Perceptron Neural Network (PNN)

The learning process using neural networks is a process that simulates the mechanism of neurons in the human body. The goal of training different types of neural networks is for classification and regression purposes, such as the convolutional neural network CNN, the recurrent neural network RNN, and the perceptron neural network PNN. These networks can be used to classify or predict all types of data, whether these data are images, numbers, texts, or videos. The perceptron neural network, as shown in Figure 1 below, is a single-layer network and is considered the basis for all networks. Understanding the operations of this network is a good basis for understanding other networks that are more complex. In order to create and train this network for classification purposes, the data must first be prepared by dividing it into training and test data sets, then training the network on the training data, and finally calculating the classification accuracy for the test data.

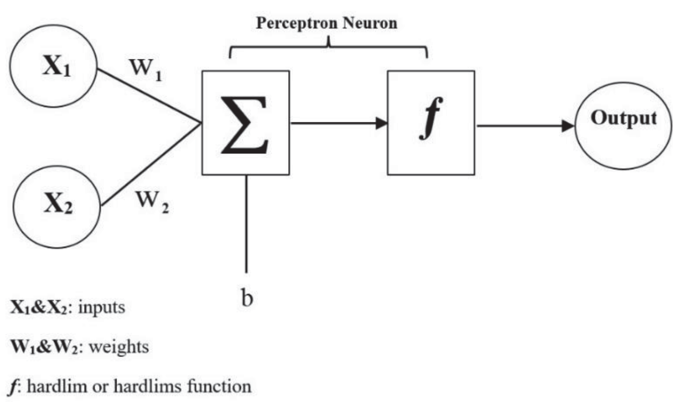

Figure 1: PNN

The perceptron neural network consists of an input layer in which the variables x_i are entered and a single neuron called the perceptron that receives the inputs by giving them an appropriate initial weight w_i so that all the inputs are multiplied by the default weights and added together and the bias value is added which can be represented by the additive function in equation (1) below.

$$SUM_q = \sum_{q=1}^{s} w_q x_q + b \qquad (1)$$

Where q represents the input variable, and s represents the number of variables or inputs.

Then one of the transfer functions of single-layer networks is applied to the additive function in equation (1).

The transfer functions of the PNN method are as follows.

1. The hard-limit transfer function is used in single-layer networks, including the perceptron neural network, and works to convert the additive function into binary outputs 0 or 1, as in Equation (2) and Figure 2.

$$f(\text{SUM}) = \begin{cases} 0 & \text{SUM} < 0 \\ 1 & \text{SUM} \geq 0 \end{cases} \qquad (2)$$

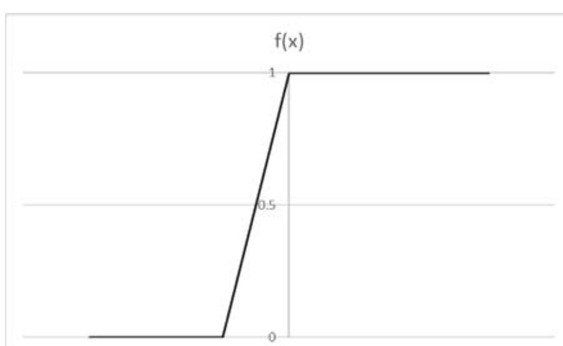

Figure 2: hardlim transfer function

2The symmetric hard-limit transfer function is used in the perceptron neural network to convert the additive function into outputs of -1 or 1 as in Equation (3) and Figure 3 below.

$$f(\text{SUM}) = \begin{cases} -1 & \text{SUM} < 0 \\ 1 & \text{SUM} \geq 0 \end{cases} \quad (3)$$

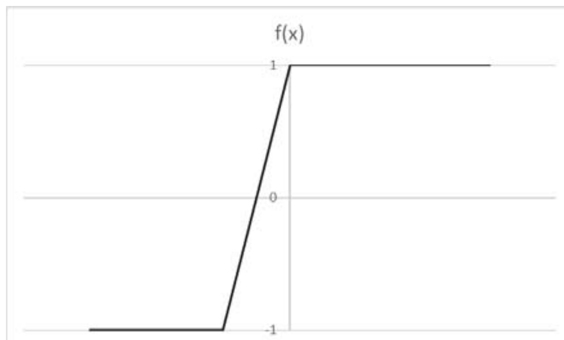

Figure 3: hardlims transfer function

The transformation function basically converts equation (1) into a binary output called the output layer. In this layer, the error value is calculated and then the error is reduced by updating the weight value (w) and bias value (b), which is an iterative process until the optimal values are reached. After obtaining the optimal values for the weights and bias through the classification layer, the final procedure performed by the PNN is to classify the time series by comparing the variable $\hat{y}_{final}$ with the original values variable, which is the target variable. The accuracy of the classification model will be calculated with respect to the actual values of the time series using the classification accuracy evaluation metric (Glorot & Bengio, 2010). Figure 4 below illustrates the general framework of the PNN algorithm.

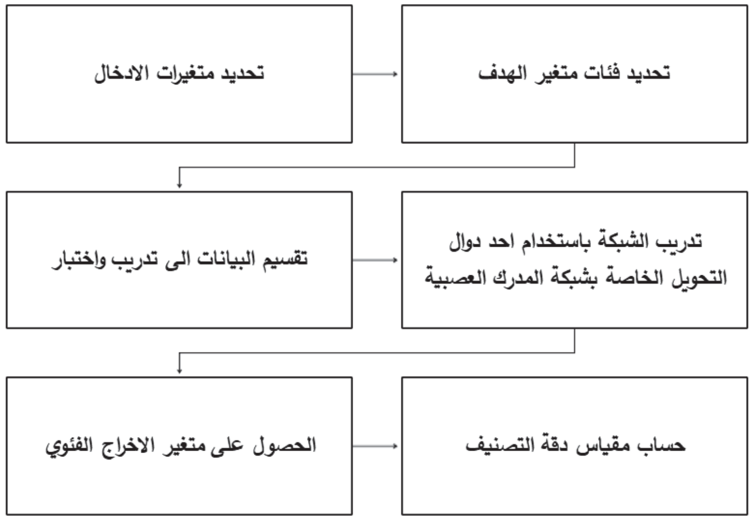

Figure 4: General structure of the perceptron neural network algorithm