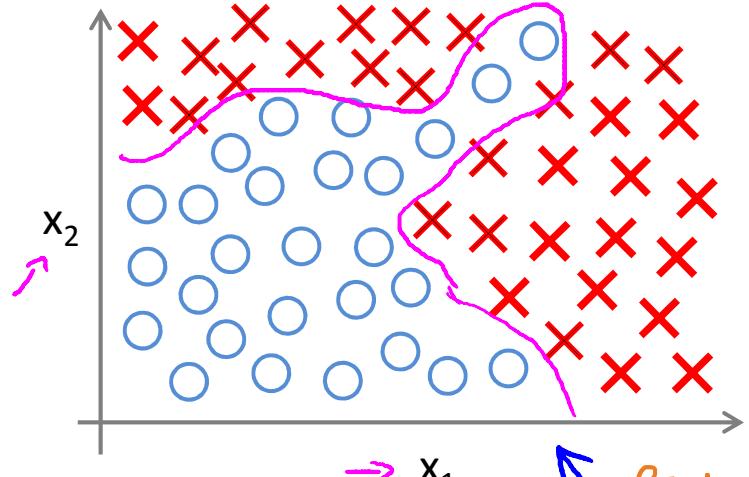


Machine Learning

Neural Networks: Representation

Non-linear hypotheses

Non-linear Classification



$\rightarrow \underline{x_1}$ = size
 $\underline{x_2}$ = # bedrooms
 $\underline{x_3}$ = # floors
 x_4 = age
 \dots
 $x_{100} -$

This hypothesis separates +ve and -ve examples
 $\rightarrow \underline{x_1^2}, \underline{x_2^2}, \underline{x_3^2}, \dots, \underline{x_{100}^2}$
 $\underline{x_1^2}, \underline{x_2 x_3} \dots$
 $\underline{h = 100 \text{ features}}$
 With these features we want to find whether our house will be sold/not

sigmoid function

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

$$\rightarrow \underline{x_1^2}, \underline{x_1 x_2}, \underline{x_1 x_3}, \underline{x_1 x_4} \dots \underline{x_1 x_{100}} \\ \underline{x_2^2}, \underline{x_2 x_3} \dots$$

≈ 5000 feature

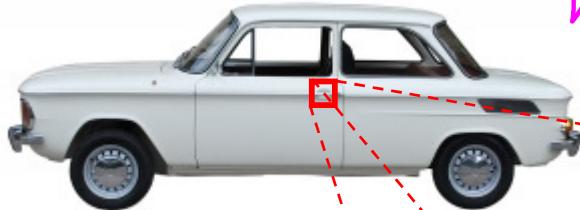
$$\mathcal{O}(n^2) \quad \approx \frac{n^2}{2} \quad 10$$

$$\mathcal{O}(n^3) \quad 170,000$$

What is this?

Normal people will this image as a car
and various parts of it

You see this:



How Computer see this car?

Computer perspective

Computer sees the door handle

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

These are the matrix
of pixel intensity
values to represent
the door handle.



Computer Vision: Car detection



Cars



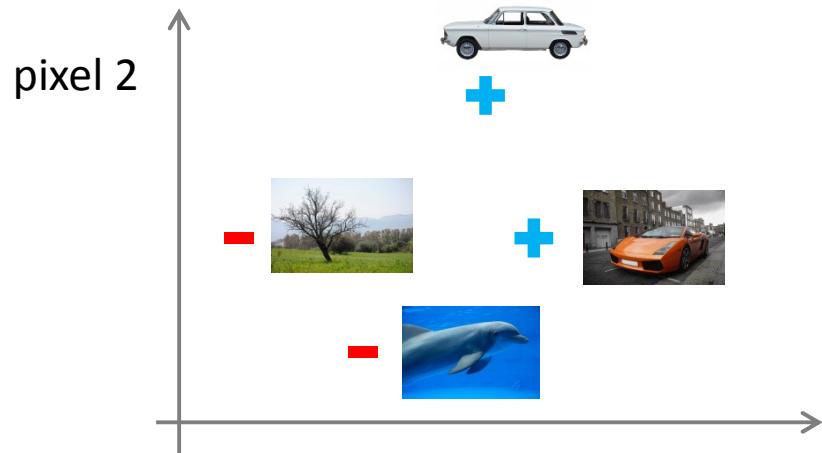
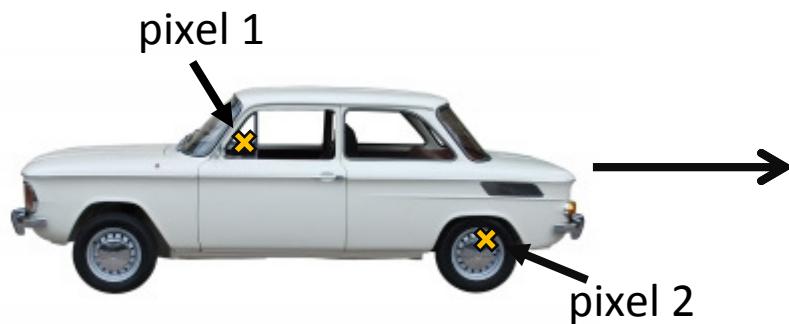
Not a car

- 1) Data-set \Rightarrow labelled training set
- 2) Fit the data into Model.

3) Testing:



