

Softmax classification

- $Y = \{1, 2, \dots, n\}$
- Given an input $x^{(i)}$, the softmax function which outputs the likelihood of Y being in class j is

$$p(Y = j \mid x^{(i)}) = \frac{e^{\langle \theta_j, x^{(i)} \rangle}}{\sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}}$$

- Softmax classification is also known as multinomial logistic regression.
- The term "softmax" comes from the fact that if we introduce a parameter c into the formula in the previous slide, i.e. if

$$p(Y = j \mid x^{(i)}) = \frac{e^{c\langle \theta_j, x^{(i)} \rangle}}{\sum_{k=1}^n e^{c\langle \theta_k, x^{(i)} \rangle}},$$

then the softmax function converges to a max function as $c \rightarrow \infty$.

Overparameterized

- Softmax classification for n classes does not actually require n weight vectors $\theta_1, \dots, \theta_n$:

$$\begin{aligned} p(Y = j \mid x^{(i)}) &= \frac{e^{\langle \theta_j, x^{(i)} \rangle}}{\sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}} \\ &= \frac{1}{1 + \sum_{k \neq j} e^{\langle \theta_k - \theta_j, x^{(i)} \rangle}} \\ &= \frac{1}{1 + \sum_{k \neq j} e^{-\langle \tilde{\theta}_k, x^{(i)} \rangle}}, \end{aligned}$$

where we denote $\tilde{\theta}_k$ as $\theta_j - \theta_k$. Hence, only $n - 1$ weight vectors are needed.

- In fact, in the 2-class case, this is precisely the formula for the sigmoid function, as thus softmax classification is a generalization of logistic regression:

$$p(Y = 1 \mid x^{(i)}) = \frac{e^{\langle \theta_1, x^{(i)} \rangle}}{e^{\langle \theta_1, x^{(i)} \rangle} + e^{\langle \theta_2, x^{(i)} \rangle}} = \frac{1}{1 + e^{-\langle \theta_1 - \theta_2, x^{(i)} \rangle}}$$

$$\begin{aligned} p(Y = 2 \mid x^{(i)}) &= \frac{e^{\langle \theta_2, x^{(i)} \rangle}}{e^{\langle \theta_1, x^{(i)} \rangle} + e^{\langle \theta_2, x^{(i)} \rangle}} = \frac{1}{1 + e^{-\langle \theta_2 - \theta_1, x^{(i)} \rangle}} \\ &= 1 - \frac{1}{1 + e^{-\langle \theta_1 - \theta_2, x^{(i)} \rangle}} \end{aligned}$$

Log-likelihood

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^m \log p(Y = y^{(i)} \mid x^{(i)}) \\ &= \sum_{i=1}^m \log \frac{e^{\langle \theta_{y^{(i)}}, x^{(i)} \rangle}}{\sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}} \\ &= \sum_{i=1}^m \langle \theta_{y^{(i)}}, x^{(i)} \rangle - \log \sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}\end{aligned}$$

Gradient of softmax log-likelihood

$$\begin{aligned}\nabla_{\theta_p} \ell(\theta) &= \sum_{i=1}^m \delta_{p y^{(i)}} x^{(i)} - \nabla_{\theta_p} \log \sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle} \\&= \sum_{i=1}^m \delta_{p y^{(i)}} x^{(i)} - \frac{\nabla_{\theta_p} \sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}}{\sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}} \\&= \sum_{i=1}^m \delta_{p y^{(i)}} x^{(i)} - \frac{e^{\langle \theta_p, x^{(i)} \rangle} x^{(i)}}{\sum_{k=1}^n e^{\langle \theta_k, x^{(i)} \rangle}} \\&= \sum_{i=1}^m \left(\delta_{p y^{(i)}} - p(Y = p \mid x^{(i)}) \right) x^{(i)}\end{aligned}$$

DEEP LEARNING



DEFINITIONS

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



MACHINE LEARNING

Machine learning begins to flourish.



DEEP LEARNING

Deep learning breakthroughs drive AI boom.



Milestones in AI and Deep Learning

- 1943: First artificial neural networks (McCulloch, Pitts)
- 1956: First conference defining “AI” in Dartmouth (Shannon, Minsky etc.)
- 1986: Training multi-layer neural nets using backpropagation (Hinton, Rumelhart, Williams)
- 1989: Convolutional neural networks (LeCun)
- 1997: Deep blue beats Kasporov in chess
- 2009: Big bang of deep learning, use of GPUs for accelerated training
- 2012: Numerous competitions won with superhuman performance in computer vision using deep CNNs (Krizhevsky, Ciresan etc.)
- 2016: Alphago beats Lee Sedol in Go

THE EXPANDING UNIVERSE OF MODERN AI

"THE BIG BANG"

Big Data
GPU
Algorithms

RESEARCH

Berkeley
Carnegie Mellon University
DEEPMIND
Massachusetts Institute of Technology
NYU
UNIVERSITY OF OXFORD
UNIVERSITY OF TORONTO

CORE TECHNOLOGY / FRAMEWORKS

Preferred Networks
facebook. torch
Université de Montréal
theano
Google TensorFlow
Berkeley Caffe
Microsoft CNTK
UNIVERSITY OF OXFORD cuDNN
NVIDIA cuDNN

AI-as-a-PLATFORM

amazon webservices

IBM Watson

Google

Microsoft Azure

START-UPS

api.ai

Personal Assistants
conversational interface

BLUE RIVER TECHNOLOGY

Agriculture
crop-yield optimization

clarifai

Tech
visual recognition platform

deep genomics

Genomics
genetic interpretation

drive.ai

Automotive
AI-as-a-service

SADAKO

Waste Management
sorting robots

Morpho

Tech
computer vision

Orbital Insight

Geospatial
predictions from images

nervana

Tech
AI-as-a-service

MetaMind

eCommerce & Medical
recommendation engines

SocialEyes*

Medical
diabetic retinopathy

HOW ARE YOU?

