Introduction to Deep Learning

Neural Network

Simple Neural Network Model

- Logistic regression model:

$$\hat{y} = \sigma(w_0 + w_1 * x_1 + w_2 * x_2)$$

- Two steps:

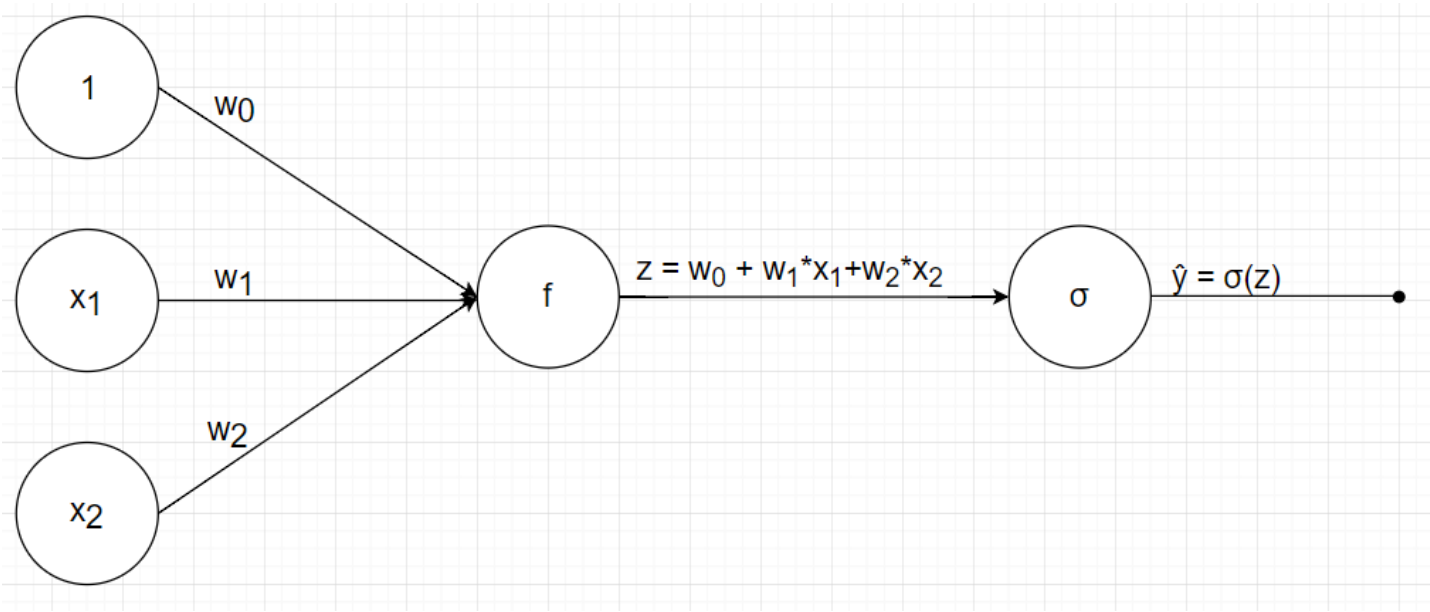
(1) Linear sum:

$$z = 1 * w_0 + x_1 * w_1 + x_2 * w_2$$

(2) Apply sigmoid function:

