Machine Learning and AI via Brain simulations

Andrew Ng

Stanford University & Google

Thanks to:

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

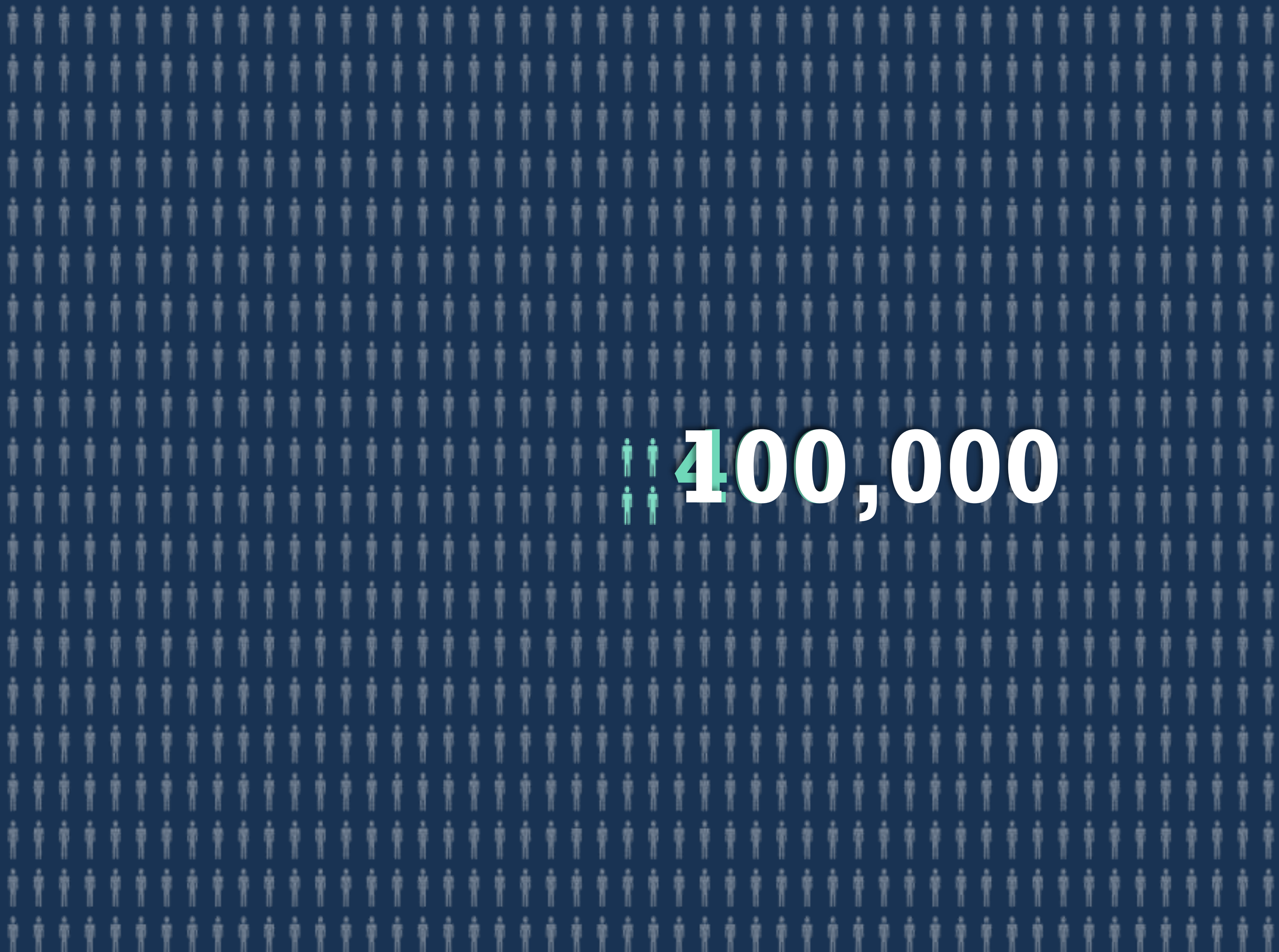

Matthieu Devin


Rajat Monga


Marc'Aurelio Ranzato


Paul Tucker


Kay Le


100,000

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.

Vision is not only mine; shared with many researchers:

E.g., Samy Bengio, Yoshua Bengio, Tom Dean, Jeff Dean, Nando de Freitas, Jeff Hawkins, Geoff Hinton, Quoc Le, Yann LeCun, Honglak Lee, Tommy Poggio, Ruslan Salakhutdinov, 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”
Suggest tags
Image search
…

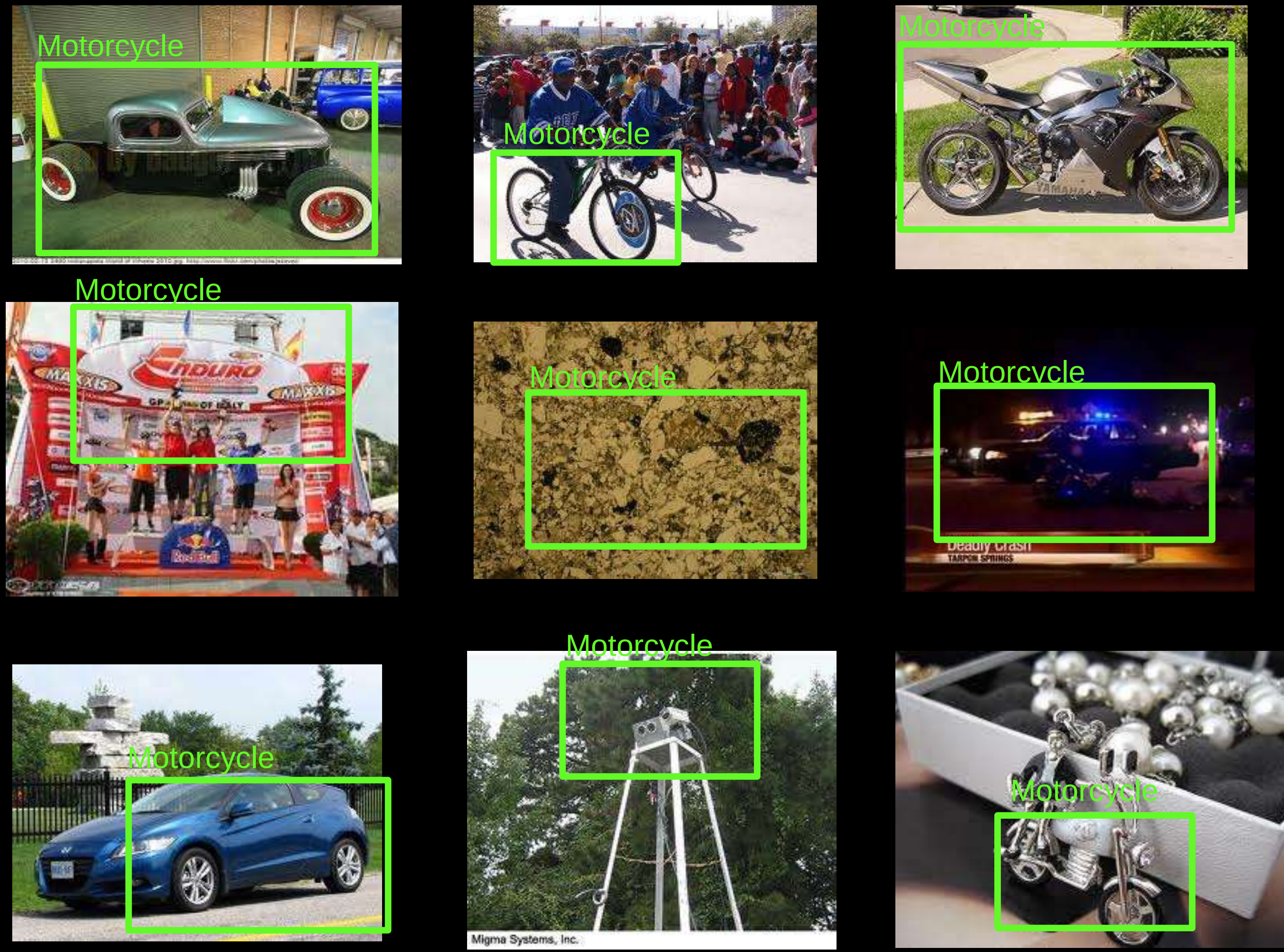
Audio

Speech recognition
Music classification
Speaker identification
…

Text

Web search
Anti-spam
Machine translation
…

Computer vision is hard!



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

The New York Times

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 for image classification

“Motorcycle”

This talk: Develop ideas using images and audio.
Ideas apply to other problems (e.g., text) too.

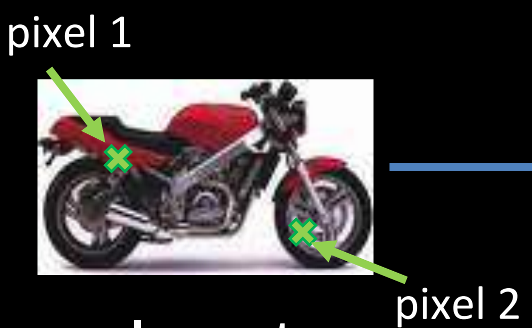
Why is this hard?

You see this:

But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Machine learning and feature representations

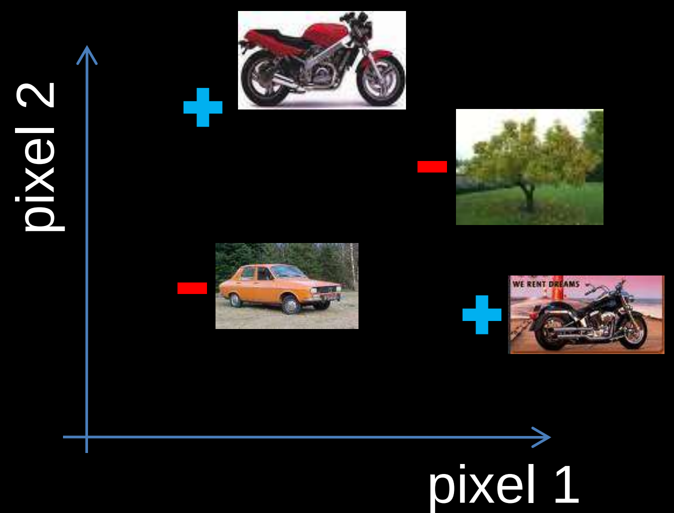


Input

Learning algorithm

Motorbikes
"Non"-Motorbikes

Raw image



Machine learning and feature representations

pixel 1

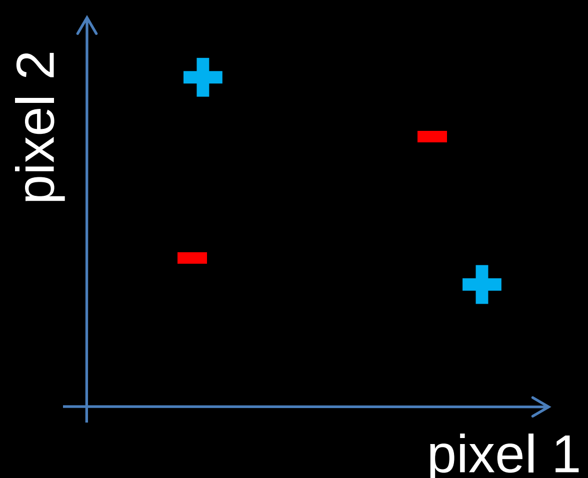
pixel 2

Input

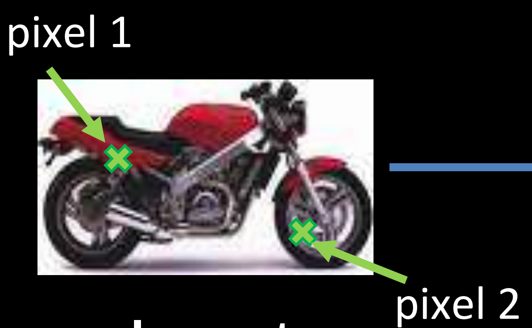
Learning algorithm

Motorbikes
“Non”-Motorbikes

Raw image



Machine learning and feature representations

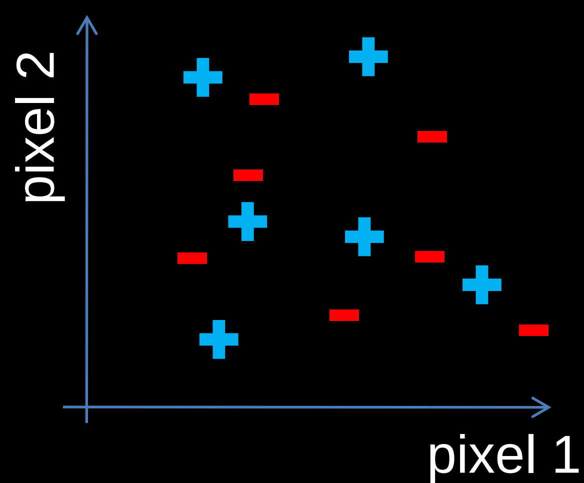


Input

Learning algorithm

Raw image

✚ Motorbikes
– “Non”-Motorbikes



What we want

Input

Feature representation

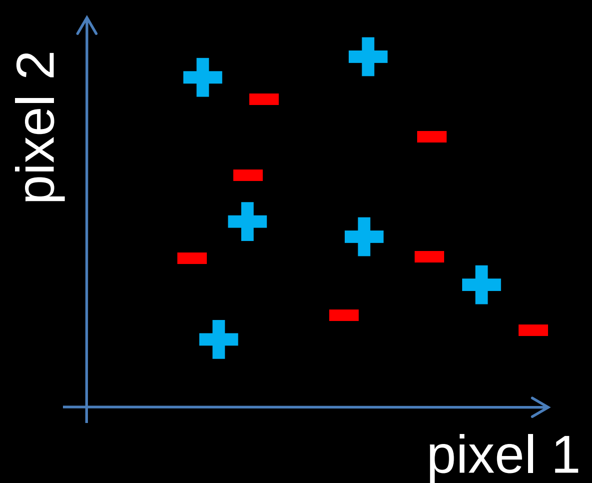
Learning algorithm

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

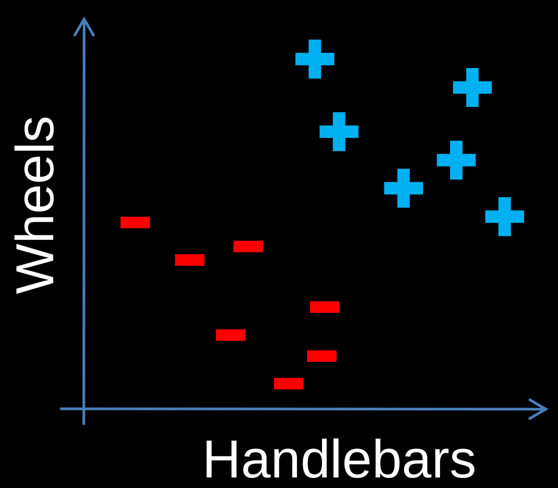
Motorbikes

"Non"-Motorbikes

Raw image



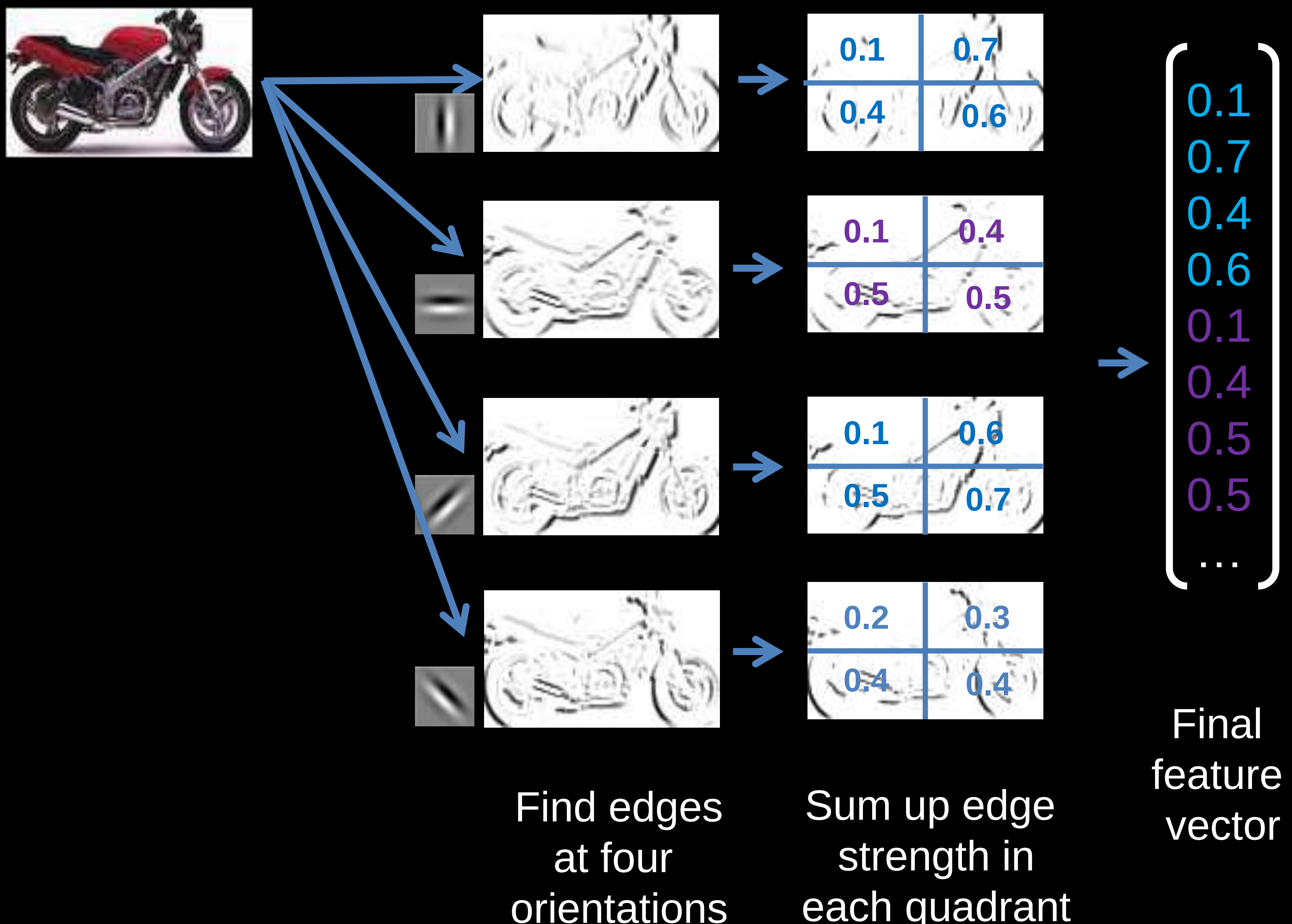
Features



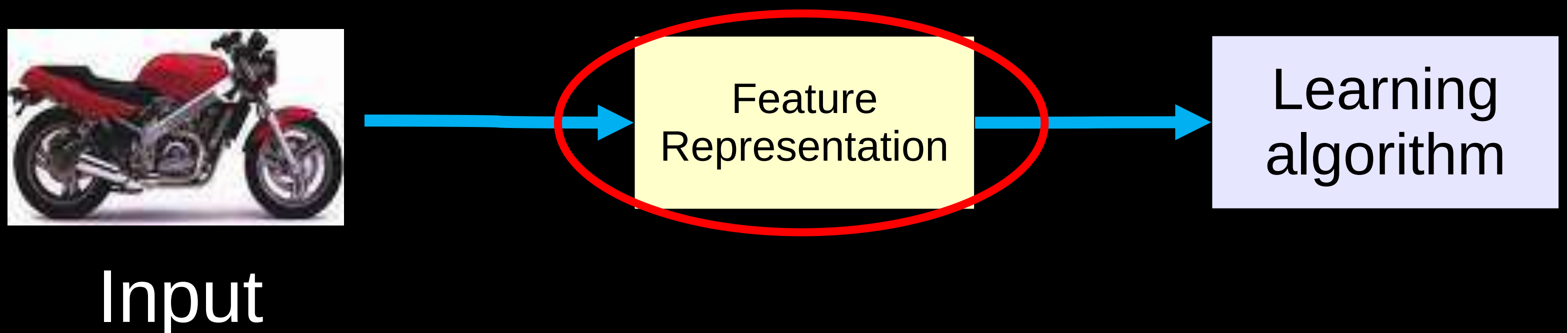
Andrew Ng

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.



Feature representations



How is computer perception done?

Images/video

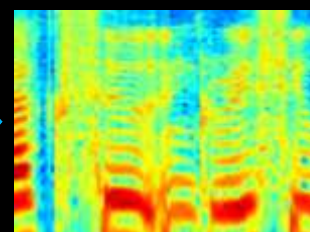
Image

Vision features

Detection

Audio

Audio

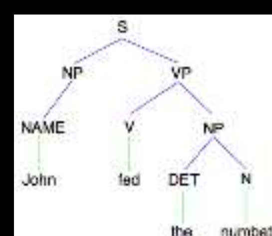


Audio features

Speaker ID

Text

Text



Text features

Text classification,
Machine translation,
Information retrieval,
....