Orbital Insight

Education
teaching robots

1,000+ AI START-UPS

\$5B IN FUNDING

Source: Venture Scanner

INDUSTRY LEADERS

Ford

target

SIEMEN

GE

TESL

gsk

TOYO

Audi

Baidu

THE HOME DEPOT

Bloomberg

UBER

MASSACHUSETTS GENERAL HOSPITAL

Mercedes-Benz

charles SCHWAB

VOLVO

MERCK

Walmart

Pinterest

YAHOO

ebay

Schlumberger

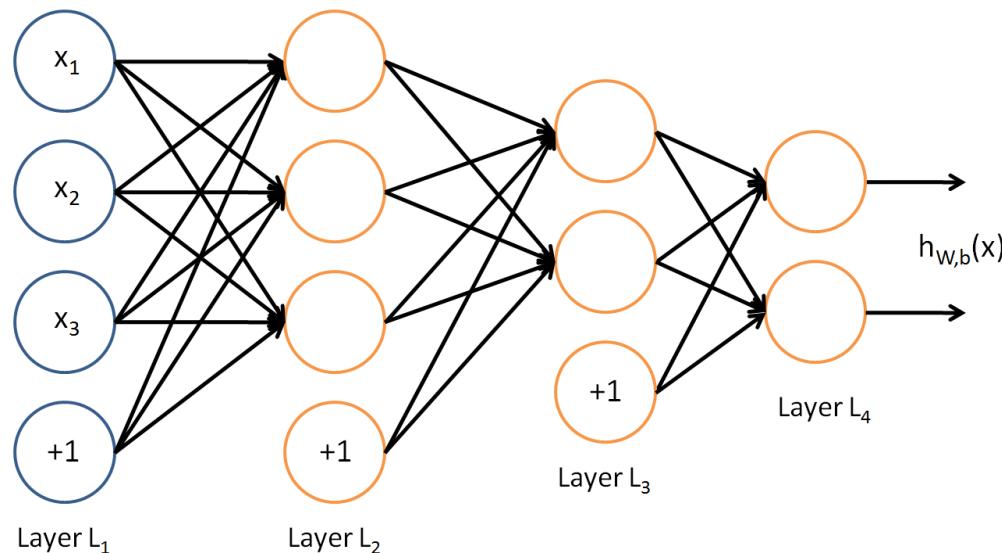
FANUC ROBOTICS

Yandex

yelp

WHAT IS DEEP LEARNING?

Biologically-inspired multilayer neural networks

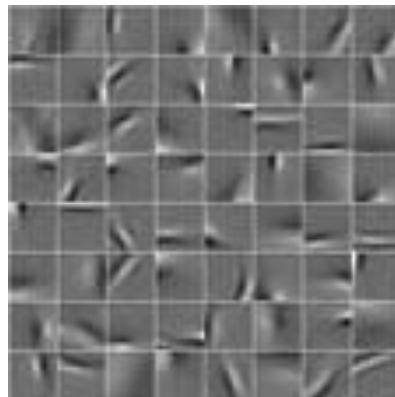


Both supervised and unsupervised

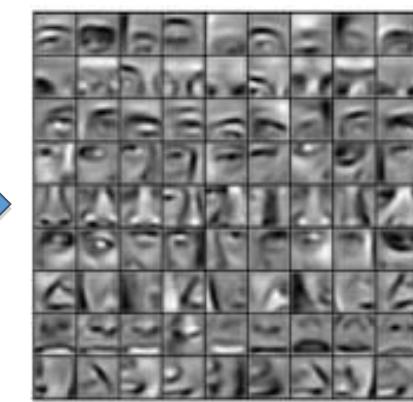
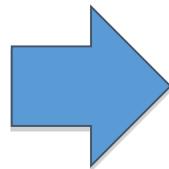


WHAT IS DEEP LEARNING?

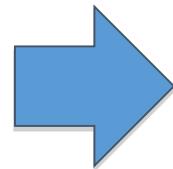
Example. Face recognition (Facebook)



