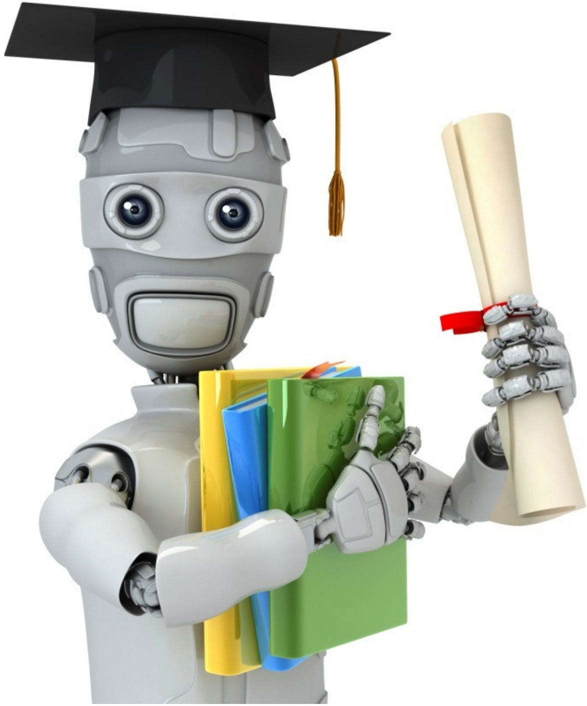
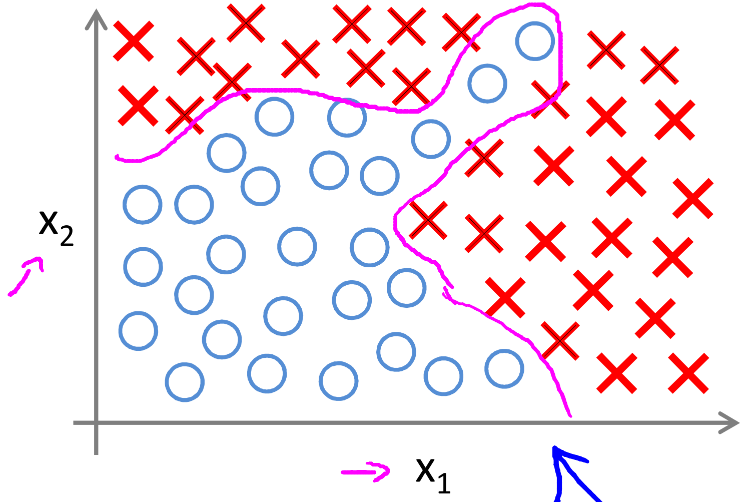


Machine Learning

Neural Networks: Representation

Non-linear hypotheses

Non-linear Classification



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

$x_1 =$ size
$x_2 =$ # bedrooms
$x_3 =$ # floors
$x_4 =$ age
...
x_{100}

$n = 100$

$\rightarrow x_1^2, x_1x_2, x_1x_3, x_1x_4 \ldots x_1x_{100}$
$x_2^2, x_2x_3 \ldots$

≈ 5000 feature $O(n^2)$

$\rightarrow x_1^2, x_2^2, x_3^2, \ldots, x_{100}^2$ $\approx \frac{n^2}{2}$

$\rightarrow x_1x_2x_3, x_1^2x_2, x_{10}x_{11}x_{17}, \ldots$

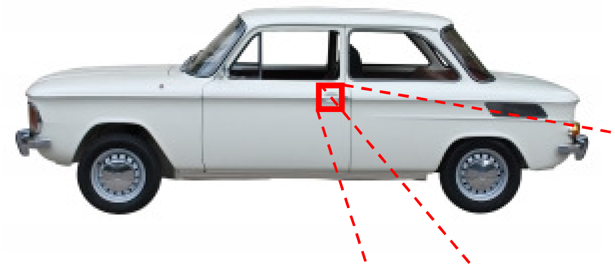
$O(n^3)$

170,000

Andrew Ng

What is this?

You see this:

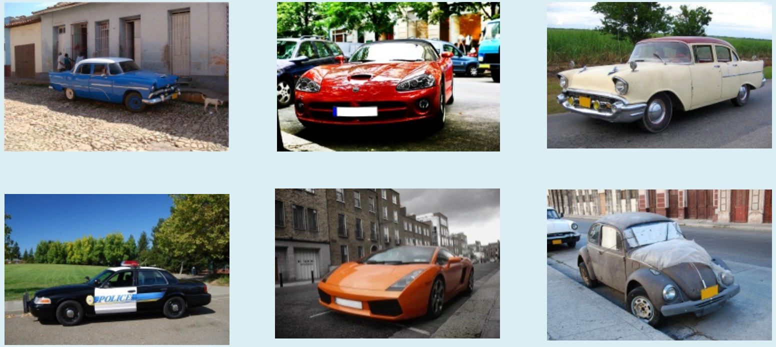


But the camera sees this:

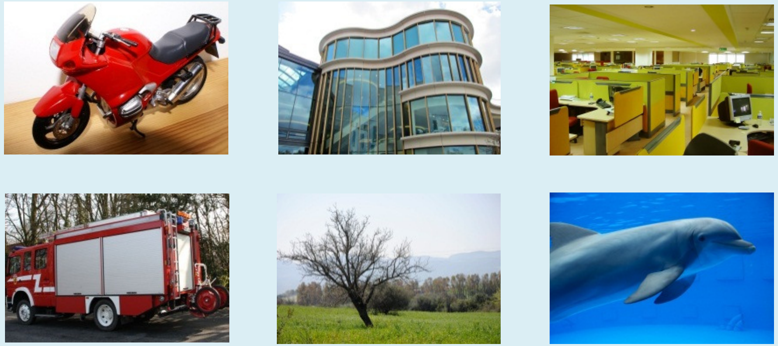
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	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

Andrew Ng

Computer Vision: Car detection



Cars

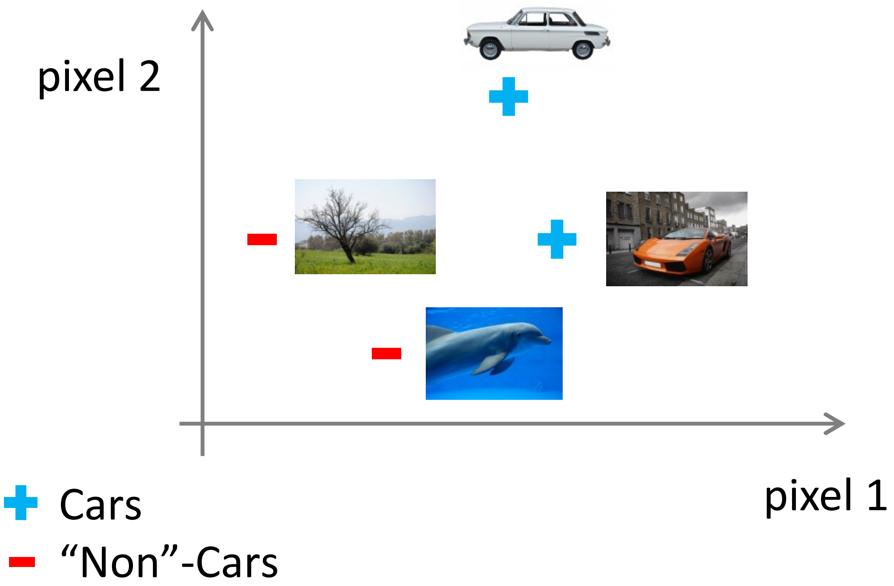


Not a car

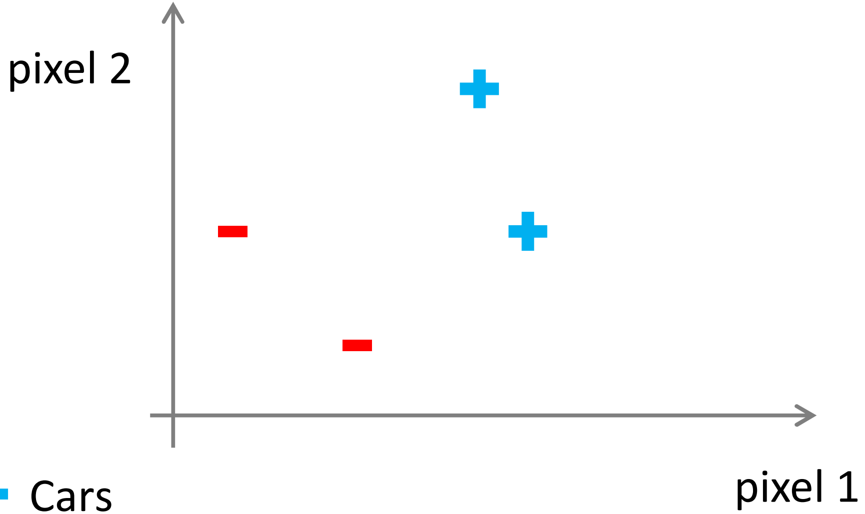
Testing:

What is this?