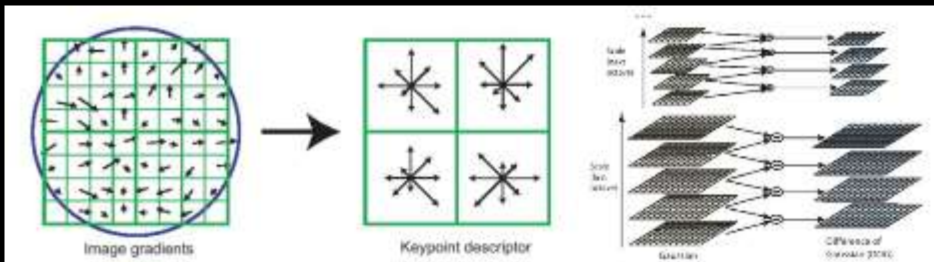
Feature representations

Input

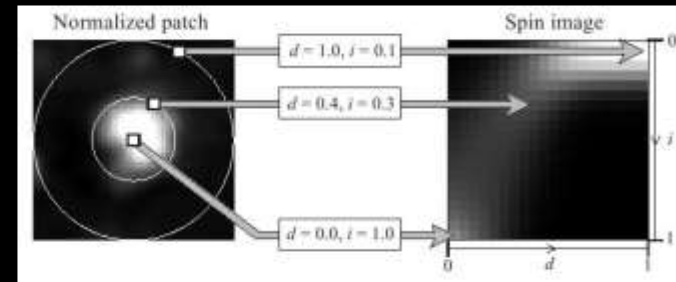
Feature Representation

Learning algorithm

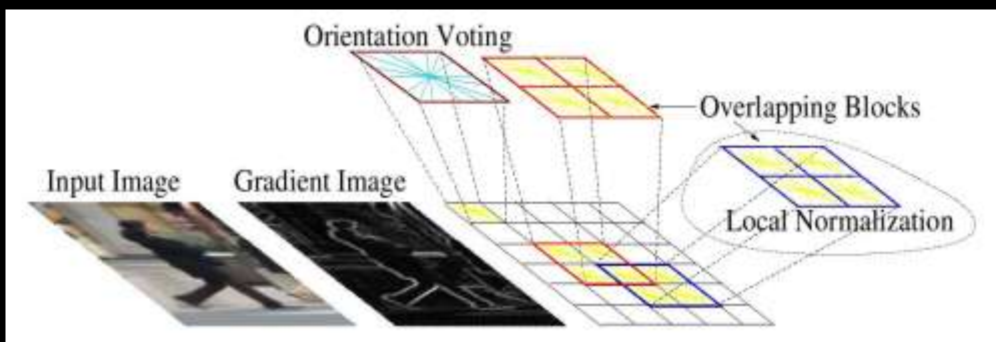
Computer vision features



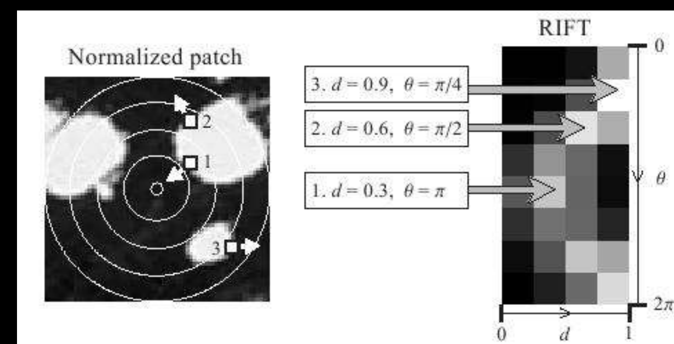
SIFT



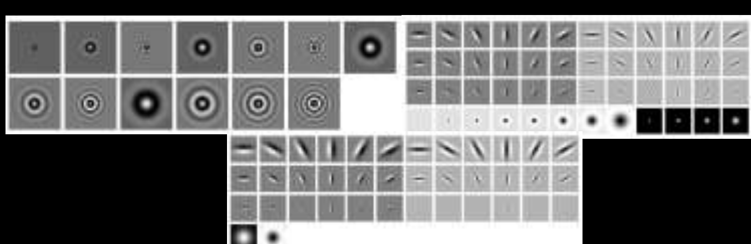
Spin image



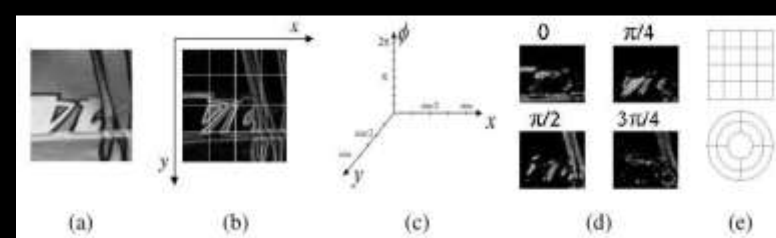
HoG



RIFT

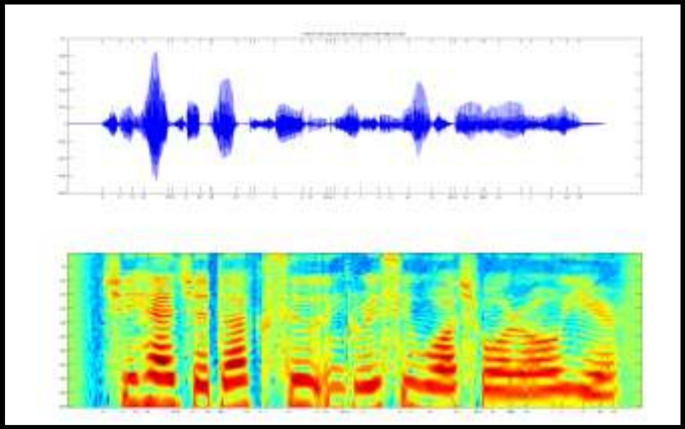


Textons

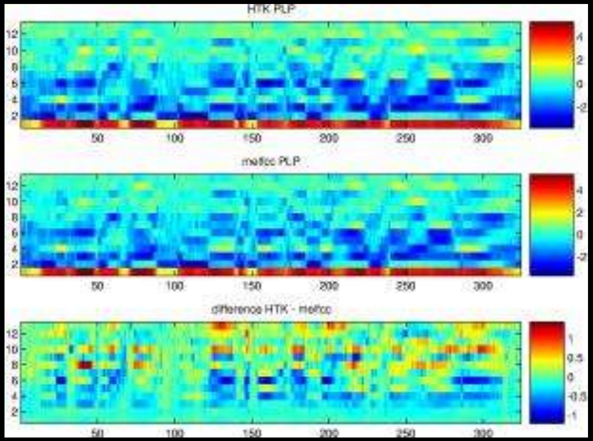


GLOH

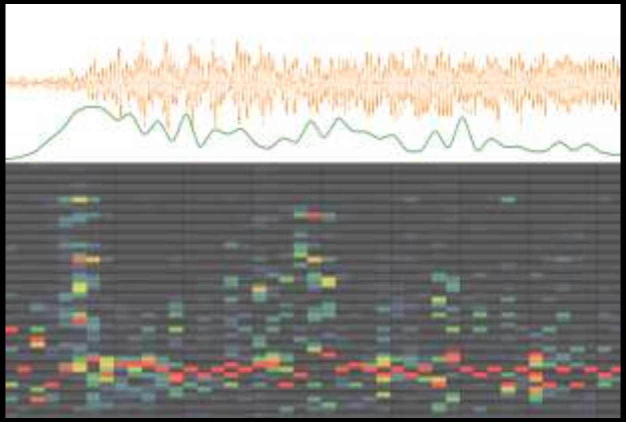
Audio features



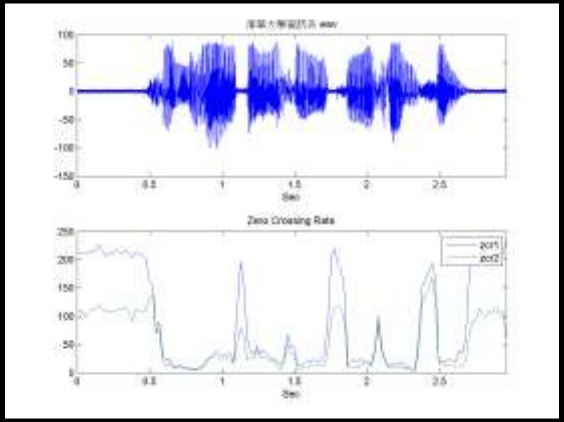
Spectrogram



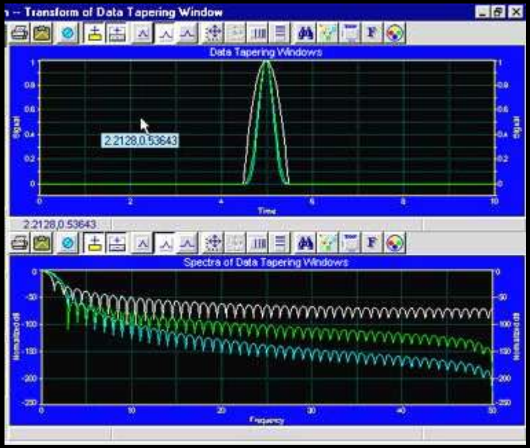
MFCC



Flux

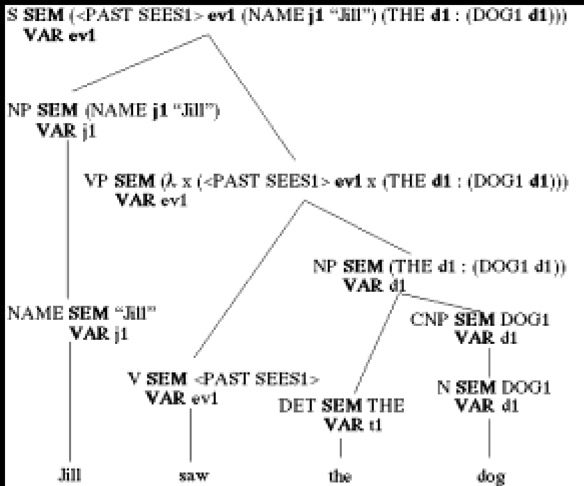


ZCR

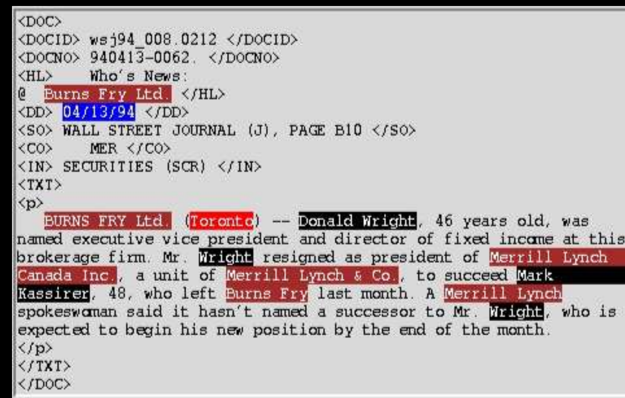


Rolloff

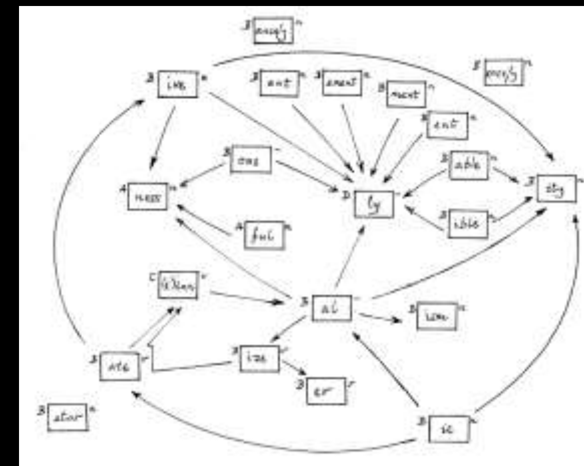
NLP features



Pars



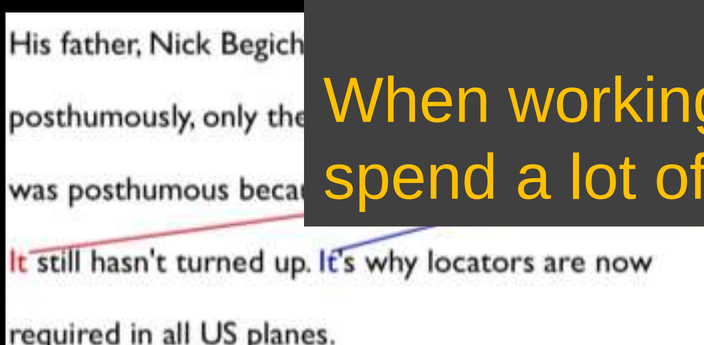
Named entity recognition



Stemming

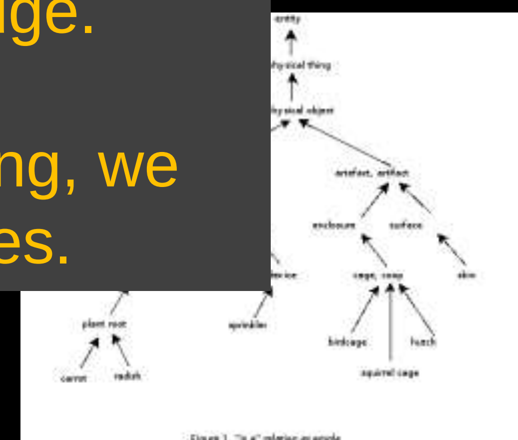
Coming up with features is difficult, time-consuming, requires expert knowledge.

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



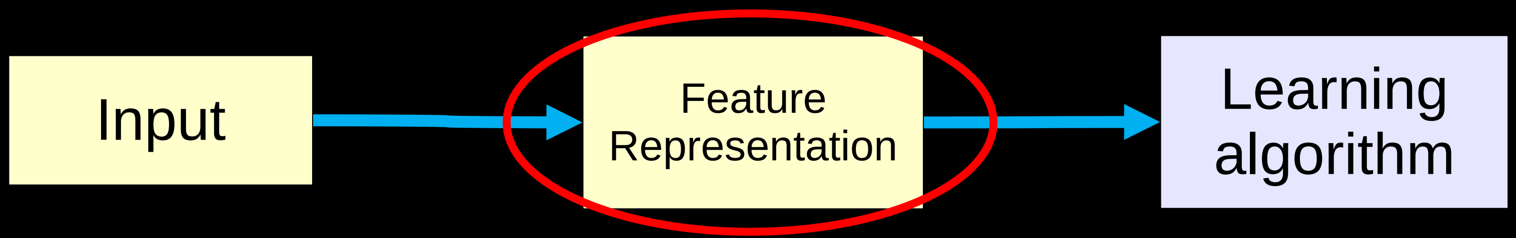
Anaphora

Part of speech

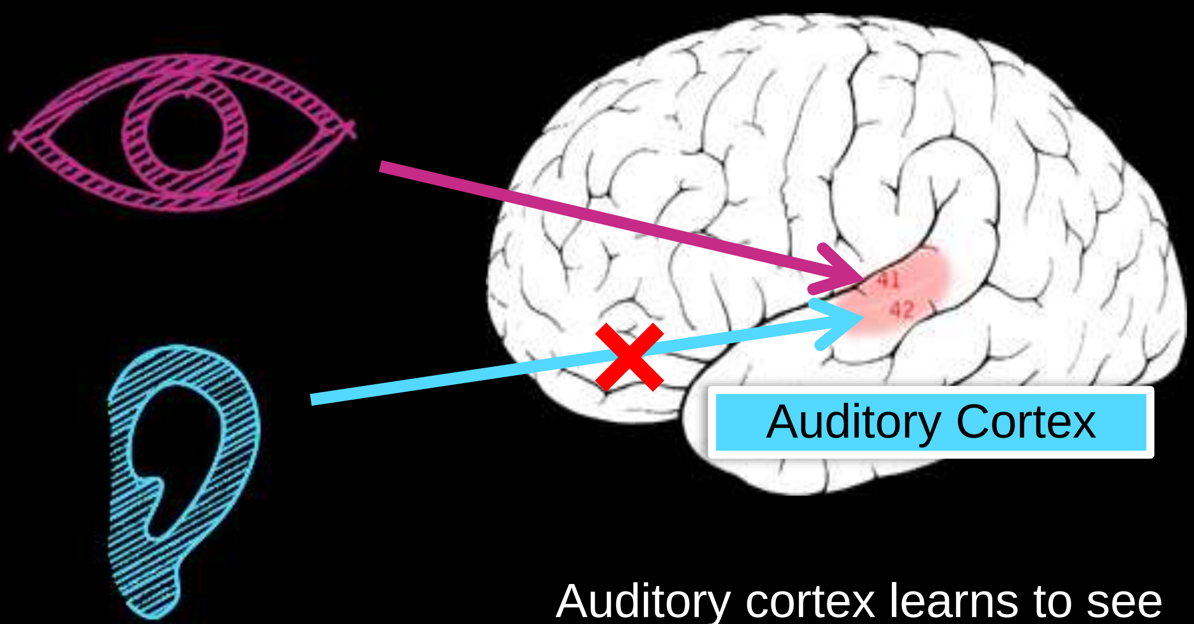


Ontologies (WordNet)

Feature representations



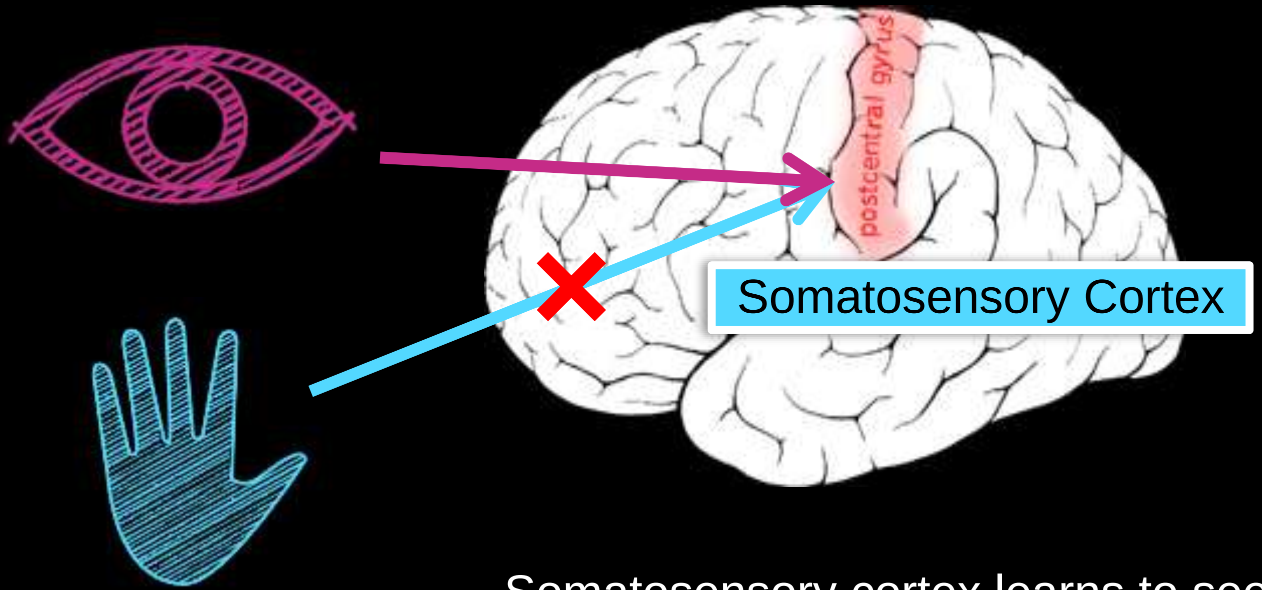
The “one learning algorithm” hypothesis



Auditory cortex learns to see

[Roe et al., 1992]

The “one learning algorithm” hypothesis



Somatosensory cortex learns to see

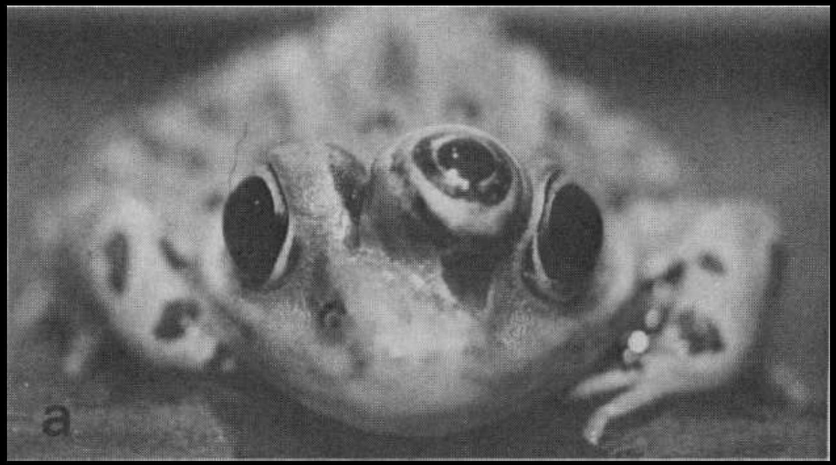
[Metin & Frost, 1989]

Sensor representations in the brain

Seeing with your tongue

Human echolocation (sonar)

Haptic belt: Direction sense



Implanting a 3rd eye

[BrainPort; Welsh & Blasch, 1997; Nagel et al., 2005; Constantine-Paton & Law, 2009]

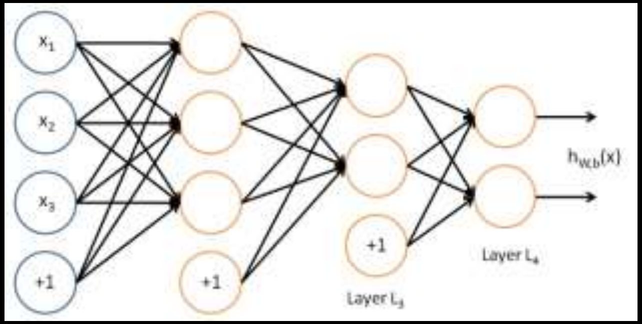
On two approaches to computer perception

The adult visual system computes an incredibly complicated function of the input.

We can try to directly implement most of this incredibly complicated function (hand-engineer features).

Can we learn this function instead?

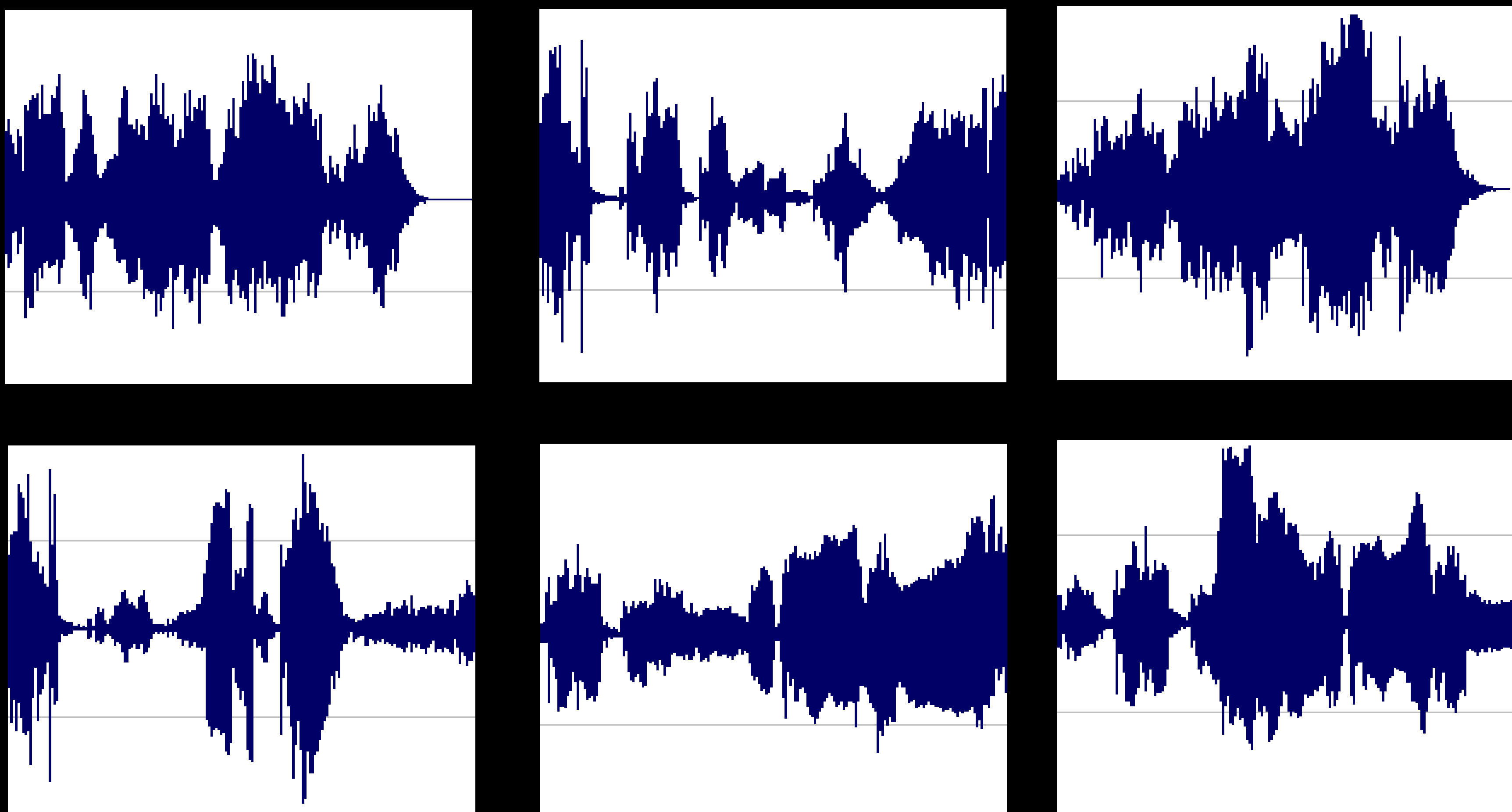
A trained learning algorithm (e.g., neural network, boosting, decision tree, SVM,…) is very complex. But the learning algorithm itself is usually very simple. The complexity of the trained algorithm comes from the data, not the algorithm.



Learning input representations

Find a better way to represent images than pixels.

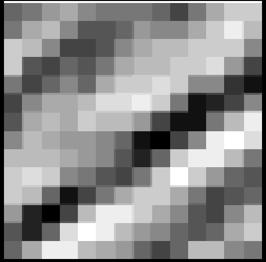
Learning input representations



Find a better way to represent audio.

Feature learning problem

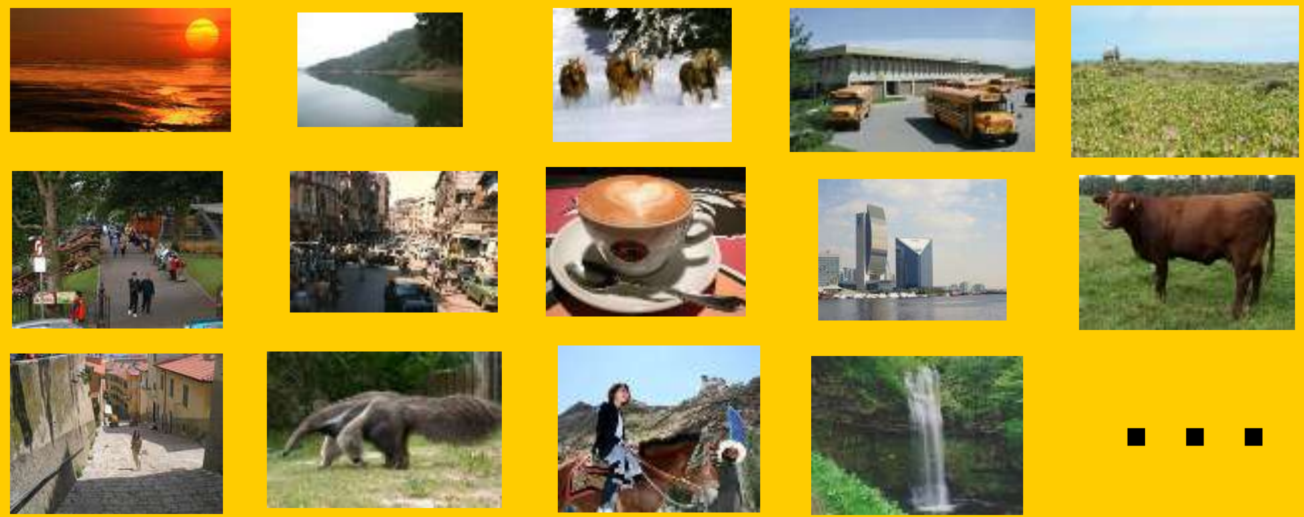
- Given a 14x14 image patch x, can represent it using 196 real numbers.



$$\begin{pmatrix} 255 \\ 98 \\ 93 \\ 87 \\ 89 \\ 91 \\ 48 \\ \ldots \end{pmatrix}$$

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

Self-taught learning (Unsupervised Feature Learning)



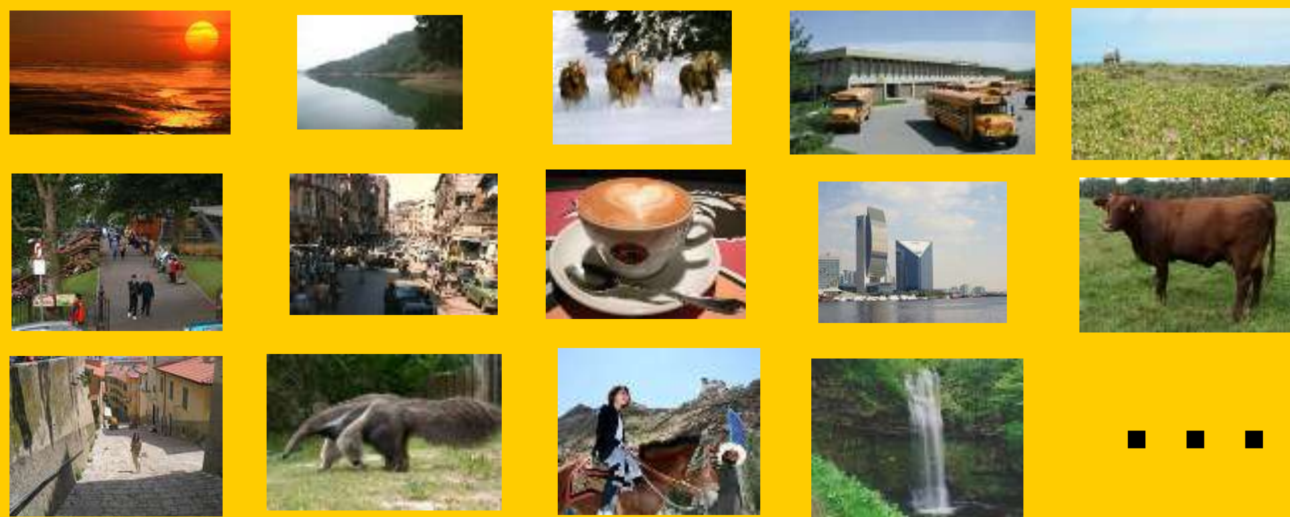
Unlabeled images

Motorcycles

Not motorcycles

Testing:
What is this?

