Machine Learning Study Week 4

Neural Networks: Representation

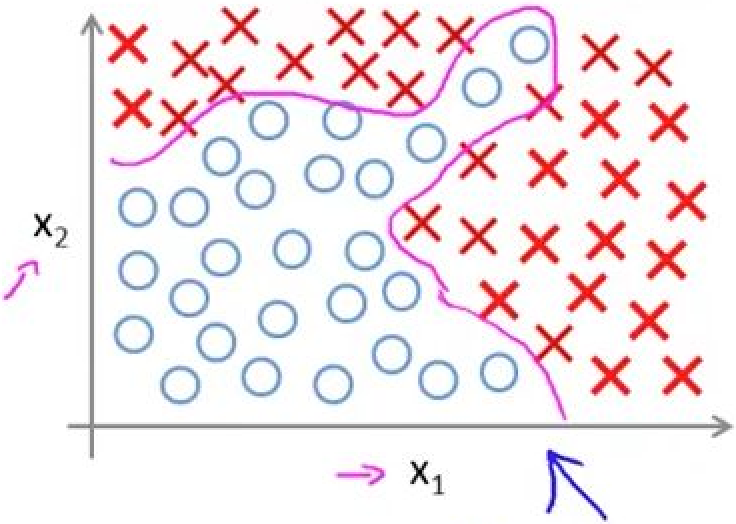
구 본 규 (protocolstack9@gmail.com)