- 4) What is this? *The model should predict whether this is a car/not*



+

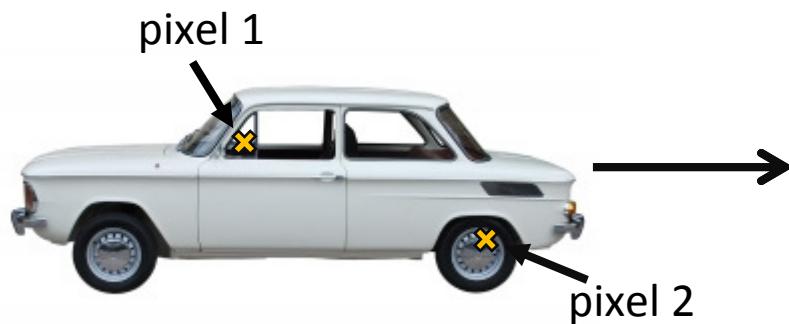
Cars

-

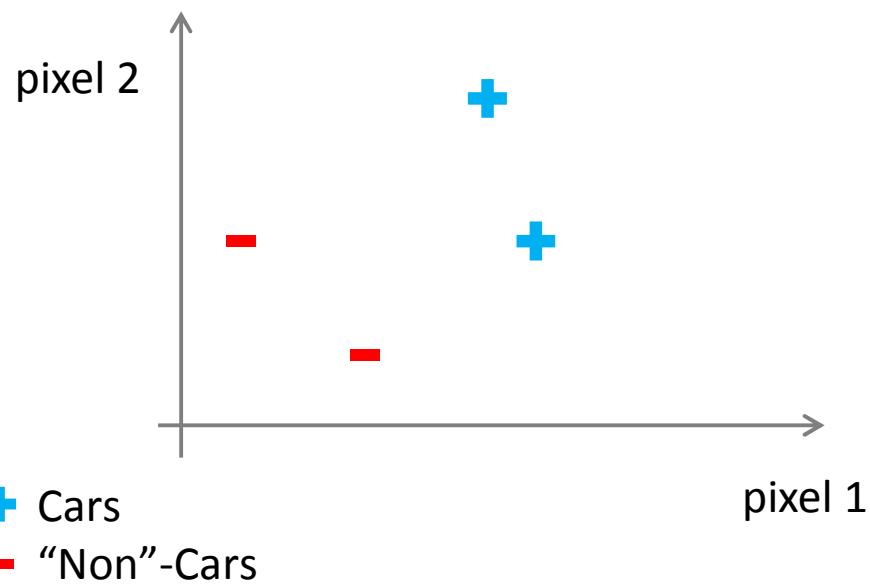
"Non"-Cars

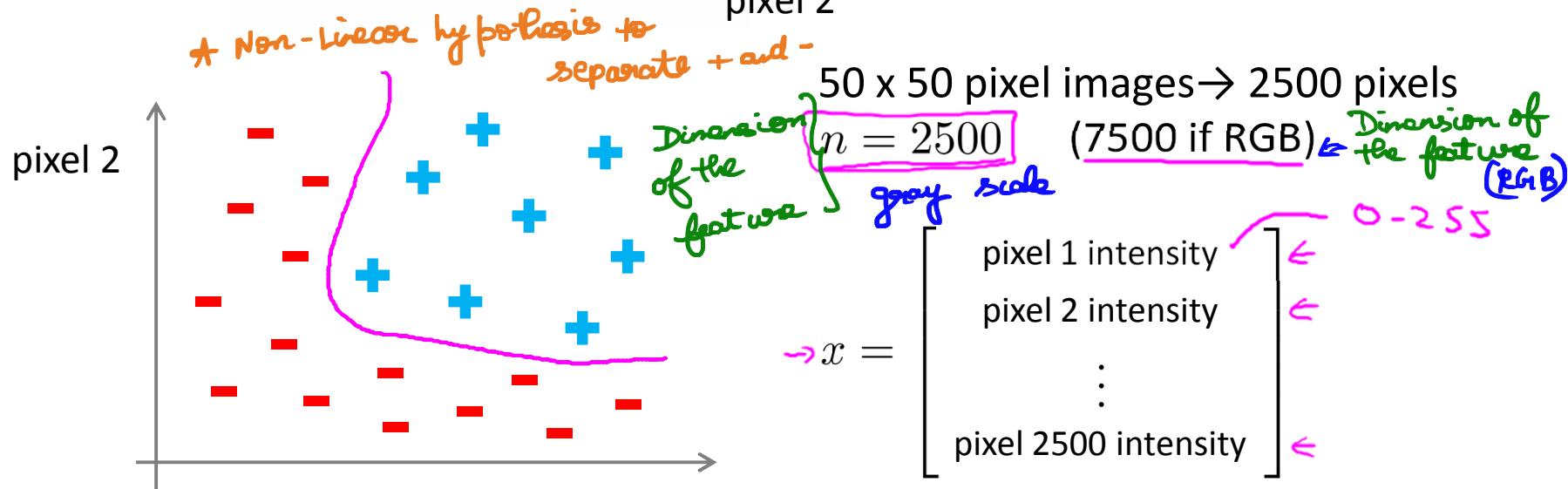
pixel 1

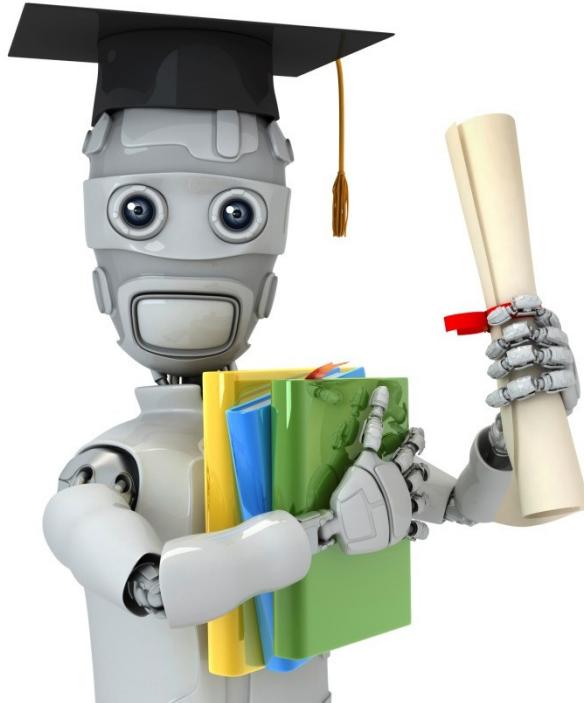
Learning
Algorithm



Learning
Algorithm







Machine Learning

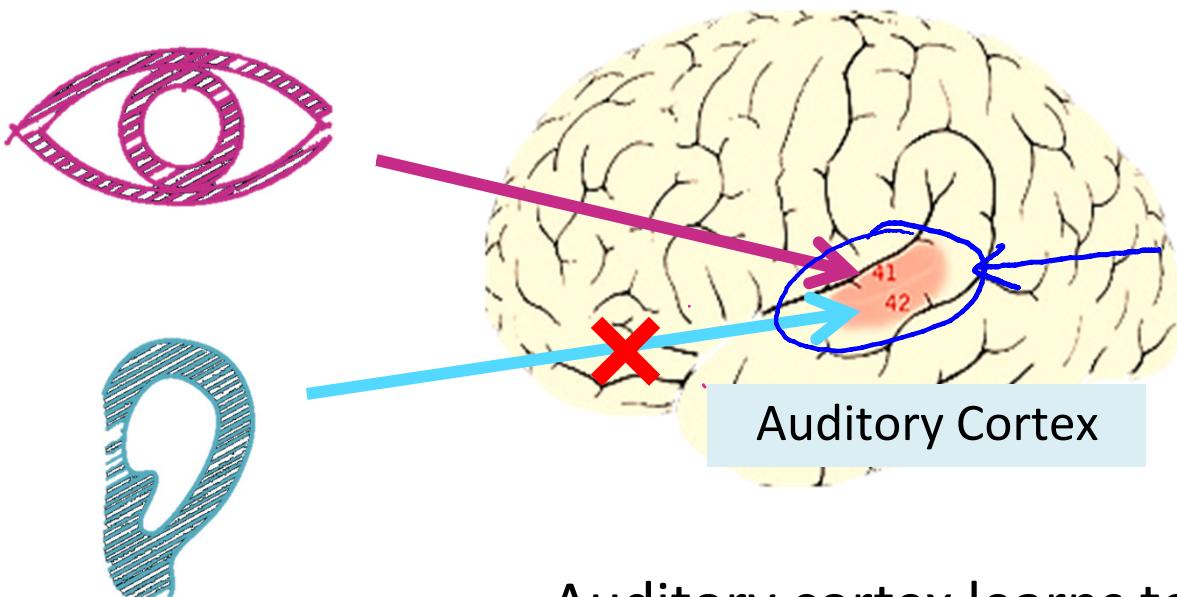
Neural Networks: Representation

Neurons and the brain

Neural Networks

- Origins: Algorithms that try to mimic the brain.
- Was very widely used in 80s and early 90s; popularity diminished in late 90s.
- Recent resurgence: State-of-the-art technique for many applications

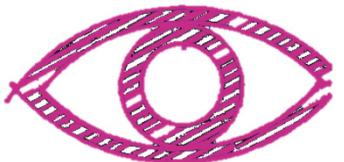
The “one learning algorithm” hypothesis



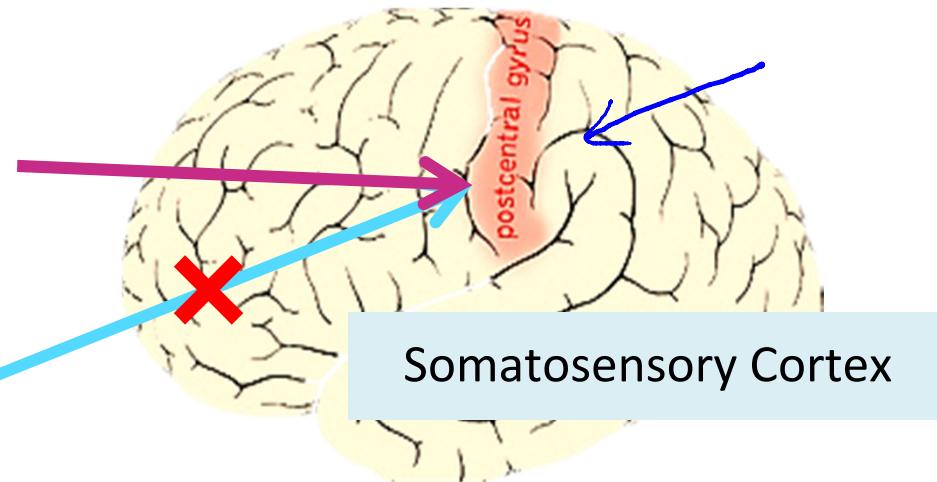
Auditory cortex learns to see

The “one learning algorithm” hypothesis

Now able
to hear



Feeling the
touch



Somatosensory cortex learns to see
(for sensory organs)

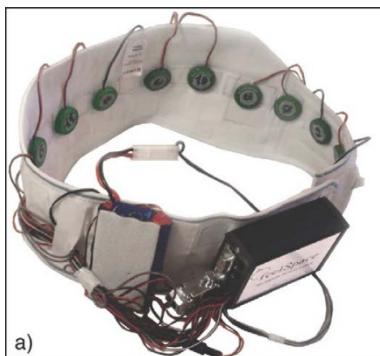
Sensor representations in the brain



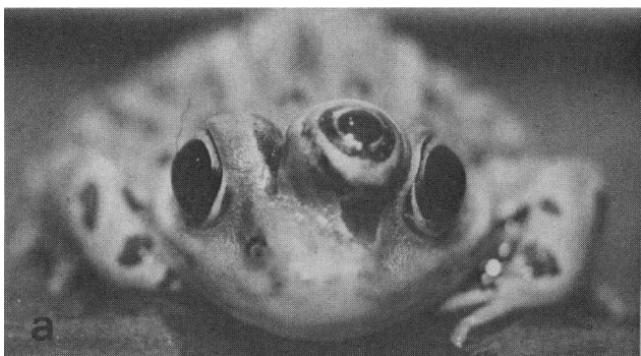
Seeing with your tongue



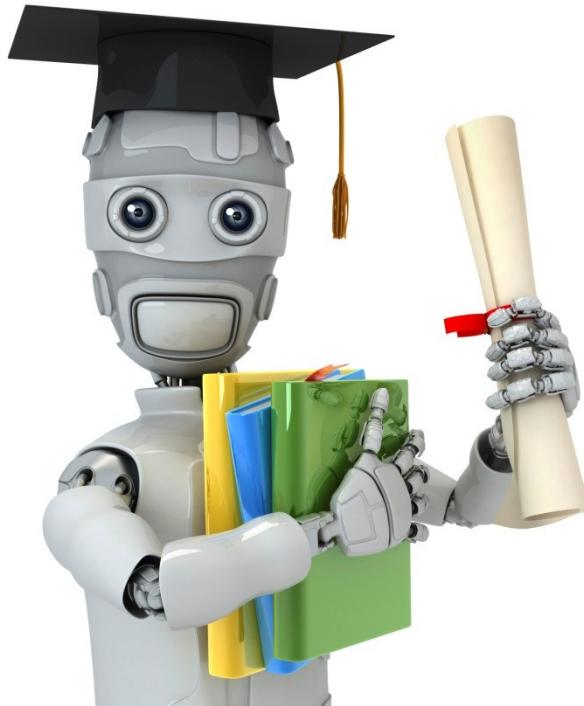
Human echolocation (sonar)