Self-taught learning (Unsupervised Feature Learning)



. . .

Unlabeled images

Motorcycles

Not motorcycles

Testing:
What is this?

First stage of visual processing: V1

V1 is the first stage of visual processing in the brain.

Neurons in V1 typically modeled as edge detectors:

Neuron #1 of visual cortex
(model)

Neuron #2 of visual cortex
(model)

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 $R^{n \times n}$)

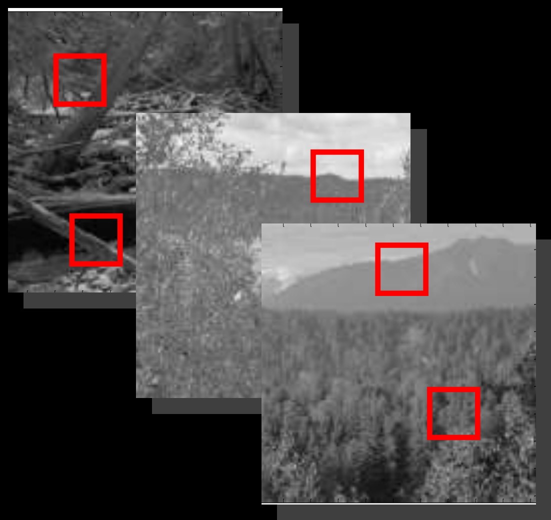
Learn: Dictionary of bases $\phi_1, \phi_2, \ldots, \phi_k$ (also $R^{n \times n}$), so that each input x can be approximately decomposed as:

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

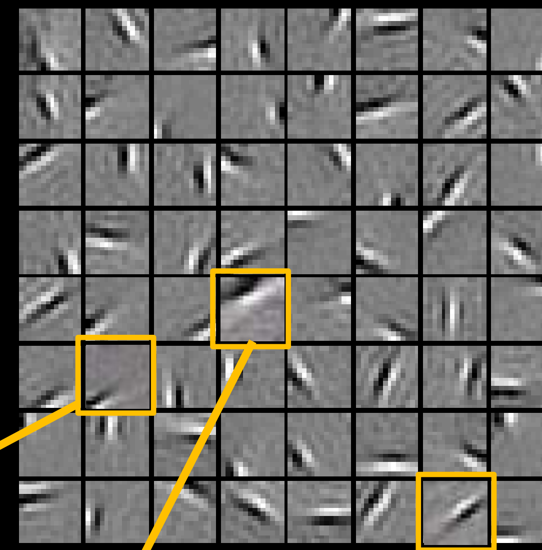
s.t. a_j's are mostly zero ("sparse")

Sparse coding illustration

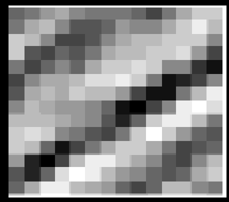
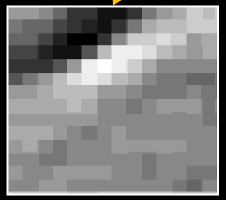
Natural Images



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



Test example

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

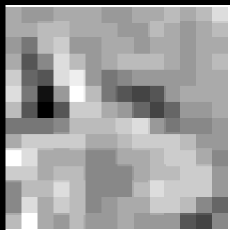
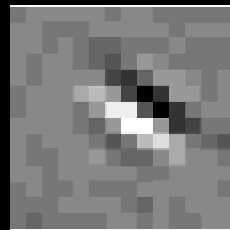
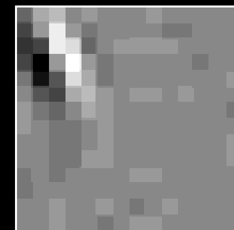
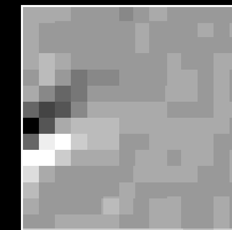
$$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$$

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

(feature representation)

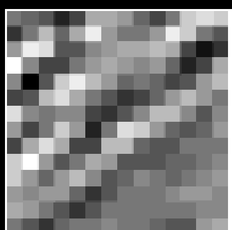
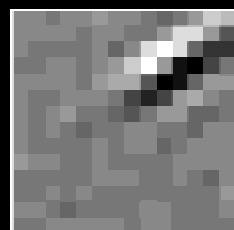
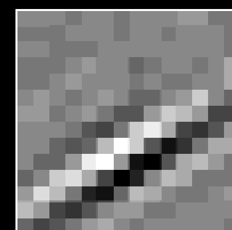
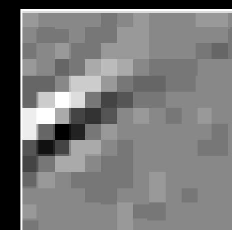
More succinct, higher-level, representation.

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].

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

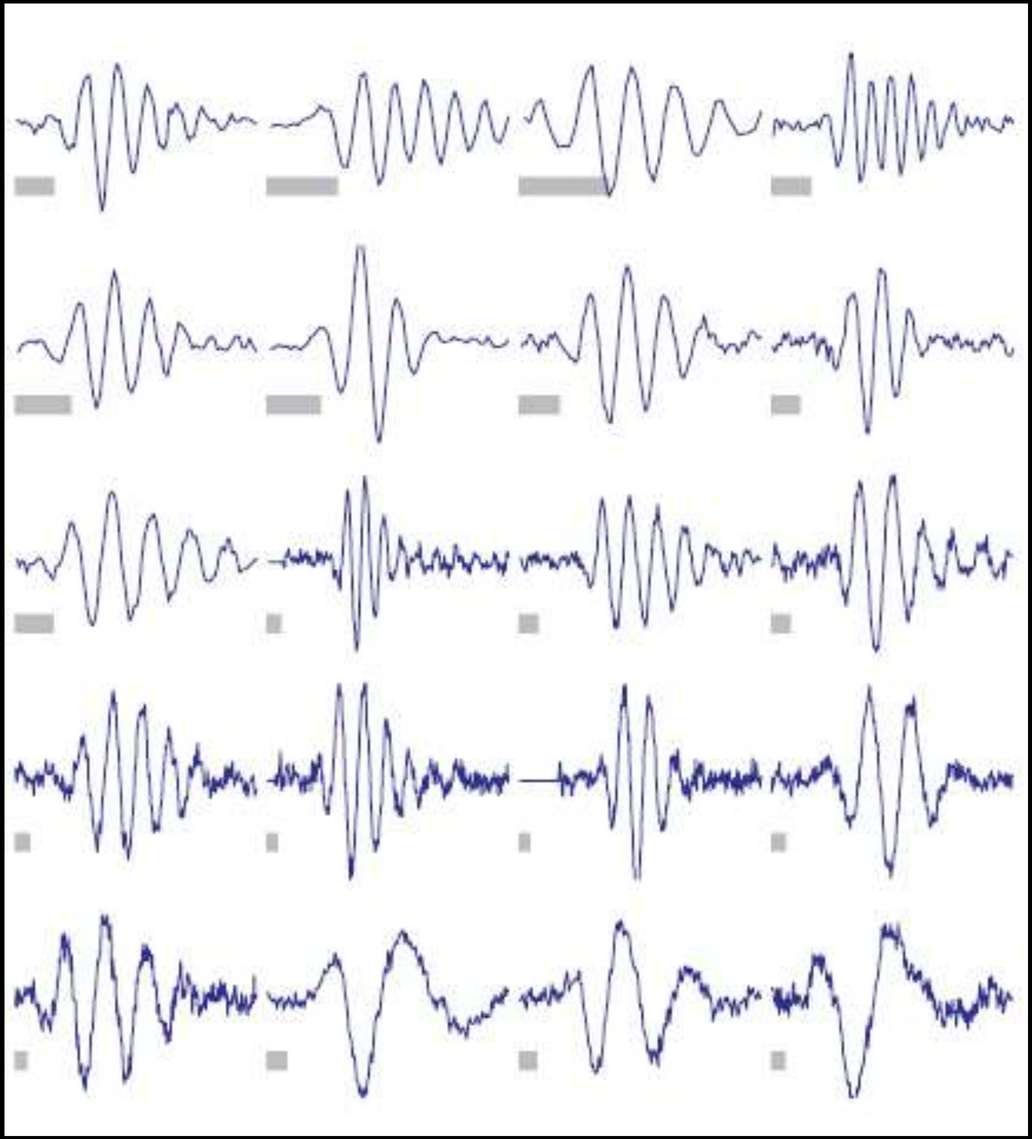
ϕ_{5} ϕ_{18} ϕ_{29}

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

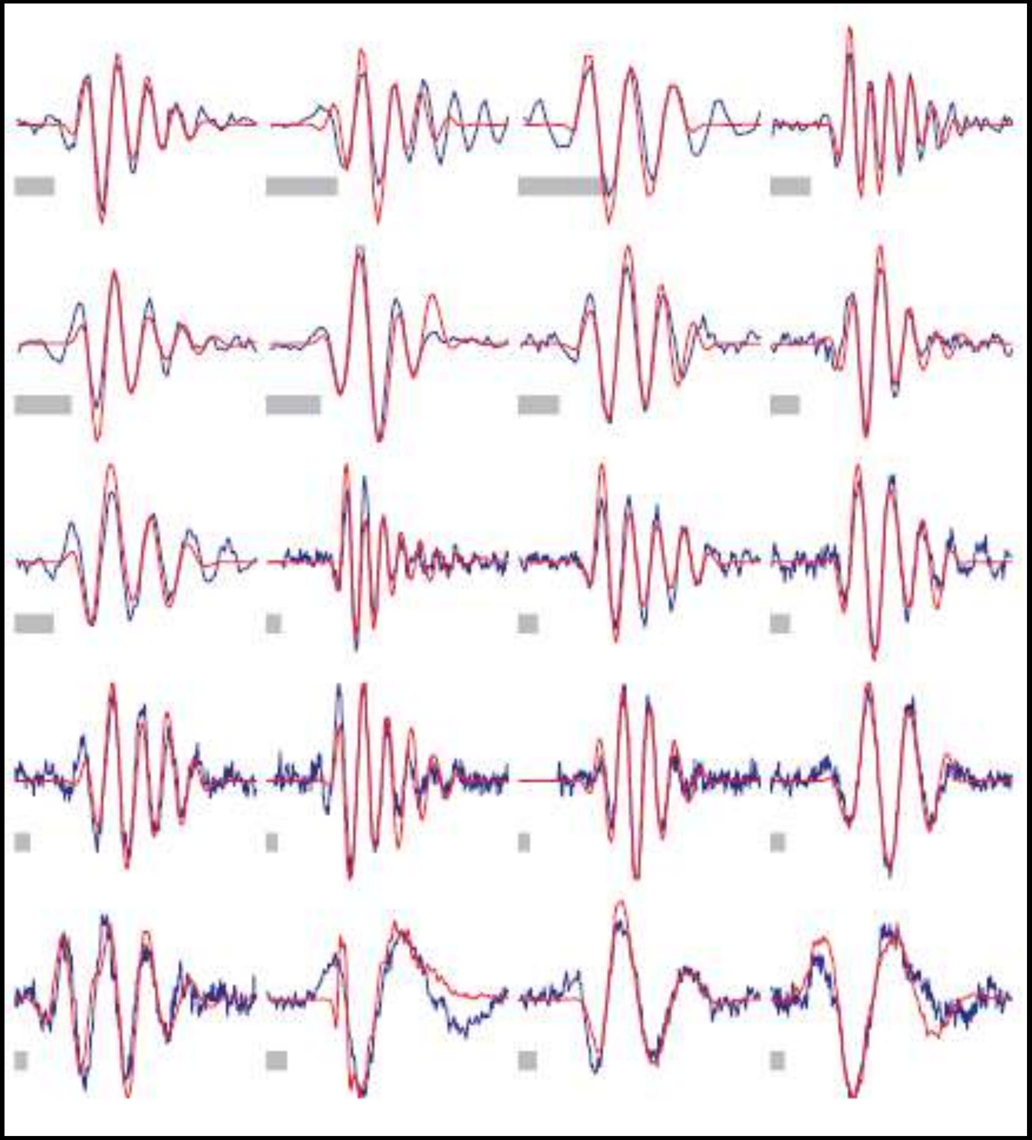
Image shows 20 basis functions learned from unlabeled audio.



[Evan Smith & Mike Lewicki, 2006]

Sparse coding applied to audio

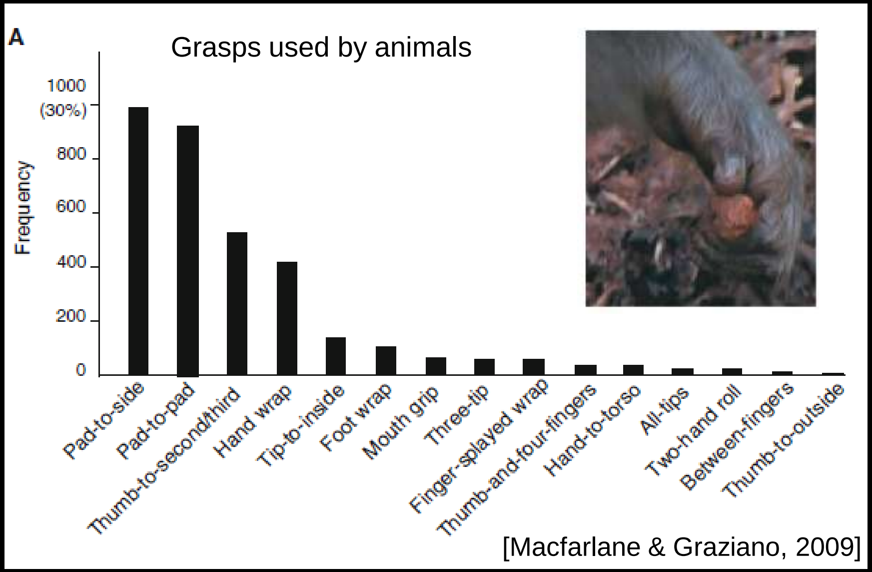
Image shows 20 basis functions learned from unlabeled audio.



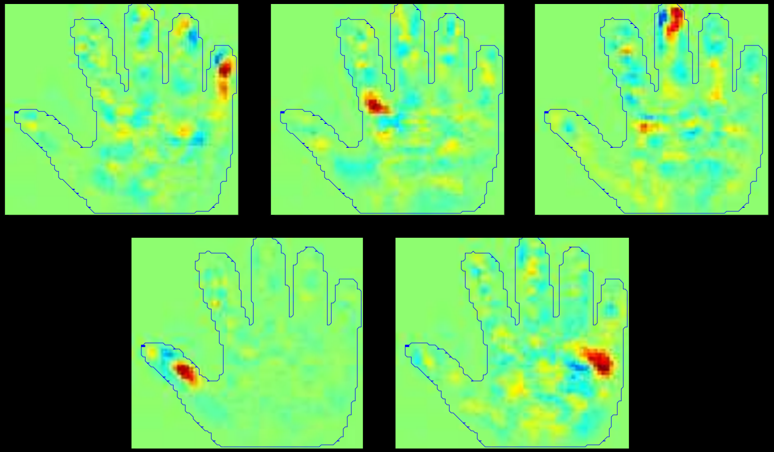
[Evan Smith & Mike Lewicki, 2006]

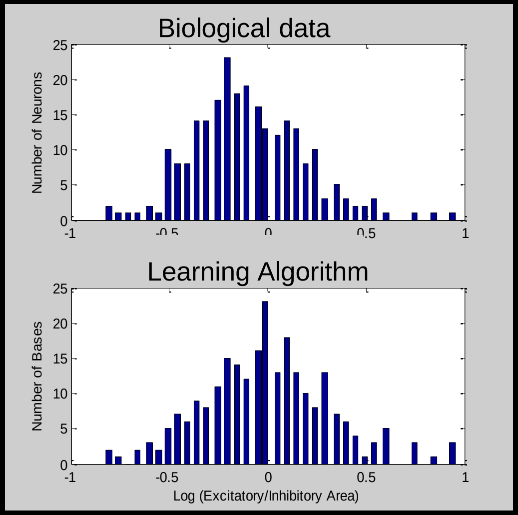
Sparse coding applied to touch data

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



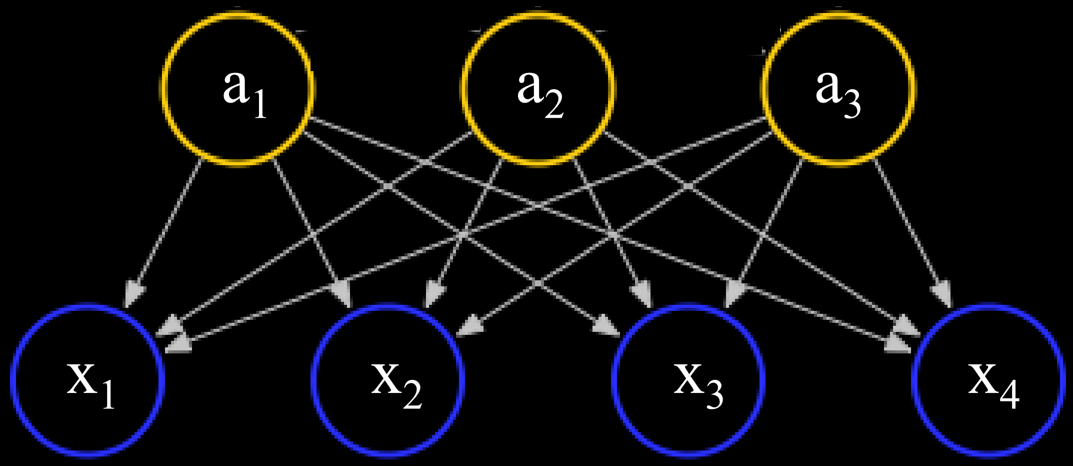
Example learned representations





[Andrew Saxe]

Learning feature hierarchies



Higher layer
(Combinations of edges;
cf V2)

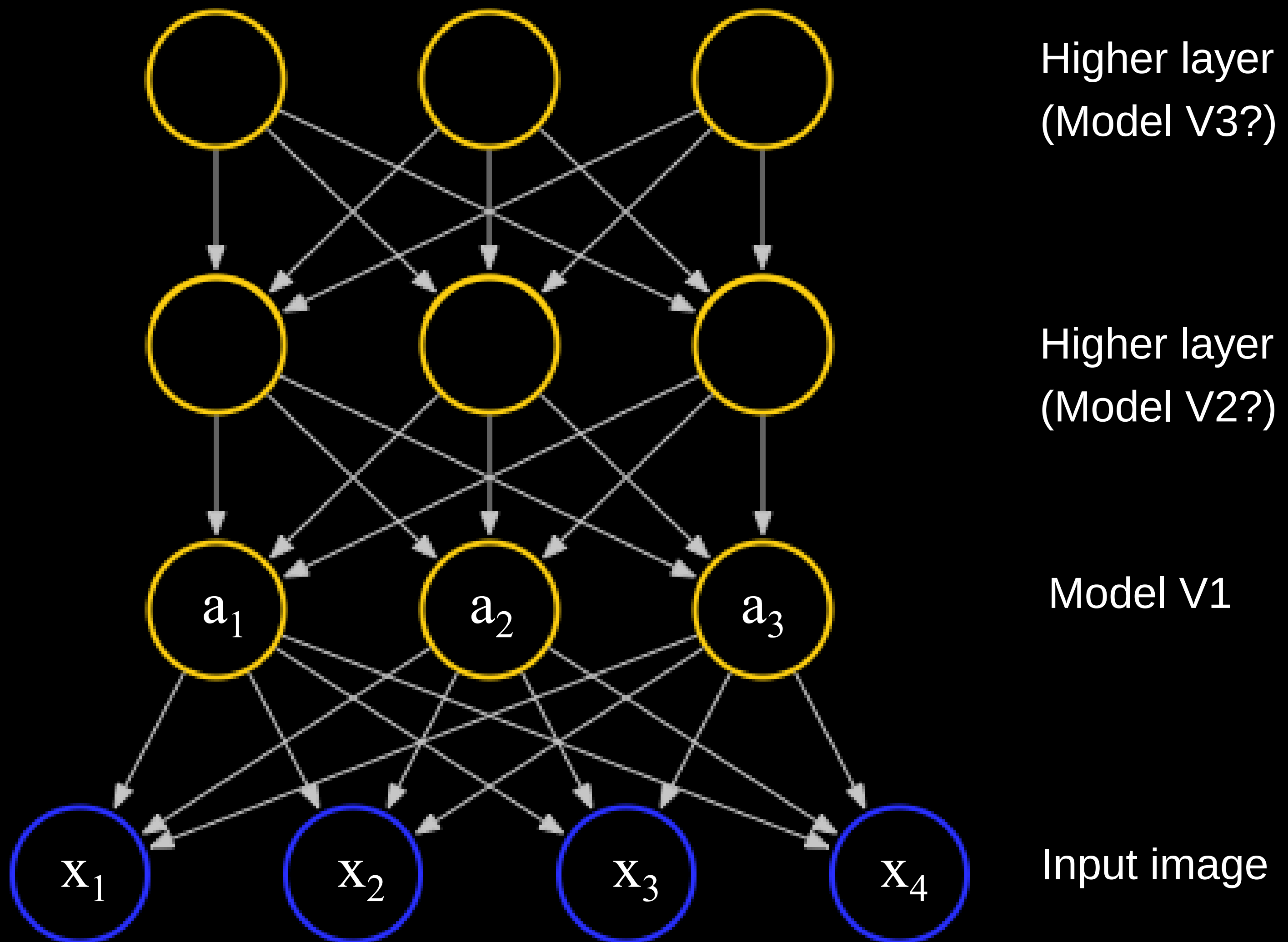
"Sparse coding"
(edges; cf. V1)

Input image (pixels)

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

[Lee, Ranganath & Ng, 2007]

Learning feature hierarchies

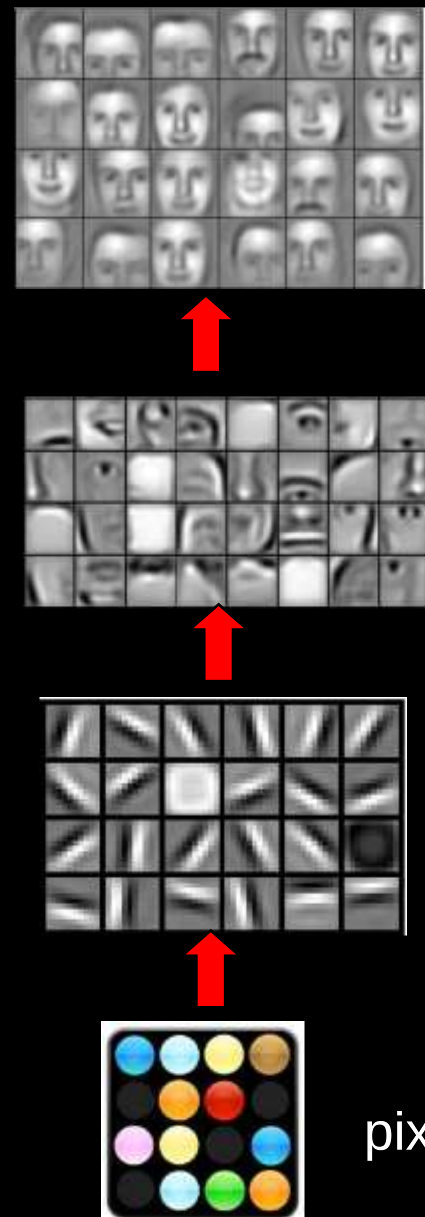


[Technical details: Sparse autoencoder or sparse version of Hinton's DBN.]

[Lee, Ranganath & Ng, 2007]

Hierarchical Sparse coding (Sparse DBN): Trained on face images

Training set: Aligned images of faces.



object models

object parts (combination of edges)

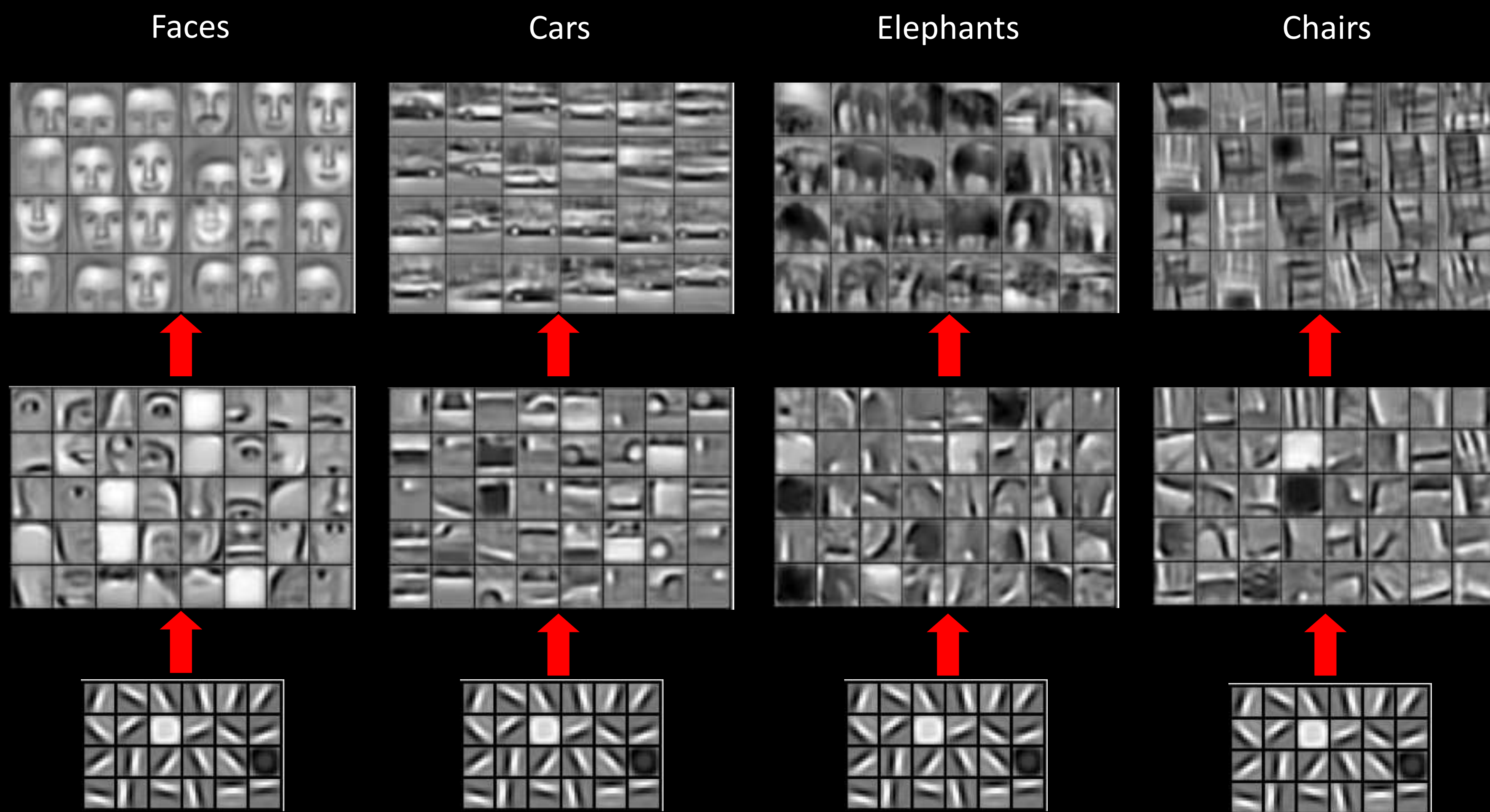
edges

pixels

[Honglak Lee]

Hierarchical Sparse coding (Sparse DBN)

Features learned from training on different object classes.