Machine Learning

Neural Networks: Representation

Non-linear hypotheses

Non-linear Classification



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

$\rightarrow x_1^2, x_1x_2, x_1x_3, x_1x_4 \dots x_1x_{100}$

$x_2^2, x_2x_3 \dots$

$\approx$ 5000 feature $\quad O(n^2)$

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

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

$O(n^3)$

170,000

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

$n = 100$

10

Computer Vision: Car detection

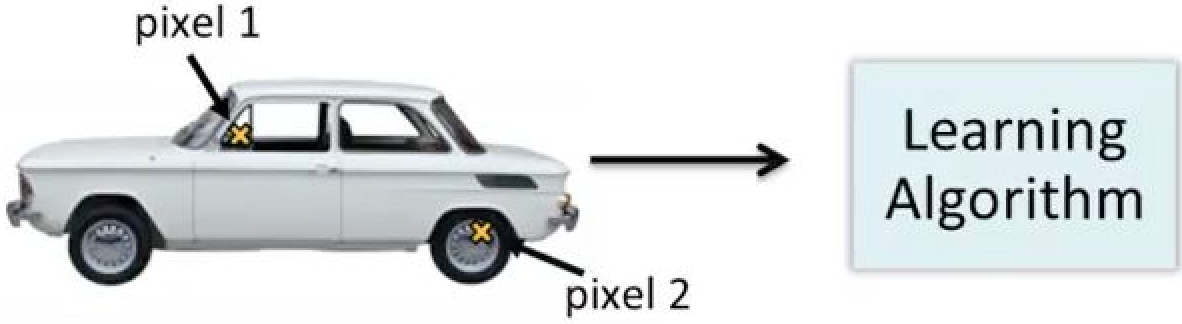
Cars

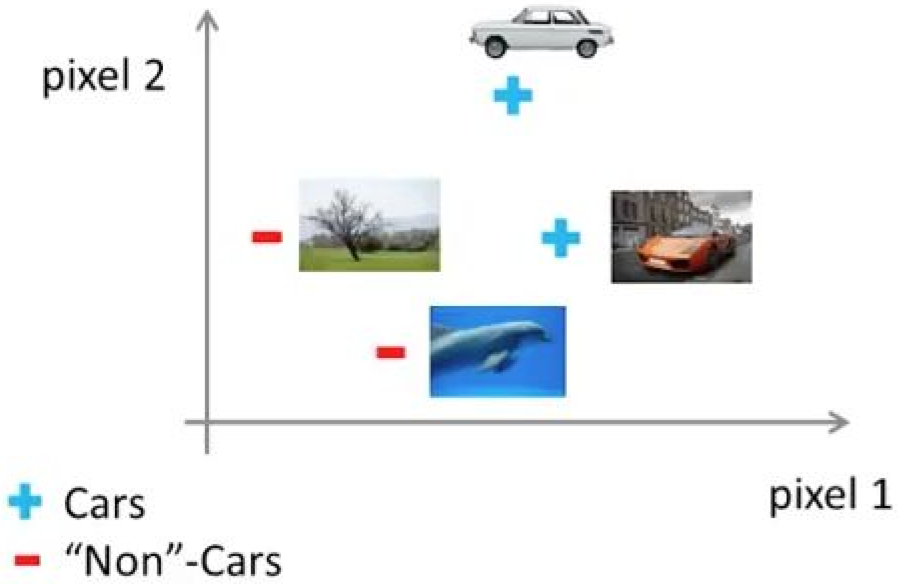
Not a car

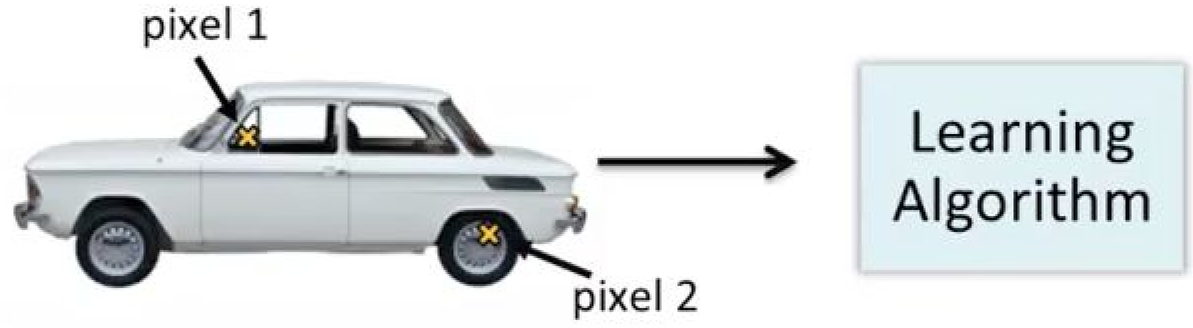
label example

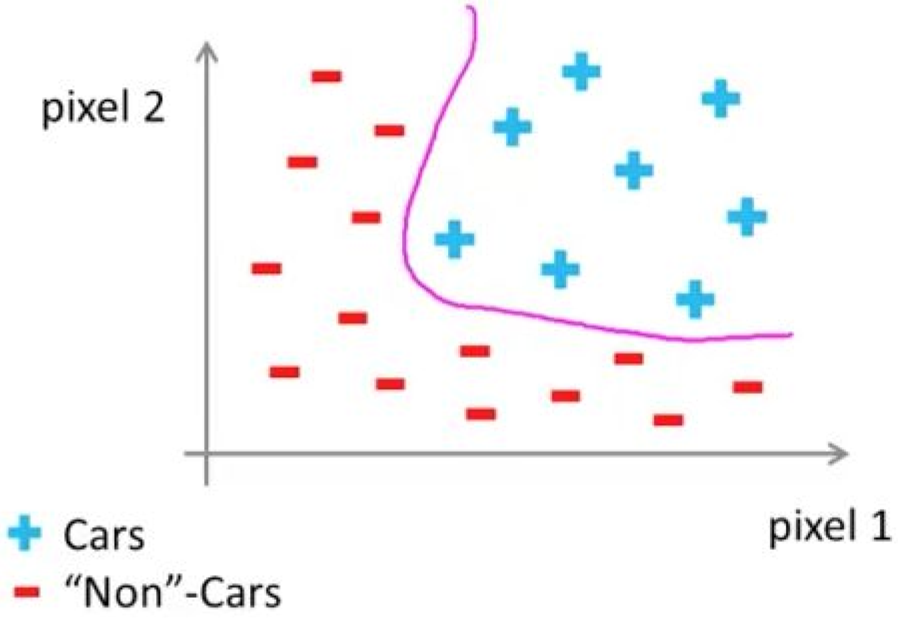
Testing: 

What is this?


pixel 1
pixel 2
Learning
Algorithm


pixel 2
pixel 1
Cars
"Non"-Cars





50 x 50 pixel images→ 2500 pixels

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

0-255

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

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

Machine Learning

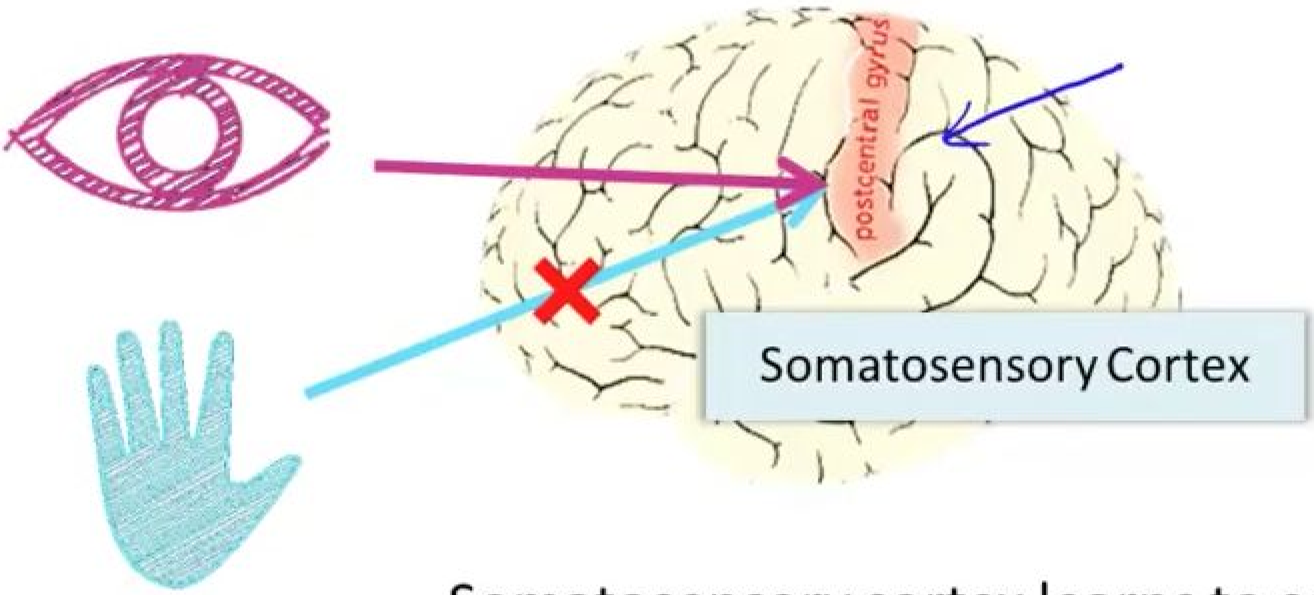
Neural Networks: Representation

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



Somatosensory cortex learns to see

Sensor representations in the brain

Seeing with your tongue

Human echolocation (sonar)

