< Back-Propagation training algorithm >

Step 1: Initialization

Set all the weights and threshold levels of the network to random numbers uniformly distributed inside a small range (Haykin, 1999) :

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i}\right)$$

where F_i is the total number of inputs of neuron i in the network. The weight initialization in done on a neuron-by-neuron basis.

Step 2: Activation

Activate the back-propagation neural network by applying inputs $x_1(p), x_2(p), \ldots, x_n(p)$ and desired outputs $y_{d,1}(p), y_{d,2}(p), \ldots, y_{d,n}(p)$.

(a) Calculate the actual outputs of the neurons in the hidden layer:

$$y_j(p) = \text{sigmoid}\left[\sum_{i=1}^{n} x_i(p) \times w_{i}\right.$$

where n is the number of the inputs of neuron j in the hidden layer, and sigmoid is the sigmoid activation function.

(b) Calculate the actual output of the neurons in the output layer:

$$y_k(p) = \text{sigmoid}\left[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}\right.$$

where m is the number of the inputs of neuron k in the output layer.

Step 3: Weight training

Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

(a) Calculate the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)$$

where

$$e_k(p) = y_{d,k}(p$$

Calculate the weight corrections:

$$\Delta w_{jk}(p) = \alpha \times y_j(p$$

Update the weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) +$$

(b) Calculate the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p)$$

Calculate the weight corrections:

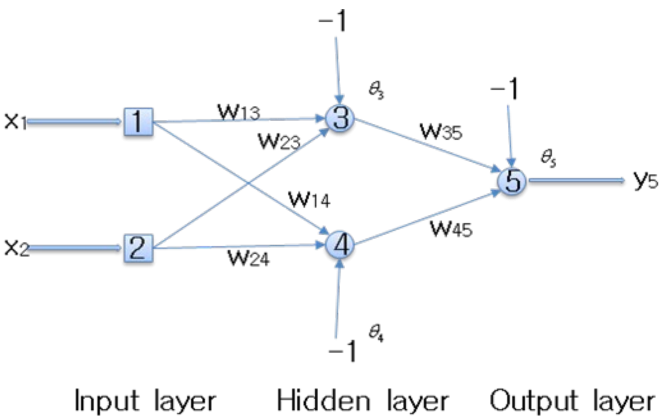
$$\Delta w_{ij}(p) = \alpha \times x_i(p$$

Update the weights at the hidden neurons:

$$w_{ij}(p+1) = w_{ij}(p)$$

Step 4: Iteration

Increase iteration p by one, go back to Step 2 and repeat the process until the selected error criterion is satisfied.



The initial weights and threshold levels are set randomly as follows.

$w_{13} = 0.5, w_{14} = 0.9, w_{23} = 0.4, w_{24} = 1.0, w_{35} = -1.2, w_{45} = 1.1, \theta_3 = 0.8, \theta_4 = -0.1$ and $\theta_5 = 0.3$

Consider a training set where inputs x_1 and x_2 are equal to 1 and desired output $y_{d,5}$ is 0.

The actual outputs of neurons 3 and 4 in the hidden layer are calculated as

$$y_3 = \text{sigmoid}[x_1 \times w_{13} + x_2 \times w_{23} - \theta_3] = \frac{1}{1 + e^{-(1\times 0.5 + 1\times 0.4 - 1\times 0.8)}} = 0.5250$$

$$y_4 = \text{sigmoid}[x_1 \times w_{14} + x_2 \times w_{24} - \theta_4] = \frac{1}{1 + e^{-(1\times 0.9 + 1\times 1.0 + 1\times 0.1)}} = 0.8808$$

Not the actual output of neuron 5 in the output layer is determined as

$$y_5 = \text{sigmoid}[y_3 \times w_{35} + y_4 \times w_{45} - \theta_5] = \frac{1}{1 + e^{-(-0.5250\times 1.2 + 0.8808\times 1.1 - 1\times 0.3)}} = 0.5097$$

>> Evaluate the next weight training step. (where $\alpha = 0.1$)