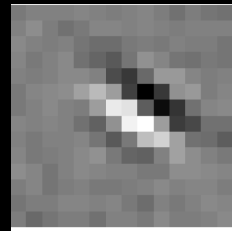
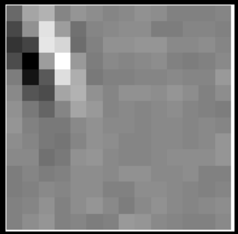
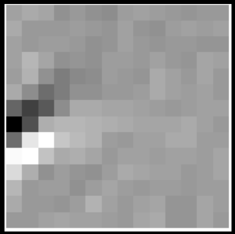
$$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$$

$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$

(feature representation)

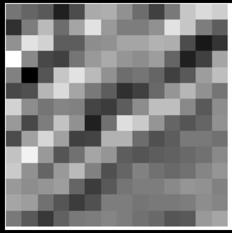
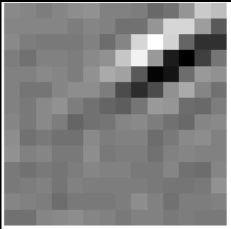
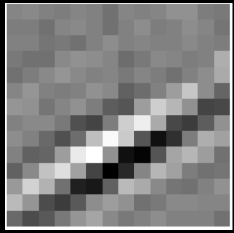
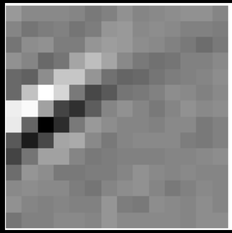
More succinct, higher-level, representation.

More examples

 $\approx 0.6 *$  $+ 0.8 *$  $+ 0.4 *$ 

ϕ_{15} ϕ_{28} ϕ_{37}

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

 $\approx 1.3 *$  $+ 0.9 *$  $+ 0.3 *$ 

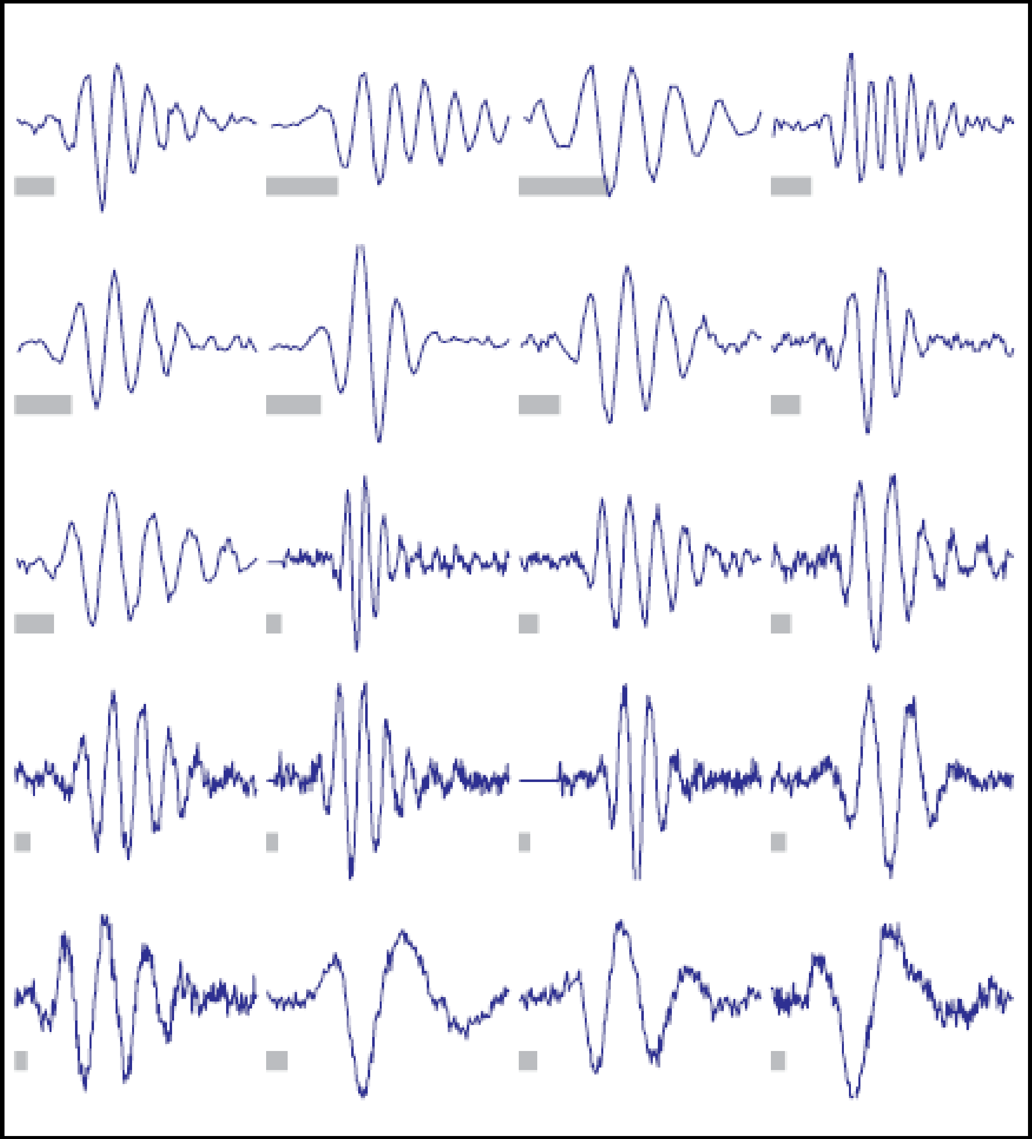
ϕ_{5} ϕ_{18} ϕ_{29}

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

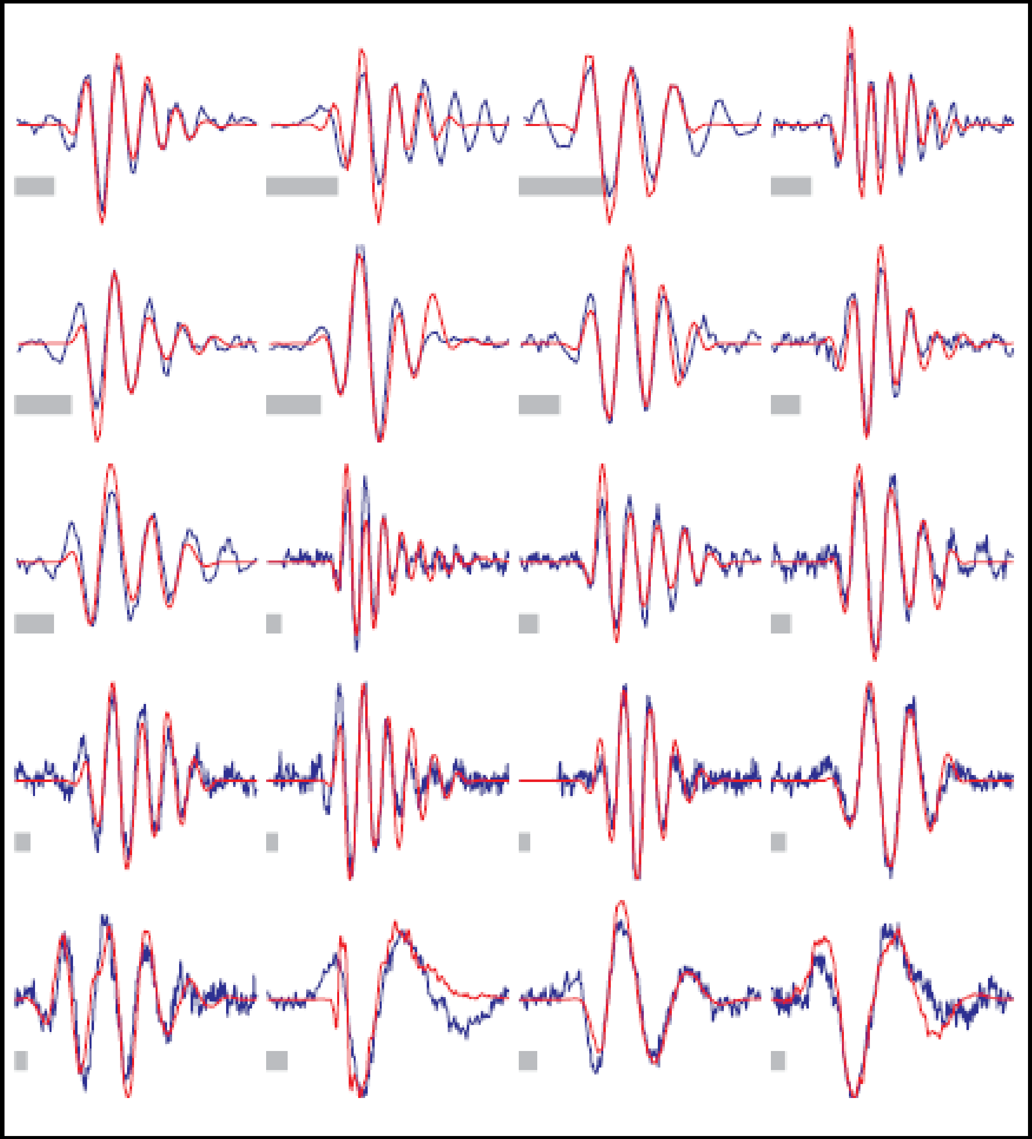
Image shows 20 basis functions learned from unlabeled audio.



[Evan Smith & Mike Lewicki, 2006]

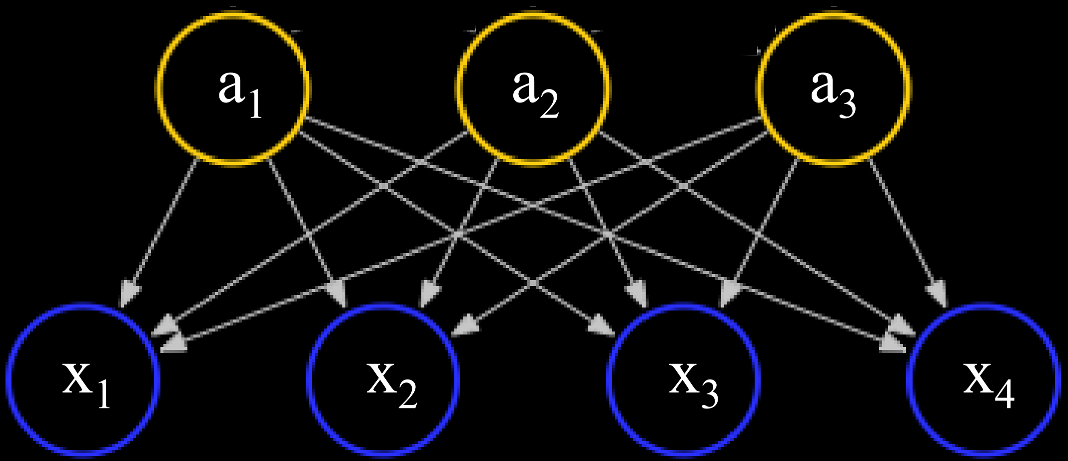
Sparse coding applied to audio

Image shows 20 basis functions learned from unlabeled audio.