Haptic belt: Direction sense

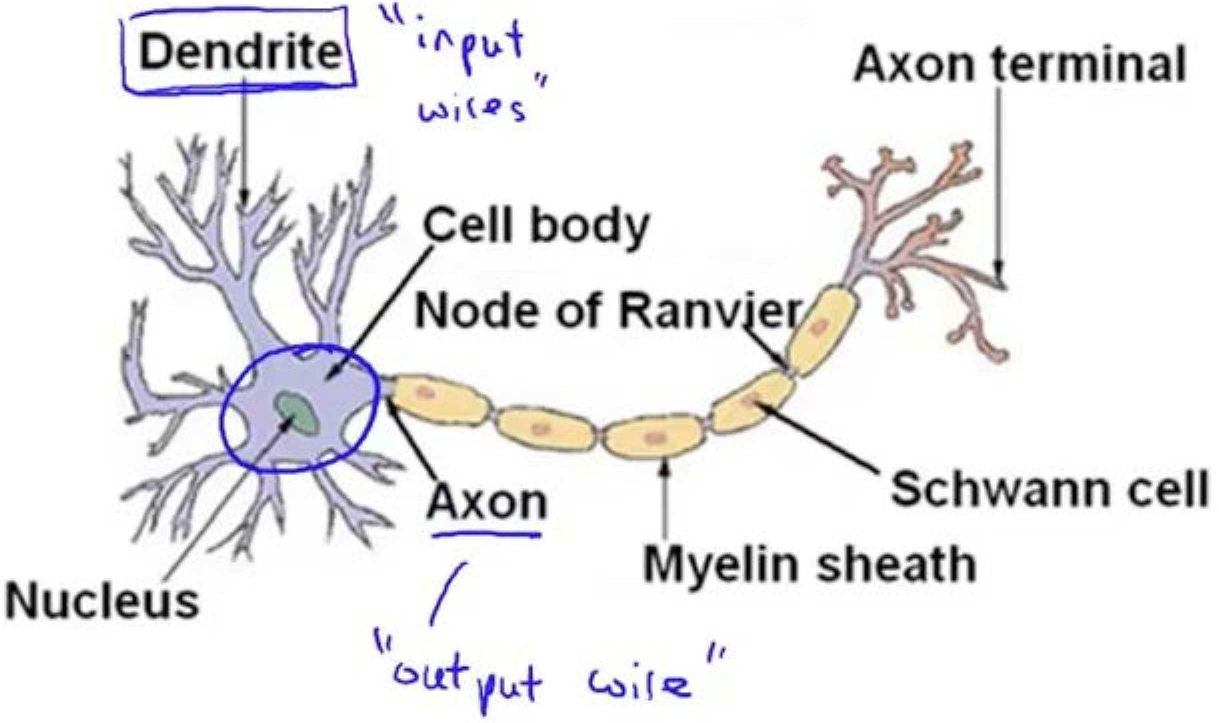
Implanting a 3rd eye

Machine Learning

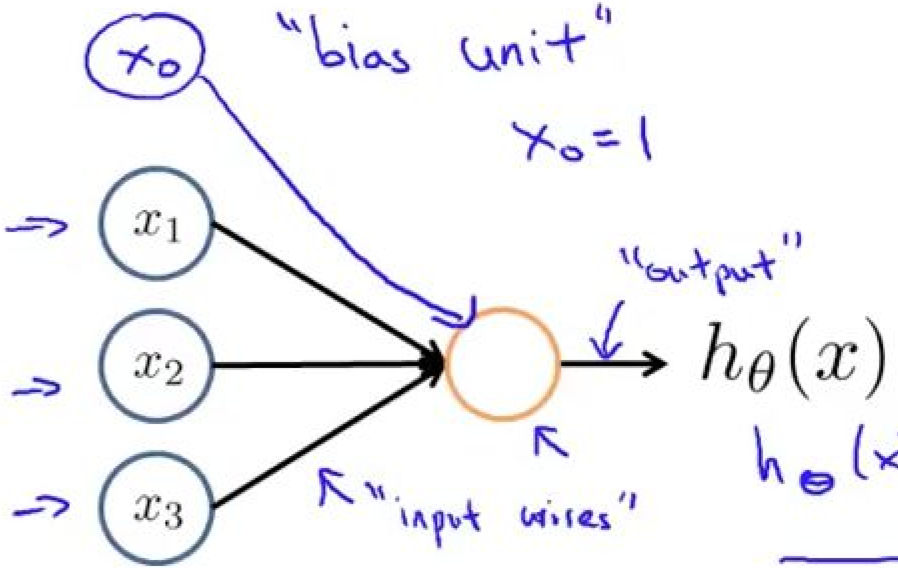
Neural Networks: Representation

Model representation I

Neuron in the brain



Neuron model: Logistic unit



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

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

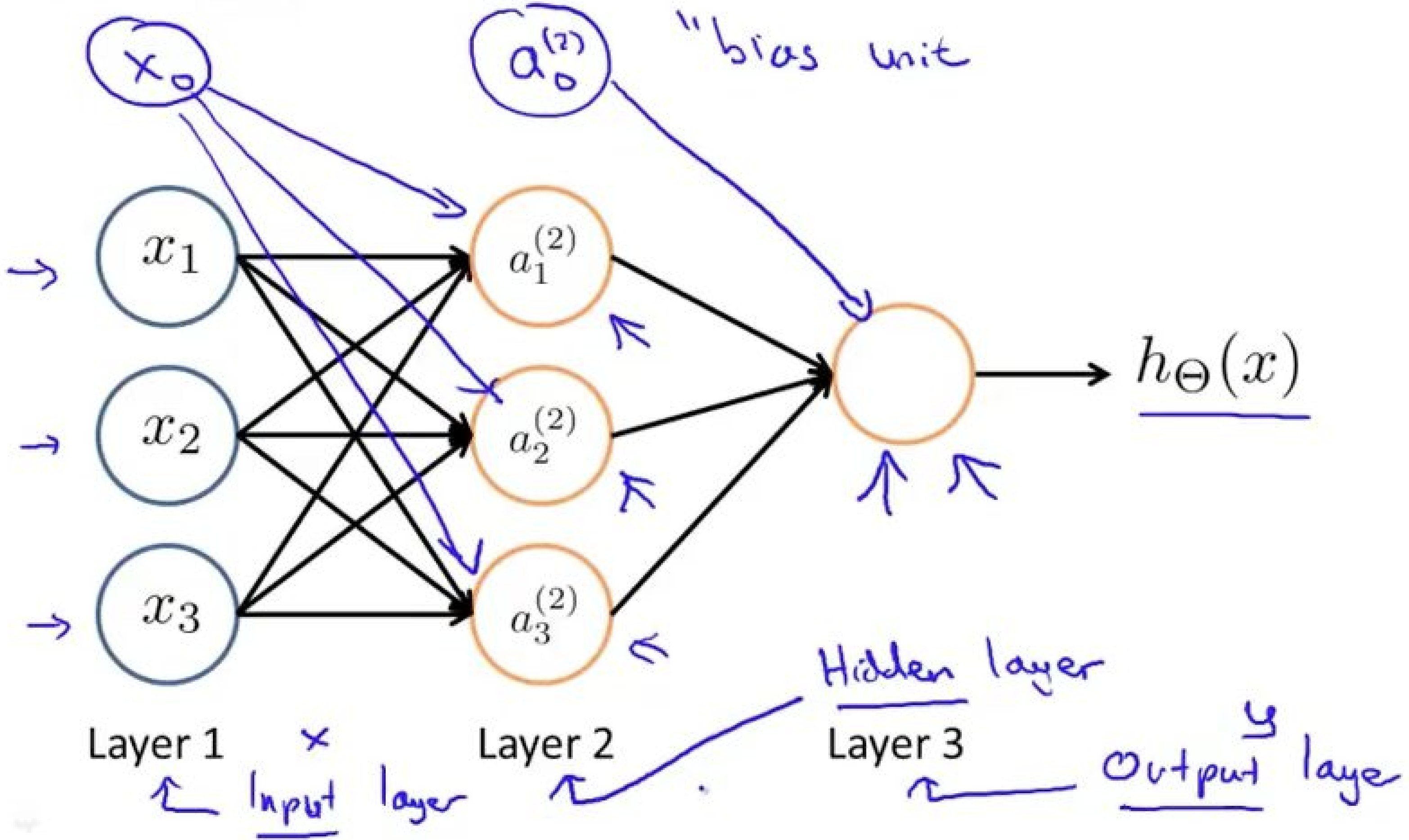
"weights"

(parameters

Sigmoid (logistic) activation function.