Learning Algorithm



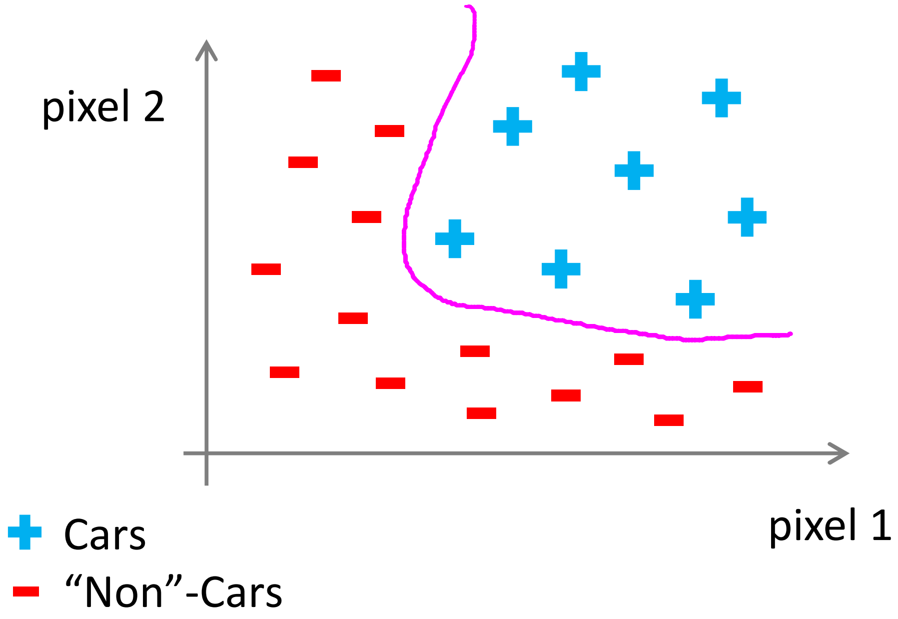
Learning Algorithm



+ Cars

– “Non”-Cars

Learning Algorithm



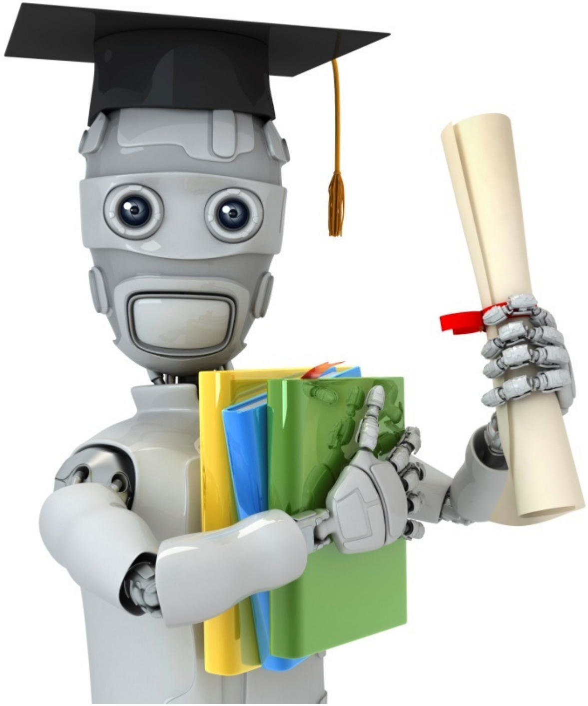
50 x 50 pixel images→ 2500 pixels

$\boxed{n = 2500}$ (7500 if RGB)

0-255

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Quadratic features ($x_i \times x_j$): ≈3 million features



Machine Learning

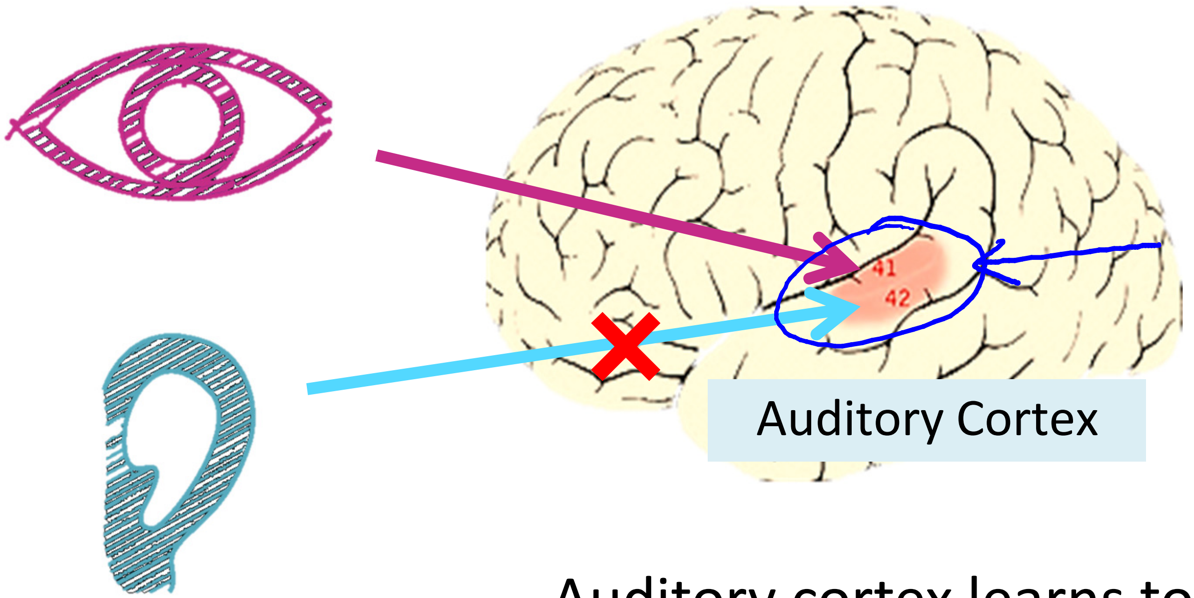
Neural Networks: Representation

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

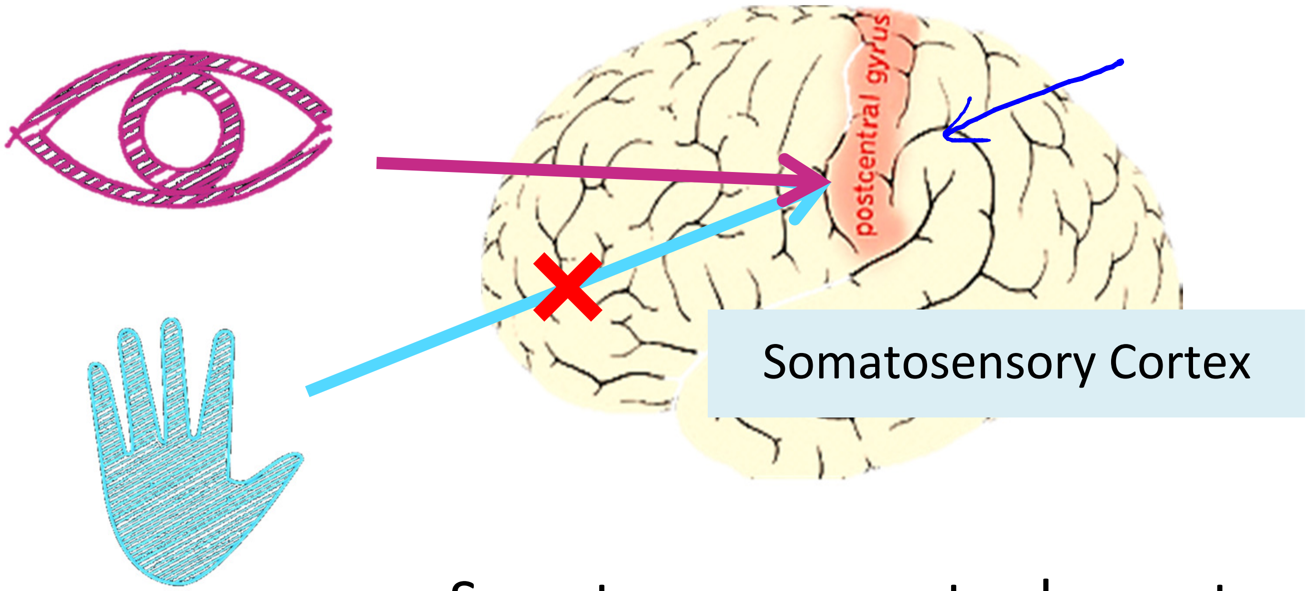
The "one learning algorithm" hypothesis



Auditory cortex learns to see

[Roe et al., 1992]

