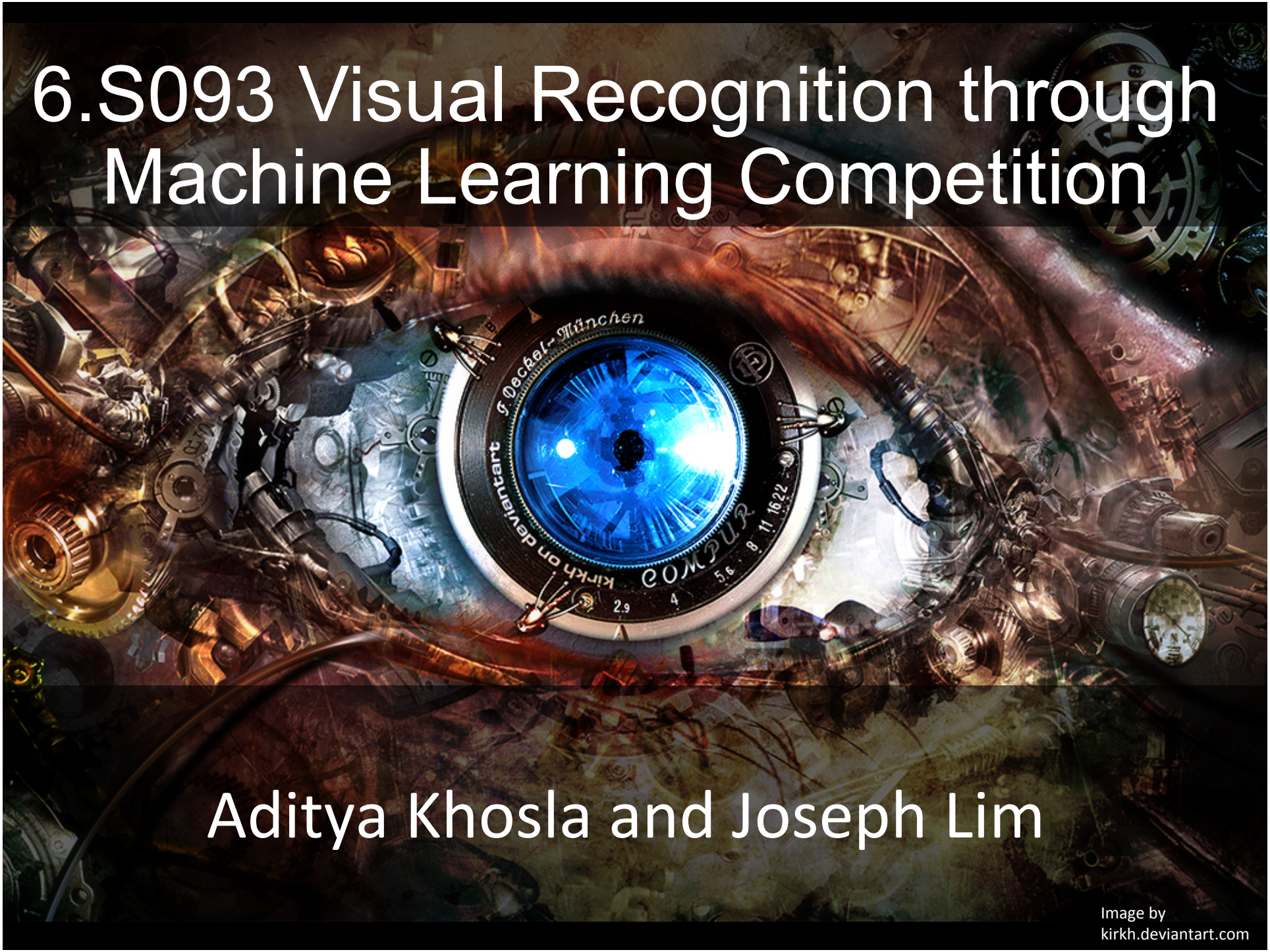
6.S093 Visual Recognition through Machine Learning Competition



Aditya Khosla and Joseph Lim

Image by
kirkh.deviantart.com

Today's class

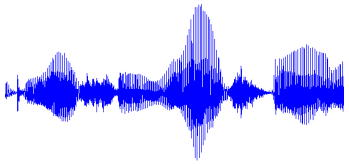
- Part 1: Introduction to deep learning
  - What is deep learning?
  - Why deep learning?
  - Some common deep learning algorithms

- Part 2: Deep learning tutorial
  - Please install Python++ now!

Slide credit

- Many slides are taken/adapted from Andrew Ng's

Typical goal of machine learning

input			output
images/video		ML	Label: “Motorcycle” Suggest tags Image search …
audio		ML	Speech recognition Music classification Speaker identification …
text		ML	Web search Anti-spam Machine translation …

Typical goal of machine learning

Feature engineering: most time consuming!

input

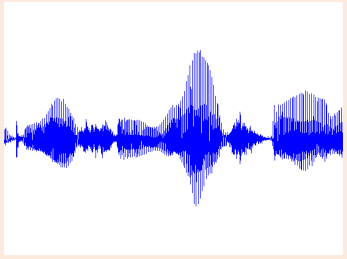
tput

images/video

ML

Label: “Motorcycle”
Suggest tags
Image search
…

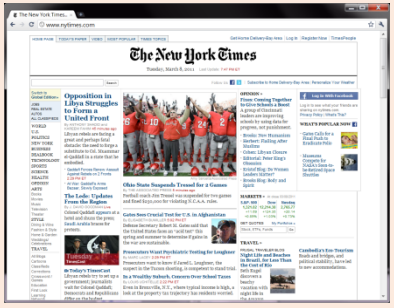
audio



ML

Speech recognition
Music classification
Speaker identification
…

text



ML

Web search
Anti-spam
Machine translation
…

Our goal in object classification

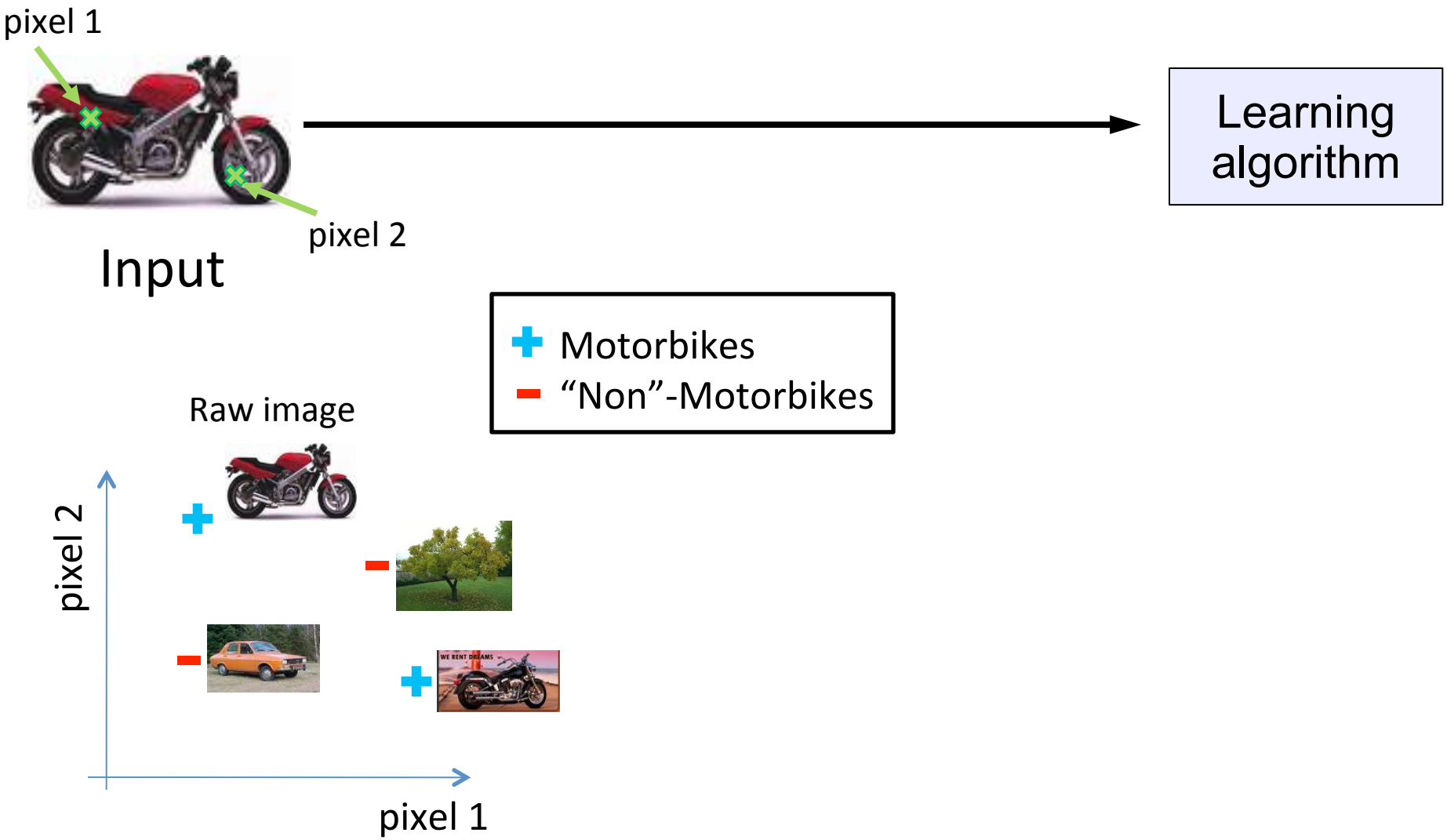
Why is this hard?

You see this:

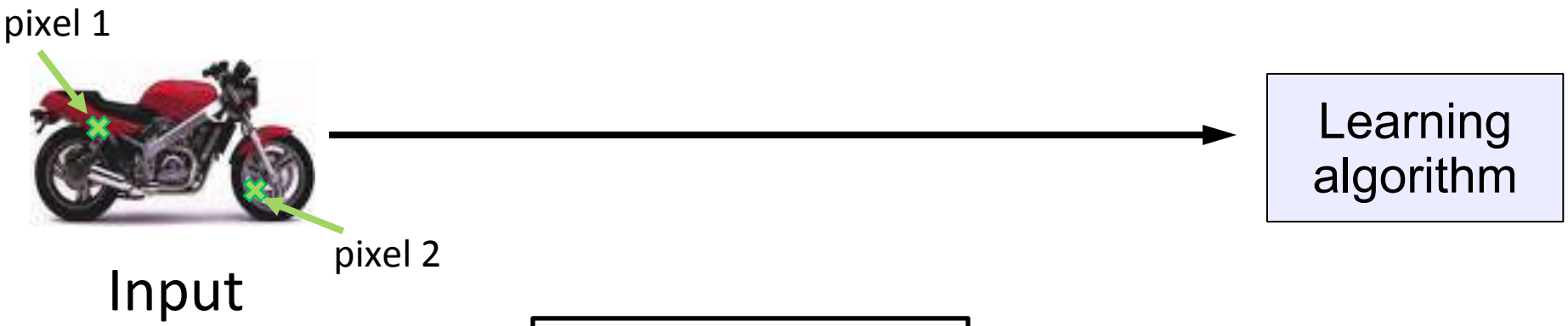
But the camera sees this:

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

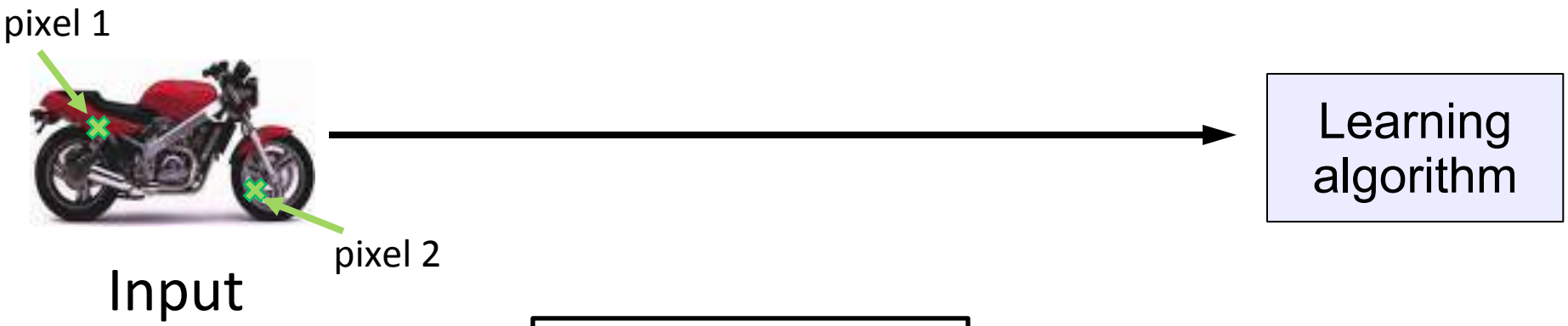
Pixel-based representation



Pixel-based representation



Pixel-based representation



What we want

Input

Feature representation

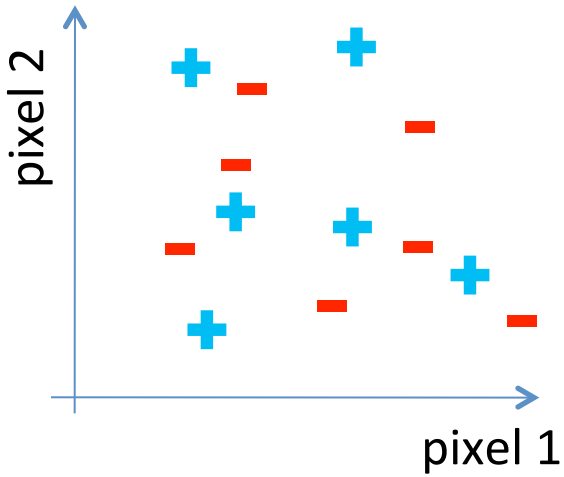
E.g., Does it have Handlebars? Wheels?

