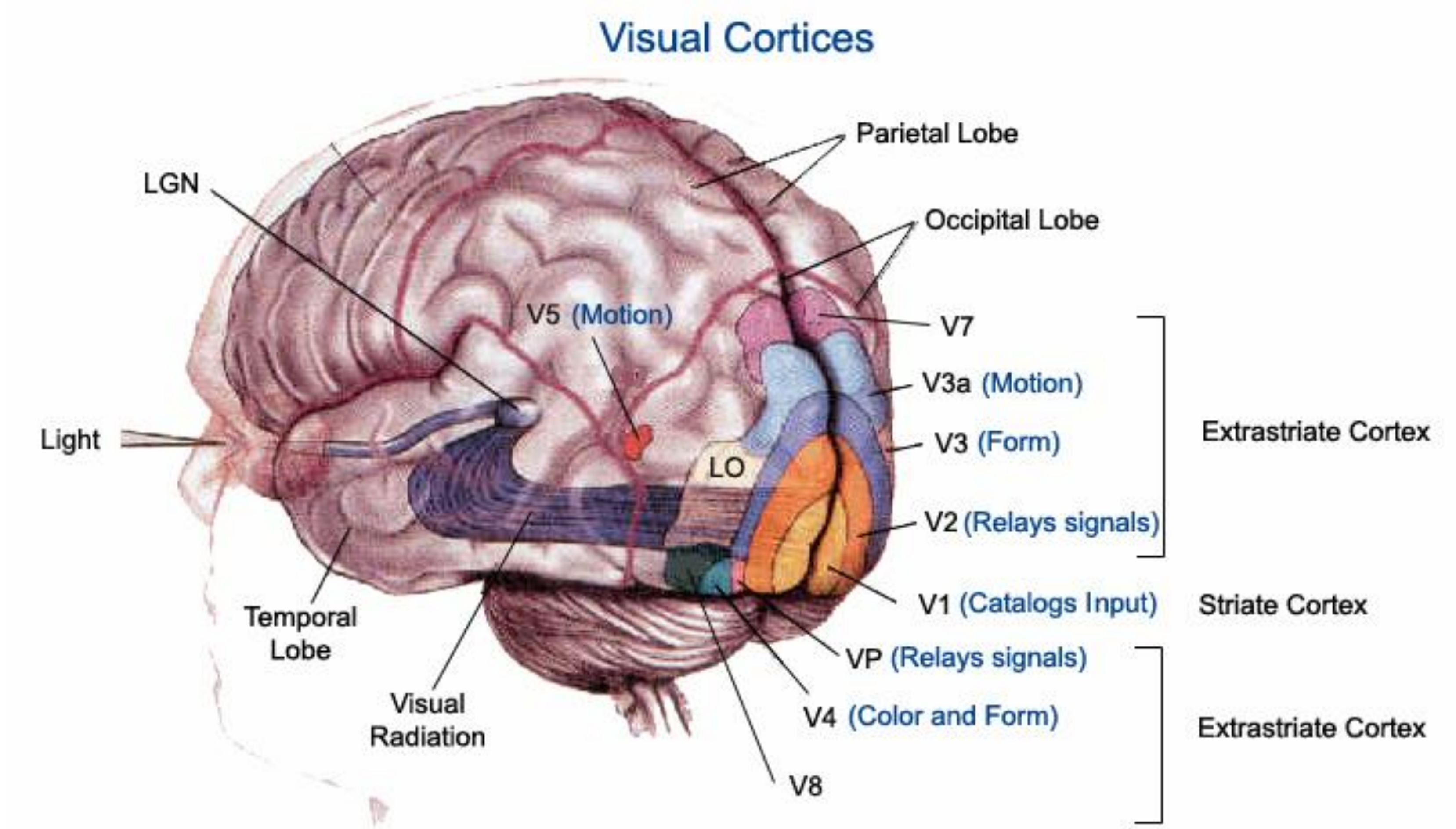
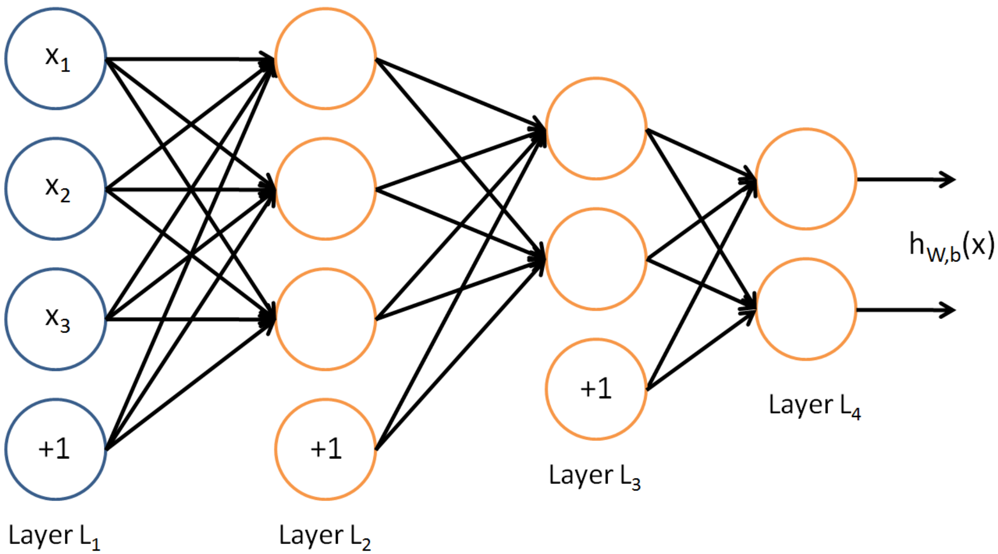
DEEP LEARNING

WHAT IS DEEP LEARNING?



WHAT IS DEEP LEARNING?

Biologically-inspired multilayer neural networks

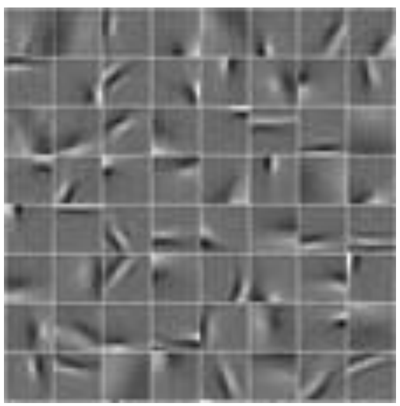


Both supervised and unsupervised

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science 313.5786 (2006): 504-507.

WHAT IS DEEP LEARNING?