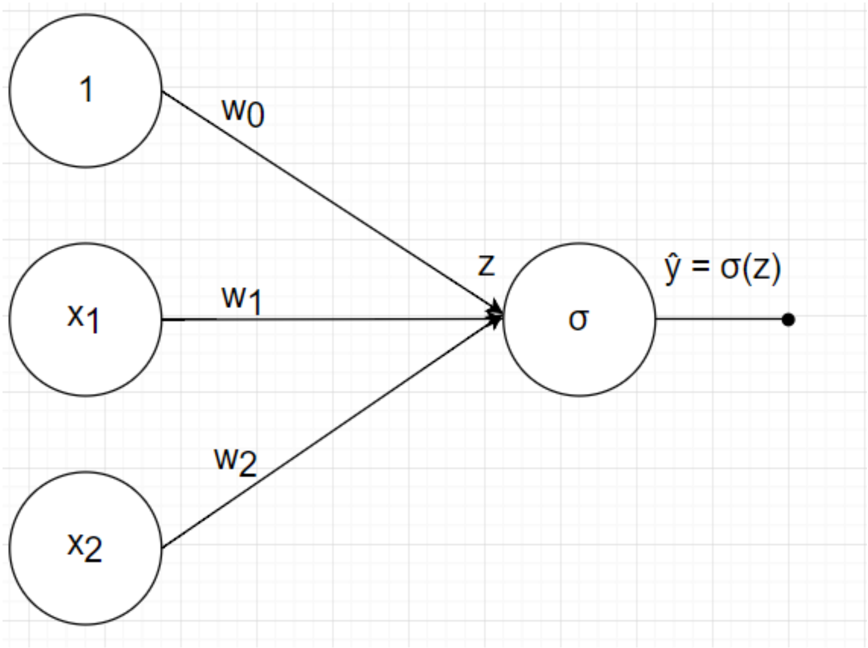
$$\hat{y} = \sigma(z)$$

Simple Neural Network Model



Flow chart of logistic regression model

Simple Neural Network Model

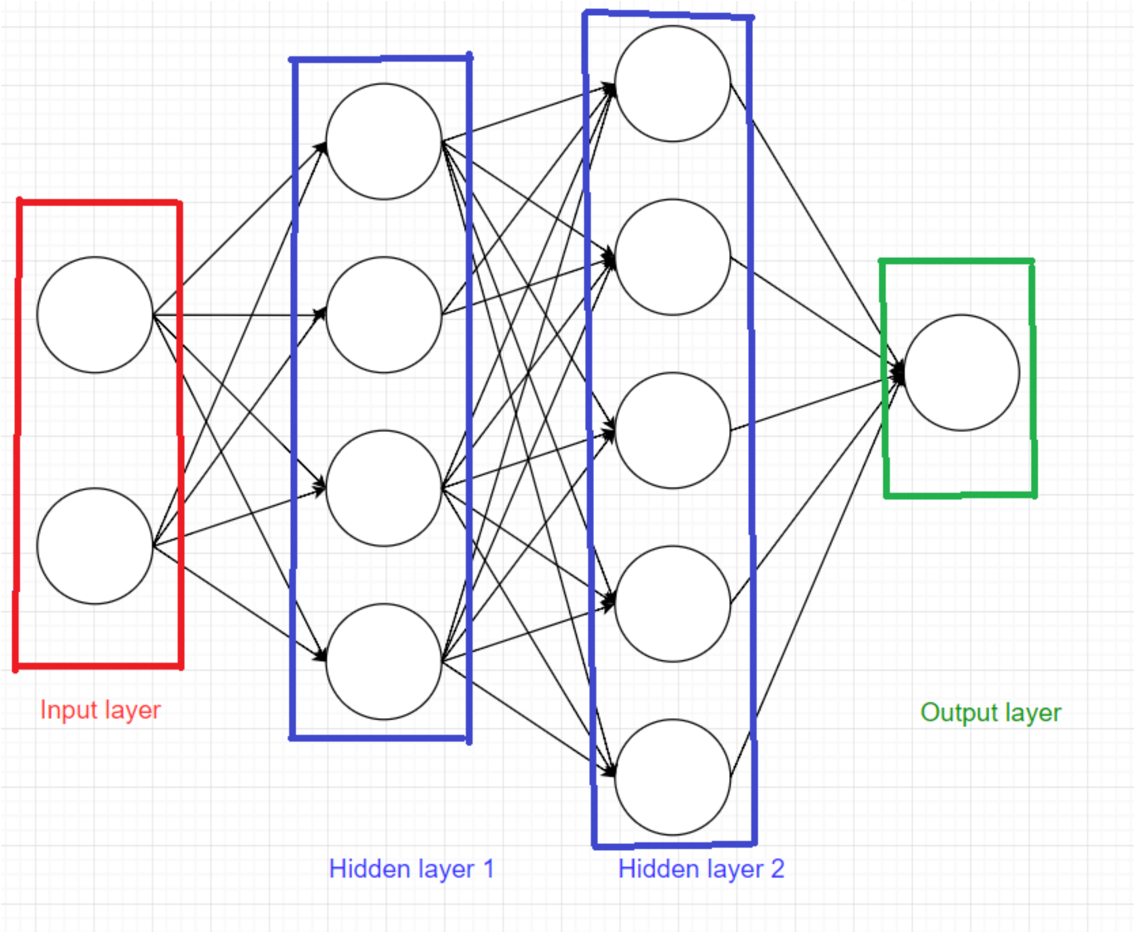


Flow chat of logistic regression model

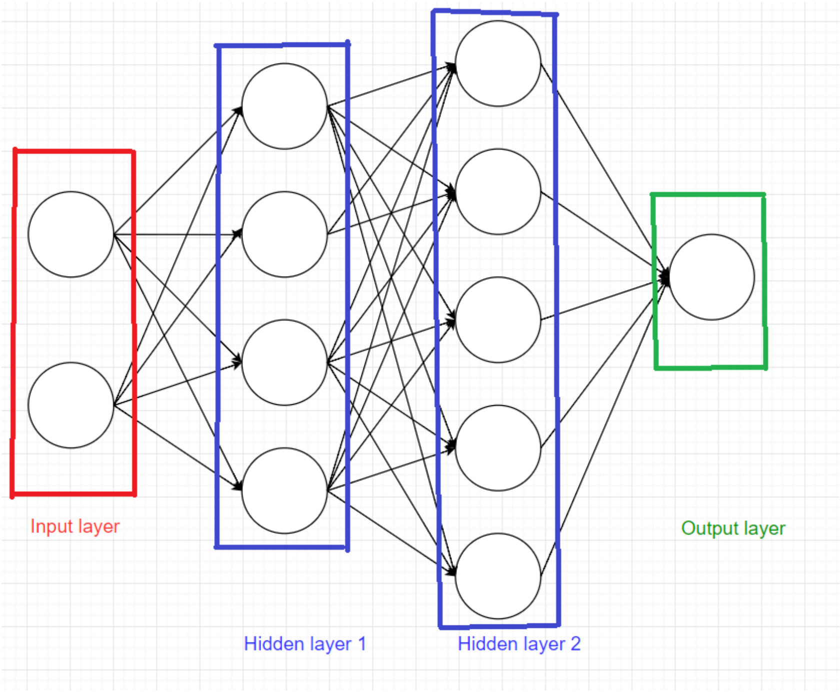
- W_0 is called bias coefficient, or free coefficient
- Sigmoid function is called activation function

General Neural Network Model

- Input layer and output layer are required
- Hidden layers are optional
- Total layers = # layers – 1
- Each circle is called one node