[Evan Smith & Mike Lewicki, 2006]

Learning feature hierarchies



Higher layer
(Combinations of edges; cf V2)

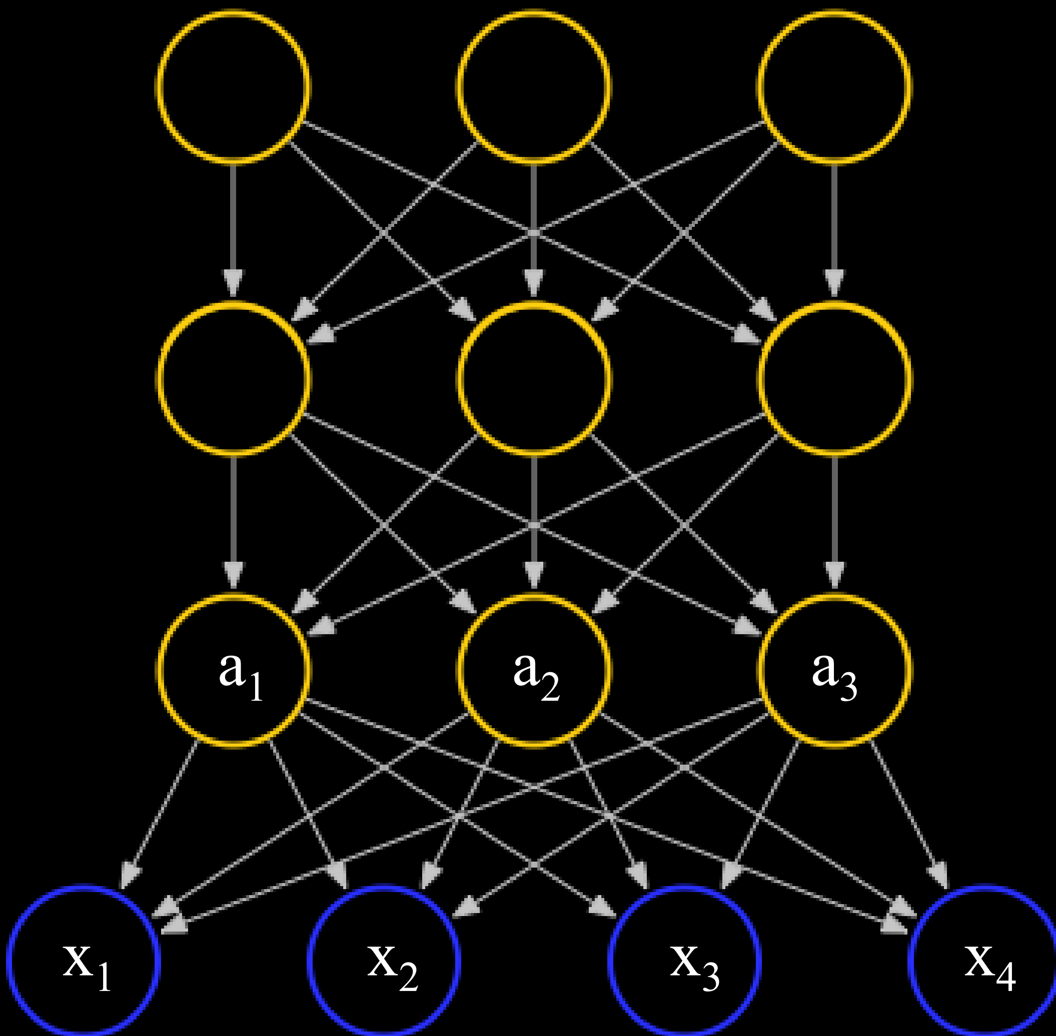
"Sparse coding"
(edges; cf. V1)

Input image (pixels)

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

[Lee, Ranganath & Ng, 2007]

Learning feature hierarchies



Higher layer
(Model V3?)

Higher layer
(Model V2?)

Model V1

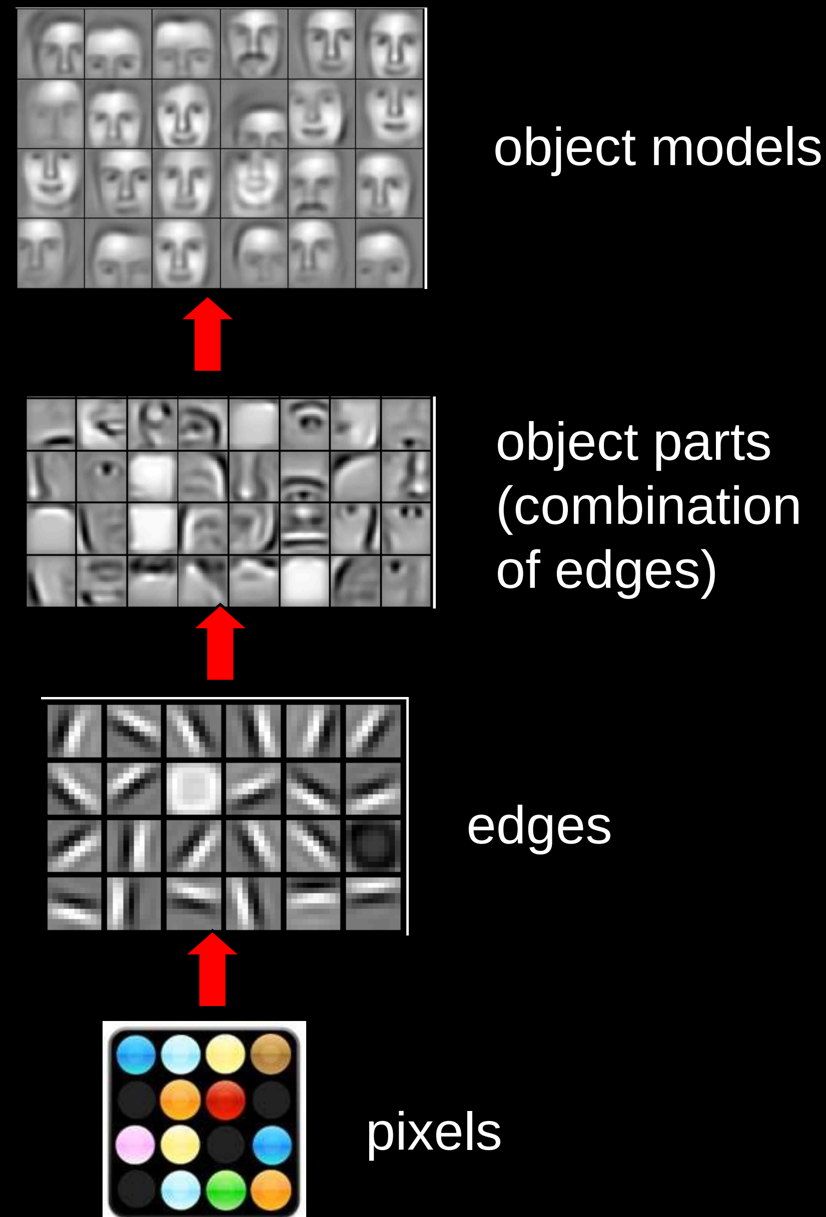
Input image

[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.



[Honglak Lee]

State-of-the-art Unsupervised feature learning

Images

CIFAR Object classification	Accuracy
Prior art (Ciresan et al., 2011)	80.5%
Stanford Feature learning	**82.0%**

NORB Object classification	Accuracy
Prior art (Scherer et al., 2010)	94.4%
Stanford Feature learning	**95.0%**

Video

Hollywood2 Classification	Accuracy
Prior art (Laptev et al., 2004)	48%
Stanford Feature learning	**53%**

YouTube	Accuracy
Prior art (Liu et al., 2009)	71.2%
Stanford Feature learning	**75.8%**

KTH	Accuracy
Prior art (Wang et al., 2010)	92.1%
Stanford Feature learning	**93.9%**

UCF	Accuracy
Prior art (Wang et al., 2010)	85.6%
Stanford Feature learning	**86.5%**

Text/NLP

Paraphrase detection	Accuracy
Prior art (Das & Smith, 2009)	76.1%
Stanford Feature learning	**76.4%**

Sentiment (MR/MPQA data)	Accuracy
Prior art (Nakagawa et al., 2010)	77.3%
Stanford Feature learning	**77.7%**

Multimodal (audio/video)

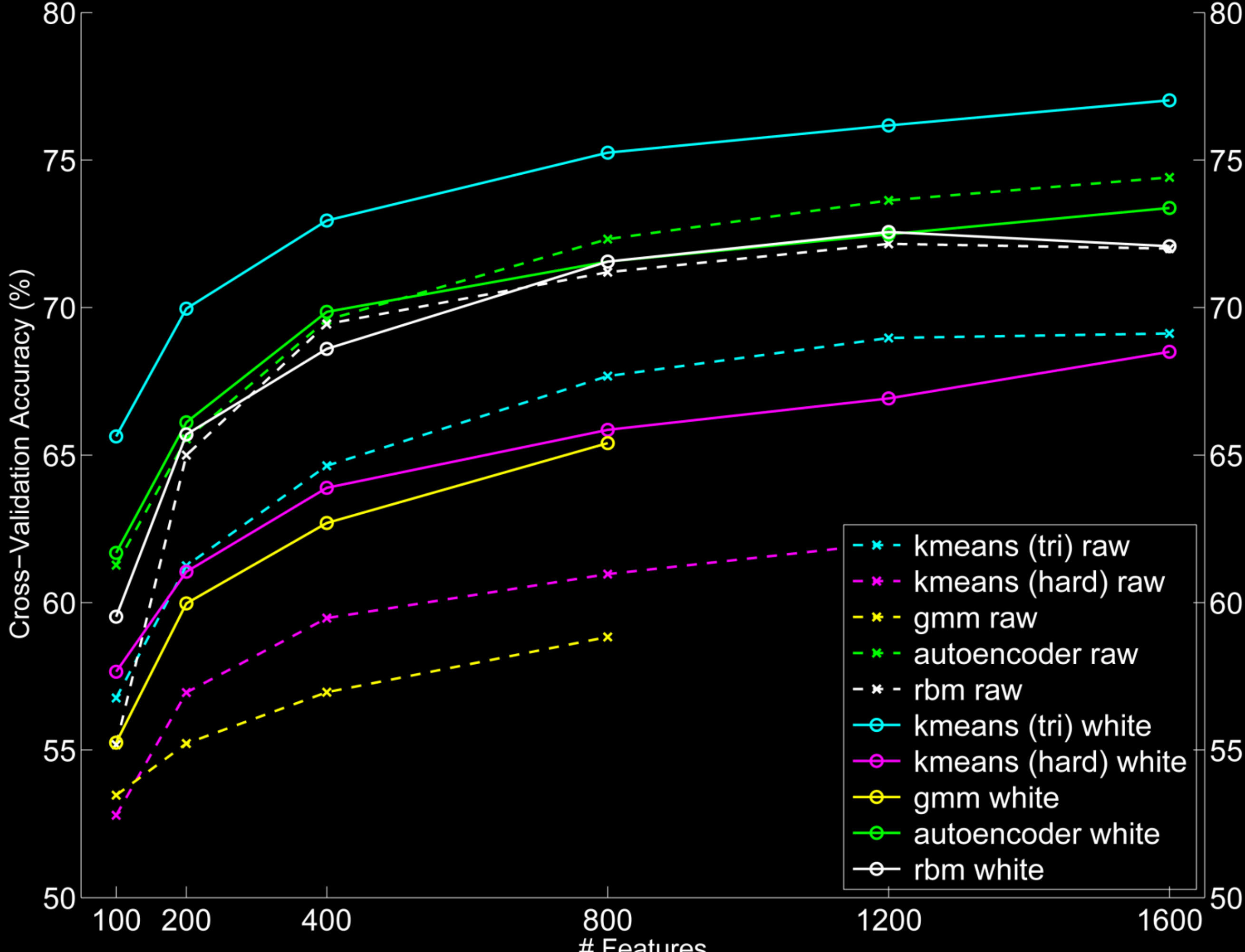
AVLetters Lip reading	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Stanford Feature learning	**65.8%**

Other unsupervised feature learning records:
Pedestrian detection (Yann LeCun)
Speech recognition (Geoff Hinton)
PASCAL VOC object classification (Kai Yu)

Technical challenge: Scaling up

Scaling and classification accuracy (CIFAR-10)

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.

