function [jVal, gradient] = costFunction(theta)

jVal = [code to compute $J(\theta)$];

gradient(1) = [code to compute $\frac{\partial}{\partial \theta_1} J(\theta)$];

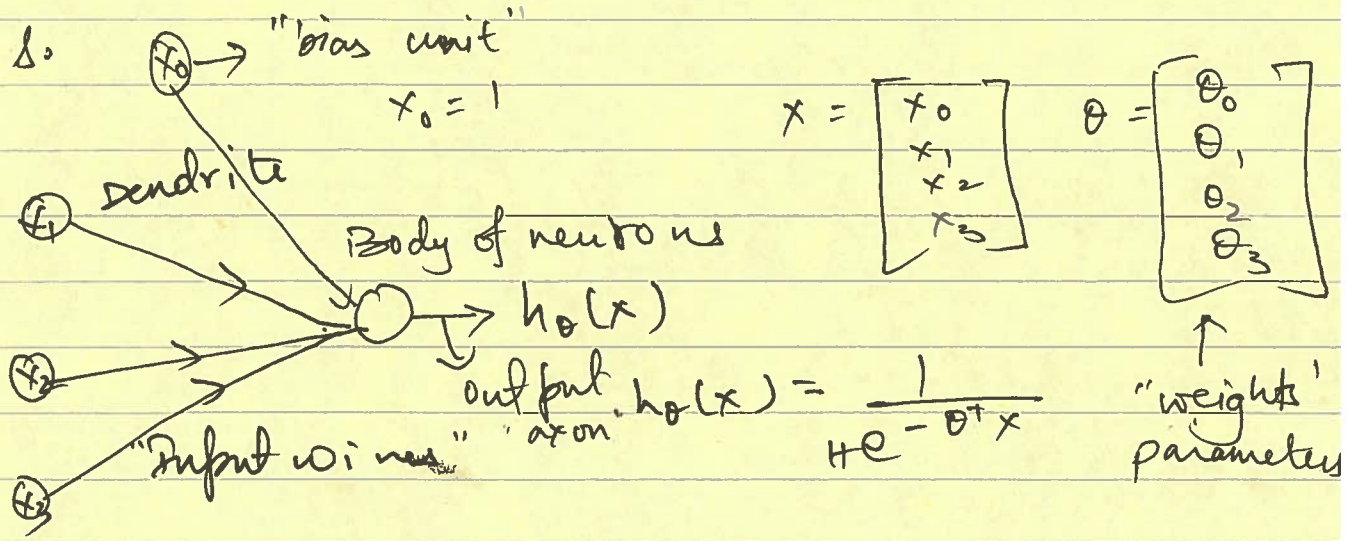
gradient(2) = [code to compute $\frac{\partial}{\partial \theta_2} J(\theta)$];

⋮

gradient(n+1) = [code to compute $\frac{\partial}{\partial \theta_n} J(\theta)$];

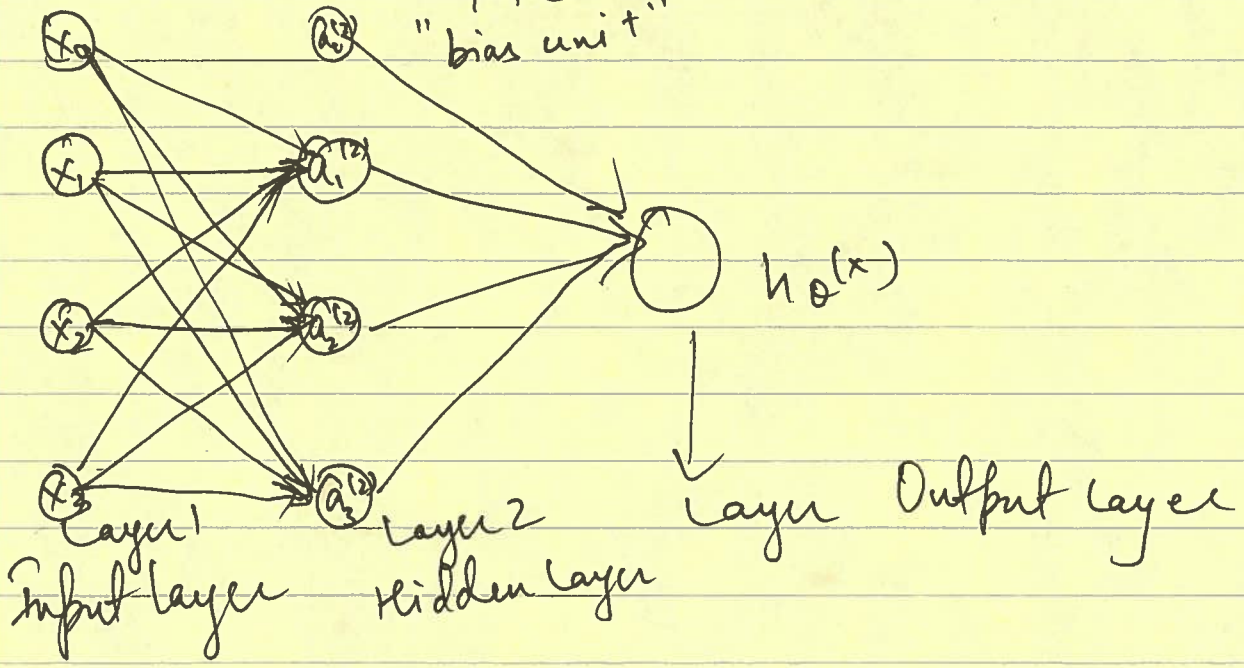
Neurons → Dendrite (Input Wires)

