## ANN Approach

Hu (1964) was the first that investigated the weather forecasting using a neural network. A lot of studies came later to investigate using ANN to forecast the weather data generally and wind speed time series data in particular. The main aim of using ANN is not just the forecasting, but for enhancing the forecasting accuracy, especially for wind speed and other nonlinear time series data, by handling the nonlinearity problem. ANN can have single or multiple layers. The proposed ANN in this study was able to handle the nonlinearity in wind speed data and this enhances the forecasting accuracy. Basic type of neural network is MLP. MLP is a feedforward neural network with one or more hidden layers between input layer and output layer. Feed-forward is an algorithm to calculate the output from the input. Feedforward means that data are produced in one direction from input layer to output layer. This type of network is trained by using the backpropagation learning algorithm. The multilayer feed-forward back propagation (FFBP) can be regarded as the workhorse of the ANN (Hagan and Menhaj, 1994; Johansson *et al.*, 1991).

In this study, the FFBP is considered to contain two layers in addition to inputs layer. First one is the hidden layer, and the second one is the output layer. In the inputs layer, there are R inputs. These inputs are oftentimes weighted randomly. For each hidden layer there are also M neurons. Practically, the best number of neurons in the hidden layer is equaled to $R \times 2 + 1$ (Palit and Popovic, 2006; Sheela and Deepa, 2013). Each input variable Z is separately weighted by a random weight w automatically. The weighted inputs for R variables and M neurons are summed with the biased value b to formulate the input of the transfer function. The input variable SUM of the transfer function f can be formulated such as follows.

$$SUM = \sum_{i=1}^{M} \sum_{j=1}^{R} w_{i,j} Z_j + b.$$ (0.1)

The most used types of transfer functions for hidden and output layers are commonly tan-sigmoid that generates outputs between -1 and +1, log-sigmoid that generates outputs between 0 and 1, and linear transfer functions that generates outputs between -1 and +1. Selecting a transfer function for the hidden and output layers is important for obtaining good results and depends on the nature of the data. Figure 3.1 displays the different types of transfer function that can be specified using ANN toolbox in MATLAB.
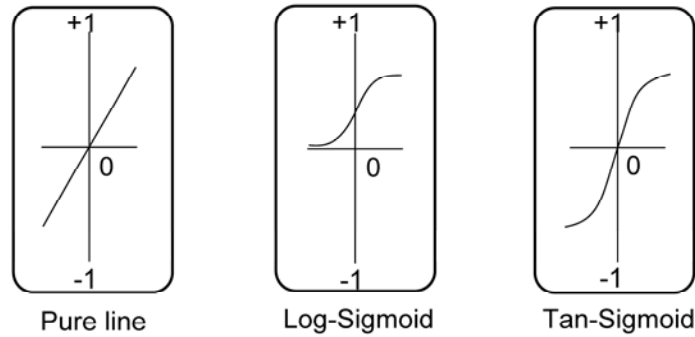
Figure 3.1 Transfer function types of ANN

The transfer function in the hidden layer is used to reflect the type of the relationship between the input and output variables, while the transfer function in the output layer is commonly employed to constrain the observation values of ANN's output variable. The mathematical expressions for linear, logistic sigmoid, and tangent sigmoid transfer functions, respectively, are as follows.
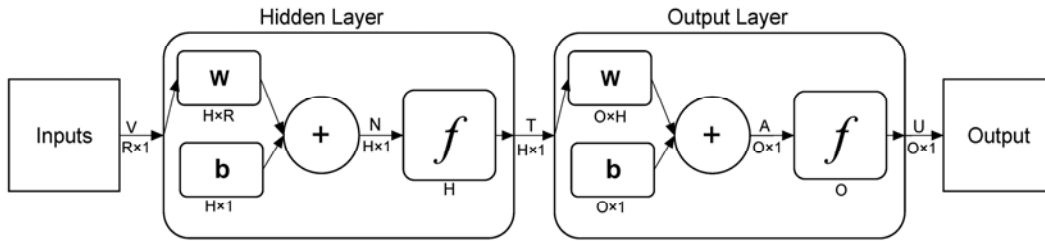
$$f(SUM) = SUM$$ (0.2)

$$f(SUM) = \frac{1}{1 + e^{-SUM}} \qquad\qquad (0.3)$$

and

$$f(SUM) = \frac{2}{1 + e^{-2SUM}} - 1 \qquad\qquad (0.4)$$

where SUM was defined in equation (0.1) (Dawson and Wilby, 2001; Shrestha *et al.*, 2005; Yonaba *et al.*, 2010).