Example. Face recognition (Facebook)



Edges

Eyes, Noses, Mouths

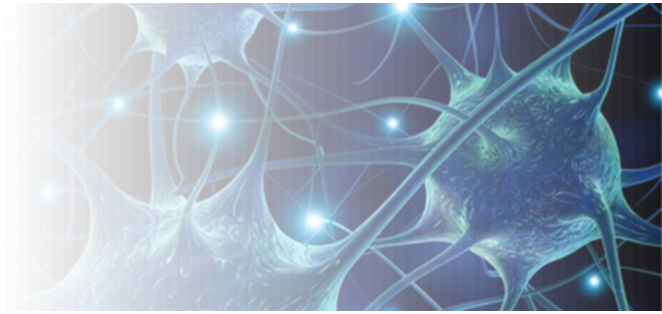
Faces

Deeper layers learn higher-order features

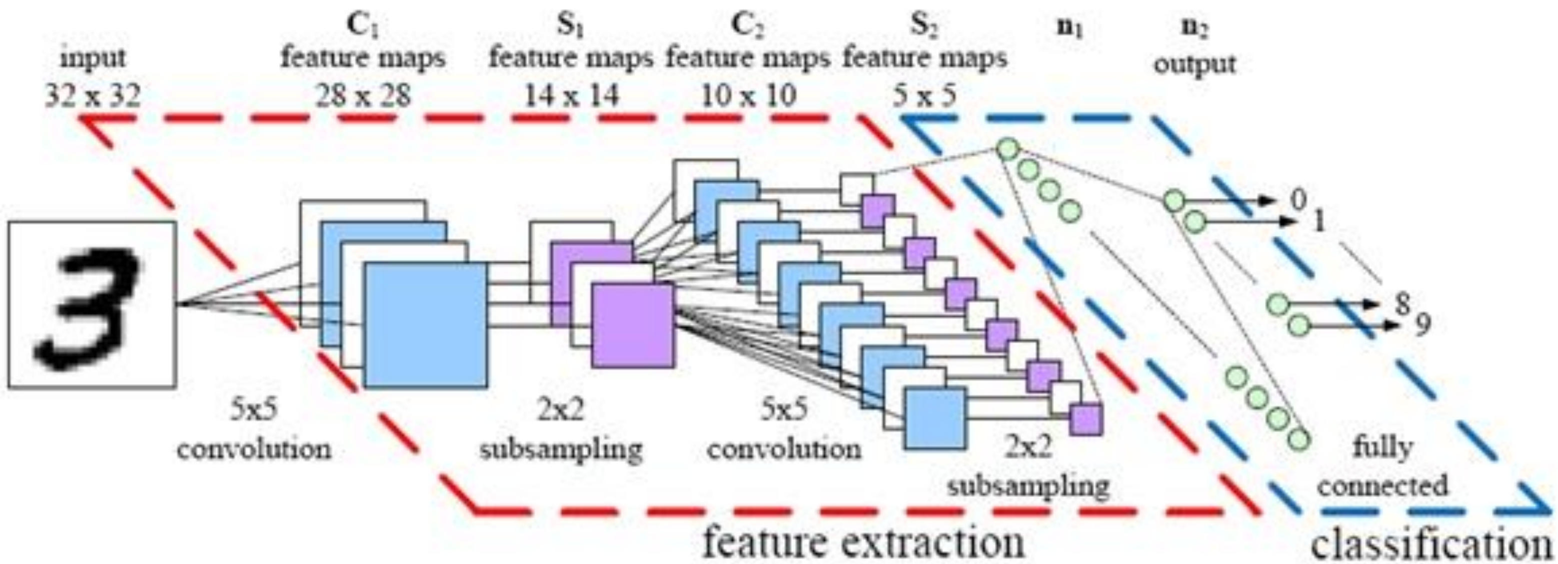
LISA lab, “Deep Learning Tutorials.” (http://www.deeplearning.net/tutorial/) accessed 2014.

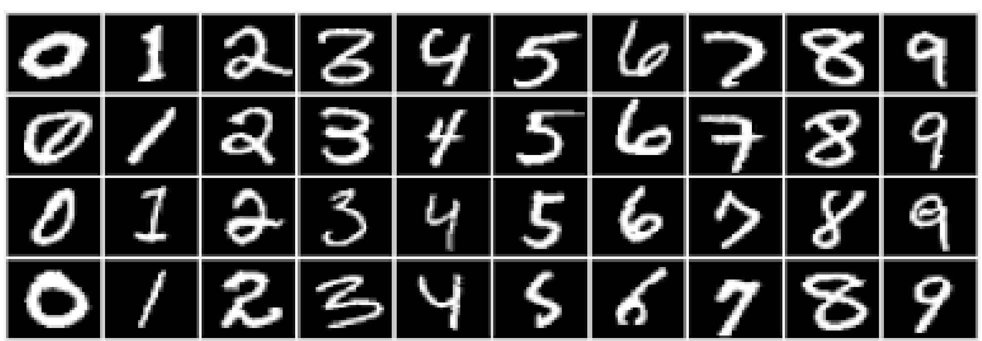
Lee, Honglak, et al. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009.