Learning algorithm

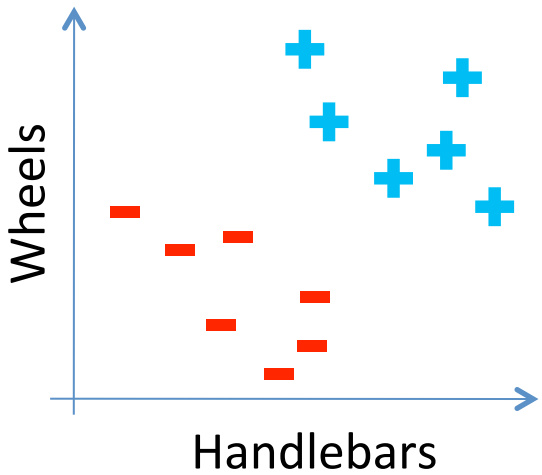
Motorbikes

"Non"-Motorbikes

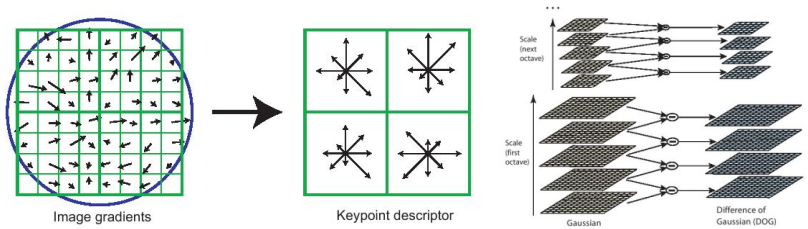
Raw image



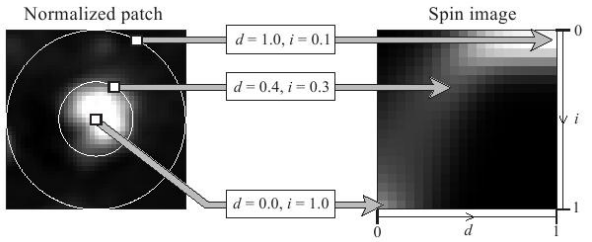
Features



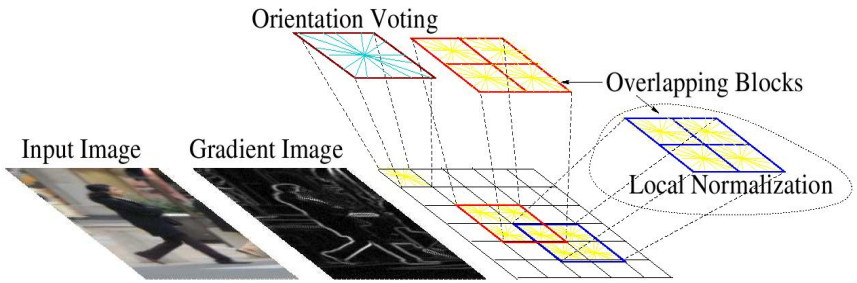
Some feature representations



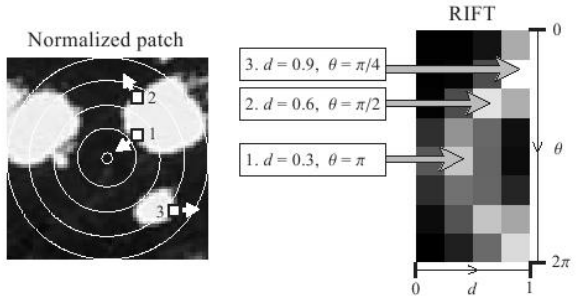
SIFT



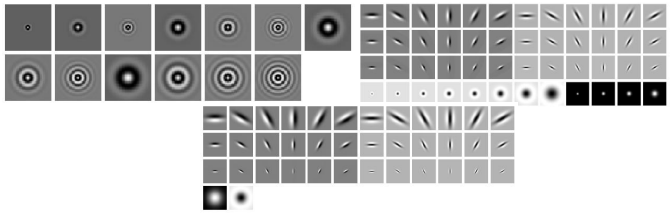
Spin image



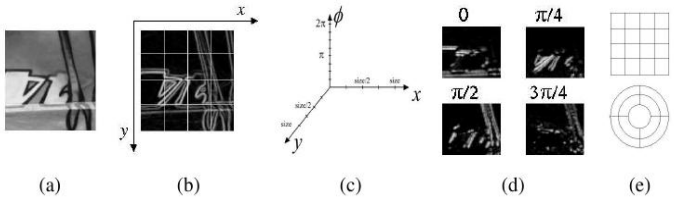
HoG



RIFT

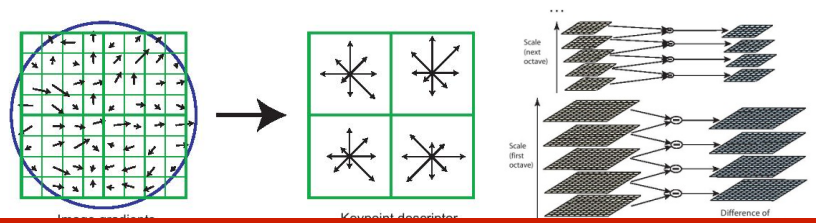


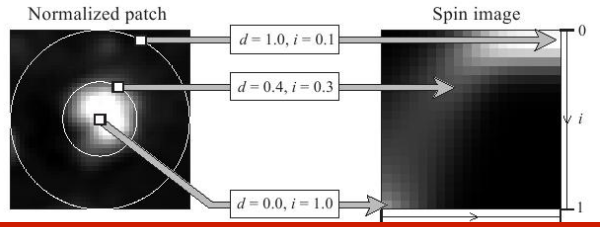
Textons



GLOH

Some feature representations

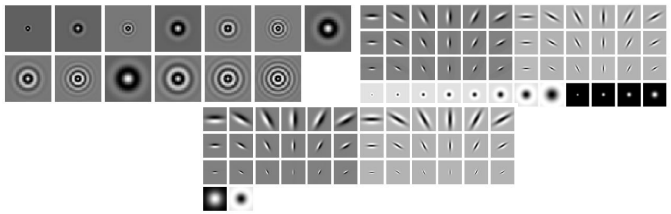




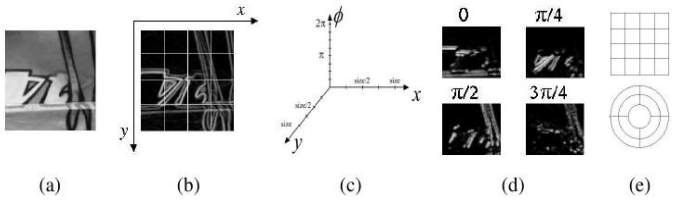
Coming up with features is often difficult, time-consuming, and requires expert knowledge.

HoG

RIFT

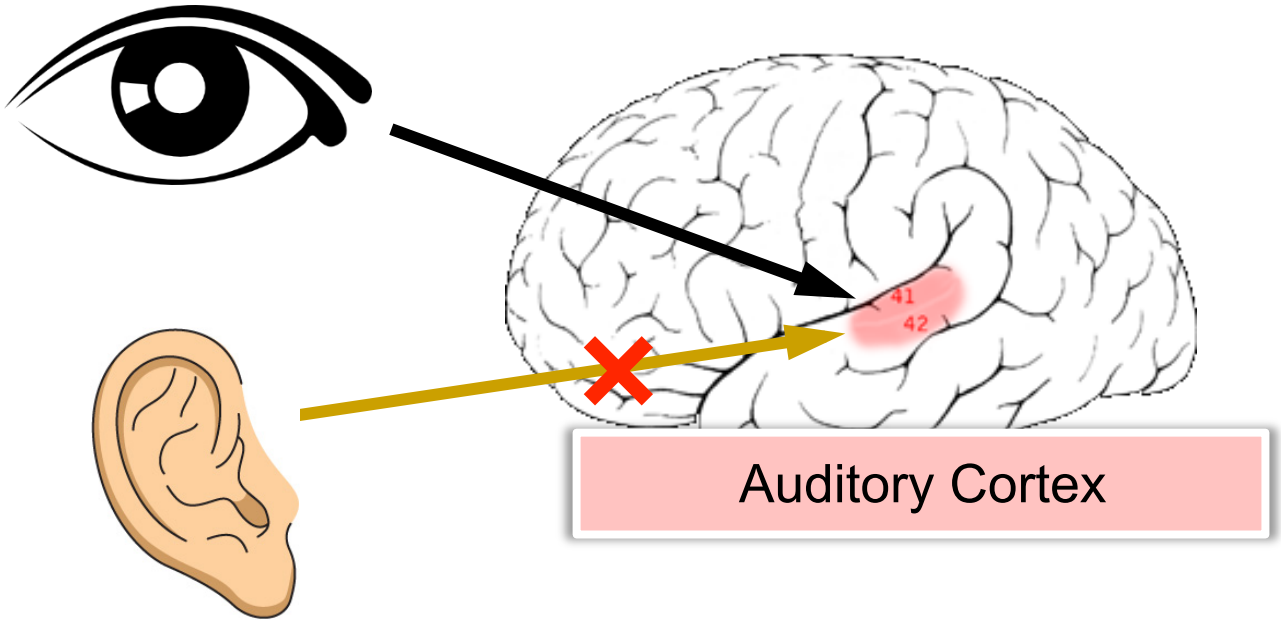


Textons



GLOH

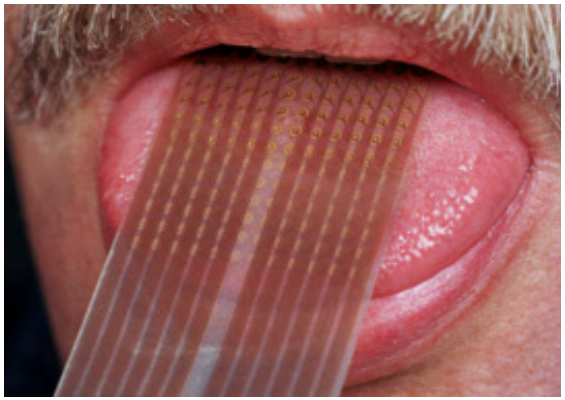
The brain:
potential motivation for deep learning



Auditory cortex learns to see!

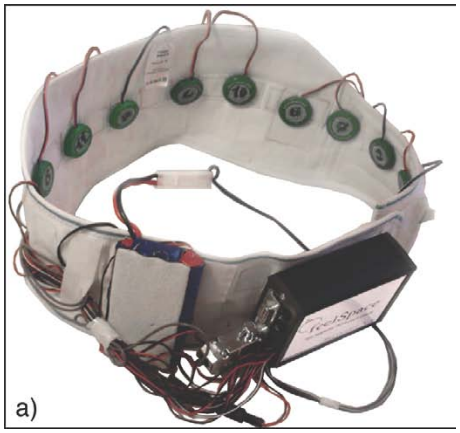
[Roe et al., 1992]

The brain adapts!

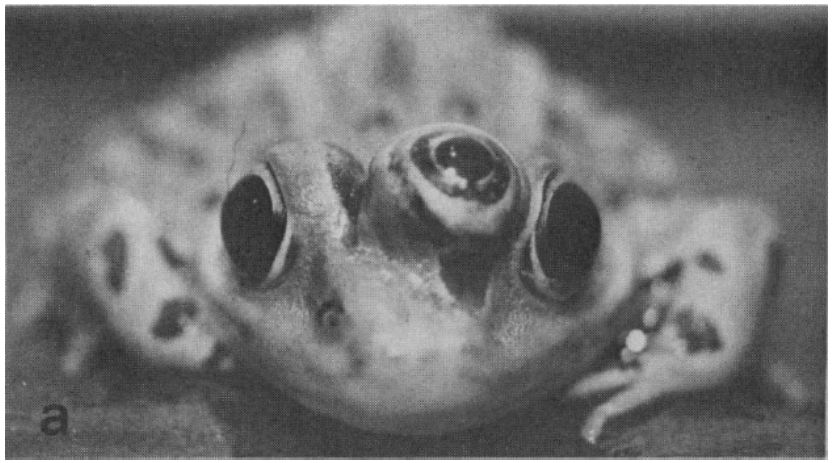


Seeing with your tongue

Human echolocation (sonar)



Haptic belt: Direction sense



Implanting a 3rd eye

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

Basic idea of deep learning

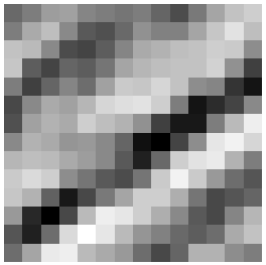
- Also referred to as representation learning or unsupervised feature learning (with subtle distinctions)

- Is there some way to extract **meaningful features** from data **even without knowing the task** to be performed?

- Then, throw in some hierarchical ‘stuff’ to make it ‘deep’

Feature learning problem

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



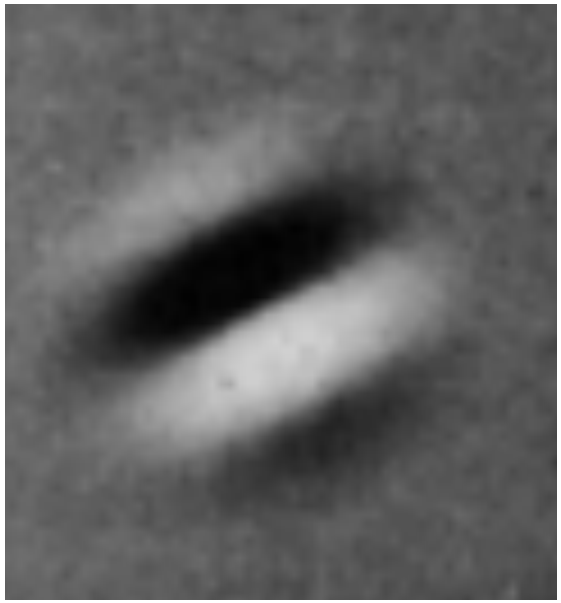
$$\begin{pmatrix} 255 \\ 98 \\ 93 \\ 87 \\ 89 \\ 91 \\ 48 \\ \dots \end{pmatrix}$$

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

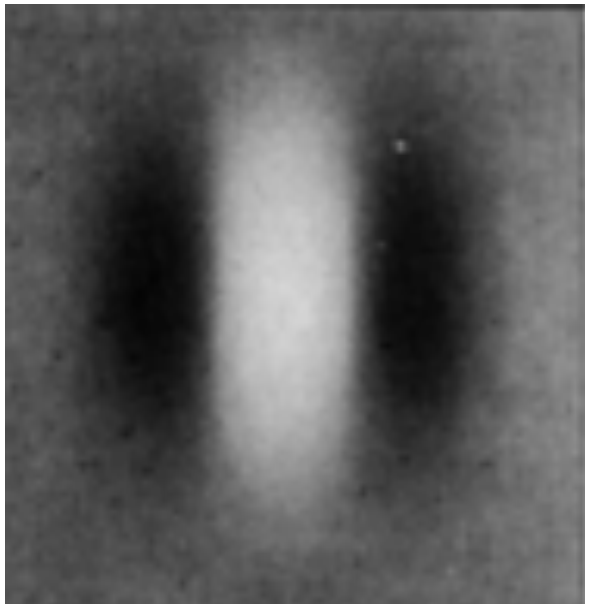
First stage of visual processing: V1

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)

Learning sensor representations

Sparse coding (Olshausen & Field,1996)

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:

