1. Neural Network: Back Propagation Networks

1.1.Statement of the Problem

Write a computer program to train a multilayer feed forward back propagation artificial neural network (ANN).

1.2.Background

Step 1: Normalize the inputs and outputs with respect to their maximum values. It is proved that the neural networks work better if input and outputs lie between 0-1. For each training pair, assume there are 'I' inputs given by $\underset{I \times 1}{\{I\}_I}$ and 'n' outputs $\underset{n \times 1}{\{O\}_O}$ in a normalised form.

Step 2: Assume the number of neurons in the hidden layer to lie between $I < m < 2I$

Step 3: $[V]$ Represents the weights of synapses connecting input neurons and hidden neurons and $[W]$ represents weights of synapses connecting hidden neurons and output neurons. Initialize the weights to small random values usually from -1 to 1. For general problems, λ can be assumed as 1 and the threshold values can be taken as zero.

$$[V]^0 = [\text{random weights}]$$

$$[W]^0 = [\text{random weights}]$$

$$[\Delta V]^0 = [\Delta W]^0 = [0] \qquad (3.51)$$

Step 4: For the training data, present one set of inputs and outputs. Present the pattern to the input layer $\{I\}_I$ as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\underset{I \times 1}{\{O\}_I} = \underset{I \times 1}{\{I\}_I} \qquad (3.52)$$

Step 5: Compute the inputs to the hidden layer by multiplying corresponding weights of synapses as

$$\underset{m \times 1}{\{I\}_H} = \underset{m \times l}{[V]^T} \underset{l \times 1}{\{O\}_I} \tag{3.53}$$

Step 6: Let the hidden layer units evaluate the output using the sigmoidal function as

$$\{O\}_H = \begin{Bmatrix} \cdot \\ \cdot \\ \dfrac{1}{(1+e^{-I_{Hi}})} \\ \cdot \\ \cdot \end{Bmatrix}_{m \times 1} \tag{3.54}$$

Step 7: Compute the inputs to the output layer by multiplying corresponding weights of synapses as

$$\underset{n \times 1}{\{I\}_O} = \underset{n \times m}{[W]^T} \underset{m \times 1}{\{O\}_H} \tag{3.55}$$

Step 8: Let the output layer units evaluate the output using sigmoidal function as

$$\{O\}_O = \begin{Bmatrix} \cdot \\ \cdot \\ \dfrac{1}{\left(1+e^{-I_{oj}}\right)} \end{Bmatrix} \tag{3.56}$$

The above is the network output.

Step 9: Calculate the error and the difference between the network output and the desired output as for the ith training set as

$$E^p = \frac{\sqrt{\Sigma\,(T_j - O_{oj})^2}}{n} \tag{3.57}$$

Step 10 : Find $\{d\}$ as

$$\{d\} = \begin{Bmatrix} \vdots \\ (T_k - O_{ok})O_{ok}(1 - O_{ok}) \\ \vdots \end{Bmatrix}_{n \times 1} \tag{3.58}$$

Step 11 : Find $[Y]$ matrix as

$$\underset{m \times n}{[Y]} = \underset{m \times 1}{\{O\}_H} \underset{1 \times n}{\langle d \rangle} \tag{3.59}$$

Step 12 : Find $[\Delta W]^{t+1} = \alpha[\Delta W]^t + \eta[Y]$ (3.60)

$m \times n \quad m \times n \quad m \times n$

Step 13 : Find $\underset{m \times 1}{\{e\}} = \underset{m \times n}{[W]} \underset{n \times 1}{\{d\}}$ (3.61a)

$$\{d^*\} = \begin{Bmatrix} \vdots \\ e_i(O_{Hi})(1 - O_{Hi}) \\ \vdots \end{Bmatrix}_{m \times 1} \tag{3.61b}$$

Find $[X]$ matrix as

$$[X] = \{O\}_I \langle d^* \rangle = \{I\}_I \langle d^* \rangle \quad (3.62)$$
$$l \times m \quad l \times 1 \; 1 \times m \quad l \times 1 \; 1 \times m$$

Step 14 : Find $[\Delta V]^{t+1} = \alpha[\Delta V]^t + \eta[X]$ (3.63)
$l \times m \quad l \times m \quad l \times m$

Step 15 : Find

$$[V]^{t+1} = [V]^t + [\Delta V]^{t+1}$$
$$[W]^{t+1} = [W]^t + [\Delta W]^{t+1} \quad (3.64)$$

Step 16 : Find error rate as

$$\text{error rate} = \frac{\Sigma E_p}{\text{nset}} \quad (3.65)$$

Step 17 : Repeat steps 4-16 until the convergence in the error rate is less than the tolerance value.

End Algorithm BPN

1.3.Methodology

Consider that for a particular problem there are five training sets as shown in

Table 3.3 Training sets

S. no.	Inputs		Output
	I_1	I_2	O
1	0.4	–0.7	0.1
2	0.3	–0.5	0.05
3	0.6	0.1	0.3
4	0.2	0.4	0.25
5	0.1	–0.2	0.12

In this problem, there are two inputs and one output and already, the values lie between –1 to 1 and hence, there is no need to normalize the values. Assume two neurons in the hidden layer. The neural network architecture is shown in Fig. 3.16.

With the data of the first training set.