APPLICATIONS

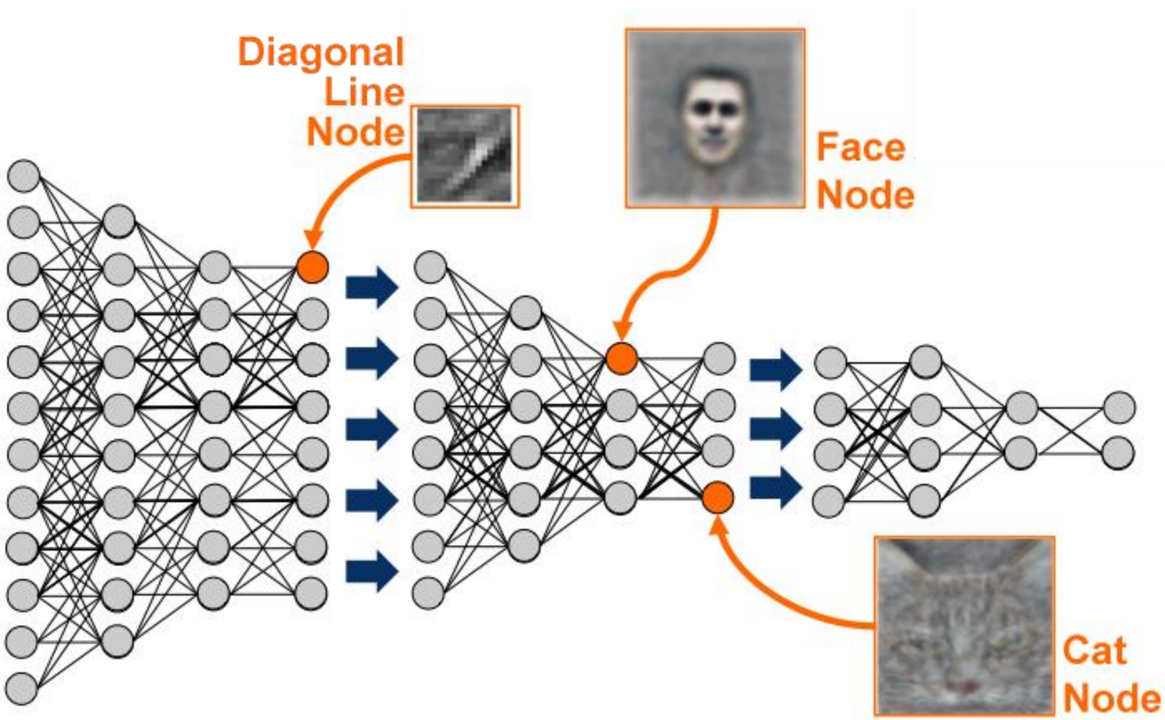


HANDWRITING RECOGNITION



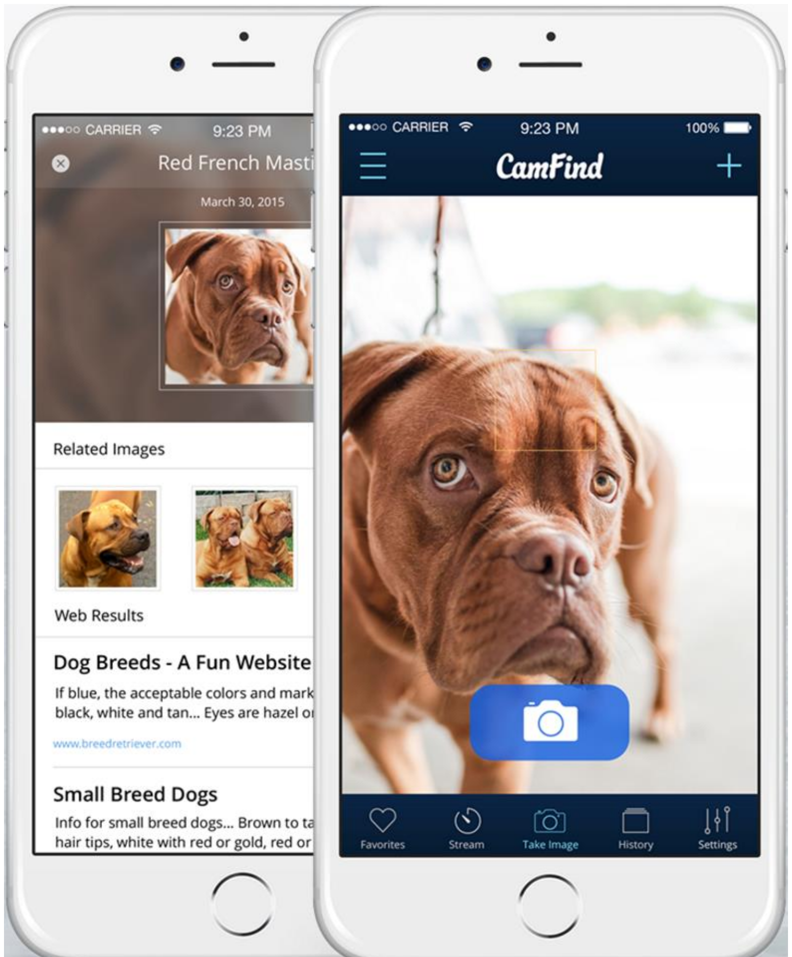


GOOGLE CAT VIDEOS



Deep Learning with GPUs

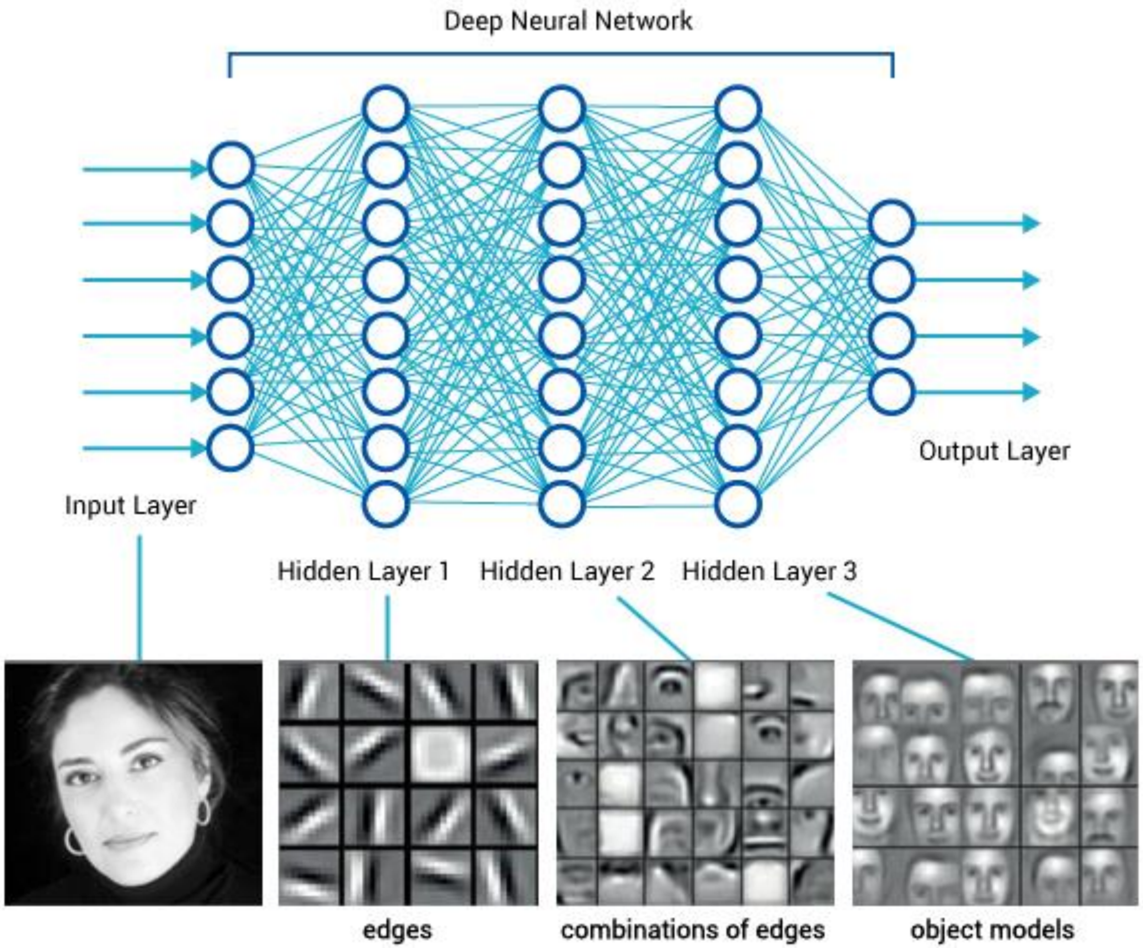
IMAGE RECOGNITION



CamFind

Visual Search Engine
(available on iOS, Android)