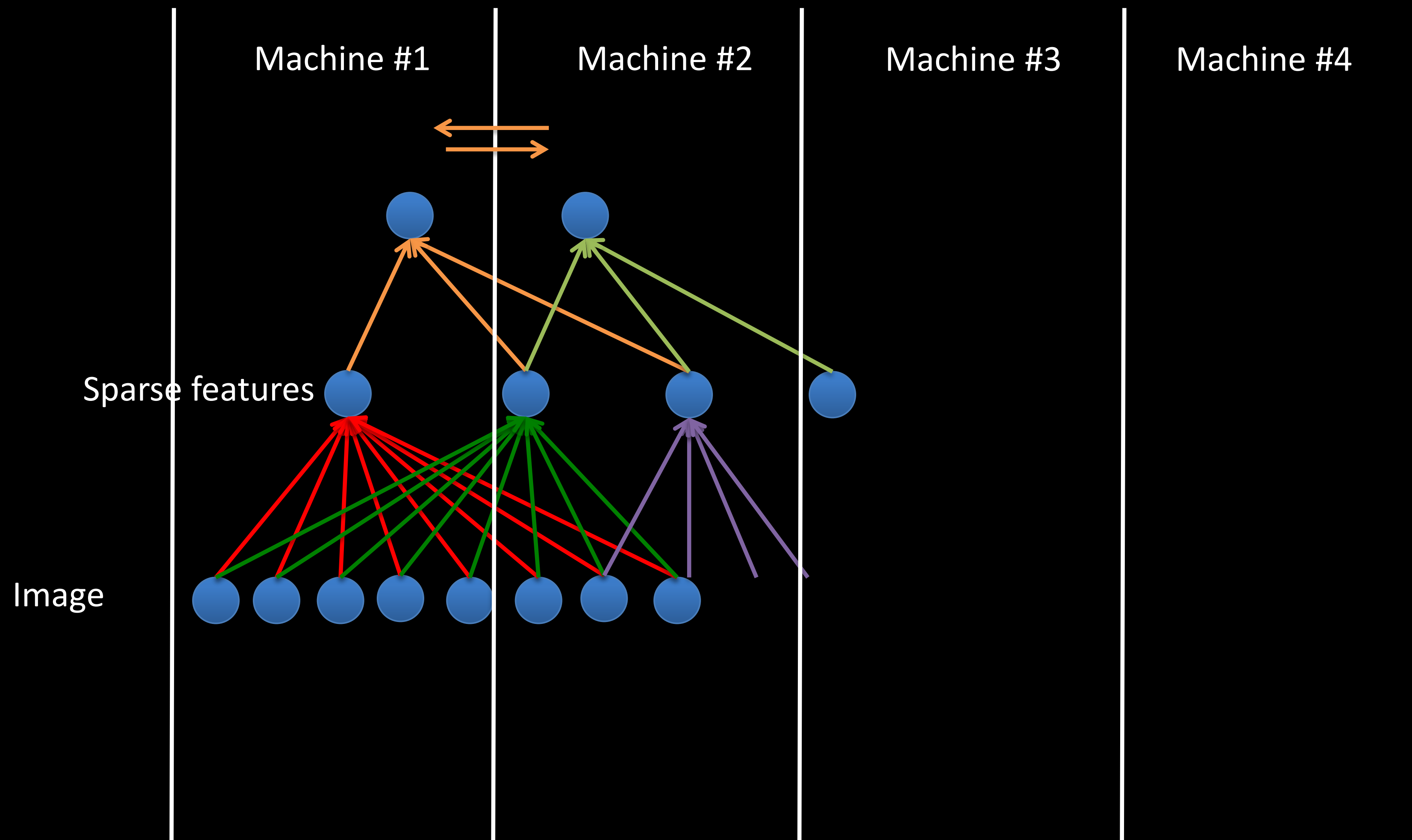


[Adam Coates]

Scaling up: Discovering object classes

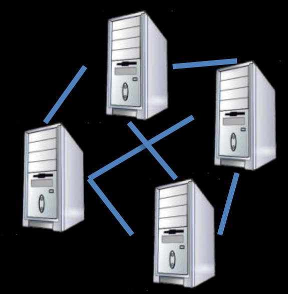
[Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Greg Corrado, Matthieu Devin, Kai Chen, Jeff Dean]

Local Receptive Field networks



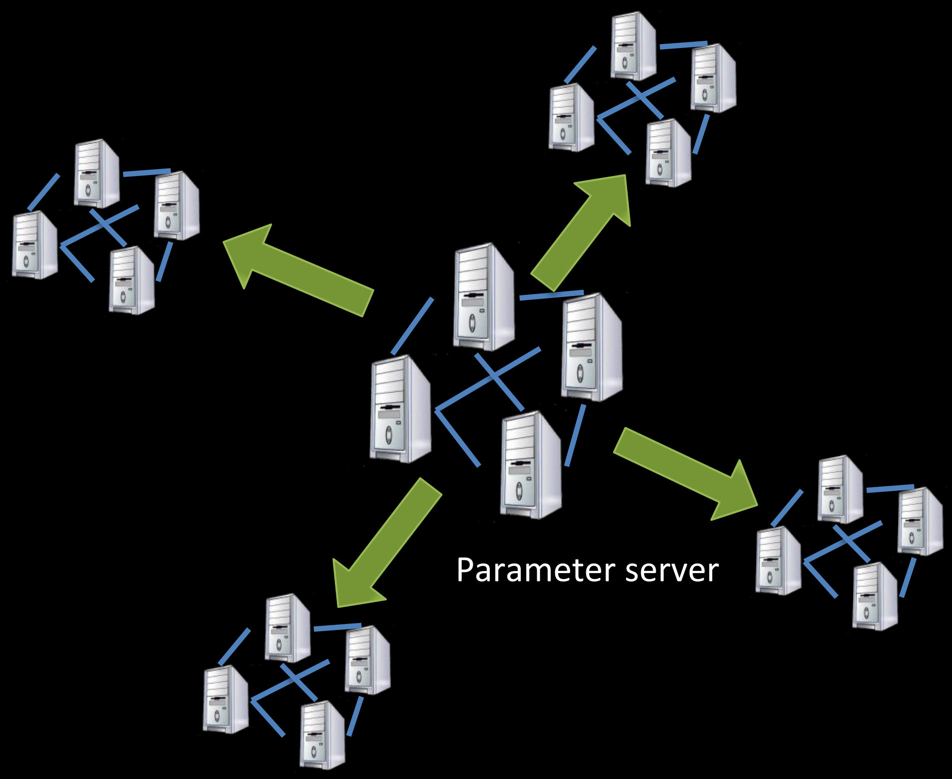
Le, et al., *Tiled Convolutional Neural Networks*. NIPS 2010

Asynchronous Parallel SGD



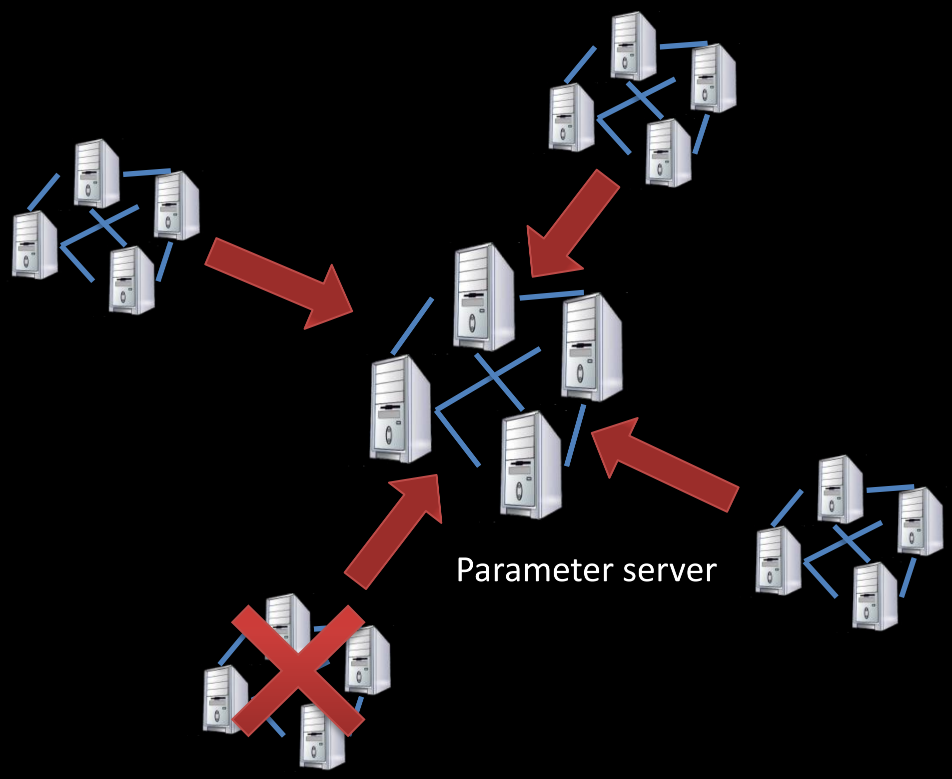
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

Asynchronous Parallel SGD



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

Asynchronous Parallel SGD

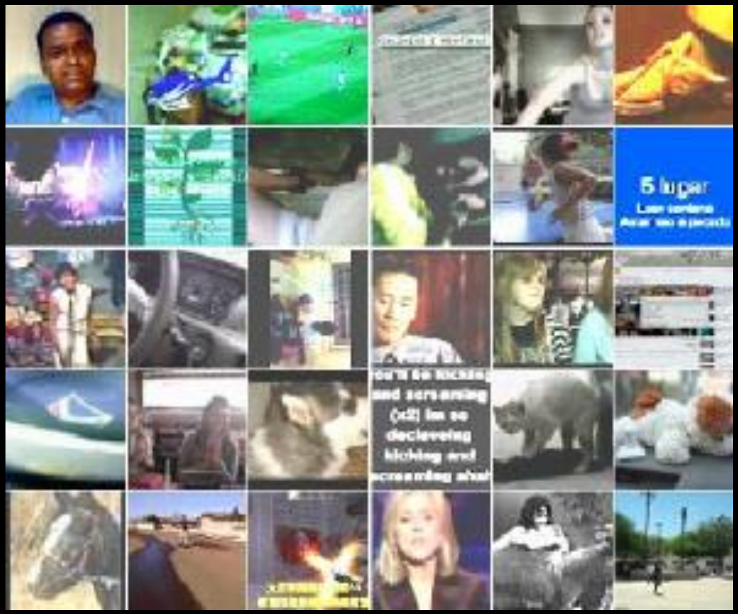


Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

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)

Andrew Ng

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 obseus
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
…

Stingray

Mantaray

0.005%

Random guess

9.5%

State-of-the-art (Weston, Bengio '11)

?

Feature learning From raw pixels

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

0.005%	9.5%	19.2%
Random guess	State-of-the-art (Weston, Bengio ‘11)	Feature learning From raw pixels

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 on Android

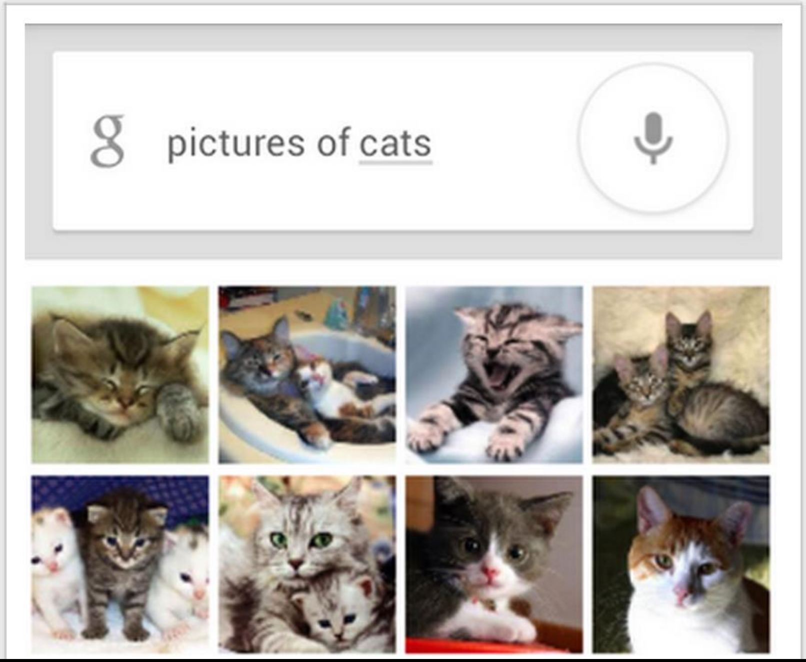
AUG
6

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

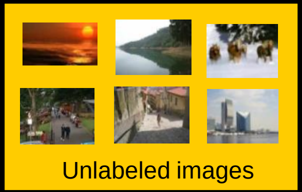
- Deep Learning : Lets learn rather than manually design our features.
- Discover the fundamental computational principles that underlie perception.
- Deep learning very successful on vision and audio tasks.
- Other variants for learning recursive representations for text.

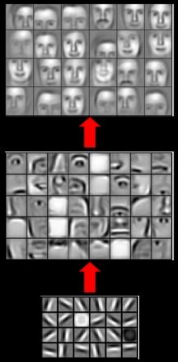
Thanks to: Adam Coates, Quoc Le, Brody Huval, Andrew Saxe, Andrew Maas, Richard Socher, Tao Wang

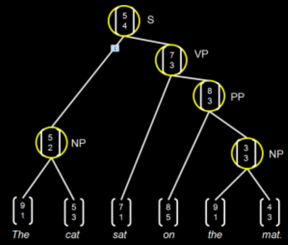
Conclusion

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.