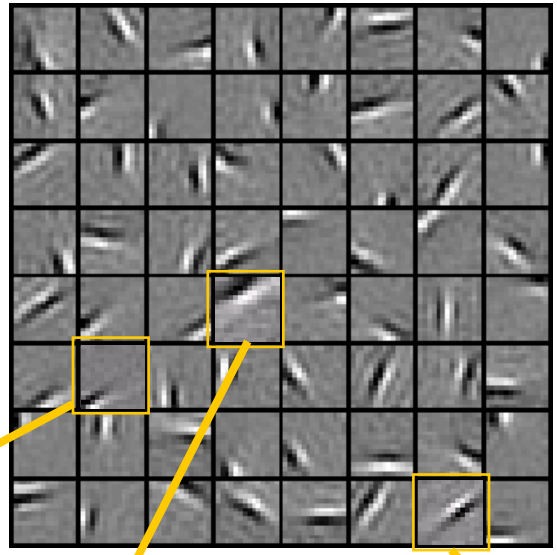
$$x \approx \sum_{j=1}^{k} a_j \phi_j$$

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

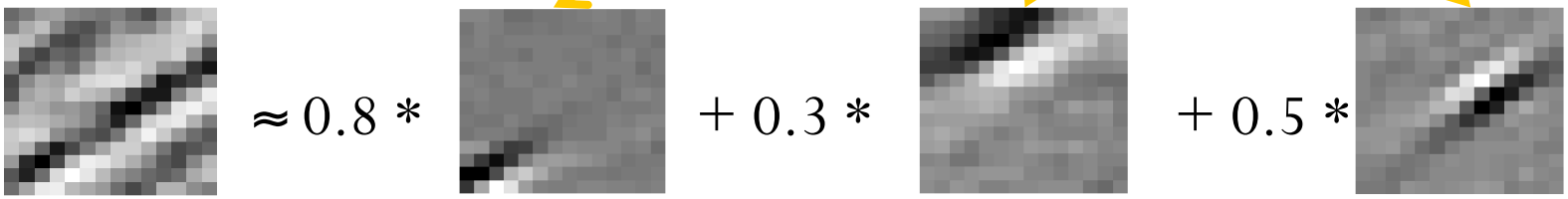
Sparse coding illustration

Natural Images

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



Test example

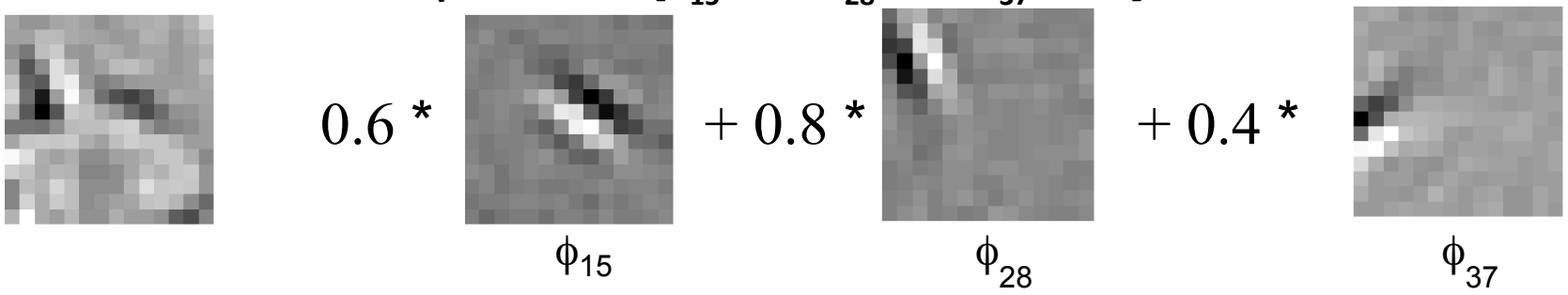


$$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$$

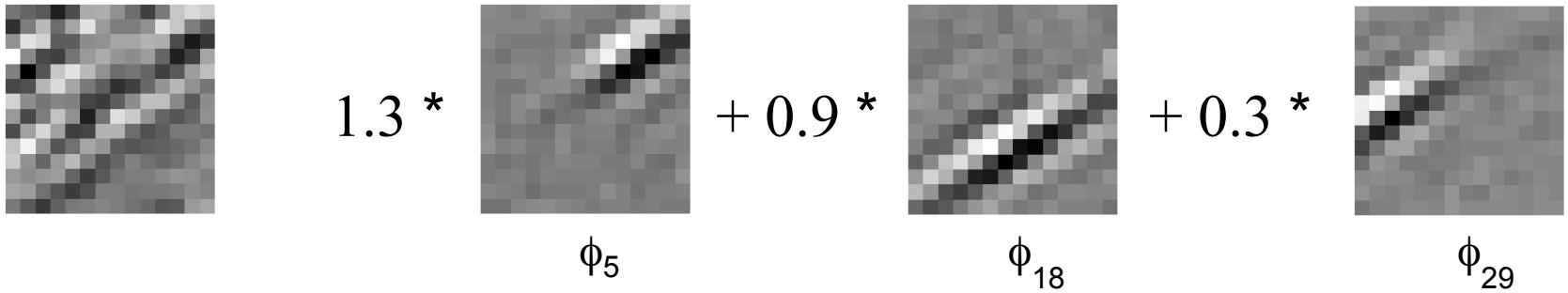
$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

Sparse coding illustration

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



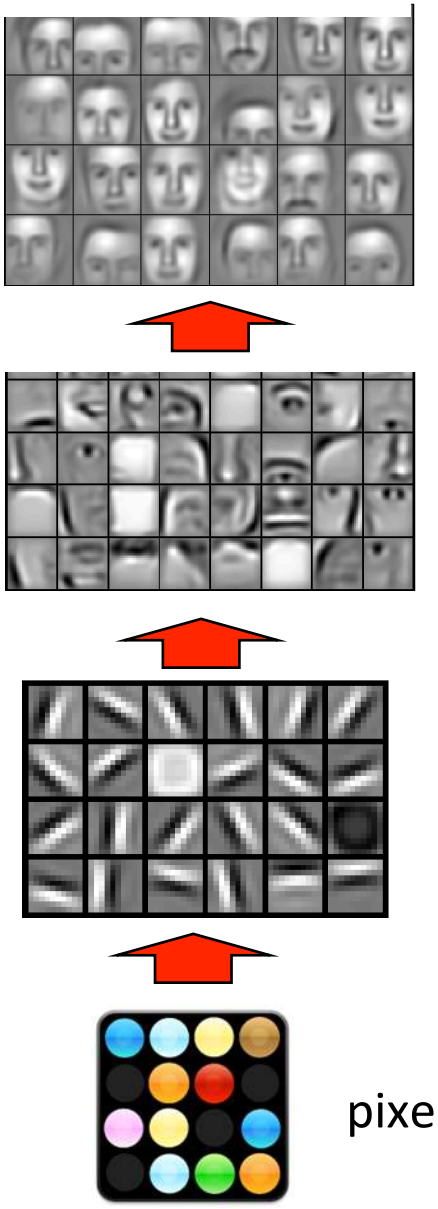
Represent as: [a_5=1.3, a_{18}=0.9, a_{29} = 0.3]



- **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.

Going deep

Training set: Aligned images of faces.



object models

object parts (combination of edges)

edges

pixels

[Honglak Lee]

Why deep learning?

Task: video activity recognition

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%**

[Le, Zhou & Ng, 2011]

Audio

TIMIT Phone classification	Accuracy
Prior art (Clarkson et al.,1999)	79.6%
Feature learning	**80.3%**

TIMIT Speaker identification	Accuracy
Prior art (Reynolds, 1995)	99.7%
Feature learning	**100.0%**

Images

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

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

Video

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

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

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

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

Text/NLP

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

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

Speech recognition on Android

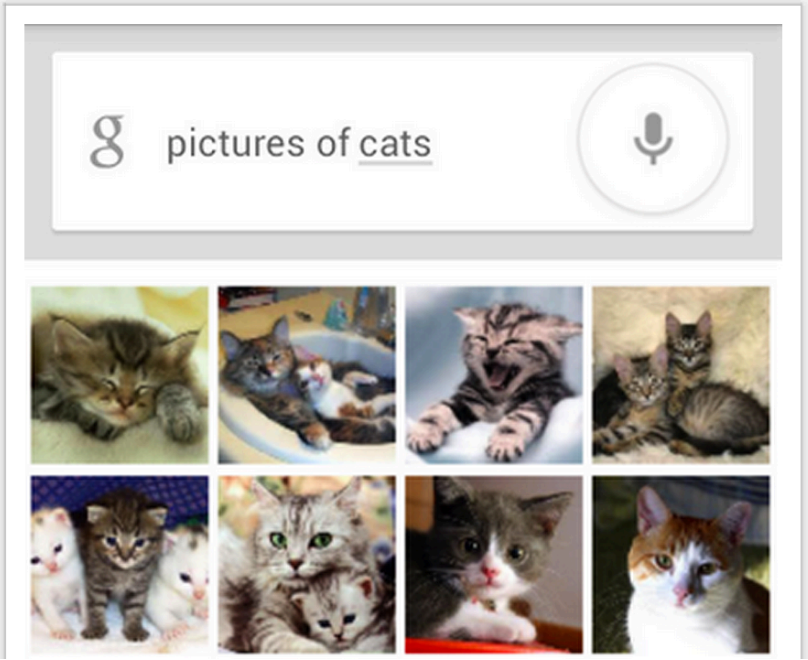
AUG
6

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!

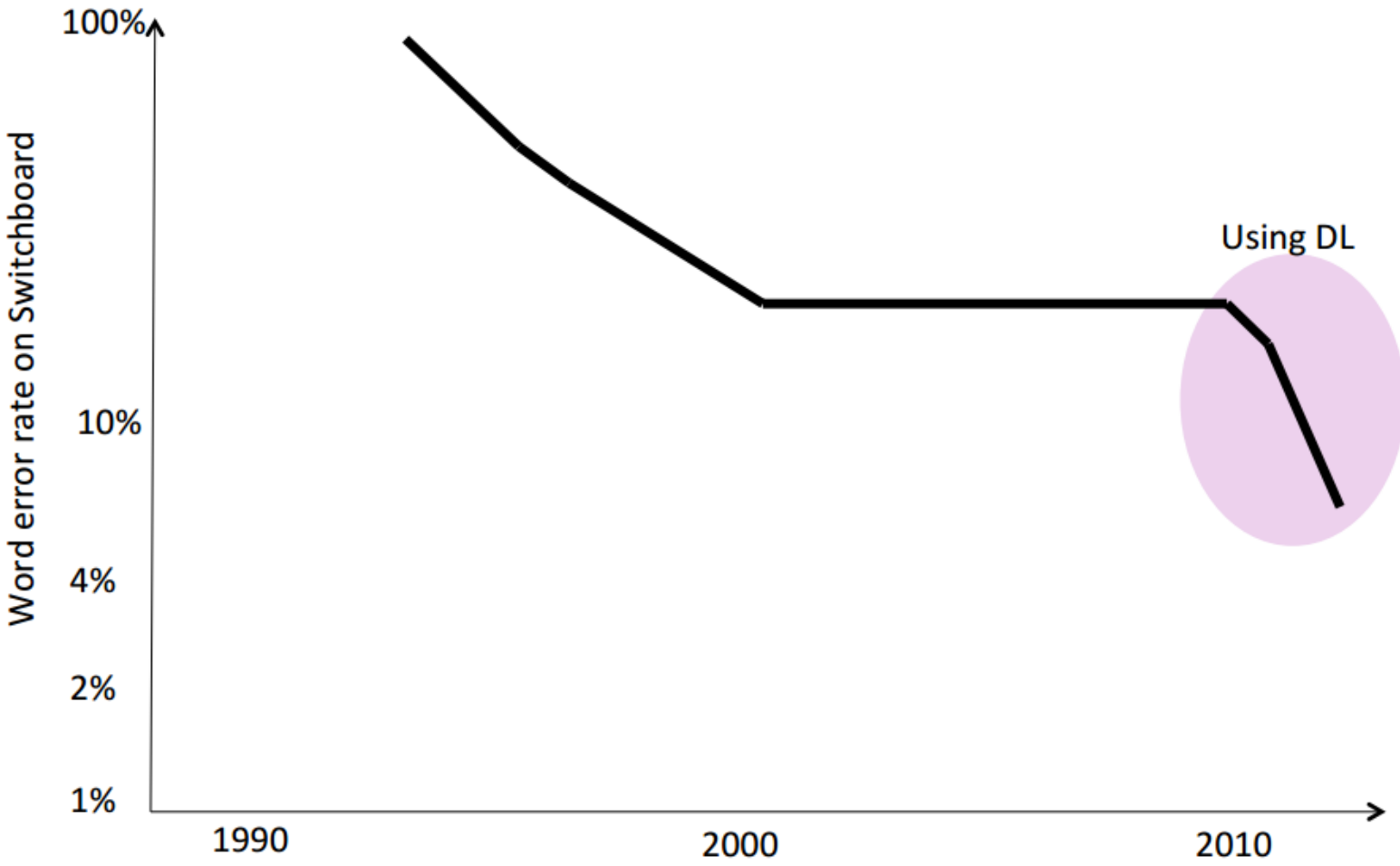
What's the point of building a gigantic cat detector you might ask? When you combine large amounts of data, large-scale distributed computing and powerful machine learning algorithms, you can apply the technology to address a large variety of practical problems.