FACE RECOGNITION



SPEECH TRANSLATION

From Hidden Markov Models to Recurrent Neural Networks

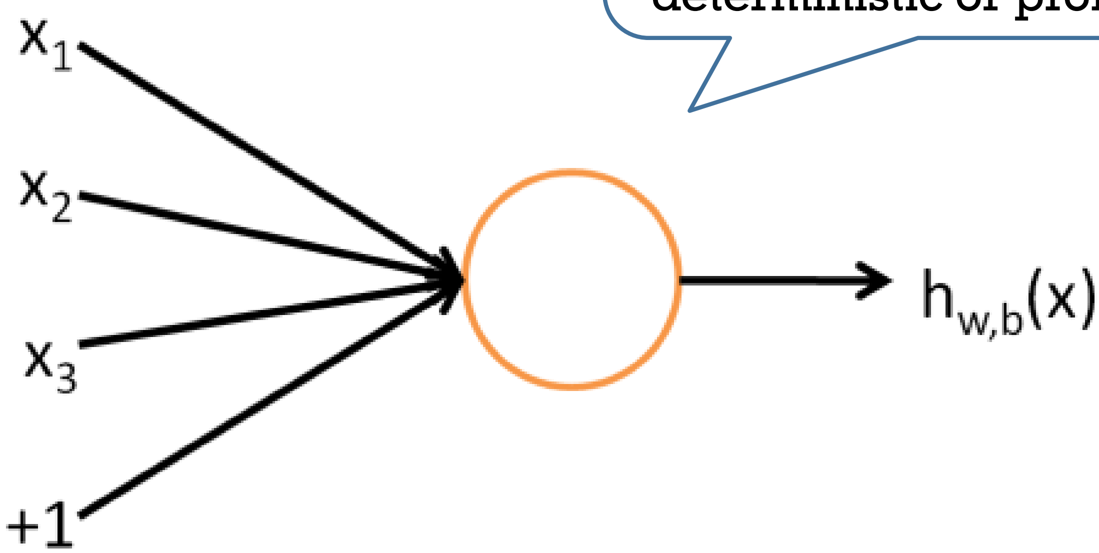
https://www.youtube.com/watch?v=mWTySUGXR2k

FEEDFORWARD NETWORKS

NEURON

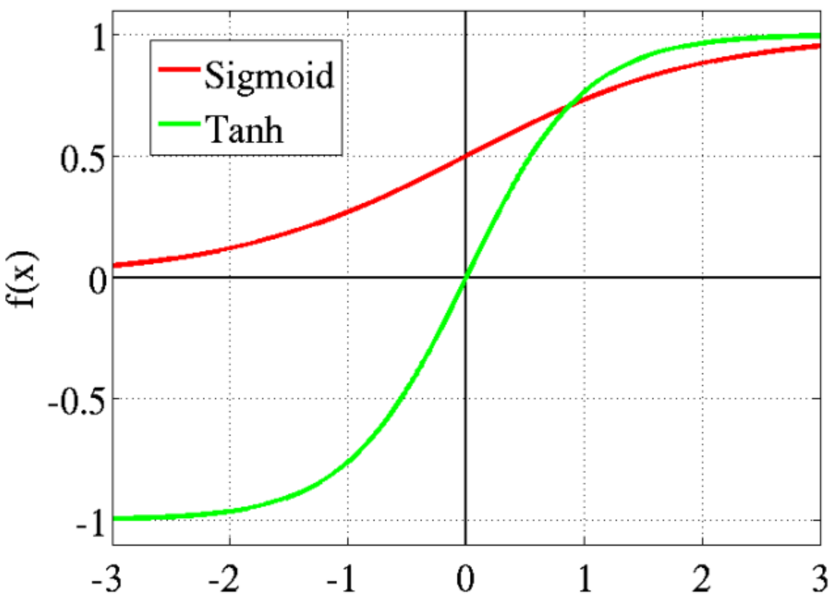
Perceptron.

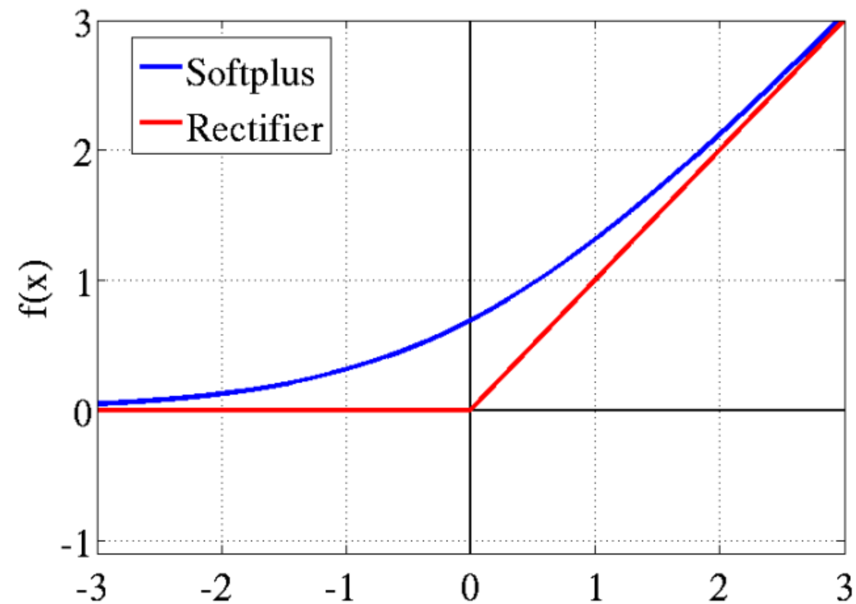
Depending on function f,
neurons can be:
real-valued or binary-valued;
deterministic or probabilistic.