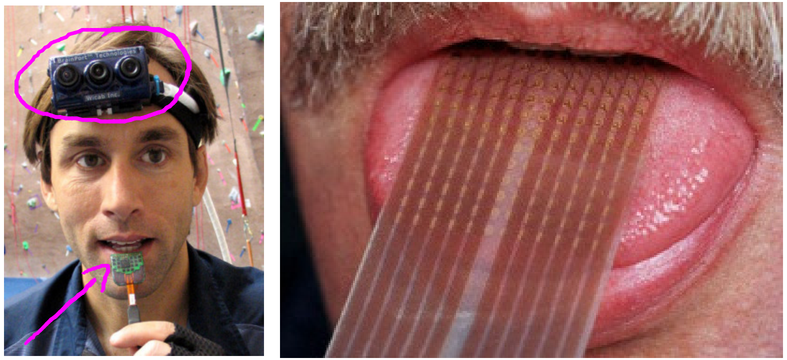
The "one learning algorithm" hypothesis



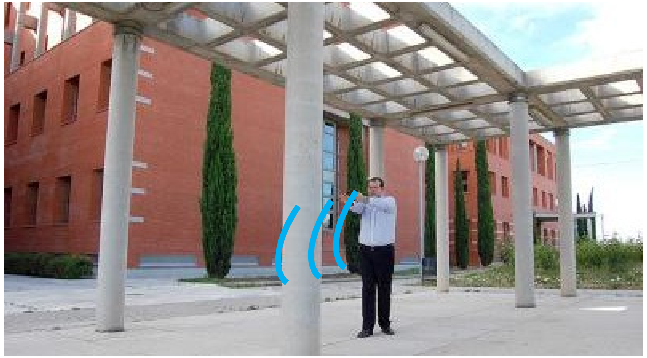
Somatosensory cortex learns to see

[Metin & Frost, 1989]

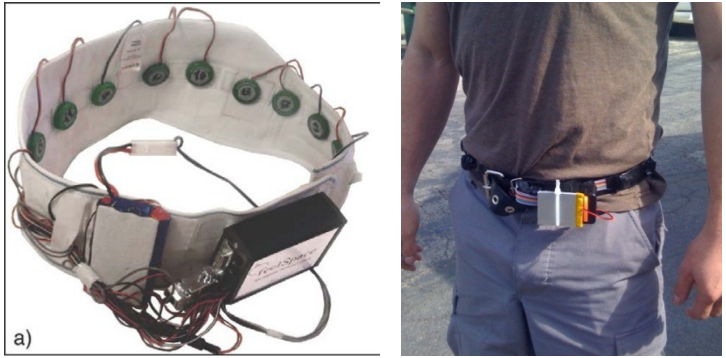
Sensor representations in the brain



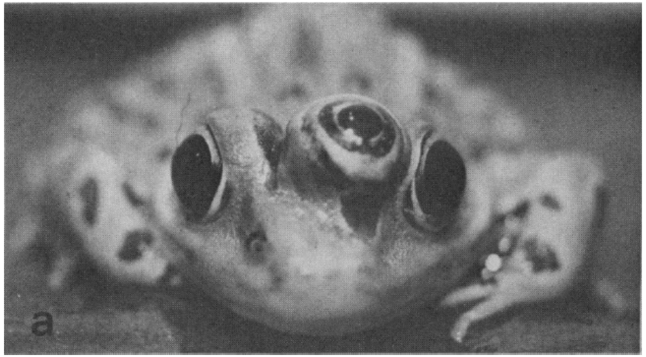
Seeing with your tongue



Human echolocation (sonar)