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

Impact on speech recognition



Application to Google Streetview

ImageNet classification: 22,000 classes

…
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

ImageNet Classification: 14M images, 22k categories

0.005%	9.5%	?
Random guess	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

ImageNet Classification: 14M images, 22k categories

0.005%	9.5%	21.3%
Random guess	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

Some common deep architectures

- Autoencoders
- Deep belief networks (DBNs)
- Convolutional variants
- Sparse coding

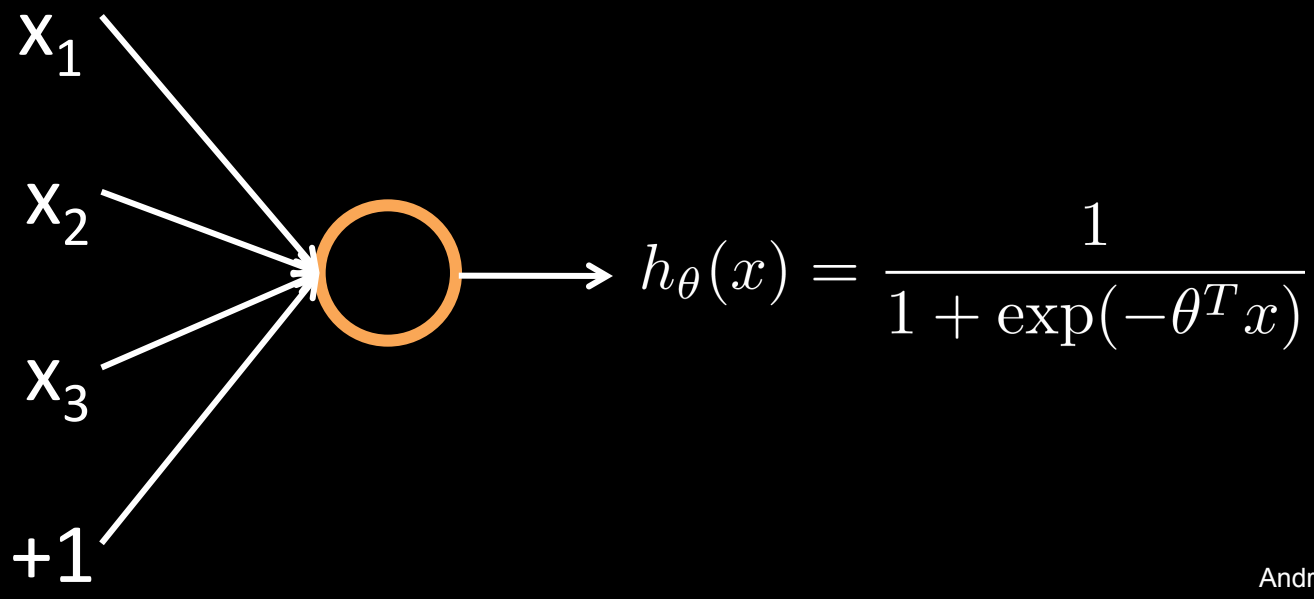
Logistic regression

Logistic regression has a learned parameter vector θ.
On input x, it outputs:

$$\begin{aligned} h_\theta(x) &= \sigma(\theta^T x) \\ &= \frac{1}{1+\exp(-\theta^T x)} \end{aligned}$$

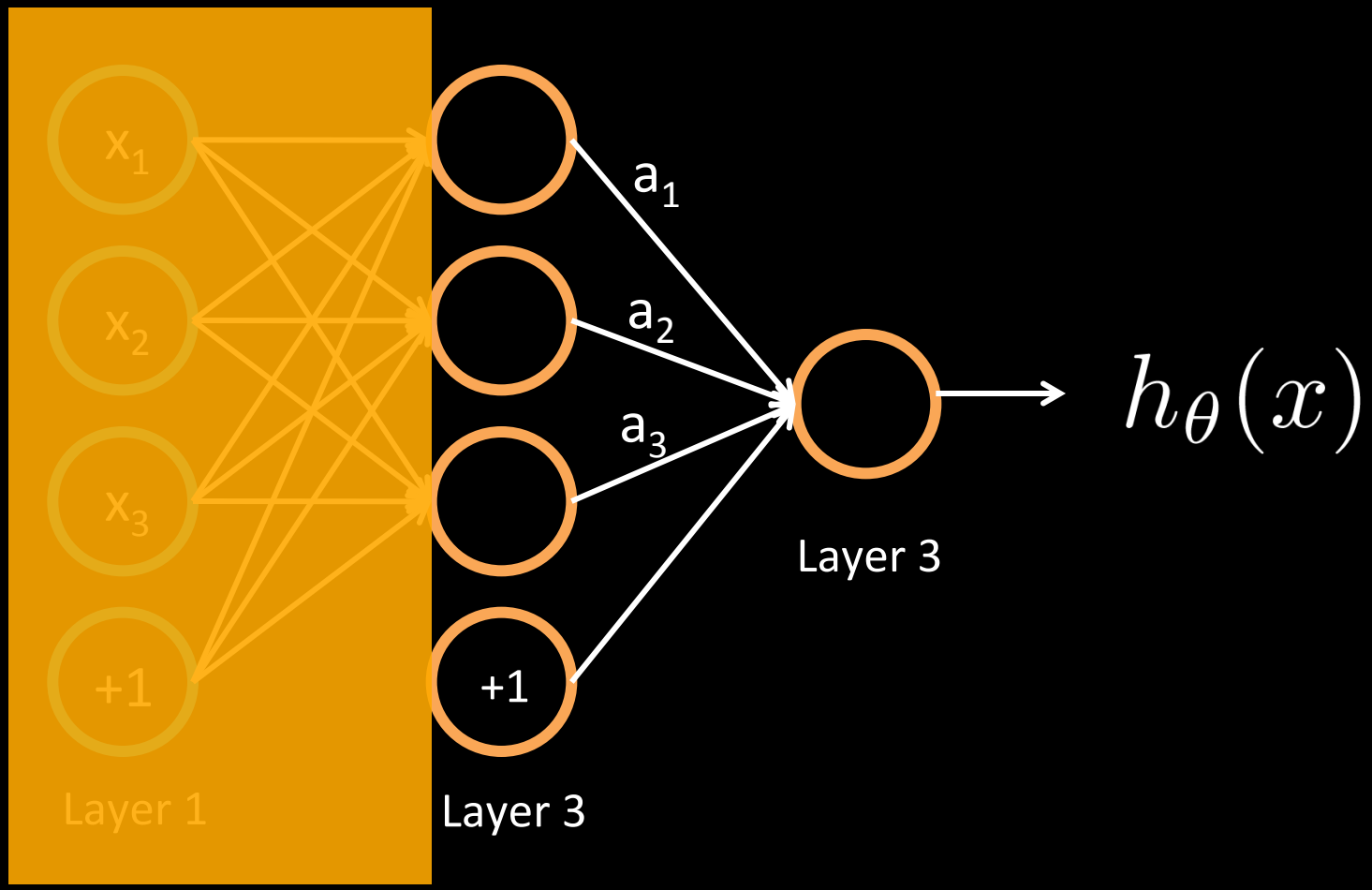
where $\sigma(z) = 1/(1+\exp(-z))$

Draw a logistic regression unit as:



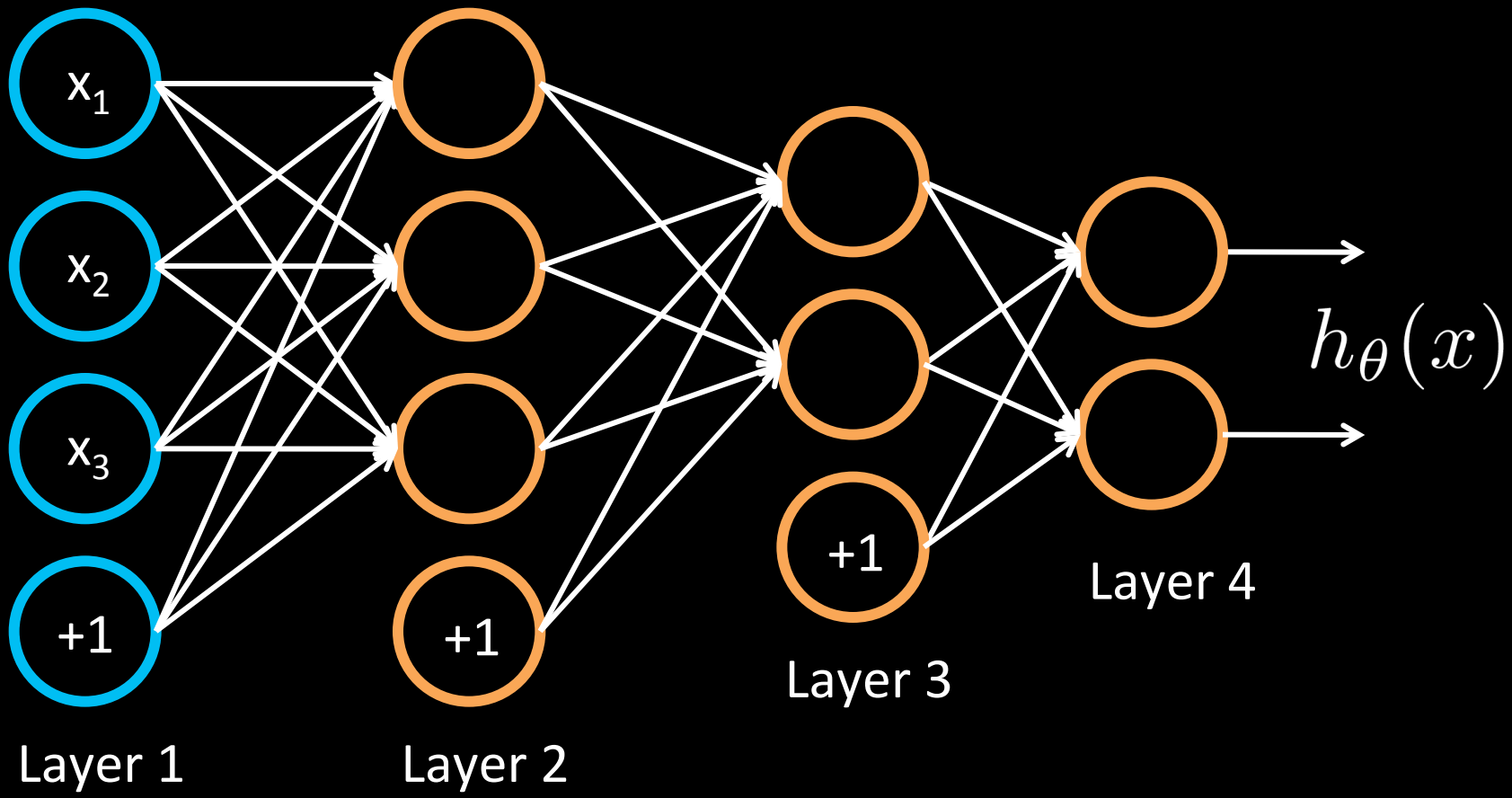
Neural Network

String a lot of logistic units together. Example 3 layer network:

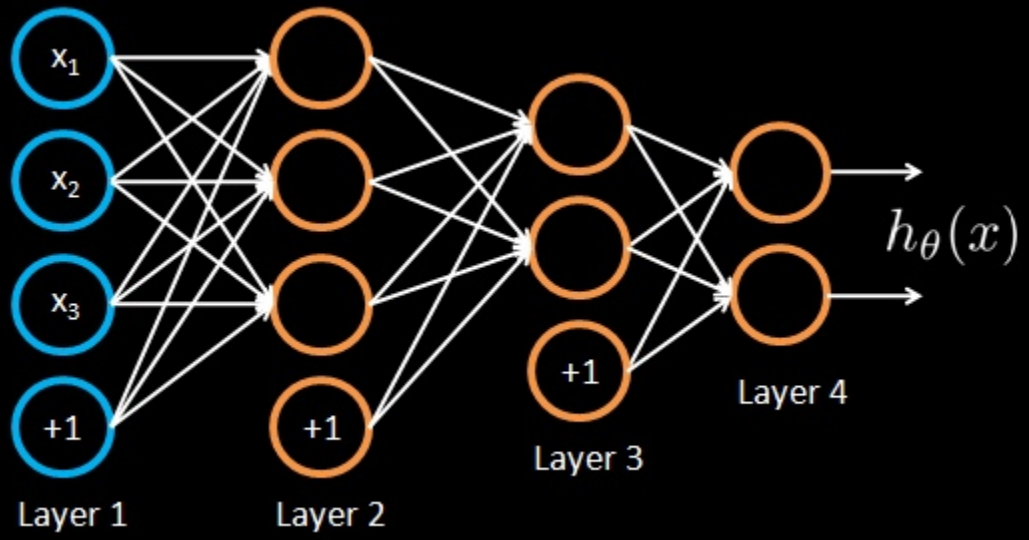


Neural Network

Example 4 layer network with 2 output units:



Training a neural network



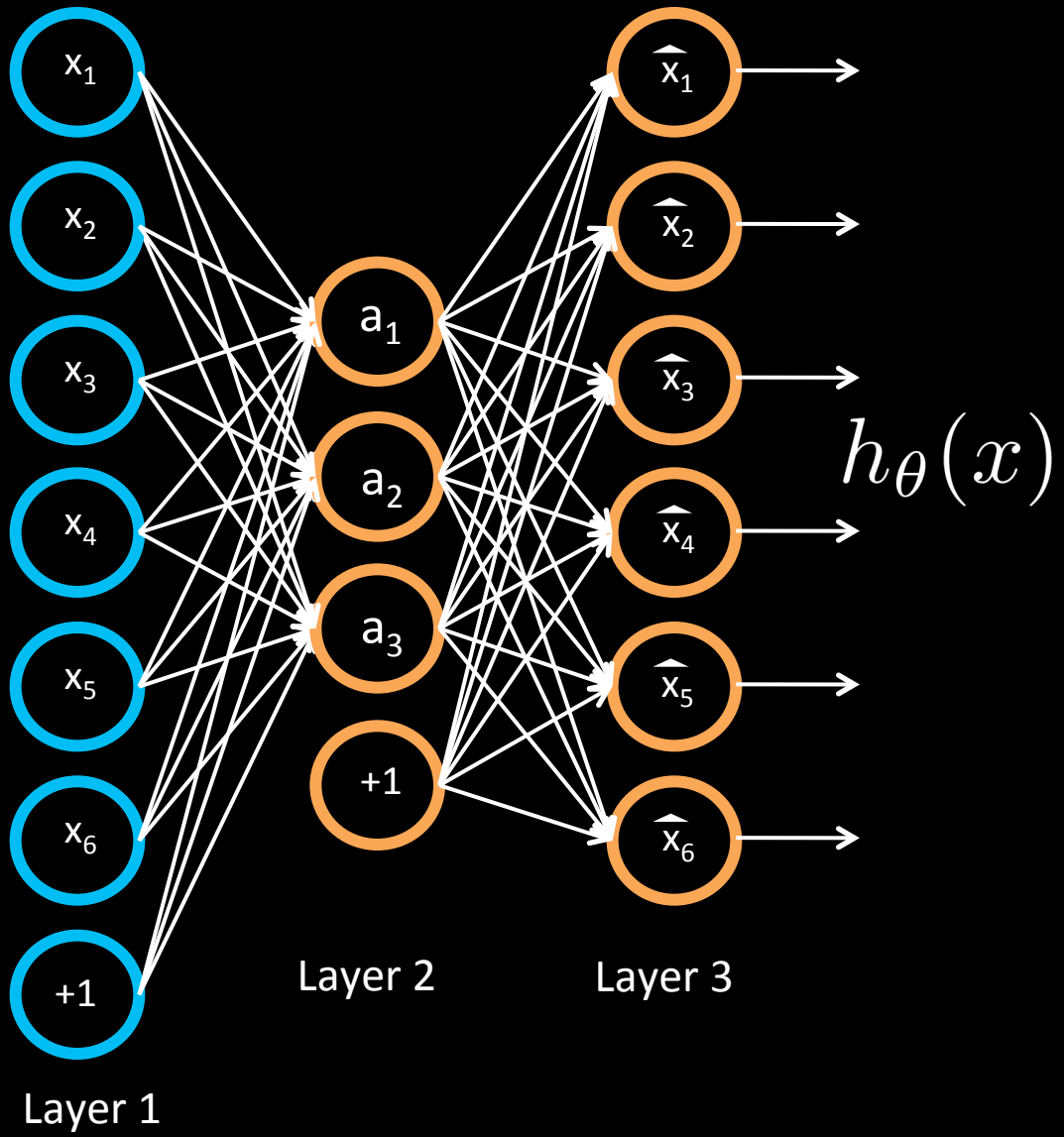
Given training set (x_1, y_1), (x_2, y_2), (x_3, y_3), ….

Adjust parameters θ (for every node) to make:

$$h_\theta(x_i) \approx y_i$$

(Use gradient descent. “Backpropagation” algorithm. Susceptible to local optima.)