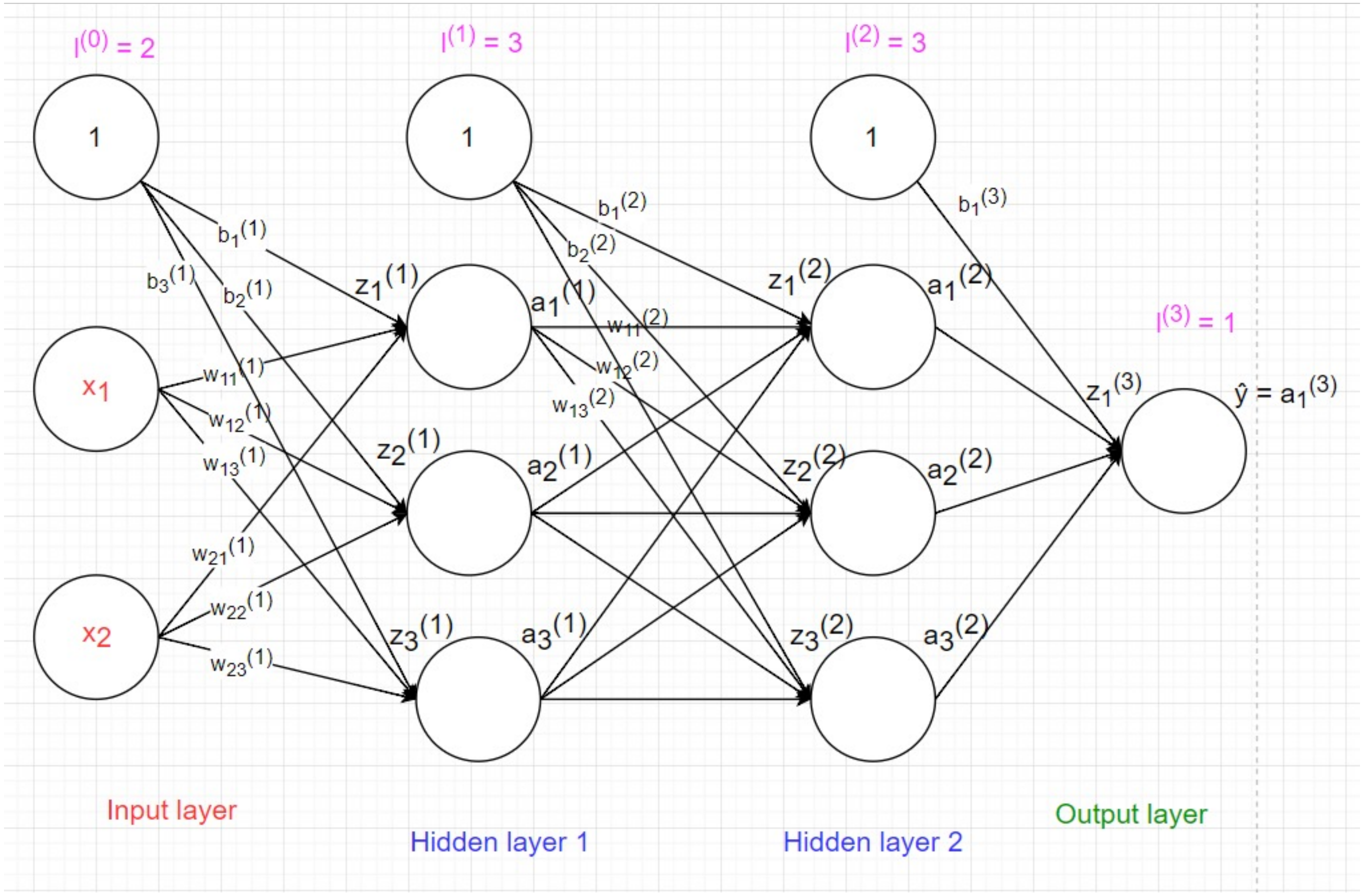
General Neural Network Model



Each node in hidden layer and output layer:

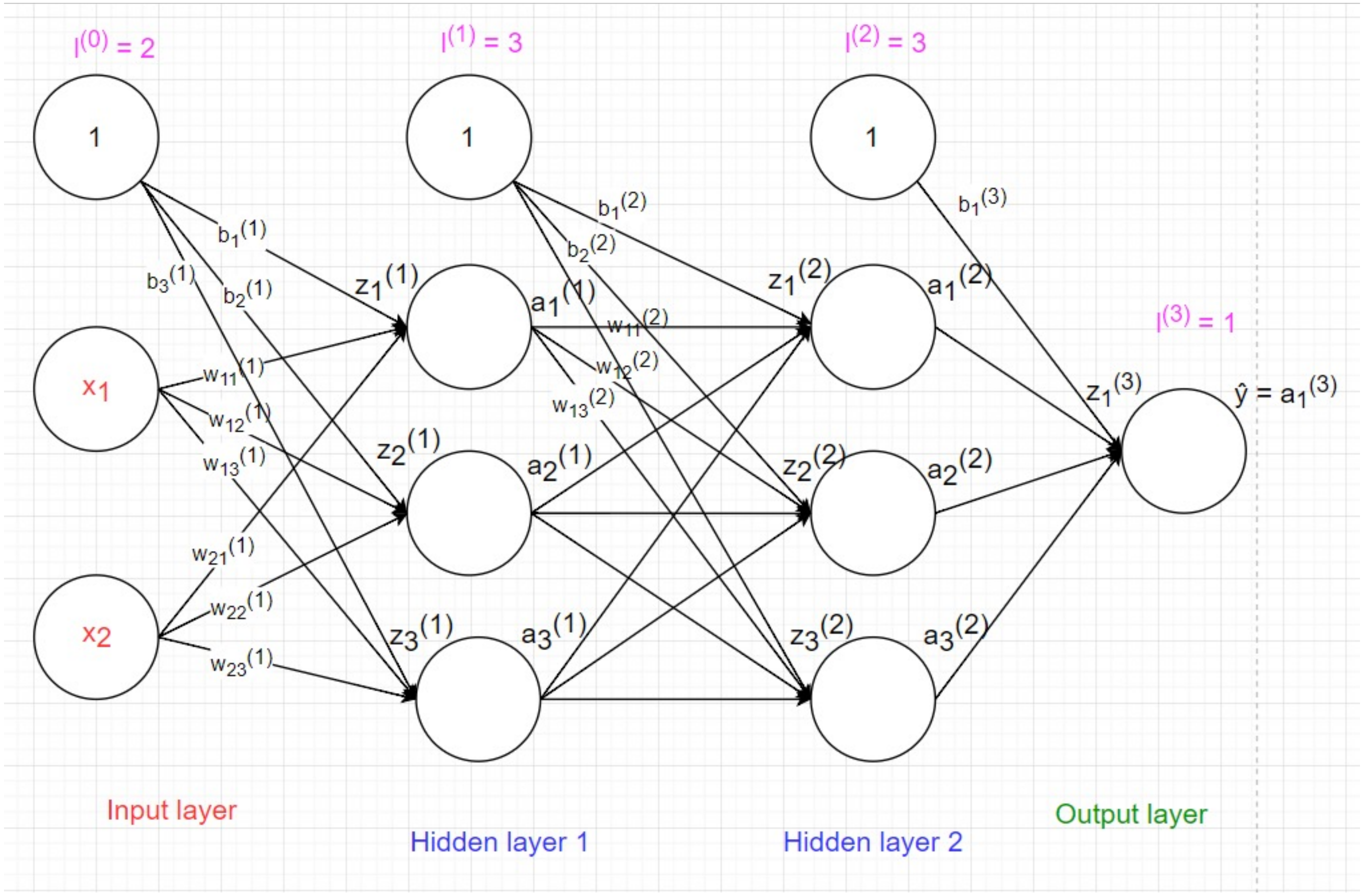
- Is connected with all nodes with previous layer with the coefficents w
- Has a bias coefficient w0
- Follows two steps of linear sum and appliance of activation function (sigmoid)