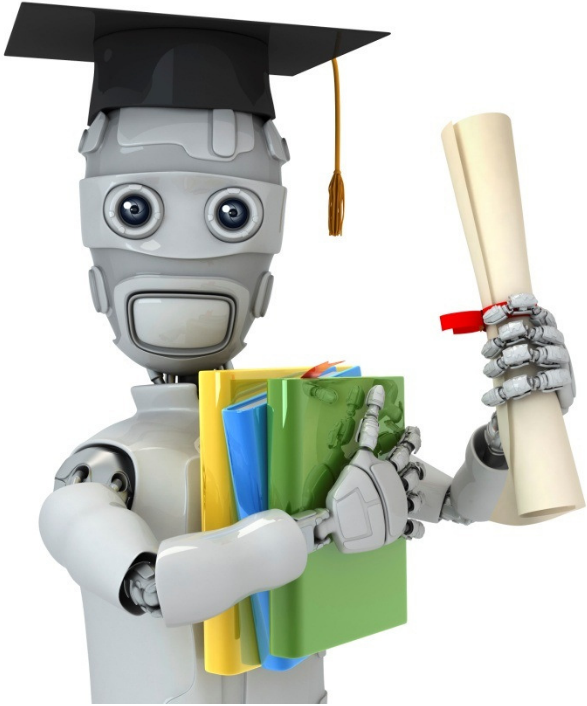


Haptic belt: Direction sense



Implanting a 3rd eye

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

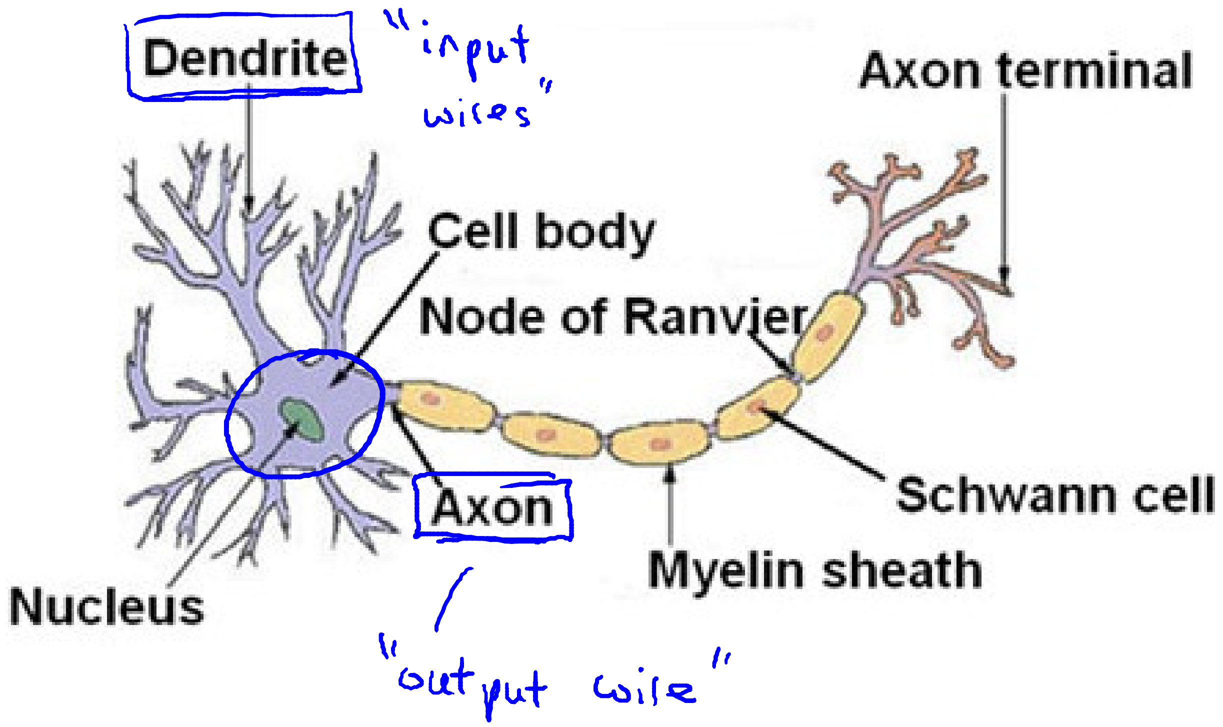


Machine Learning

Neural Networks: Representation

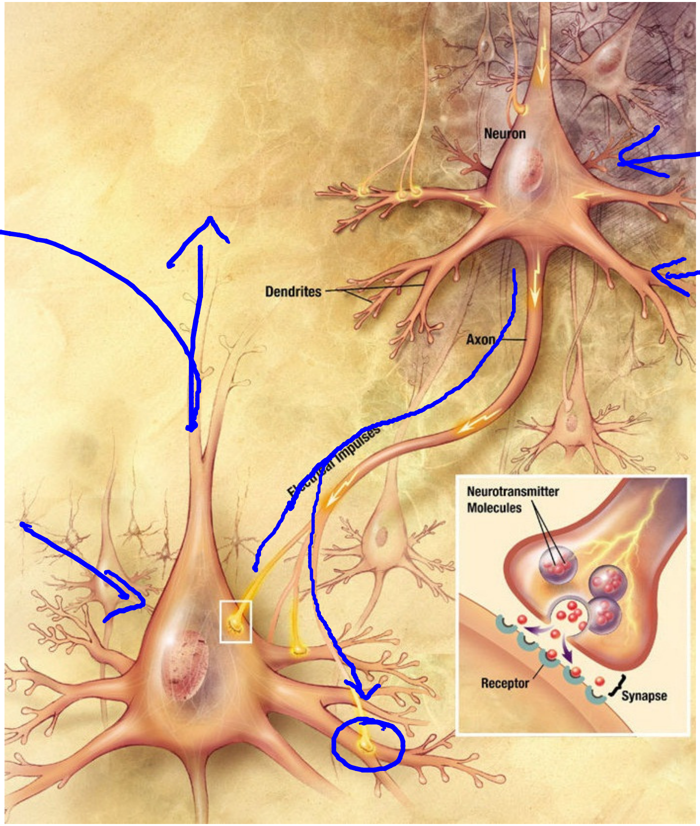
Model representation I

Neuron in the brain



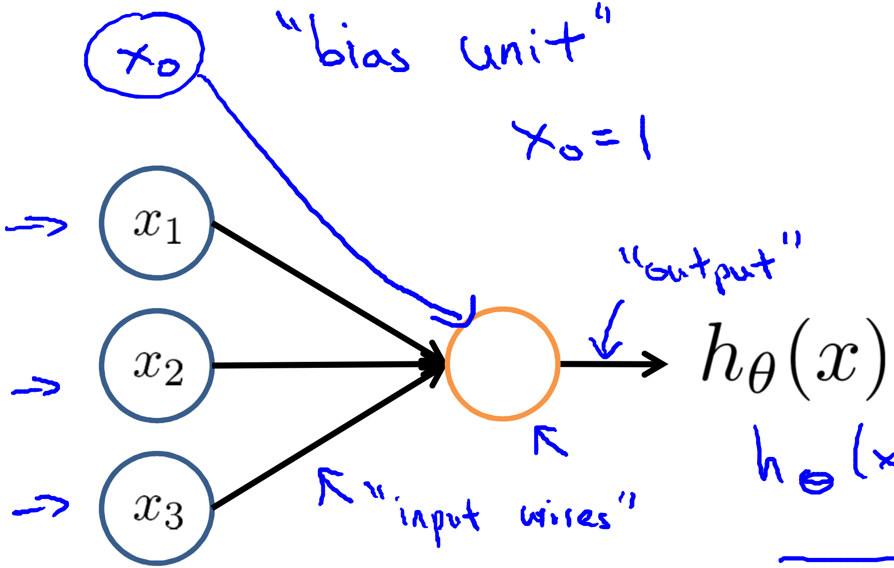
Neurons in the brain

muscle



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

Neuron model: Logistic unit



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

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

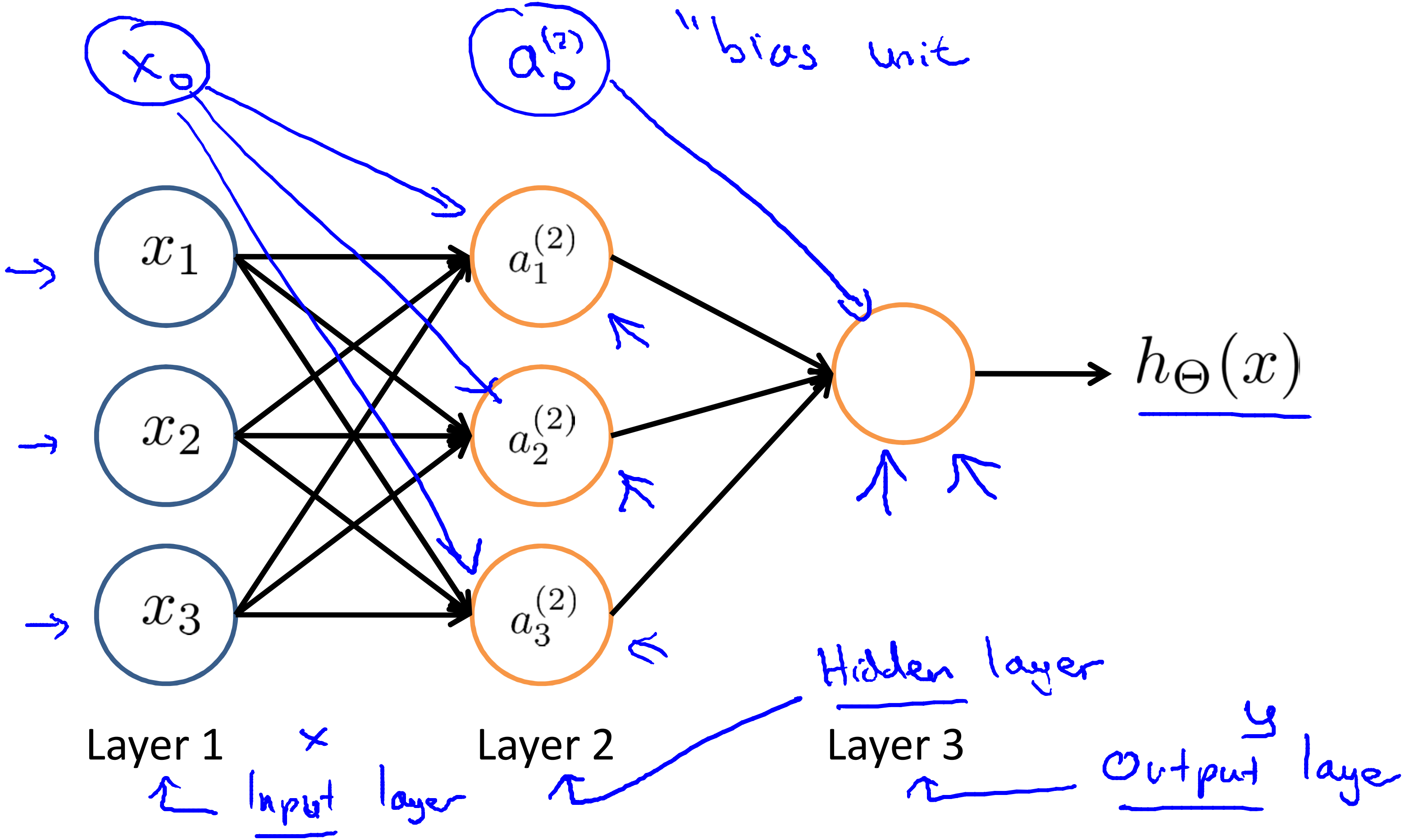
"weights"

(parameters

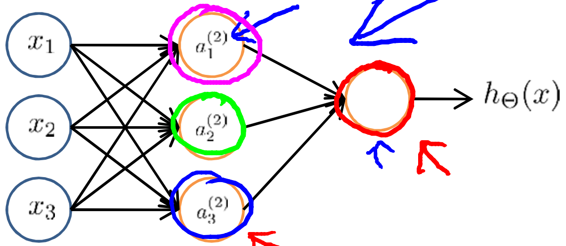
Sigmoid (logistic) activation function.

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

Neural Network



Neural Network



3 units

3 hiddn units

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

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

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

$h_\Theta(x)$

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

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

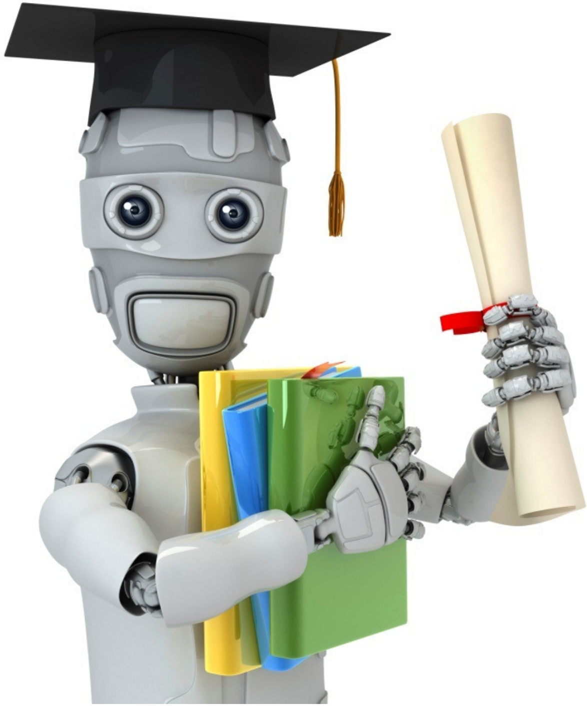
$$\rightarrow a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$

$\Theta^{(2)}$

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

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

$S_{j+1} \times (S_j + 1)$

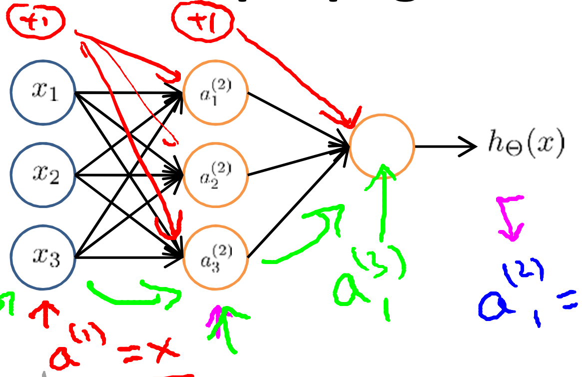


Machine Learning

Neural Networks: Representation

Model representation II

Forward propagation: Vectorized implementation



$a_1^{(2)} = g(z_1^{(2)})$

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$
$$h_\Theta(x) = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

$z_1^{(2)}$ $z_2^{(2)}$ $z_3^{(2)}$ $z^{(3)}$

$\Theta^{(1)}$ x

$a_3^{(2)} = g(z_3^{(2)})$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$z^{(2)} = \Theta^{(1)} \cancel{x}\, a^{(1)}$$
$$\boxed{a^{(2)}} = g(z^{(2)})$$