→ Axon (Output Wire)




Sigmoid (logistic) activation function

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



# Neural Network Rep

## Model Rep-I

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

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

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

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

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

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

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

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

$$s_2 \times (s_1 + 1)$$

Layer 1 ($s_j = 2$) Layer 2. ($s_{j+1} = 4$)

$$\underbrace{s_{j+1}}_{4} \times s_j + 1$$

$$4 \times (2+1)$$

# Neural Network Rep
## Model Rep - II

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

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

$$z_k^{(2)} = \Theta_{k,0}^{(1)} x_0 + \Theta_{k1}^{(1)} x_1 + \dots + \Theta_{kn}^{(1)} x_n$$

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

$$z^{(2)} = \Theta^{(1)} x$$
$$a^{(2)} = g(z^{(2)})$$
$\hookrightarrow \mathbb{R}^3 \uparrow \mathbb{R}^3$

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

Bias Unit

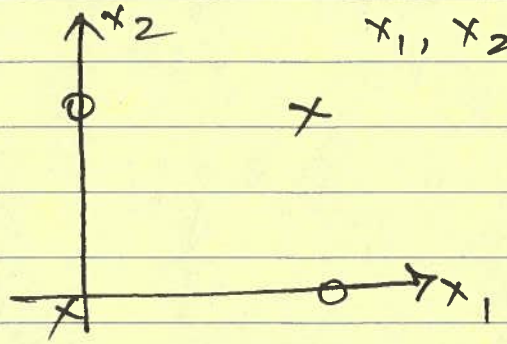
Add $a_0^{(2)} = 1 \rightarrow a^{(2)} \in \mathbb{R}^4$

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

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

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

# Examples & Intuition.




$x_1, x_2$ are binary (0 or 1)

$y = x_1$ XOR $x_2$

$x_1$ XNOR $x_2$

NOT ($x_1$ XOR $x_2$)

XOR

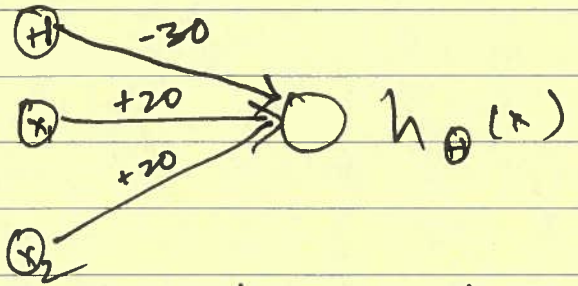
| A | B | A XOR B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

XNOR

| A | B | A XNOR B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

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

$y = x_1$ AND $x_2$.




$$h_\theta(x) = g(-30 + 20x_1 + 20x_2)$$

$\theta_{10}^{(1)}$ $\quad$ $H_{11}^{(1)}$ $\quad$ $H_{12}^{(1)}$

| $x_1$ | $x_2$ | $h_\theta x$ |
|---|---|---|
| 0 | 0 | $g(-30) \approx 0$ |
| 0 | 1 | $g(-10) \approx 0$ |
| 1 | 0 | $g(-10) \approx 0$ |
| 1 | 1 | $g(0) \approx 1$ |