Unsupervised feature learning with a neural network



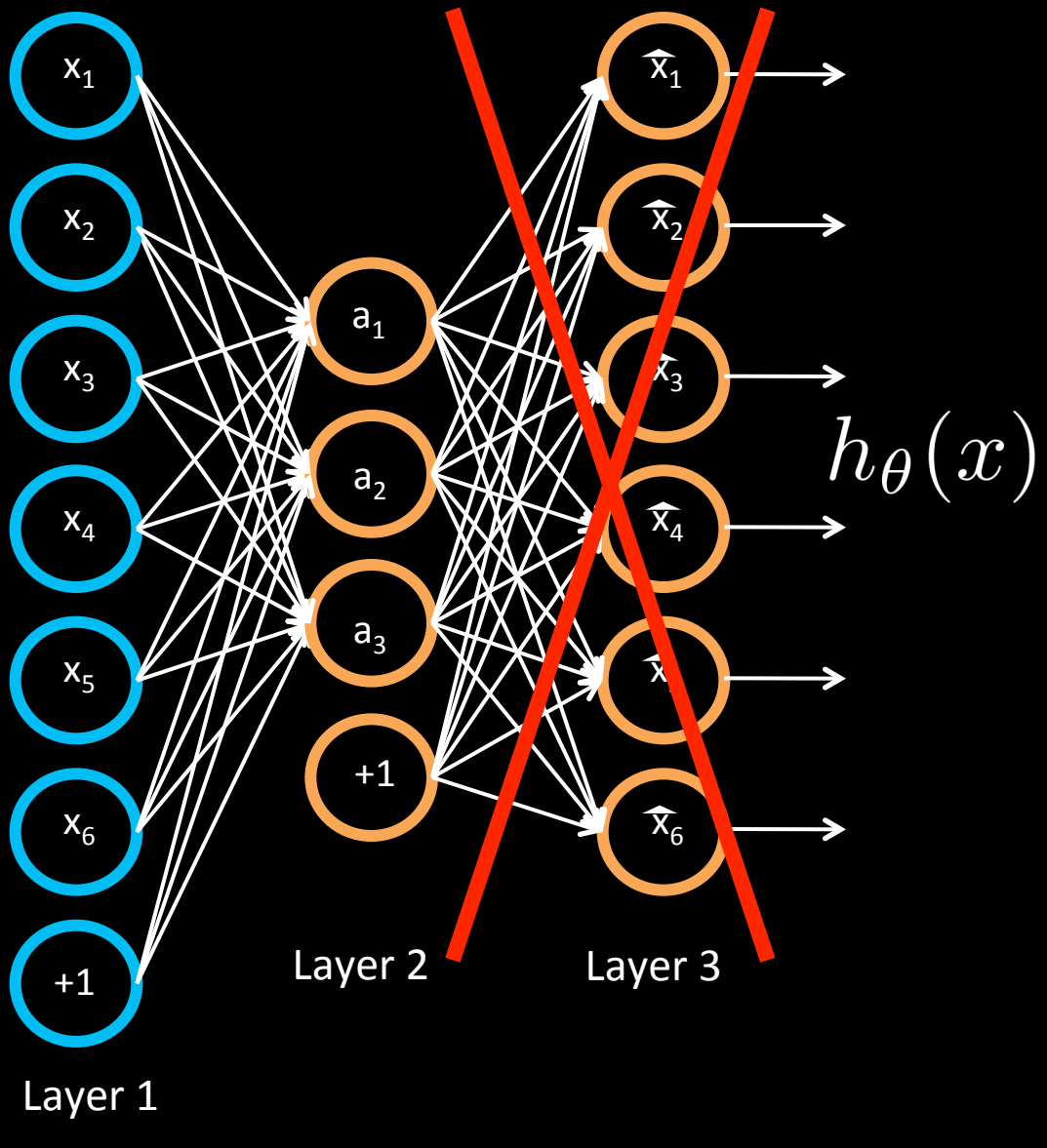
Autoencoder.

Network is trained to output the input (learn identify function).

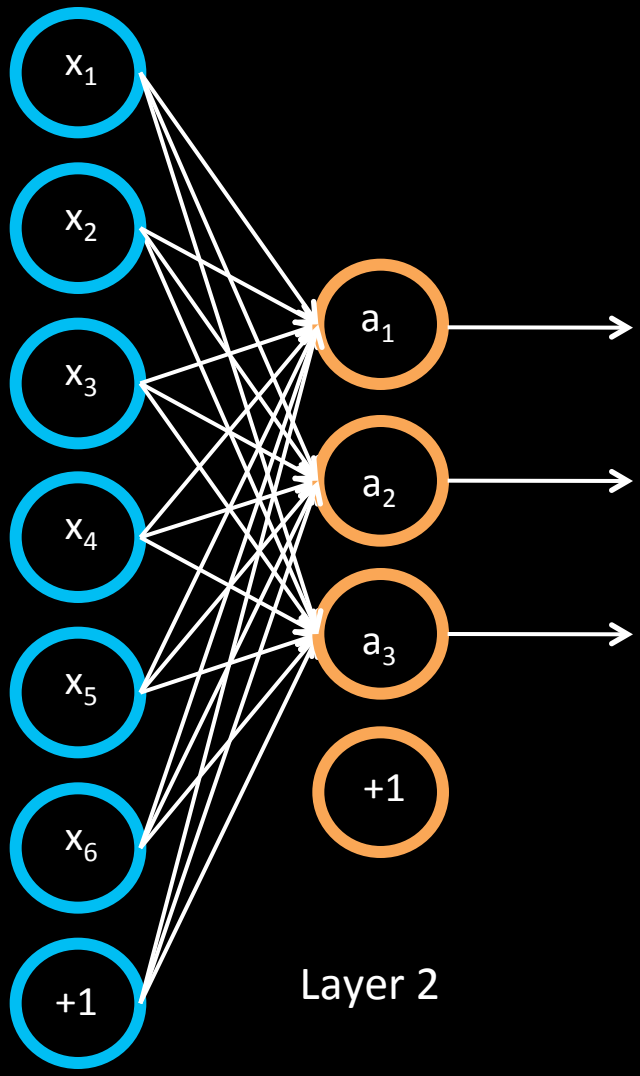
$$h_\theta(x) \approx x$$

Trivial solution unless:
- Constrain number of units in Layer 2 (learn compressed representation), or
- Constrain Layer 2 to be **sparse**.

Unsupervised feature learning with a neural network



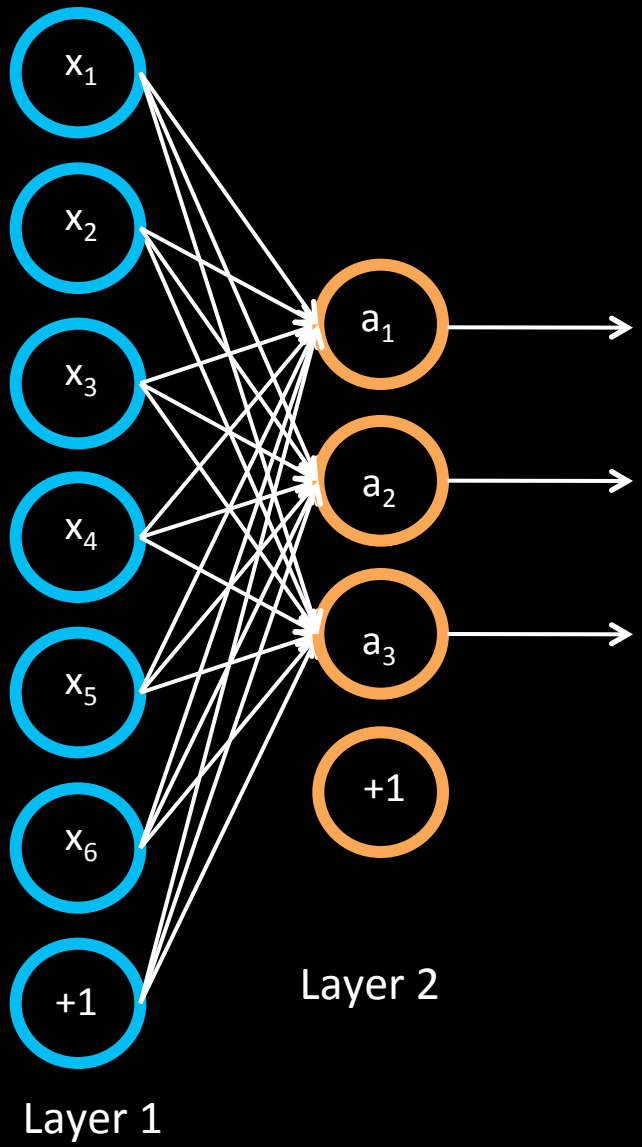
Unsupervised feature learning with a neural network



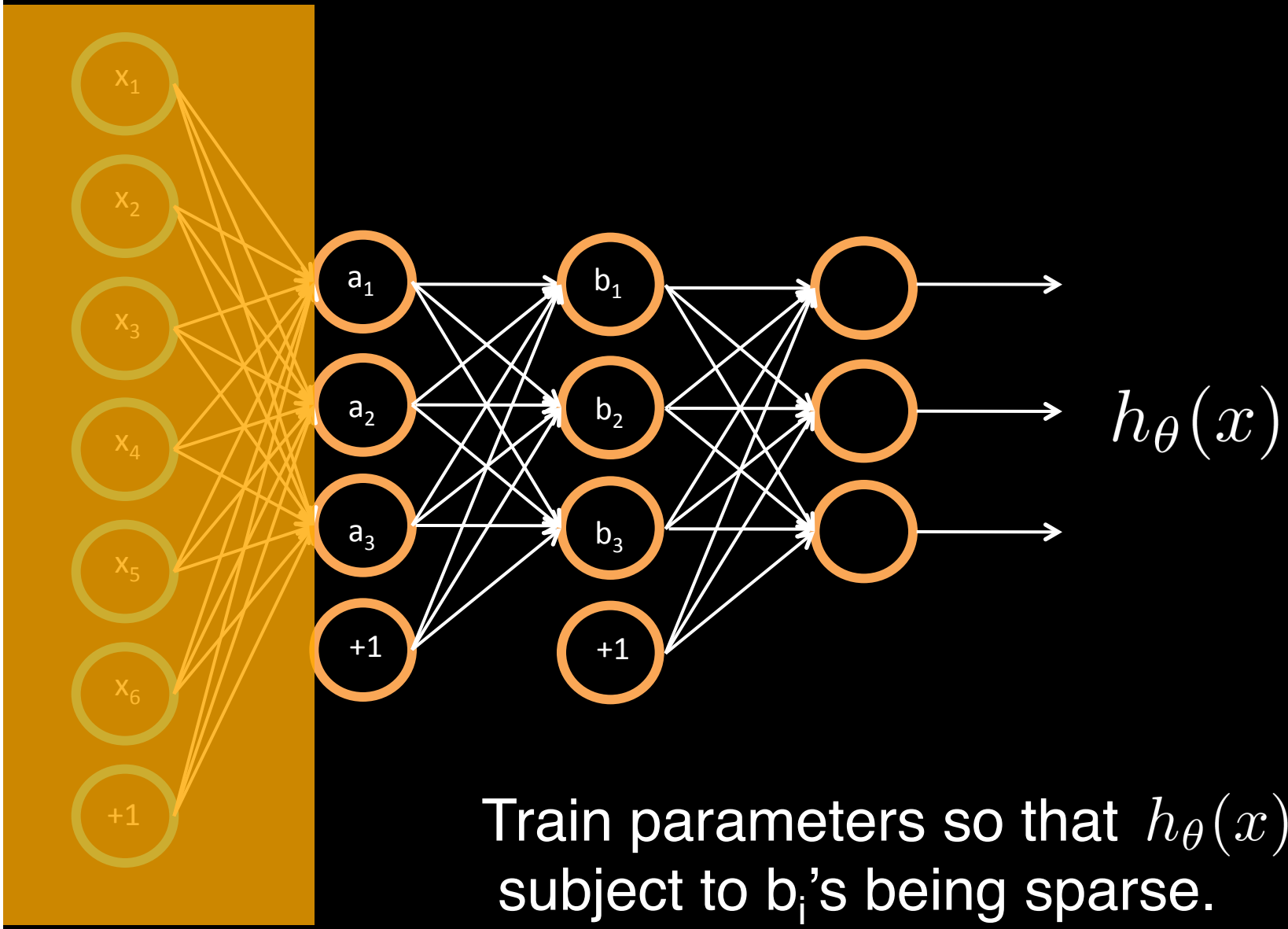
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

New representation for input.