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

ACTIVATION FUNCTIONS





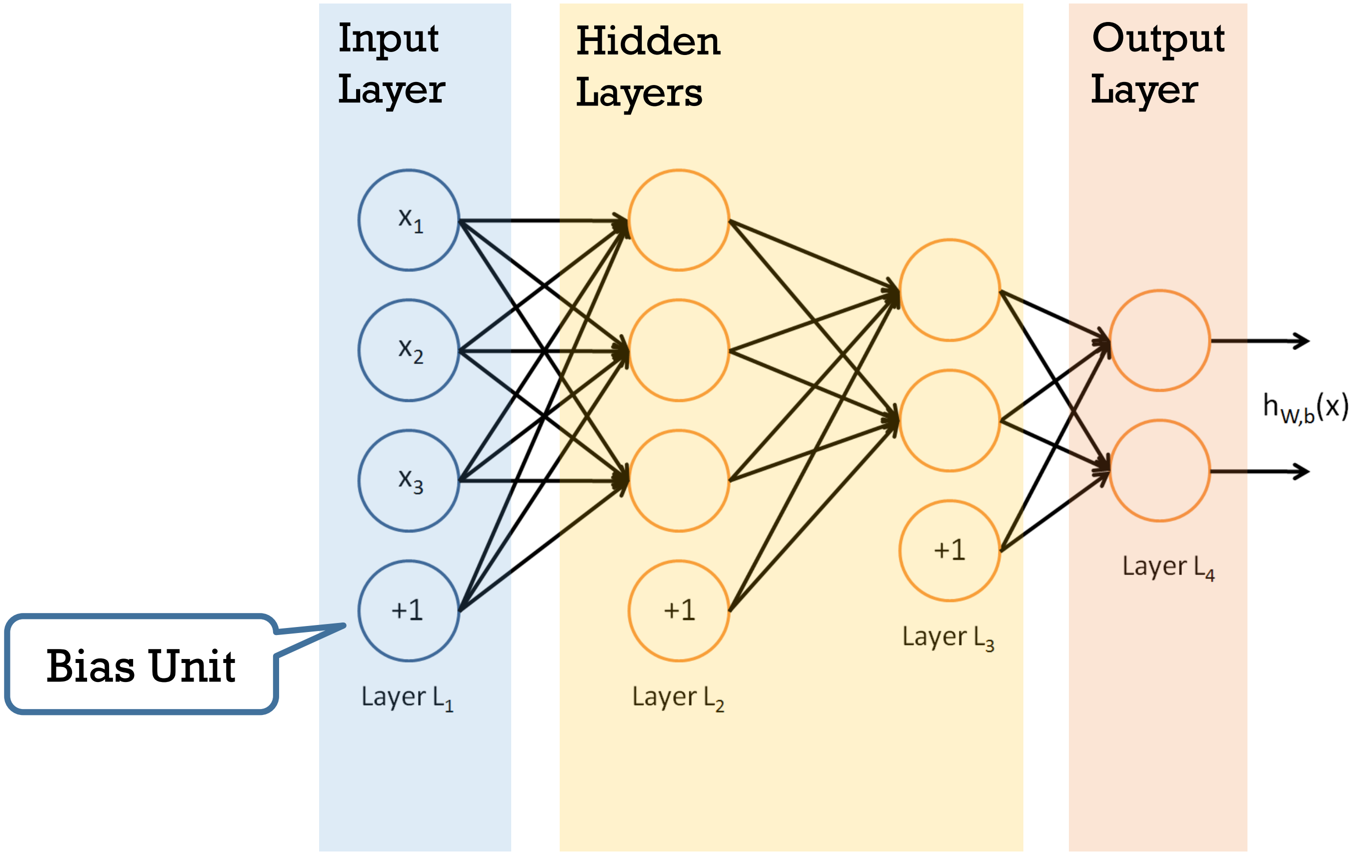
sigmoid $f(z) = \frac{1}{1+e^{-z}}$

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

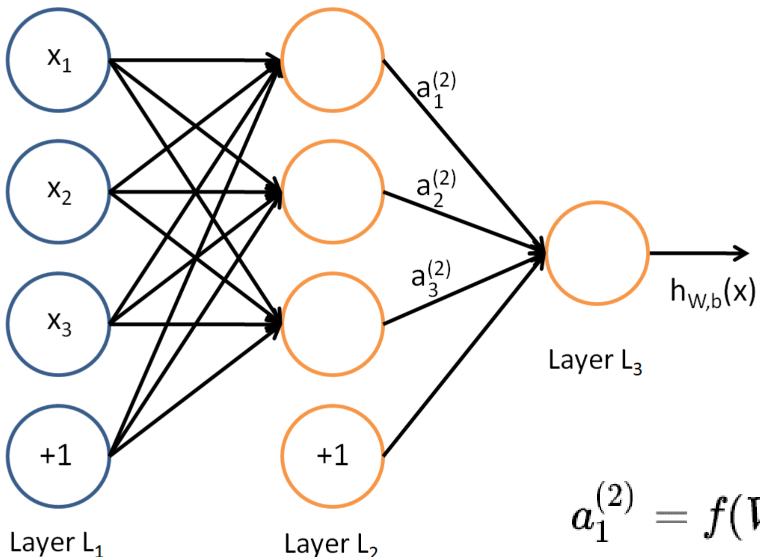
softplus $f(z) = \ln(1 + e^{-z})$

rectified linear unit (ReLU) $f(z) = \max(0, z)$

MULTI-LAYER NEURAL NETWORK

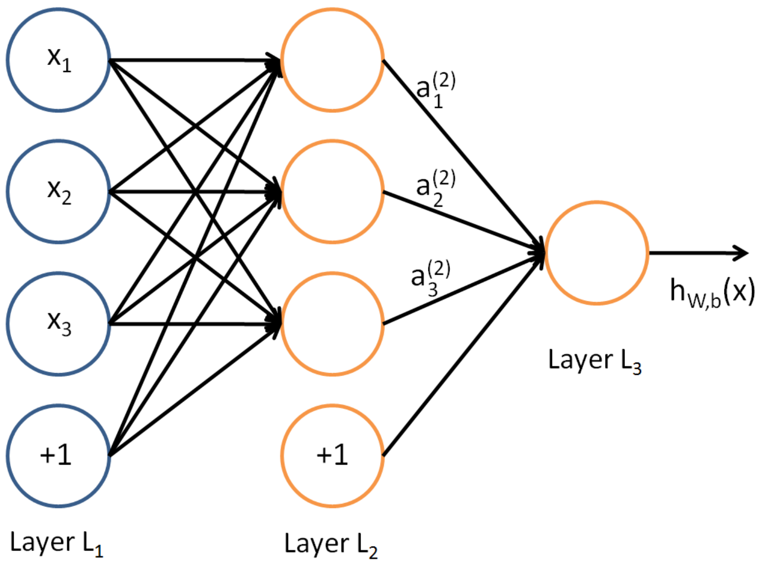


MULTI-LAYER NEURAL NETWORK



$$
\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) = a_1^{(3)} &= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

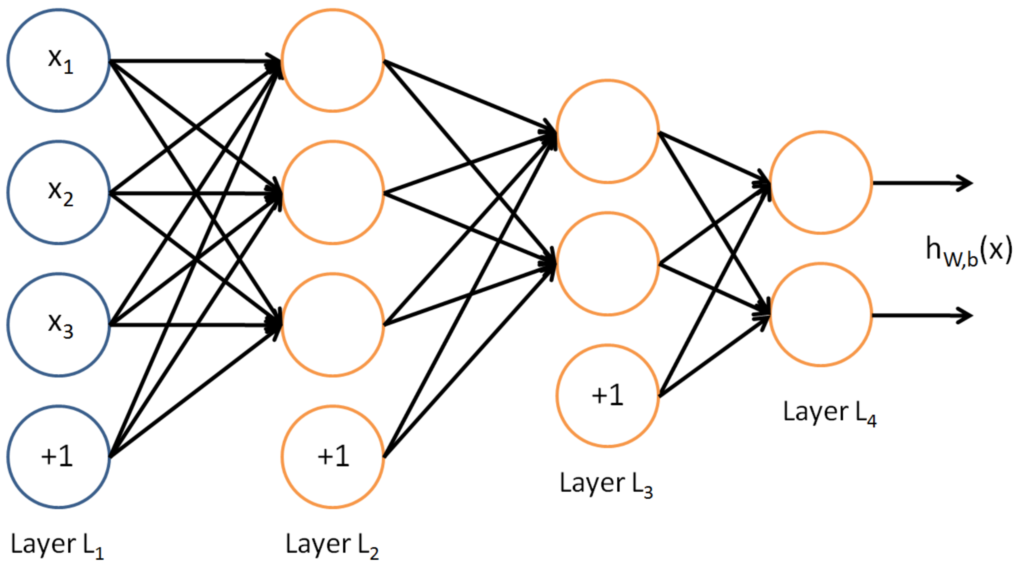
MULTI-LAYER NEURAL NETWORK



Forward Propagation.

$$
\begin{aligned}
z^{(2)} &= W^{(1)}x + b^{(1)} \\
a^{(2)} &= f(z^{(2)}) \\
z^{(3)} &= W^{(2)}a^{(2)} + b^{(2)} \\
h_{W,b}(x) &= a^{(3)} = f(z^{(3)})
\end{aligned}
$$

MULTI-LAYER NEURAL NETWORK



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

Activation

Feedforward Neural Network.

