Neural network

Ngô Minh Nhựt

2021

Part 1

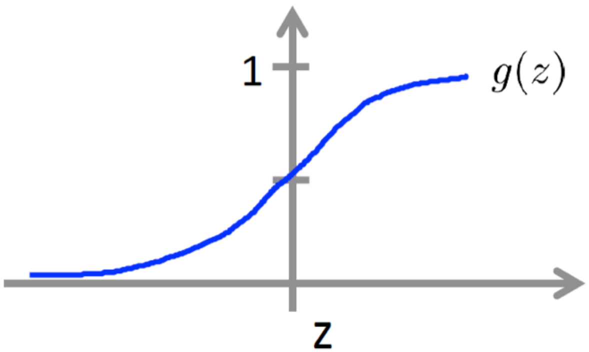
Model representation and forward propagation

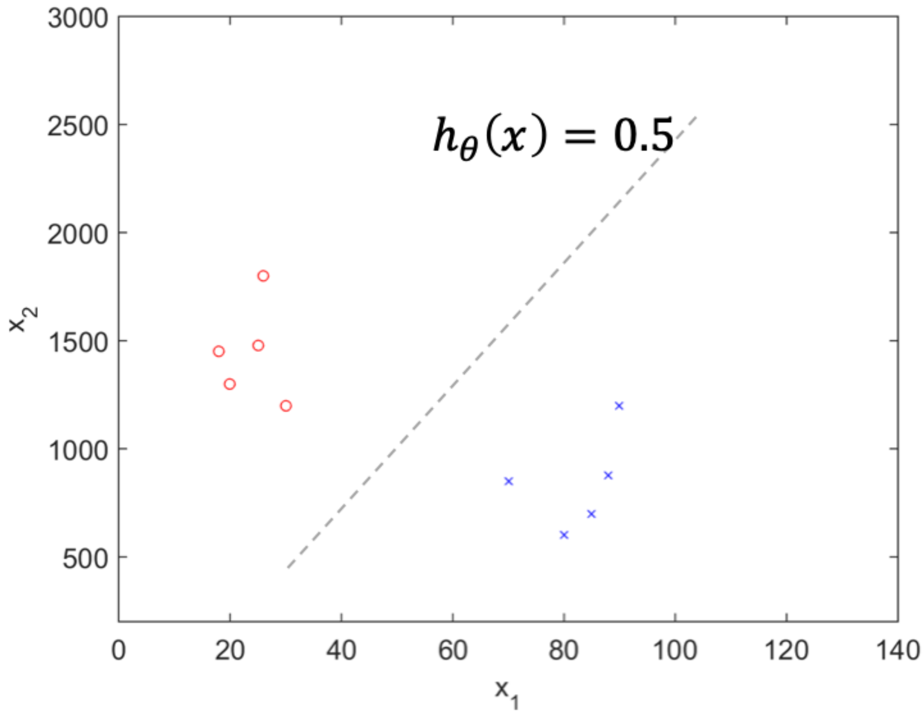
Logistic regression

- Linear classifier

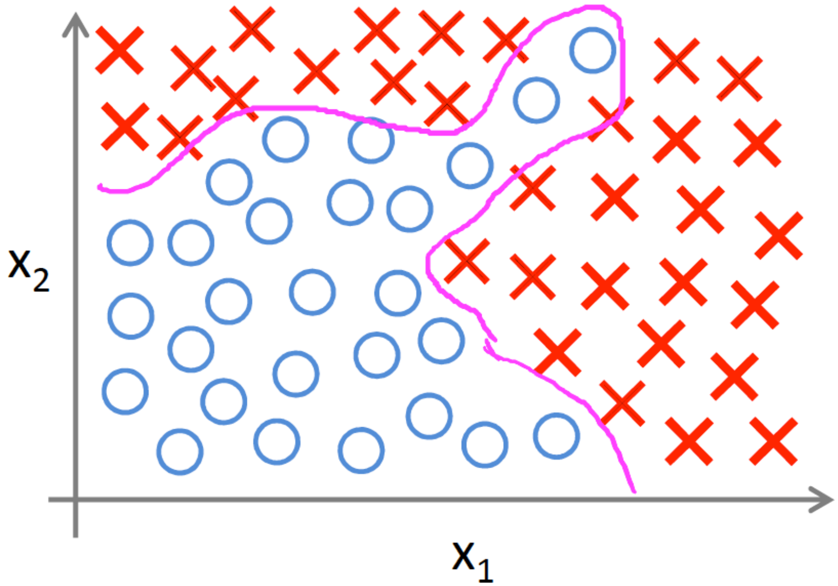
$$h_\theta(x) = g(\theta^T x)$$

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





Non-linear classifier



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

Feature mapping

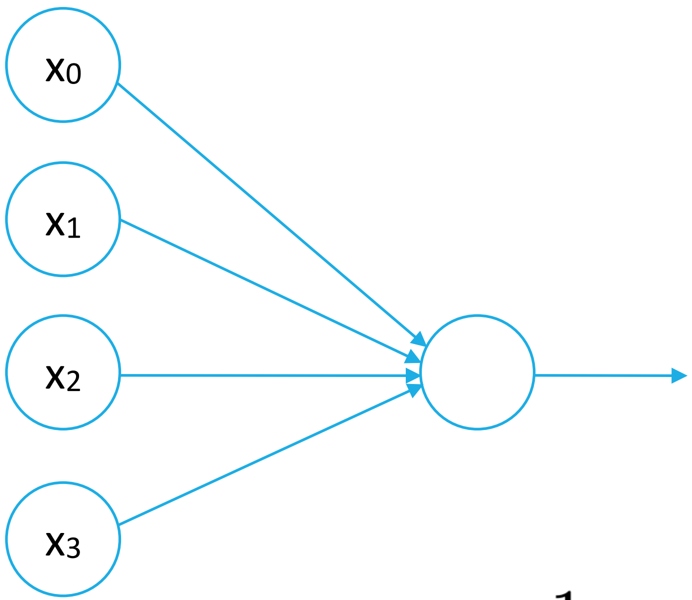
x_1 = size
x_2 = number of bedrooms
x_3 = number of floors
x_4 = age
…
x_{100}

What mappings?
How many mappings?
How many features?