Unsupervised feature learning with a neural network

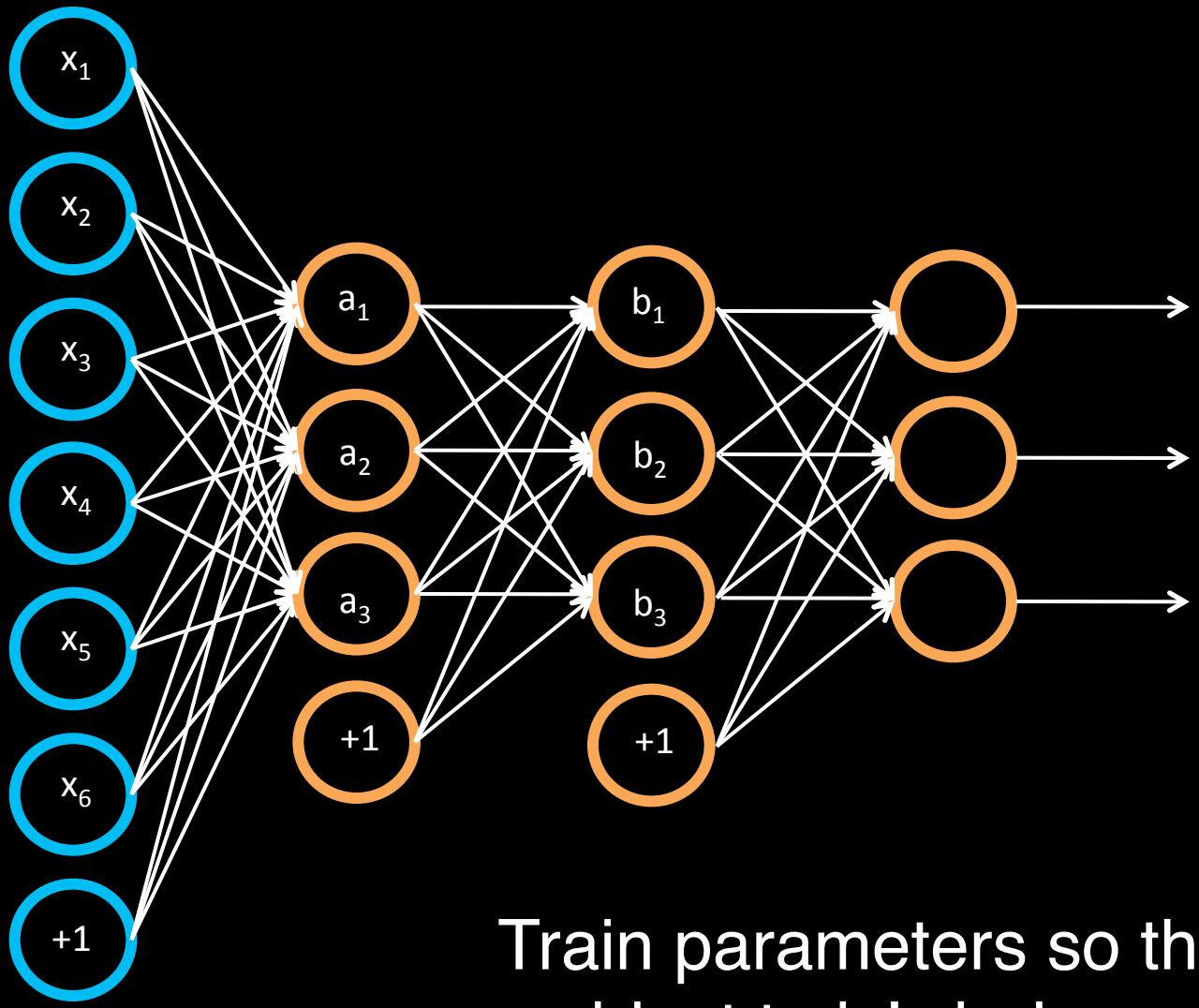


Unsupervised feature learning with a neural network



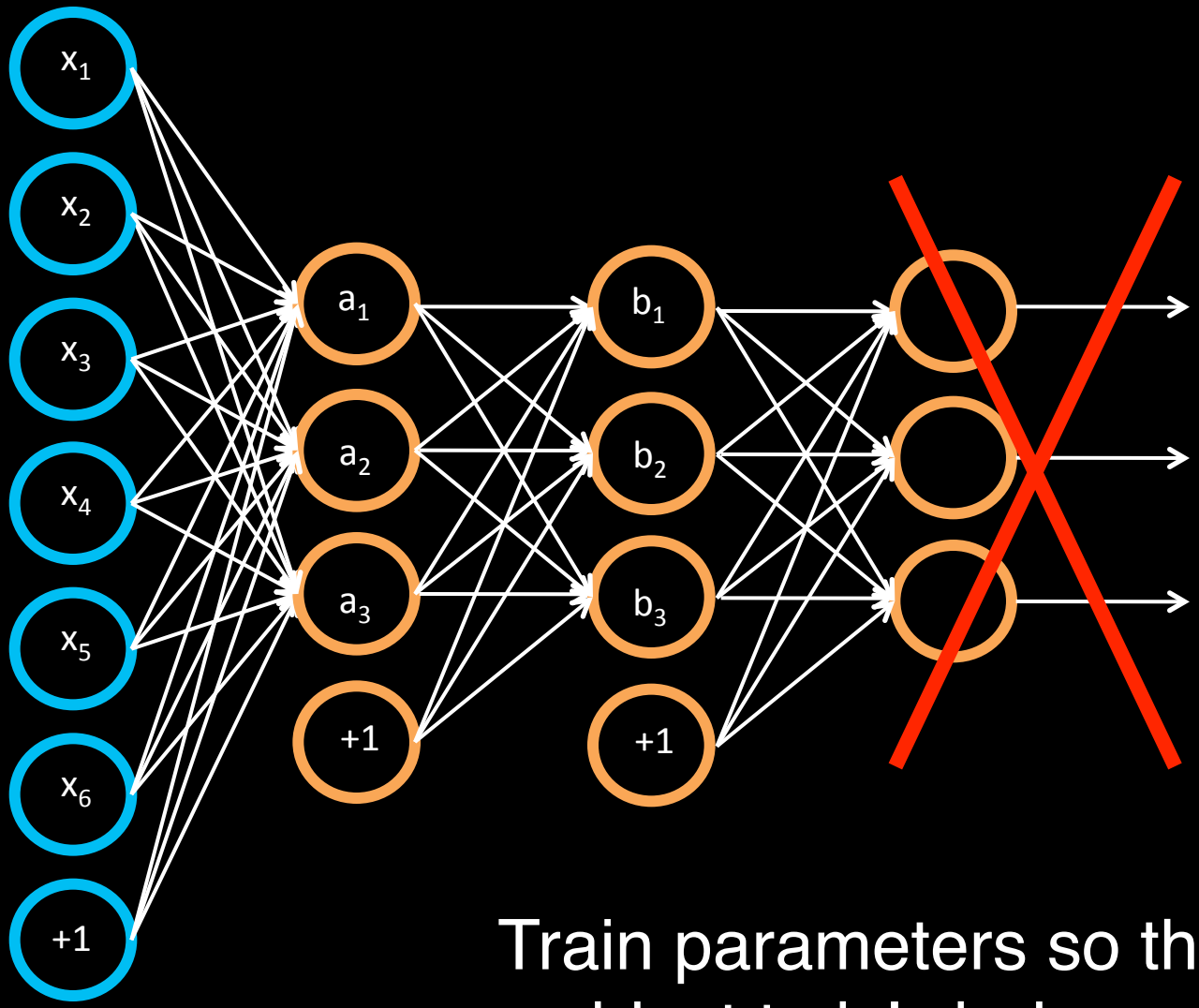
Train parameters so that $h_\theta(x) \approx a$, subject to b_i's being sparse.

Unsupervised feature learning with a neural network



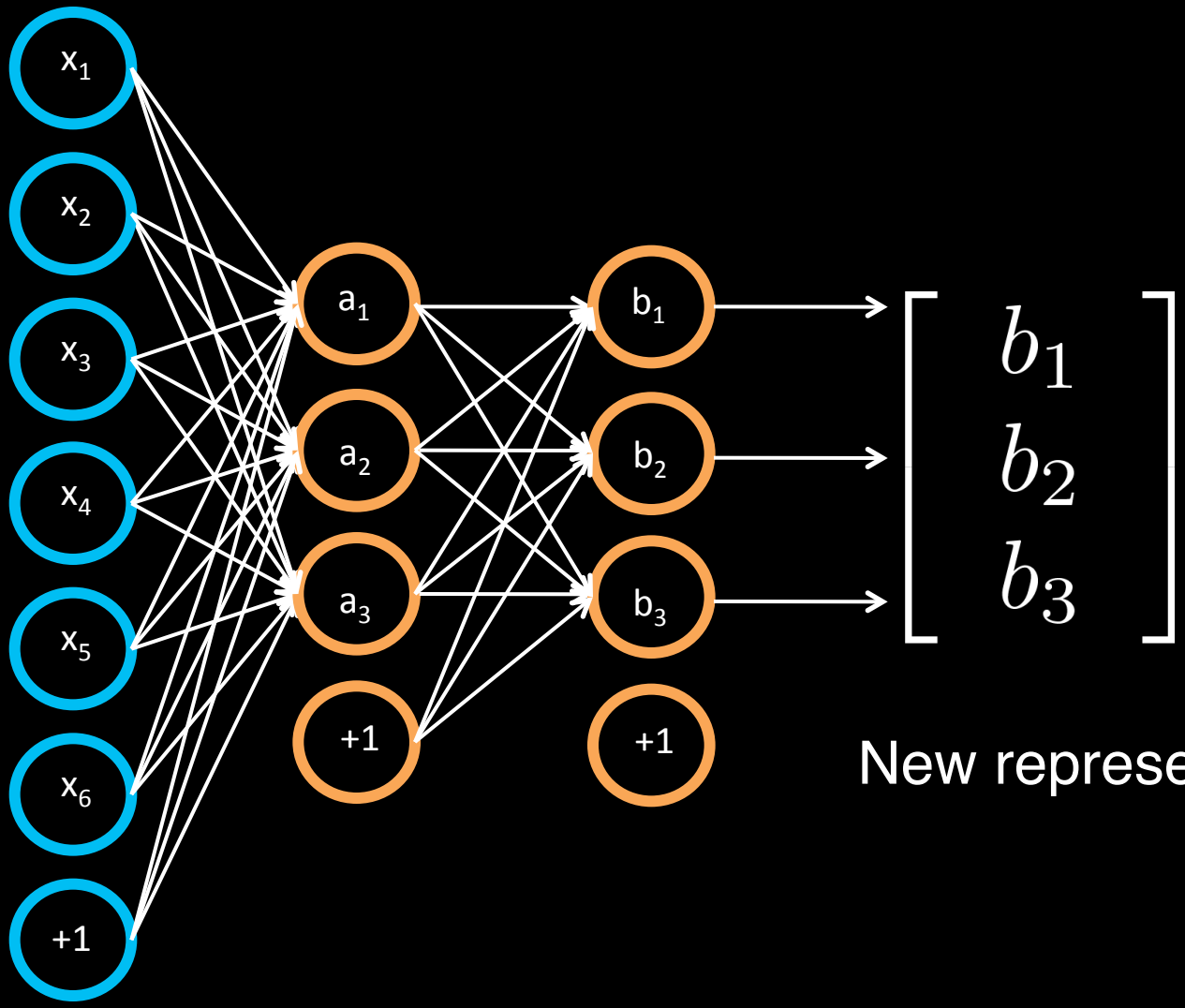
Train parameters so that $h_\theta(x) \approx a$, subject to b_i's being sparse.

Unsupervised feature learning with a neural network



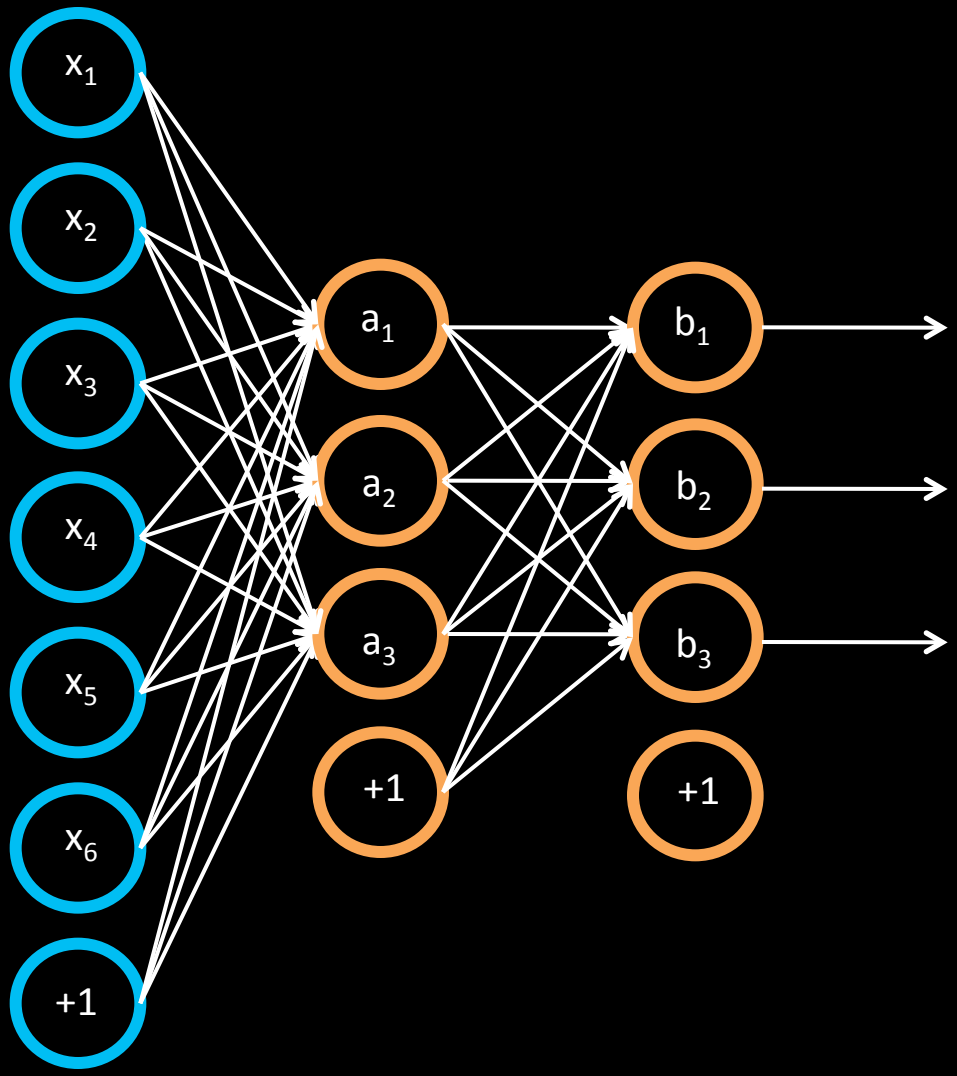
Train parameters so that $h_\theta(x) \approx a$, subject to b_i's being sparse.

Unsupervised feature learning with a neural network

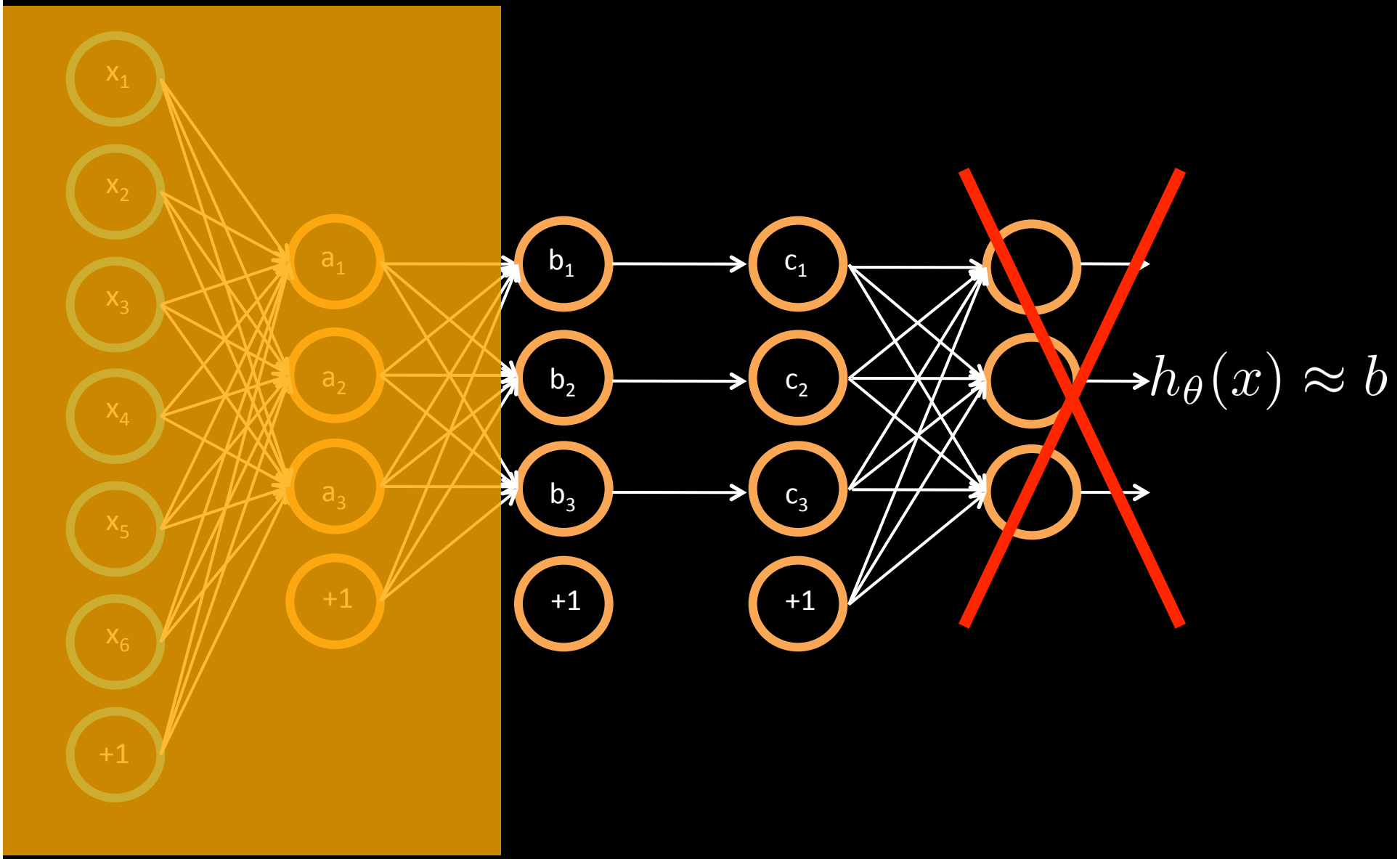


New representation for input.