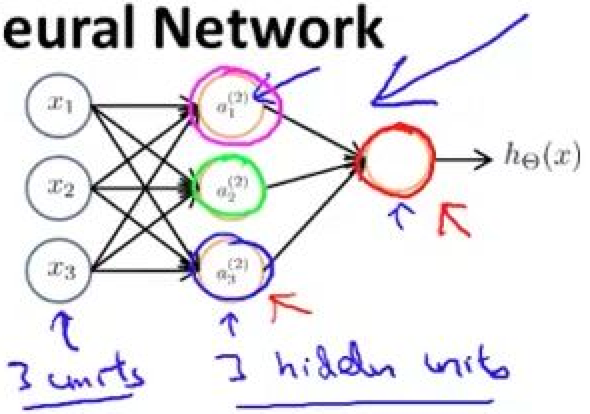
$$g(z) = \frac{1}{1+e^{-z}}$$

Neural Network



Andrew

Neural Network



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

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

$\Theta^{(1)} \in \mathbb{R}^{3\times4}$

$h_\Theta(x)$

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

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

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

$\Theta^{(2)}$

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

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

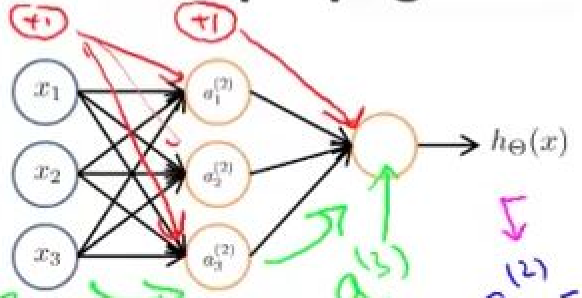
$S_{j+1} \times (S_j + 1)$

Machine Learning

Neural Networks: Representation

Model representation II

Forward propagation: Vectorized implementation



$a^{(1)} = x$

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

$$a_1^{(2)} = g(\underbrace{\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3}_{z_1^{(2)}})$$

$$a_2^{(2)} = g(\underbrace{\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3}_{z_2^{(2)}})$$

$$a_3^{(2)} = g(\underbrace{\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3}_{z_3^{(2)}})$$