Edges



Eyes, Noses, Mouths



Faces

Deeper layers learn higher-order features



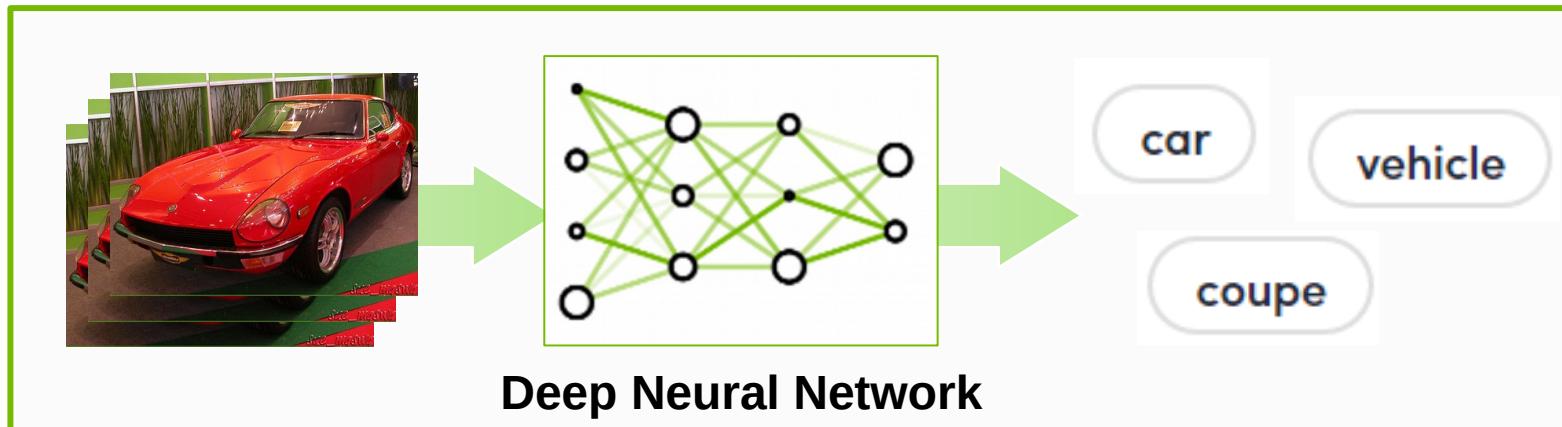
A NEW COMPUTING MODEL

Algorithms that Learn from Examples



Traditional Approach

- Requires domain experts
- Time consuming
- Error prone
- Not scalable to new problems



Deep Learning Approach

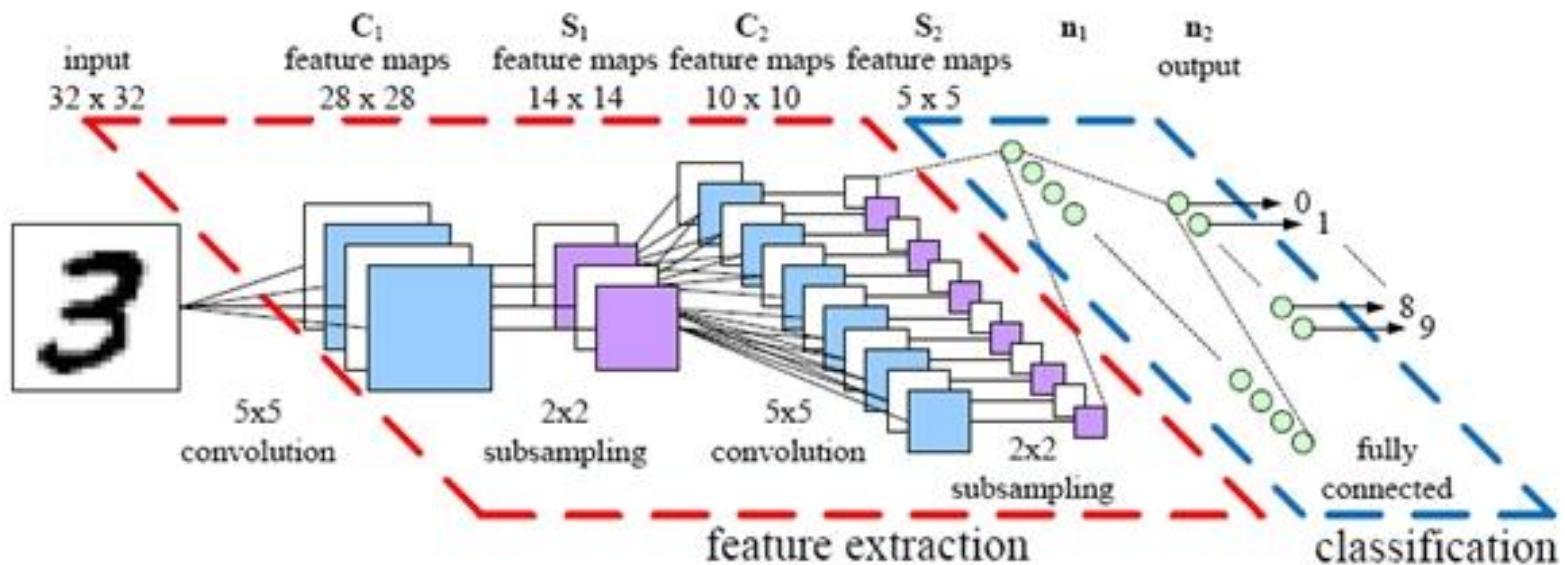
- ✓ Learn from data
- ✓ Easily to extend
- ✓ Speedup with GPUs