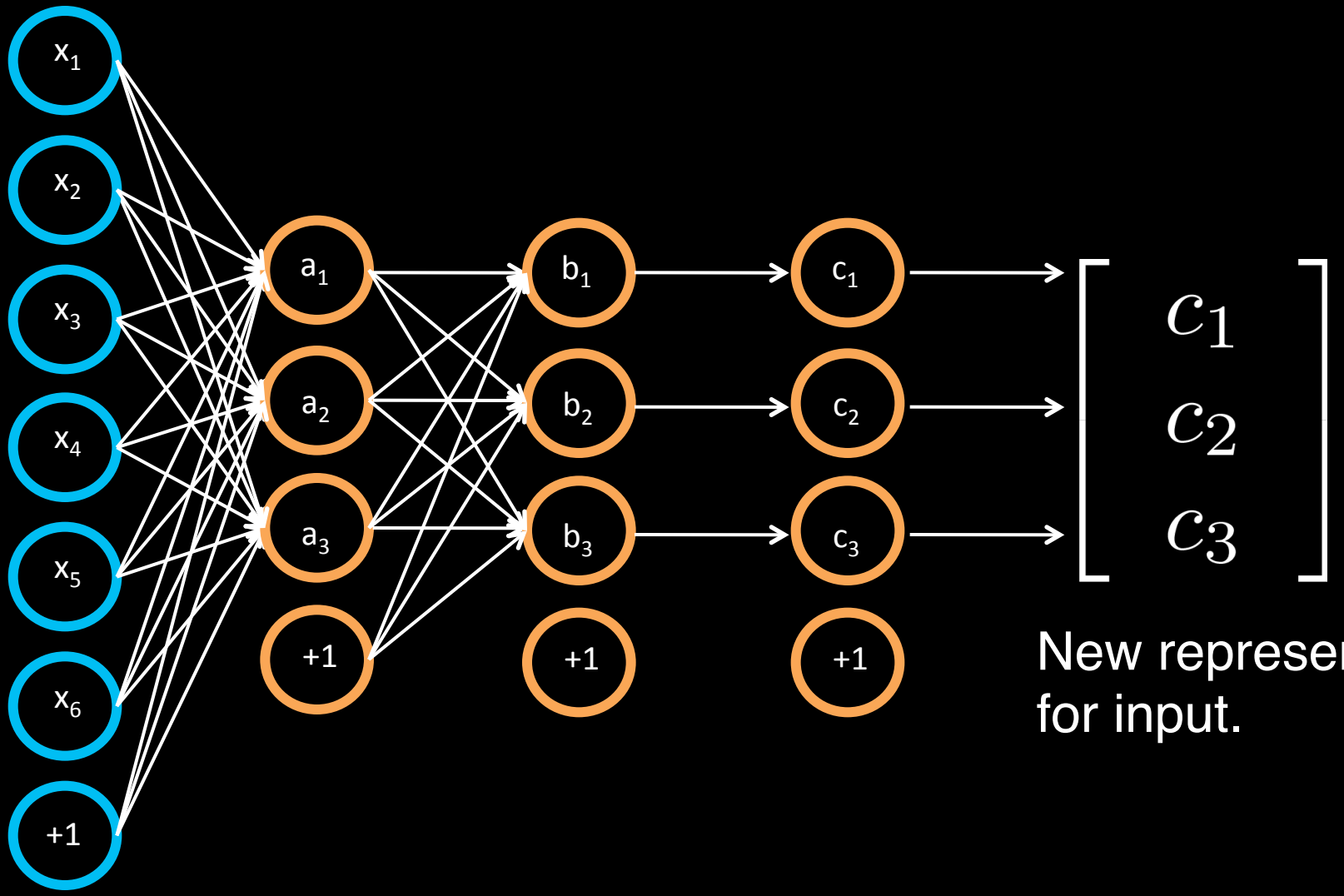
Unsupervised feature learning with a neural network



Unsupervised feature learning with a neural network



Unsupervised feature learning with a neural network



New representation for input.

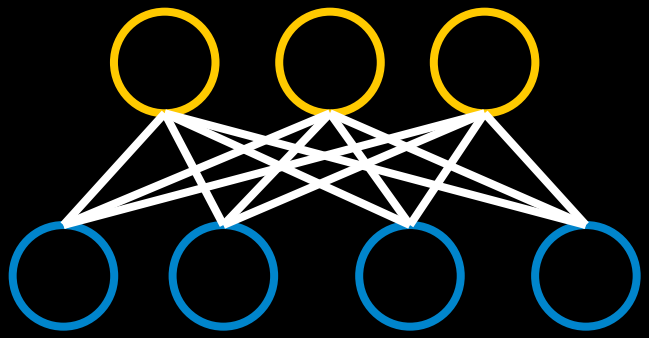
Use $[c_1, c_3, c_3]$ as representation to feed to learning algorithm.

Andrew Ng

Deep Belief Net

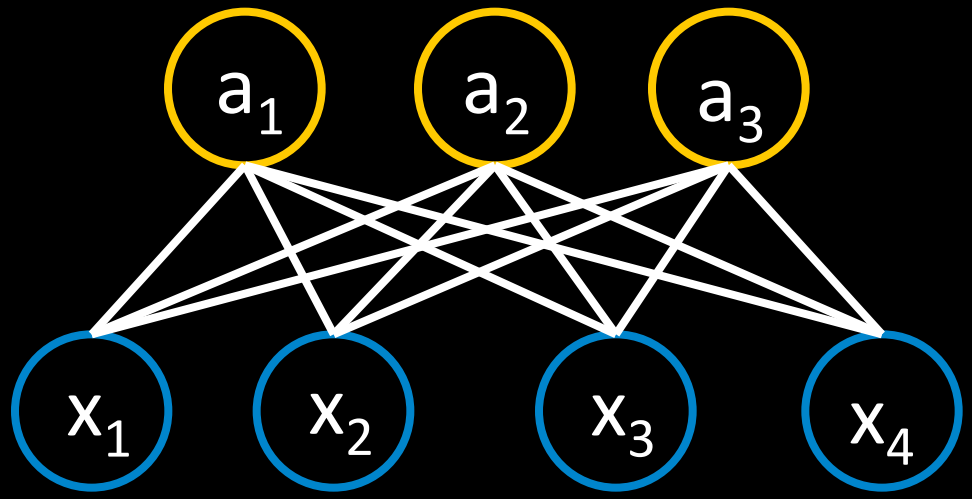
Deep Belief Net (DBN) is another algorithm for learning a feature hierarchy.

Building block: 2-layer graphical model (Restricted Boltzmann Machine).



Can then learn additional layers one at a time.

Restricted Boltzmann machine (RBM)



Layer 2. [a_1, a_2, a_3] (binary-valued)

Input [x_1, x_2, x_3, x_4]

MRF with joint distribution:

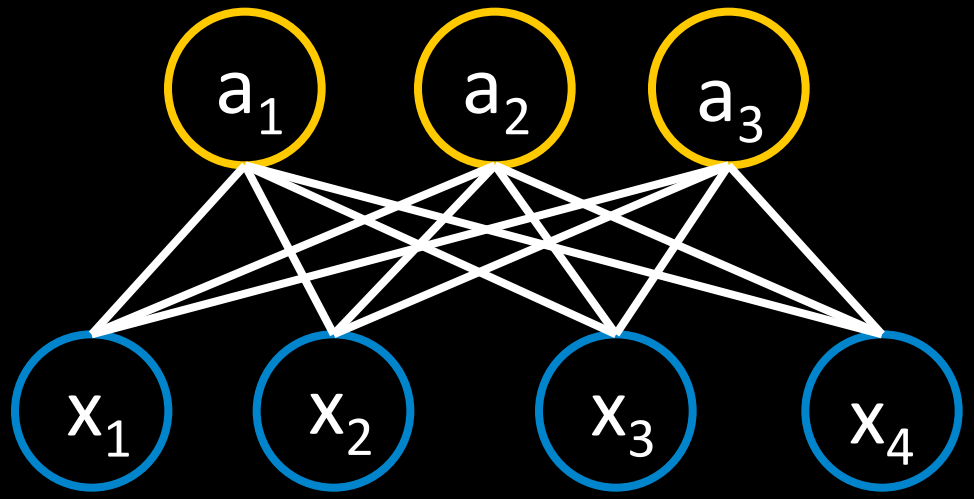
$$P(x, a) \propto \exp\left(-\sum_{i,j} x_i a_j W_{ij}\right)$$

Use Gibbs sampling for inference.

Given observed inputs x, want maximum likelihood estimation:

$$\max_W P(x) = \max_W \sum_a P(x, a)$$

Restricted Boltzmann machine (RBM)



Layer 2. [a_1, a_2, a_3] (binary-valued)

Input [x_1, x_2, x_3, x_4]

Gradient ascent on log P(x) :

$$\Delta W_{ij} = \alpha \left([x_i a_j]_{\text{obs}} - [x_i a_j]_{\text{prior}} \right)$$

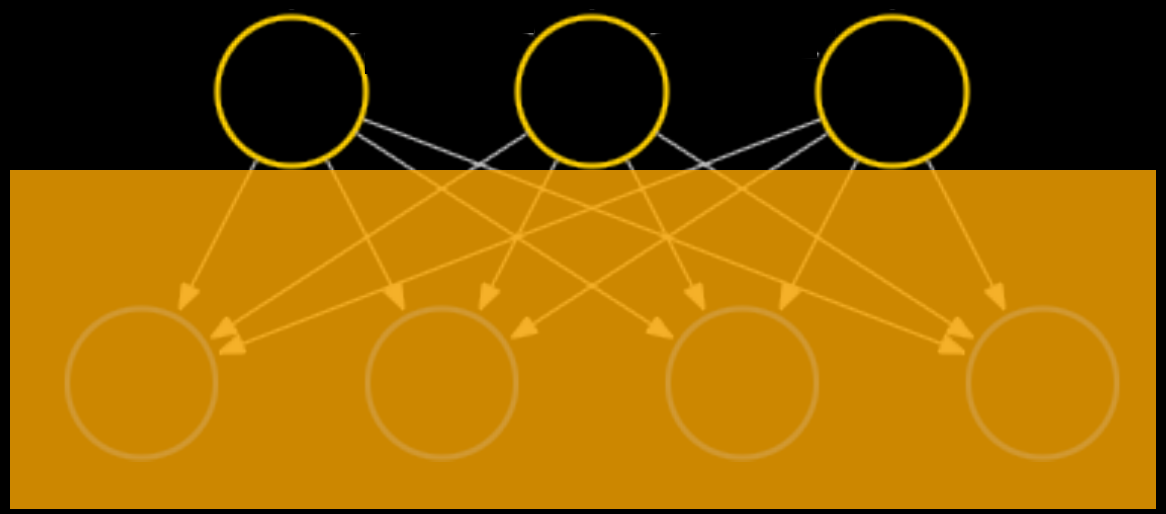
$[x_i a_j]_{obs}$ from fixing x to observed value, and sampling a from P(a|x).

$[x_i a_j]_{prior}$ from running Gibbs sampling to convergence.

Adding sparsity constraint on a_i's usually improves results.

Deep Belief Network

Similar to a sparse autoencoder in many ways.
Stack RBMs on top of each other to get DBN.

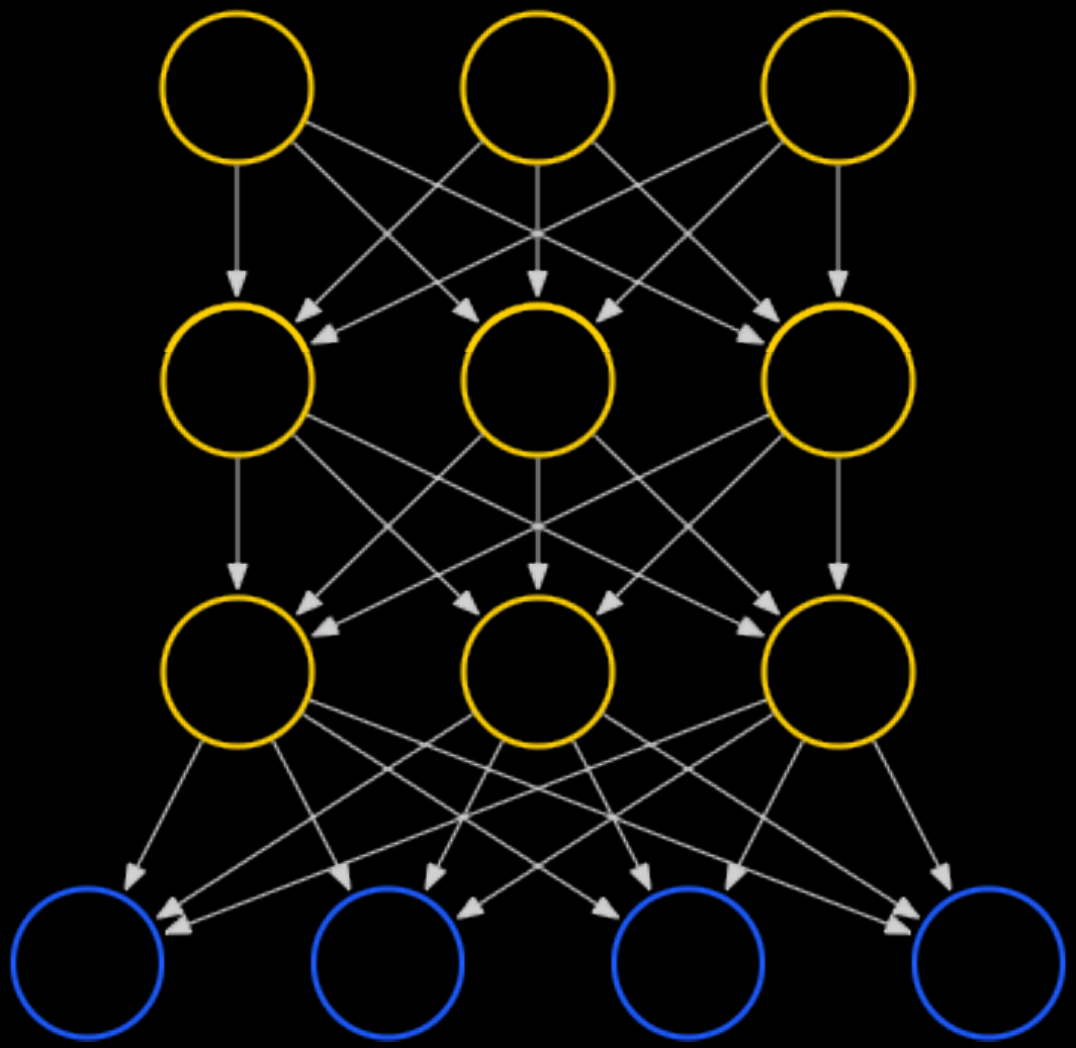


Layer 3. $[b_1, b_2, b_3]$

Layer 2. $[a_1, a_2, a_3]$

Input $[x_1, x_2, x_3, x_4]$

Deep Belief Network



Layer 4. [c_1, c_2, c_3]