Haptic belt: Direction sense



Implanting a 3rd eye



Machine Learning

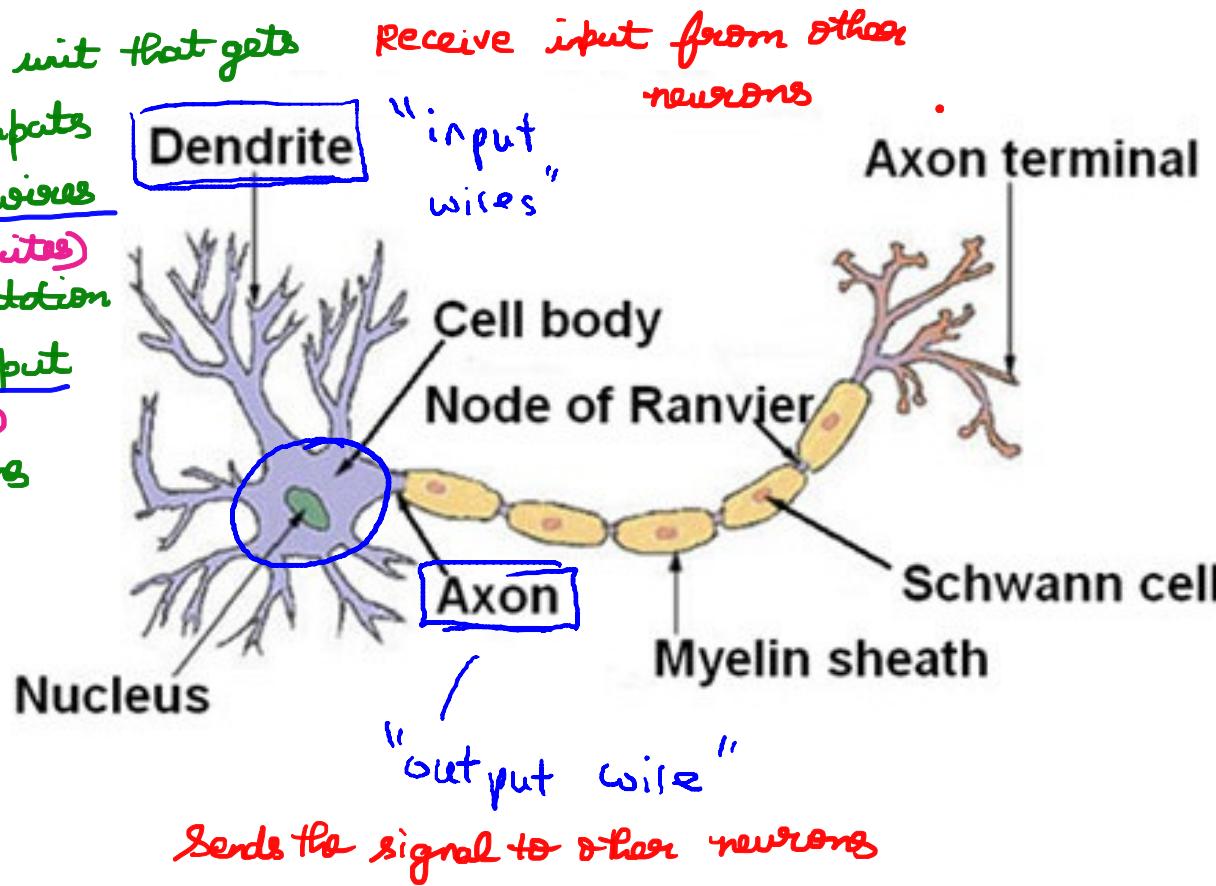
Neural Networks: Representation

Model representation I

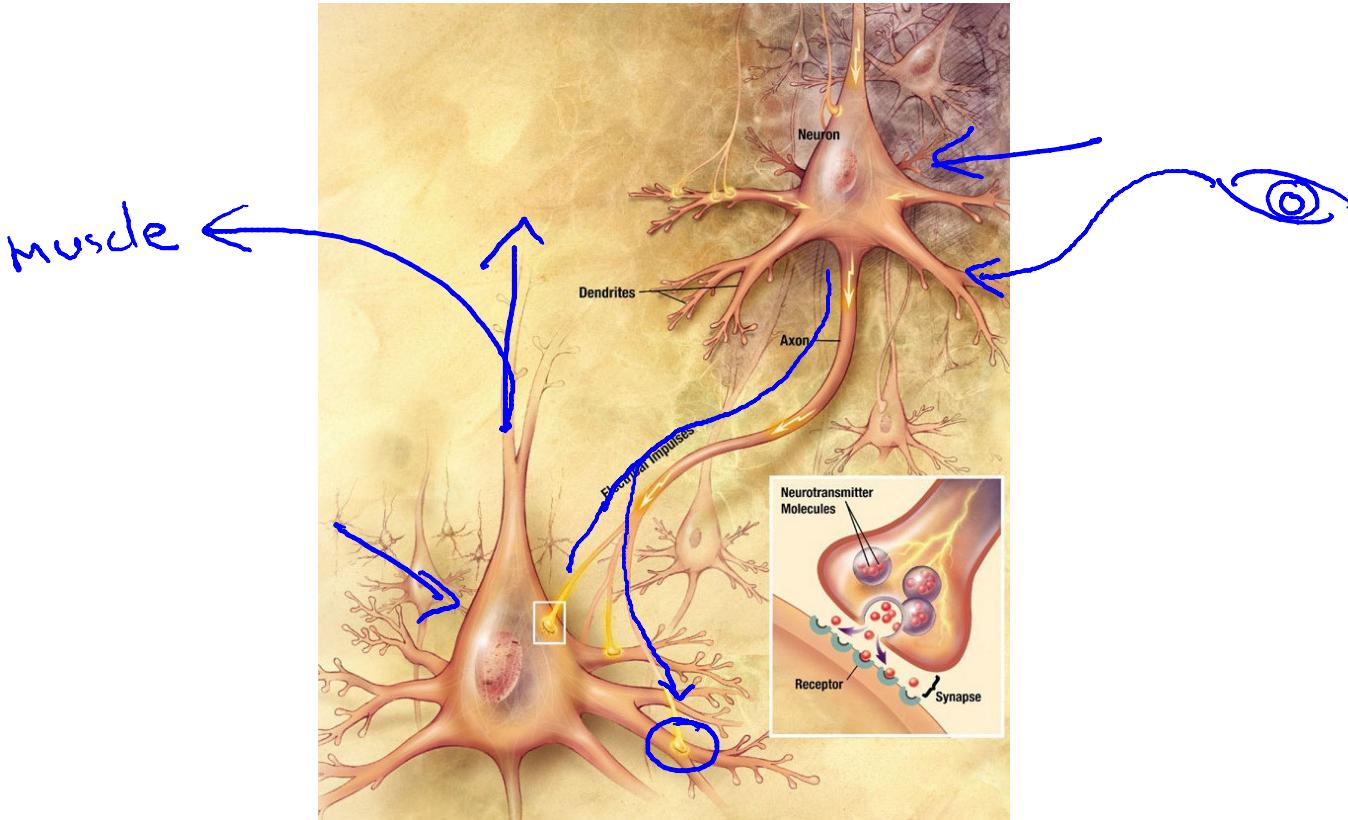
Neuron in the brain

Computational unit that gets
a number of inputs
through input wires
(via Dendrite)
does some computation
and sends out put
(via Axon)
to other neurons

Neural networks were developed as simulating neurons or
networks of neurons in the brain



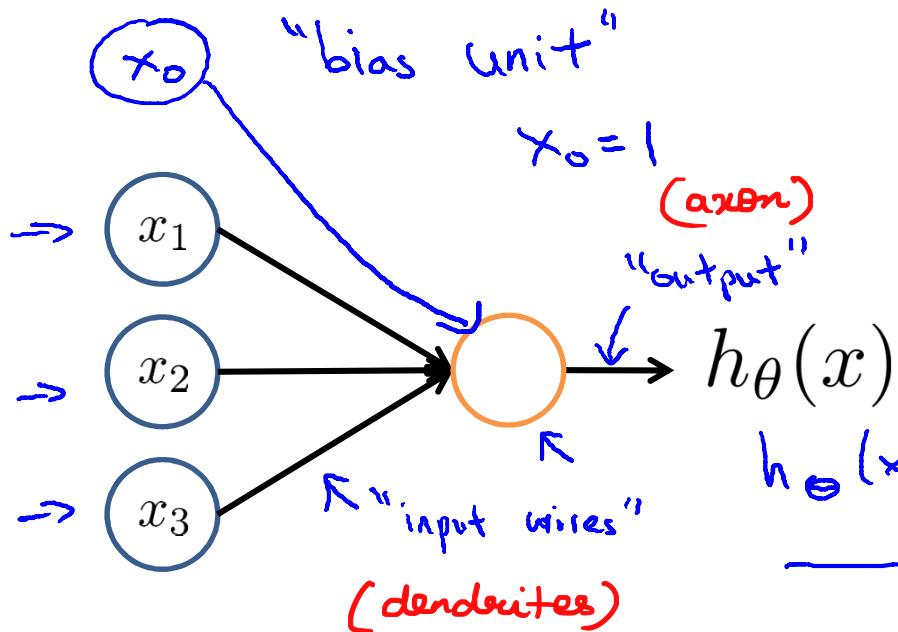
Neurons in the brain



[Credit: US National Institutes of Health, National Institute on Aging]

Andrew Ng

Neuron model: Logistic unit



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

↑
"weights" ←
(parameters ←)