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

$\Theta^{(1)}$ x

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

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

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

$\boxed{a^{(2)}} = g(z^{(2)})$

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

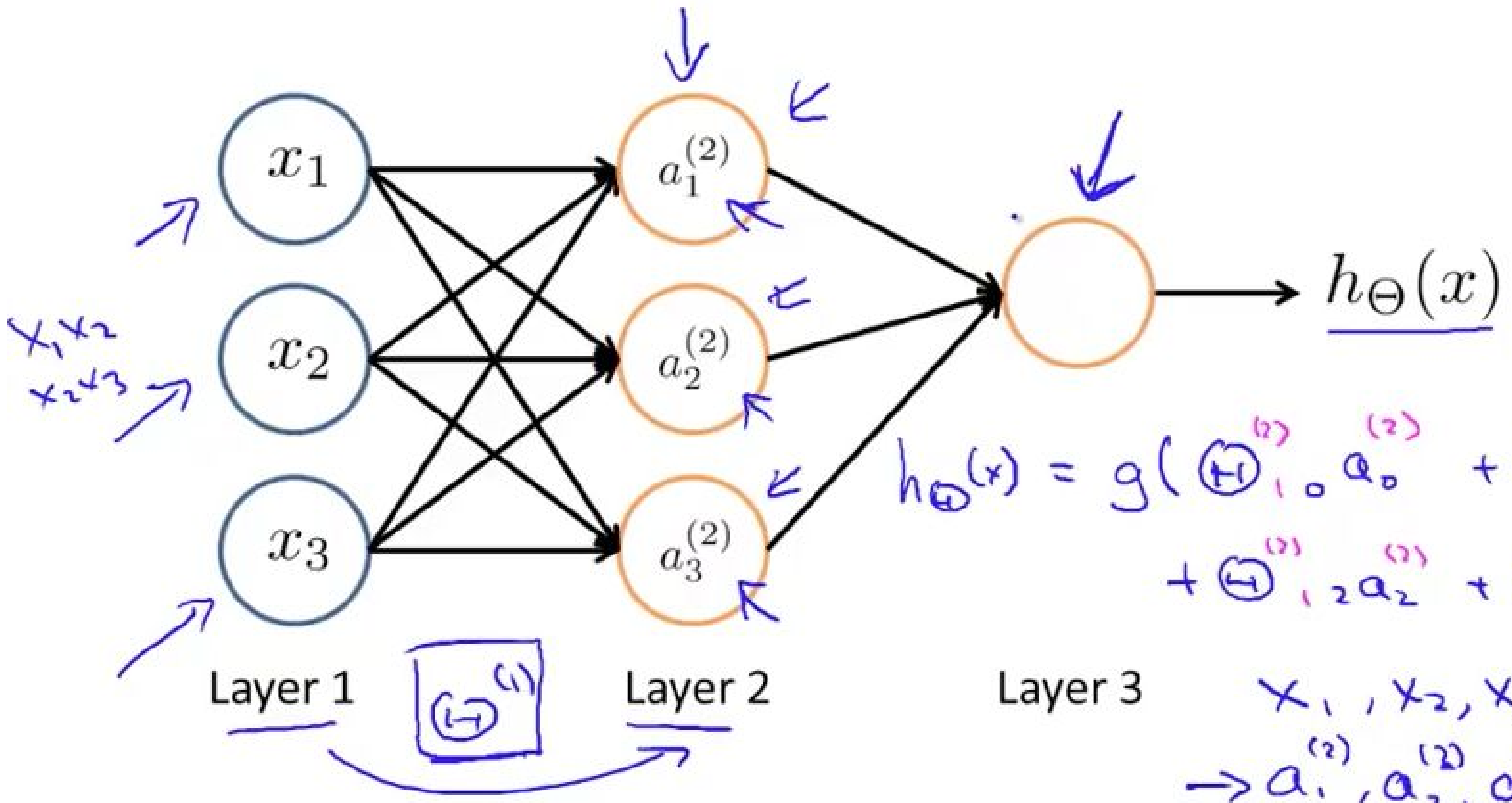
$a_1^{(2)}$
$a_2^{(2)}$
$a_3^{(2)}$

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

$z^{(3)} = \Theta^{(2)}a^{(2)}$

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

Neural Network learning its own features

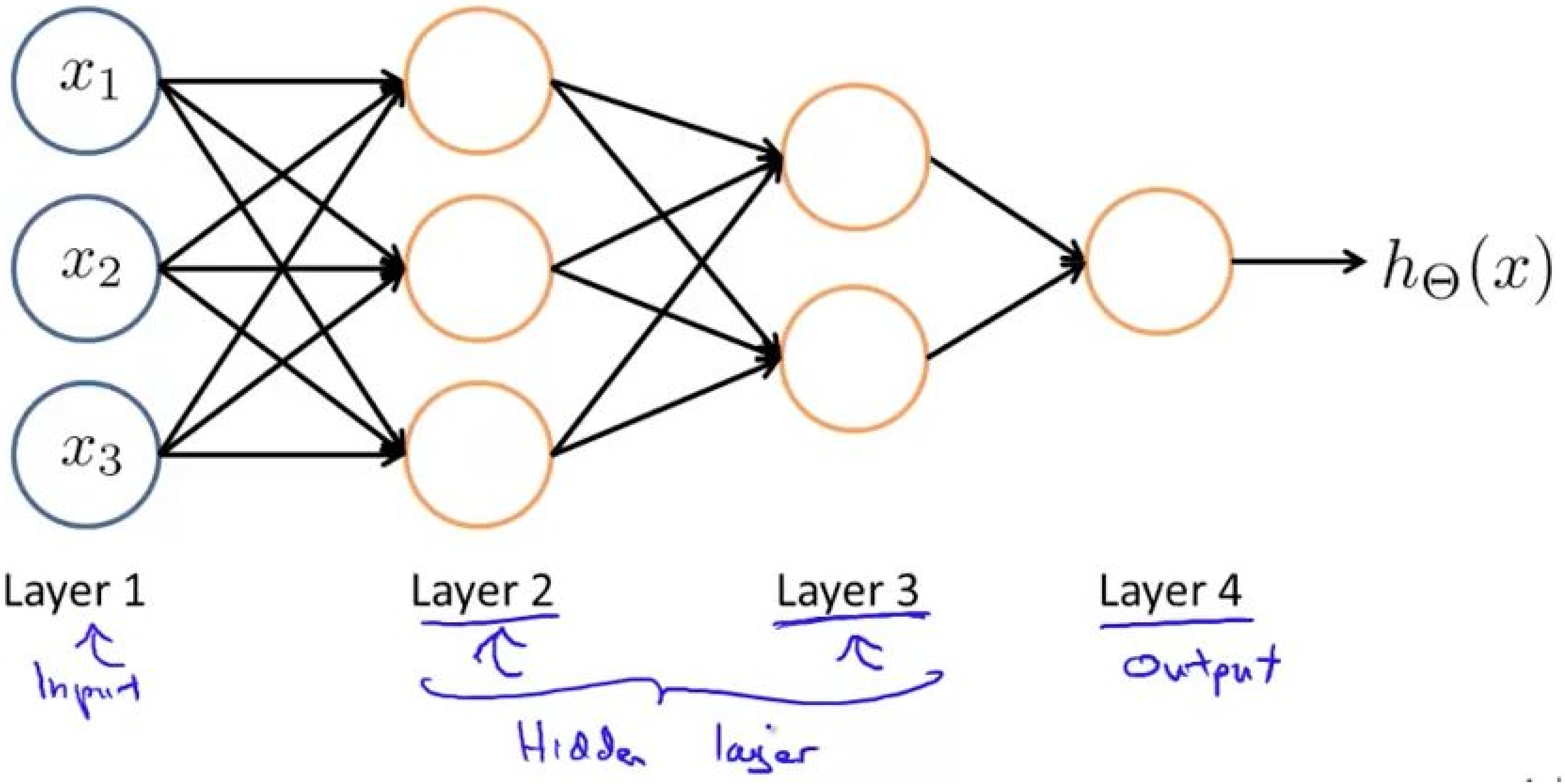


$$h_\Theta(x) = g(\Theta^{(2)}_{10} a^{(2)}_0 + \Theta^{(1)}_{11} a^{(2)}_1 + \Theta^{(2)}_{12} a^{(2)}_2 + \Theta^{(2)}_{13} a^{(2)}_3)$$

$$x_1, x_2, x_3$$

$$\rightarrow a^{(2)}_1, a^{(2)}_2, a^{(2)}_3$$

Other network architectures



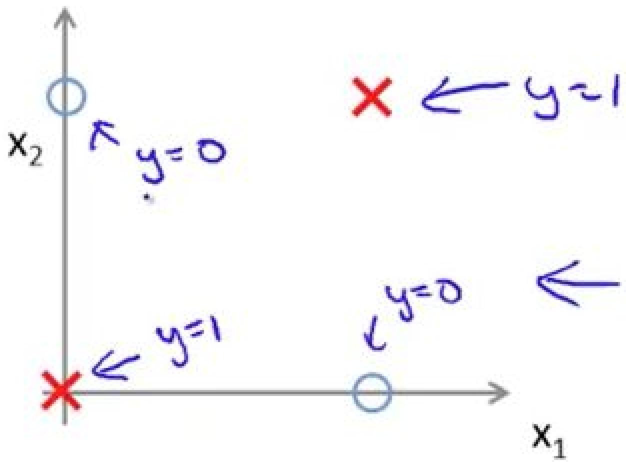
Machine Learning

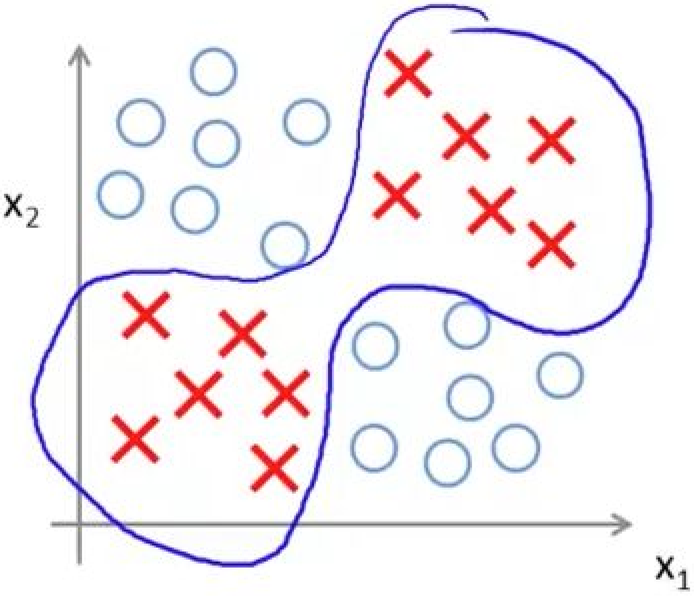
Neural Networks: Representation

Examples and intuitions I

Non-linear classification example: XOR/XNOR

x_1, x_2 are binary (0 or 1).

