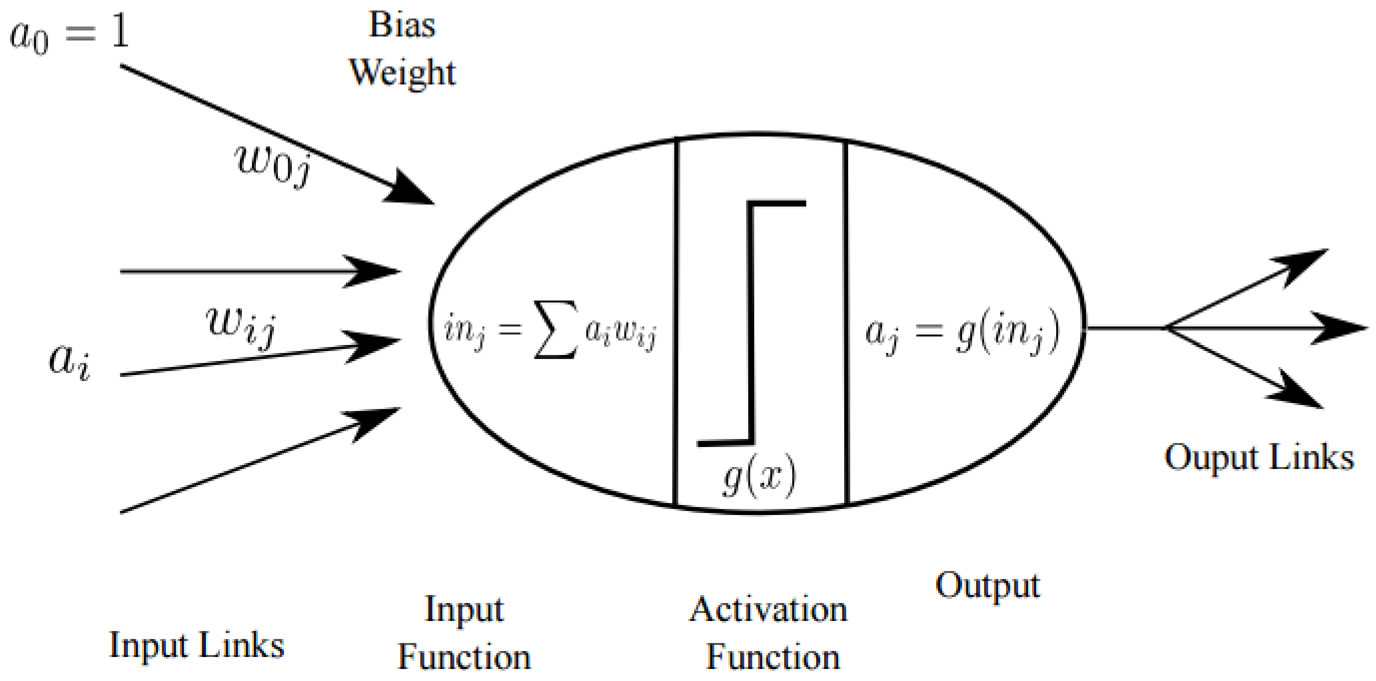
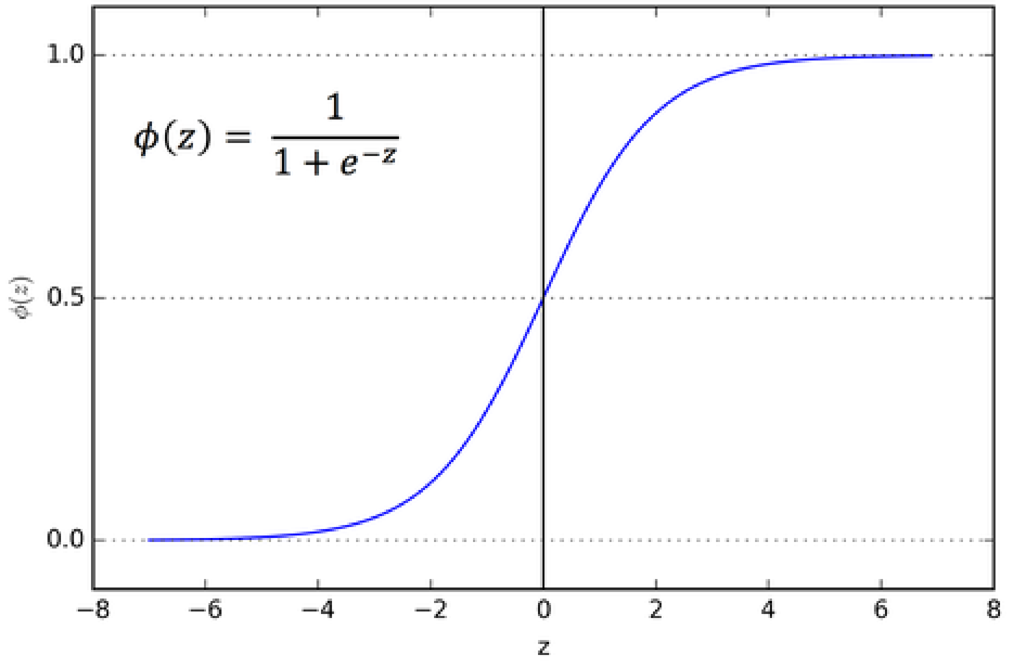
# A Neuron Model