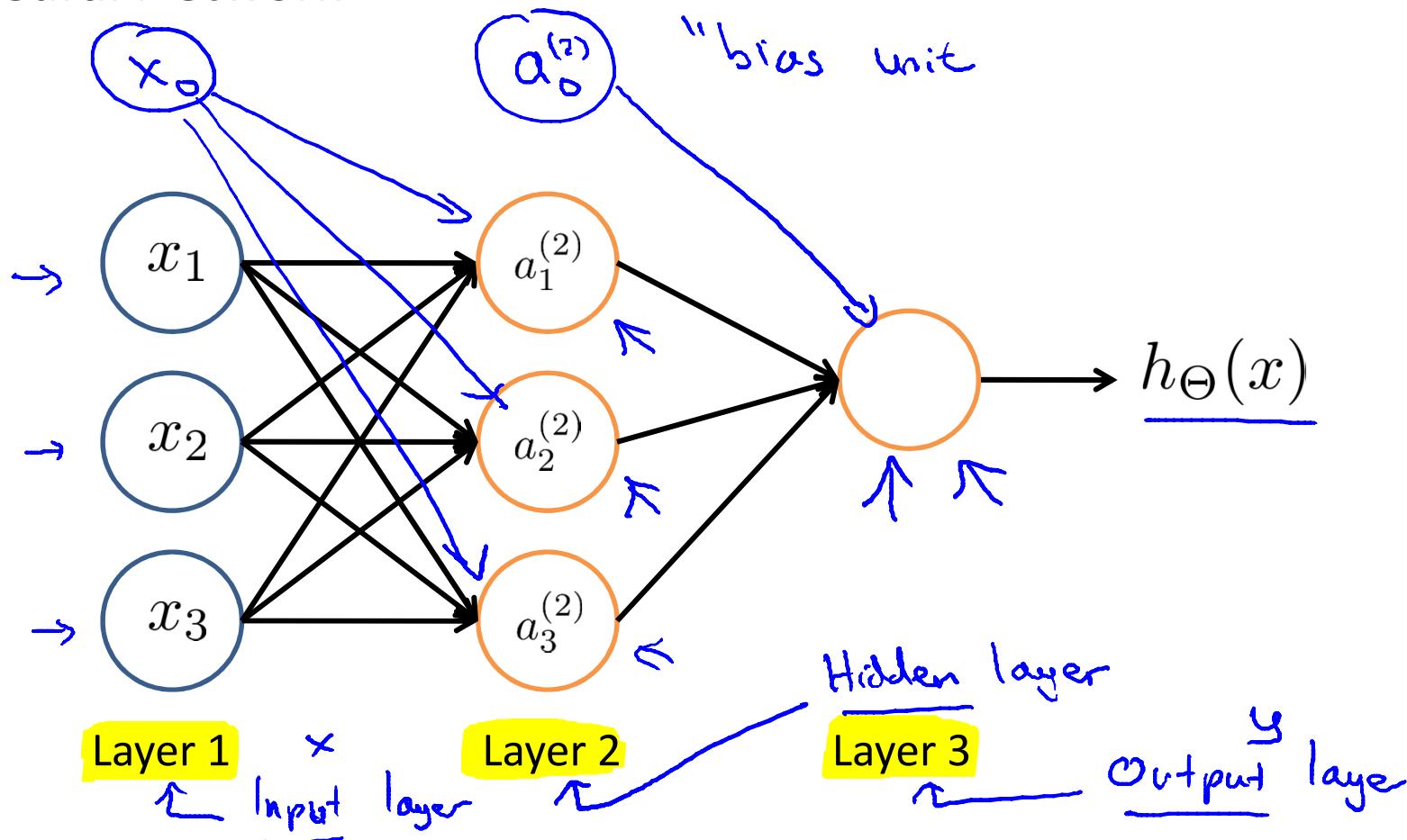
$$h_{\theta}(x) = \frac{1}{1+e^{-\theta^T x}}$$

Sigmoid (logistic) activation function.

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

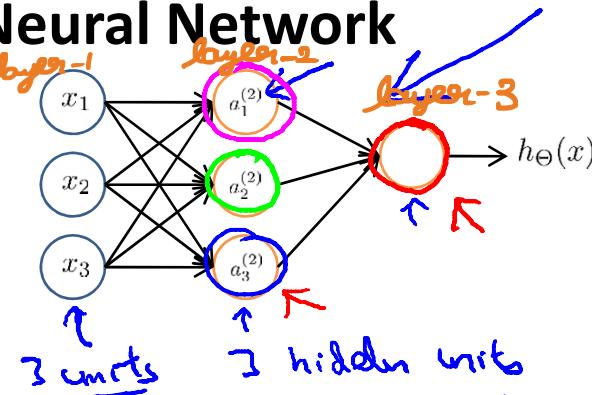
(Group of different neuron strong together)

Neural Network



$x_0 \rightarrow \text{bias}$

Neural Network



$\rightarrow a_i^{(j)}$ = "activation" of unit i in layer j

$\rightarrow \Theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j + 1$

$$\Theta^{(1)} \in \mathbb{R}^{3 \times 4} \quad h_{\Theta}(x)$$

$$\rightarrow a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$\rightarrow a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$\rightarrow a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

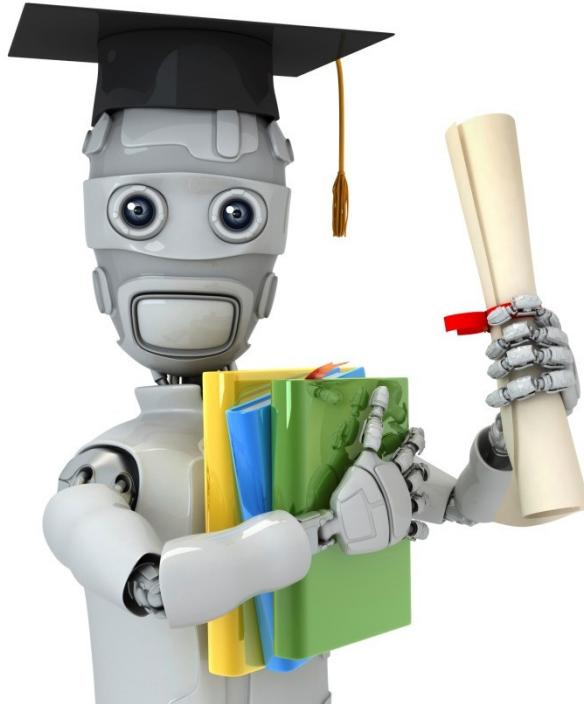
$$\rightarrow h_{\Theta}(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

\rightarrow If network has s_j units in layer j , s_{j+1} units in layer $j + 1$, then $\Theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$.