[Honglak Lee]

Machine learning applications

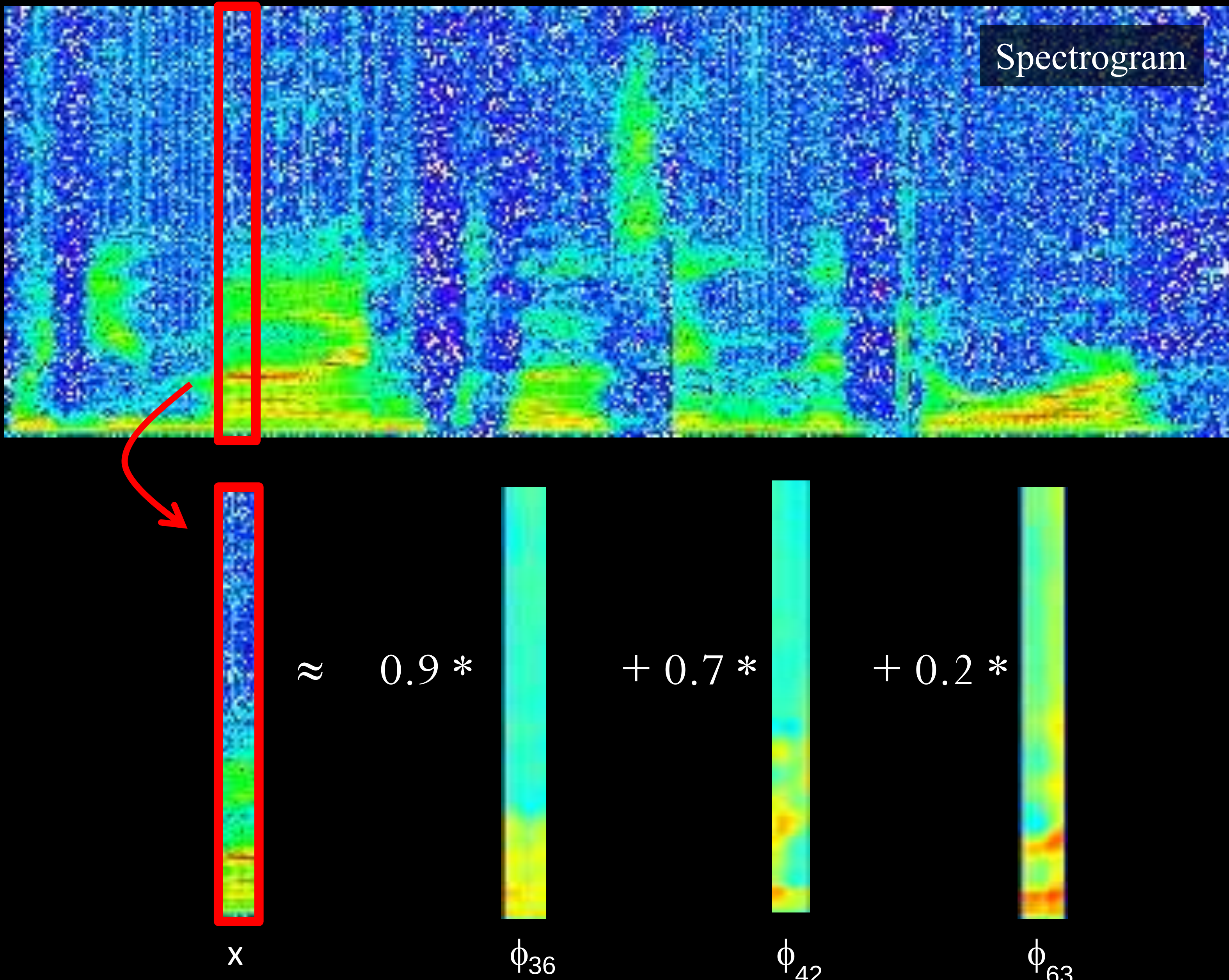
Video Activity recognition (Hollywood 2 benchmark)

Method	Accuracy
Hessian + ESURF [Williems et al 2008]	38%
Harris3D + HOG/HOF [Laptev et al 2003, 2004]	45%
Cuboids + HOG/HOF [Dollar et al 2005, Laptev 2004]	46%
Hessian + HOG/HOF [Laptev 2004, Williems et al 2008]	46%
Dense + HOG / HOF [Laptev 2004]	47%
Cuboids + HOG3D [Klaser 2008, Dollar et al 2005]	46%
Unsupervised feature learning (our method)	**52%**

Unsupervised feature learning significantly improves
on the previous state-of-the-art.

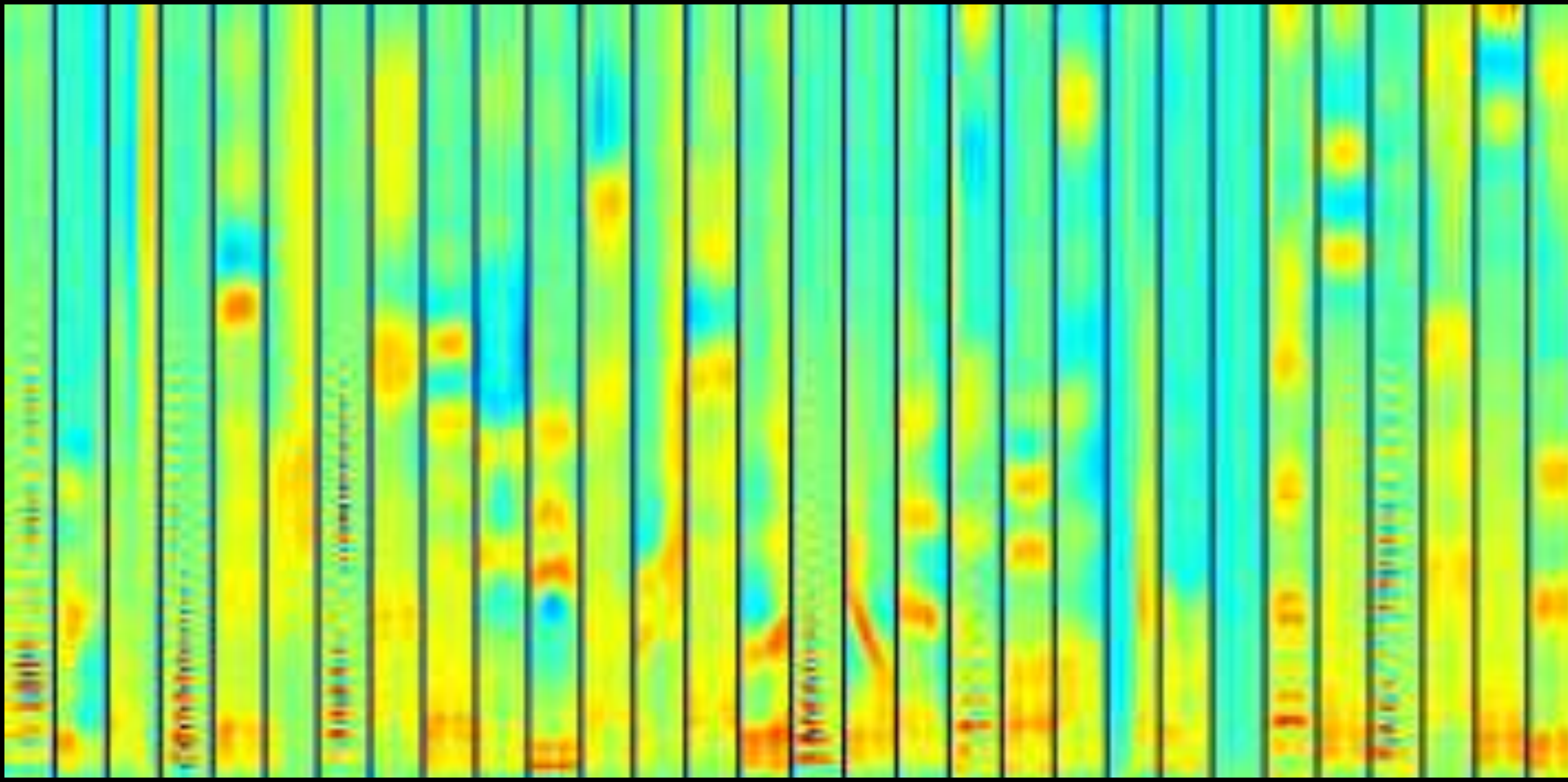
[Le, Zhou & Ng, 2011]

Sparse coding on audio (speech)



Andrew Ng

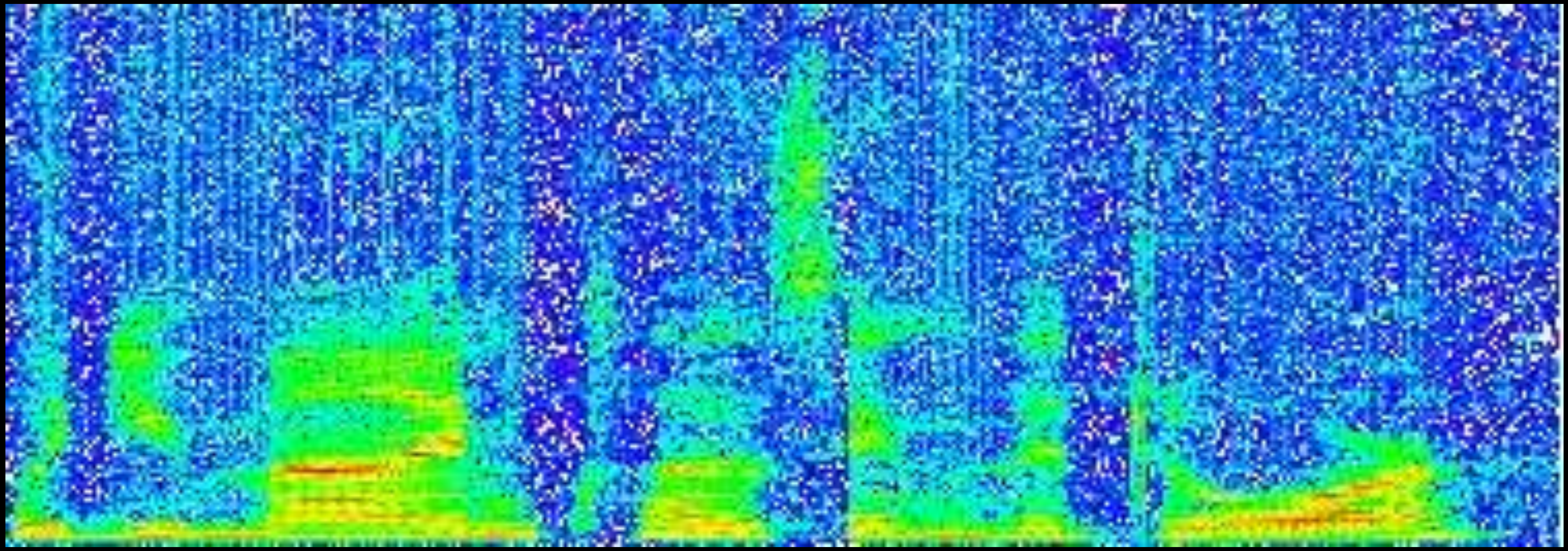
Dictionary of bases ϕ_i learned for speech



Many bases seem to correspond to phonemes.

[Honglak Lee]

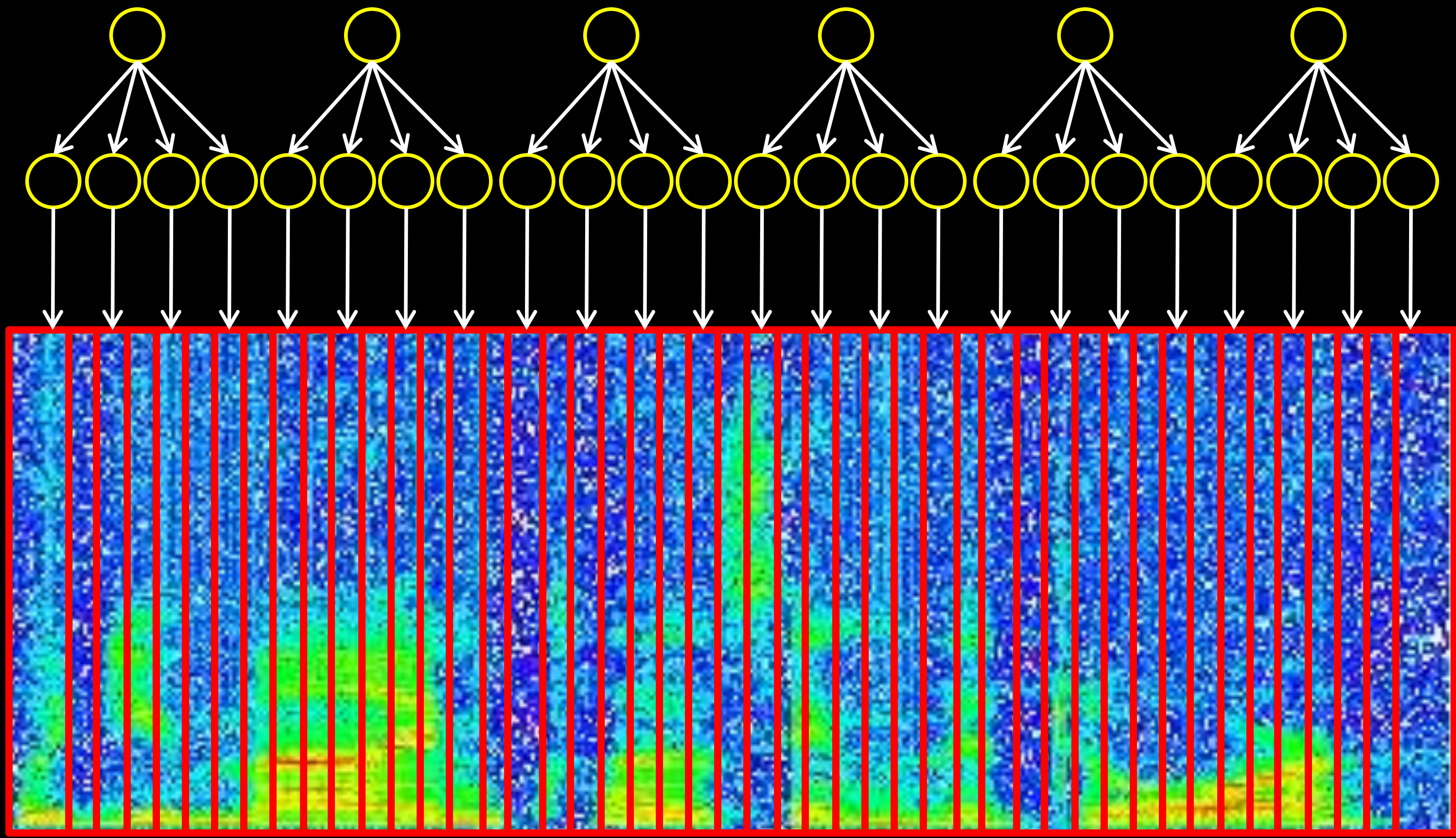
Hierarchical Sparse coding (sparse DBN) for audio



Spectrogram

[Honglak Lee]

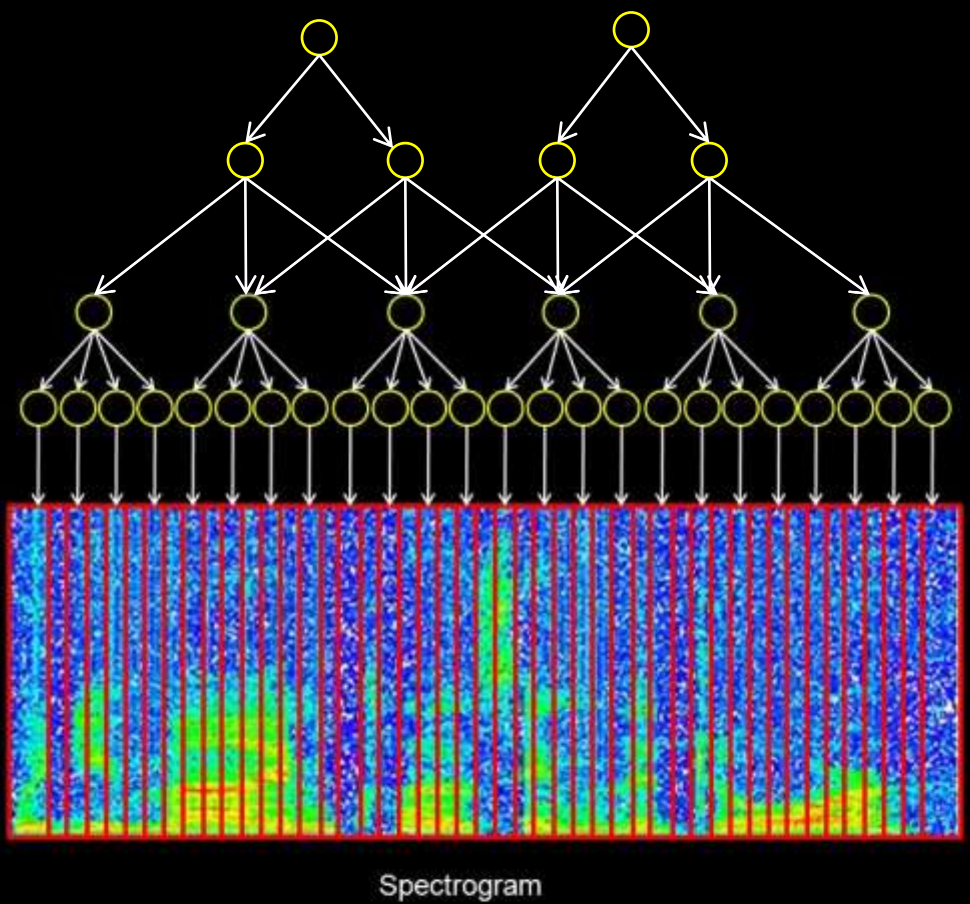
Hierarchical Sparse coding (sparse DBN) for audio



Spectrogram

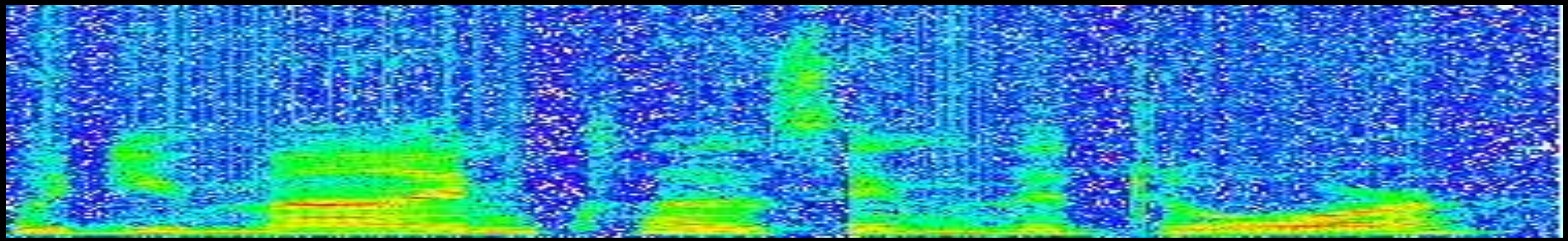
[Honglak Lee]

Hierarchical Sparse coding (sparse DBN) for audio



[Honglak Lee]

Phoneme Classification (TIMIT benchmark)



Method	Accuracy
Clarkson and Moreno (1999)	77.6%
Gunawardana et al. (2005)	78.3%
Sung et al. (2007)	78.5%
Petrov et al. (2007)	78.6%
Sha and Saul (2006)	78.9%
Yu et al. (2006)	79.2%
Unsupervised feature learning (our method)	**80.3%**

Unsupervised feature learning significantly improves on the previous state-of-the-art.

[Lee et al., 2009]

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)

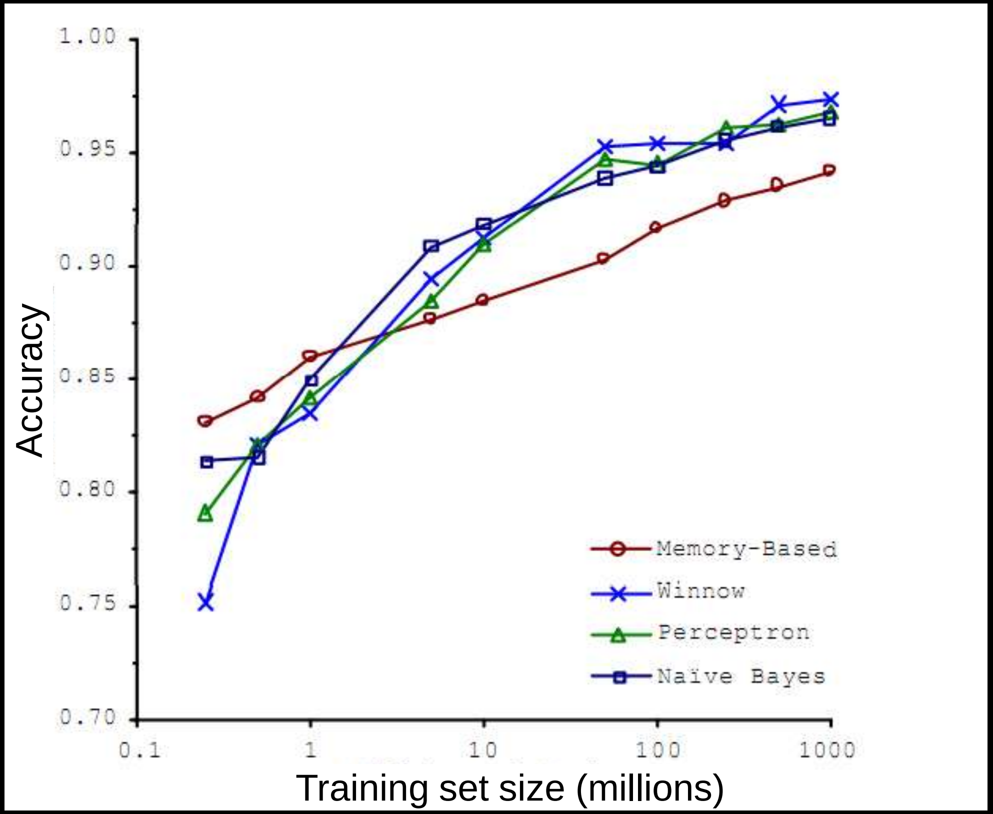
AVLetters Lip reading	Accuracy
Prior art (Zhao et al., 2009)	58.9%
Stanford Feature learning	**65.8%**

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

Technical challenge: Scaling up

Supervised Learning

- Choices of learning algorithm:
  - Memory based
  - Winnow
  - Perceptron
  - Naïve Bayes
  - SVM
  - ….
- What matters the most?

