CHAPTER 3

METHODOLOGY

3.0 Introduction

This chapter will discuss about the methods of collecting data, methods of analyzing data and the modelling to be done. This research will focus on using Artificial Neural Network to analyze data and come up with a model for the purpose of prediction. This chapter will serve an important role in implementing this research study accordingly inside the scope of study.

3.1 Artificial Neural Network

The data analysis tool to be used in this research is Feed-Foward Back Propagation Artificial Neural Network, FF-ANN. The feed forward network, which can be considered as common purpose non-linear functions for aligning two sets of variables, form the basis for most present day applications.

The coding tool to provide this network in this research is python programming tool. It is a widely used, general purpose high-level programming language. It can be used to program multiple models including object-oriented programming, imperative programming and functional programming.
The ability of a network to generalize and its complexity depends on the network topology. The network is designed by trial and error. A larger-than-necessary network is bound to over-fit the training data, while a smaller-than-necessary network will be hard to train.

3.1.1 Feed Forward Neural Network

The model of neural network that will be used in this research is the feed forward neural network that was popularized by David Rumelhart. The reason for this decision is because of the success shown by the network in construction forecasting model, and the simplicity in implementation.

The data will be served at the input neuron, where it will be multiplied by a certain weight which will be the strength of the respective signals, and then computed by an activation function. When it reaches a certain threshold, the output neuron will compute the data to give an output. There can be numerous hidden layers, but there can be only a single input and output layer in a single network.

**Figure 5 McCulloch-Pitts model of a single neuron that sums the inputs \( x_i \), then transforms the sums of inputs using a non-linear activation function to give a final output.**
The weight determines the strength of the input multiplied by it. It can be either in positive or negative value. The computation of the neuron differs according to the weight. The output can be adjusted to the value intended by adjusting the weights multiplying the inputs. However for inputs of hundreds and thousands, the weights are adjusted by using algorithms to obtain the desired output, which is called training or learning. In this research, the algorithm used is back-propagation algorithm, that will be discussed further in the next topic.

3.1.1.1 Non-linear mapping function

In this research, the feed forward network is regarded as a general purpose non-linear function for mapping two sets of variables. The networks can be better understood by learning the mapping structure of it. Figure 4 shows a non-linear function which takes \(n\) input variables \(x_1, x_2, x_3, \ldots, x_n\), and then maps them onto \(m\) output variables \(y_1, y_2, y_3, \ldots, y_m\). In neural networks, variables \(x\) are called inputs while variables \(y\) represent outputs.

![Figure 6 A schematic illustration showing a non-linear function mapping \(n\) \(x\) variables onto \(m\) \(y\) variables](image)

3.1.1.2 Analogies of ANN with polynomial curve fitting

When input and output variables are collected and respective vectors are formed, which are denoted as \(x = (x_1, x_2, x_3, \ldots, x_d)\) and \(y = (y_1, y_2, y_3, \ldots, y_d)\), the precise model that maps \(x\) onto \(y\) is determined by the architecture of the network and the values of weights multiplier, denoted as \(w_1, w_2, w_3, \ldots, w_d\). Thus the network mapping can be written in the form \(y = y(x; w)\), which describes \(y\) as a function of \(x\) that is multiplied by parameter \(w\).