$$s_{j+1} \times (s_j + 1)$$

layer 1 \rightarrow layer 2
 $(j=1)$

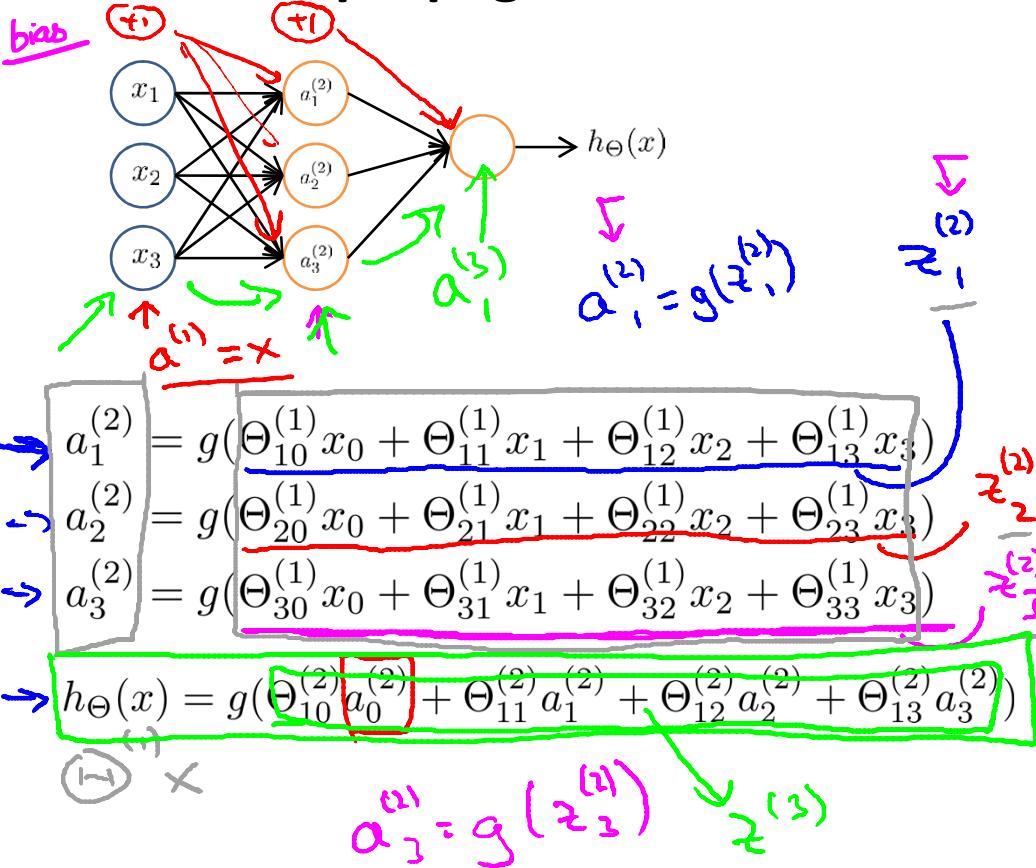
$(j=2)$
layer-2
 \rightarrow
layer-3



Machine Learning

Neural Networks: Representation --- Model representation II

Forward propagation: Vectorized implementation

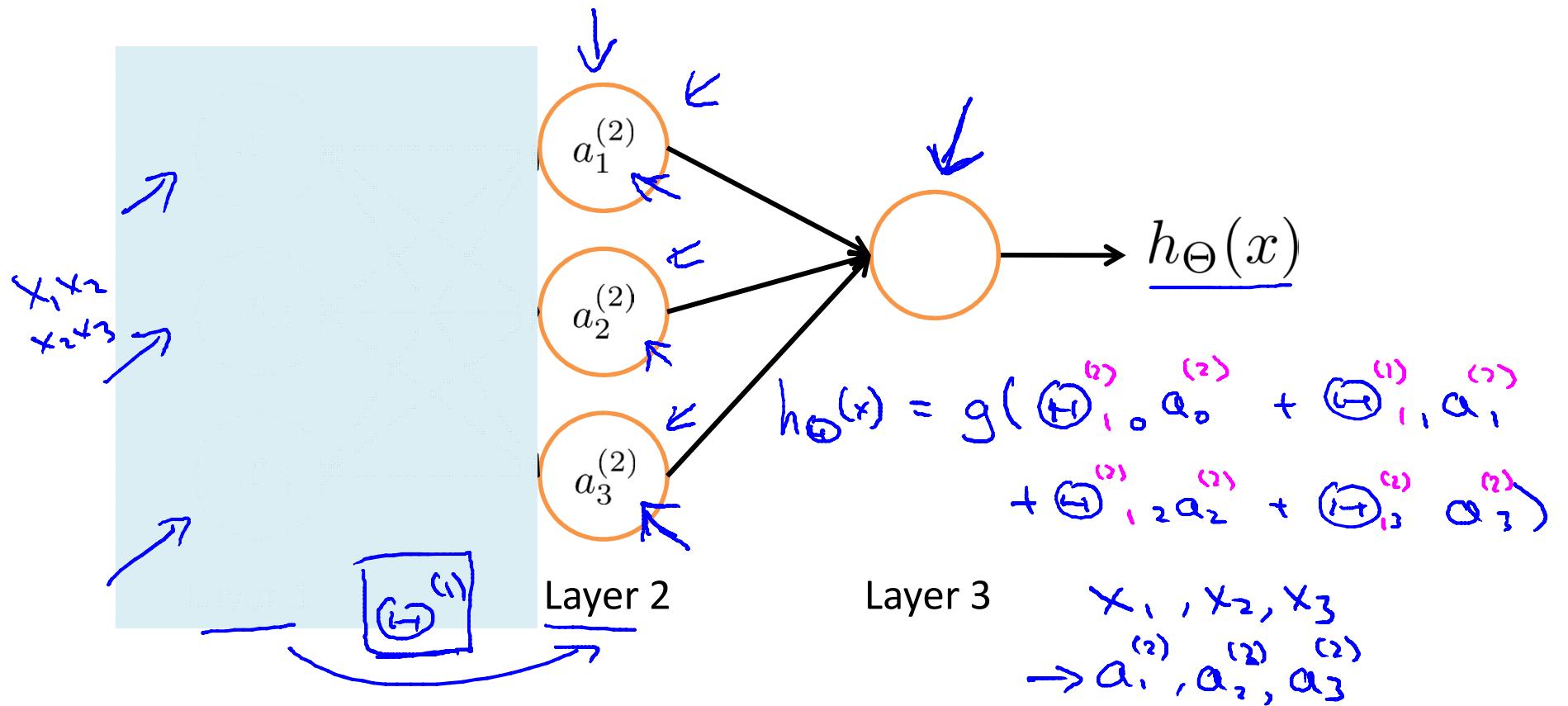


feature vector $\xrightarrow{\text{bias}}$ $x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\xrightarrow{\text{3D vector}}$ $z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$

Forward Propagation Steps:

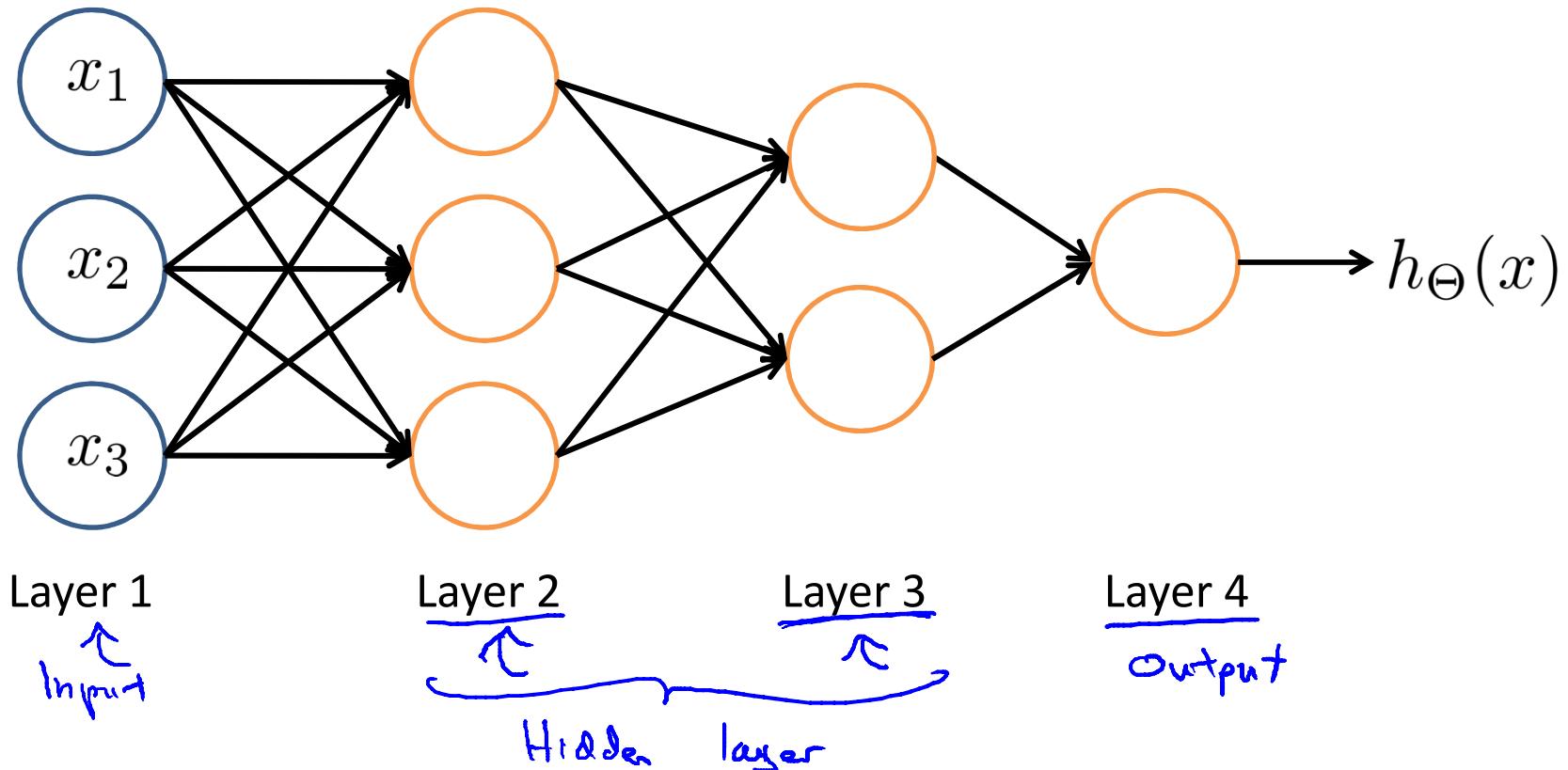
- $z^{(2)} = \Theta^{(1)} x$ (~~$a^{(1)}$~~) $\in \mathbb{R}^3$
- $a^{(2)} = g(z^{(2)})$ $\in \mathbb{R}^3$
- Add $a_0^{(2)} = 1.$ $\rightarrow a^{(2)} \in \mathbb{R}^4$ bias unit
- $z^{(3)} = \Theta^{(2)} a^{(2)}$
- $h_\Theta(x) = a^{(3)} = g(z^{(3)})$

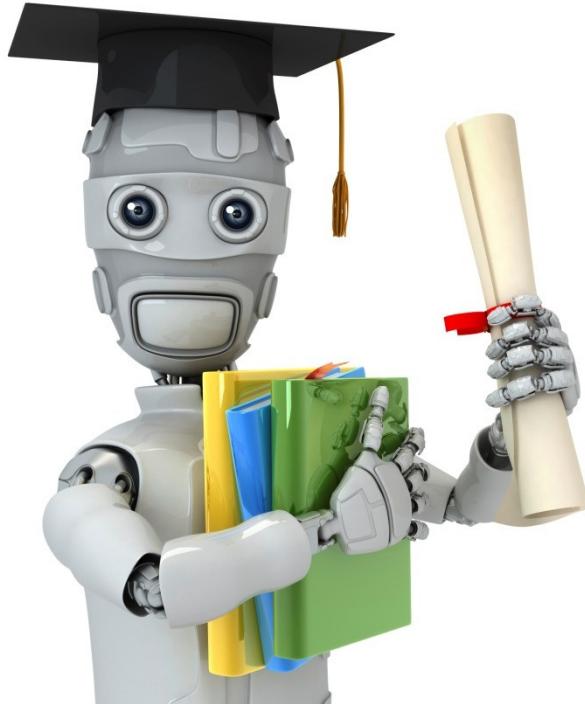
Neural Network learning its own features



Other network architectures

How different Neural Networks are connected to each other?





