ML OL Course

Week 4

Neural Network

Forward

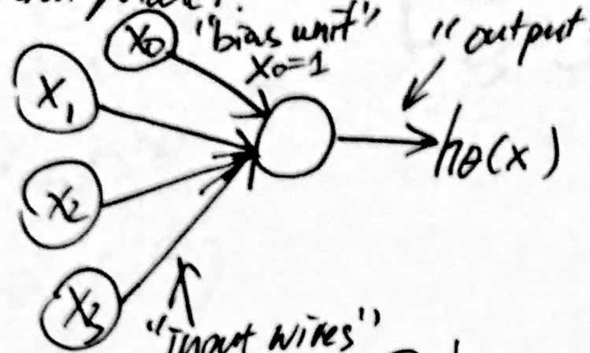
Neural Networks Motivation??? Neuroplasticity

parameters
weights

A start of are techniques for many current algorithms.

Neuron Model:

Single Neuro

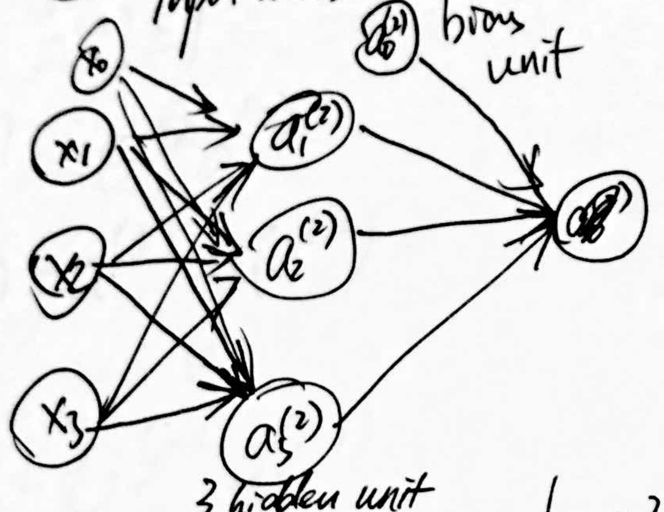


$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Sigmoid (logistic) activation function

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

Neural Network



Layer 1 — x input layer

Layer 2 — Hidden layer

Layer 3 — y output layer

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

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

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

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

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

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

$$, \Theta^{(2)}$$

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

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

Forward propagation: Vectorized implementation

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

根据上图

$z^{(2)} = \Theta^{(1)} X$

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

Add $a_0^{(2)} = 1$

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

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

$z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$

$a^{(j)} = g(z^{(j)})$ add $a_0^{(j)}$ $\leftarrow a_0^{(j)}$

$z^{(j+1)} = \Theta^{(j)} a^{(j)}$

$h_\Theta(x) = a^{(j+1)} = g(z^{(j+1)})$

Non linear classification example: XOR/XNOR

x_1, x_2 are binary (0 or 1).