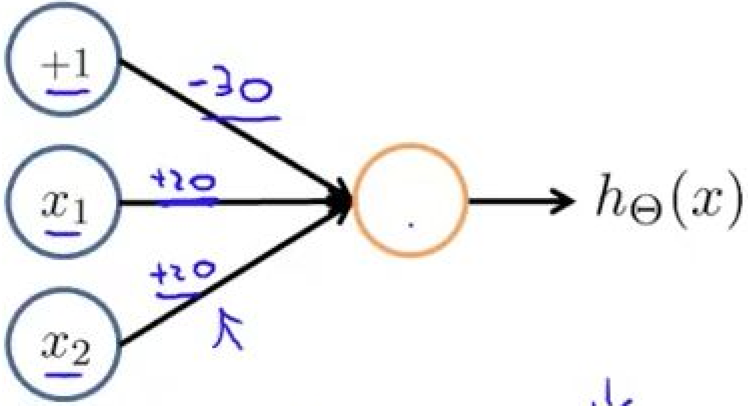


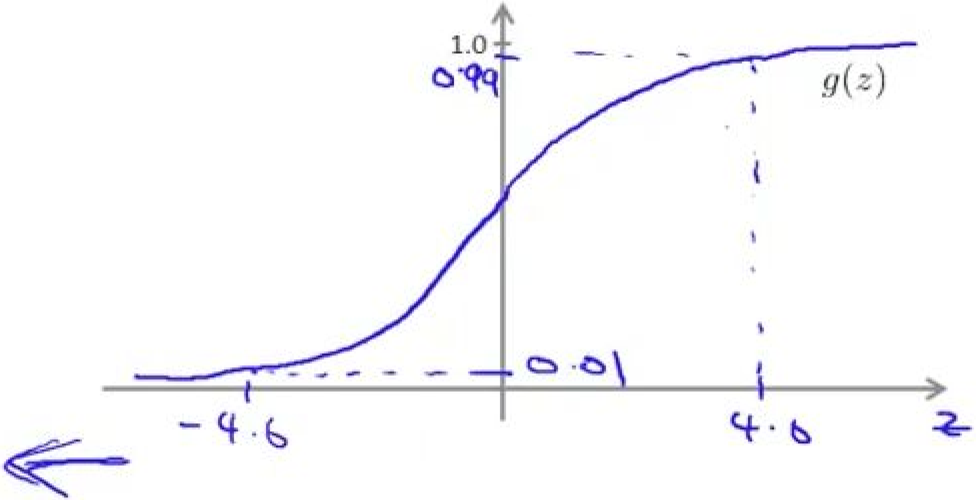


$$y = x_1 \text{ XOR } x_2$$
$$x_1 \text{ XNOR } x_2$$
$$\text{NOT } (x_1 \text{ XOR } x_2)$$

Simple example: AND

- $x_1, x_2 \in \{0, 1\}$
- $y = x_1 \text{ AND } x_2$



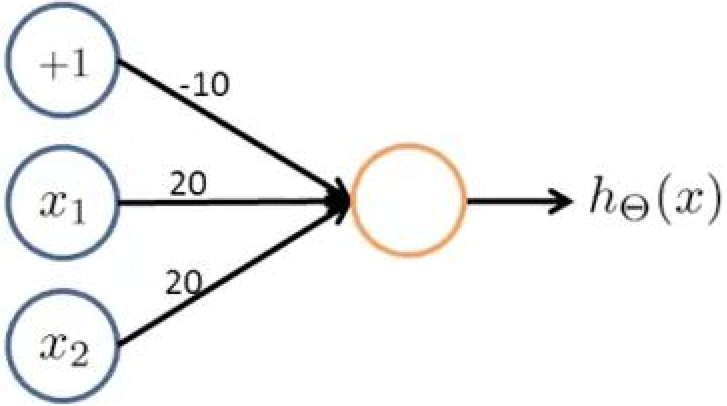


$$h_\Theta(x) = g(\underbrace{-30}_{\Theta^{(1)}_{10}} + \underbrace{20}_{\Theta^{(1)}_{11}} x_1 + \underbrace{20}_{\Theta^{(1)}_{12}} x_2)$$

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

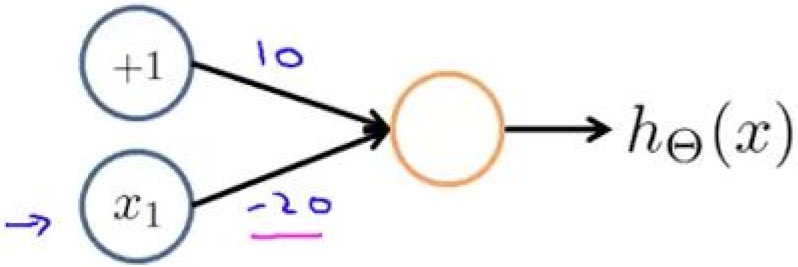
Neural Networks: Representation

Examples and intuitions II

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

Negation:

$\text{NOT } x_1$



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

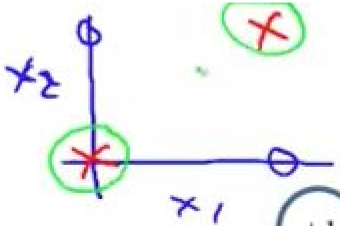
$h_\Theta(x) = g(10 - 20x_1)$

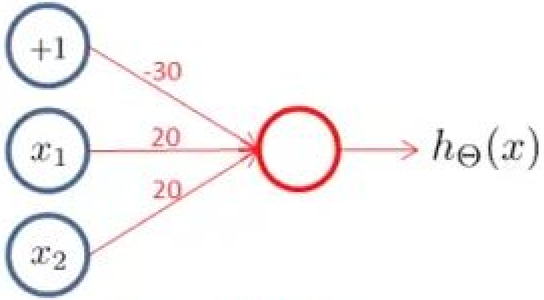
$\rightarrow (\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$