Non-linear classification with logistic regression requires a lot of features

Source: Andrew Ng

Logistic regression



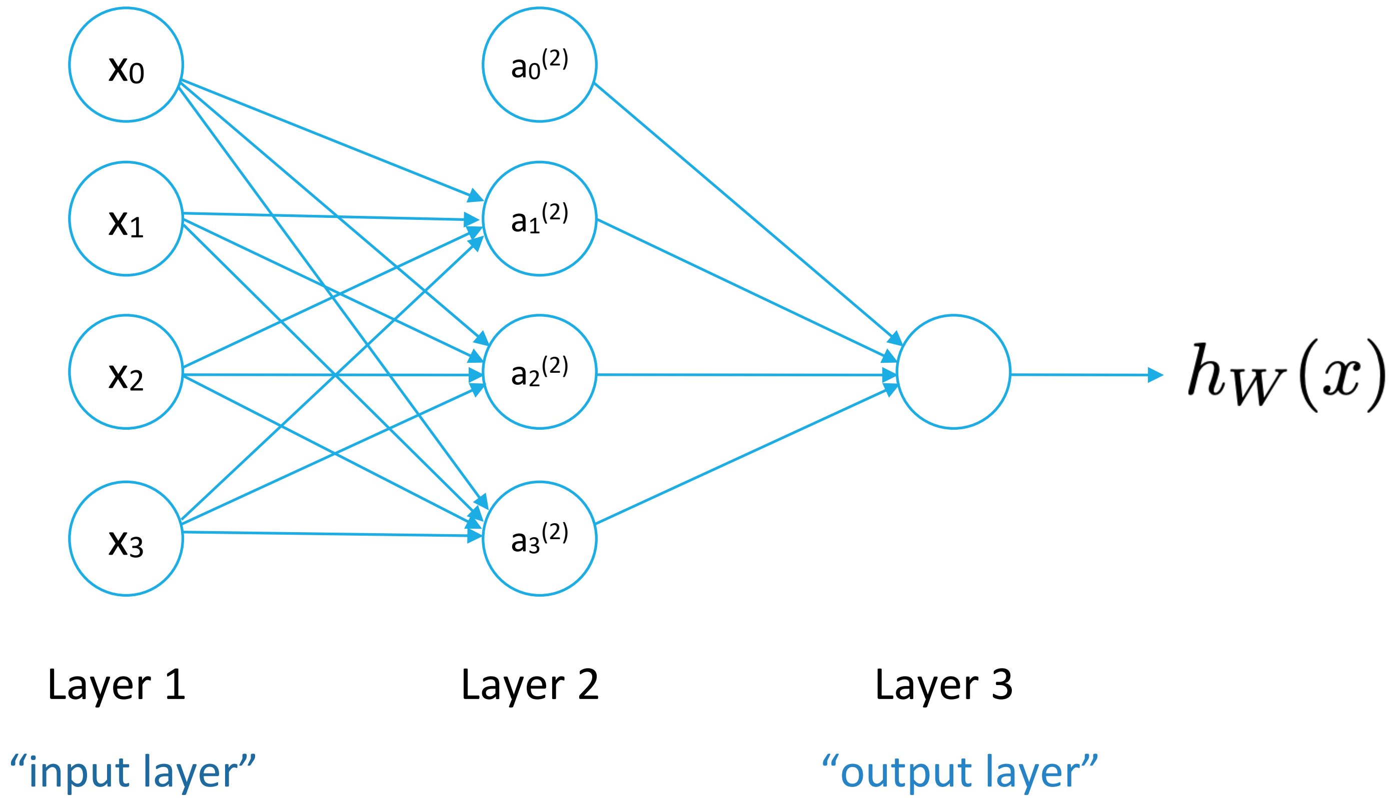
$$h_\theta(x)$$

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

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

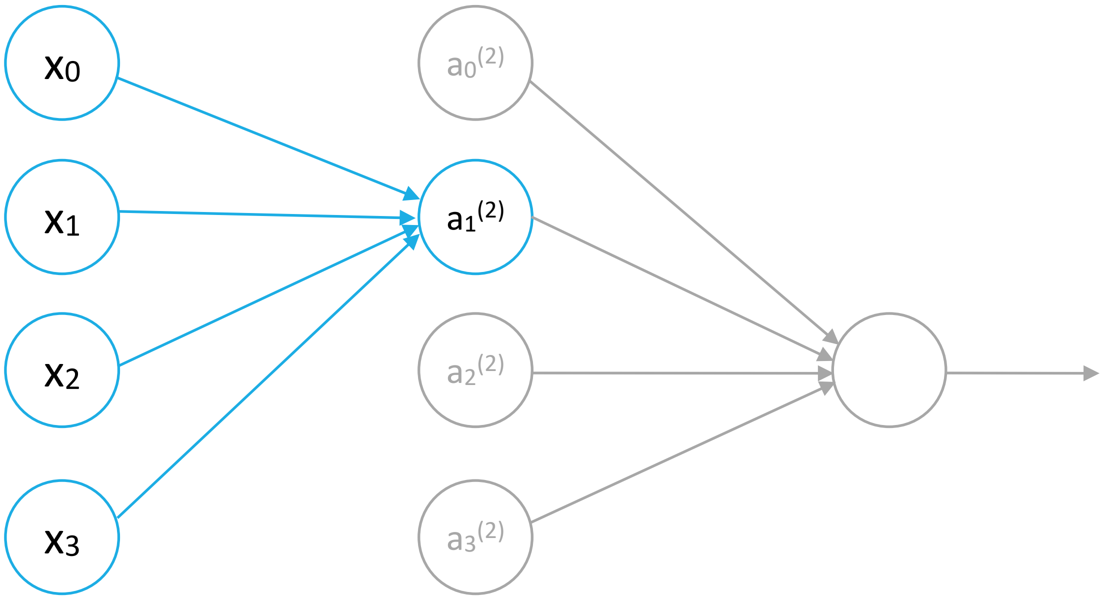
Sigmoid activation function

Neural network



Activation at node

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

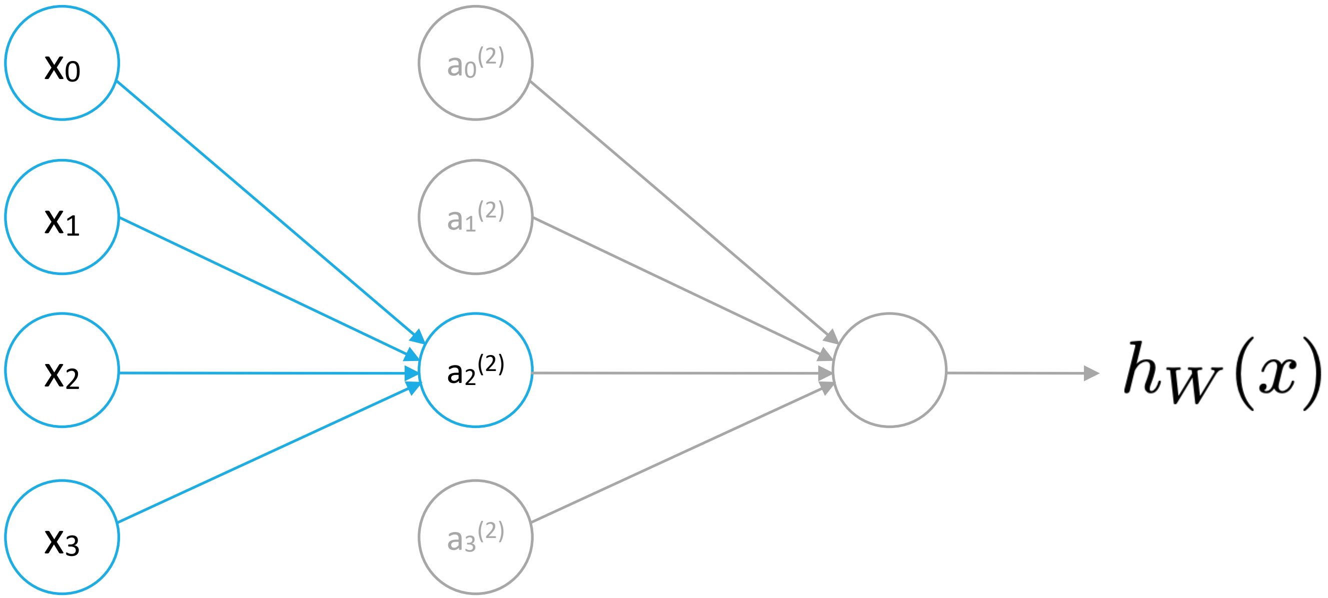