Stanford

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

Google: Kai Chen | Greg Corrado | Jeff Dean | Matthieu Devin | Andrea Frome | Rajat Monga | Marc'Aurelio Ranzato | Paul Tucker | Kay Le

Advanced Topics

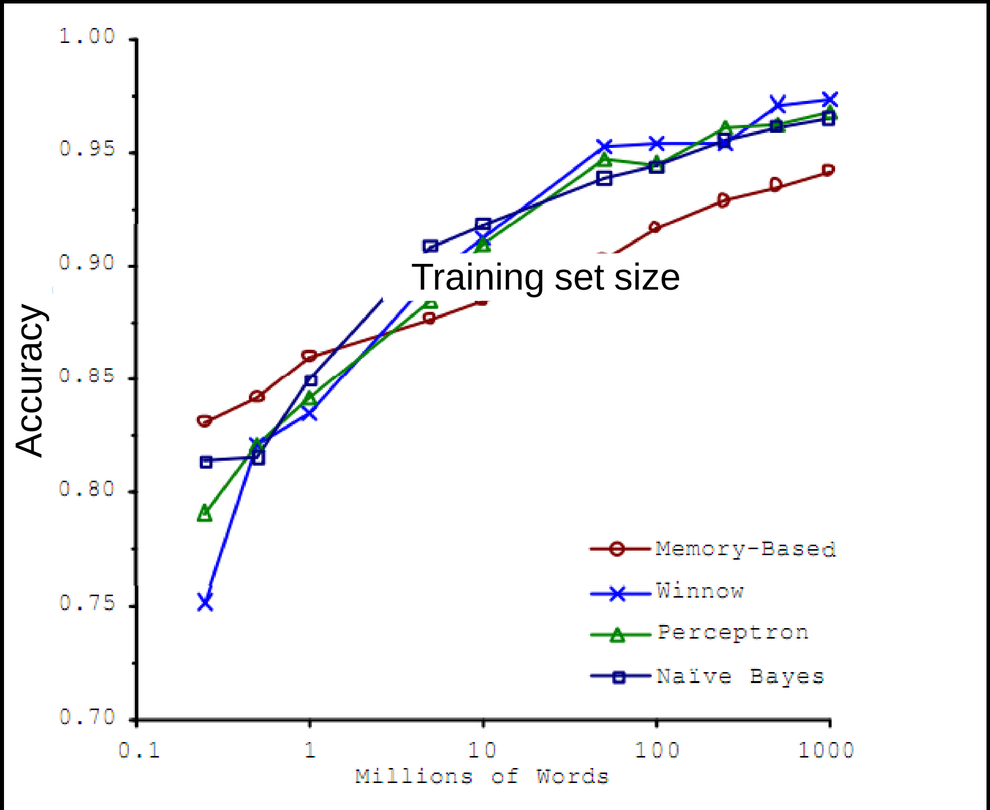
Andrew Ng
Stanford University & Google

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?



[Banko & Brill, 2001]

"It's not who has the best algorithm that wins.
It's who has the most data."

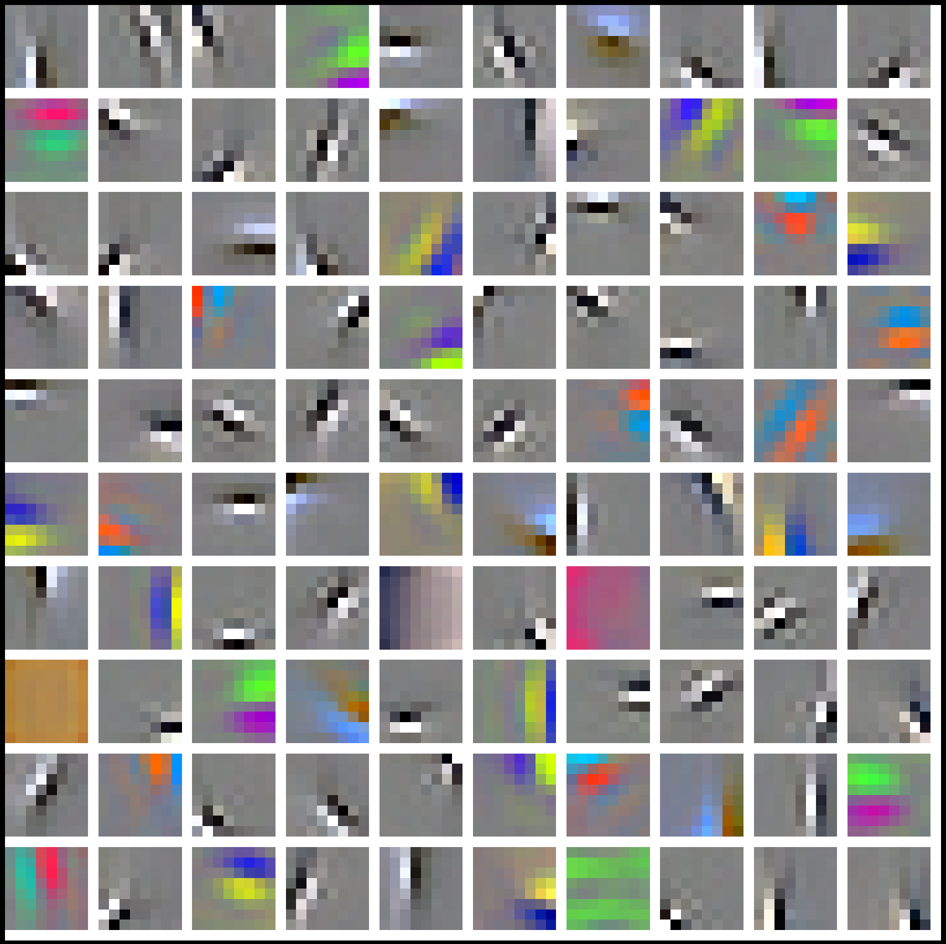
Unsupervised Feature Learning

- 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.

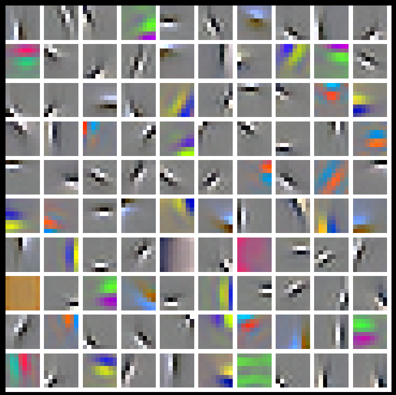


Sparse auto-encoder

Unsupervised feature learning

Weights learned with and without whitening.

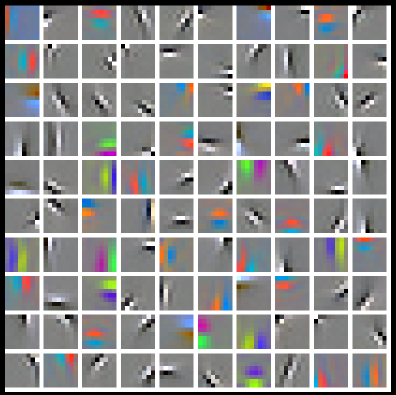
with whitening



without whitening

Sparse auto-encoder

with whitening



without whitening

Sparse RBM

with whitening

without whitening

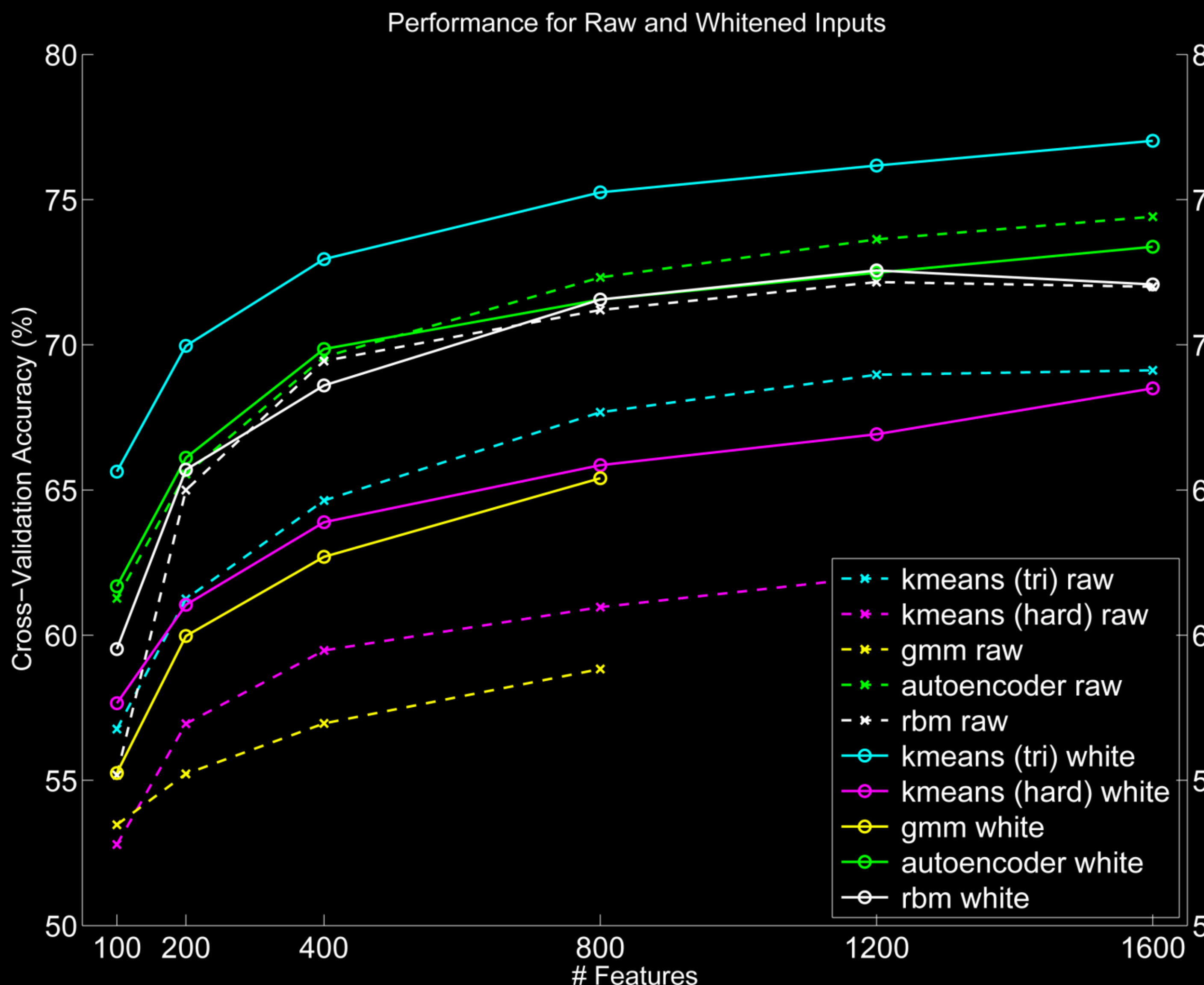
K-means

with whitening

without whitening

Gaussian mixture model

Scaling and classification accuracy (CIFAR-10)



Andrew Ng

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.

CIFAR-10 Test accuracy	
Raw pixels	37.3%
RBM with back-propagation	64.8%
3-Way Factored RBM (3 layers)	65.3%
Mean-covariance RBM (3 layers)	71.0%
Improved Local Coordinate Coding	74.5%
Convolutional RBM	78.9%
Sparse auto-encoder	73.4%
Sparse RBM	72.4%
K-means (Hard)	68.6%
K-means (Triangle, 1600 features)	77.9%
K-means (Triangle, 4000 features)	$\boxed{79.6\%}$

NORB Test accuracy (error)	
Convolutional Neural Networks	93.4% (6.6%)
Deep Boltzmann Machines	92.8% (7.2%)
Deep Belief Networks	95.0% (5.0%)
Jarrett et al., 2009	94.4% (5.6%)
Sparse auto-encoder	96.9% (3.1%)
Sparse RBM	96.2% (3.8%)
K-means (Hard)	96.9% (3.1%)
K-means (Triangle)	$\boxed{97.0\% \ (3.0\%)}$