$(= 1$ if and only if

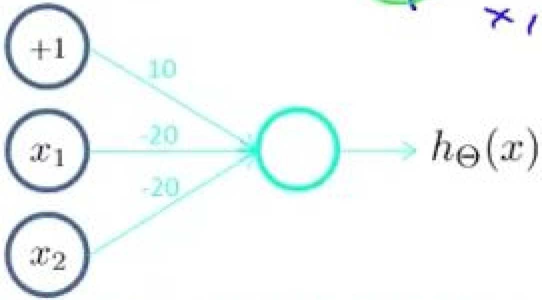
$x_1 = x_2 = 0$

Putting it together: x_1 XNOR x_2

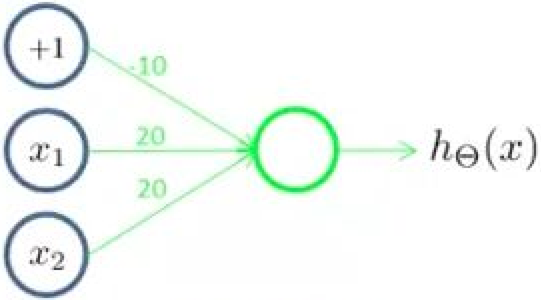




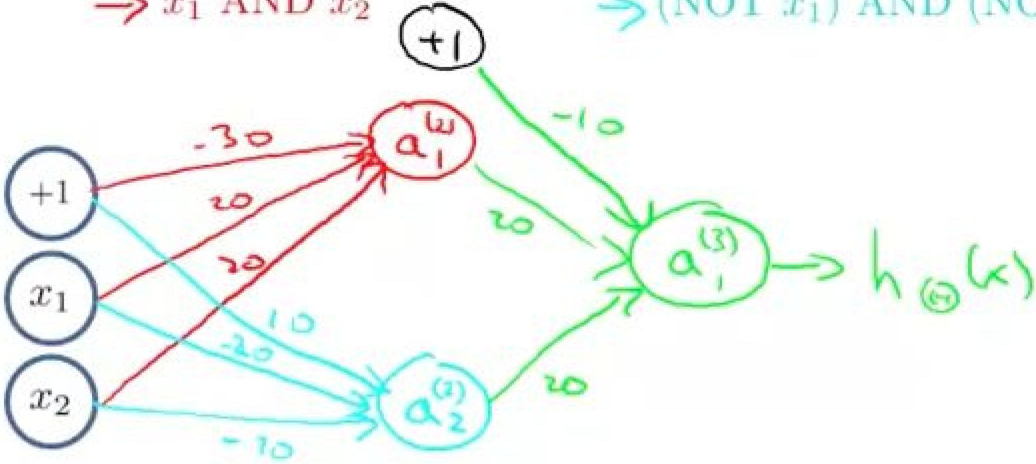
$\rightarrow x_1$ AND x_2



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

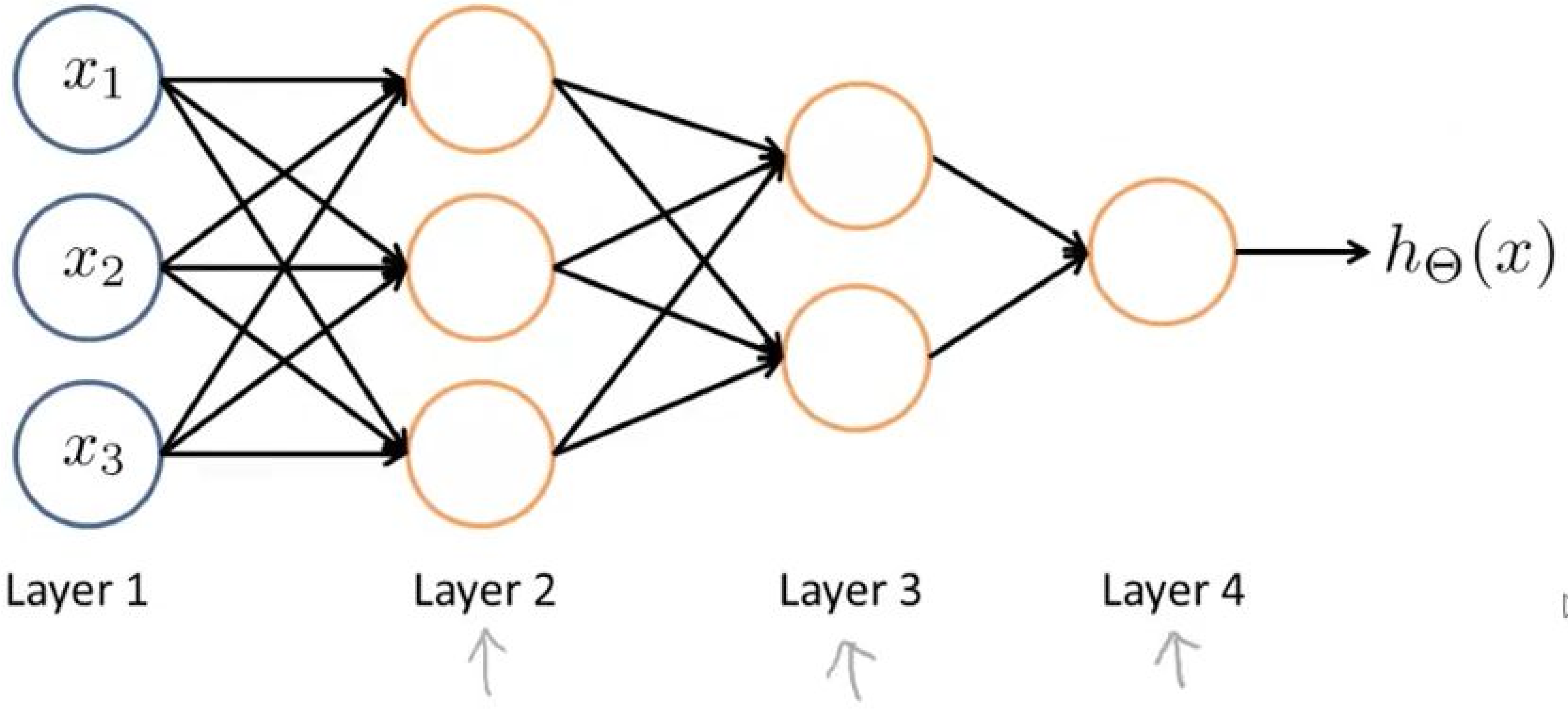


$\rightarrow x_1$ OR x_2

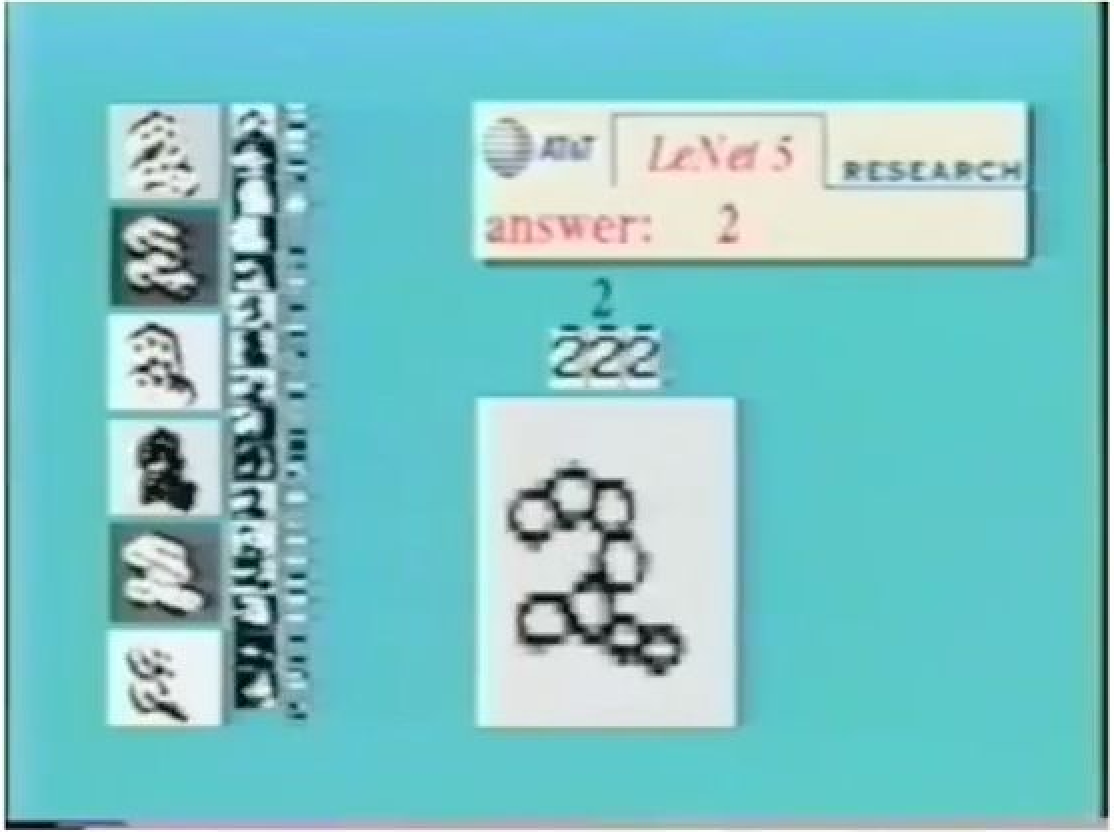


x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

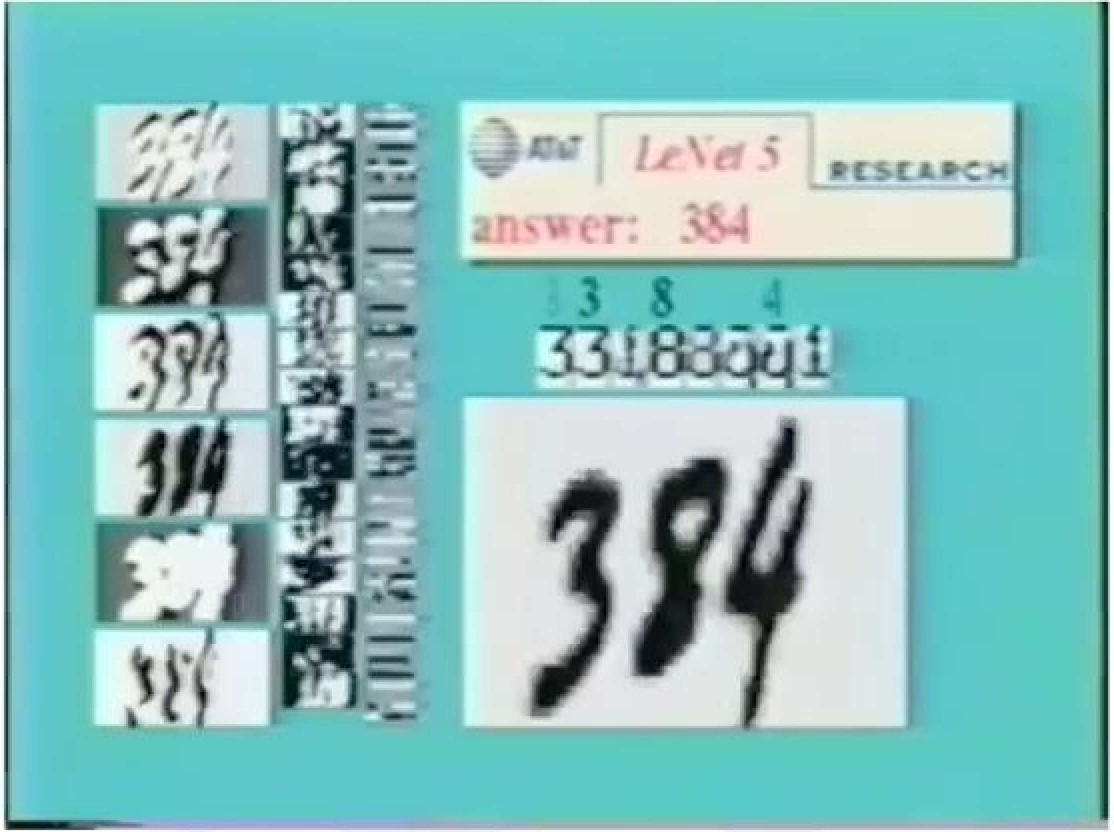
Neural Network intuition



Handwritten digit classification



Handwritten digit classification



Machine Learning