$$h_W(x)$$

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

Activation at node

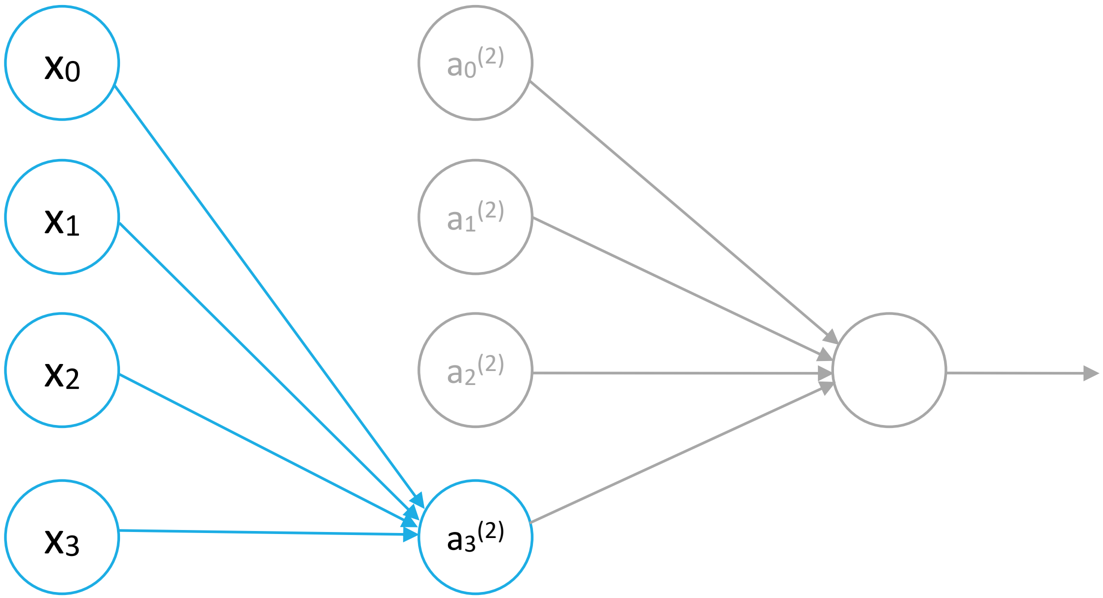
$$g(z) = \frac{1}{1+e^{-z}}$$



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

Activation at node

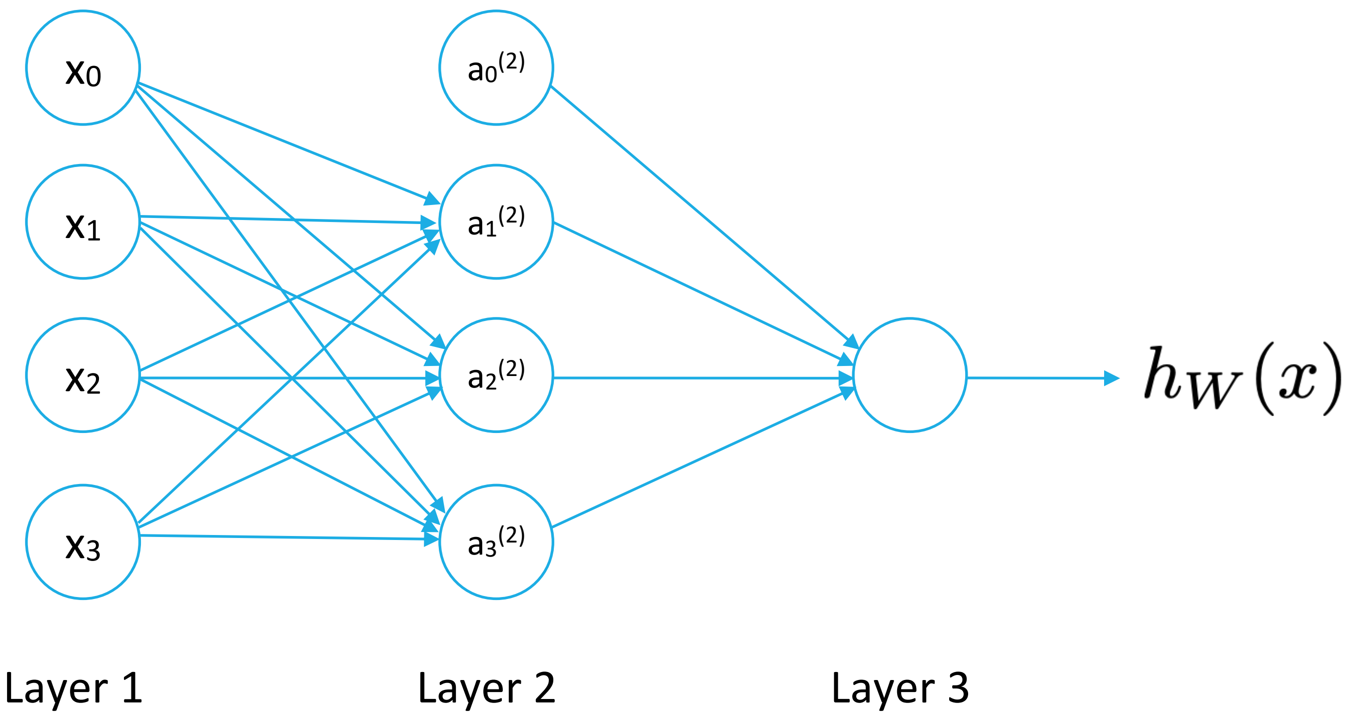
$$g(z) = \frac{1}{1 + e^{-z}}$$



$$h_W(x)$$

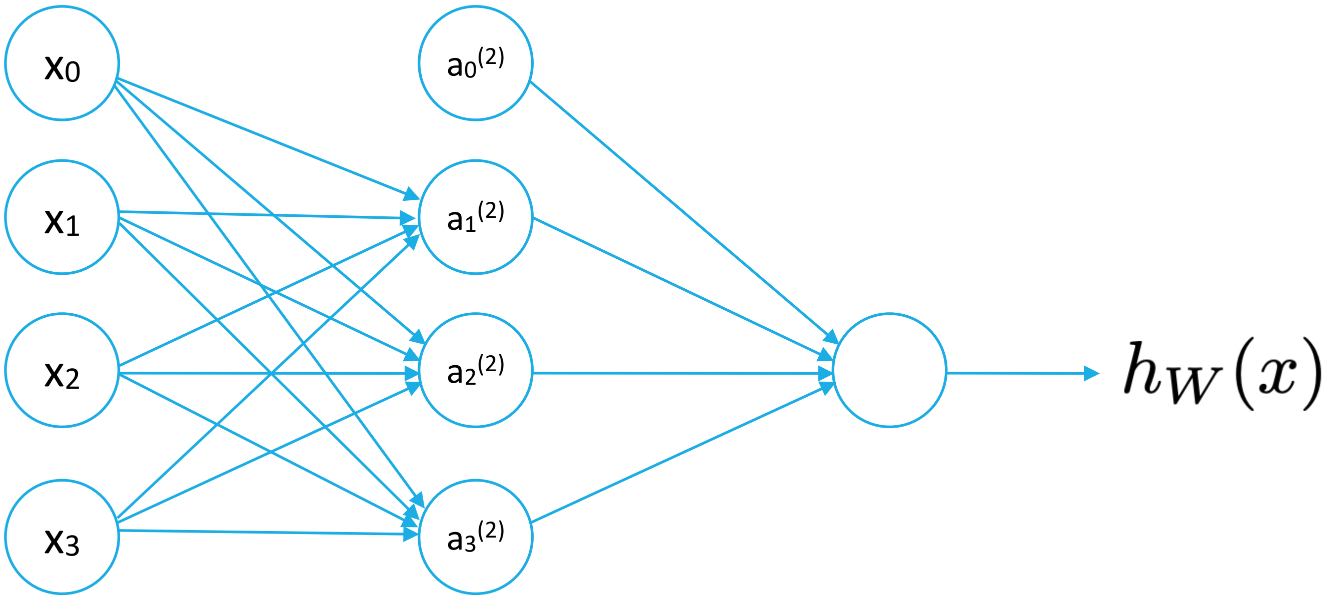
$$a_3^{(2)} = g\left(W_{30}^{(1)} x_0 + W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3\right)$$

Weight matrix



$a_i^{(j)}$: activation of i^{th} node in j^{th} layer
$W^{(j)}$: weight matrix mapping activation in j^{th} layer to $j+1^{th}$ layer
If neural network has s_j nodes in j^{th} layer and s_{j+1} nodes in $j+1^{th}$ layer, $W^{(j)}$ has a size of s_{j+1} x $(s_j + 1)$

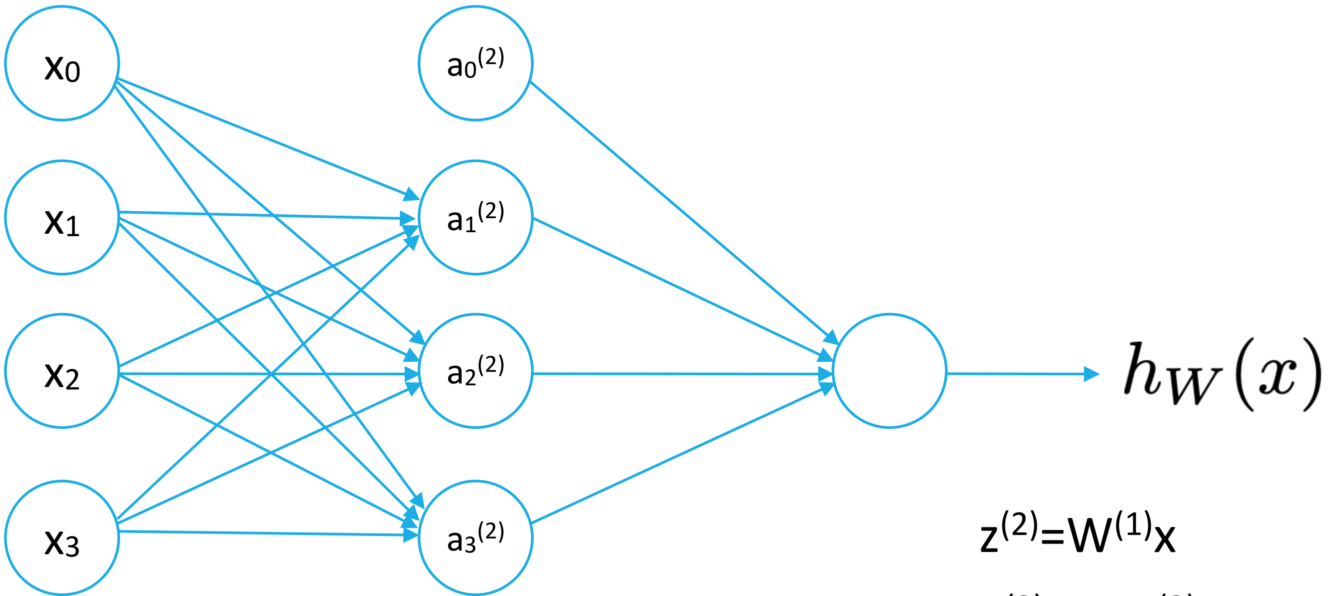
Forward propagation



$$a_1^{(2)} = g\left(W_{10}^{(1)}x_0 + W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3\right)$$
$$a_2^{(2)} = g\left(W_{20}^{(1)}x_0 + W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3\right)$$
$$a_3^{(2)} = g\left(W_{30}^{(1)}x_0 + W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3\right)$$

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

Representation by vector



$z^{(2)} = W^{(1)}x$

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

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

$z^{(3)} = W^{(2)}a^{(2)}$

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

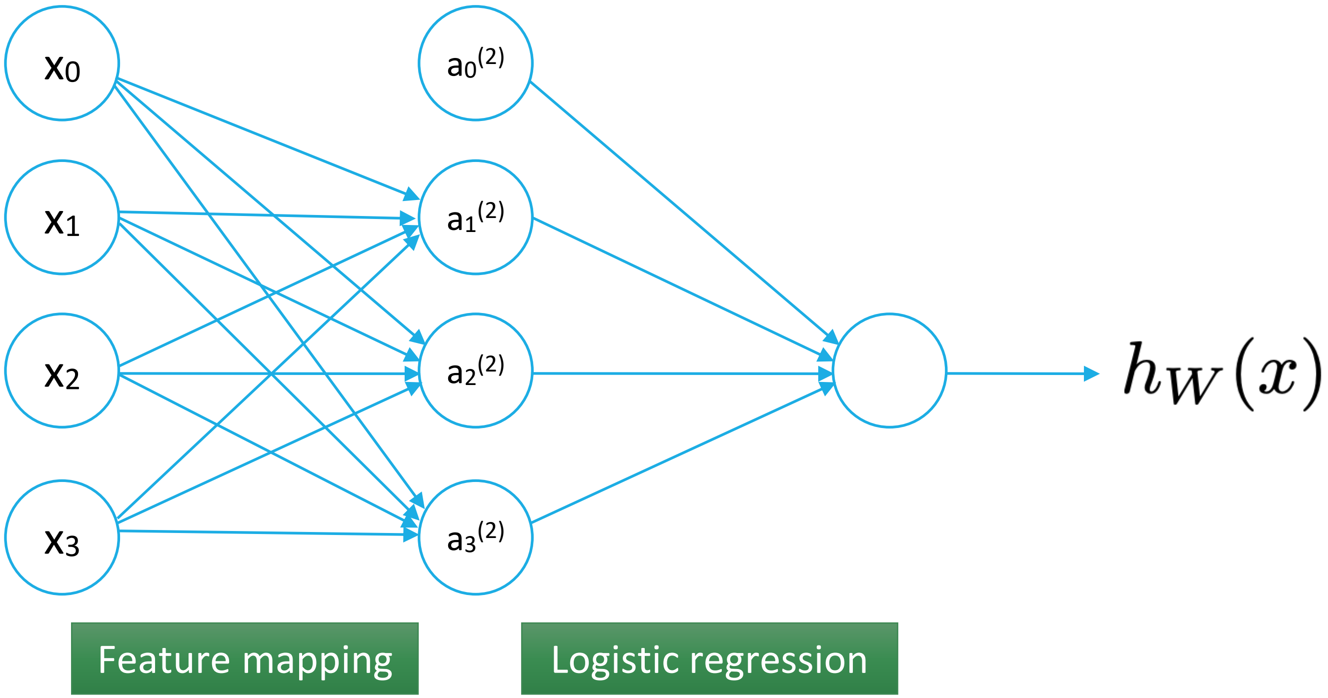
$$a_1^{(2)} = g\left(W_{10}^{(1)}x_0 + W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3\right)$$

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

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

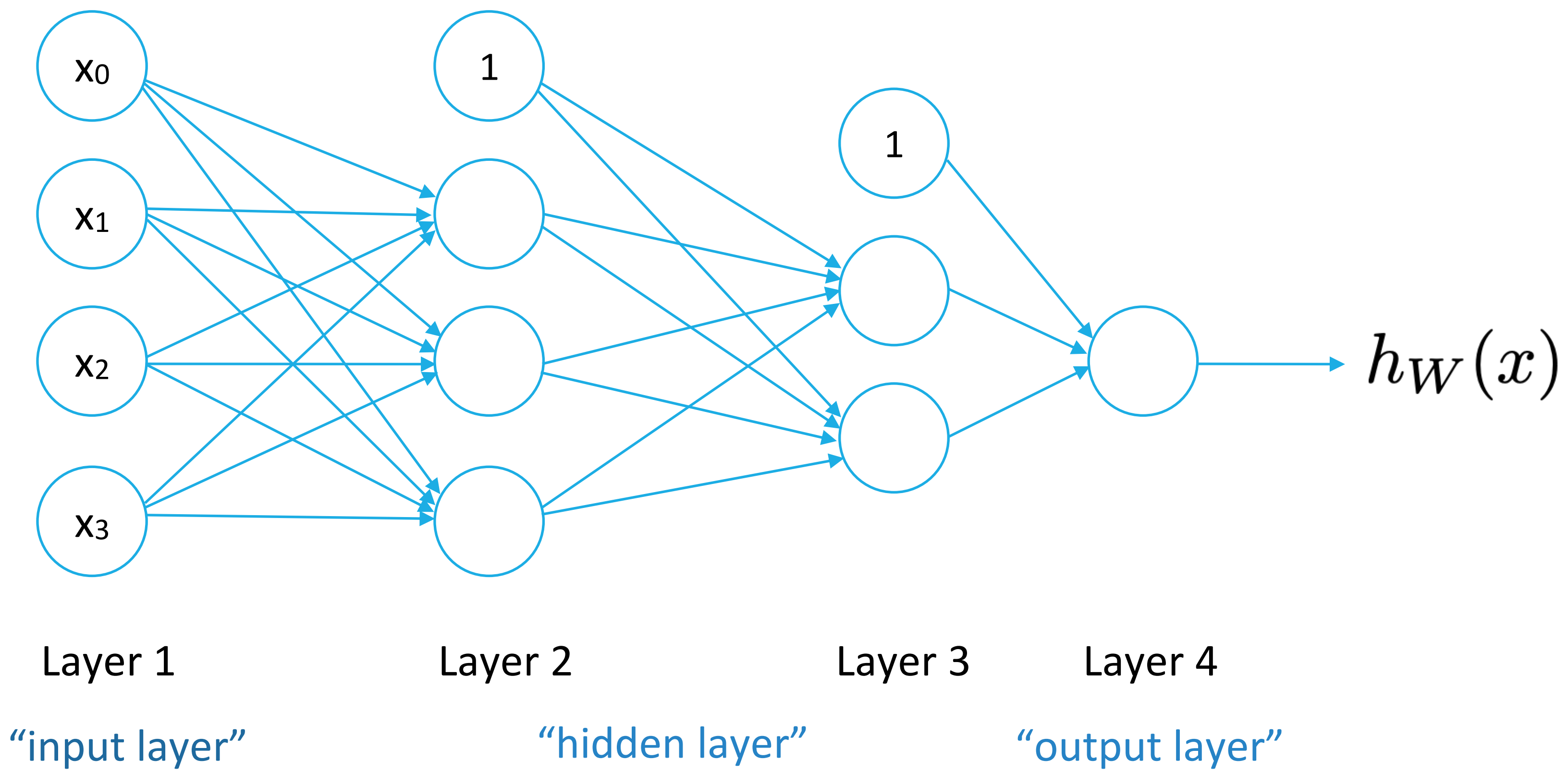
$$h_W(x) = a_1^{(3)} = g\left(W_{10}^{(2)}a_0^{(2)} + W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)}\right)$$

Feature self-learning network



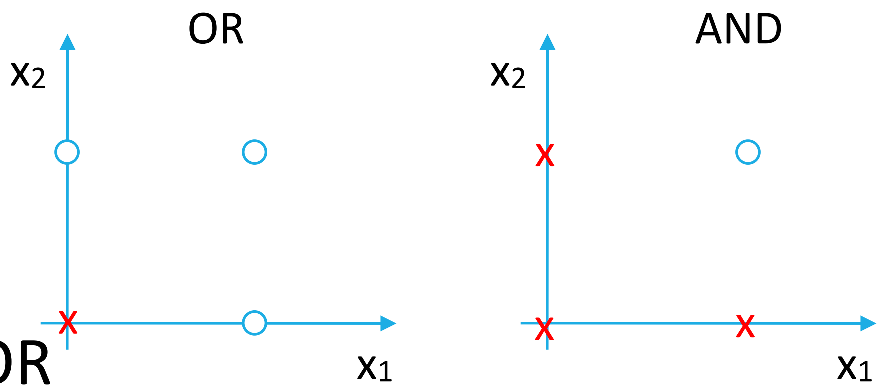
$$h_W(x) = a_1^{(3)} = g\left(W_{10}^{(2)} a_0^{(2)} + W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)}\right)$$

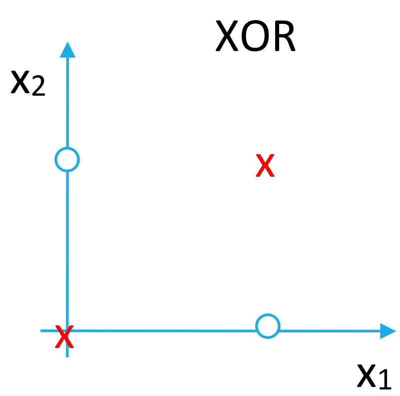
Other architectures



Non-linear classifier

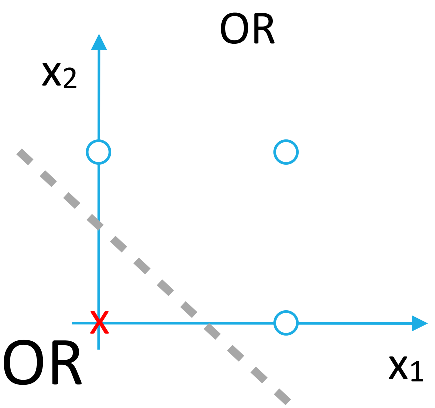
- Linear function
  - OR
  - AND
- Non-linear function XOR
  - XOR(x1, x2)
  - a1 = AND(x1, NOT(x2))
  - a2 = AND(NOT(x1), x2)
  - y = OR(a1, a2)

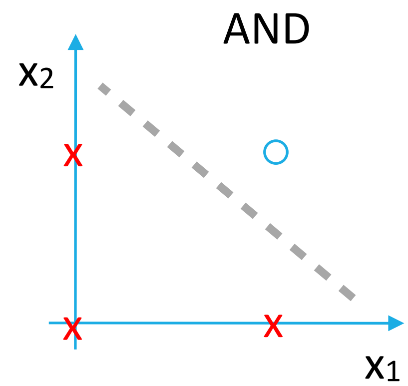


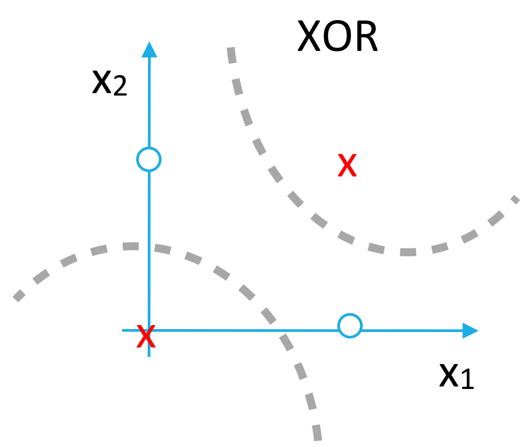


Non-linear classifier

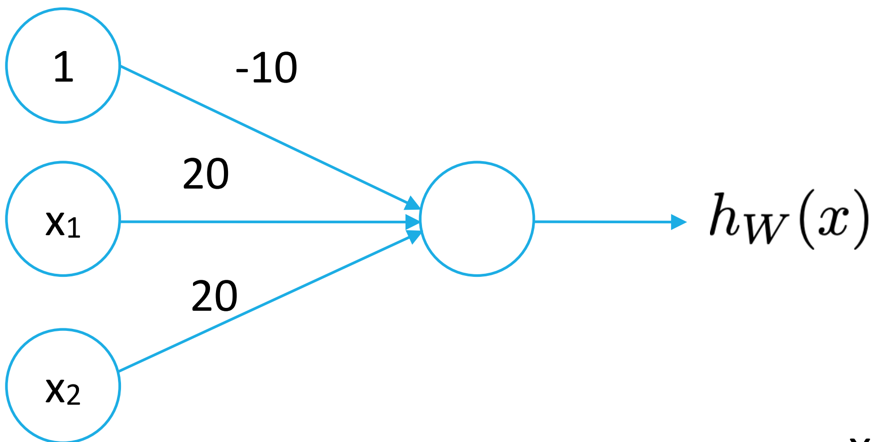
- Linear function
  - OR
  - AND
- Non-linear function XOR
  - XOR(x1, x2)
  - a1 = AND(x1, NOT(x2))
  - a2 = AND(NOT(x1), x2)
  - y = OR(a1, a2)

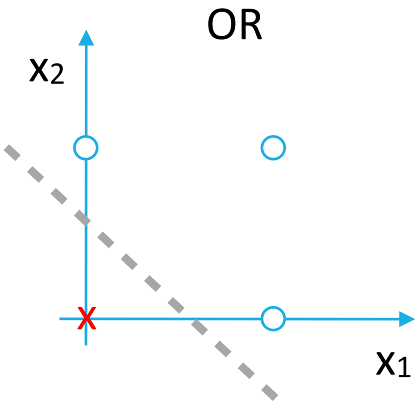




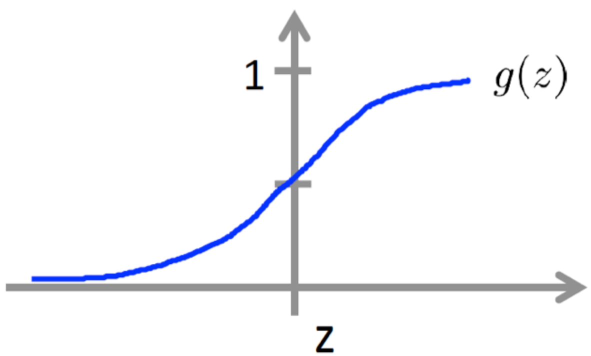


Function OR



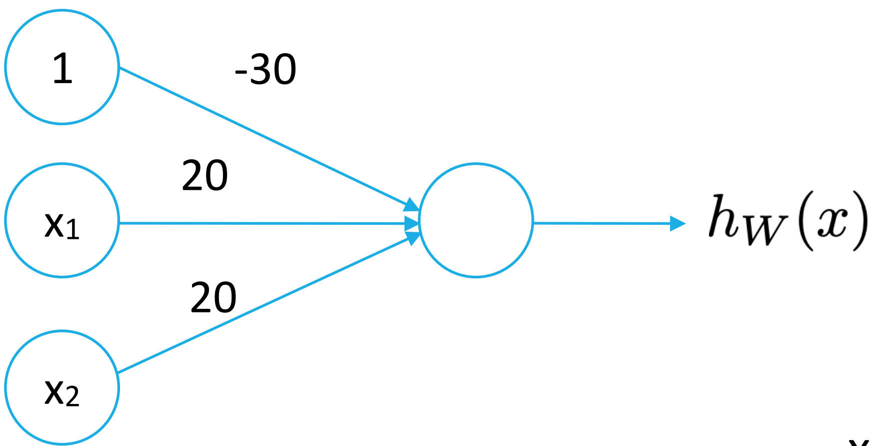


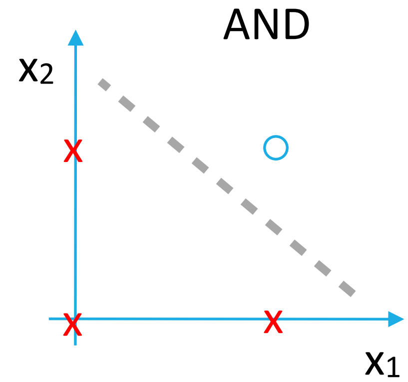
$$h_W(x) = g(-10 + 20x_1 + 20x_2)$$



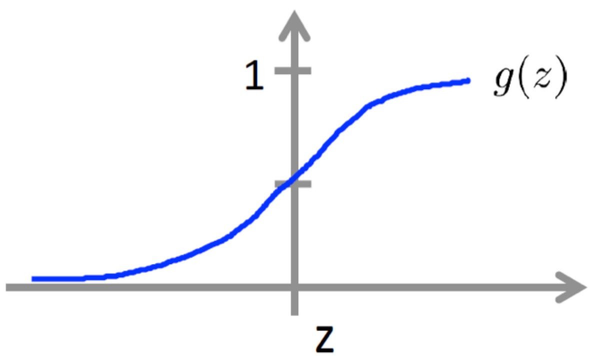
x_1	x_2	$h_W(x)$
0	0	0
0	1	1
1	0	1
1	1	1

Function AND



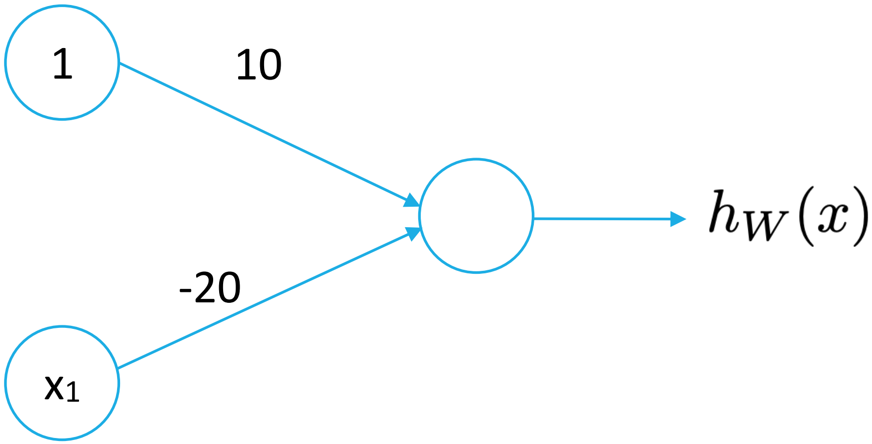


$$h_W(x) = g(-30 + 20x_1 + 20x_2)$$

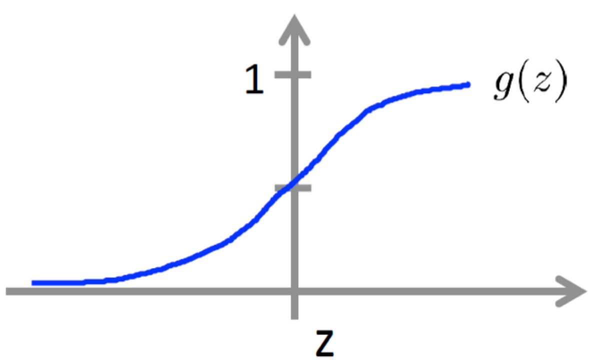


x_1	x_2	$h_W(x)$
0	0	0
0	1	0
1	0	0
1	1	1

Function NOT

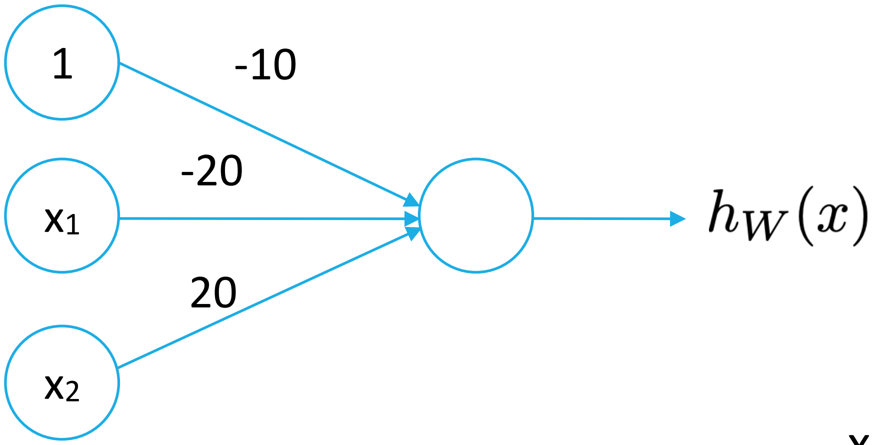


$$h_W(x) = g(10 - 20x_1)$$

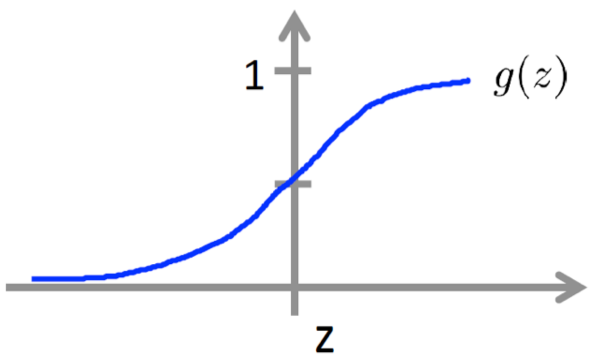


x_1	$h_W(x)$
0	1
1	0

Function AND(NOT(x1), x2)

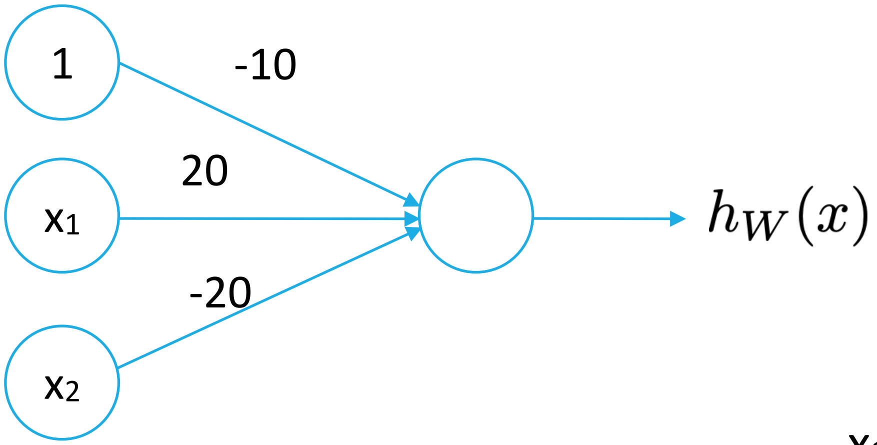


$$h_W(x) = g(-10 - 20x_1 + 20x_2)$$

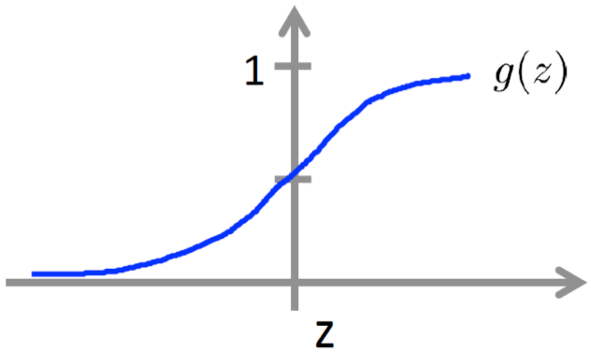


x_1	x_2	$h_W(x)$
0	0	0
0	1	1
1	0	0
1	1	0

Function AND(x1, NOT(x2))

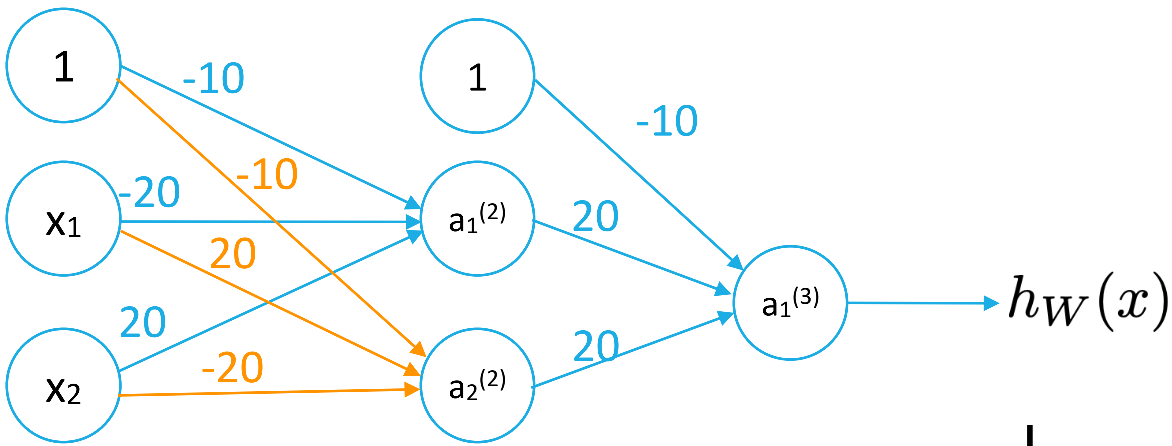


$$h_W(x) = g(-10 + 20x_1 - 20x_2)$$



x_1	x_2	$h_W(x)$
0	0	0
0	1	0
1	0	1
1	1	0

Function XOR



x_1	x_2	$h_W(x)$
0	0	0
0	1	1
1	0	1
1	1	0

$XOR(x_1, x_2) = OR(AND(NOT(x_1), x_2), AND(x_1, NOT(x_2)))$

Part 2

Training and backward propagation

Cost function

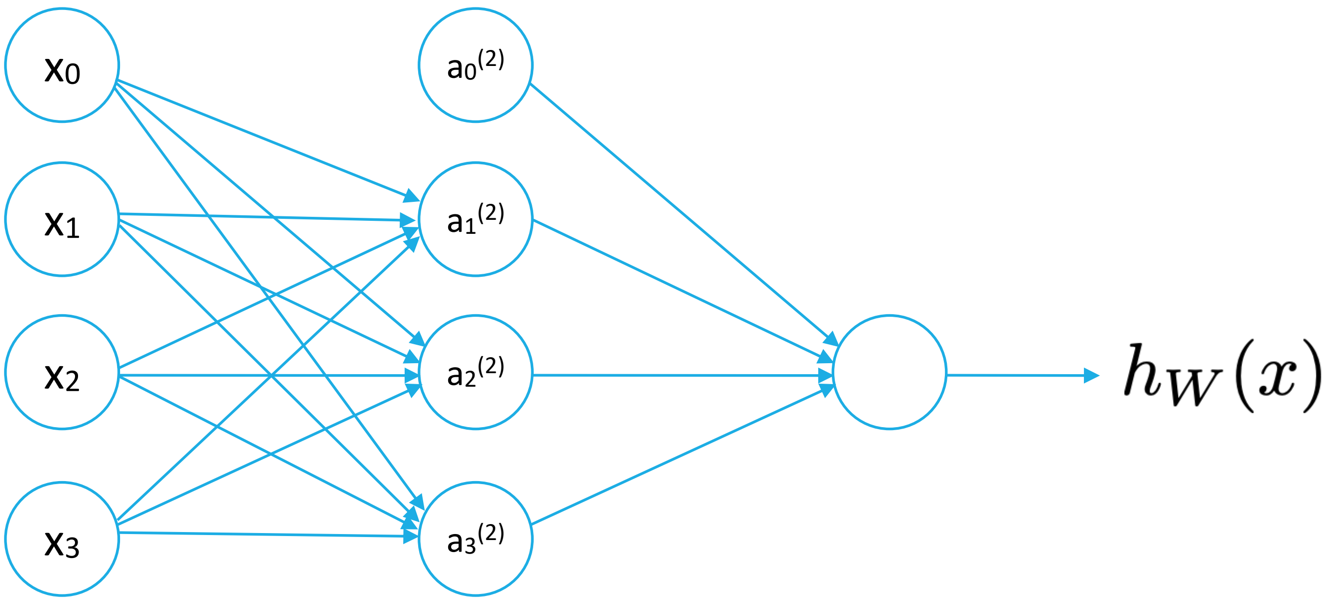
$$J(W) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log(h_W(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_W(x^{(i)}))\right]$$