General Neural Network Model



The neural network has: 3 layers, 2 nodes in input layer, 3 nodes in hidden layer 1, 3 nodes in hidden layer 2, 1 node in output layer

General Neural Network Model



Note: node 1 is not considered as a node since it is used to calculate bias of the node in the next layer

General Neural Network Model

- Node i in layer l with bias $b_i^{(l)}$ has 2 steps:

(1) Linear sum: $z_i^{(l)} = \sum_{j=1}^{l^{(l-1)}} a_j^{(l-1)} * w_{ji}^{(l)} + b_i^{(l)}$

(2) Apply activation function: $a_i^{(l)} = \sigma(z_i^{(l)})$

General Neural Network Model

- At node 2 of layer 1, we have:

$$z_2^{(1)} = x_1 * w_{12}^{(1)} + x_2 * w_{22}^{(1)} + b_2^{(1)}$$
$$a_2^{(1)} = \sigma(z_2^{(1)})$$

- At node 3 of layer 2, we have:

$$z_3^{(2)} = a_1^{(1)} * w_{13}^{(2)} + a_2^{(1)} * w_{23}^{(2)} + a_3^{(1)} * w_{33}^{(2)}$$
$$a_3^{(2)} = \sigma(z_3^{(2)})$$

Feedforward

- Let call input layer x = $a^{(0)}$, with size 2*1, we have:

$$z^{(1)} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \end{bmatrix} = \begin{bmatrix} a_1^{(0)} * w_{11}^{(1)} + a_2^{(0)} * w_{21}^{(1)} + a_3^{(0)} * w_{31}^{(1)} + b_1^{(1)} \\ a_1^{(0)} * w_{12}^{(1)} + a_2^{(0)} * w_{22}^{(1)} + a_3^{(0)} * w_{32}^{(1)} + b_2^{(1)} \\ a_1^{(0)} * w_{13}^{(1)} + a_2^{(0)} * w_{23}^{(1)} + a_3^{(0)} * w_{33}^{(1)} + b_3^{(1)} \end{bmatrix}$$

$$= (W^{(1)})^T * a^{(0)} + b^{(1)}$$

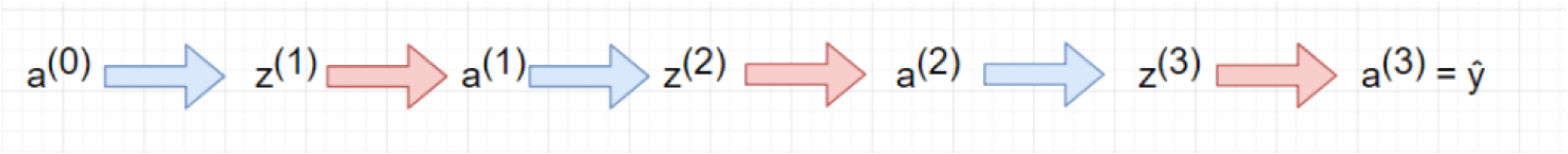
$$a^{(1)} = \sigma(z^{(1)})$$

Feedforward

- Similarly, we have:

$$z^{(2)} = (W^{(2)})^T * a^{(1)} + b^{(2)}$$
$$a^{(2)} = \sigma(z^{(2)})$$
$$z^{(3)} = (W^{(3)})^T * a^{(2)} + b^{(3)}$$
$$\hat{y} = a^{(3)} = \sigma(z^{(3)})$$

Feedforward



feedforward neural network

Loss function

- Gradient descent algorithm
- Step of derivative calculation of coefficients of loss function is done with the backpropagation algorithm