Neural Network Architecture.
Arrangement of neurons, e.g. number of neurons in each layer.

BACKPROPAGATION

TRAINING LOSS

For binary neurons, other loss functions are used.

Point Loss

$$J(W,b;x,y)=\frac{1}{2}\left\|h_{W,b}(x)-y\right\|^2+\frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2$$

weight decay regularization

Training Loss

$$\begin{aligned}J(W,b)&=\left[\frac{1}{m}\sum_{i=1}^{m}J(W,b;x^{(i)},y^{(i)})\right]\\&=\left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}(x^{(i)})-y^{(i)}\right\|^2\right)\right]+\frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2\end{aligned}$$

λ weight decay parameter

BACKPROPAGATION

Chain Rule for Neural Networks.

$$\frac{\partial}{\partial W_{ij}^{(l)}}\left(\frac{1}{2}\left\|a^{(n_l)}-y\right\|^2\right)=\left(a^{(n_l)}-y\right)\frac{\partial}{\partial W_{ij}^{(l)}}f\left(z^{(n_l)}\right)$$

$$=\left(a^{(n_l)}-y\right)f'\left(z^{(n_l)}\right)\frac{\partial}{\partial W_{ij}^{(l)}}\left(W^{(n_l-1)}a^{(n_l-1)}+b^{(n_l-1)}\right)$$

$$=\left(a^{(n_l)}-y\right)f'\left(z^{(n_l)}\right)W^{(n_l-1)}\frac{\partial}{\partial W_{ij}^{(l)}}a^{(n_l-1)}$$

$$=\underbrace{\underbrace{\left(a^{(n_l)}-y\right)f'\left(z^{(n_l)}\right)}_{\delta^{(n_l)}}W^{(n_l-1)}f'\left(z^{(n_l-1)}\right)}_{\delta^{(n_l-1)}}\frac{\partial}{\partial W_{ij}^{(l)}}z^{(n_l-1)}$$

BACKPROPAGATION

1. Perform a feed-forward pass, computing the activations layer by layer.

2. For the output layer (layer n_l), set

$$\delta^{(n_l)} = -(y - a^{(n_l)}) \bullet f'(z^{(n_l)})$$

3. For $l = n_l - 1, n_l - 2, n_l - 3, \ldots, 2$, set

$$\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \bullet f'(z^{(l)})$$

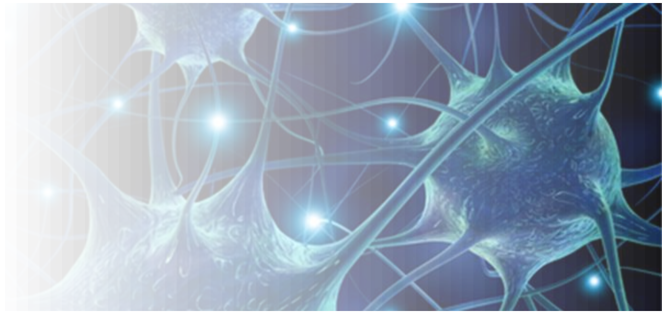
4. Compute the desired partial derivatives:

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^T,$$
$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)}.$$

$\bullet$ denotes element-wise multiplication

Note that with weight decay, $\nabla_{W^{(l)}} J(W, b) = \nabla_{W^{(l)}} J(W, b; x, y) + \lambda W^{(l)}$

AUTOENCODERS

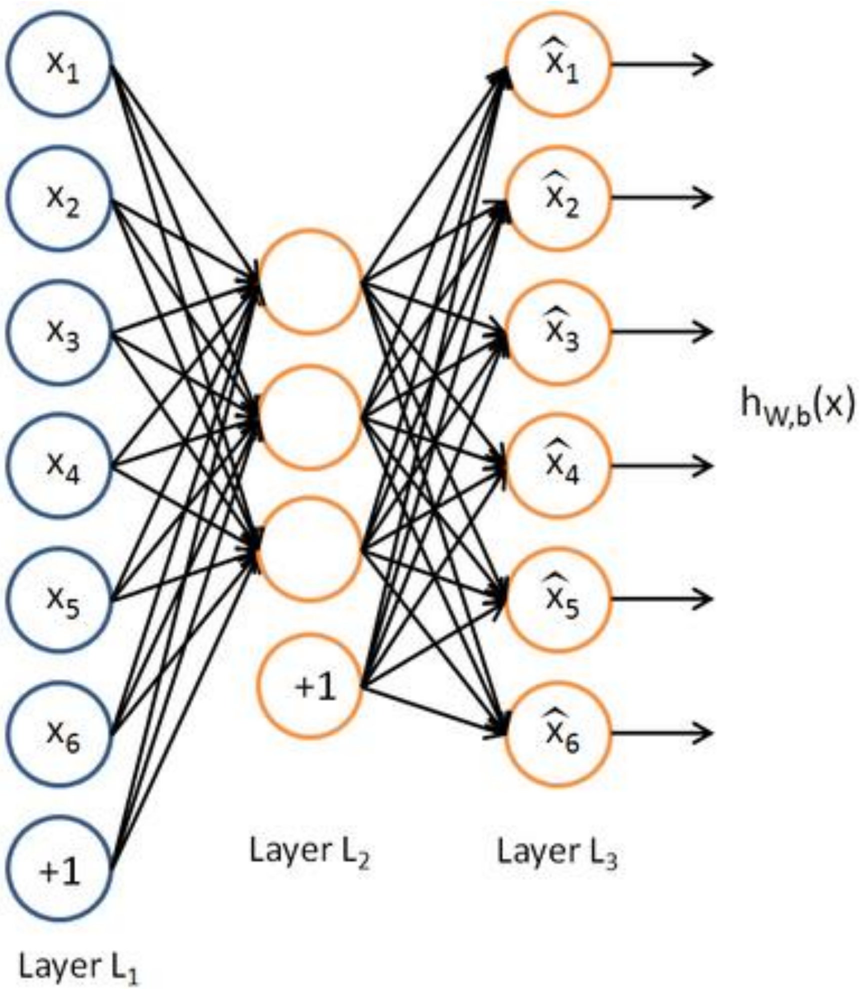


AUTOENCODERS

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

Strategies for Dimensionality Reduction

- Few hidden neurons
- Sparse activations



* not in syllabus

SPARSE AUTOENCODER

Sparsity Penalty.