APPLICATIONS



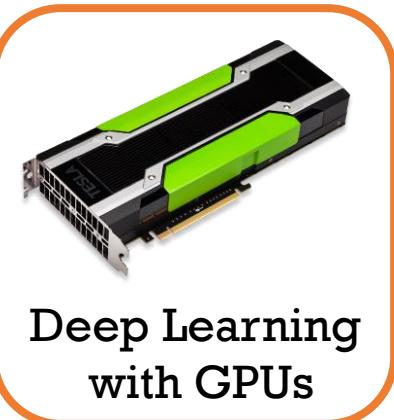
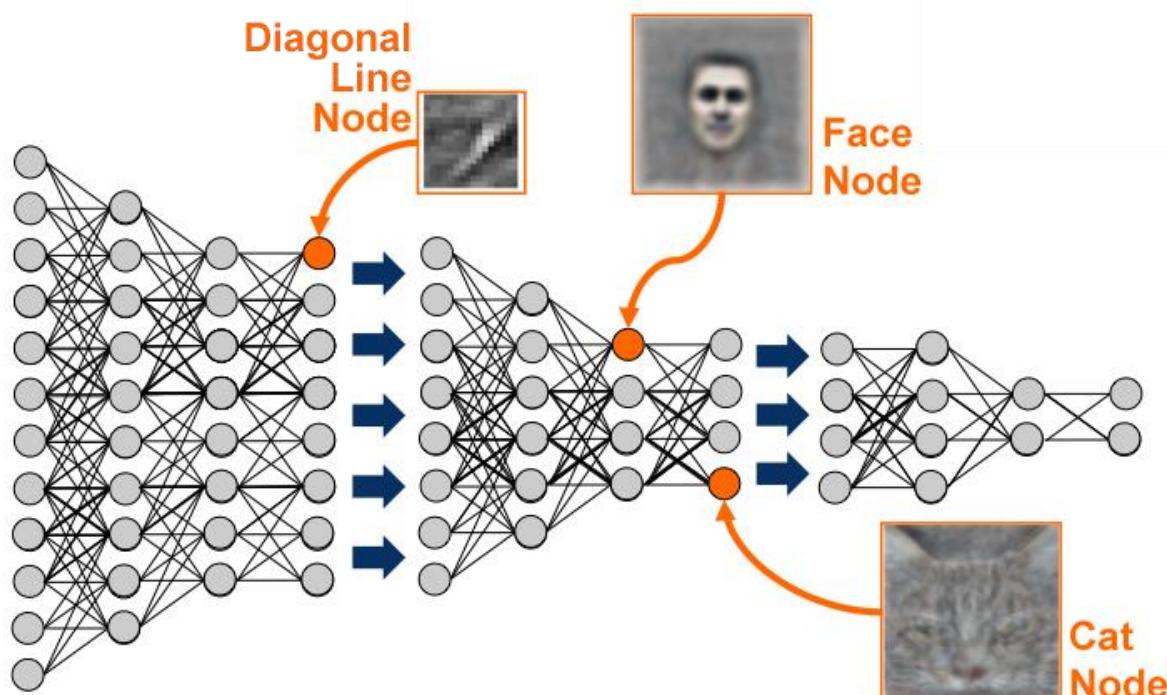
HANDWRITING RECOGNITION