 Will be taught in the next lecture

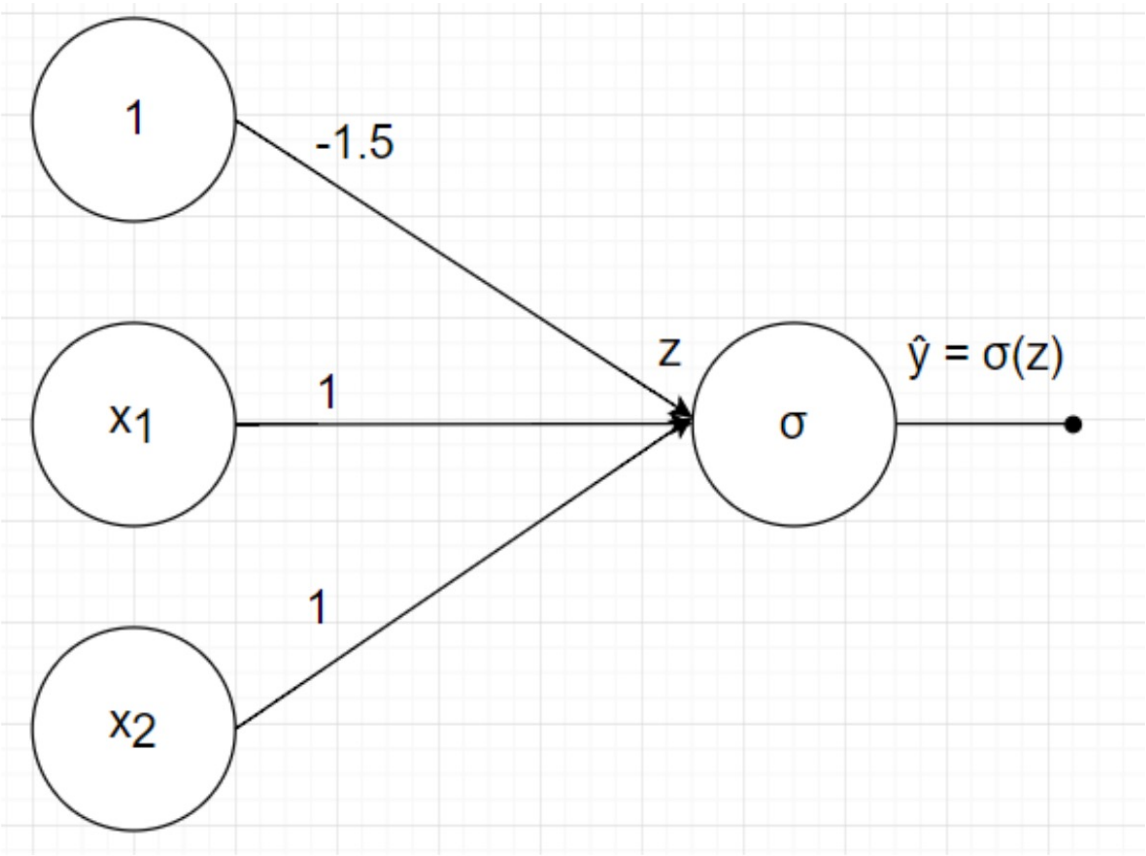
Logistic Regression vs. Neural Network

Problem: AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

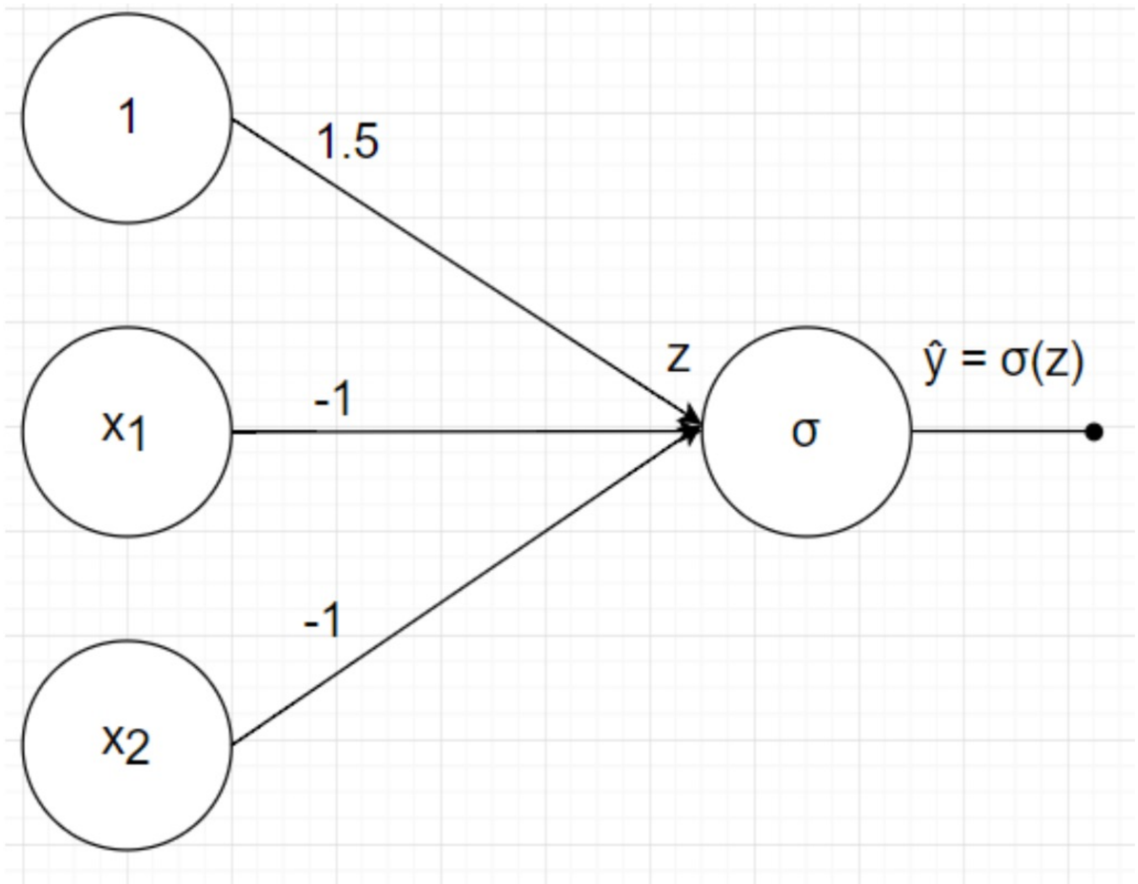