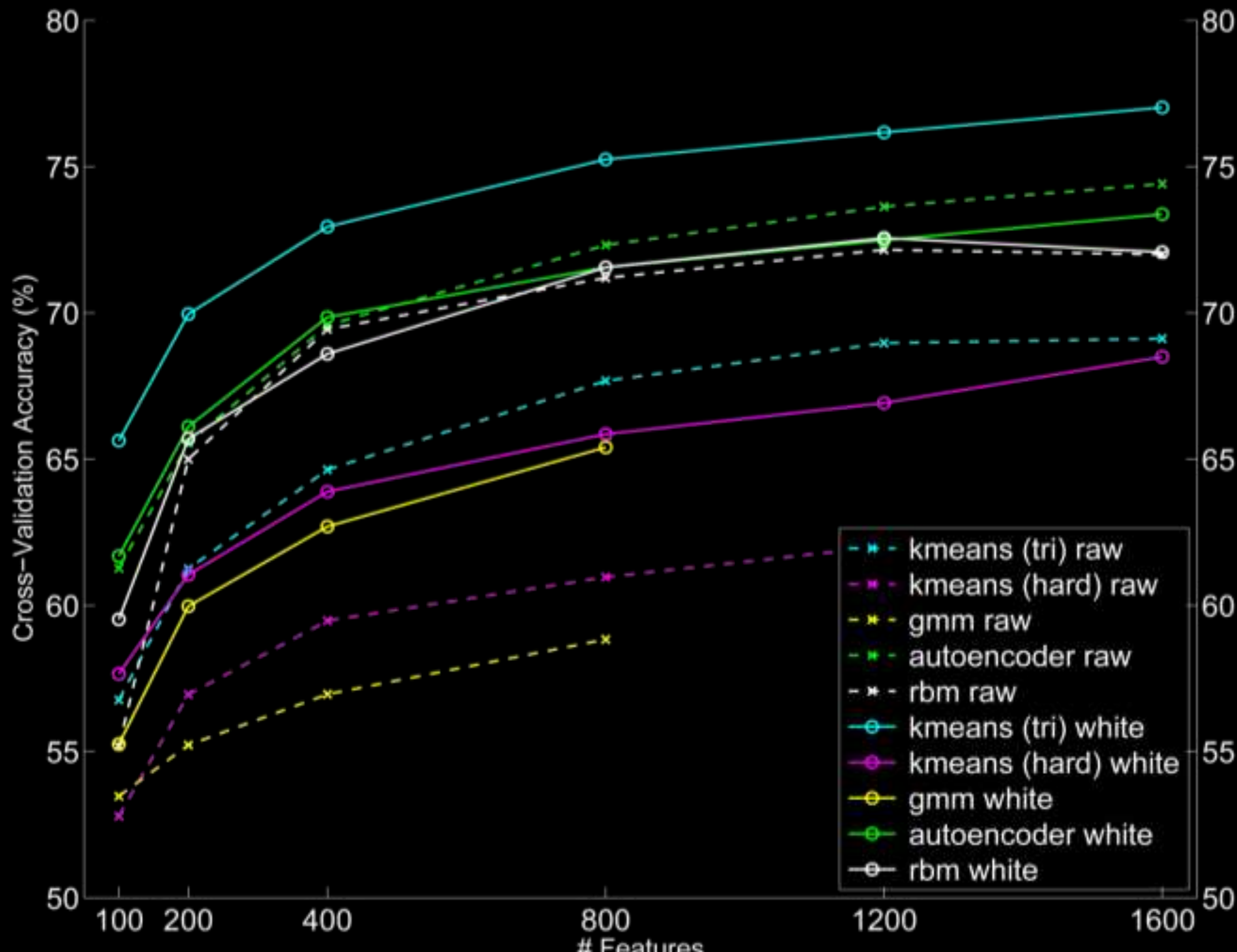


[Banko & Brill, 2001]

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

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]

Attempts to scale up

Significant effort spent on algorithmic tricks to get algorithms to run faster.

- Efficient sparse coding. [LeCun, Ng, Yu]
- Efficient posterior inference [Bengio, Hinton]
- Convolutional Networks. [Bengio, de Freitas, LeCun, Lee, Ng]
- Tiled Networks. [Hinton, Ng]
- Randomized/fast parameter search. [DiCarlo, Ng]
- Massive data synthesis. [LeCun, Schmidhuber]
- Massive embedding models [Bengio, Collobert, Hinton, Weston]
- Fast decoder algorithms. [LeCun, Lee, Ng, Yu]
- GPU, FPGA and ASIC implementations. [Dean, LeCun, Ng, Olukotun]

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%**
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%**

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)

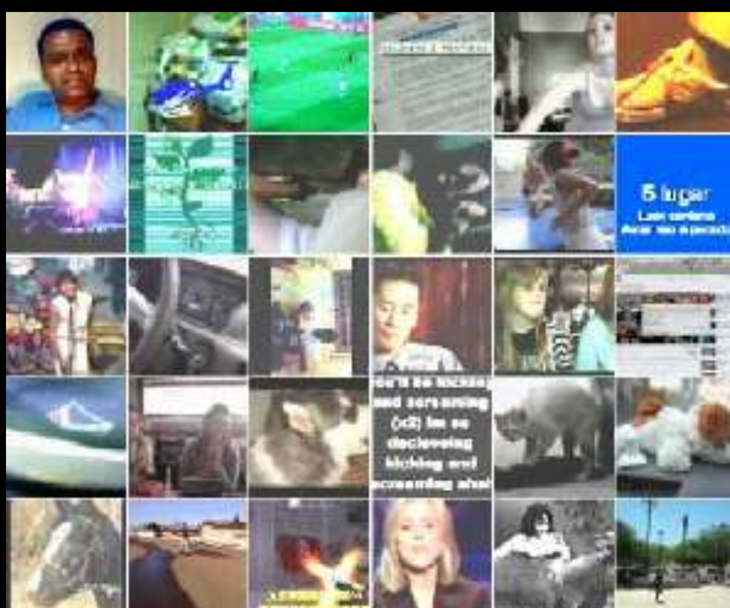
Scaling up: Discovering object classes

[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)

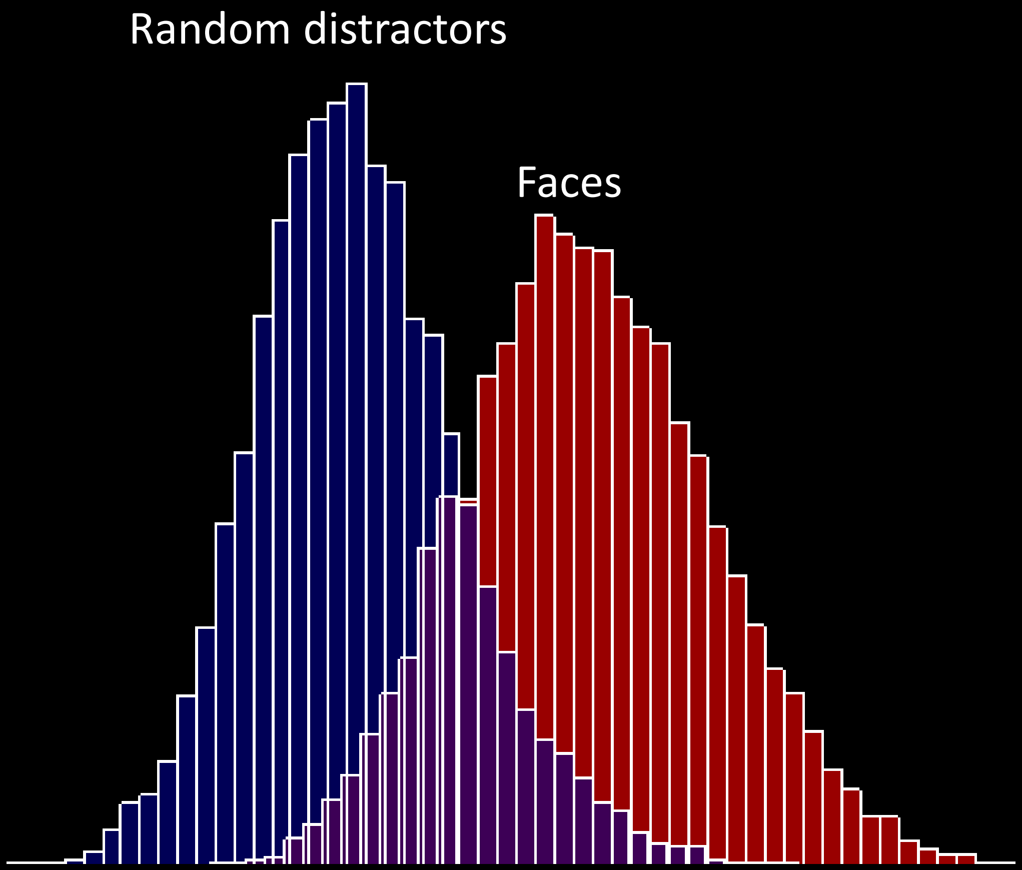
Test set (FITW + ImageNet)

Andrew Ng

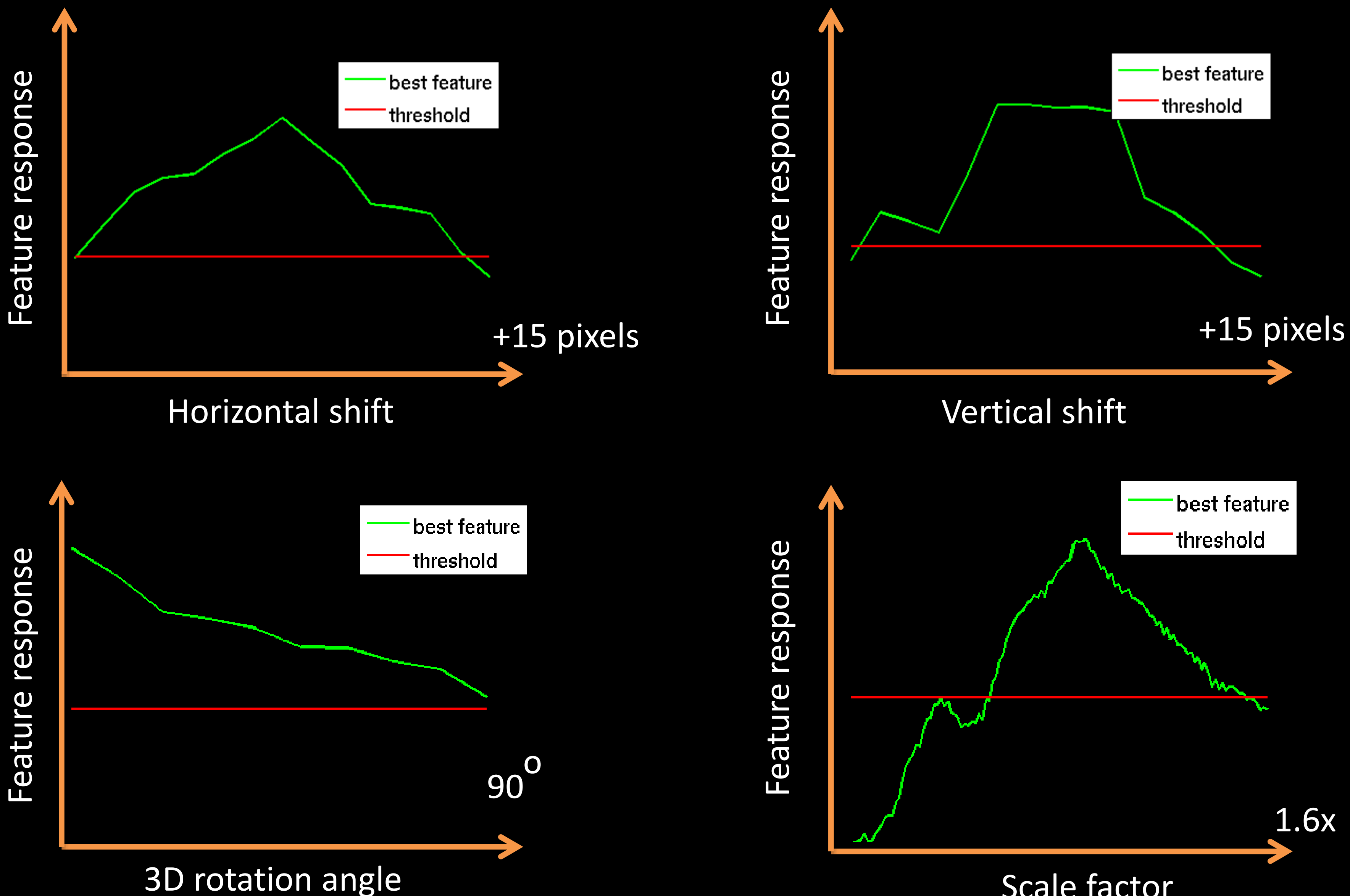
Face neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization


Random distractors
Faces

Invariance properties

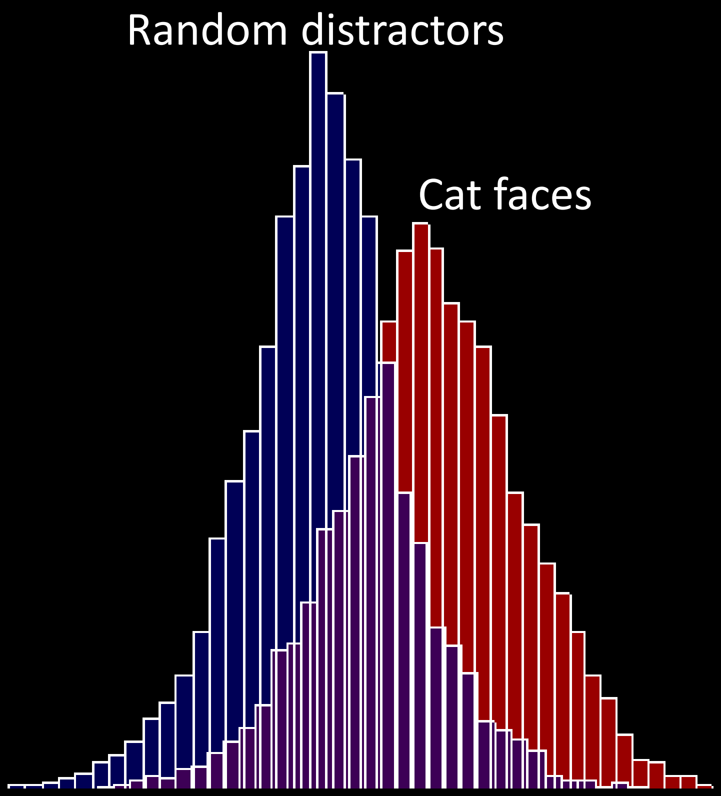


Cat neuron

Top Stimuli from the test set

Optimal stimulus by numerical optimization

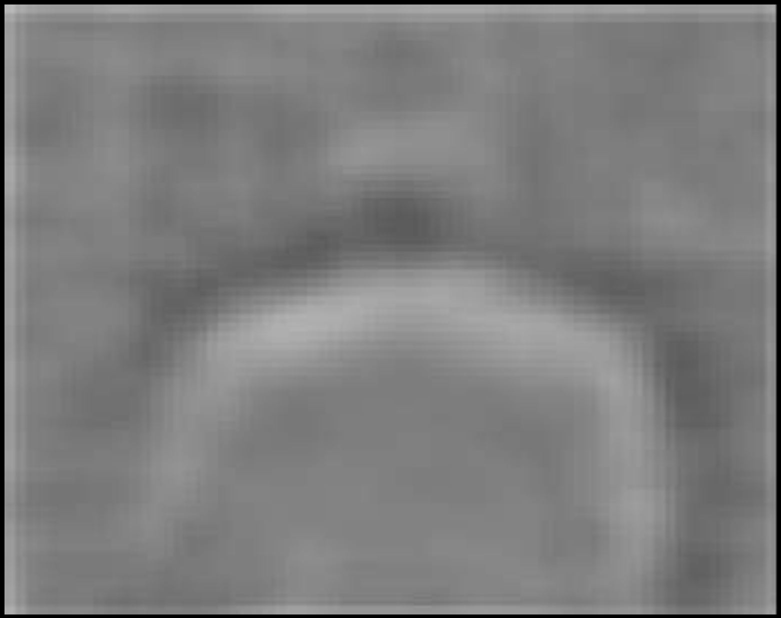
Cat face neuron



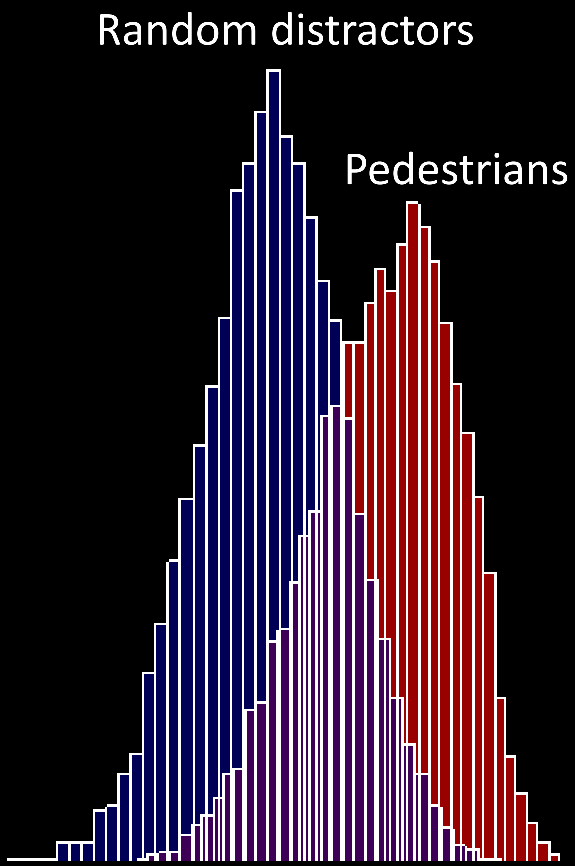
Visualization

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!

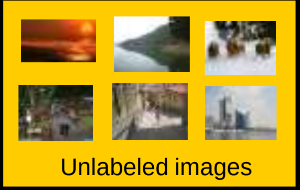
- We don't understand the learned features.

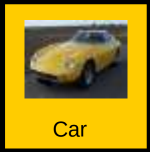
  A: True. Though many vision/audio/etc. features also suffer from this (e.g, concatenations/combinations of different features).

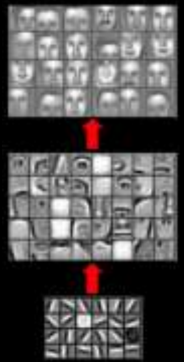
Conclusion

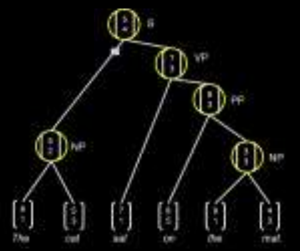
Unsupervised Feature Learning Summary

- Deep Learning and Self-Taught 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.
- To get this to work for yourself, see online tutorial:
  http://deeplearning.stanford.edu/wiki









Thanks to:

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 Rajat Monga Marc'Aurelio Ranzato Paul Tucker Kay Le

Advanced topics + Research philosophy

Andrew Ng

Stanford University & Google

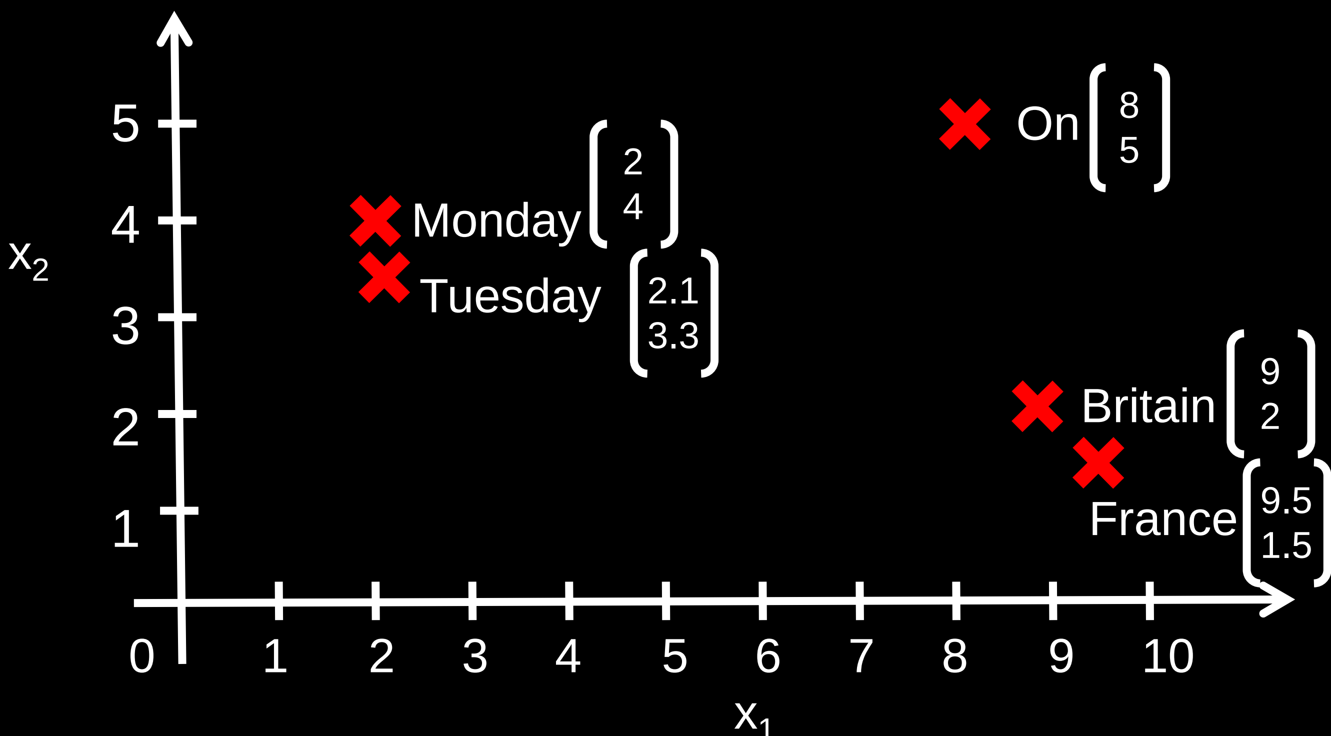
Learning Recursive Representations

Feature representations of words

Imagine taking each word, and computing an n-dimensional feature vector for it.

[Distributional representations, or Bengio et al., 2003, Collobert & Weston, 2008.]

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



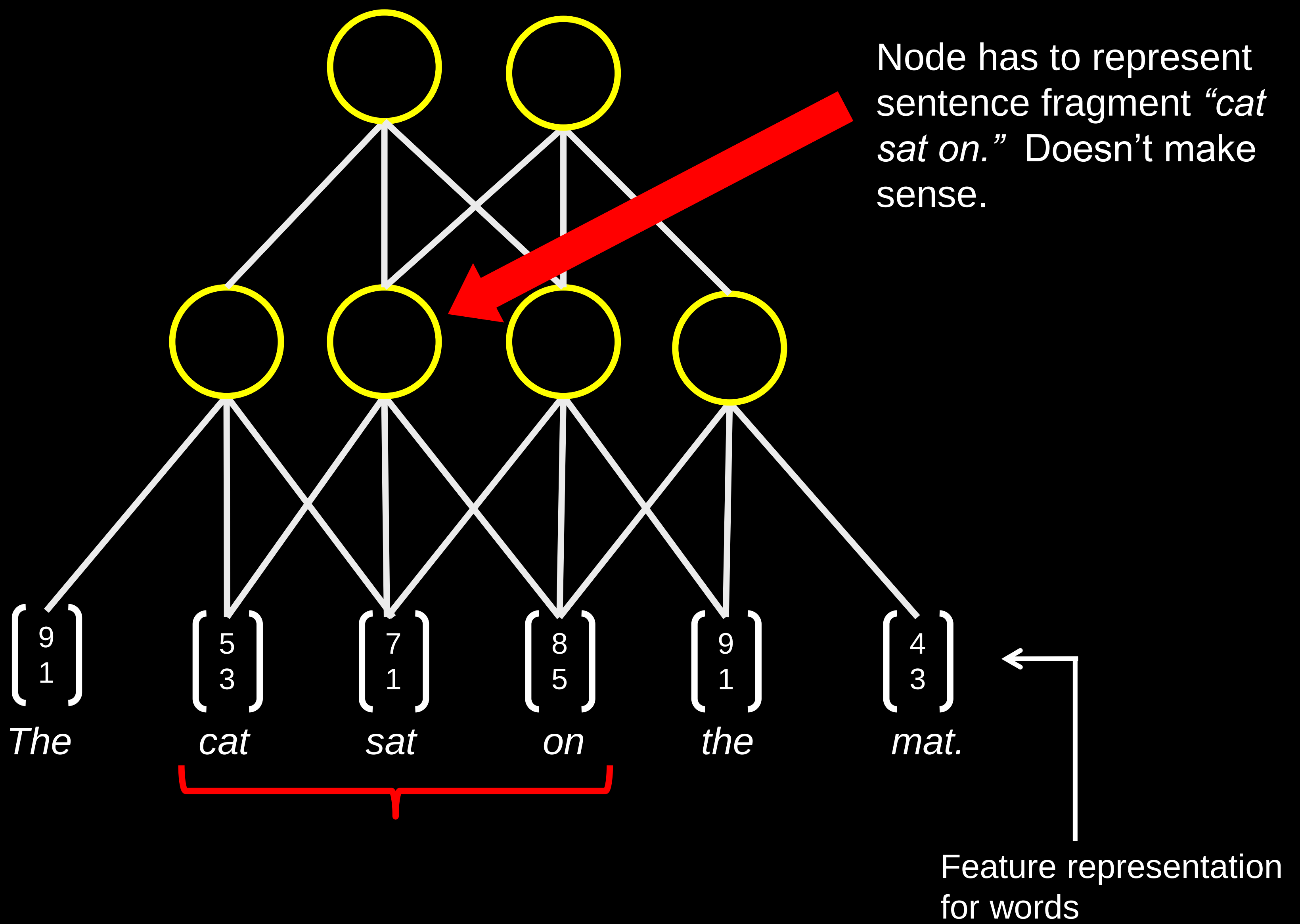
$$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} \qquad \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

Monday Britain

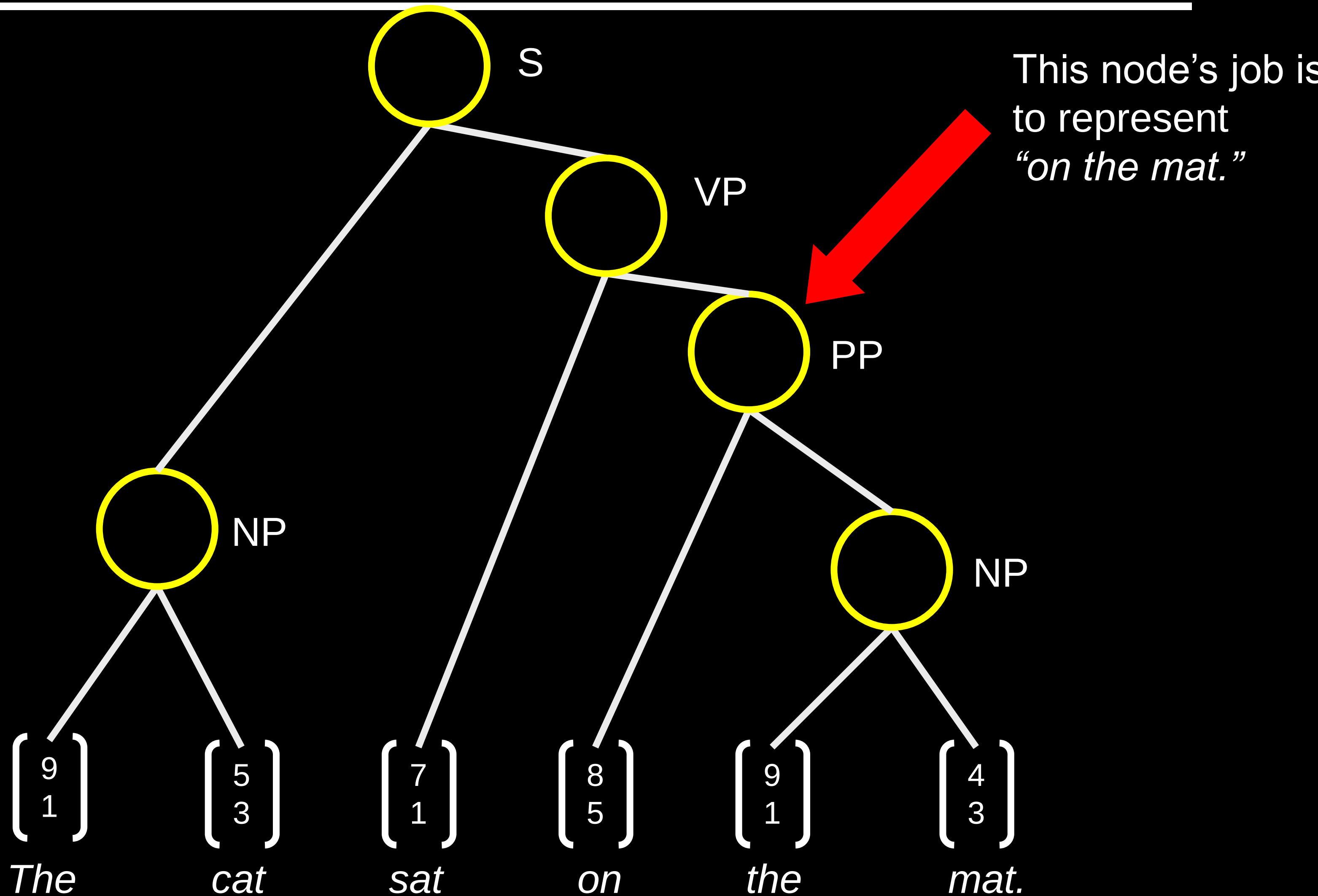
On Monday, Britain ….