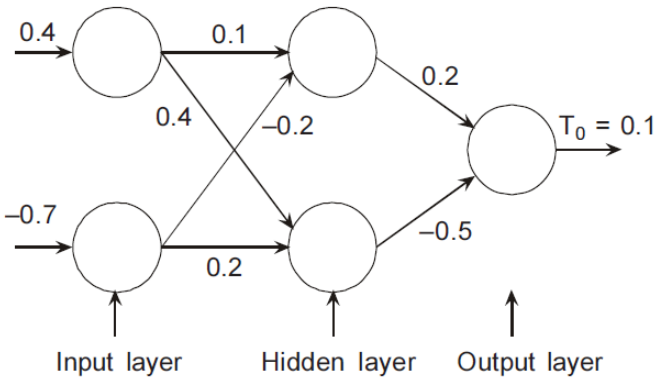

Fig. 3.16 MFNN architecture for the illustration.

Step 1: $\{O\}_I = \{I\}_I = \begin{Bmatrix} 0.4 \\ -0.7 \end{Bmatrix}$

Step 2: Initialize the weights as

$$[V]^0 = \begin{bmatrix} 0.1 & 0.4 \\ -0.2 & 0.2 \end{bmatrix}_{2\times 2}; \quad [W]^0 = \begin{Bmatrix} 0.2 \\ -0.5 \end{Bmatrix}_{2\times 1}$$

Step 3: Find $\{I\}_H = [V]^T \{O\}_I$ as

$$= \begin{bmatrix} 0.1 & -0.2 \\ 0.4 & 0.2 \end{bmatrix} \begin{Bmatrix} 0.4 \\ -0.7 \end{Bmatrix} = \begin{Bmatrix} 0.18 \\ 0.02 \end{Bmatrix}$$

Step 4: $\{O\}_H = \begin{Bmatrix} \dfrac{1}{1+e^{-0.18}} \\ \dfrac{1}{1+e^{-0.02}} \end{Bmatrix} = \begin{Bmatrix} 0.5448 \\ 0.505 \end{Bmatrix}$

Step 5: $\{I\}_O = [W]^T \{O\}_H = \langle 0.2 - 0.5 \rangle \begin{Bmatrix} 0.5448 \\ 0.505 \end{Bmatrix} = -0.14354$

Step 6: $\{O\}_O = \left(\dfrac{1}{1+e^{0.14354}} \right) = 0.4642$

Step 7: Error $= (T_O - O_O)^2 = (0.1 - 0.4642)^2 = 0.13264$

Step 8: Let us adjust the weights

First find $d = (T_O - O_{O1})\,(O_{O1})(1 - O_{O1})$

$$= (0.1 - 0.4642)\,(0.4642)\,(0.5358) = -0.09058$$

$$[Y] = \{O\}_H \langle d \rangle = \begin{Bmatrix} 0.5448 \\ 0.505 \end{Bmatrix} \langle -0.09058 \rangle = \begin{Bmatrix} -0.0493 \\ -0.0457 \end{Bmatrix}$$

Step 9:

$$[\Delta W]^1 = \alpha[\Delta W]^0 + \eta[Y] \quad (\text{assume } \eta = 0.6)$$

$$= \begin{Bmatrix} -0.02958 \\ -0.02742 \end{Bmatrix}$$

Step 10: $\{e\} = [W]\ \{d\} = \begin{Bmatrix} 0.2 \\ -0.5 \end{Bmatrix} (-0.09058) = \begin{Bmatrix} -0.018116 \\ 0.04529 \end{Bmatrix}$

Step 11: $\{d^*\} = \begin{Bmatrix} (-0.018116)\,(0.5448)\,(1 - 0.5448) \\ (0.04529)\,(0.505)\,(1 - 0.505) \end{Bmatrix} = \begin{Bmatrix} -0.00449 \\ 0.01132 \end{Bmatrix}$

Step 12: $[X] = \{O\}_I \langle d^* \rangle = \begin{Bmatrix} 0.4 \\ -0.7 \end{Bmatrix} \langle -0.00449 \quad 0.01132 \rangle$

$$= \begin{bmatrix} -0.001796 & 0.004528 \\ 0.003143 & -0.007924 \end{bmatrix}$$

Step 13: $[\Delta V]^1 = \alpha[\Delta V]^0 + \eta[X] = \begin{bmatrix} -0.001077 & 0.002716 \\ 0.001885 & -0.004754 \end{bmatrix}$

Step 14:
$$[V]^1 = \begin{bmatrix} 0.1 & 0.4 \\ -0.2 & 0.2 \end{bmatrix} + \begin{bmatrix} -0.001077 & 0.002716 \\ 0.001885 & -0.004754 \end{bmatrix} = \begin{bmatrix} 0.0989 & 0.04027 \\ -0.1981 & 0.19524 \end{bmatrix}$$

$$[W]^1 = \begin{Bmatrix} 0.2 \\ -0.5 \end{Bmatrix} + \begin{Bmatrix} -0.02958 \\ -0.02742 \end{Bmatrix} = \begin{Bmatrix} 0.17042 \\ -0.52742 \end{Bmatrix}$$

Step 15: With the updated weights $[V]$ and $[W]$, error is calculated again and next training set is taken and the error will be adjusted.

Step 16: Iterations are carried out till we get the error less than the tolerance.

Step 17: Once weights are adjusted the network is ready for inference.