x_1 AND x_2

Logistic Regression vs. Neural Network



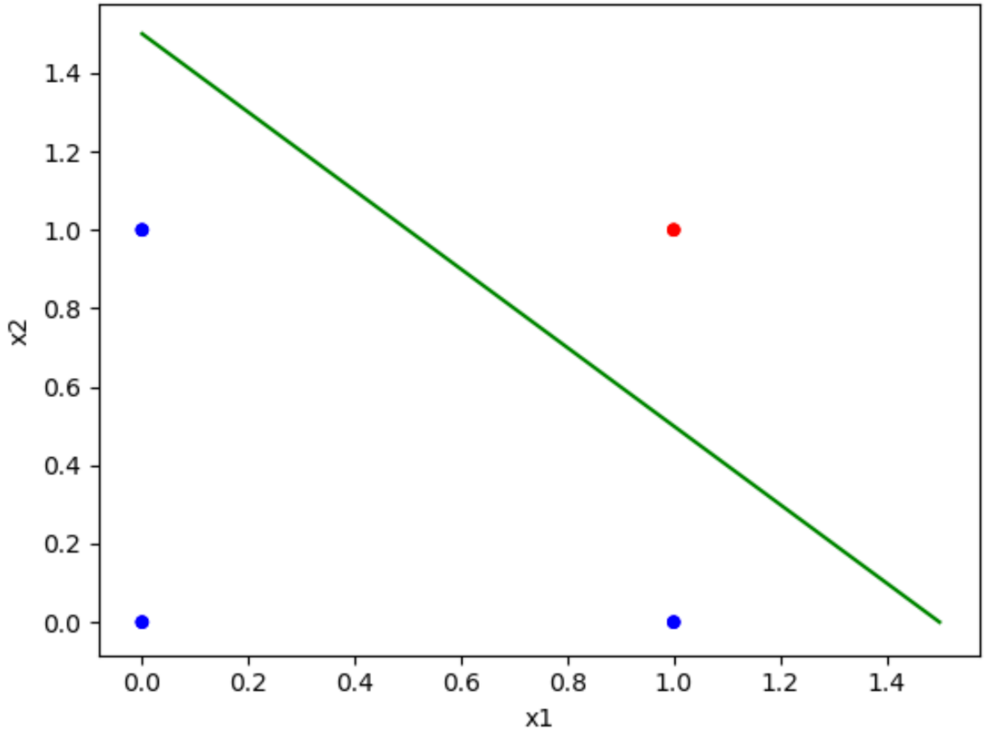
Flow chart for the problem x_1 AND x_2

Logistic Regression vs. Neural Network



Flow chart for the problem NOT (x_1 AND x_2)