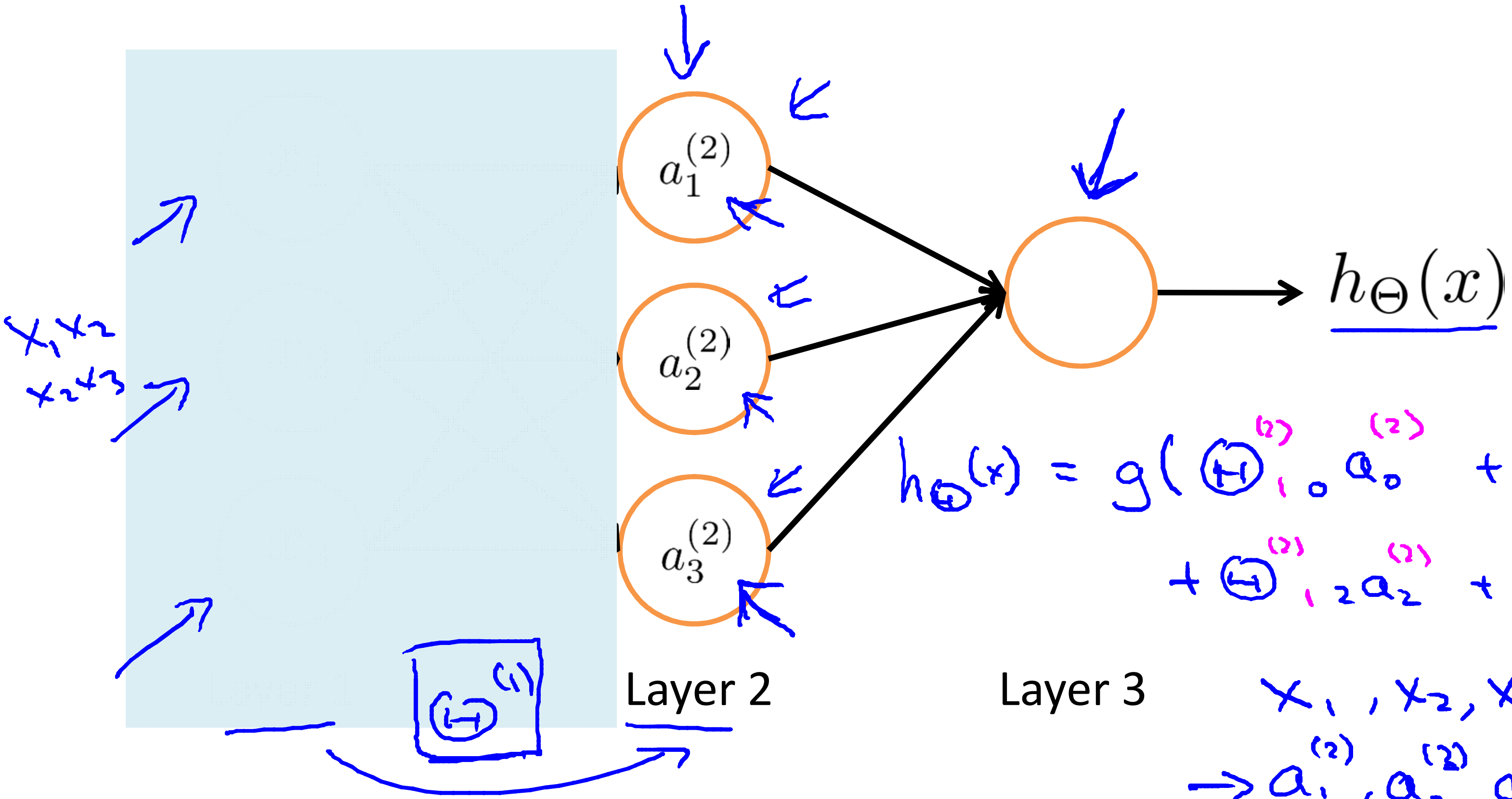
$\mathbb{R}^3$ $\mathbb{R}^3$

$a_1^{(2)}$ $a_2^{(2)}$ $a_3^{(2)}$

Add $a_0^{(2)} = 1$. $\rightarrow a^{(2)} \in \mathbb{R}^4$

$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$h_\Theta(x) = \boxed{a^{(3)}} = g(z^{(3)})$$

Neural Network learning its own features

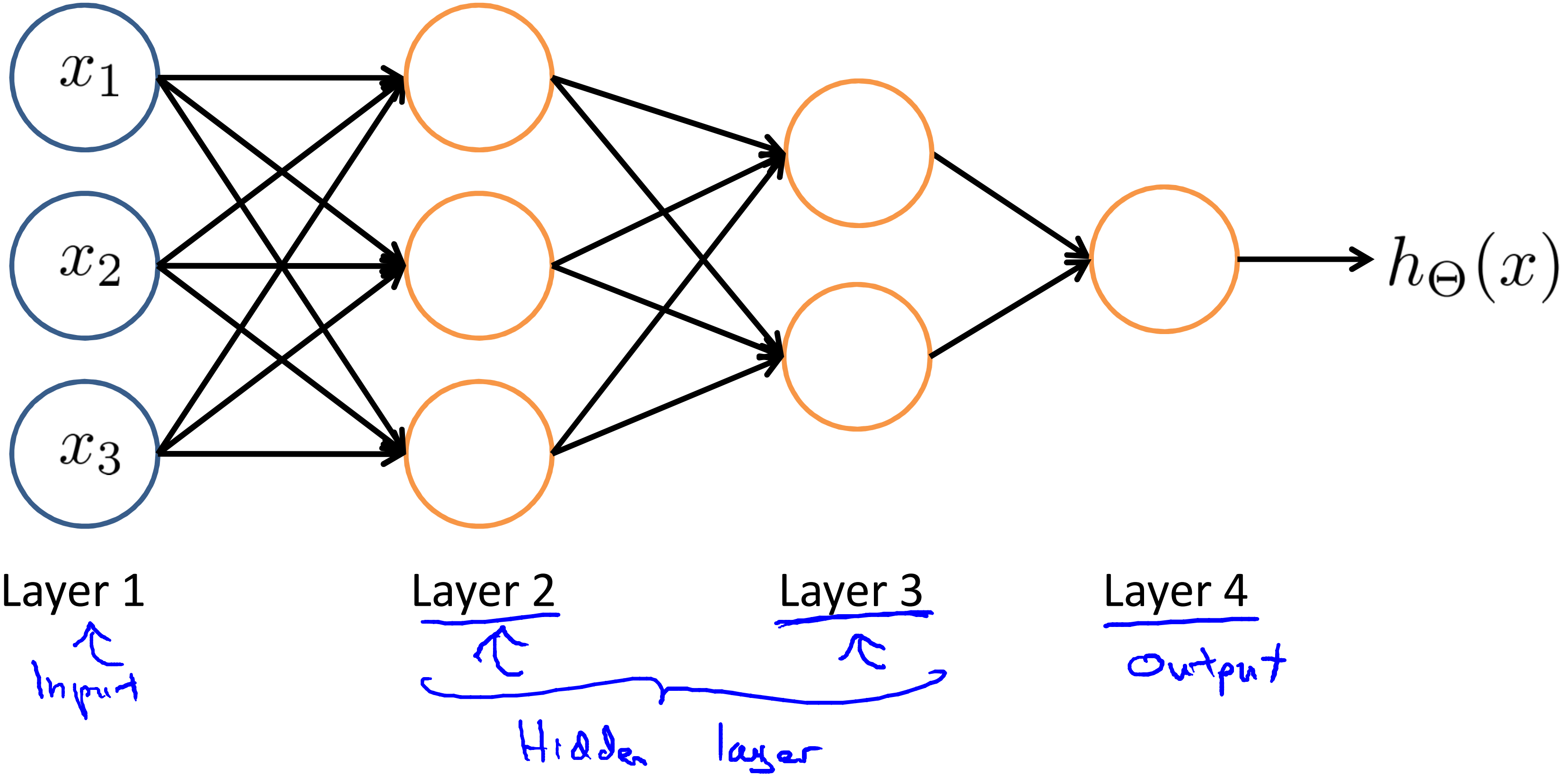


$$h_\Theta(x) = g(\Theta^{(2)}_{10} a_0^{(2)} + \Theta^{(1)}_{11} a_1^{(2)} + \Theta^{(2)}_{12} a_2^{(2)} + \Theta^{(2)}_{13} a_3^{(2)})$$