Neural Networks: Representation

Multi-class classification

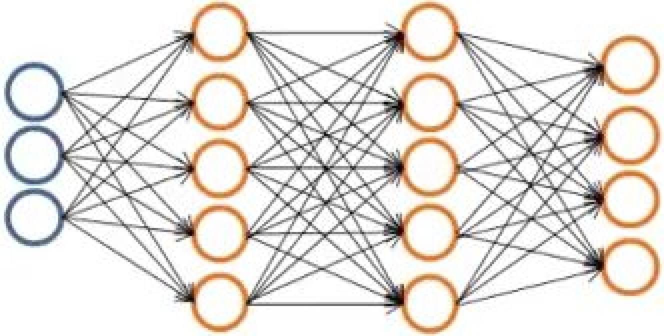
Multiple output units: One-vs-all.

Pedestrian

Car

Motorcycle

Truck

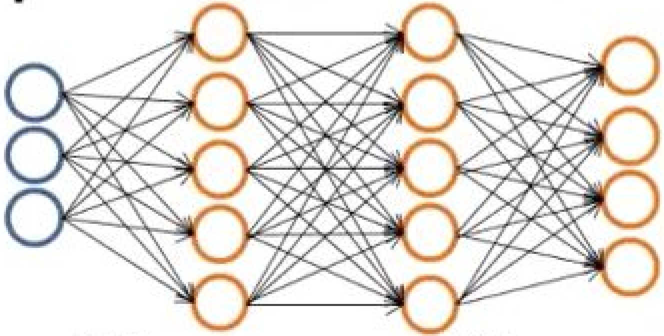


$$h_\Theta(x) \in \mathbb{R}^4$$

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

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: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

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

→ $y^{(i)}$ 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

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

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