0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9



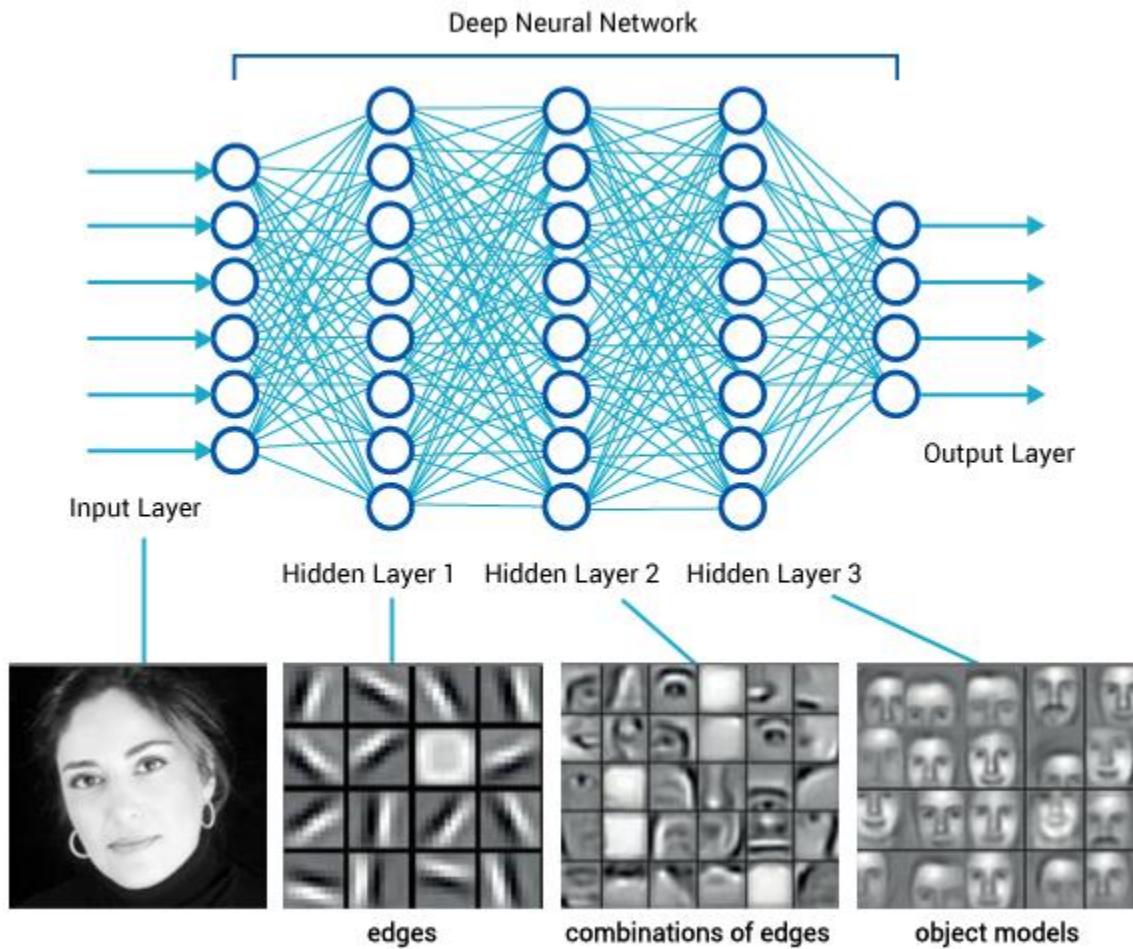
GOOGLE CAT VIDEOS



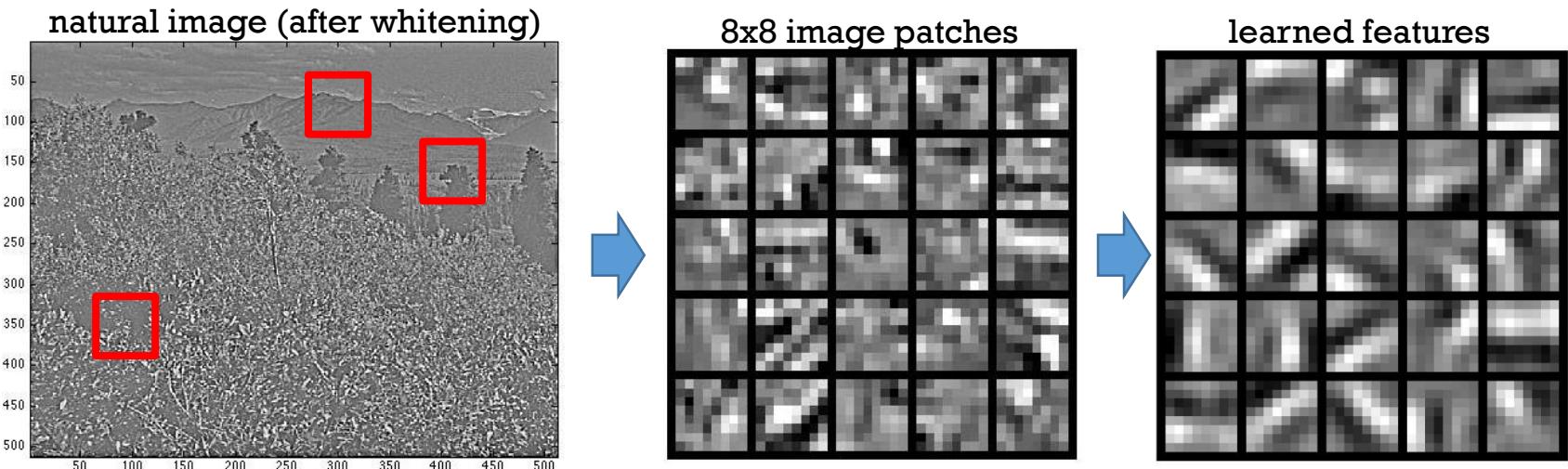
Deep Learning
with GPUs



FACE RECOGNITION



NATURAL IMAGES



Edge features similar to those from neuroscience experiments
(see Hubel & Wiesel Cat Experiment, 1959).