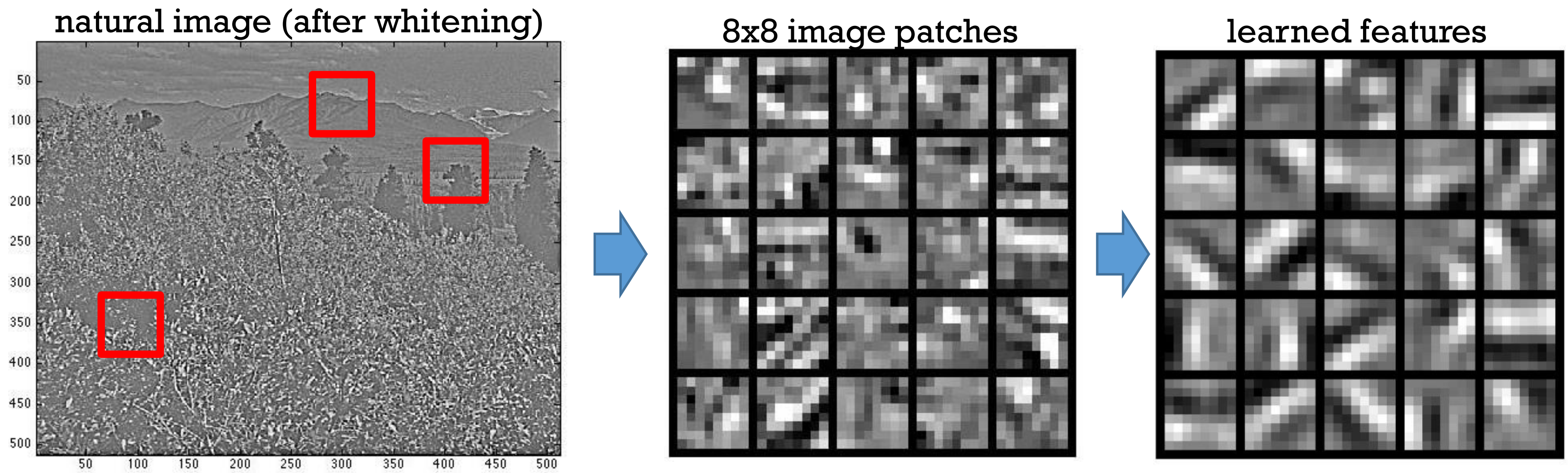
Average activation $$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right]$$

$$\mathrm{KL}(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}$$

$$J_{\mathrm{sparse}}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} \mathrm{KL}(\rho||\hat{\rho}_j),$$

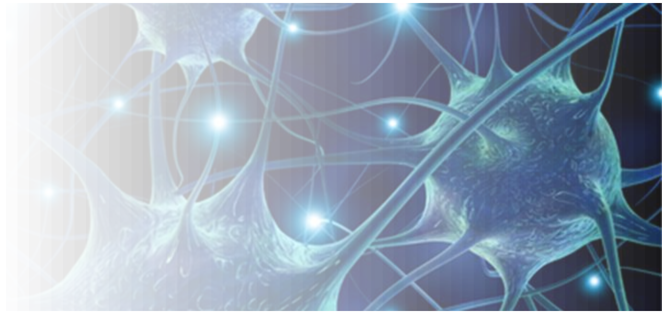
β sparsity parameter

NATURAL IMAGES



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

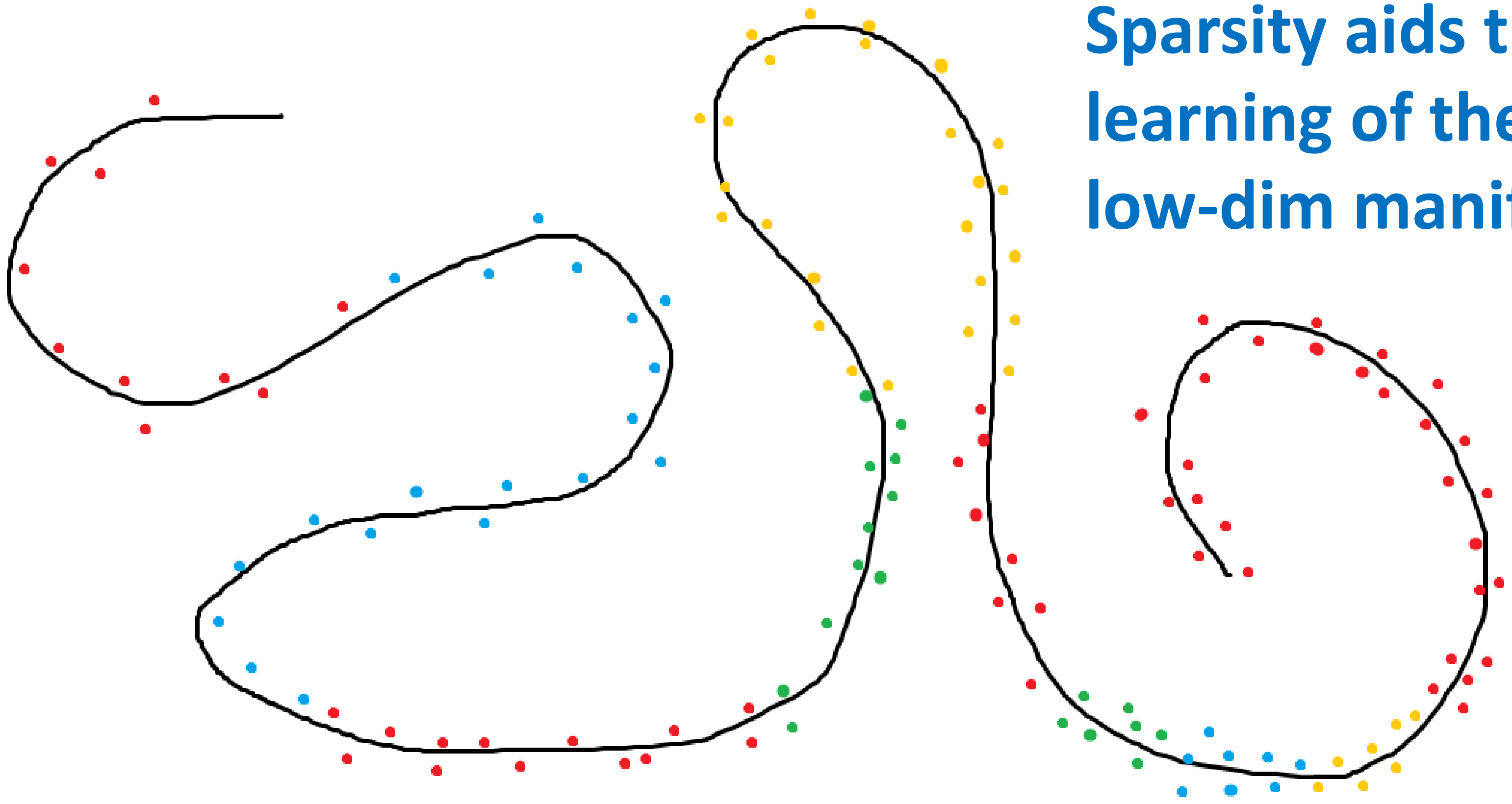
FEATURE ENGINEERING



WHY DOES DEEP LEARNING WORK?

