WEEK 4

· Neural Networks: Representation

WEEK 4 Neural Networks

Neural networks motivation

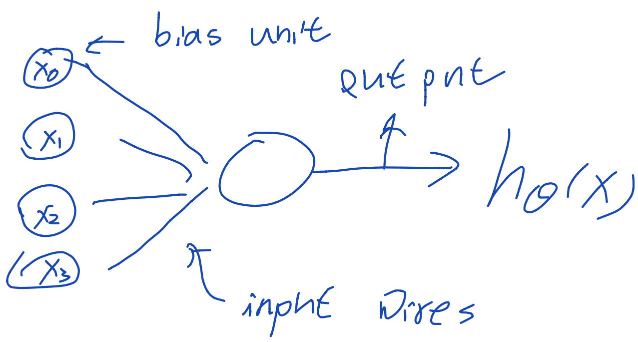
- Complex regression
- Large number of features

Origins: Algorithms ~~that~~ try to mimic ~~the~~ brain

Somatosensory Cortex "one learning hypothesis"

Neural Networks representatio

Neuron model: Logistic unit

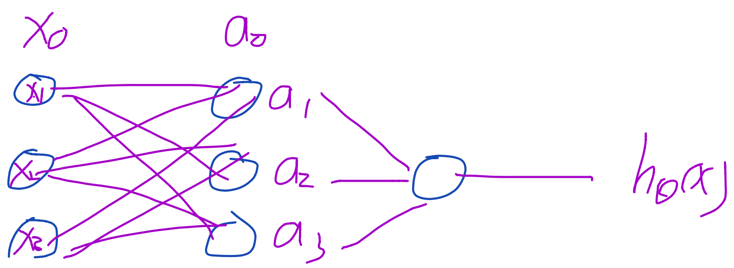


terminology:

Sigmoid (logistic) activation function $g(z)=\frac{1}{1+e^{-z}}$

$x=\begin{bmatrix}x_0\\x_1\\x_2\\x_3\end{bmatrix}$ $\theta=\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\\\theta_3\end{bmatrix}$ ← "weights" = parameters

Neural Network:



Layer 1 → Input Layer

Layer 2 → Hidden Layer

Layer 3 → Output Layer

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

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

$$a_2^{(2)} = g(\underbrace{\Theta_{20}\cdots\Theta_{21}+\cdots\Theta_{22}\cdots\cdots\Theta_{33}\cdots}) \equiv z_2^{(2)}$$

$$a_3^{(2)} = \underbrace{-\ -\ -\ -\ -\ -\ -\ -\ -\ -\ -\ -} \cdots \equiv z_3^{(2)}$$

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

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

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

Forward Propagation: Vectorized Implementation

vectorize:

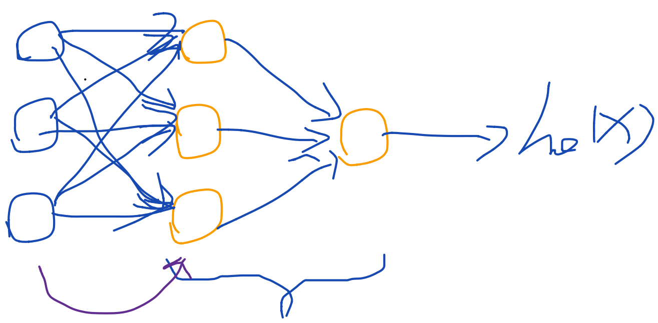
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} \qquad a^{(2)} = \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

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

also define $a^{(1)} = x$, add $a_0^{(2)} = 1$

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

Neural Network learning it own features



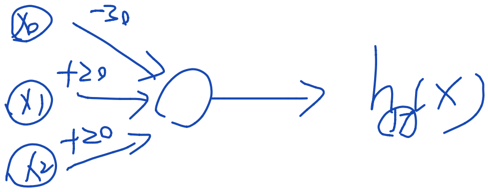
$\Theta^{(1)}$

Logistic regression $h_\theta(x) = g\left(\Theta_0^{(2)} a_0^{(2)} + \Theta_1^{(2)} a_1^{(2)} + \Theta_2^{(2)} a_2^{(2)} + \Theta_3^{(2)} a_3^{(2)}\right)$

Neural network learning its own features

Examples

(1) Logic AND function

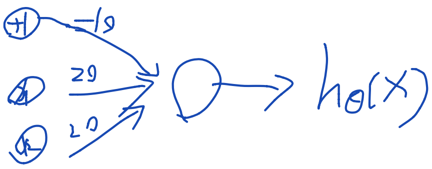


$h_\Theta(x) = g(-30 + 20x_1 + 20x_2)$

g : sigmoid function

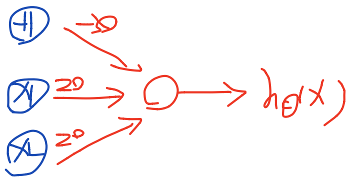
x_1	x_2	$h_\Theta(x)$
0	0	0
0	1	0
1	0	0
1	1	1

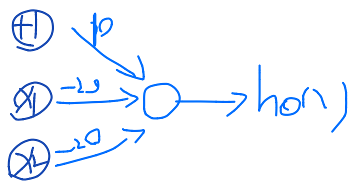
(2) Logic OR function

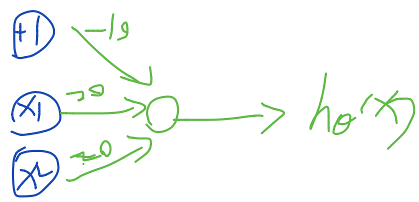


(3) Logic NOT

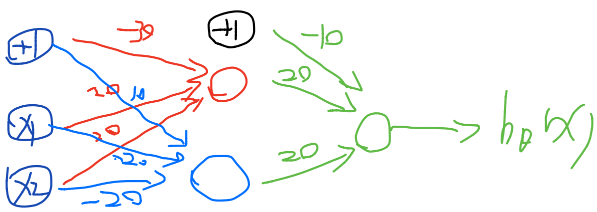
(4) XNOR







x_1 AND x_2 (NOT x_1) AND (NOT x_2) x_1 OR x_2



XNOR !!!

Multi-class classification

$$\text{output} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \dots$$