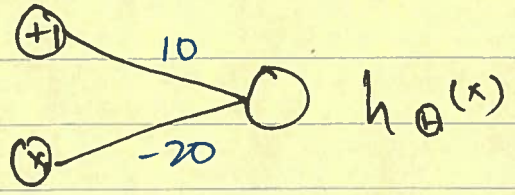
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$h_\theta(x) \approx x_1$ AND $x_2$

| $x_1$ | $x_2$ | |
|---|---|---|
| 0 | 0 | $g(-10) \approx 0$ |
| 0 | 1 | $g(20) \approx 1$ |
| 1 | 0 | $g(20) \approx 1$ |
| 1 | 1 | $g(40) \approx 1$ |

$x_1$ AND $x_2$

Not $x_1$




$g(10 - 20x_1)$

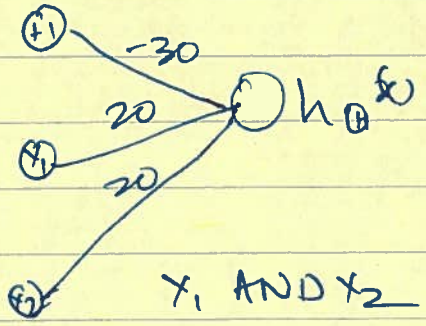
| $x_1$ | $h_\Theta(x)$ |
|---|---|
| 0 | $g(10) \cong 1$ |
| 1 | $g(-20) \cong 0$ |

$x_1$ OR $x_2$. $\{0,1\}$

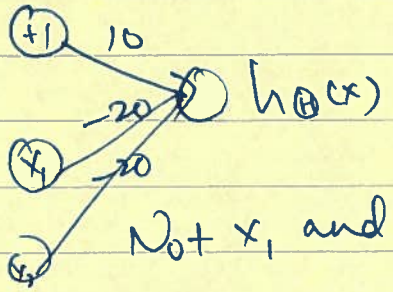
Not $x_1$ AND Not $x_2$

$= 1$ if any only if $x_1 = x_2 = 0$

Large

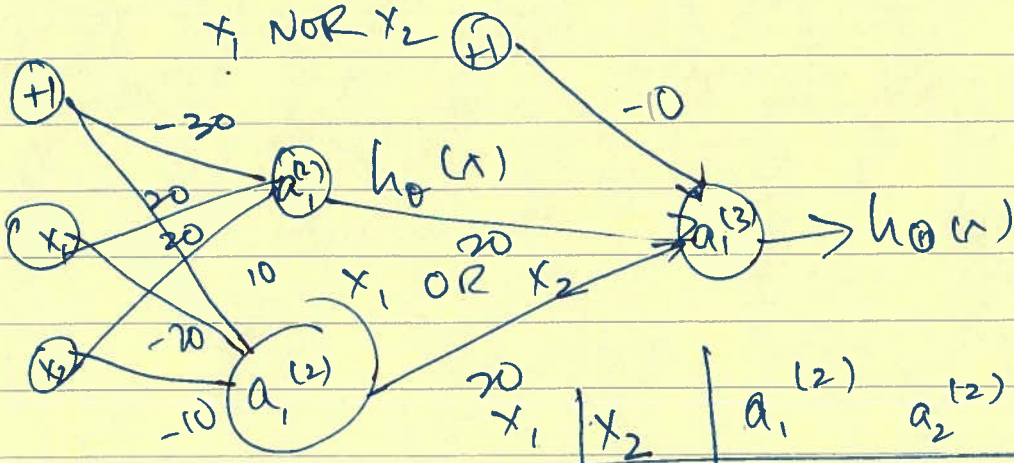



$x_1$ AND $x_2$




Not $x_1$ and Not $x_2$

$x_1$ NOR $x_2$




$a_1^{(3)} \rightarrow h_\Theta(x)$

↑ output

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

$g(30 - 20(x_1) - 20(x_2))$

$g(-20 + 30(x_1) + 30(x_2))$

| $x_1$ | $x_2$ | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| $x_1$ | $x_2$ | $h_\theta(x)$ |
|---|---|---|
| 0 | 0 | $g(30) \geq 1$ |
| 0 | 1 | $g(10) \geq 1$ |
| 1 | 0 | $g(10) \geq 1$ |
| 0 | 1 | $g(-10) \geq 0$ |

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

$L$ = total # of layers in network

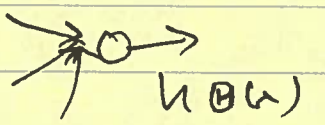
$L = 4$

$s_l$ = no. of units (not counting bias unit) in layer $l$

$s_1 = 3, s_2 = 5, s_4 = s_L = 4$

Binary classification

$y = 0$ or $1$

1 output unit 


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

$s_L = 1$ $\quad K = 1$

Multiclass

$y \in \mathbb{R}^K$ Eg $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix},$ ped car