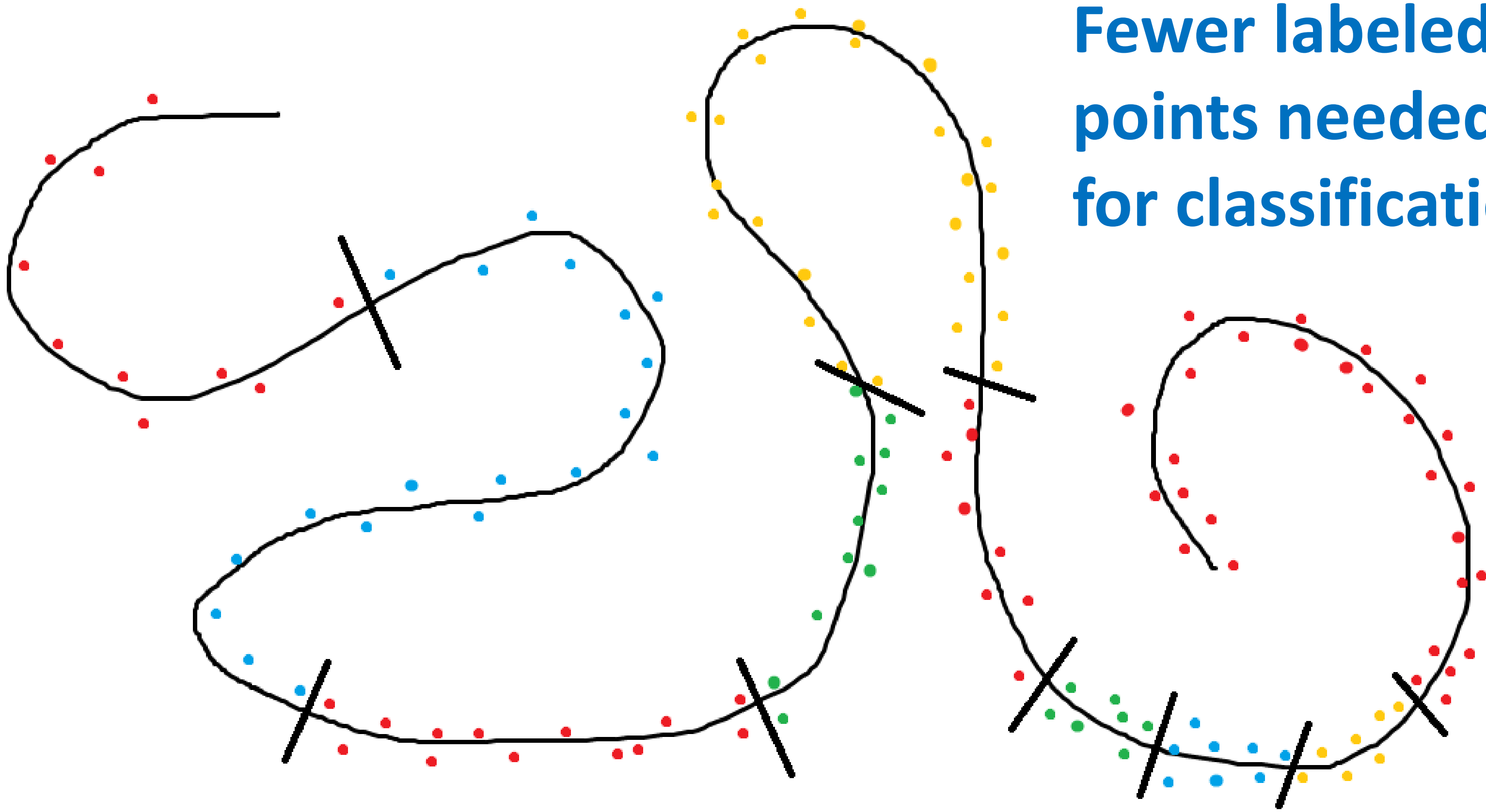
Data is often near low-dim manifold in high-dim space

WHY DOES DEEP LEARNING WORK?



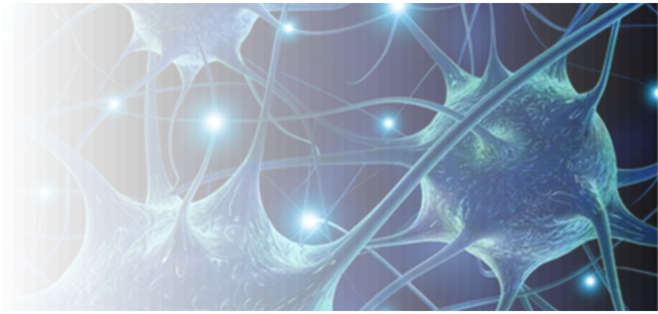
Sparsity aids the learning of the low-dim manifold

WHY DOES DEEP LEARNING WORK?



Fewer labeled points needed for classification

HISTORY



* not in syllabus

RIDICULOUSLY SIMPLIFIED HISTORY

1943 Artificial Neuron (McCullough, Pitts)

1957 Perceptrons (Rosenblatt)

1969 Problem with XOR (Minsky, Papert)

FIRST AI WINTER

1986 Backpropagation (Rumelhart, Hinton, Williams)

1989 Convolutional Neural Nets (LeCun)
Autoencoders, Belief Nets, Recurrent NN, Reinforcement

1995 Problems with Backprop.
Rise of SVMs and Random Forests.

SECOND AI WINTER

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning/

* not in syllabus

DEEP LEARNING CONSPIRACY

Yann LeCun,
Geoffrey Hinton,
Yoshua Bengio,
Andrew Ng

2006 Greedy Initialization of Layers

2009 Graphics Processing Units

2012 Dropout (ImageNet)

WHAT WAS WRONG WITH BACKPROPAGATION IN 1986?

1. Our labeled datasets were thousands of times too small.
2. Our computers were millions of times too slow.
3. We initialized the weights in a stupid way.
4. We used the wrong type of non-linearity.

Hinton, "Deep Learning," https://royalsociety.org/events/2015/05/breakthrough-science-technologies-machine-learning/

SUMMARY

- Multilayer Neural Networks
  - Neuron
  - Activation Function
  - Forward Propagation
- Learning Algorithm
  - Cost Function
  - Backpropagation
  - Autoencoders

INTENDED LEARNING OUTCOMES

Deep Learning

- Describe how the output of an artificial neuron relates to its inputs. Give examples of activation functions which are commonly used.
- Describe how the output of a multilayer neural network relates to its inputs. In particular, write down formulas for forward propagation. Given a network, identify the neural network architecture.
- Write down the training loss of a feedforward network. Derive the gradient formulas in backpropagation from the training loss.
- Give a definition of an autoencoder. Explain how they are used for dimensionality reduction. Describe two strategies for reduction.
- List some successful applications of deep learning. Give four reasons for the recent success of deep learning.