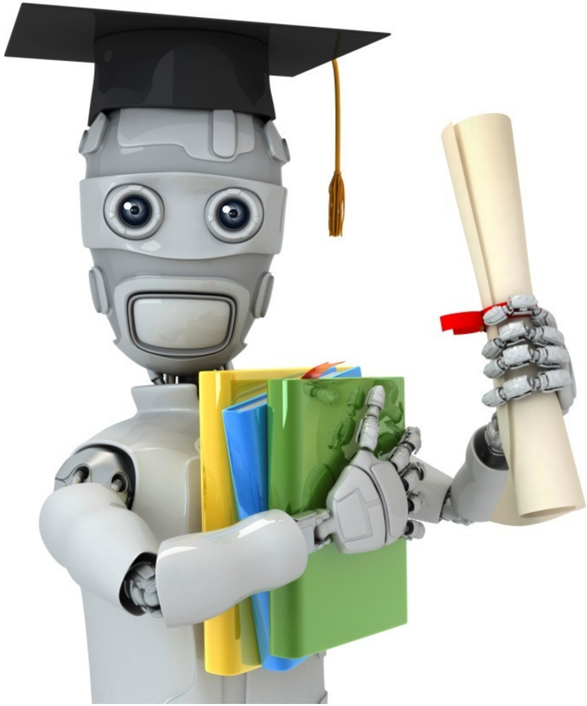
x_1, x_2, x_3

$\rightarrow a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$

Other network architectures



Andrew Ng



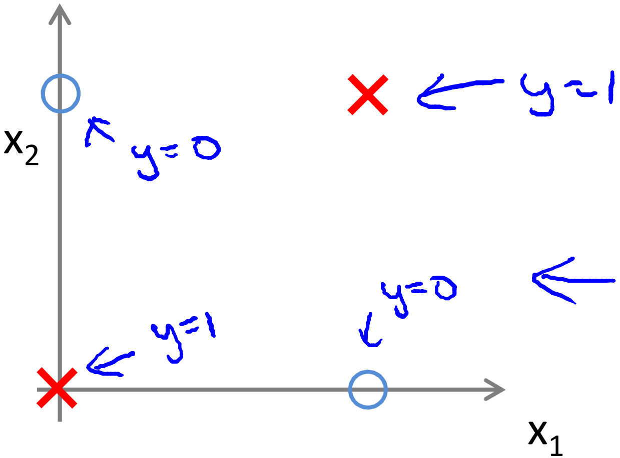
Machine Learning

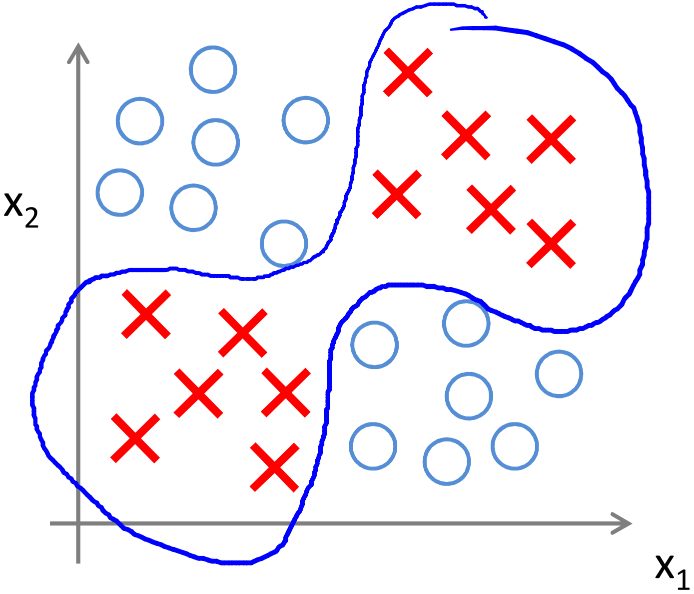
Neural Networks: Representation

Examples and intuitions I

Non-linear classification example: XOR/XNOR

x_1, x_2 are binary (0 or 1).





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

$$x_1 \text{ XNOR } x_2$$

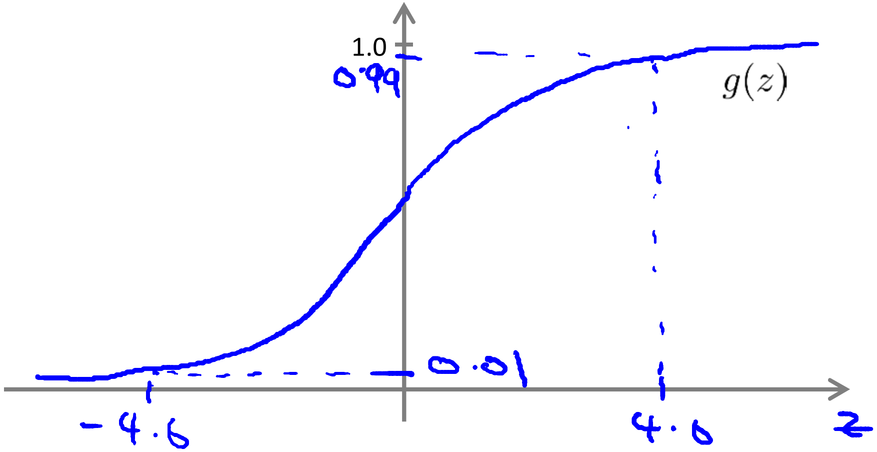
$$\text{NOT } (x_1 \text{ XOR } x_2)$$

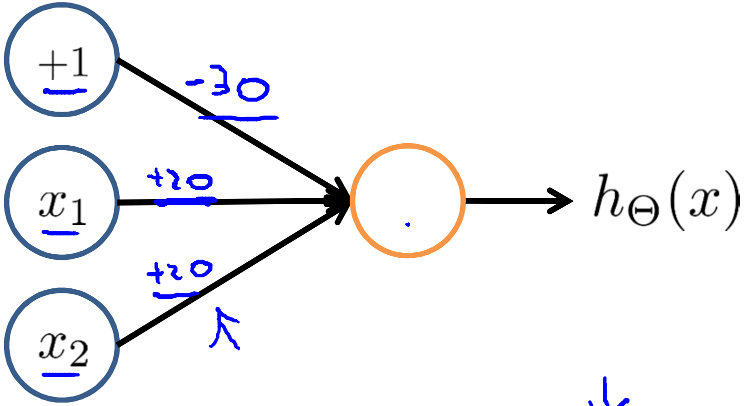
Andrew Ng

Simple example: AND

→ $x_1, x_2 \in \{0, 1\}$

→ $y = x_1 \text{ AND } x_2$



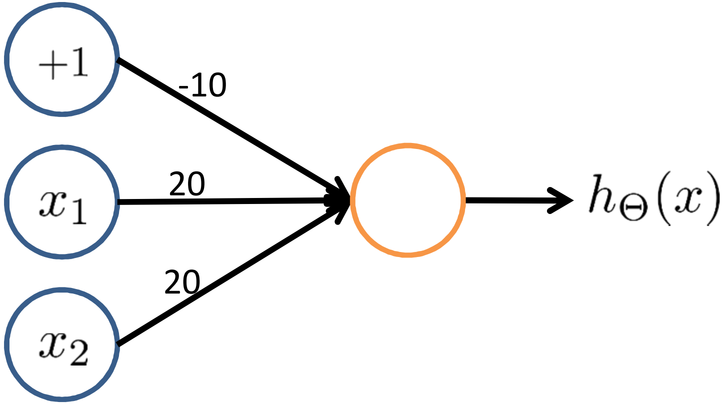


→ $h_\Theta(x) = g(\underset{\Theta^{(1)}_{10}}{-30} + \underset{\Theta^{(1)}_{11}}{20}x_1 + \underset{\Theta^{(1)}_{12}}{20}x_2)$

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	$g(10) \approx 1$

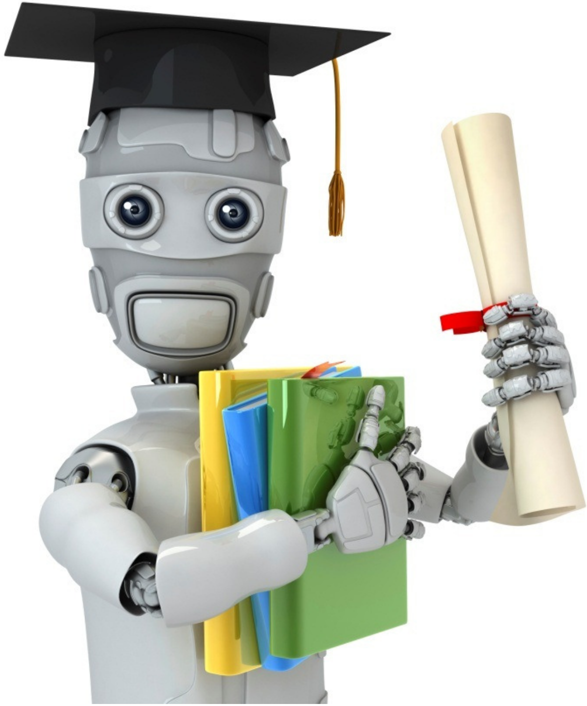
$h_\Theta(x) \approx x_1 \text{ AND } x_2$

Example: OR function



$g(-10 + 20x_1 + 20x_2)$

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

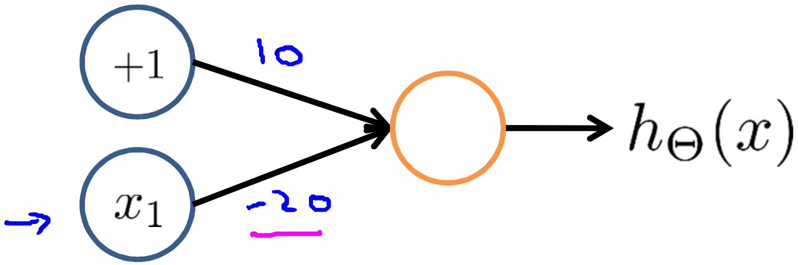
Neural Networks: Representation

Examples and intuitions II

$\rightarrow x_1$ AND x_2 $\qquad\qquad$ $\rightarrow x_1$ OR x_2 $\qquad$ $\{0,1\}$.