Representation: $\begin{bmatrix}8\\5\end{bmatrix} \quad \begin{bmatrix}2\\4\end{bmatrix} \quad \begin{bmatrix}9\\2\end{bmatrix}$

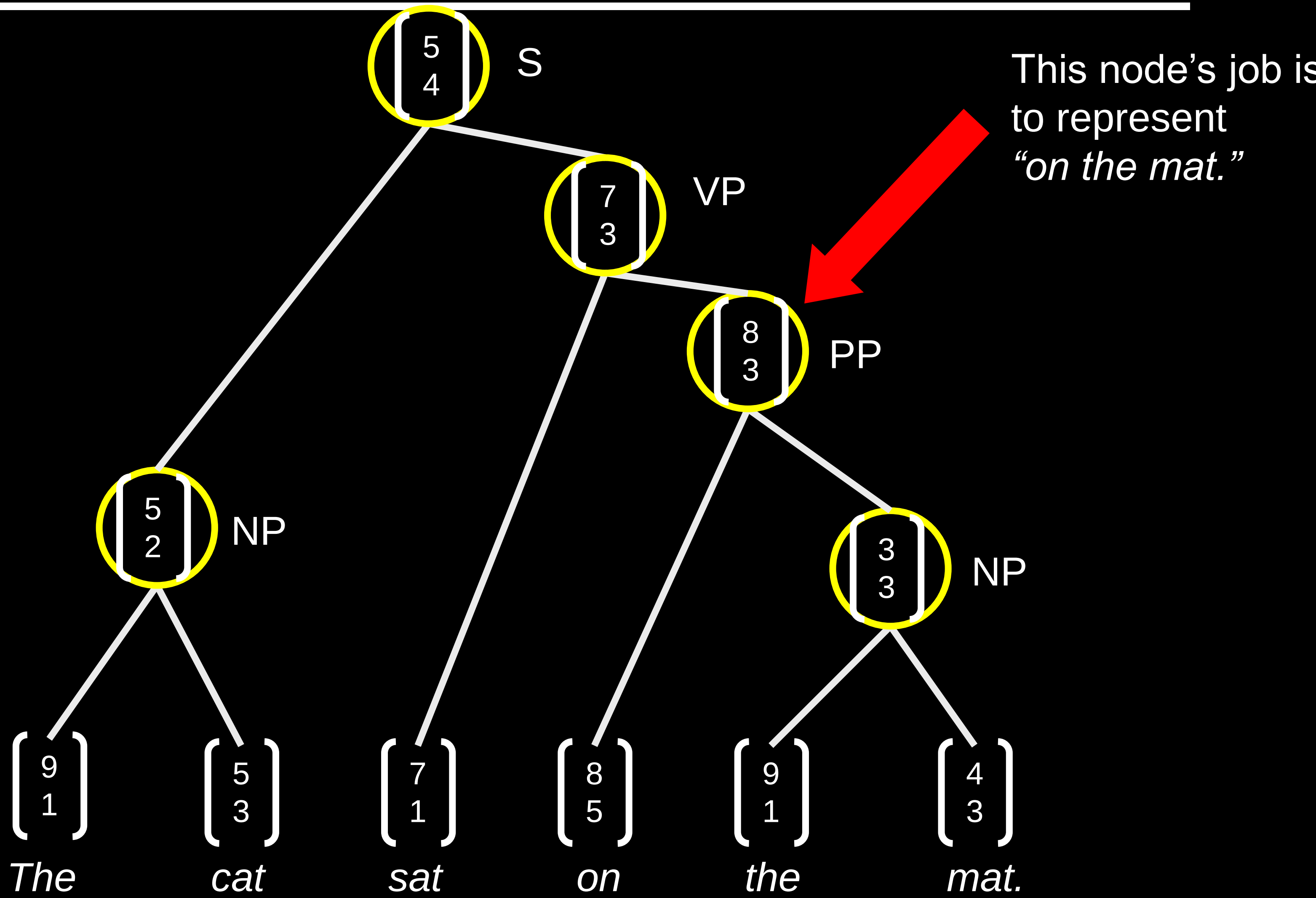
“Generic” hierarchy on text doesn’t make sense



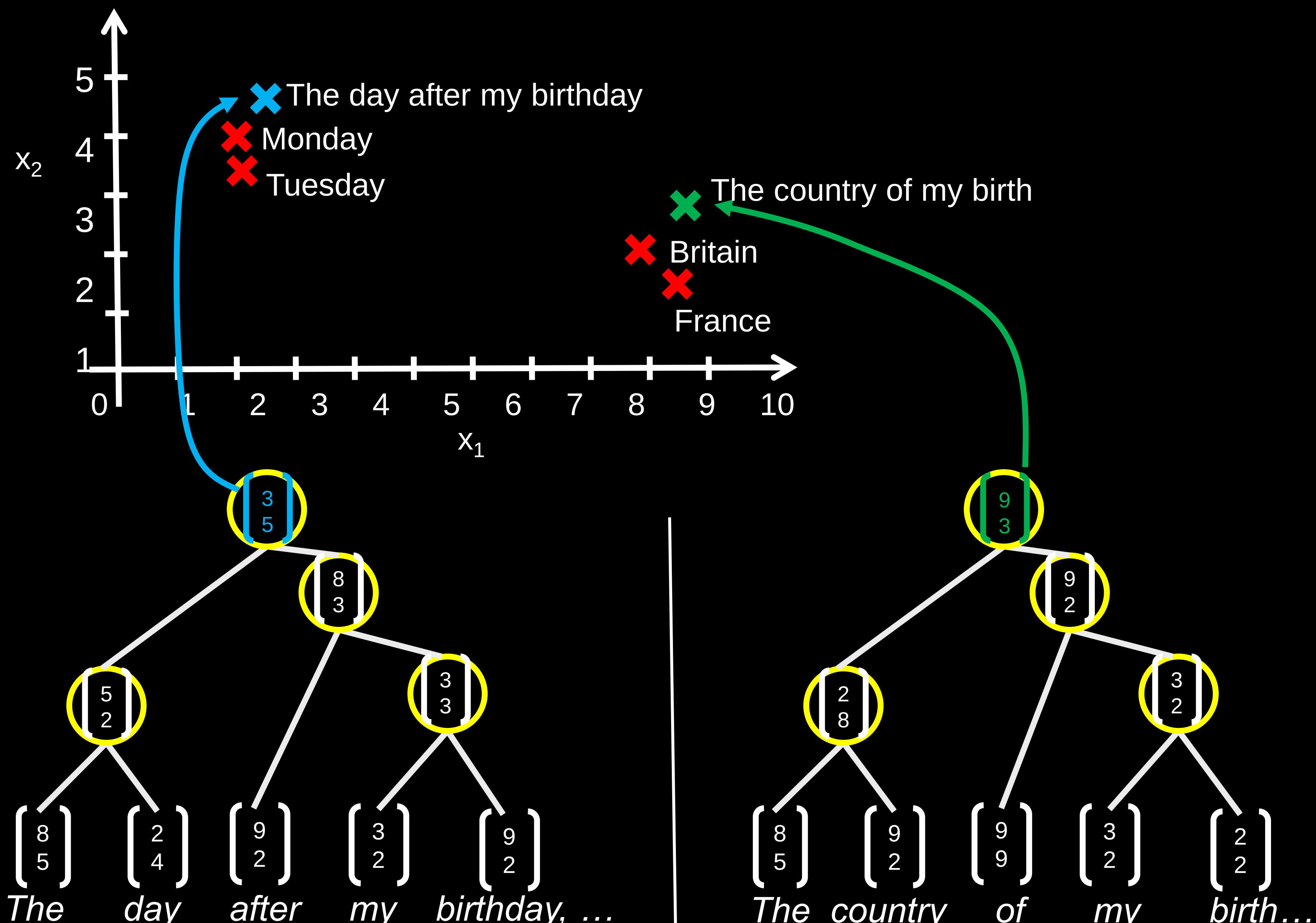
What we want (illustration)



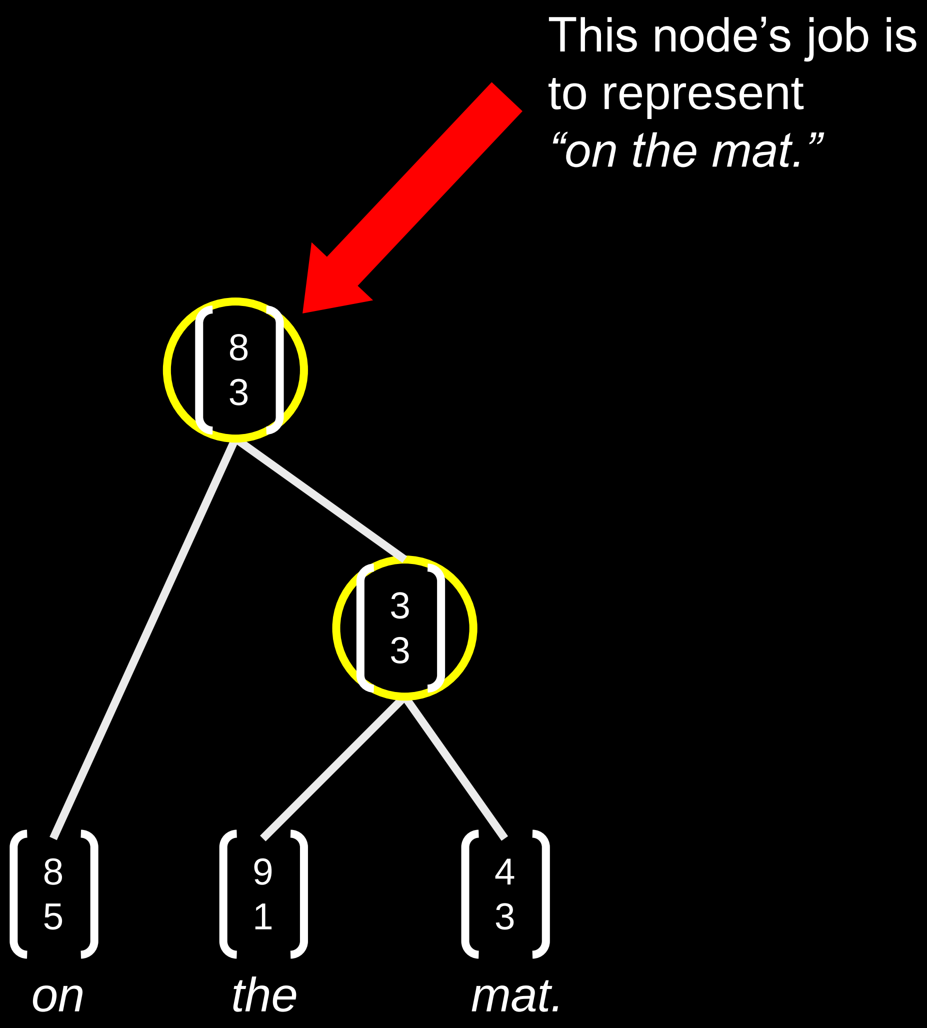
What we want (illustration)



What we want (illustration)

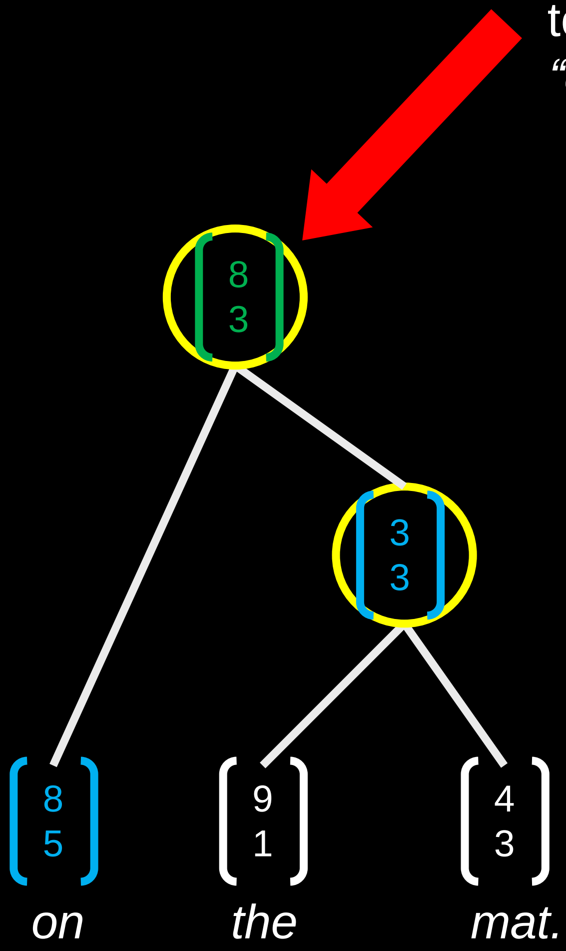


Learning recursive representations



Learning recursive representations

This node's job is to represent *"on the mat."*



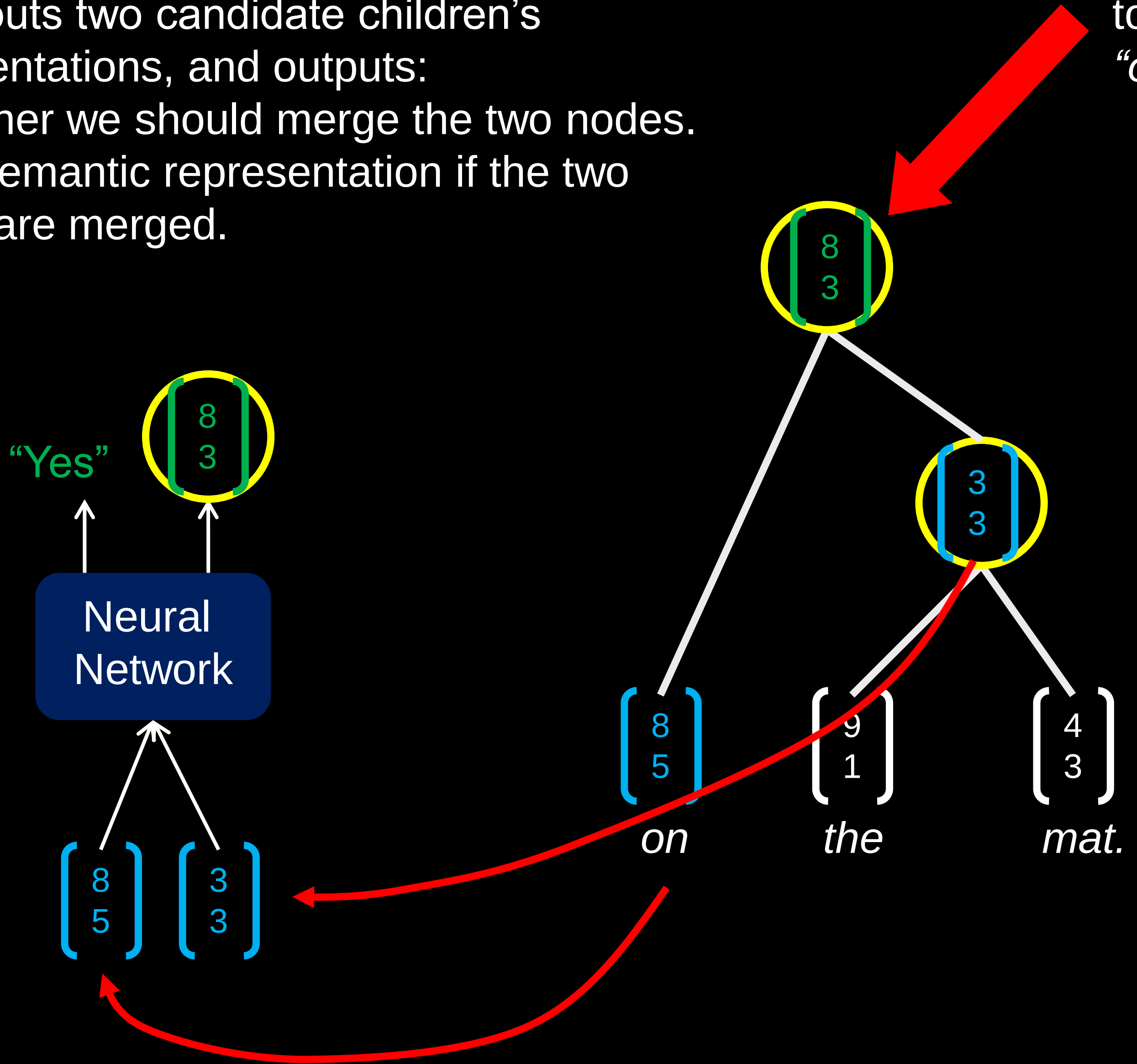
Andrew Ng

Learning recursive representations

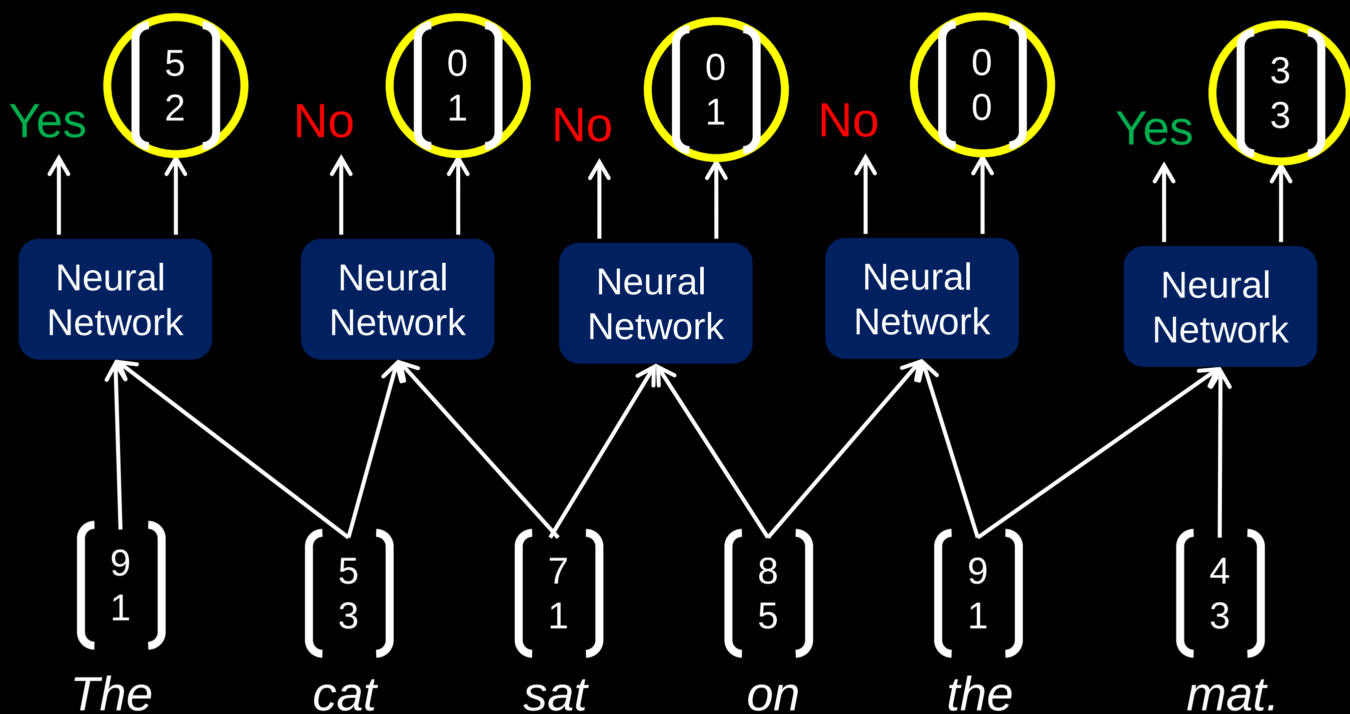
Basic computational unit: Neural Network that inputs two candidate children's representations, and outputs:

- Whether we should merge the two nodes.
- The semantic representation if the two nodes are merged.