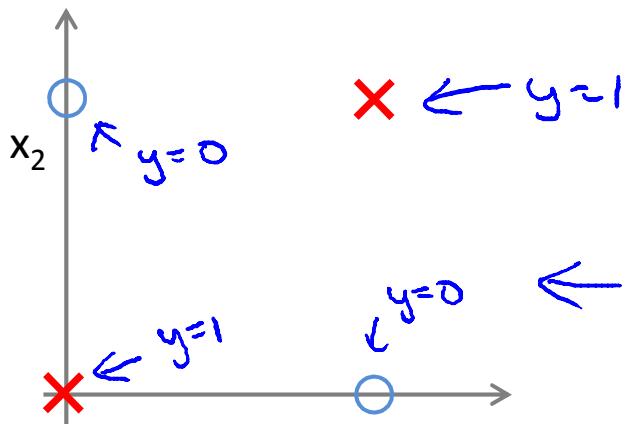
Machine Learning

Neural Networks: Representation

Examples and intuitions I

Non-linear classification example: XOR/XNOR

→ x_1, x_2 are binary (0 or 1).



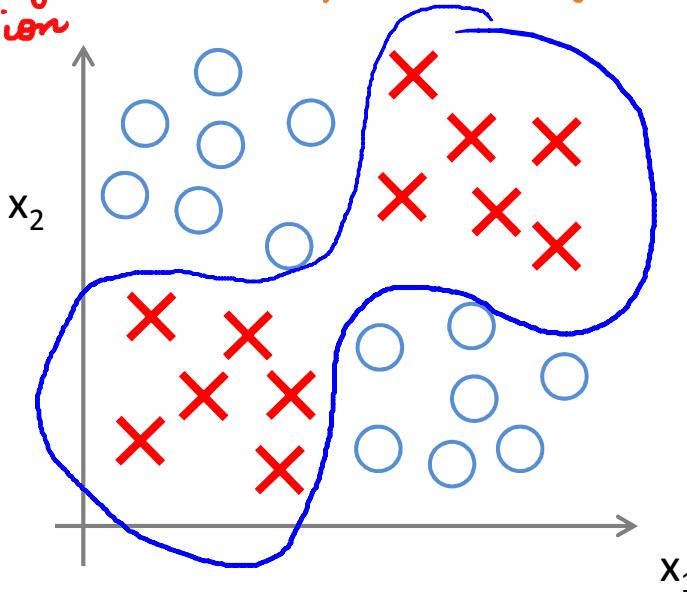
Here 2+ve and 2-ve examples

$$y = \underline{x_1 \text{ XOR } x_2}$$

$$\rightarrow \underline{x_1 \text{ XNOR } x_2} \leftarrow$$

$$\rightarrow \underline{\text{NOT } (x_1 \text{ XOR } x_2)}$$

simplified version Complex Learning problem

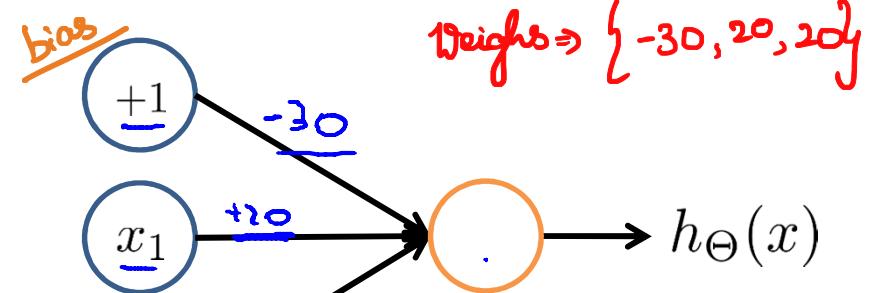


Our aim ⇒ To find a Decision boundary

Simple example: AND

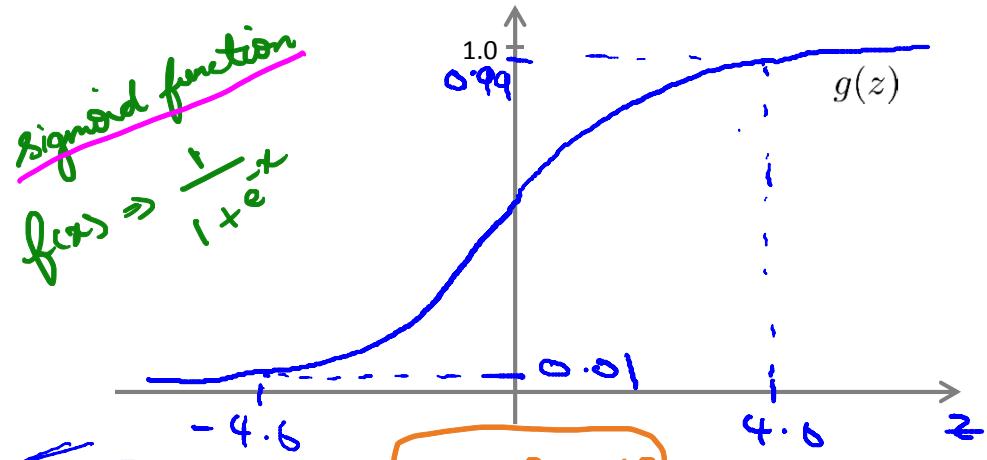
$$\rightarrow x_1, x_2 \in \{0, 1\}$$

$$\rightarrow y = x_1 \text{ AND } x_2$$



$$\rightarrow h_{\Theta}(x) = g\left(\frac{-30}{\pi} + \frac{20x_1}{\pi} + \frac{20x_2}{\pi}\right)$$

$\Theta_{1,0}$ $\Theta_{1,1}$ $\Theta_{1,2}$

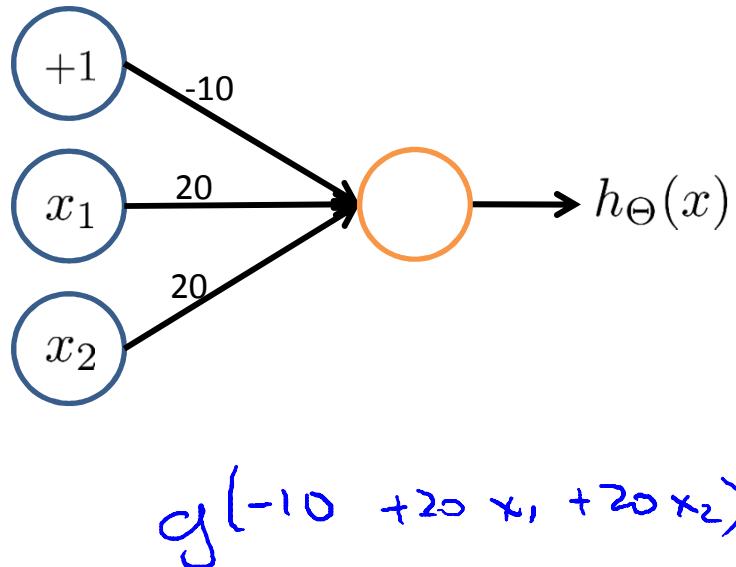


Truth table

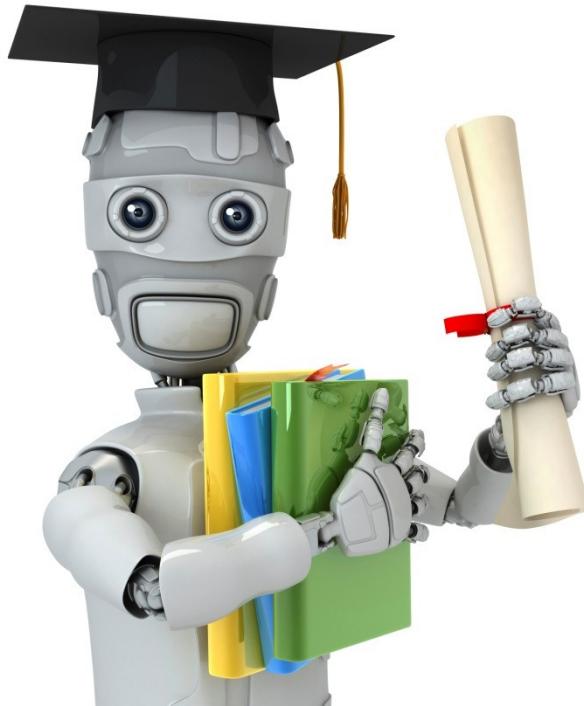
x_1	x_2	$h_{\Theta}(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

$h_{\Theta}(x) \approx x_1 \text{ AND } x_2$

Example: OR function



x_1	x_2	$h_{\Theta}(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	≈ 1
1	1	≈ 1



Machine Learning

Neural Networks: Representation

Examples and intuitions II

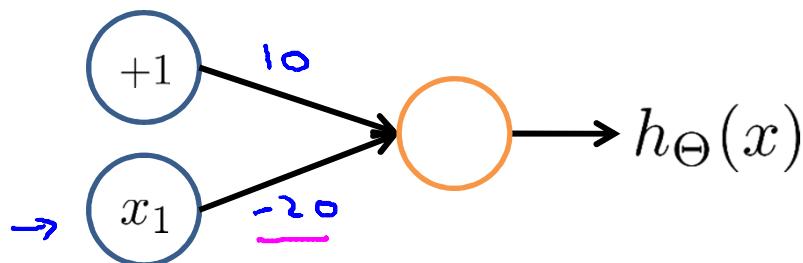
$\rightarrow x_1 \text{ AND } x_2$

$\rightarrow x_1 \text{ OR } x_2$

$\{0, 1\}$.

