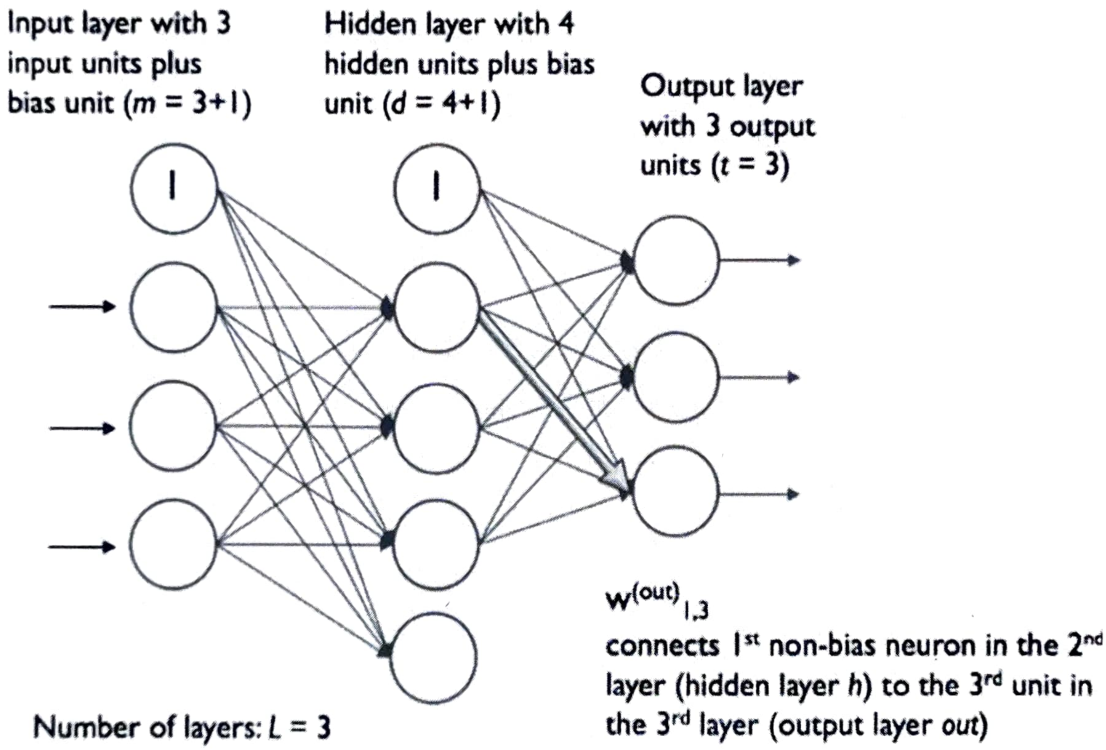
Multi-layer perceptron network

A fully connected multi-layer neural network is called a Multilayer Perceptron (MLP).



It has 3 layers including one hidden layer. If it has more than 1 hidden layer, it is called a deep ANN. An MLP is a typical example of a feedforward artificial neural network. In this figure, the ith activation unit in the lth layer is denoted as ai(l).

The number of layers and the number of neurons are referred to as hyperparameters of a neural network, and these need tuning. Cross-validation techniques must be used to find ideal values for these.

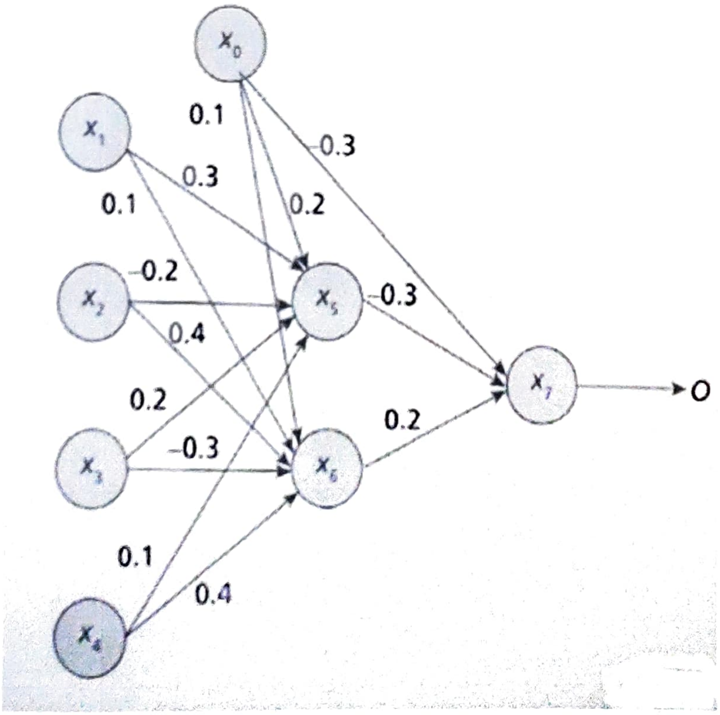
The weight adjustment training is done via backpropagation. Deeper neural networks are better at processing data. However, deeper layers can lead to vanishing gradient problems. Special algorithms are required to solve this issue.

Forward Propagation:

The MLP learning procedure is as follows:

- Starting with the input layer, propagate data forward to the output layer. This step is the forward propagation.
- Based on the output, calculate the error (the difference between the predicted and known outcome). The error needs to be minimized.
- Backpropagate the error. Find its derivative with respect to each weight in the network, and update the model.
- Repeat the three steps given above over multiple epochs to learn ideal weights.
- Finally, the output is taken via a threshold function to obtain the predicted class labels.

Forward Propagation in MLP:



Learning rate = 0.8

x_1	x_2	x_3	x_4	output
1	1	0	1	1

Calculate i/p to neuron x_5

$$\therefore I_5 = (1 * 0.3) + (1 * -0.2) + (0 * 0.2) + (1 * 0.1) + (1 * 0.2)$$

$$I_5 = 0.4$$

Calculate i/p to neuron x_6

$$I_6 = (1 * 0.3) + (1 * 0.4) + (0 * -0.3) + (1 * 0.4) + (1 * 0.1)$$

$$I_6 = 1.2$$

Calculate i/p to neuron x_7

To calculate input to neuron x_7, first calculate output from neurons x_5 and x_6. Sigmoidal activation function is used here.

$$\therefore O_5 = \frac{1}{1+e^{-I_5}} = \frac{1}{1+e^{-0.4}} = 0.599$$

$$O_6 = \frac{1}{1+e^{-I_6}} = \frac{1}{1+e^{-1.2}} = 0.769$$

Now use these outputs O_6 and O_7 to calculate input to neuron x_7.

$$\therefore I_7 = (O_5 * -0.3) + (O_6 * 0.2) + (1 * -0.3)$$
$$= (0.599 * -0.3) + (0.769 * 0.2) + (1 * -0.3)$$
$$I_7 = -0.326$$

Now use this input I_7 to calculate final output of this iteration No. 1.

$$\therefore O_7 = \frac{1}{1+e^{-I_7}} = \frac{1}{1+e^{0.326}} = 0.419$$

Calculate Error:

Desired output = 1

Output from the network = 0.419

∴ Error = 1 − 0.419 = 0.581

- To update weight vectors, this error term need to be back propogated until desired accuracy is achieved.