# A Neuron Model

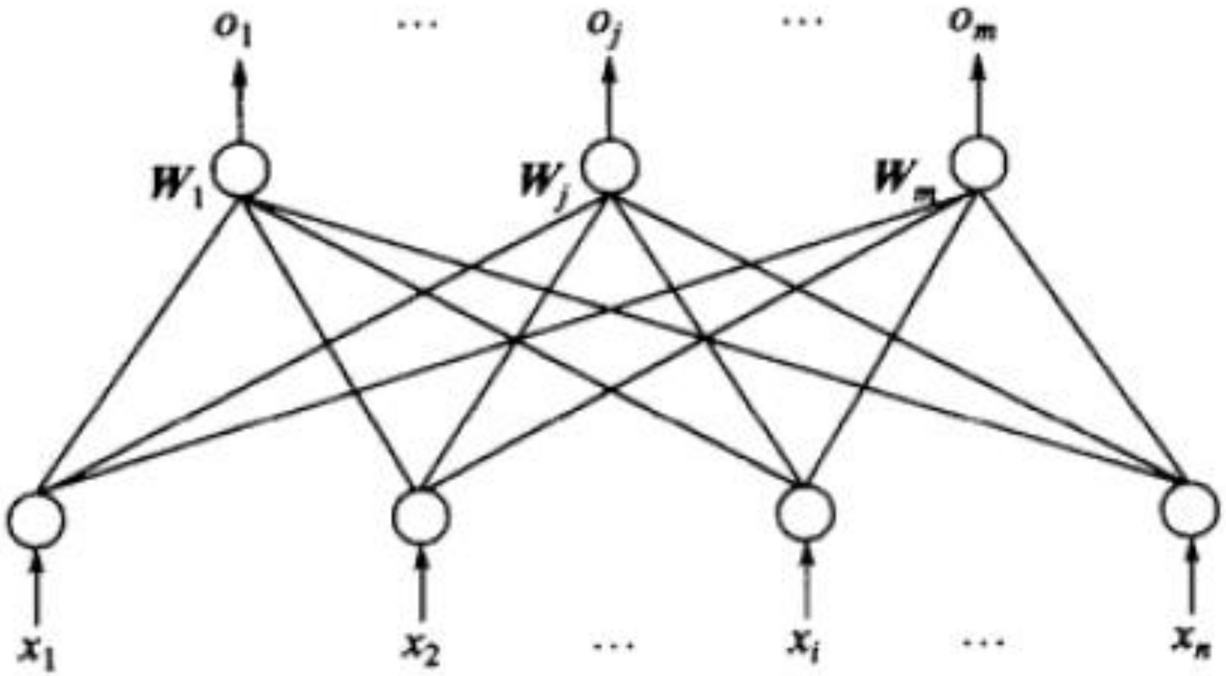
- Activation Function
  - Sigmoid



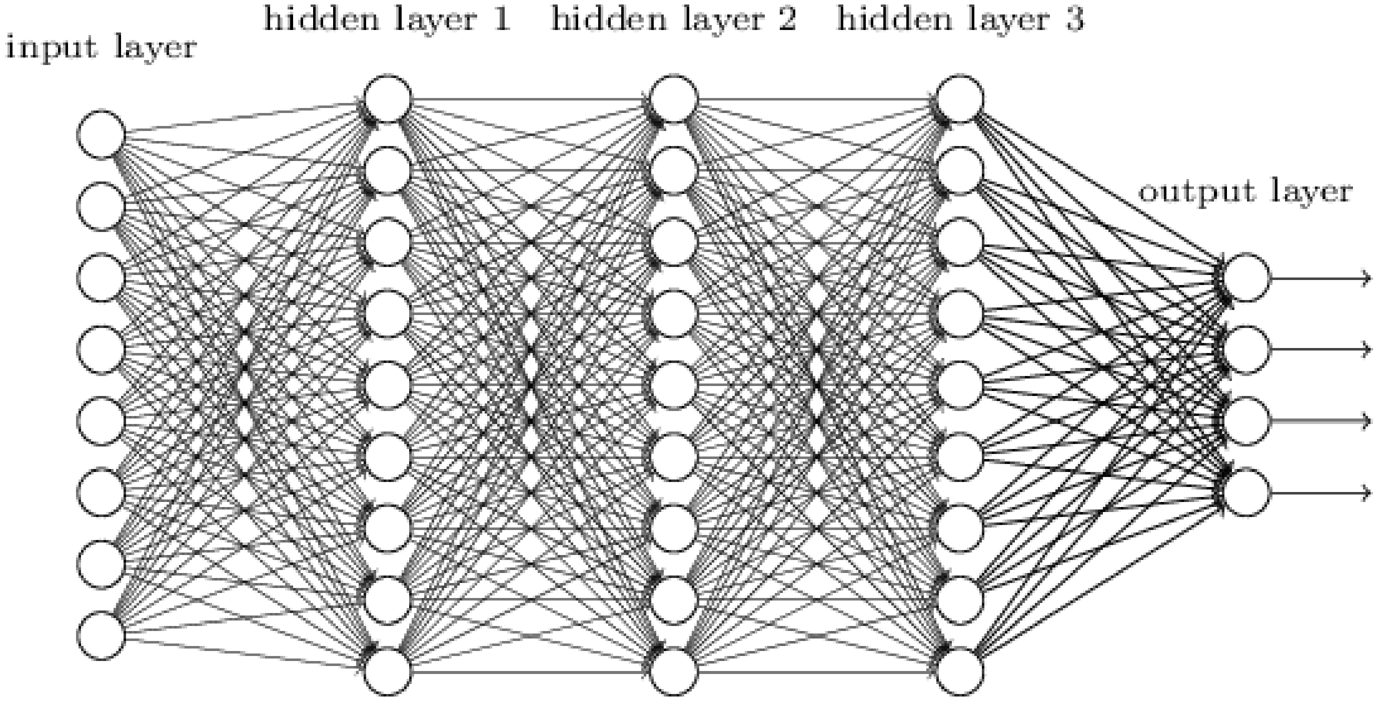
# A Neuron Model

- The **weights** of the neuron are the parameters of the model.
- The input is computed as a weighted sum of the inputs.
- The **activation function** determines the output and can be different functions.
- The output is obtained by applying the activation function to the input.
- How does the neuron work?