Logistic Regression vs. Neural Network



Separation line $y = 1.5 - 1 * x_1 - 1 * w_2$

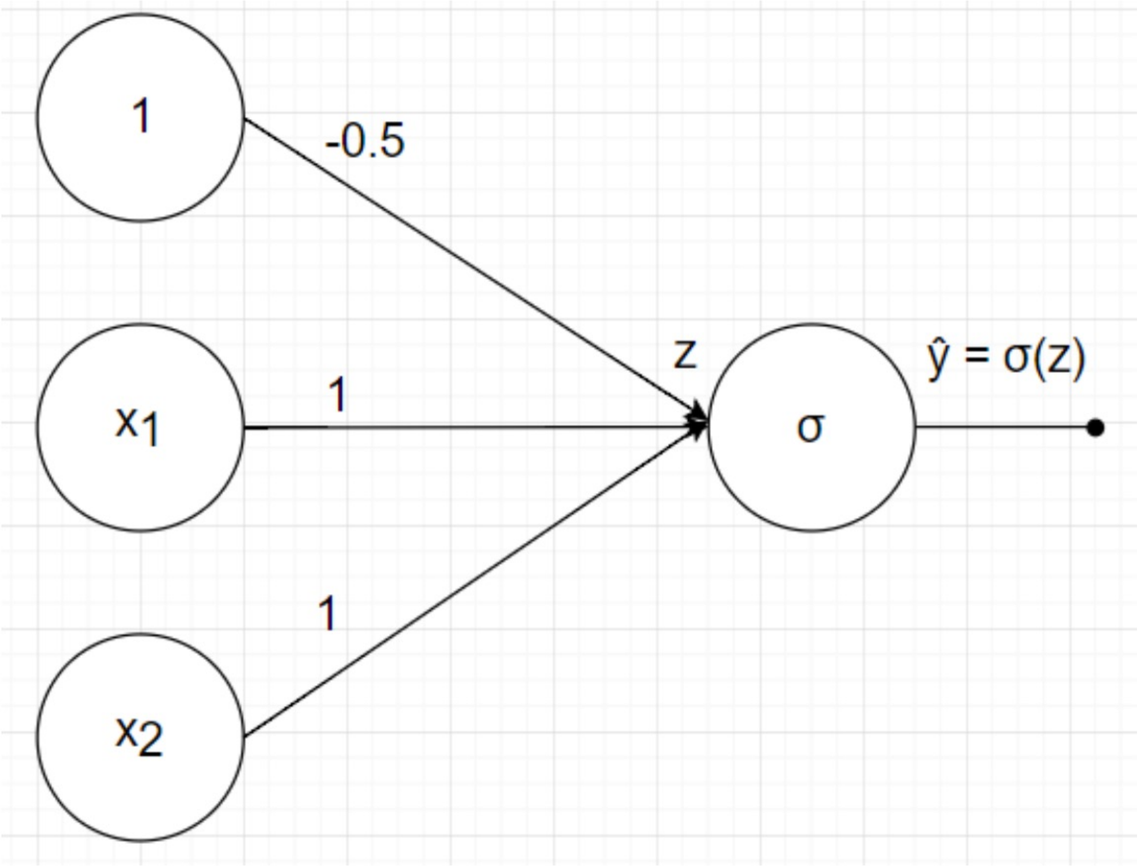
Logistic Regression vs. Neural Network

Problem: OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

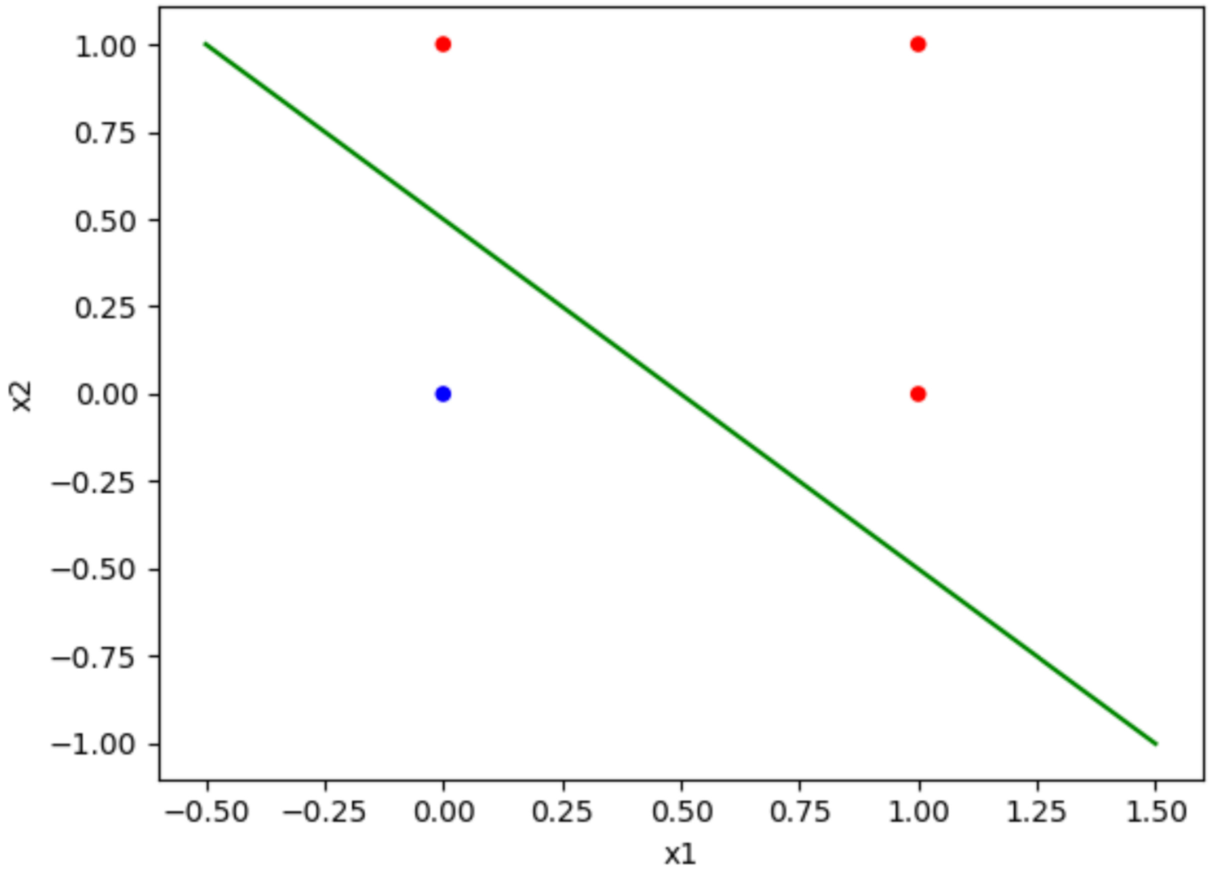
x_1 OR x_2

Logistic Regression vs. Neural Network



Flow chart for the problem x_1 OR x_2

Logistic Regression vs. Neural Network



Separation line for OR problem

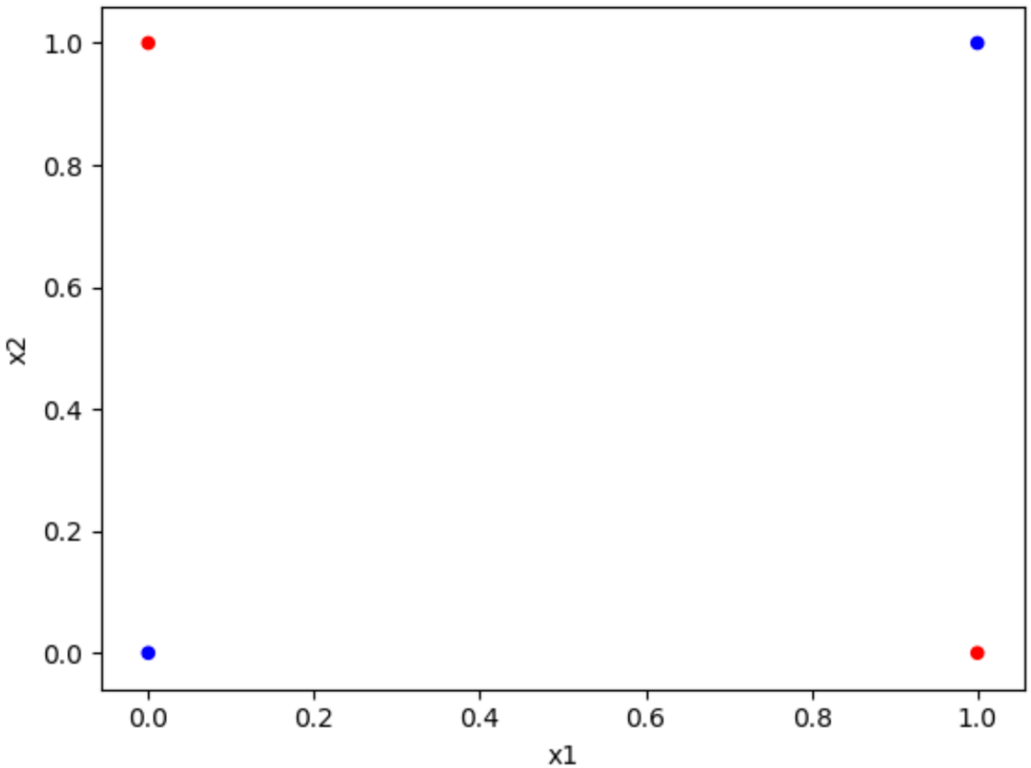
Logistic Regression vs. Neural Network

Problem: XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