$y = x_1 \text{ XOR } x_2$

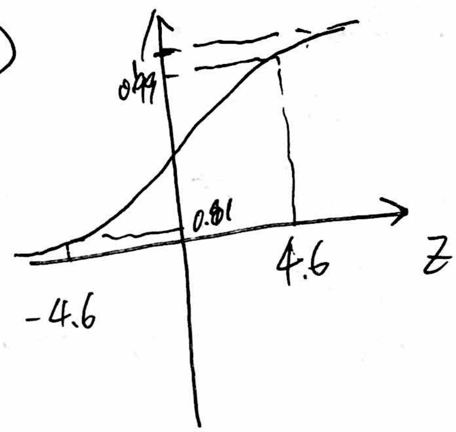
$x_1 \text{ XNOR } x_2$

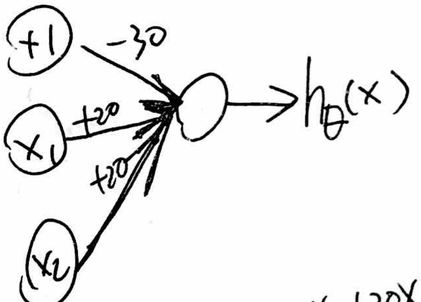
$\text{Not}(x_1 \text{ XOR } x_2)$

Simple example: AND

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

$y = x_1 \text{ AND } x_2$





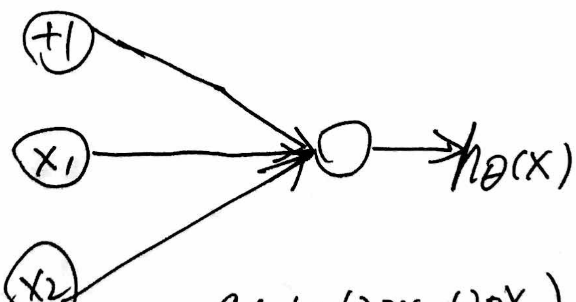
$$h_\theta(x) = g(\underset{\Theta^{(1)}_{10}}{-30} + \underset{\Theta^{(1)}_{11}}{20}x_1 + \underset{\Theta^{(1)}_{12}}{20}x_2)$$

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	$g(10) \approx 1$

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

---

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	$g(10) \approx 1$
1	1	$g(30) \approx 1$

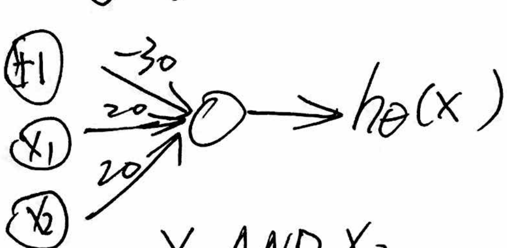
Negation: X_1 AND X_2 X_1 OR X_2

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

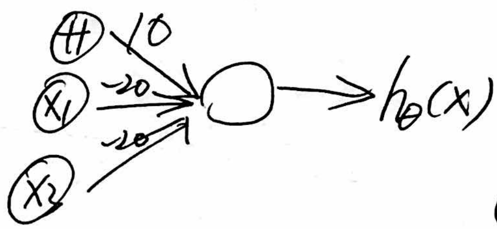
x_1	$h_\theta(x)$
0	$g(10) = 1$
1	$g(-10) = 0$

(NOT X_1) AND (NOT X_2)

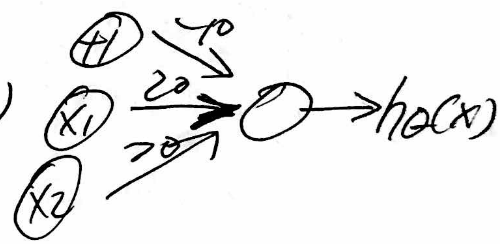
Putting together: X_1 NOR X_2



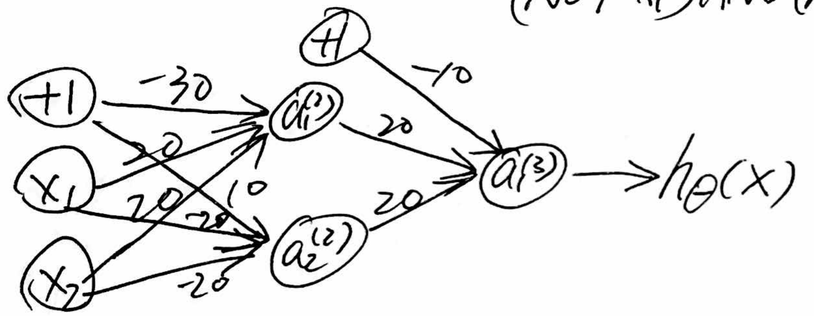
X_1 AND X_2



(NOT X_1) AND (NOT X_2)



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

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.

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

$y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ $(x^{(i)}, y^{(i)})$

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

Neural Network:

$$x = a^{(1)}$$

$$z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$$

$$a^{(j)} = g(z^{(j)})$$

$$z^{(j+1)} = \Theta^{(j)} a^{(j)}$$

$$h_\Theta(x) = a^{(j+1)} = g(z^{(j+1)})$$