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

$s_L = K \quad (K \geq 3)$

Logistic Regression

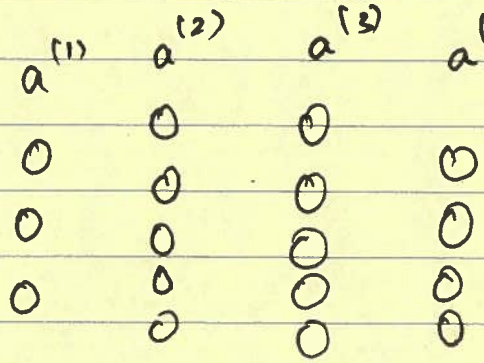
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

K → dim

i → i element

Gradient computation.

$(x, y)$ → $\delta^{(2)}$ $\delta^{(3)}$ $\delta^{(4)}$




$a^{(1)} = x$

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

$a^{(2)} = g(z^{(2)})$ (add $a_0^{(2)}$)

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

$a^{(3)} = g(z^{(3)})$

$z^{(4)} = \Theta^{(3)} a^{(3)}$

$a^{(4)} = h_\Theta(x) = g(z^{(4)})$

$\delta_j^{(l)}$ = "error" of node $j$ in layer $l$

$a_j^{(l)}$

$L = 4 \longrightarrow (h_\Theta(x))_j$

$\delta_j^{(4)} = a_j^{(4)} - y_j$ $\quad \delta^{(4)} = a^{(4)} - y$ $\quad$ $y$ vector = output unit.

$\delta_j^{(3)} = (\Theta^{(3)})^T \delta^{(4)} .* g'(z^{(3)})$ $\quad a^{(3)} .* (1 - a^{(3)})$

$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)})$ $\quad a^{(3)} .* (1 - a^{(2)})$

(No $\delta^{(1)}$)

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = a_j^{(l)} \delta_i^{(l+1)}$$

ignoring $\lambda$ if $\lambda = 0$.

Training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

$\Delta_{ij}^{(l)} = 0$ for all $l, i, j$.

use to compute $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$

For $i = 1$ to $m$

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

Perform forward prop to compute $a^{(l)}$ for $l = 2, 3, \ldots, L$

Using $y^{(i)}$ compute $\delta^{(L)} = a^{(L)} - y^{(i)}$

compute $\delta^{(L-1)}, \delta^{(L-2)}, \ldots, \delta^{(2)}$

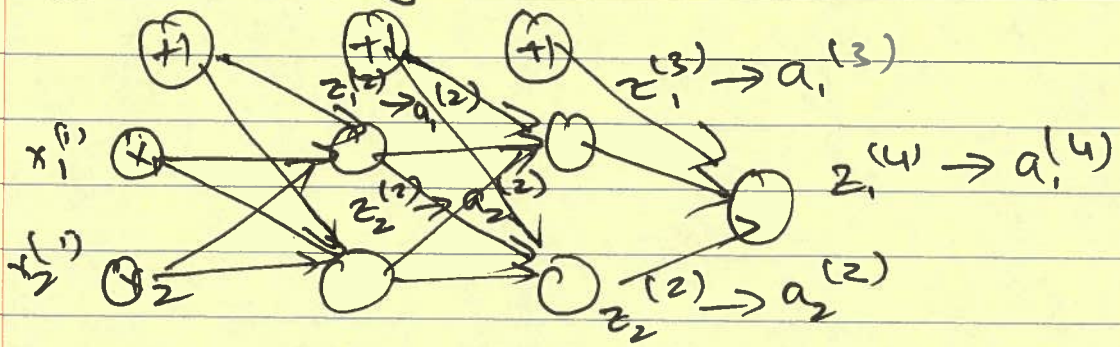
$$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$$

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)} \left(a^{(l)}\right)^T$$

$$D_{ij}^{(l)} := \frac{1}{m} \Delta_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)} \quad \text{if } j \neq 0$$

$$D_{ij}^{(l)} := \frac{1}{m} \Delta_{ij}^{(l)} \quad \text{if } j = 0$$

# Forward Propagation




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

$$z_{11}^{(3)} = \Theta_{10}^{(2)} \times 1 + \Theta_{11}^{(2)} \times a_1^{(2)} + \Theta_{12}^{(2)} a_1^{(2)}$$