x_1 XOR x_2

Logistic Regression vs. Neural Network



Cannot draw a separation line for XOR problem

→ Cannot solve the XOR problem using logistic regression

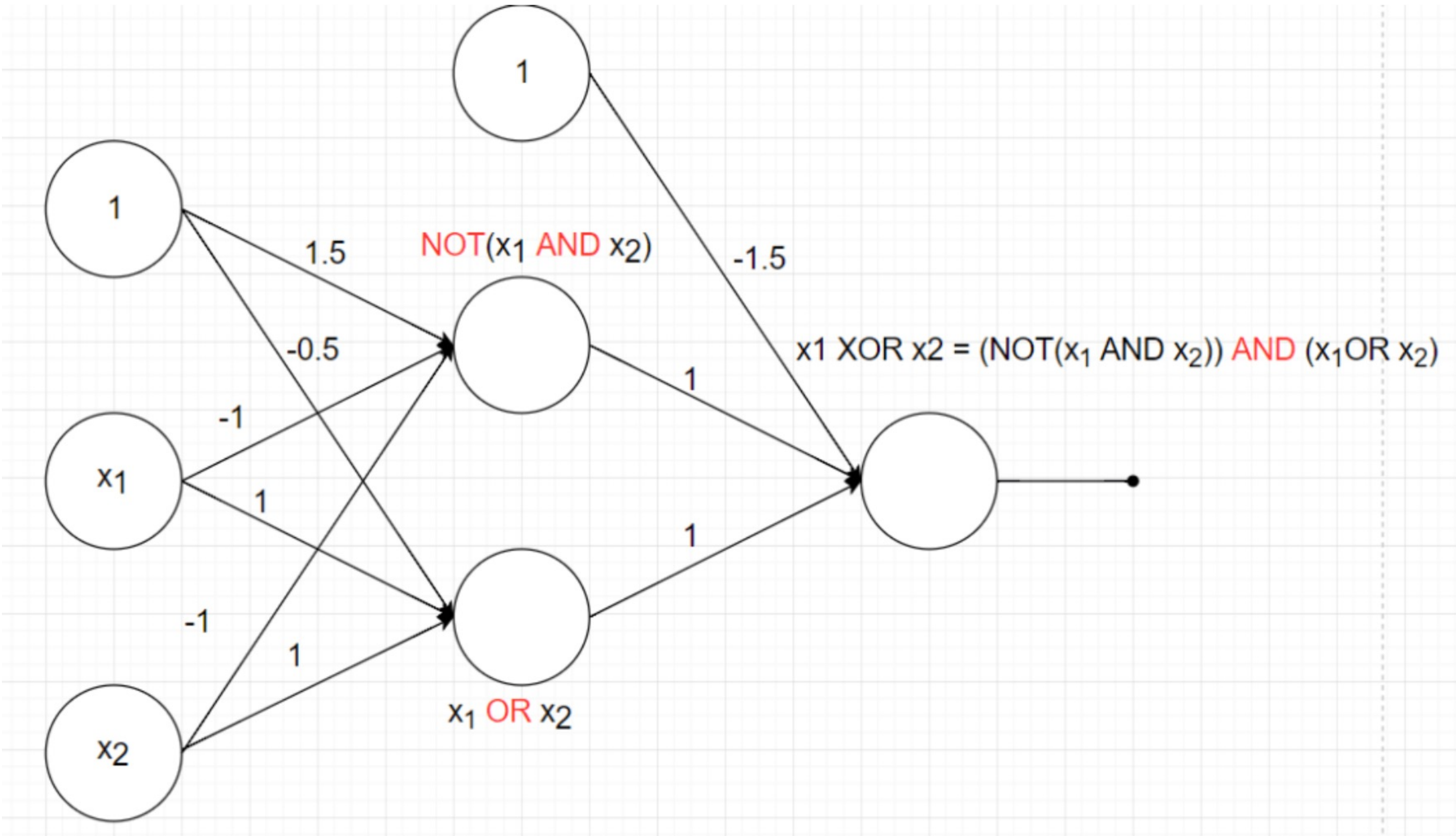
→ Neural network ???

Logistic Regression vs. Neural Network

Rewrite XOR problem:

A	B	**A XOR B**	A AND B	NOT (A AND B)	A OR B	**(NOT(A AND B) AND (A OR B))**
0	0	0	0	1	0	0
0	1	1	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	0	1	0

Logistic Regression vs. Neural Network



Solve XOR problem with several logistic regression models