SPEECH TRANSLATION



From Hidden Markov Models to Recurrent Neural Networks

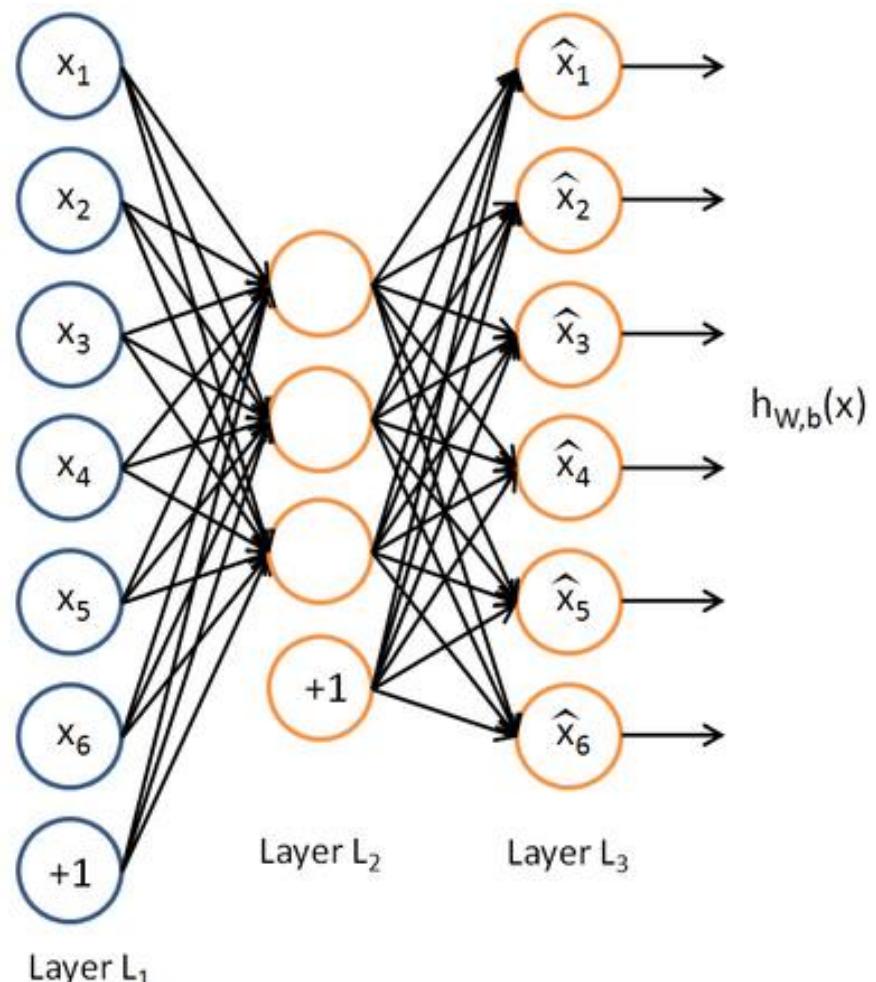


AUTOENCODERS

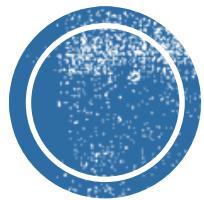
Training a multilayer neural network to reconstruct the input from a **reduced representation**.

Strategies for Dimensionality Reduction

- Few hidden neurons
- Sparse activations



5 min break

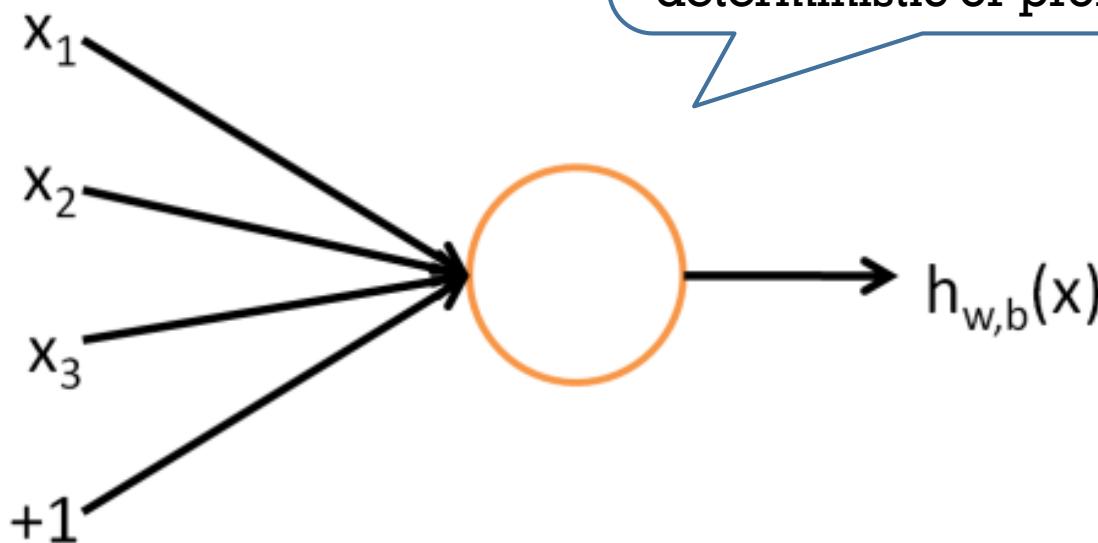


FEEDFORWARD NETWORKS



NEURON

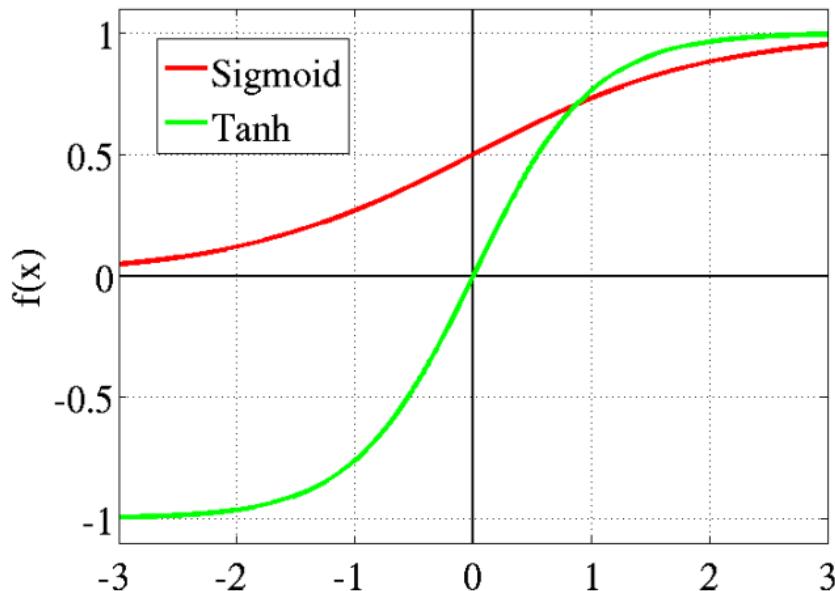
Perceptron.