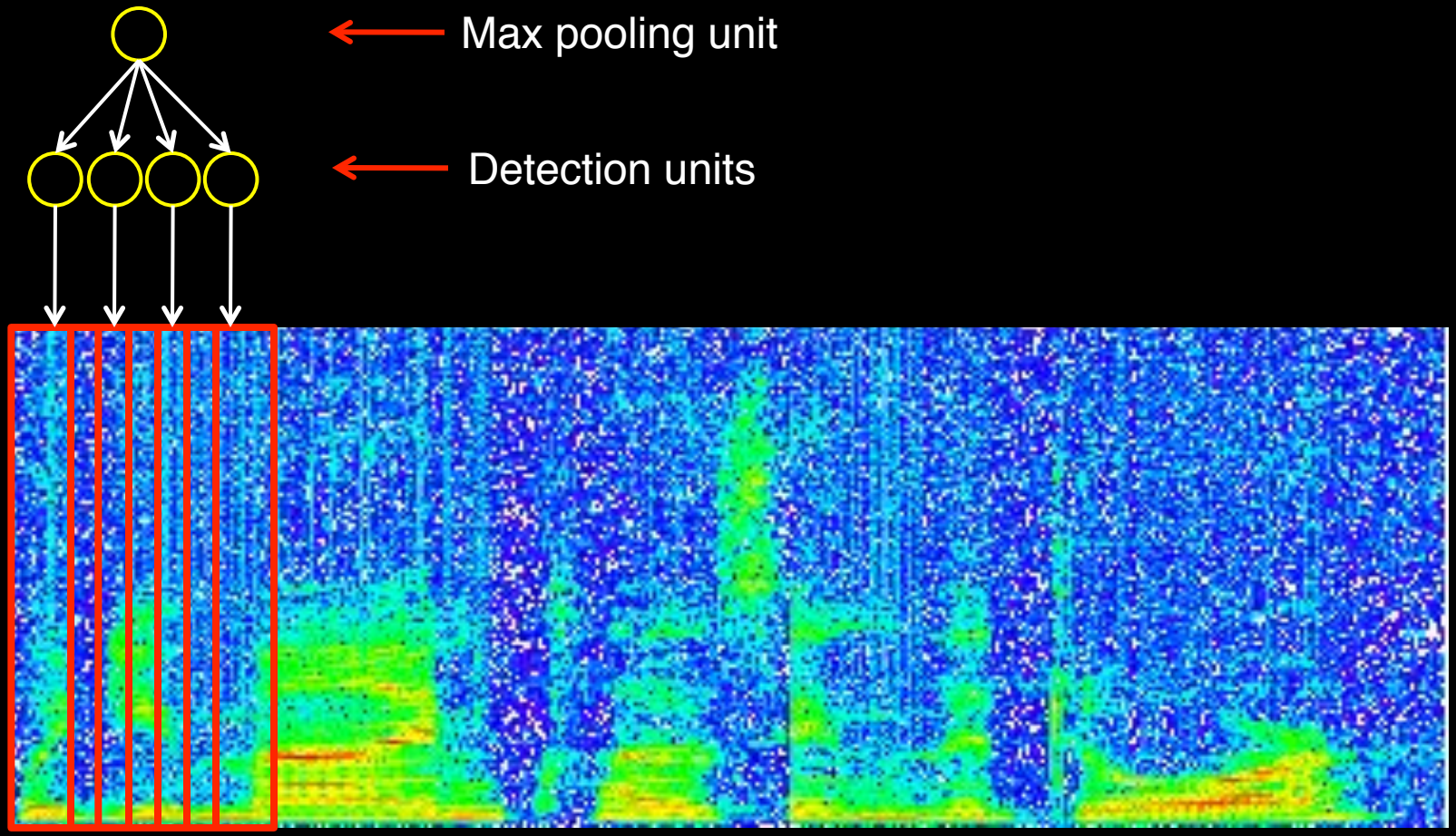
Layer 3. [b_1, b_2, b_3]

Layer 2. [a_1, a_2, a_3]

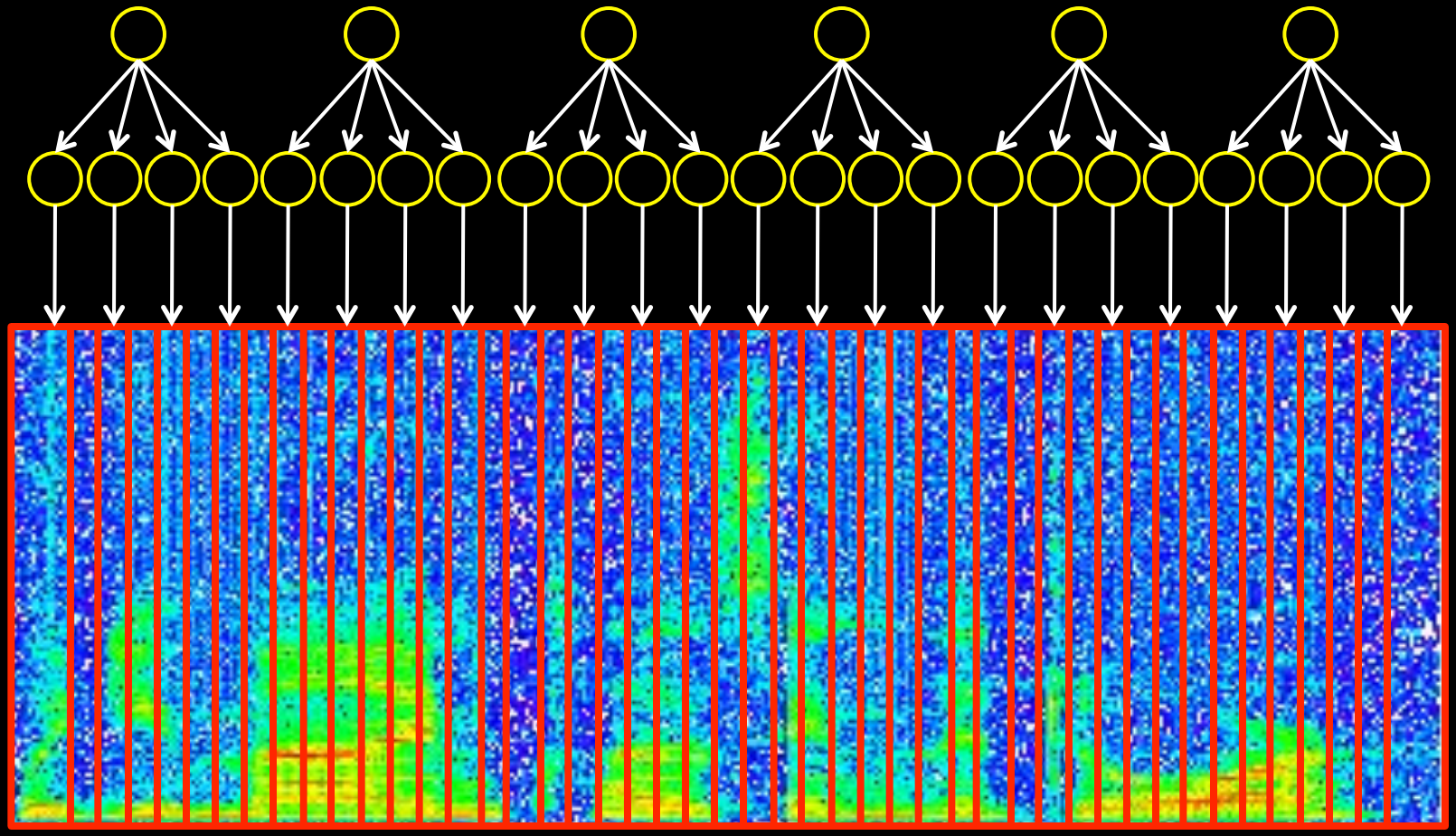
Input [x_1, x_2, x_3, x_4]

Andrew Ng

Convolutional DBN for audio

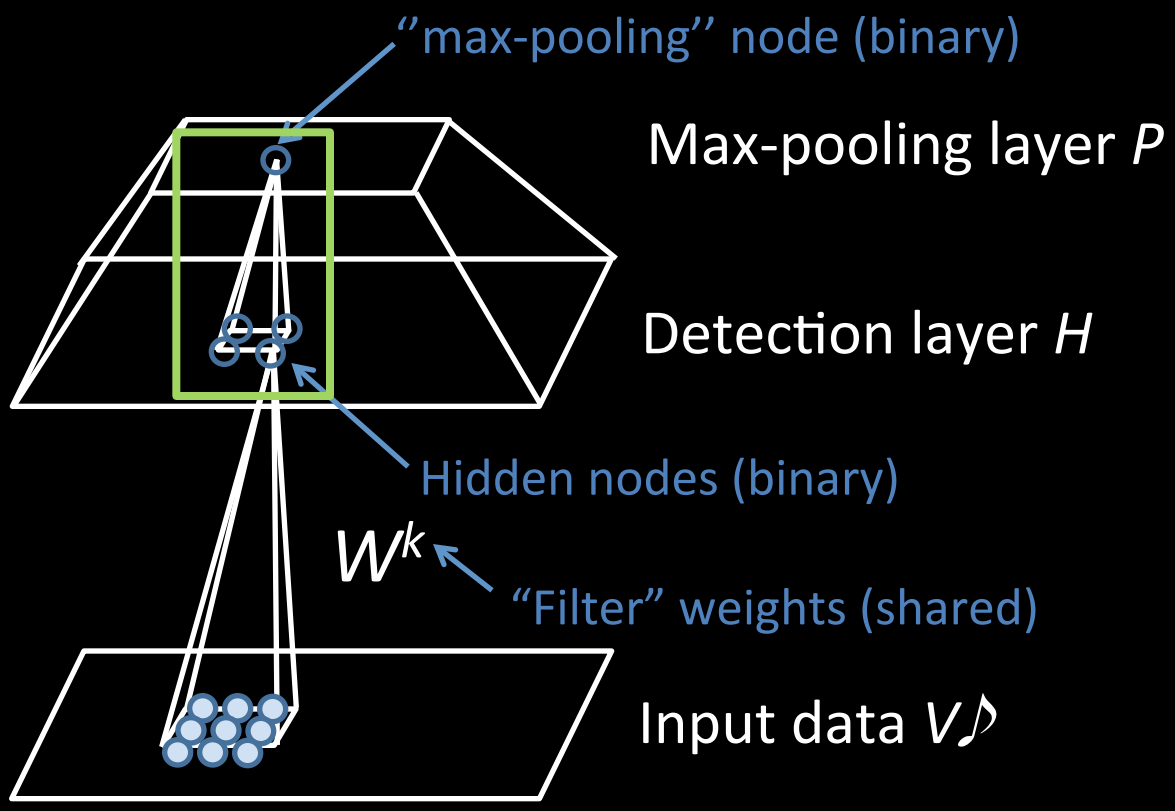


Convolutional DBN for audio

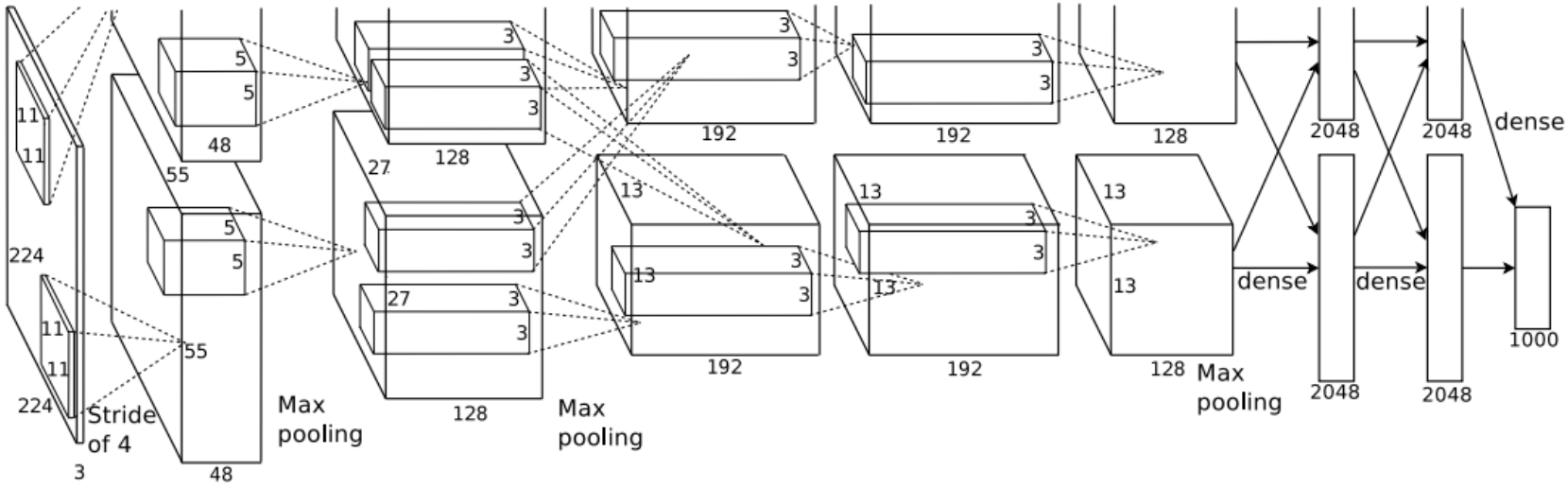


Spectrogram

Convolutional DBN for Images



Tutorial



[image classifier demo]