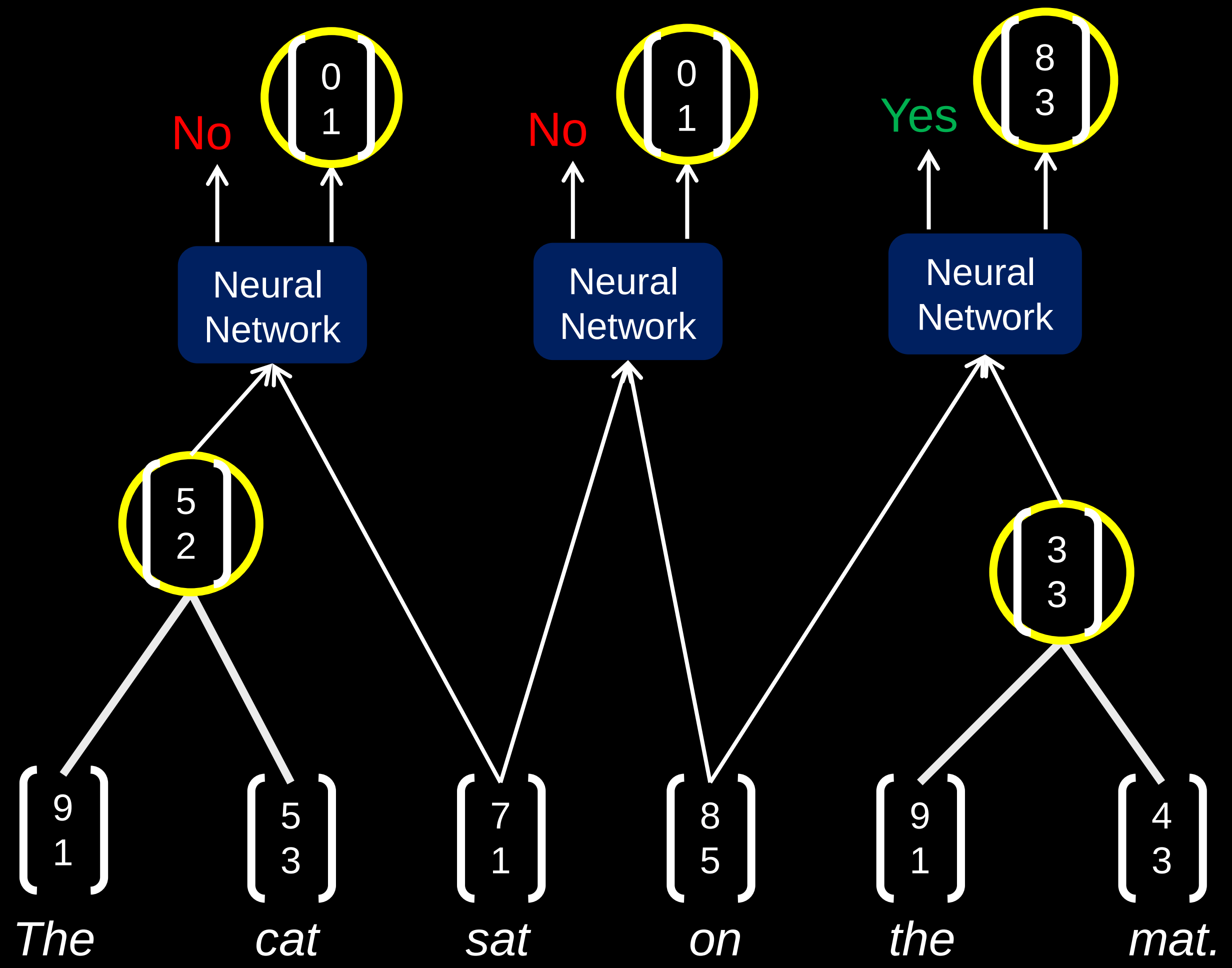
This node's job is to represent *"on the mat."*



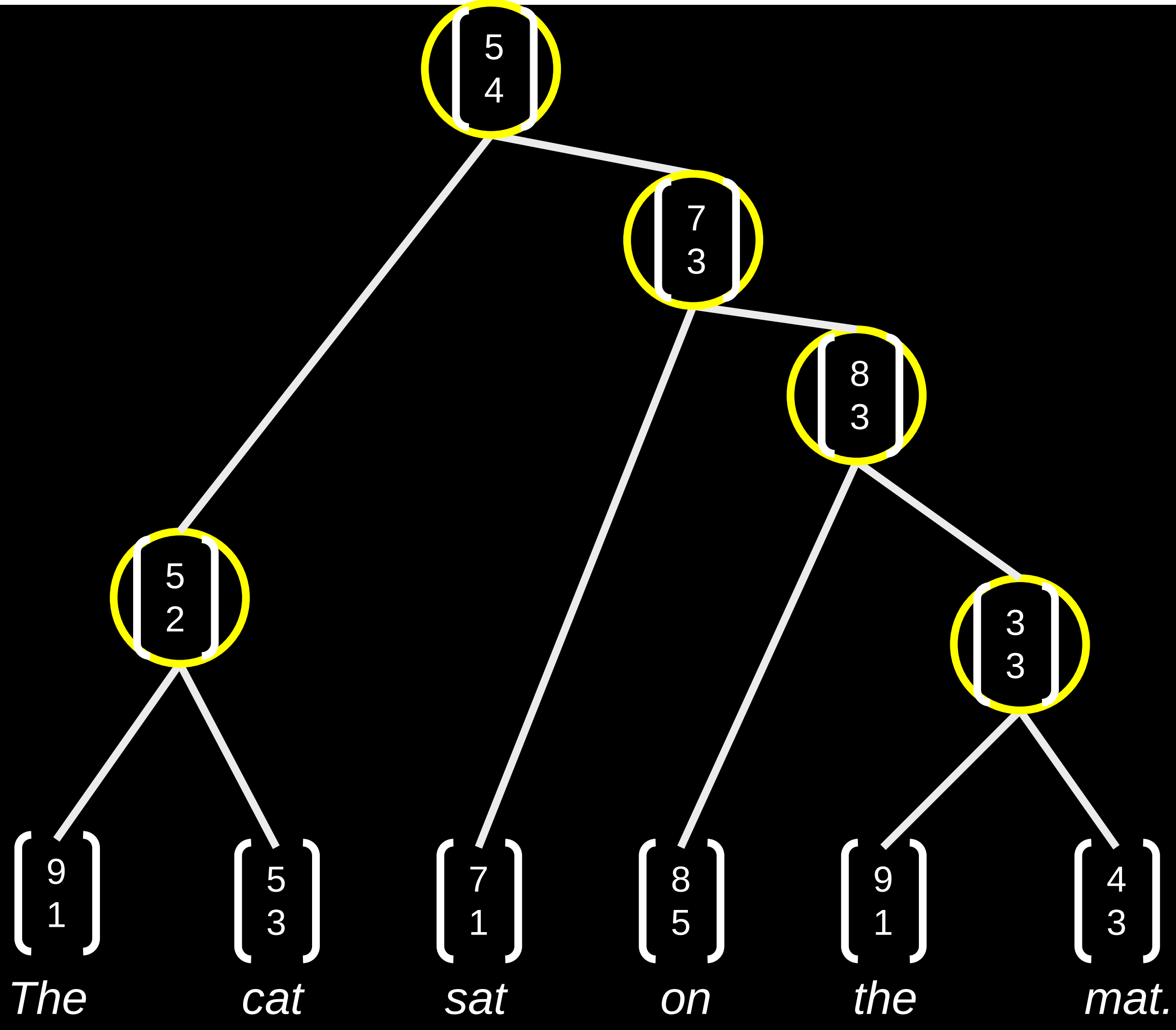
Parsing a sentence



Parsing a sentence



Parsing a sentence



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.)

Similarities	Center Sentence	Nearest Neighbor Sentences (most similar feature vector)
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 which are cities	Columbia , S.C	1. Greenville , Miss 2. UNK , Md

Finding Similar Sentences

Similarities	Center Sentence	Nearest Neighbor Sentences (most similar feature vector)
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

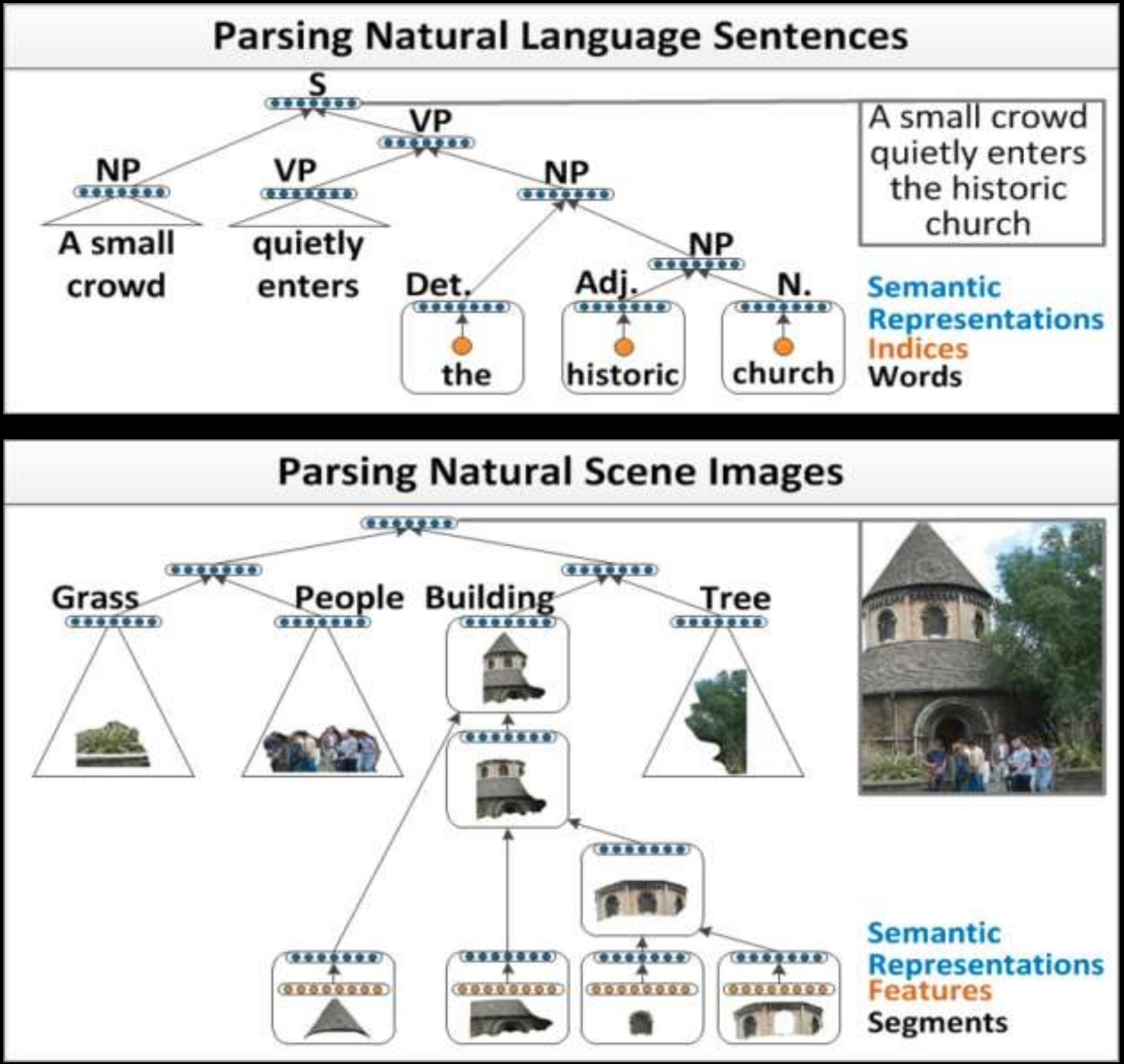
Application: Paraphrase Detection

- Task: Decide whether or not two sentences are paraphrases of each other. (MSR Paraphrase Corpus)

Method	F1
Baseline	79.9
Rus et al., (2008)	80.5
Mihalcea et al., (2006)	81.3
Islam et al. (2007)	81.3
Qiu et al. (2006)	81.6
Fernando & Stevenson (2008) (WordNet based features)	82.4
Das et al. (2009)	82.7
Wan et al (2006) (many features: POS, parsing, BLEU, etc.)	83.0
Stanford Feature Learning	**83.4**

Parsing sentences and parsing images

A small crowd quietly enters the historic church.



Each node in the hierarchy has a “feature vector” representation.

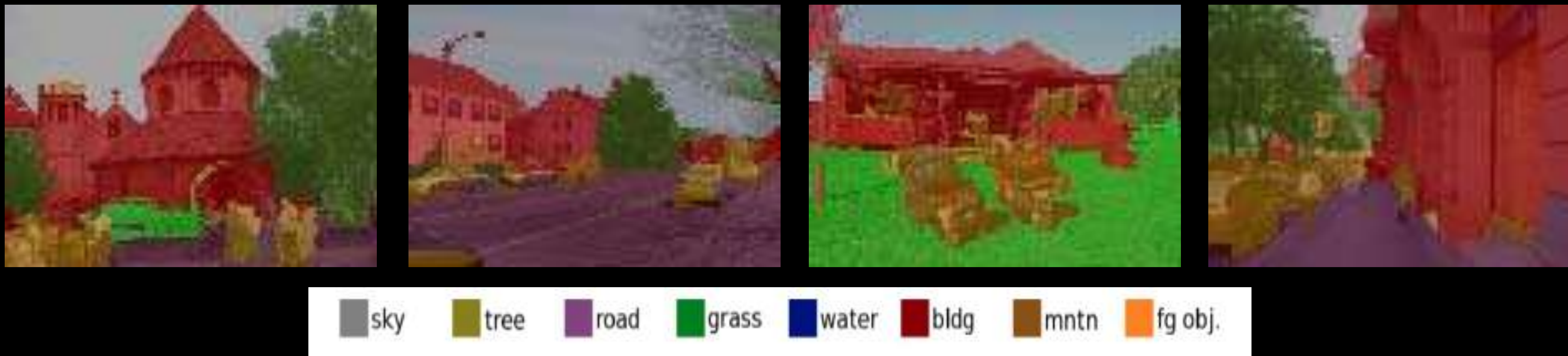
Nearest neighbor examples for image patches

- 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?

Selected patch

Nearest Neighbors

Multi-class segmentation (Stanford background dataset)



Method	Accuracy
Pixel CRF (Gould et al., ICCV 2009)	74.3
Classifier on superpixel features	75.9
Region-based energy (Gould et al., ICCV 2009)	76.4
Local labelling (Tighe & Lazebnik, ECCV 2010)	76.9
Superpixel MRF (Tighe & Lazebnik, ECCV 2010)	77.5
Simultaneous MRF (Tighe & Lazebnik, ECCV 2010)	77.5
Stanford Feature learning (our method)	**78.1**

Multi-class Segmentation MSRC dataset: 21 Classes

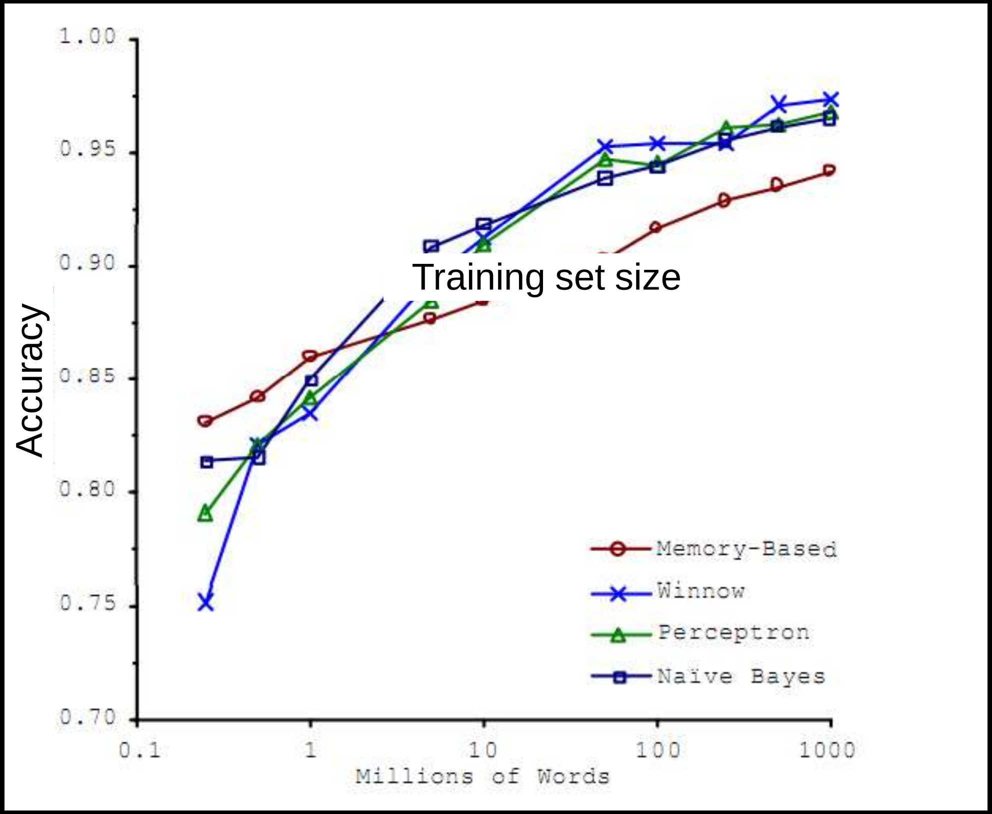
Methods	Accuracy
TextonBoost (Shotton et al., ECCV 2006)	72.2
Framework over mean-shift patches (Yang et al., CVPR 2007)	75.1
Pixel CRF (Gould et al., ICCV 2009)	75.3
Region-based energy (Gould et al., IJCV 2008)	76.5
Stanford Feature learning (out method)	**76.7**

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

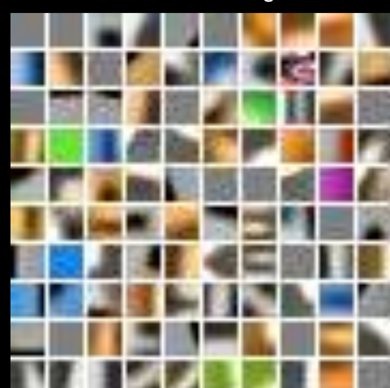
with whitening

without whitening

Sparse RBM

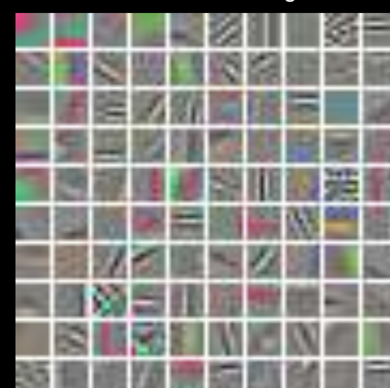
with whitening

without whitening



K-means

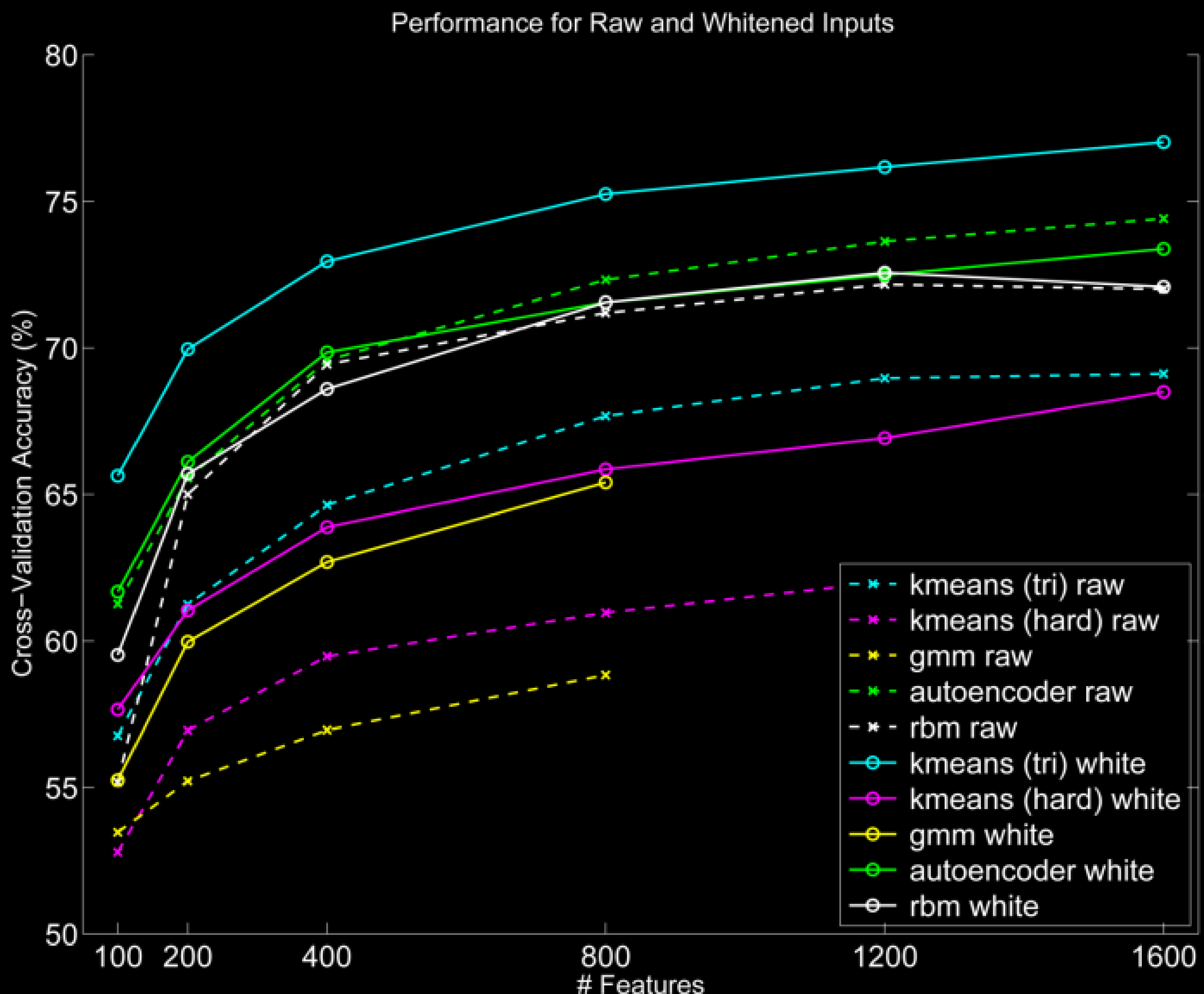
with whitening



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

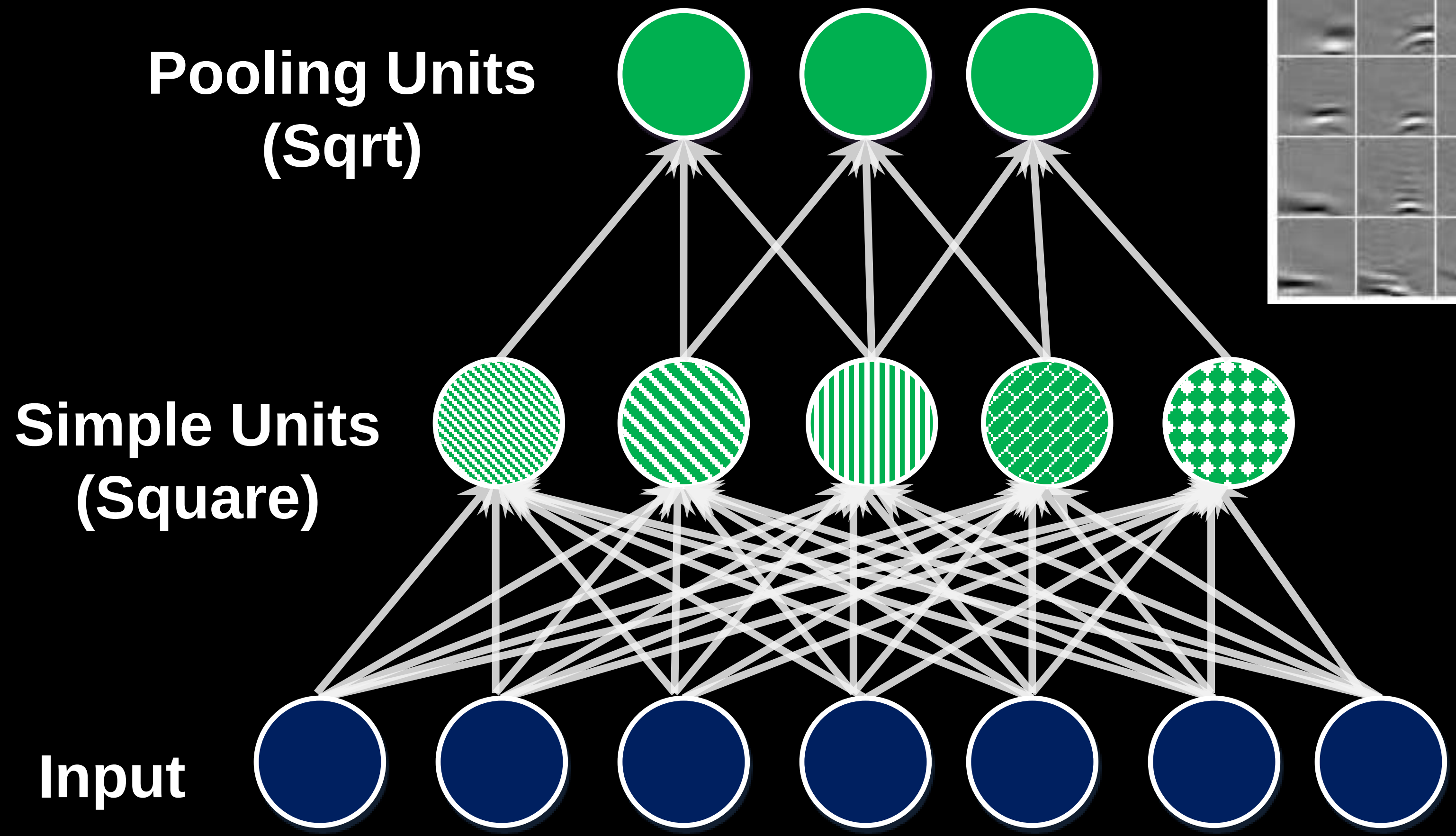
Learning Invariances

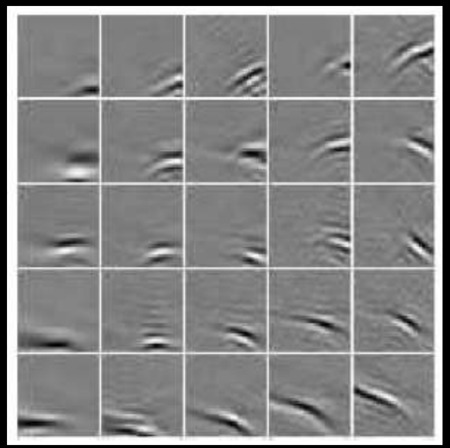
- We want to learn invariant features.

- 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.