$$h_{w,b}(x) = f(w^\top x) = f\left(\sum_{i=1}^d w_i x_i + b\right)$$

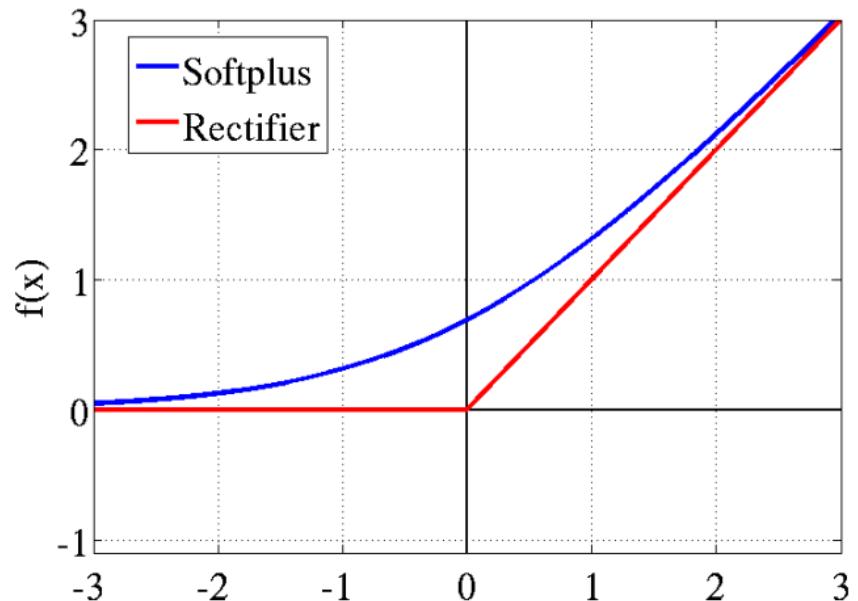


ACTIVATION FUNCTIONS



$$\text{sigmoid } f(z) = \frac{1}{1+e^{-z}}$$

$$\tanh f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

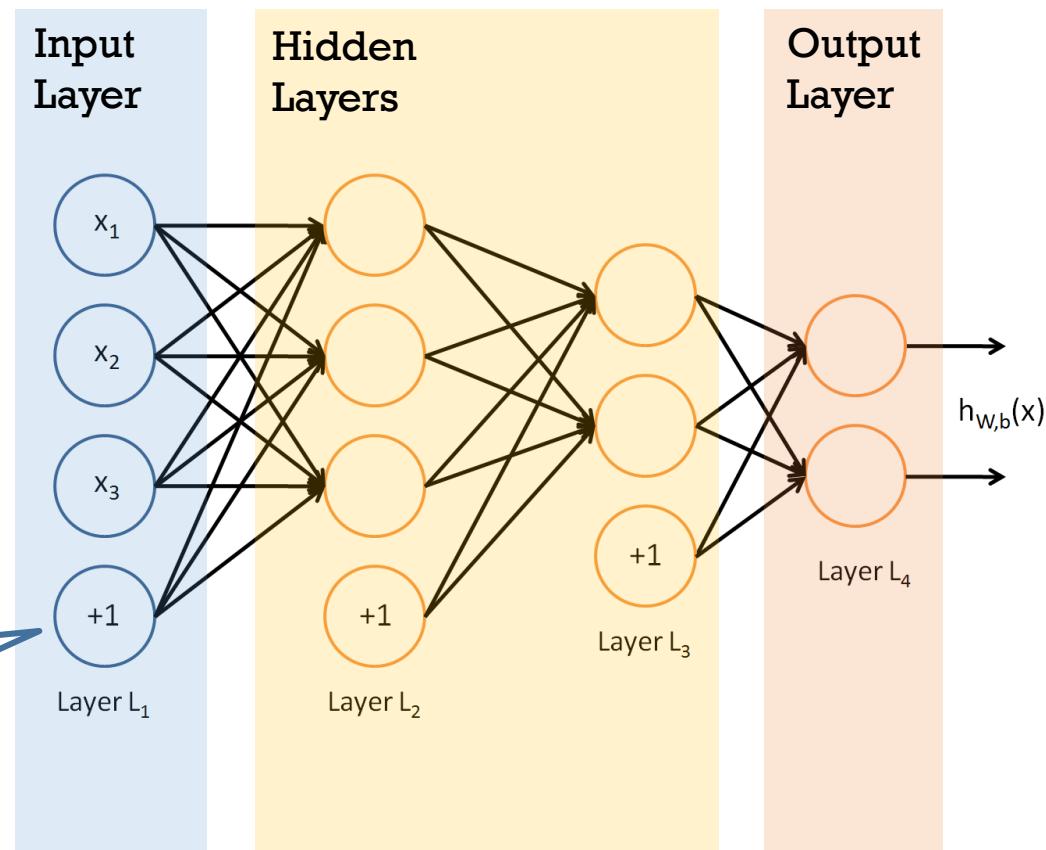


$$\text{softplus } f(z) = \ln(1 + e^{-z})$$

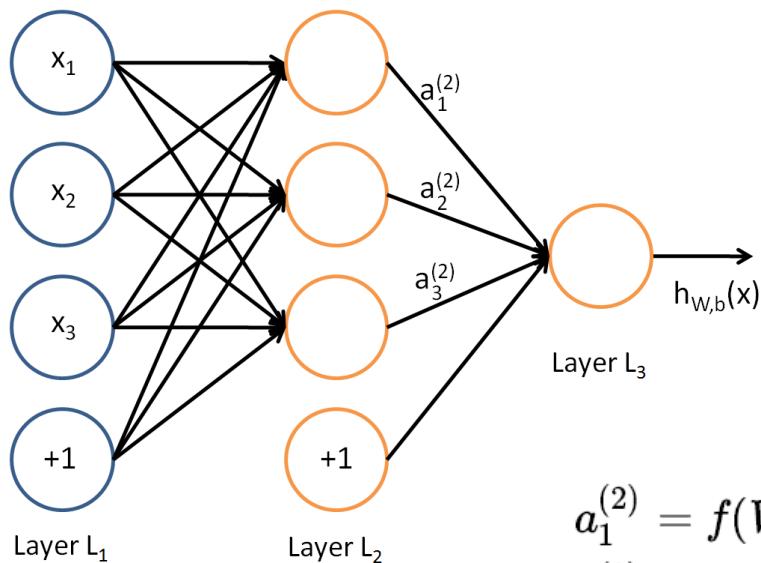
$$\begin{aligned} &\text{rectified} \\ &\text{linear unit} \\ &(\text{ReLU}) \end{aligned}$$
$$f(z) = \max(0, z)$$



MULTI-LAYER NEURAL NETWORK



MULTI-LAYER NEURAL NETWORK



$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$

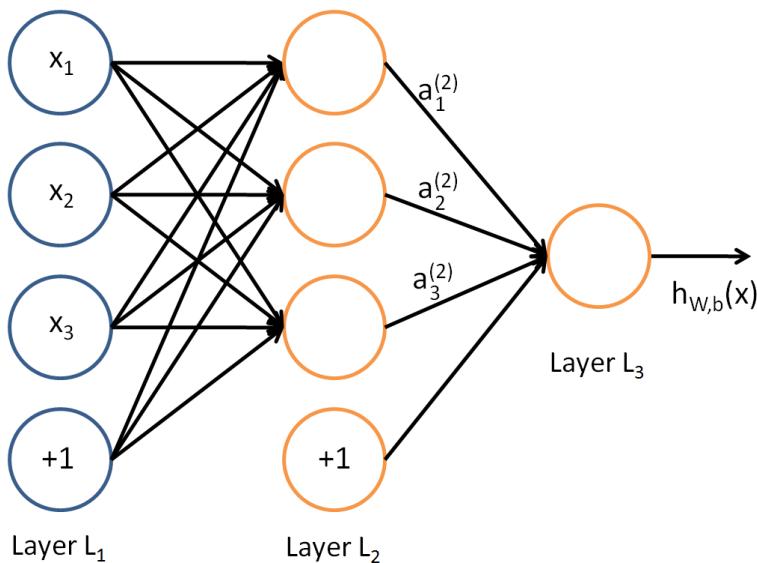