Negation:

NOT x_1



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

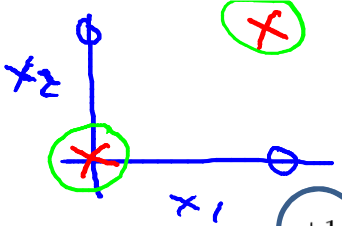
$$h_\Theta(x) = g(10 - 20x_1)$$

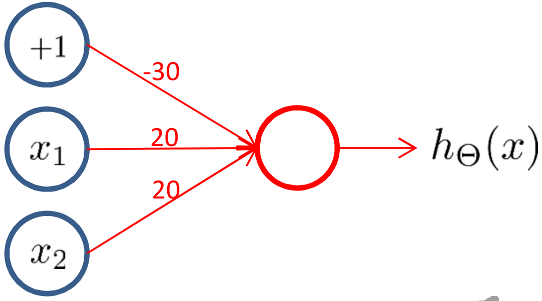
$\rightarrow$ (NOT x_1) AND (NOT x_2)

$(=1$ if and only if

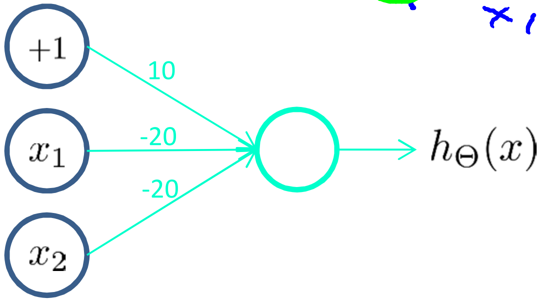
$\rightarrow x_1 = x_2 = 0$

Putting it together: $x_1 \text{ XNOR } x_2$

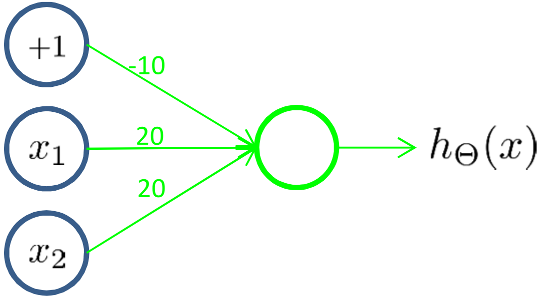




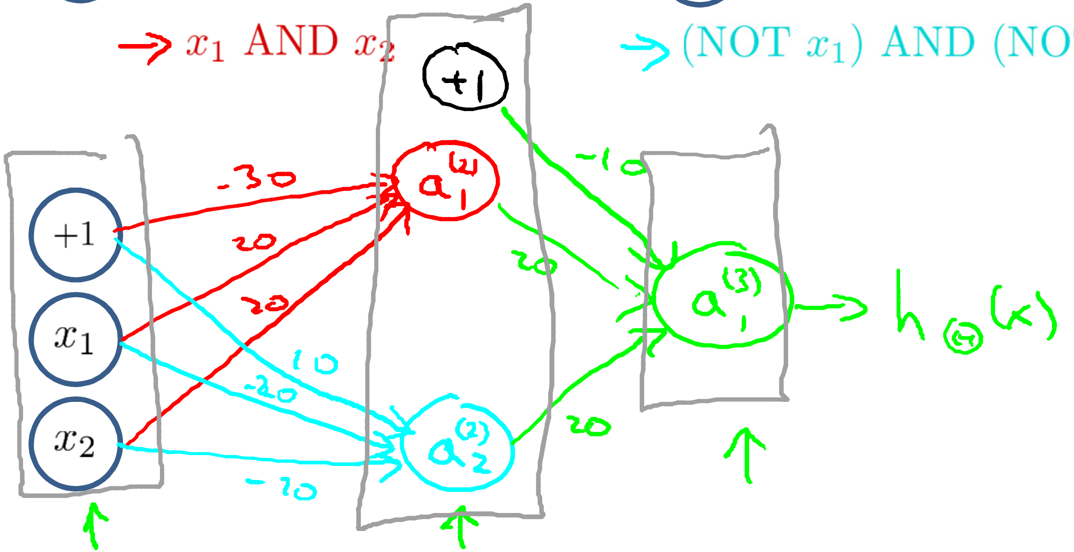
$\rightarrow x_1 \text{ AND } x_2$



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



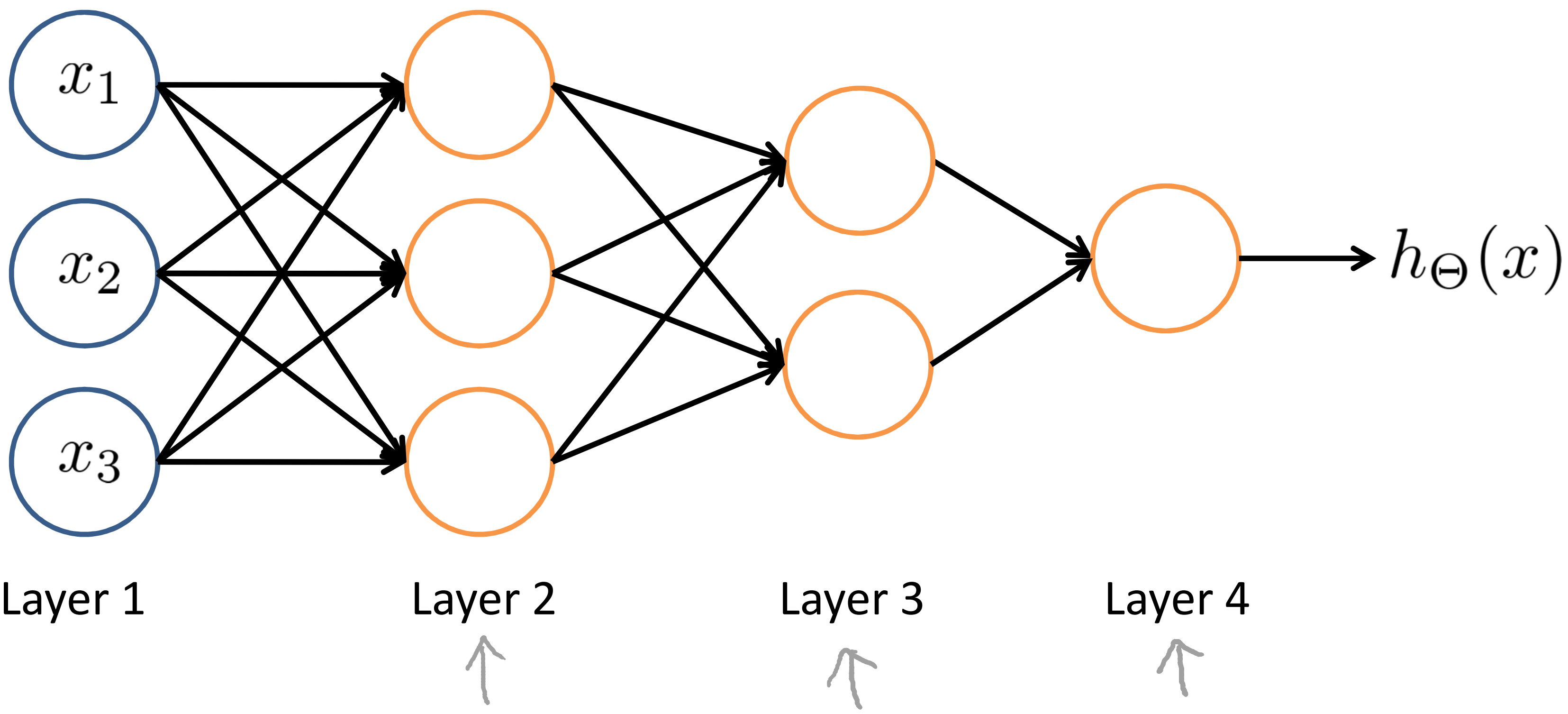
$\rightarrow x_1 \text{ OR } x_2$



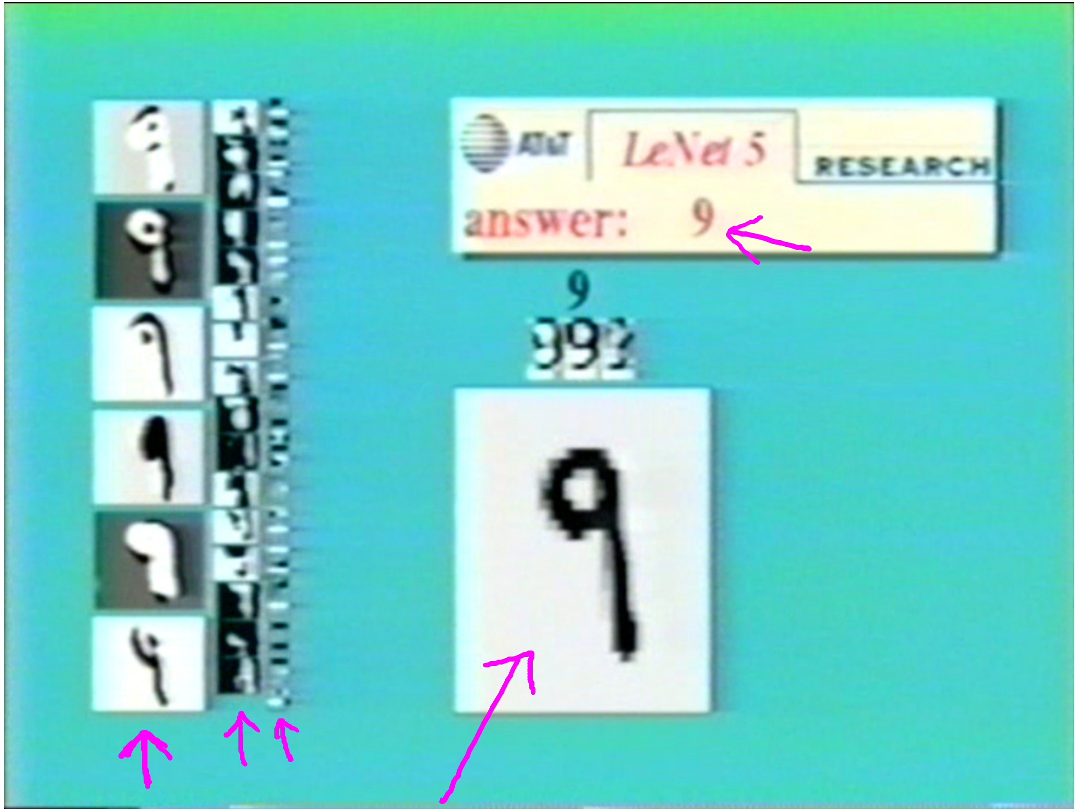
x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Andrew Ng

Neural Network intuition

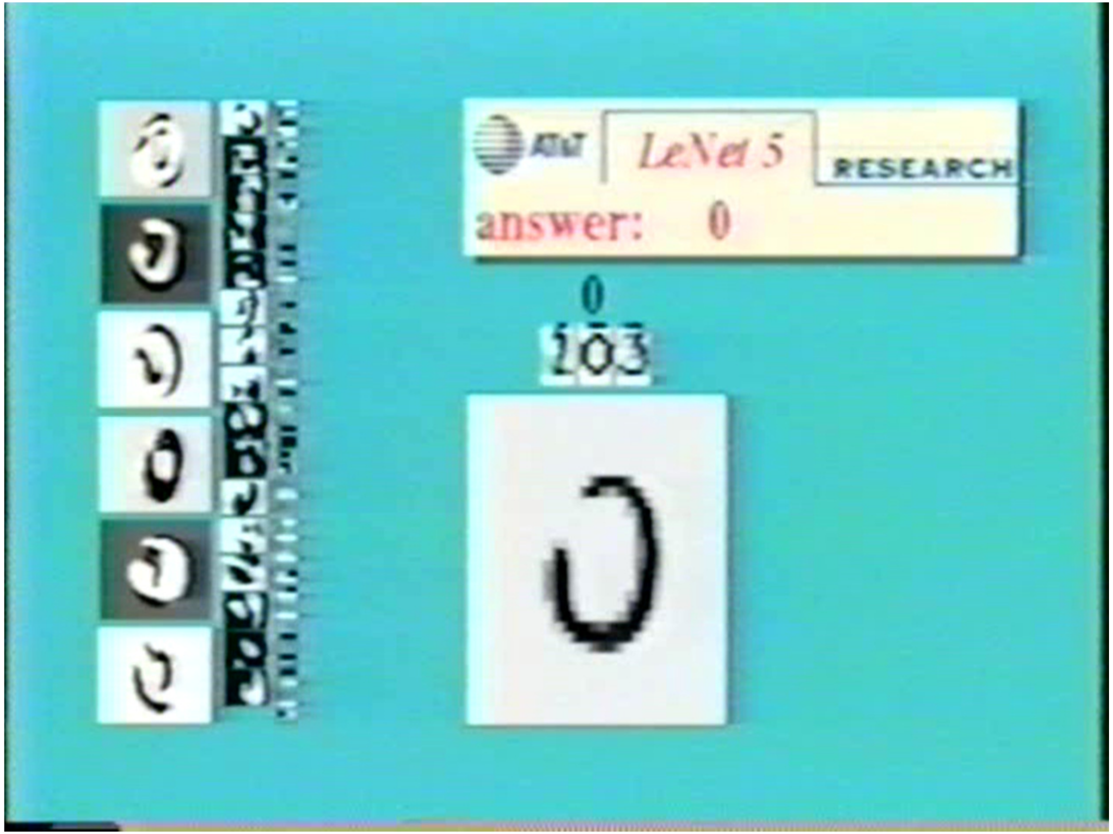


Handwritten digit classification