Using ANN toolbox the FFBP neural network that includes one hidden layer and one output layer with single output can be represented such as in Figure 3.2 (Beale *et al.*, 2012; Palit and Popovic, 2006).

Figure 3.2: The general structural of ANN

The random weights $w_{i,j}$ for *R* inputs and H neurons can be written as a matrix such as follows.

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,1} & w_{M,2} & \cdots & w_{M,R} \end{bmatrix}, \tag{0.5}$$

while the inputs vector can be formulated as follows.

$$Z = \begin{bmatrix} Z_{t1} & Z_{t2} & \cdots & Z_{tR} \end{bmatrix}' \tag{0.6}$$

where $V = Z$ as an input vector for the hidden layer, $T = Z$ as an input vector for an output layer, $T = \hat{Z} = f(N)$ as an output vector for first hidden layer, $U = \hat{Z} = f(A)$ as an output vector for an output layer, $N = SUM$ as a summation function for first hidden layer, $A = SUM$ as a summation function for an output layer, H represents the number of the neurons in the first hidden layer and the number of inputs for an output layer, and O represents the number of the neurons in an output layer and equals to 1 oftentimes.

ANN training process is employed to adjust the random weights and bias values in each layer to enhance the forecasting accuracy and obtain desirable output. Training process may be supervised or unsupervised. FFBP neural network requires a supervised training for obtaining fitting and forecasting error. Many types of training algorithm can be used to apply the training process. The best training algorithm satisfies the most minimum forecasting error. Minimizing the forecasting error also depends on the appropriate choosing of the weights and

bias values in addition to the appropriate training function. The best training functions for back propagation algorithms are Levenberg-Marquardt (LM) and Bayesian regularization (BR) (Tien Bui *et al.*, 2012; Yonaba *et al.*, 2010).

LM training function is oftentimes the fastest training algorithm and recommended to minimise MSE of a neural network (He and Ma, 2010; Kermani *et al.*, 2005; Khan *et al.*, 2013; Mukherjee and Routroy, 2012; Piotrowski and Napiorkowski, 2011). BR training function determines the weights and bias values according to LM optimization. It minimizes the forecasting error by combining the classical error's sum of squares with an two further terms called regularization (Caballero and Fernández, 2008; Hirschen and Schäfer, 2006; Jiang *et al.*, 2008; Kumar *et al.*, 2004; Lau *et al.*, 2009; Saini, 2008; Ticknor, 2013).
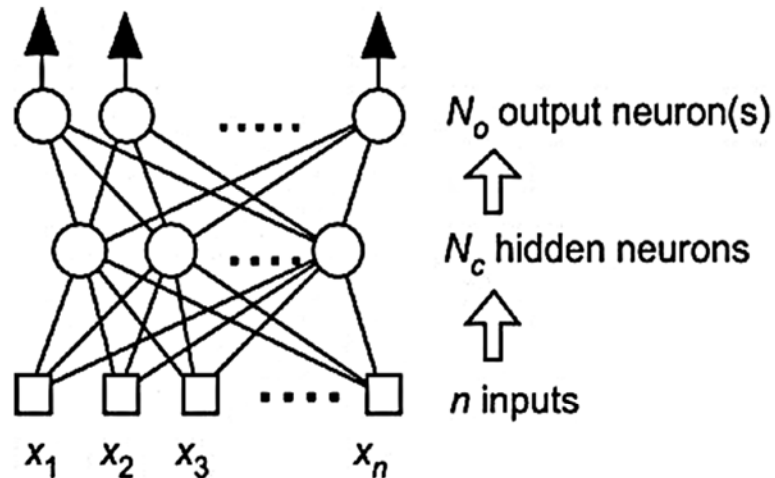
A neural network with n inputs, a layer of N_c hidden neurons, and N_o output neurons