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$

$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$



MULTI-LAYER NEURAL NETWORK



Forward Propagation.

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

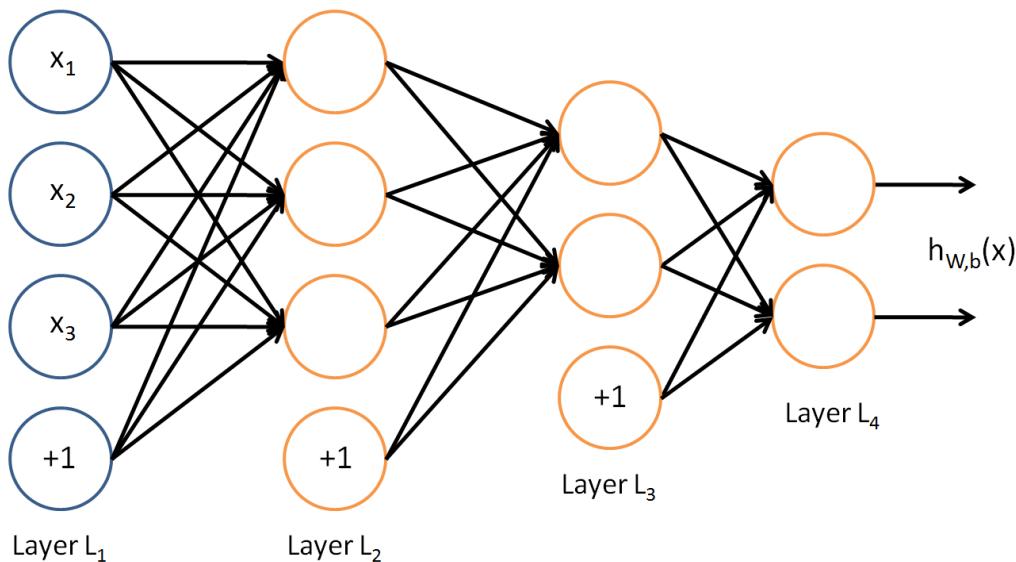
$$a^{(2)} = f(z^{(2)})$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

$$h_{W,b}(x) = a^{(3)} = f(z^{(3)})$$



MULTI-LAYER NEURAL NETWORK



Feedforward Neural Network.

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)}$$
$$a^{(l+1)} = f(z^{(l+1)})$$

Activation

Neural Network Architecture.

Arrangement of neurons, e.g. number of neurons in each layer.