Gradient

$$\frac{dJ}{dW} = \left[\frac{dJ}{dW_{10}^{(1)}}, \frac{dJ}{dW_{11}^{(1)}}, \ldots, \frac{dJ}{dW_{10}^{(L-1)}}, \ldots, \frac{dJ}{dW_{s_{l+1}s_l}^{(L-1)}}\right]$$

Derivative

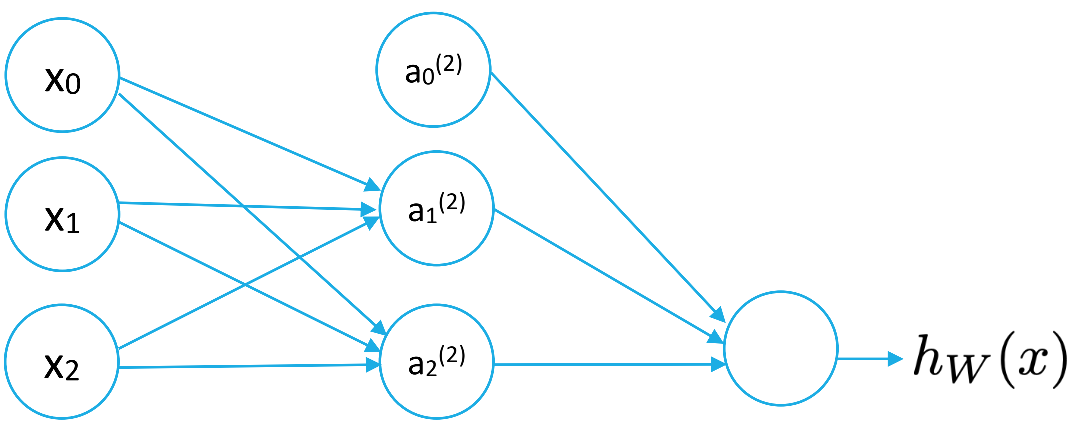
$$\frac{dJ}{dW_{ij}^{(l)}} = \sum_{k=1}^{s_{(l+1)}} \frac{dJ}{dz_k^{(l+1)}} \frac{dz_k^{(l+1)}}{dW_{ij}^{(l)}}$$



$$
\begin{aligned}
\frac{dJ}{dz_i^{(l)}} &= \sum_{j=1}^{s_{(l+1)}} \frac{dJ}{dz_j^{(l+1)}} \frac{dz_j^{(l+1)}}{dz_i^{(l)}} \\
&= \sum_{j=1}^{s_{(l+1)}} \delta_j^{(l+1)} \frac{dz_j^{(l+1)}}{dz_i^{(l)}} \\
&= \sum_{j=1}^{s_{(l+1)}} \delta_j^{(l+1)} \frac{d}{dz_i^{(l)}} \sum_{k=0}^{s_l} W_{jk}^{(l)} g(z_k^{(l)}) \\
&= \sum_{j=1}^{s_{(l+1)}} \delta_j^{(l+1)} W_{ji}^{(l)} g'(z_i^{(l)}) \\
&= g'(z_i^{(l)}) \sum_{j=1}^{s_{(l+1)}} (W^{(l)})_{ij}^T \delta_j^{(l+1)}
\end{aligned}
$$

Notation

$$
\delta_j^{(l)} = \frac{dJ}{dz_j^{(l)}}
$$



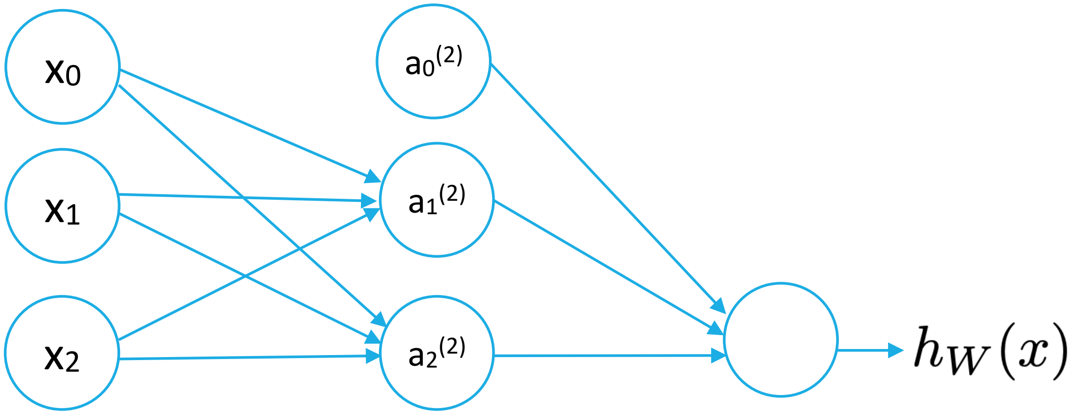
$$\frac{dJ}{dW_{ij}^{(l)}} = \sum_{k=1}^{s_{(l+1)}} \frac{dJ}{dz_k^{(l+1)}} \frac{dz_k^{(l+1)}}{dW_{ij}^{(l)}}$$

$$= \sum_{k=1}^{s_{(l+1)}} \delta_k^{(l+1)} \frac{d}{dW_{ij}^{(l)}} \sum_{t=0}^{s_l} W_{kt}^{(l)} a_t^{(l)}$$

$$= \delta_i^{(l+1)} \frac{d}{dW_{ij}^{(l)}} W_{ij}^{(l)} a_j^{(l)}$$

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

$$\frac{dJ}{dW^{(l)}} = \delta^{(l+1)} a^{(l)T}$$



Backward propagation algorithm

(1) Apply forward propagation to calculate:

$z^{(1)}, \ldots, z^{(L)}, a^{(1)}, \ldots, a^{(L)}$ và $J(z^{(L)})$

(2) $$\delta^{(L)} = \frac{dJ}{dz^{(L)}}$$

(3) for l = L-1 to 0

(4) $$\frac{dJ}{dz^{(l)}} = g'(z^{(l)})(W^{(l)T}\delta^{(l+1)})$$

(5) $$\frac{dJ}{dW^{(l)}} = \delta^{(l+1)}a^{(l)T}$$

(6) end for

Regularization

- Cost function

$$J(W) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log(h_W(x^{(i)})) + (1-y^{(i)}) \log(1-h_W(x^{(i)}))\right]$$
$$+ \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{j=1}^{s_{l+1}} \sum_{i=1}^{s_l} (W_{ji}^{(l)})^2$$

Regularization

- Gradient

$$\frac{dJ}{dz^{(l)}} = g'(z^{(l)})(W^{(l)T}\delta^{(l+1)}) + \lambda W^{(l)} \qquad \text{if} \quad j \neq 0$$

$$\frac{dJ}{dz^{(l)}} = g'(z^{(l)})(W^{(l)T}\delta^{(l+1)}) \qquad \text{if} \quad j = 0$$