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.

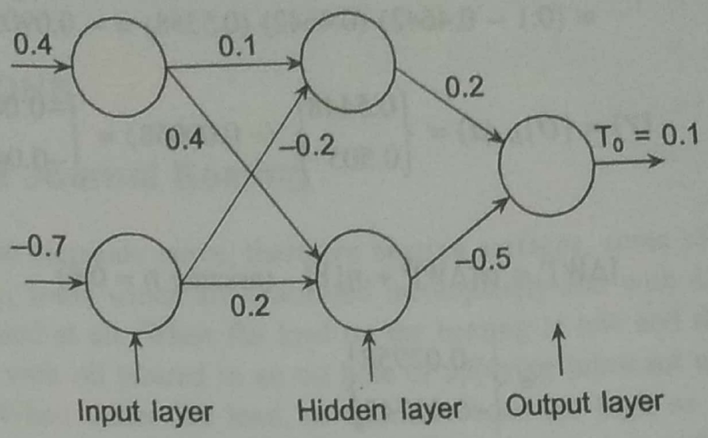


**Fig. 3.16** MFNN architecture for the illustration.

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.

A computer program "NEURONET" is developed for training the data and inferring the results using backpropagation neural network.

## 3.4 APPLICATIONS

### 3.4.1 Design of Journal Bearing

Whenever the machine elements move, there are bearing surfaces, some of which are lubricated easily and completely, some which are lubricated incompletely and with difficulty, and some of which are not lubricated at all. When the load on the bearing is low and the motion is slow, the bearing is lubricated with oil poured in an oil hole or applying lubricant with some other device from time to time. When either the load, or speed or both are high as in modern high-speed machinery, the lubrication by oil or by other fluids must be designed according to the conditions of operation. When there is a relative motion between two machine parts, one of which supporting the other, then the supporting member is called *bearing*. The bearings are classified as shown in Fig. 3.17(a).

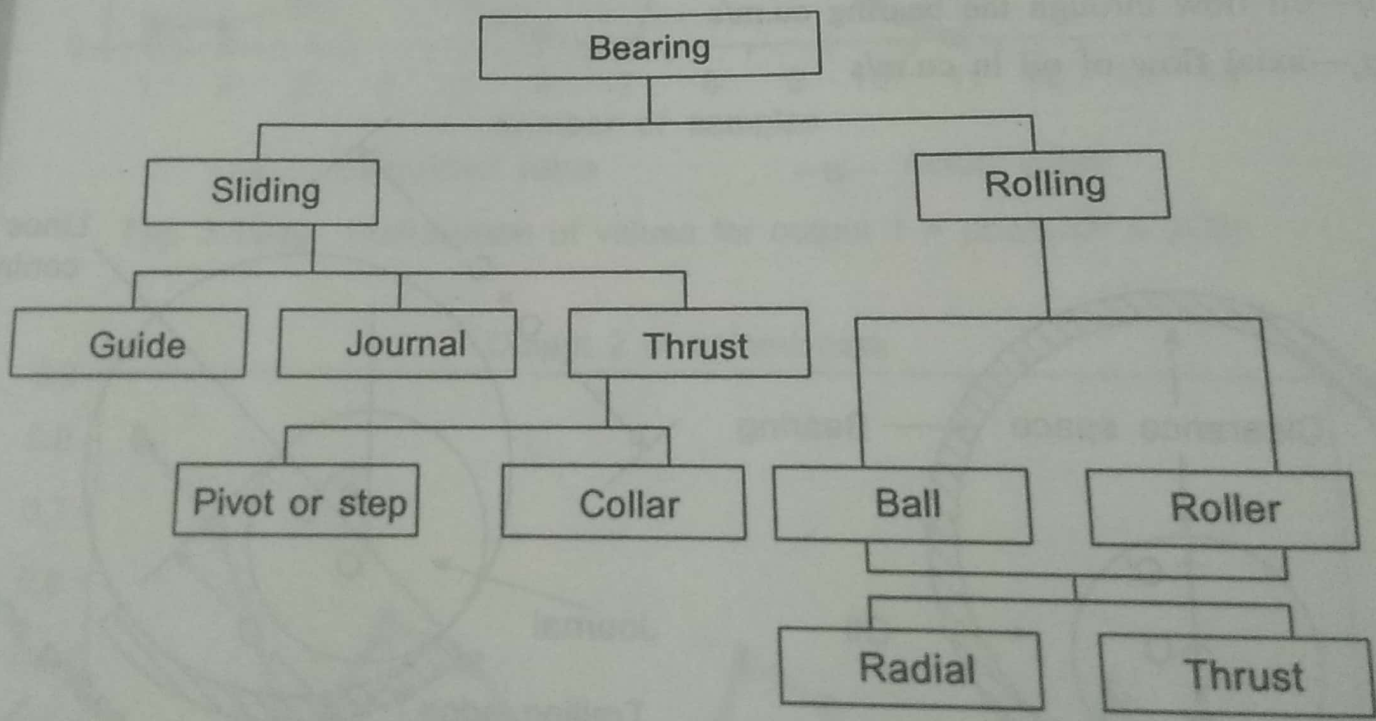


Fig. 3.17(a) Different types of bearings.