Tiled Convolution Neural Networks

Quoc Le

Jiquan Ngiam

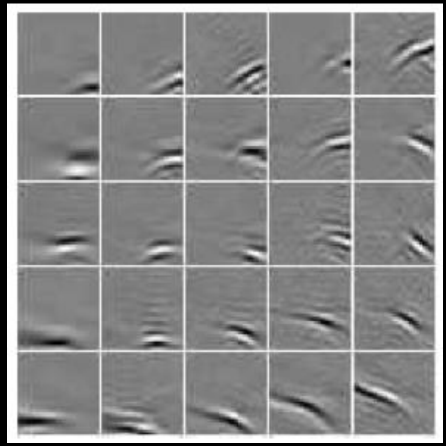
Learning Invariances

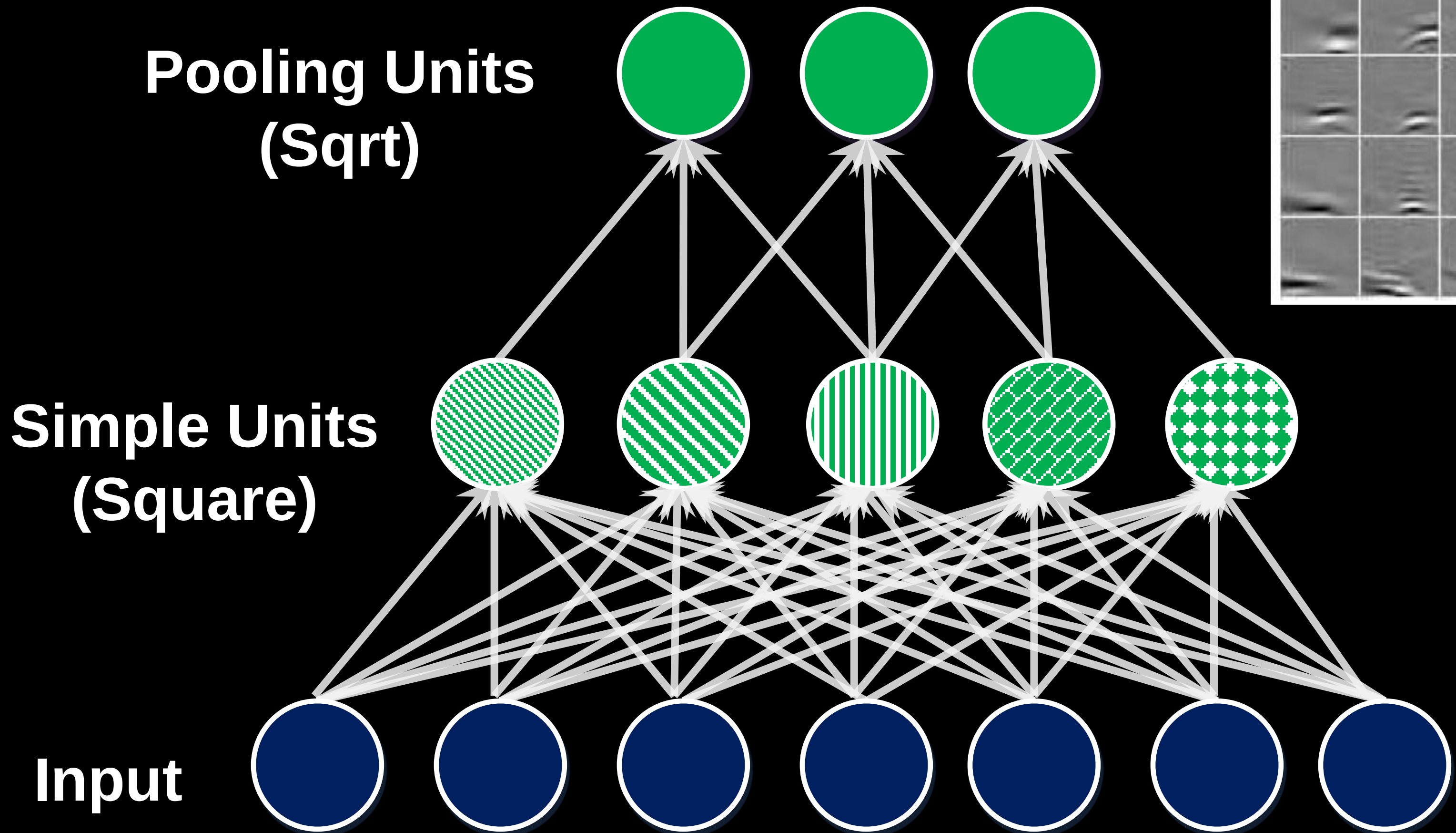
- We want to learn invariant features.

- 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.

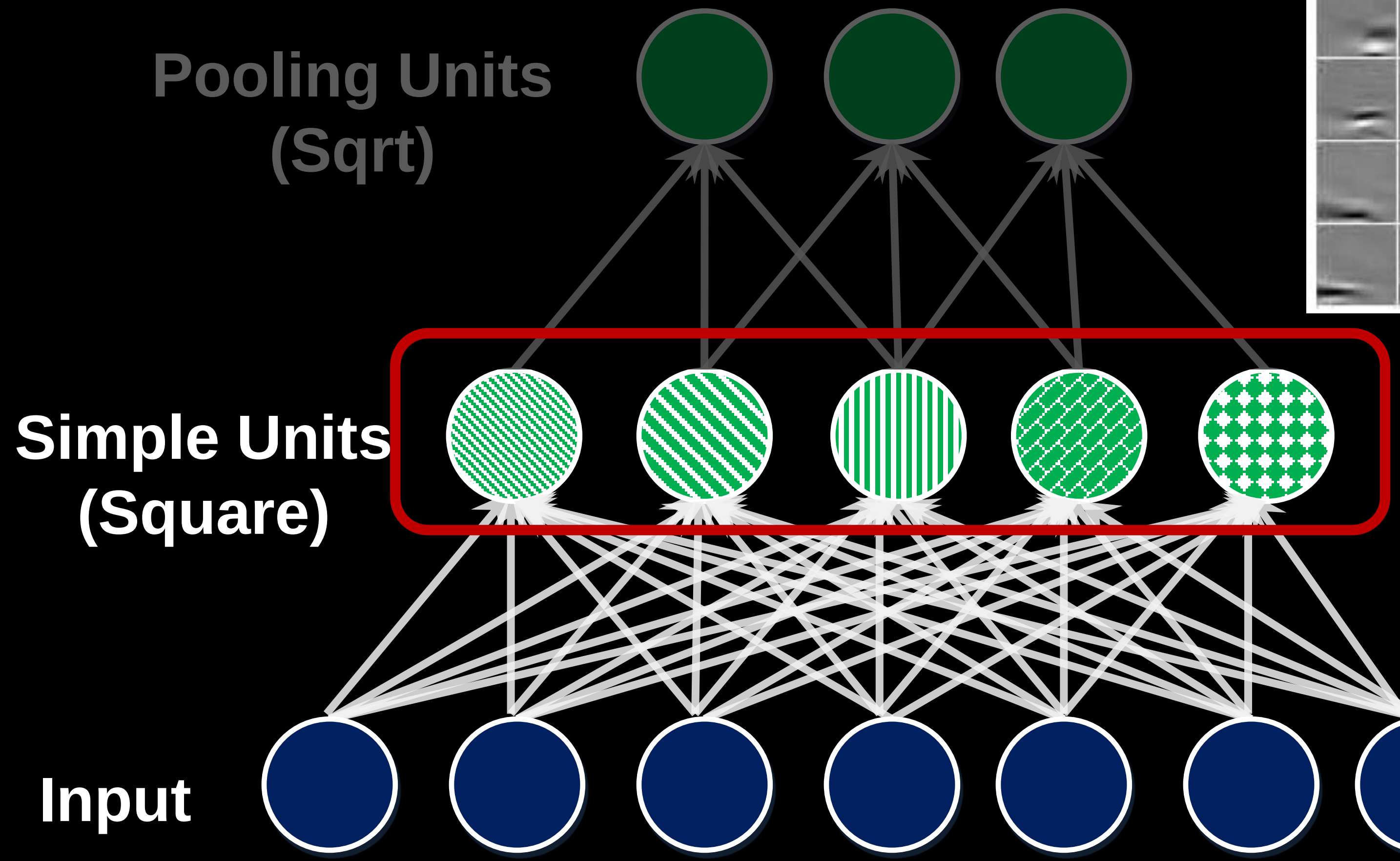
Fully Connected Topographic ICA

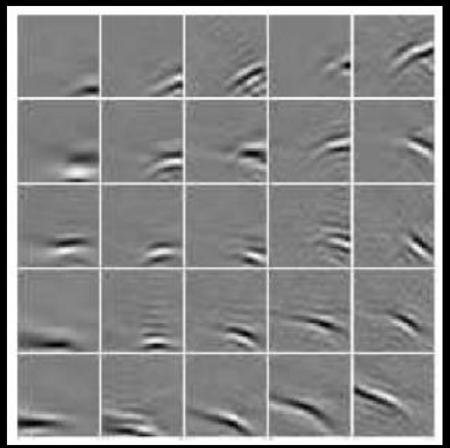




Doesn't scale to large images.

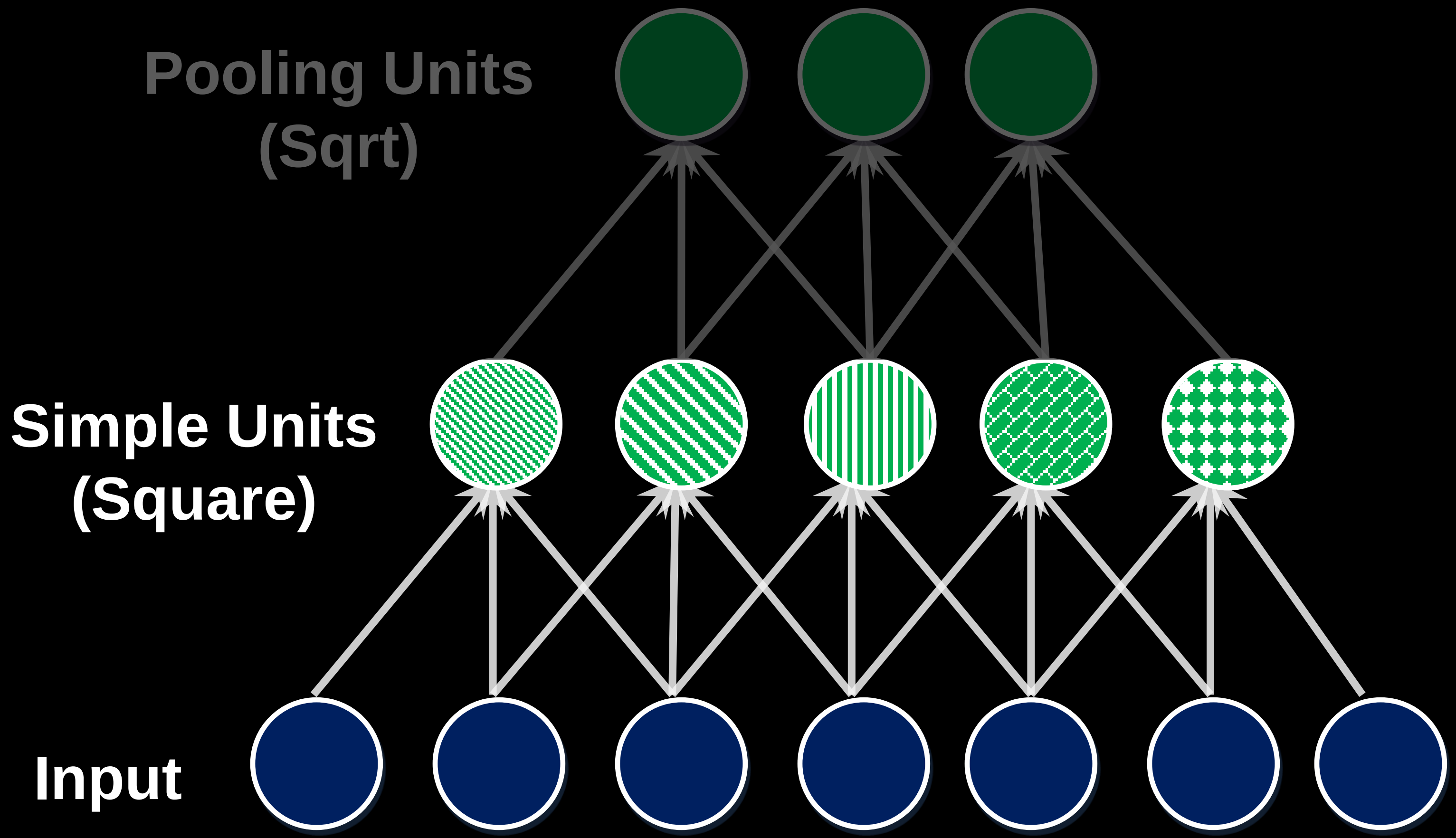
Fully Connected Topographic ICA



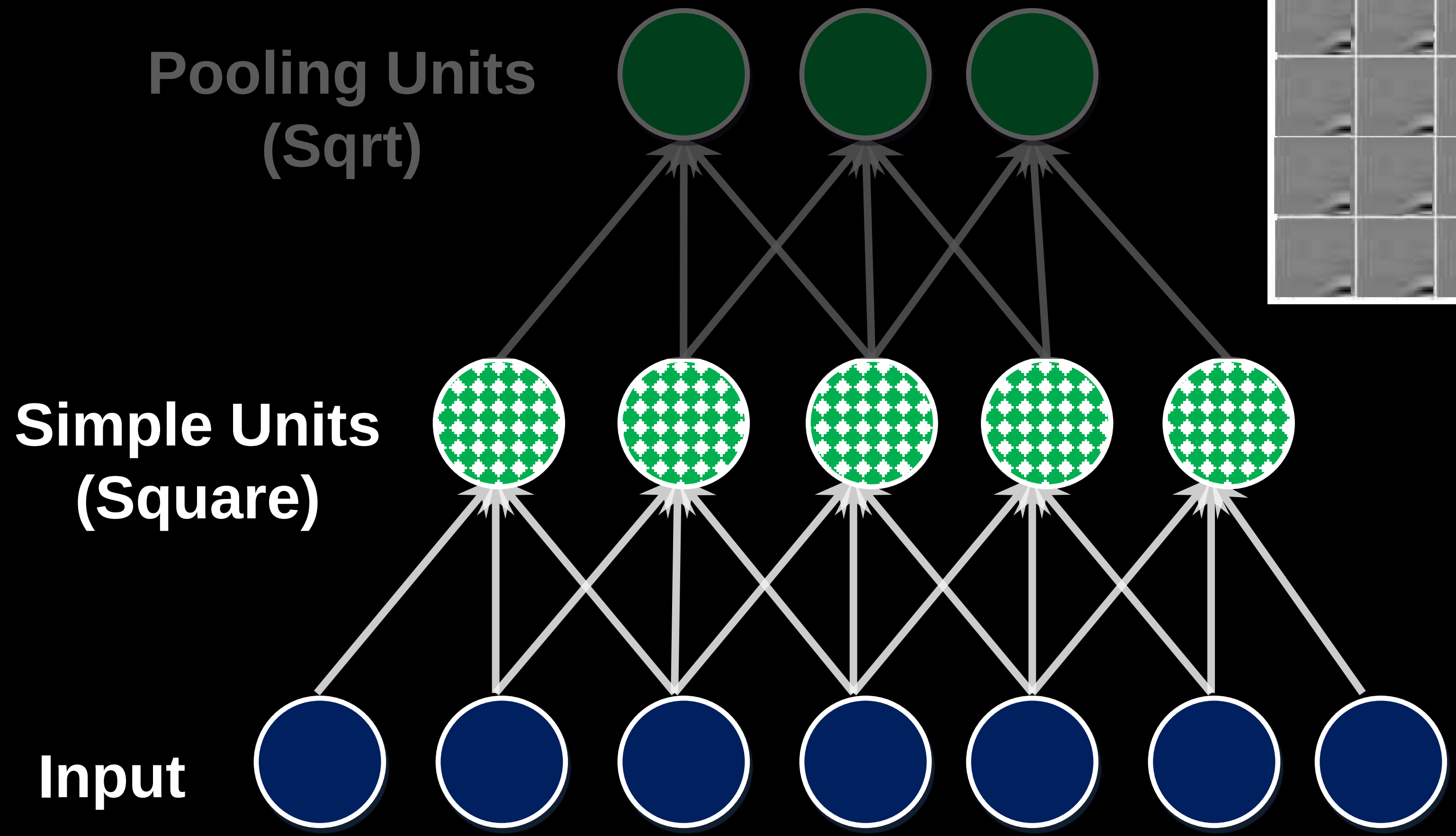


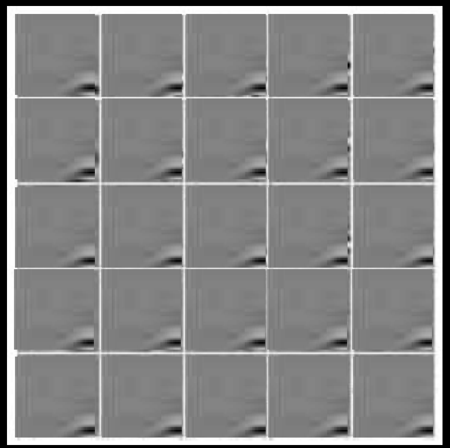
Doesn’t scale to large images.

Local Receptive Fields

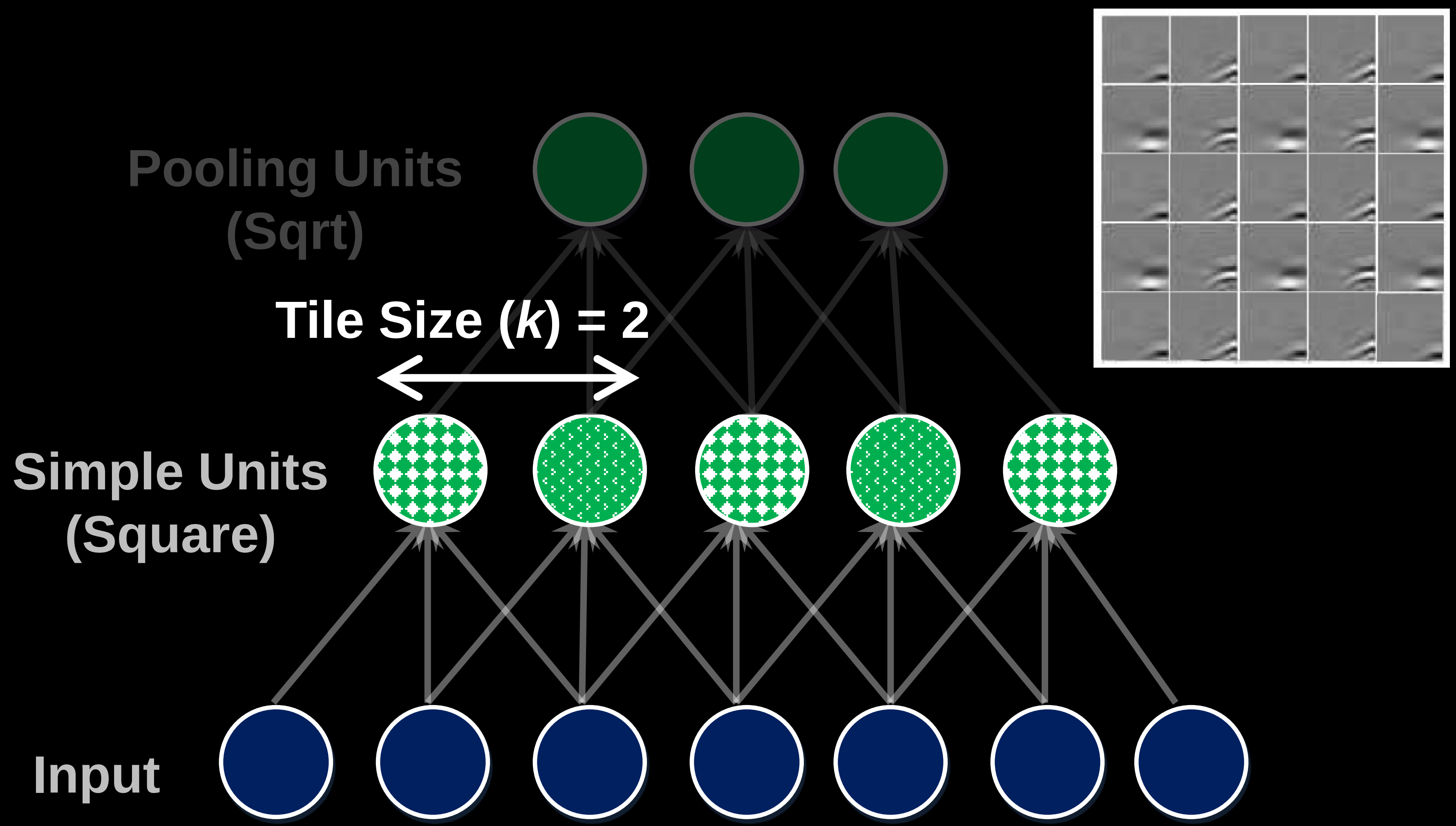


Convolution Neural Networks (Weight Tying)



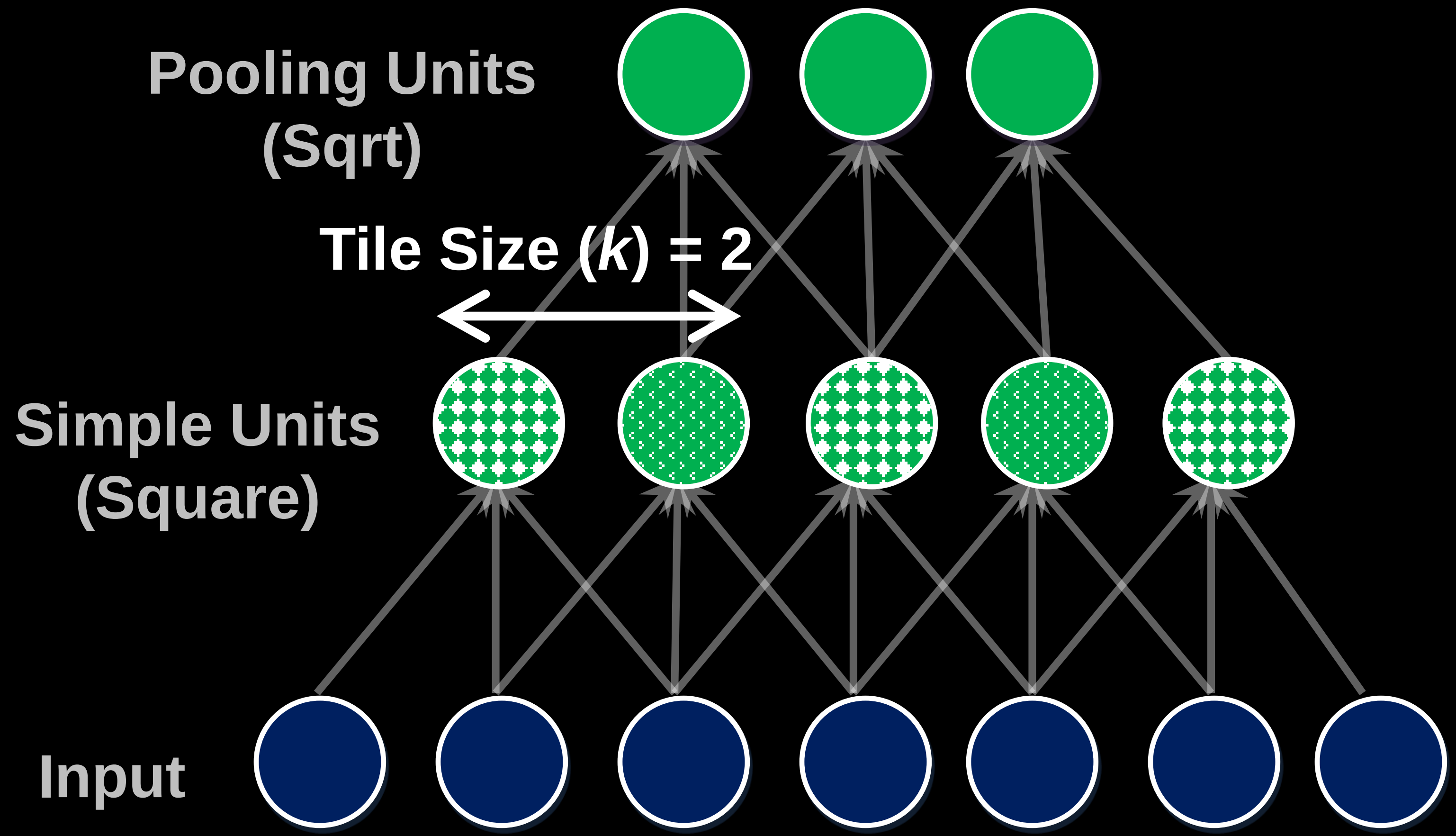


Tiled Networks (Partial Weight Tying)

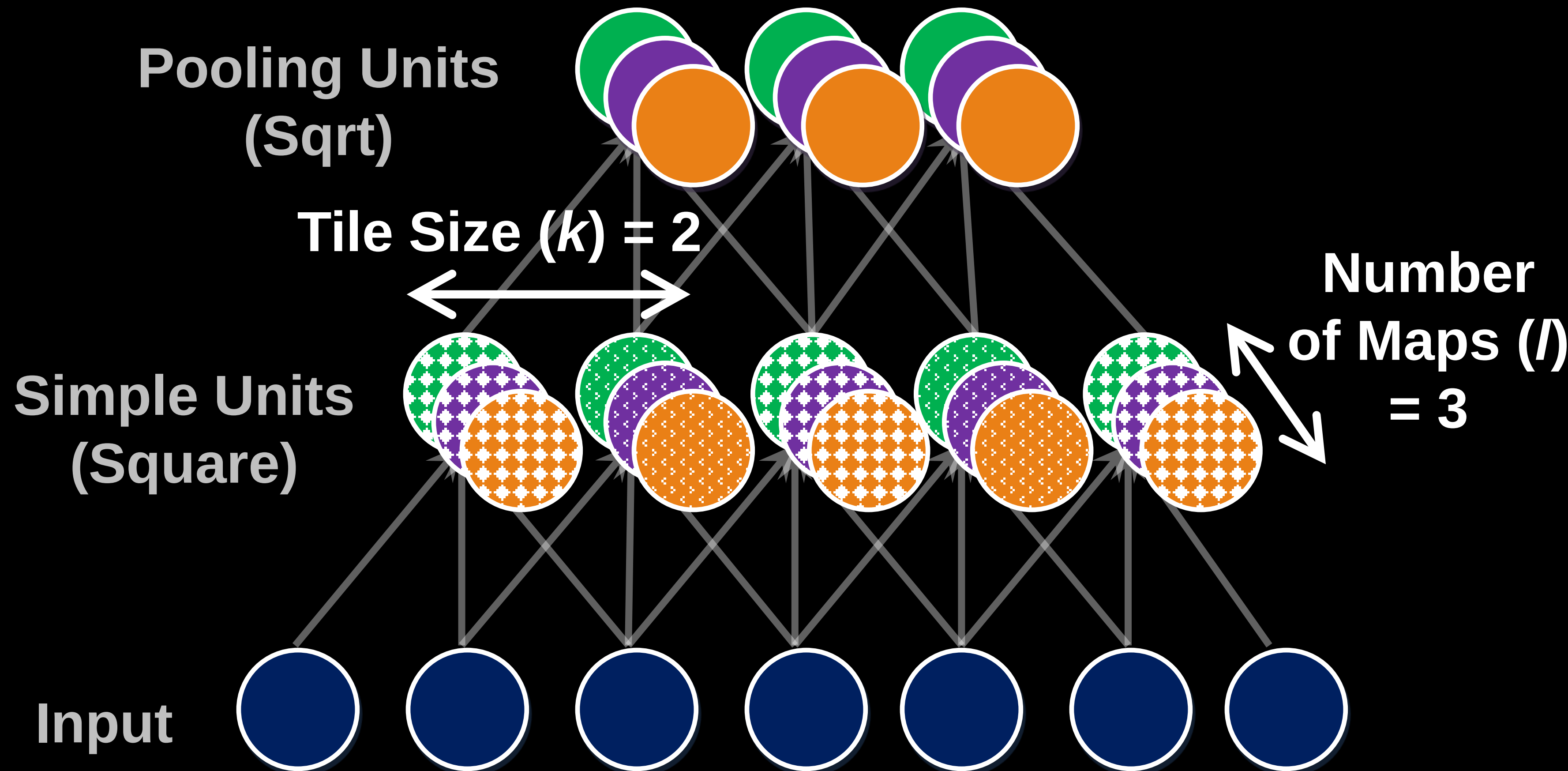


Local pooling can capture complex invariances (not just translation); but total number of parameters is small.

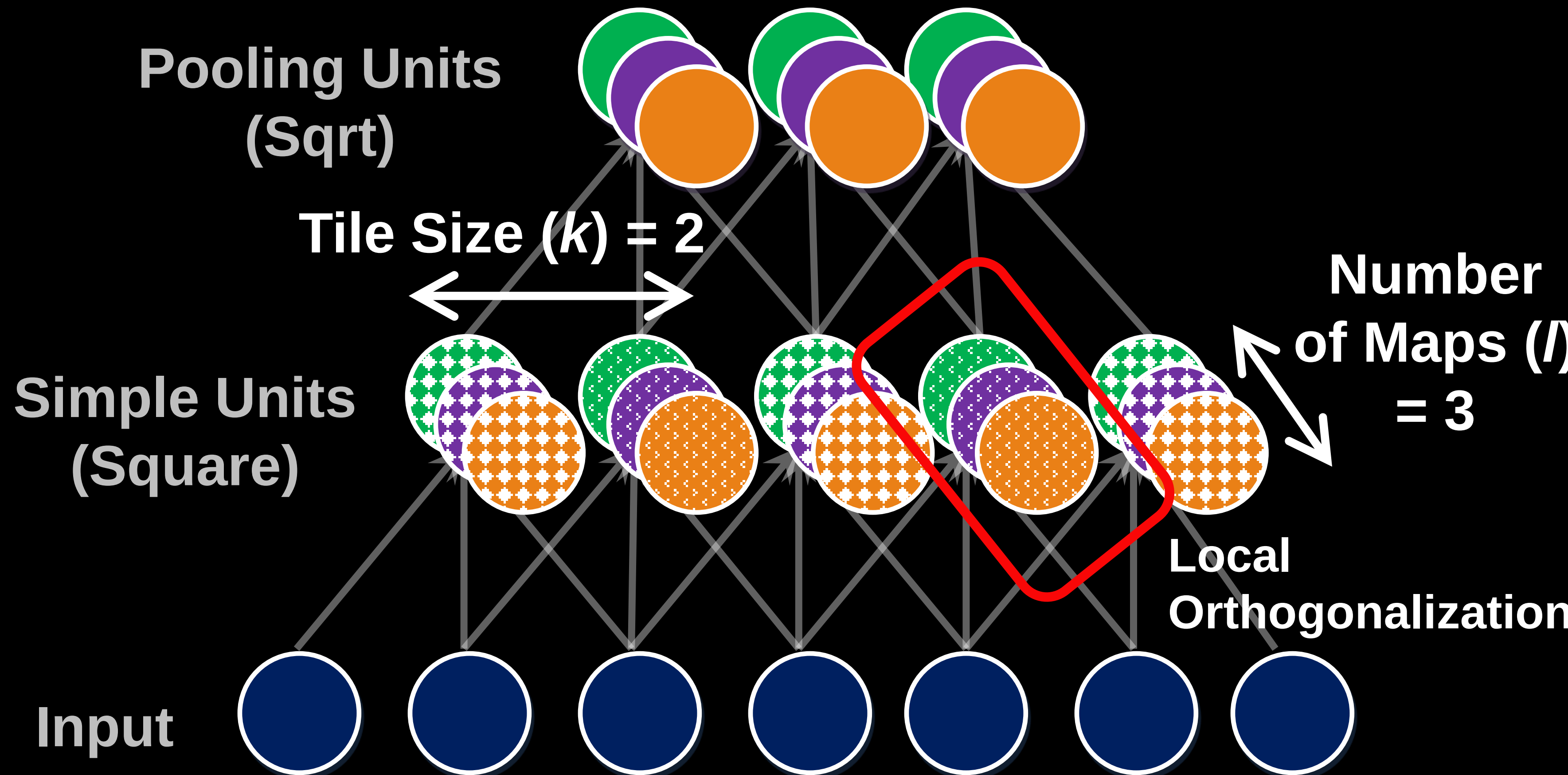
Tiled Networks (Partial Weight Tying)



Tiled Networks (Partial Weight Tying)



Tiled Networks (Partial Weight Tying)



NORB and CIFAR-10 results

Algorithms	NORB Accuracy
Deep Tiled CNNs [this work]	$\boxed{\textbf{96.1\%}}$
CNNs [Huang & LeCun, 2006]	94.1%
3D Deep Belief Networks [Nair & Hinton, 2009]	93.5%
Deep Boltzmann Machines [Salakhutdinov & Hinton, 2009]	92.8%
TICA [Hyvarinen et al., 2001]	89.6%
SVMs	88.4%

Algorithms	CIFAR-10 Accuracy
Improved LCC [Yu et al., 2010]	74.5%
Deep Tiled CNNs [this work]	$\boxed{\textbf{73.1\%}}$
LCC [Yu et al., 2010]	72.3%
mcRBMs [Ranzato & Hinton, 2010]	71.0%
Best of all RBMs [Krizhevsky, 2009]	64.8%
TICA [Hyvarinen et al., 2001]	56.1%

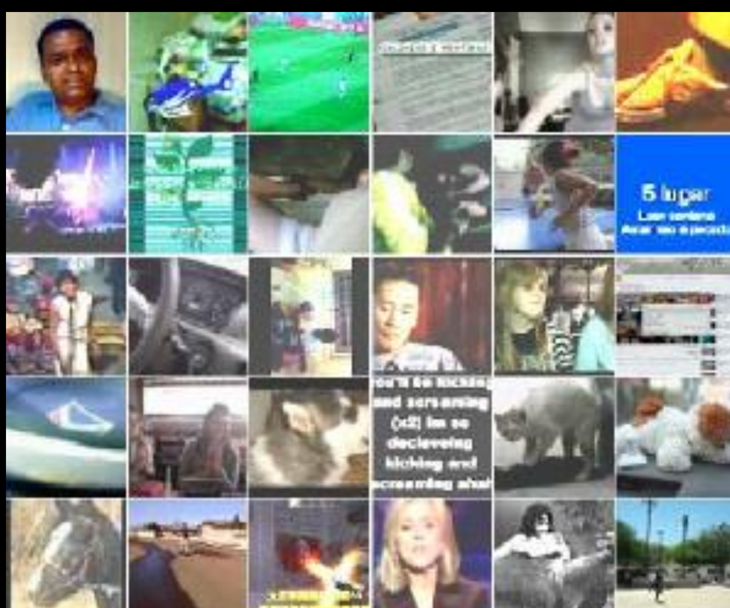
Scaling up: Discovering object classes

[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



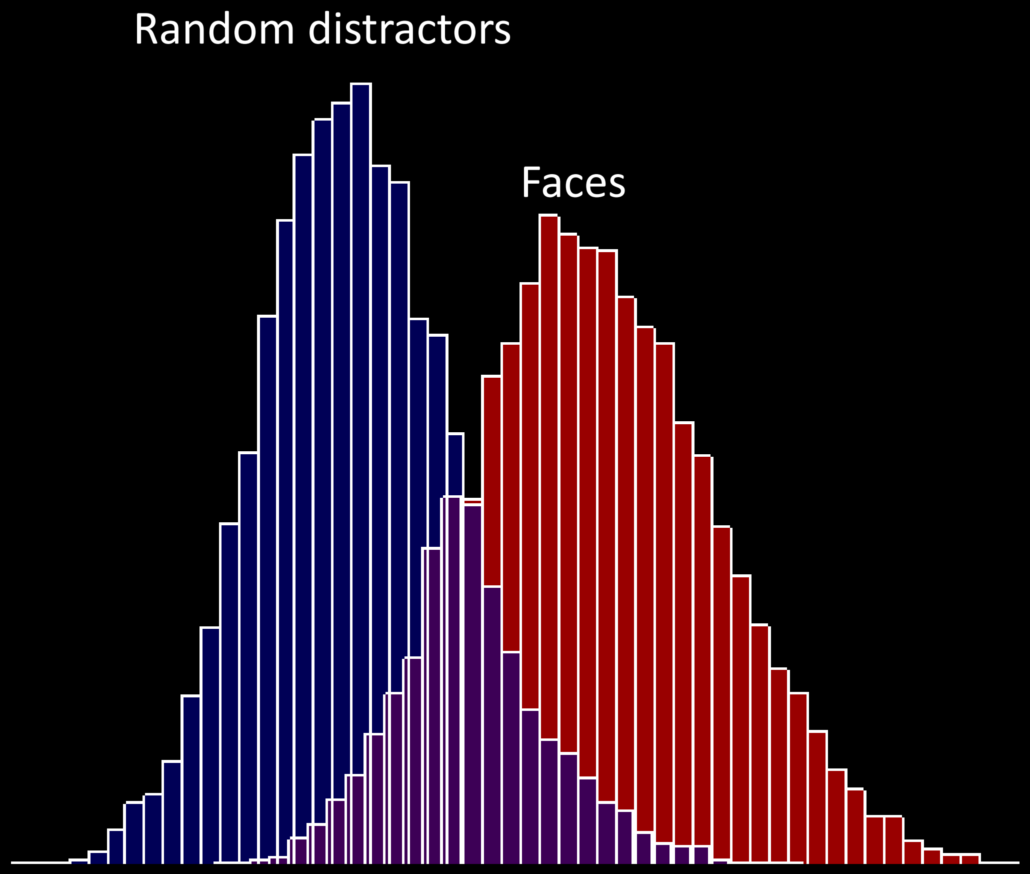
Training set (YouTube)

Test set (FITW + ImageNet)

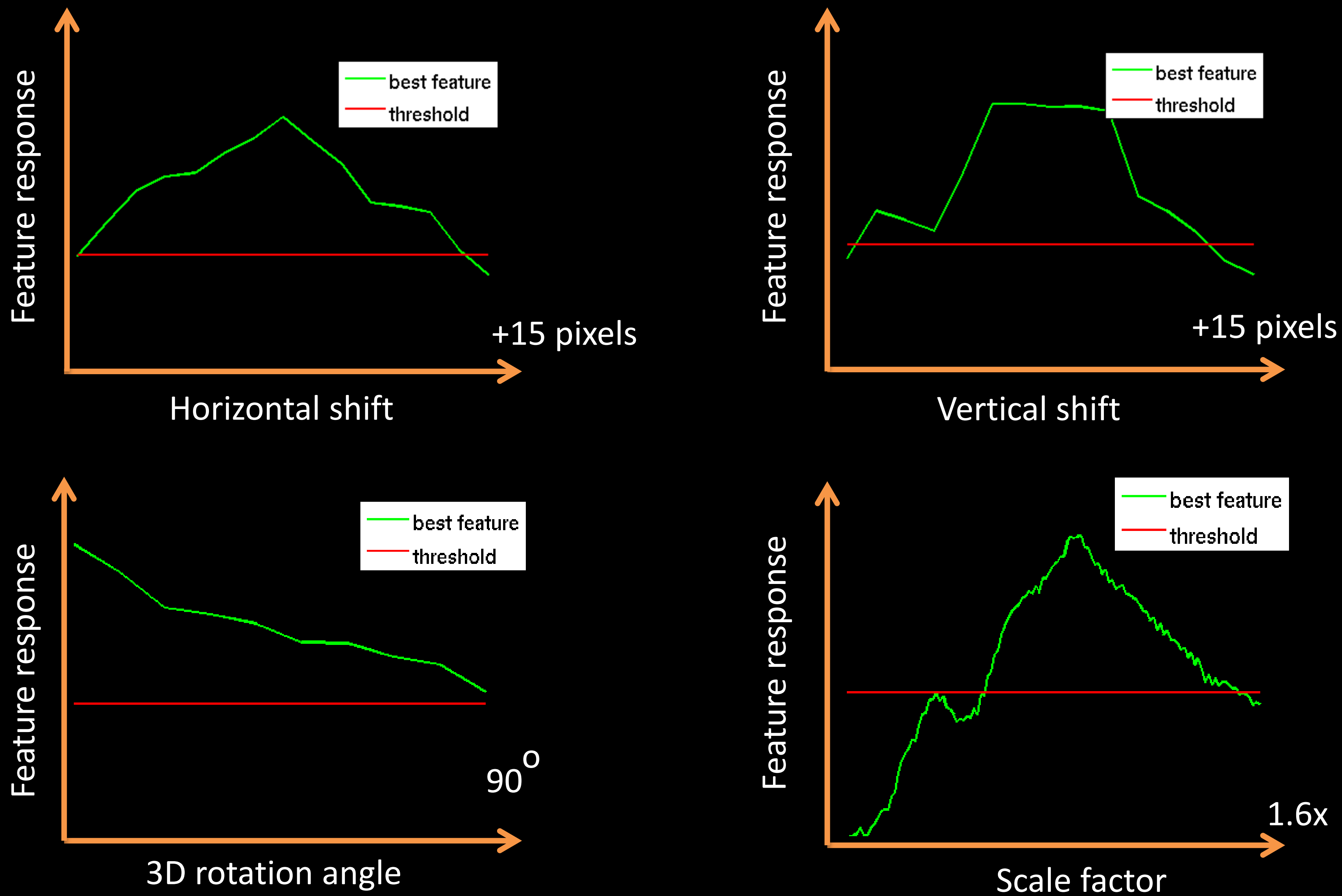
Face neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization


Random distractors
Faces

Invariance properties

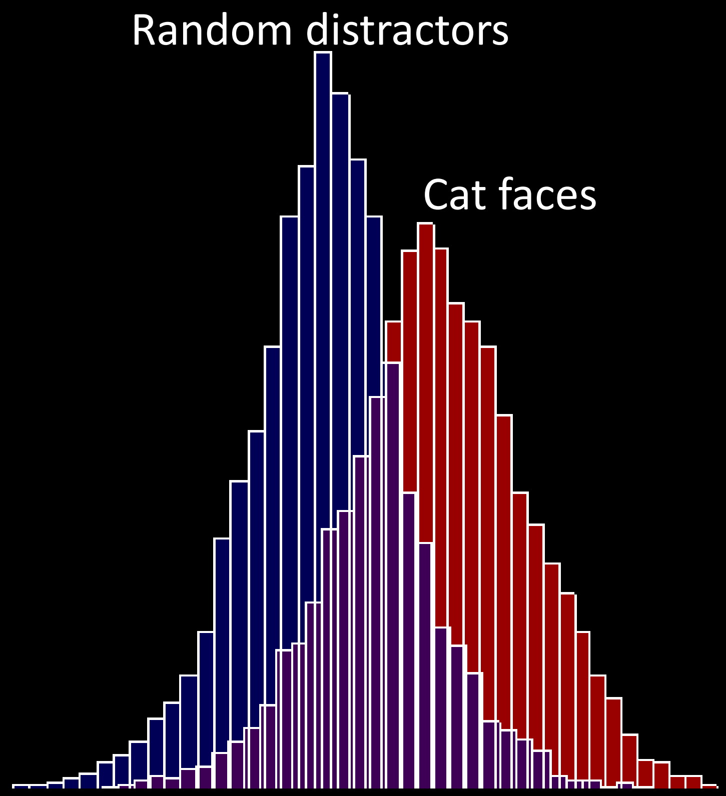


Cat neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization

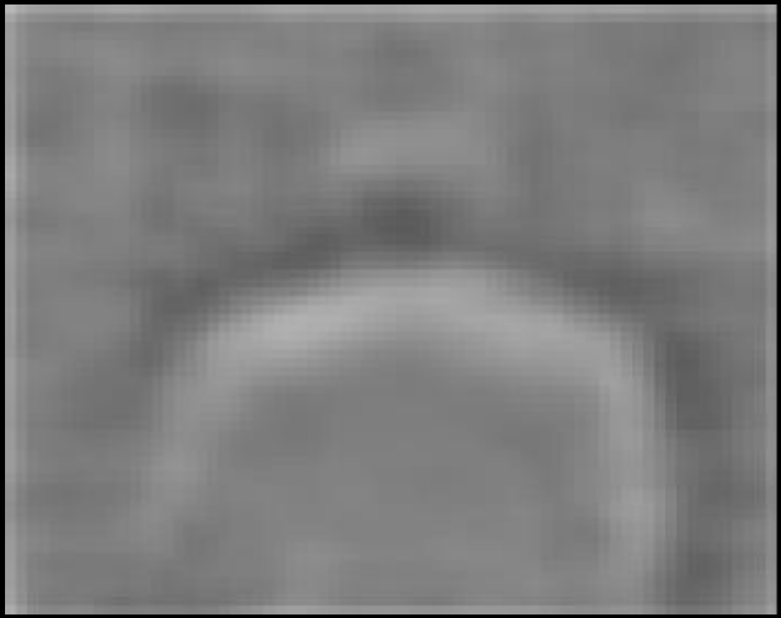
Cat face neuron



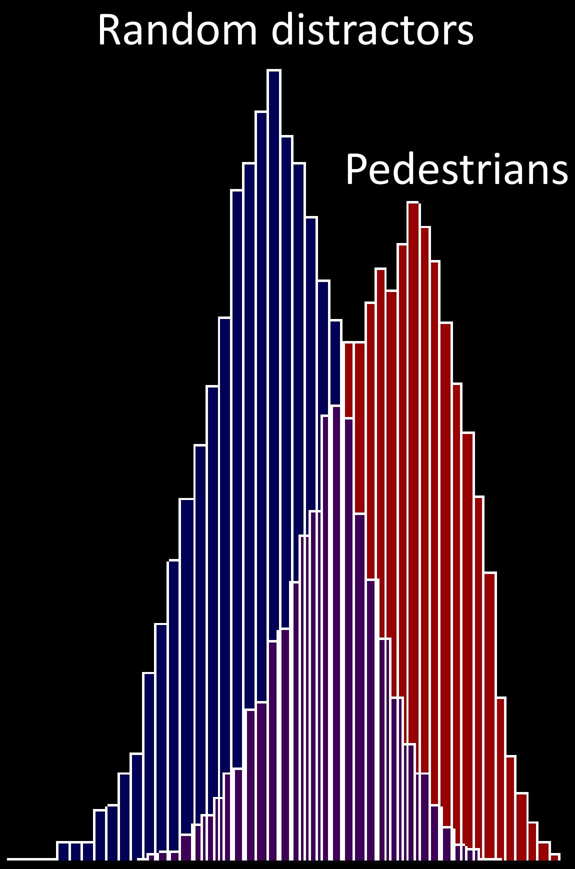
Visualization

Top Stimuli from the test set

Optimal stimulus by numerical optimization



Pedestrian neuron



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.

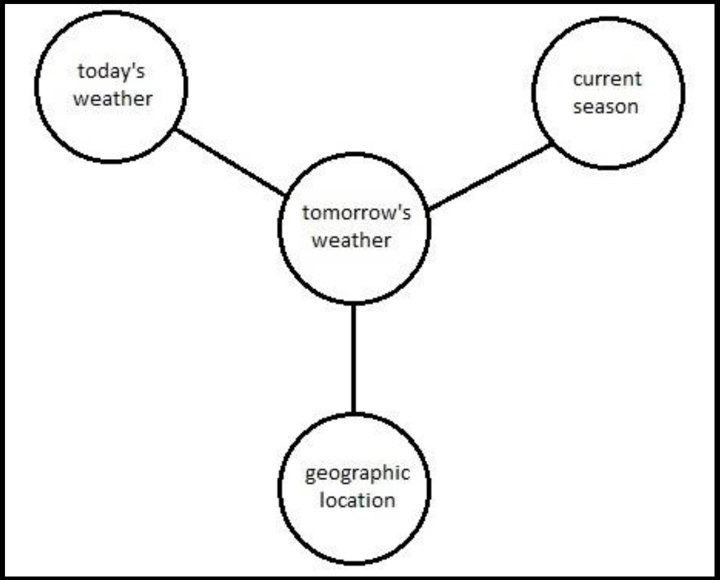
  A: Many of these were true, but not anymore (were not fundamental weaknesses). There's still work to be done though!

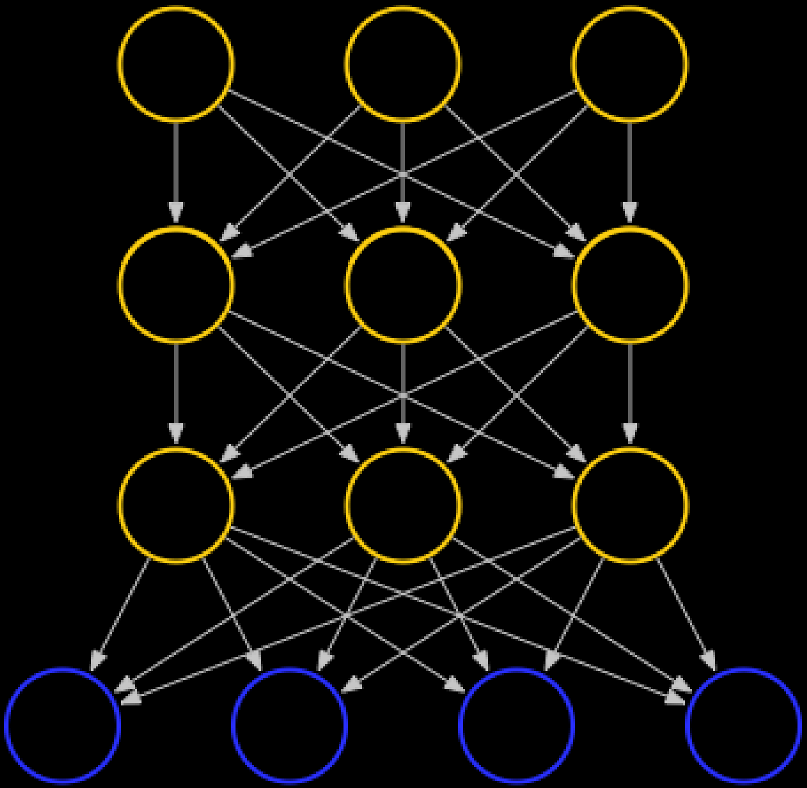
- We don't understand the learned features.

  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





Where these algorithms work

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

END END
END