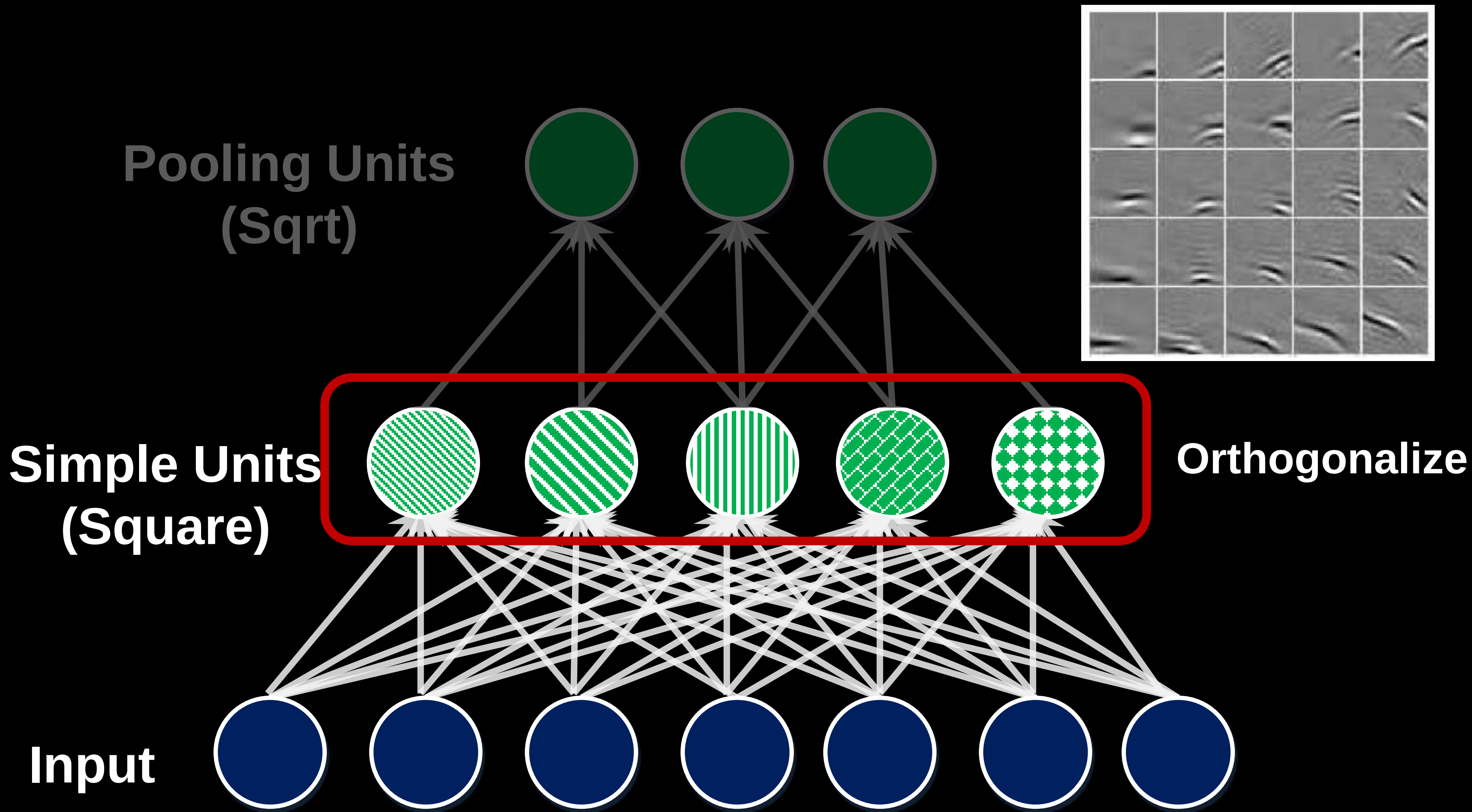
Fully Connected Topographic ICA





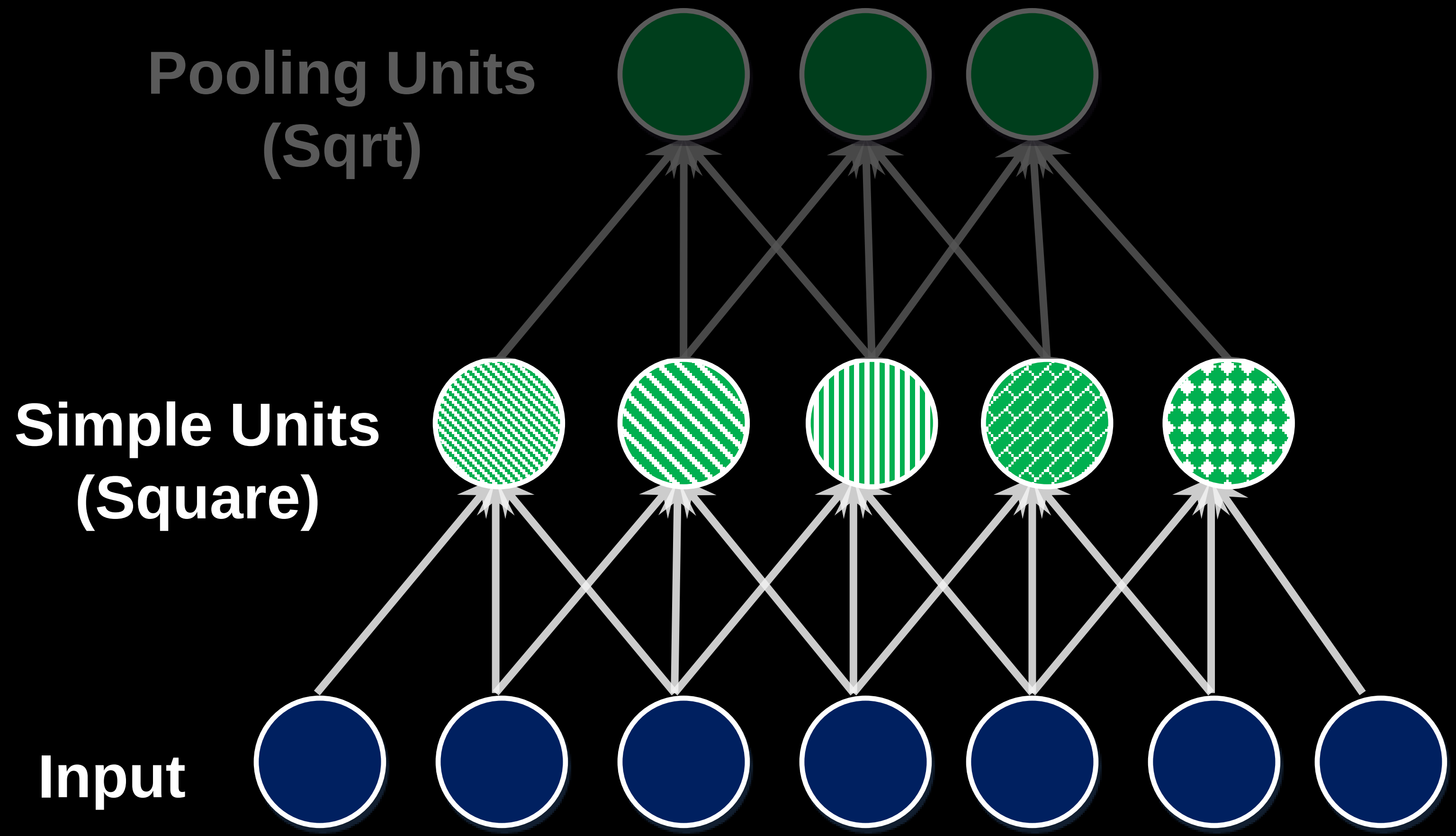
Doesn’t scale to large images.

Fully Connected Topographic ICA



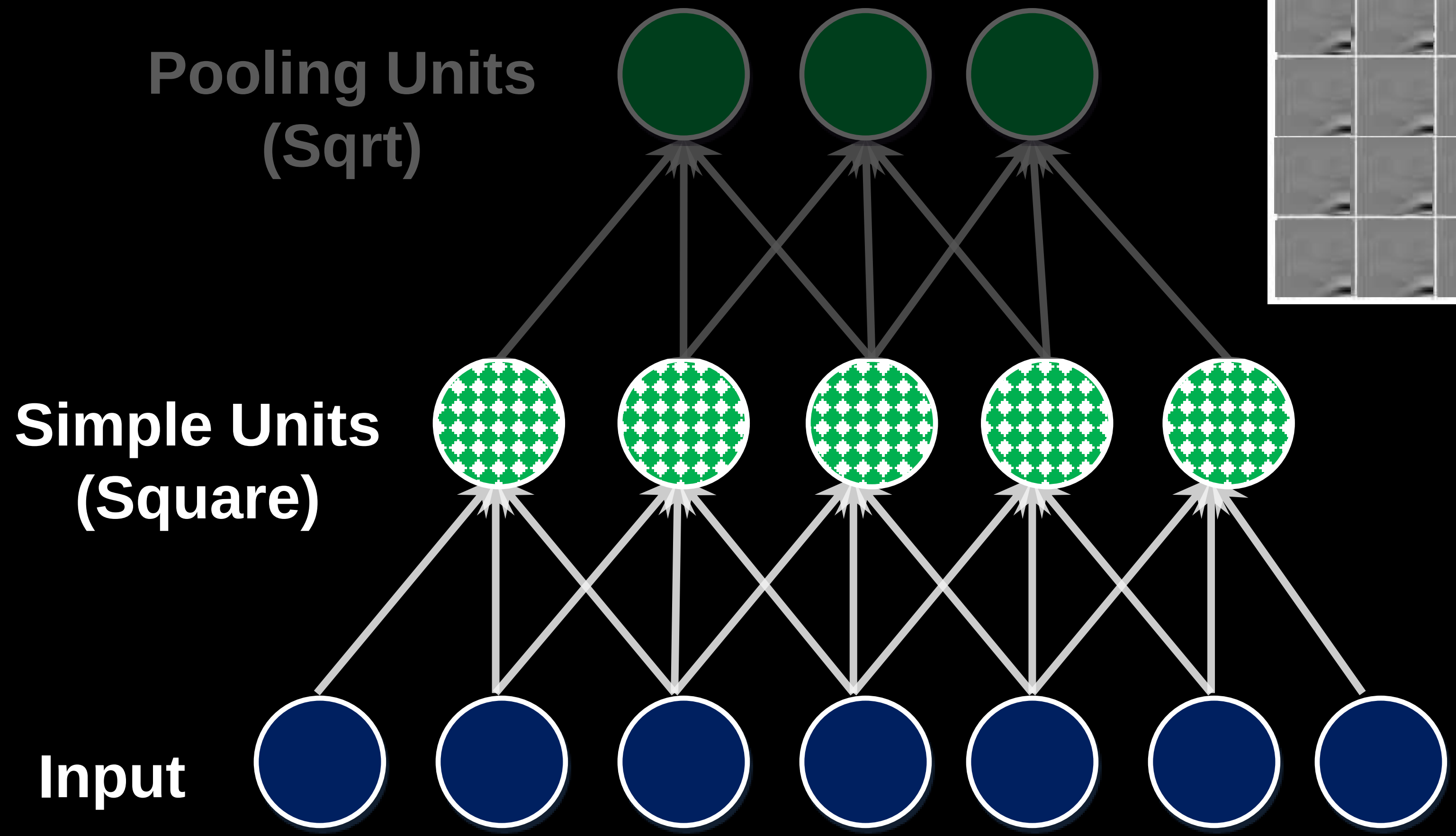
Doesn’t scale to large images.

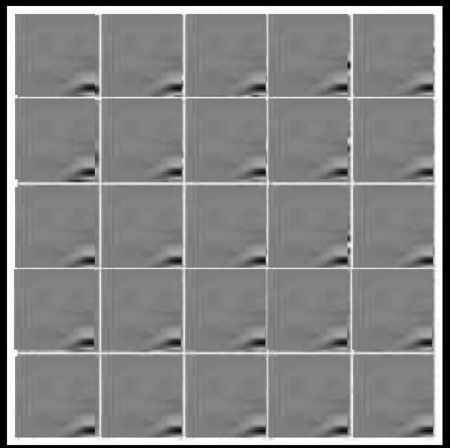
Local Receptive Fields



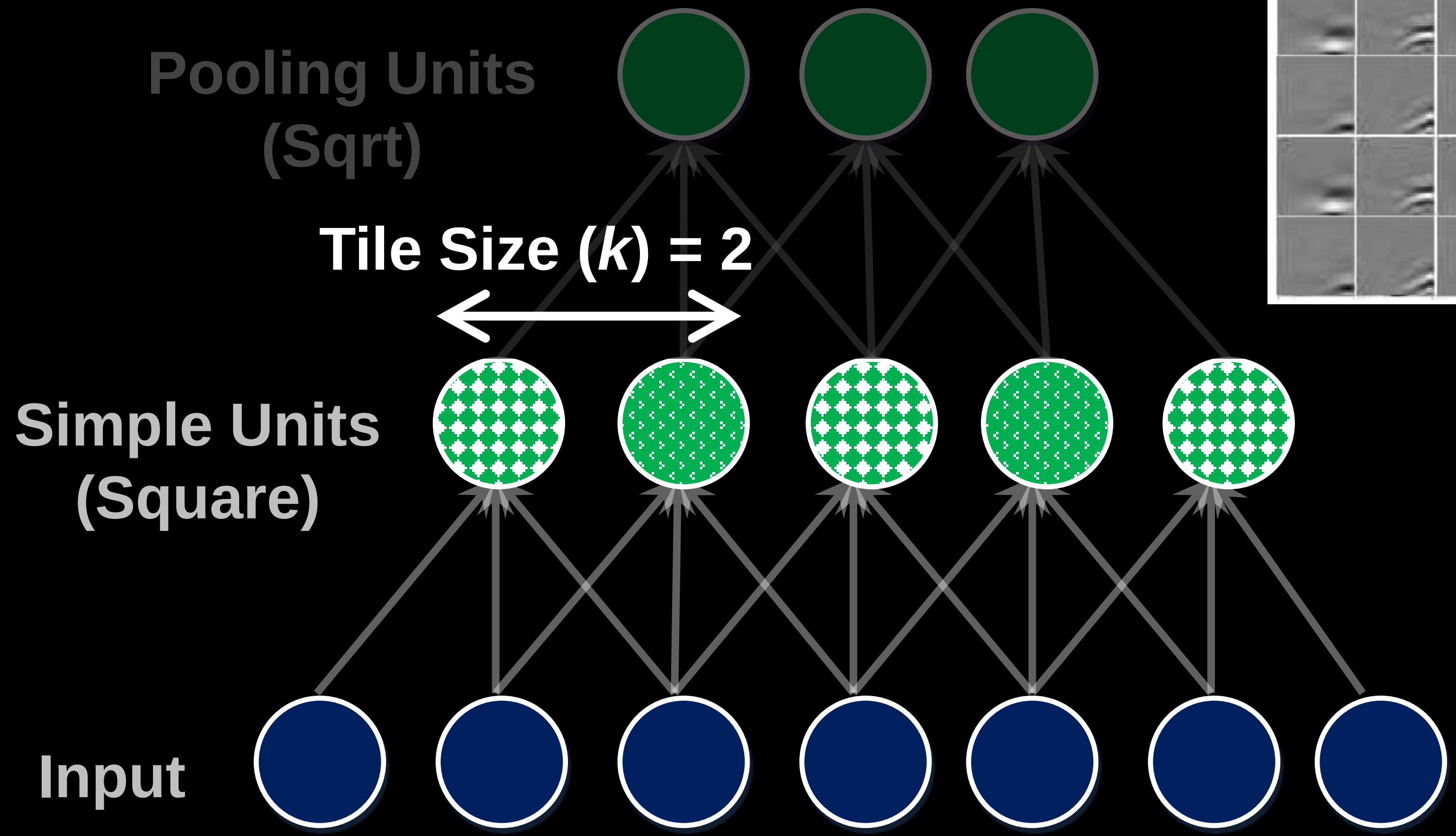
Andrew Ng

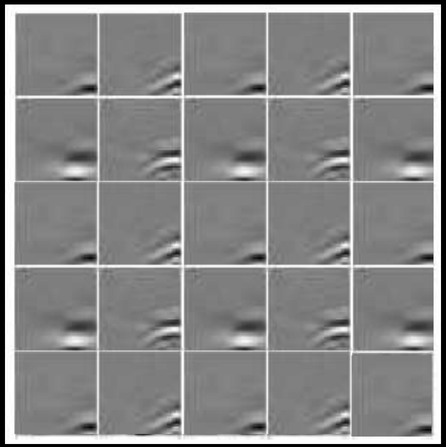
Convolution Neural Networks (Weight Tying)





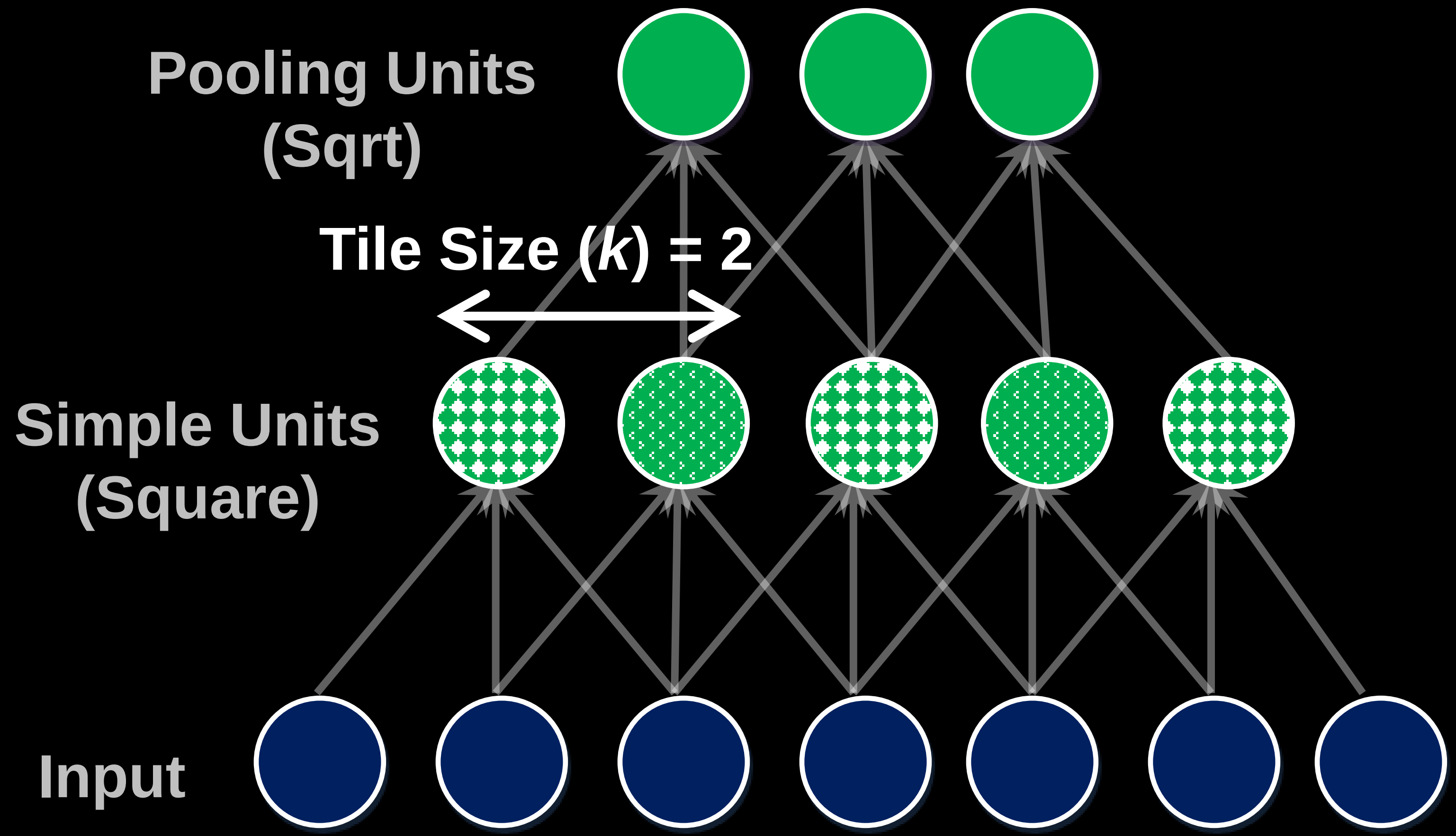
Tiled Networks (Partial Weight Tying)



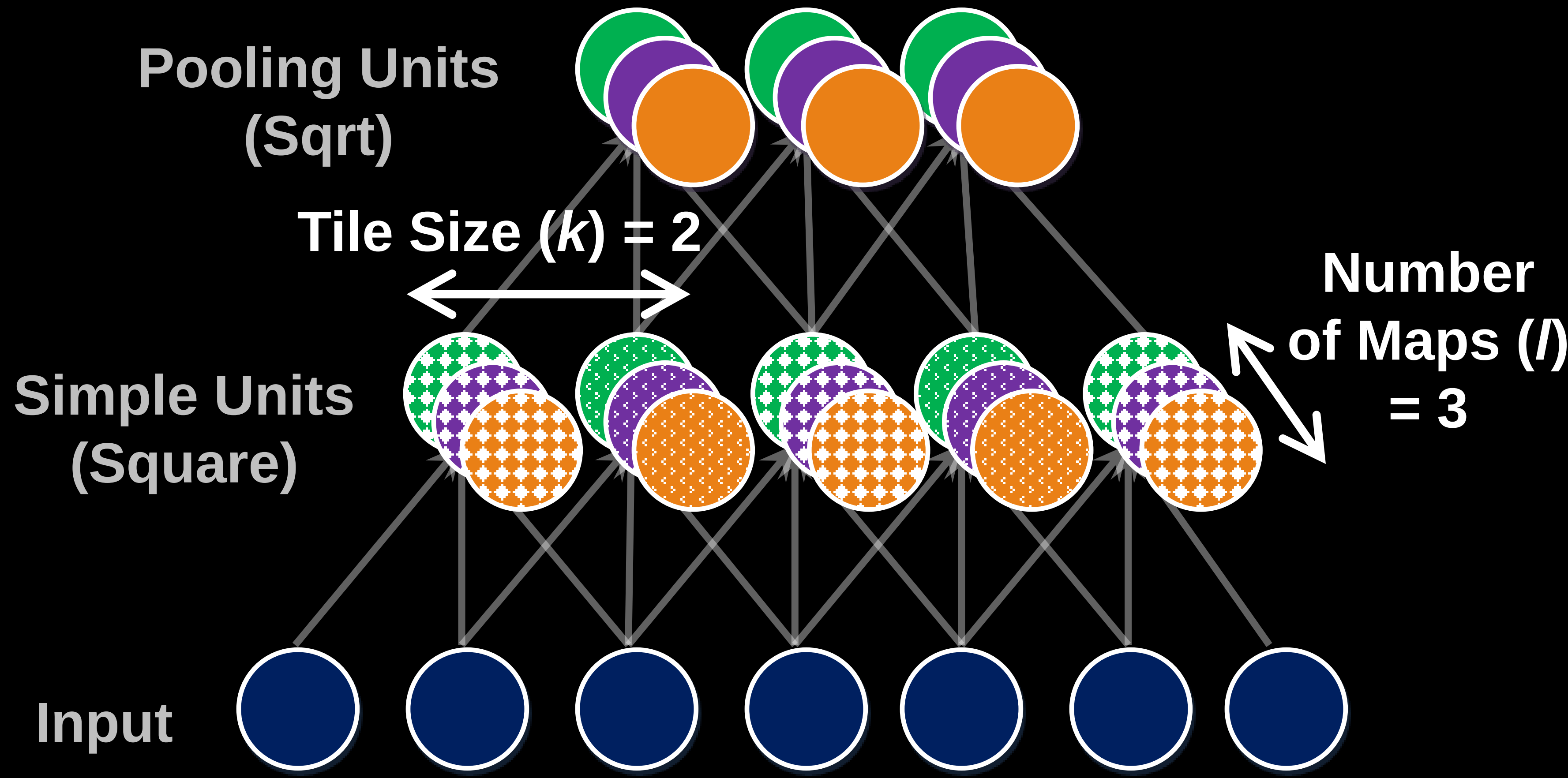


Local pooling can capture complex invariances (not just translation); but total number of parameters is small.

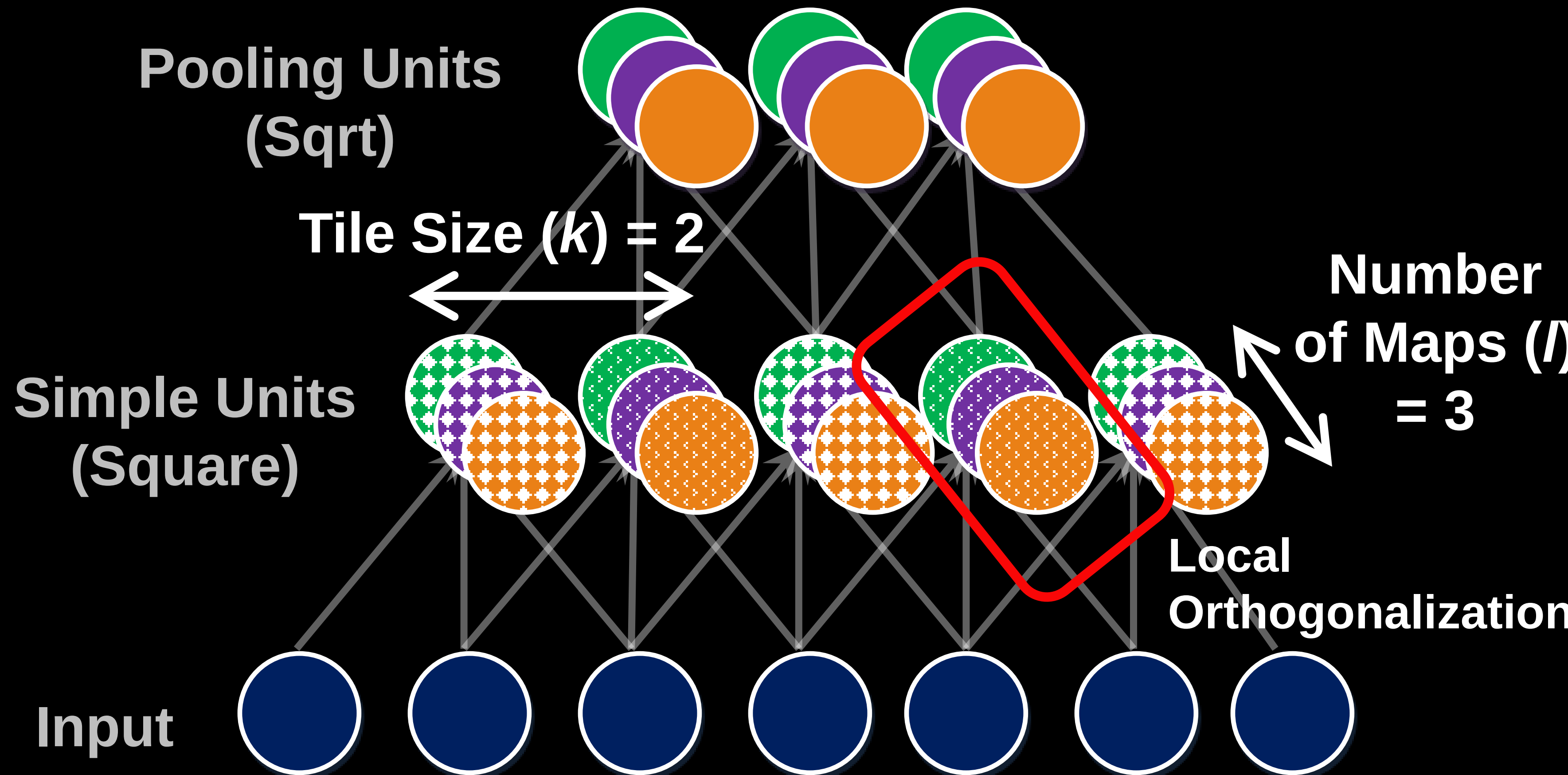
Tiled Networks (Partial Weight Tying)



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%

Summary/Big ideas

Summary/Big ideas

- 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
- Has worked well so far in two regimes (confusing to outsiders):
  - Lots of labeled data. "Train the heck out of the network."
  - Unsupervised Feature Learning/Self-Taught learning
- Scalability is important.
- Detailed tutorial: http://deeplearning.stanford.edu

END END END