Negation:

NOT x_1



x_1	$h_{\Theta}(x)$
0	$g(10) \approx 1$
1	$g(-20) \approx 0$

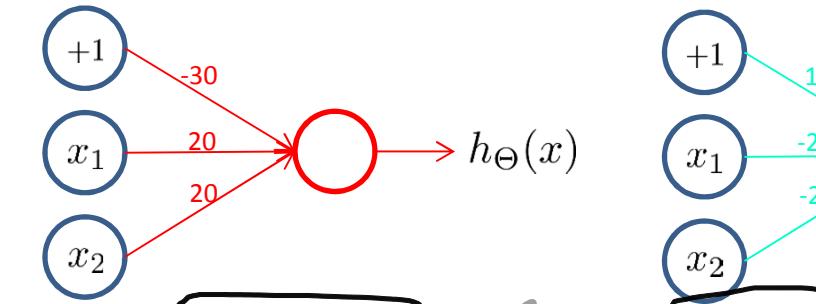
$$h_{\Theta}(x) = g(10 - 20x_1)$$

$\rightarrow (\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$

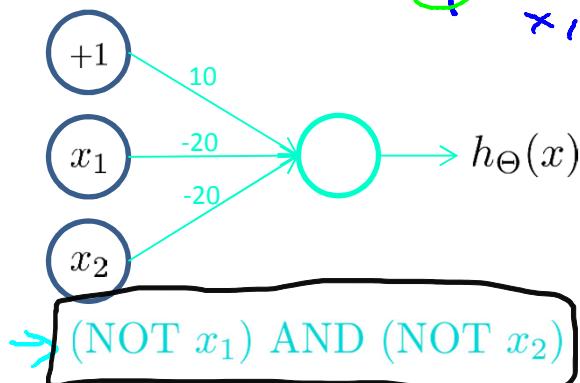
$\begin{cases} = 1 & \text{if and only if} \\ = 0 & \end{cases}$

$\rightarrow x_1 = x_2 = 0$

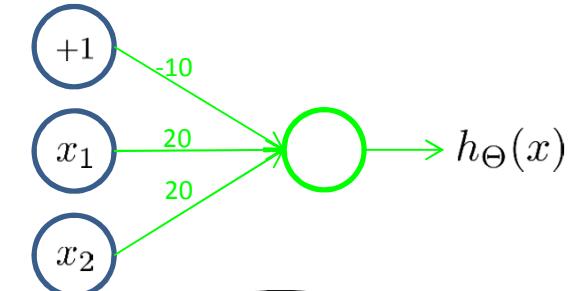
Putting it together: x_1 XNOR x_2



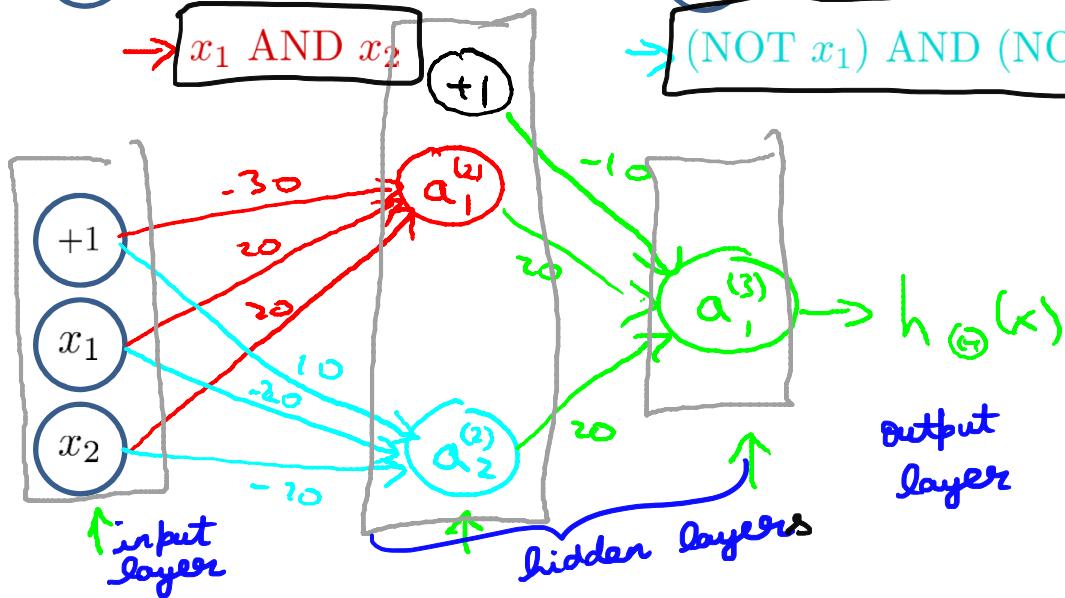
$\rightarrow x_1 \text{ AND } x_2$



$\rightarrow (\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$



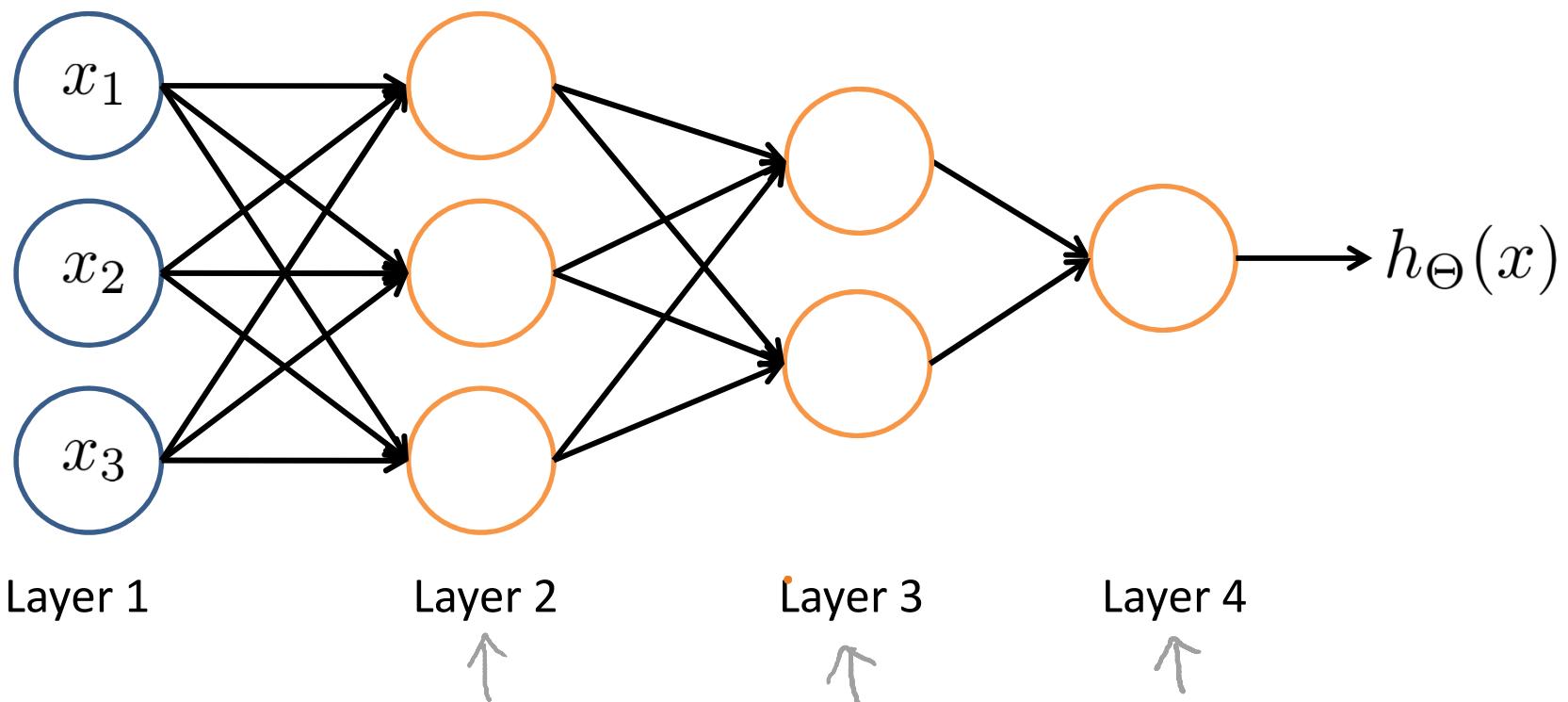
$\rightarrow x_1 \text{ OR } x_2$



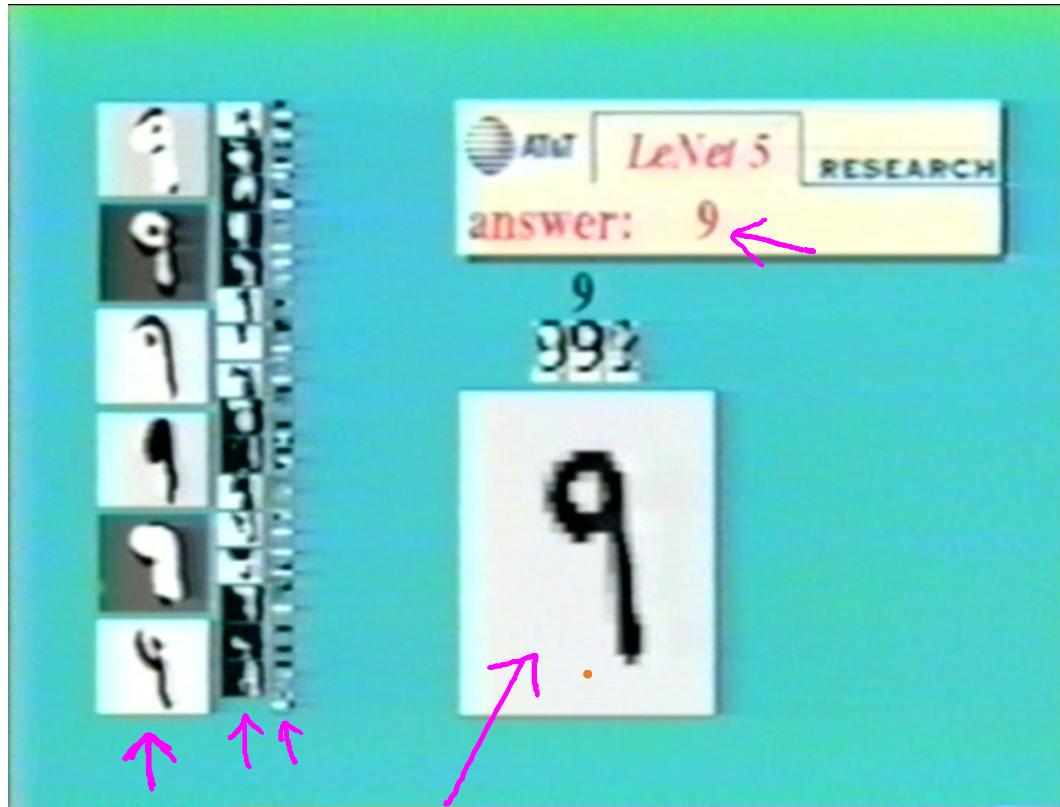
output
layer

x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_{\Theta}(x)$
0	0	0	1	1 ←
0	1	0	0	0 ←
1	0	0	0	0 ←
1	1	1	1	1 ←

Neural Network intuition



Handwritten digit classification

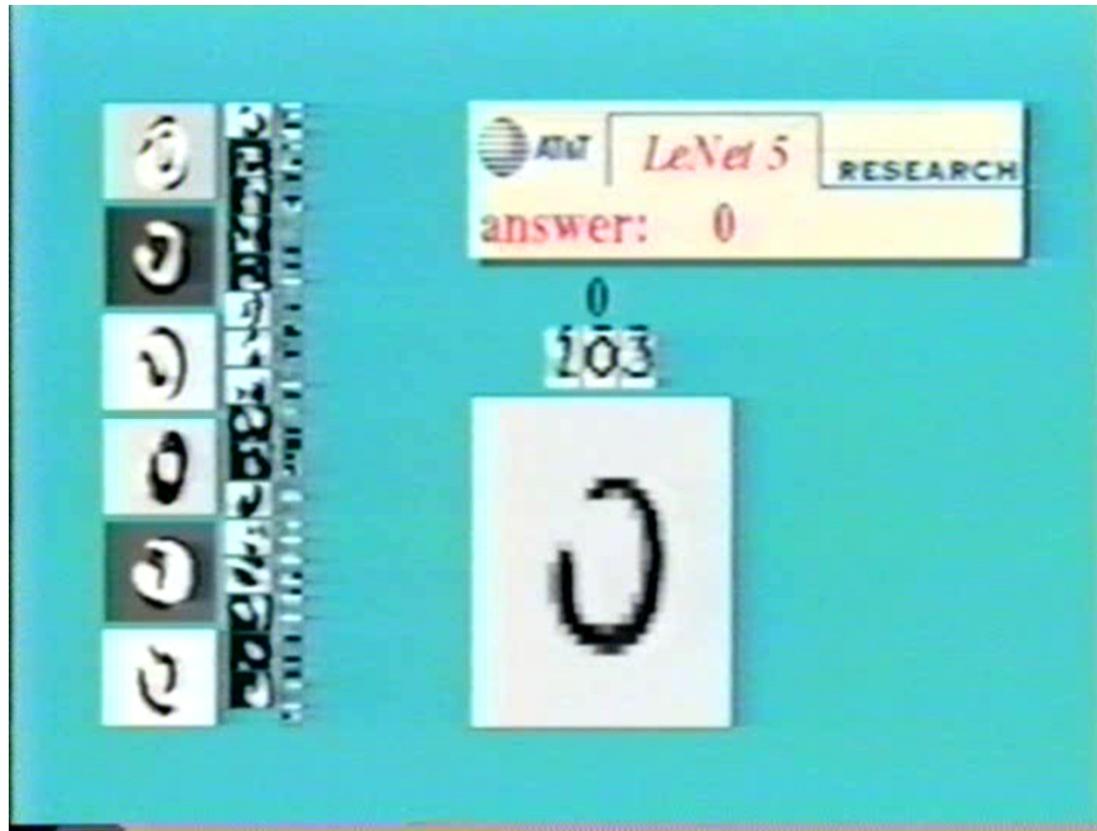


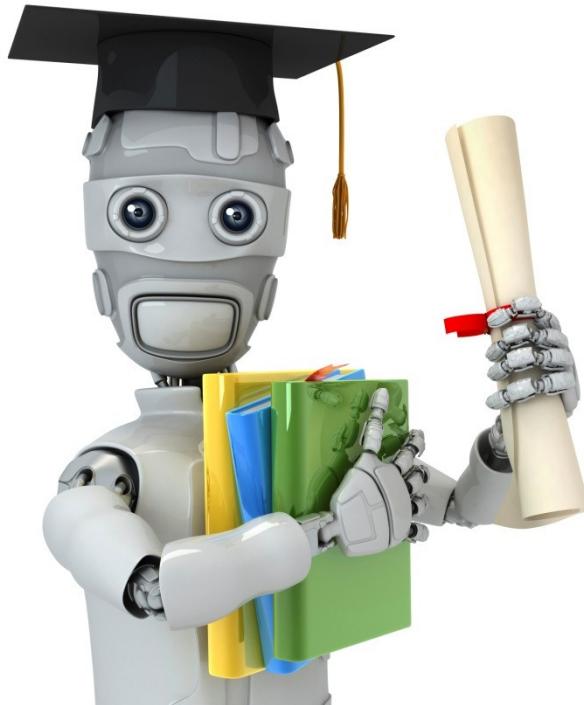
[Courtesy of Yann LeCun]



Andrew Ng

Handwritten digit classification





Machine Learning

Neural Networks: Representation

Multi-class classification

Multiple output units: One-vs-all.



Pedestrian



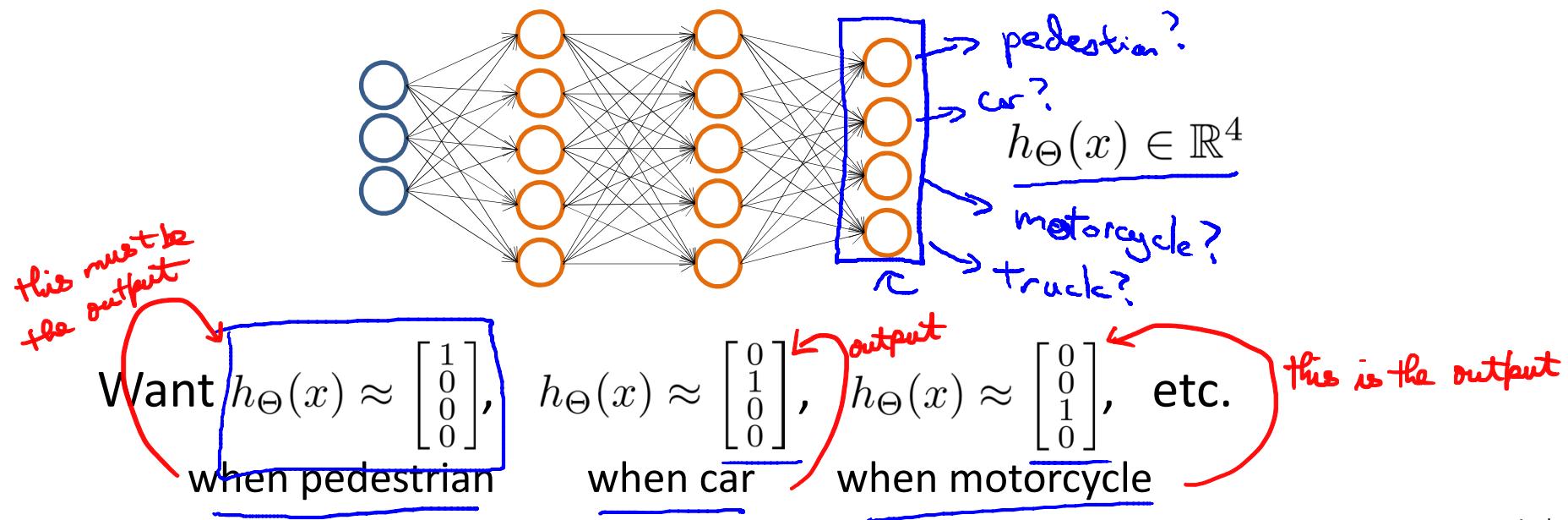
Car



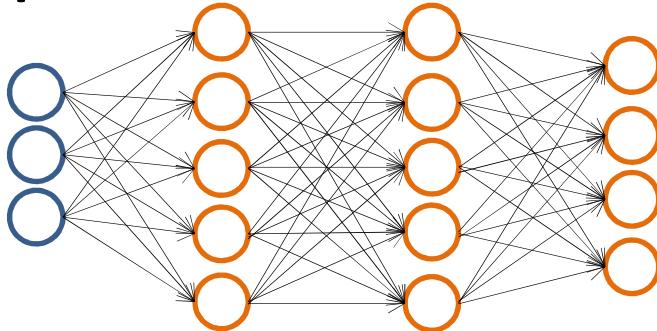
Motorcycle



Truck



Multiple output units: One-vs-all.



$$h_{\Theta}(x) \in \mathbb{R}^4$$

Want $h_{\Theta}(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, etc.
when pedestrian when car when motorcycle

Training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

→ $y^{(i)}$ one of
pedestrian car motorcycle truck

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

~~Previously~~
 $y \in \{1, 2, 3, 4\}$
 $\underline{h_{\Theta}(x^{(i)}) \approx y^{(i)}} \in \mathbb{R}^4$