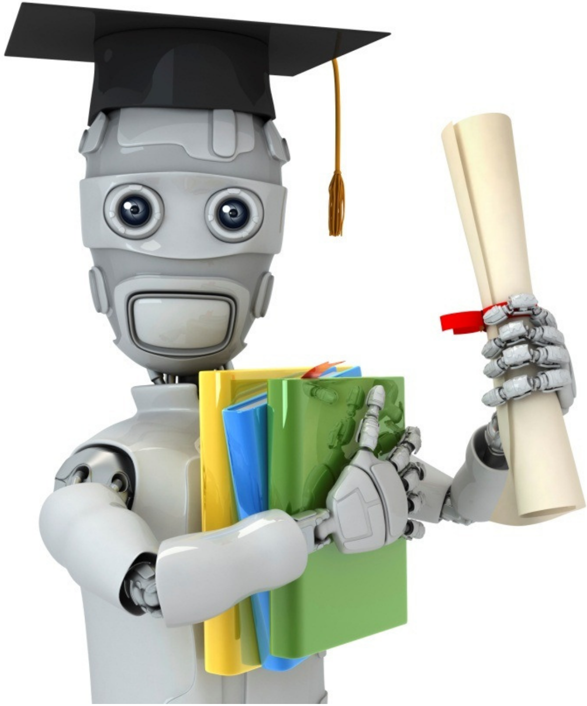


[Courtesy of Yann LeCun]

Handwritten digit classification



[Courtesy of Yann LeCun]



Machine Learning

Neural Networks: Representation

Multi-class classification

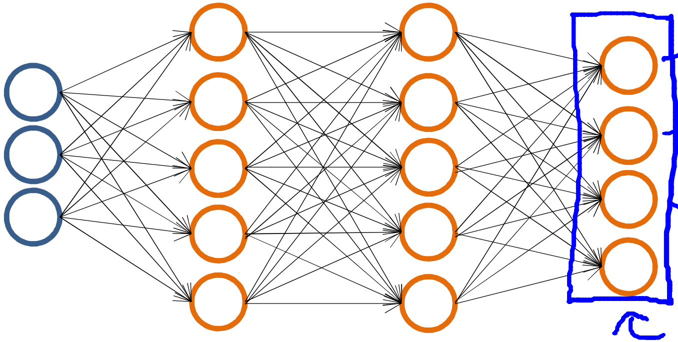
Multiple output units: One-vs-all.

Pedestrian

Car

Motorcycle

Truck



pedestrian?

car?

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

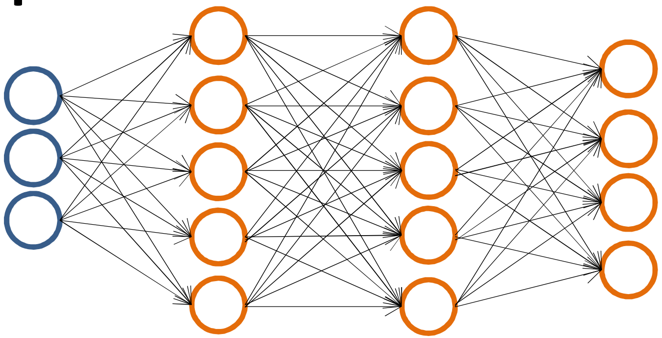
motorcycle?

truck?

Want $\boxed{h_\Theta(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, etc.

when pedestrian   when car   when motorcycle

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

pedestrian car motorcycle truck

$(x^{(i)}, y^{(i)})$

~~Previously~~

$\cancel{y \in \{1,2,3,4\}}$

$\underbrace{h_\Theta(x^{(i)})}_{\mathbb{R}^4} \approx \underline{y^{(i)}}$