# Single-layer Neural Networks



# Multiple-layers Neural Network



# Delta Learning Rule

- Based on gradient descent.
- Only applicable for continuous activation function.
- The learning signal ***r*** is called delta and defined as:

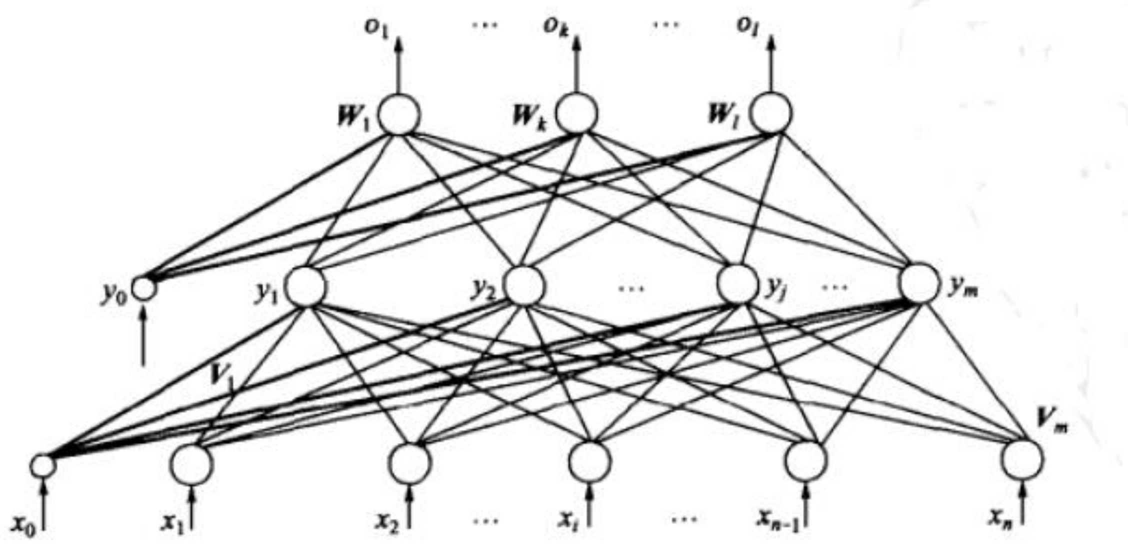
$$r = [d_j - f(\mathbf{W}_j^T \boldsymbol{X})] f'(\mathbf{W}_j^T \boldsymbol{X})$$
$$= (d_j - o_j) f'(\mathrm{net}_j)$$

- The weights will be adjusted as:

$$\Delta \mathbf{W}_j = \eta (d_j - o_j) f'(\mathrm{net}_j) \mathbf{X}$$

# Backpropagation Algorithm

- BP algorithm is a process of training a neural network
- Based on delta learning rule



# Backpropagation Algorithm

$$\delta_k^o = (d_k - o_k) o_k (1 - o_k)$$

$$\delta_j^y = \left[ \sum_{k=1}^{l} (d_k - o_k) f'(\mathrm{net}_k) w_{jk} \right] f'(\mathrm{net}_j)$$

$$= \left( \sum_{k=1}^{l} \delta_k^o w_{jk} \right) y_j (1 - y_j)$$

- For the output layer

$$\Delta w_{jk}^{h+1} = \eta \delta_k^{h+1} y_j^h = \eta (d_k - o_k) o_k (1 - o_k) y_j^h \quad j = 0,1,2,\cdots,m_h;\ k = 1,2,\cdots,l$$

- For the ***h*** *th* hidden layer

$$\Delta w_{ij}^h = \eta \delta_j^h y_i^{h-1} = \eta \left( \sum_{k=1}^{l} \delta_k^o w_{jk}^{h+1} \right) y_j^h (1 - y_j^h) y_i^{h-1} \quad i = 0,1,2,\cdots,m_{h-1};\ j = 1,2,\cdots,m_h$$

- The first hidden layer

$$\Delta w_{pq}^1 = \eta \delta_q^1 x_p = \eta \left( \sum_{r=1}^{m_2} \delta_r^2 w_{qr}^2 \right) y_q^1 (1 - y_q^1) x_p \qquad p = 0,1,2,\cdots,n;\ j = 1,2,\cdots,m_1$$

# Backpropagation Algorithm

- The flow of signal in BP algorithm: