Chapter 3

Artificial Neural Network for Time Series Forecasting

Neural networks have been motivated by the recognition that the brain computers are in an entirely different way from the conventional digital computer. A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for us. It resembles the brain in two respects [Hajek (2005)].

- Knowledge is acquired by the network through a learning process.
- Inter neuron connection strengths known as synaptic weights are used to store the knowledge.

Most of the times, financial time series is non-linear in nature. Forecasting models such as ARMA, ARIMA, ARCH and GARCH, therefore, can not predict the behaviour accurately. In such cases, ANN efficiently predicts the futuristic behaviour of time series data. Neural networks get trained with the help of a given training sample of input and output. The process of training the network is repeated for many samples in the set until the network reaches a steady state, when further significant changes will not be observed in the weights. Weights are adapted by the training itself. Neural networks have in-built ability to adjust their weights for finding out proper model.

Neural networks can be applied to all sorts of financial problems. They are not only useful be used for stock market predictions but also in forecasting of yield curves, exchange rates, bond rates etc. Especially for forecasting of financial time series, ANN is useful due to the following reasons:

1. Stock market data is highly complex and hard to model. It, therefore, requires a non linear model which can appropriately fit the data.

2. A large set of interacting input series is required to explain a specific stock.

In recent years, neural network applications in finance for Pattern Recognition, Classification and Time Series Forecasting have dramatically increased.
Neural networks are universal function approximators that can map any non-linear function. Collection of input and processing units are known as neuron or nodes. Neurons are connected with weights which, along with network architecture, store knowledge of the trained network. Neural network is similar to linear and non-linear least square regression and can also be viewed as alternative statistical approach for solving least square problems. Linear regression models work as feed forward neural network with no hidden layer and one output neuron with linear transfer function. Networks with hidden layer resembles non-linear regression model and weights represent regression curve parameter.

3.1 Designing a Neural Network

Following are the basic steps for designing a neural network.

3.1.1 Variable Selection

Designing of neural network depends on clear understanding of problem. Selecting important input variable is critical, frequency of data depends upon objectives of researcher.

3.1.2 Data Collection

Technical data is always available where as fundamental information is more difficult to obtain. All data should be checked for errors by examining day to day changes, ranges and logical consistency. For fundamental data we should consider following four issues.

1. Method of calculating fundamental indicator, should be consistent over time series.

2. Data should not retroactively revised.

3. Must be appropriately lagged.

4. Researcher should be confident that the source will continue to publish required fundamental information.

3.1.3 Data Preprocessing

Analysing data, highlighting important relationships, detection of trends are considered in data preprocessing. As we know that data representation is tough in designing a network, we cannot use raw data directly. Thus, raw data is scaled between upper and lower bounds of transfer function.

Preprocessing of data involves mainly trial and error methods. Most commonly used data transformation is first differencing and taking natural logarithm of the variables. Another transformation technique that can be used is ratio of input variables. Smoothing of input and output data by using simple or exponential moving average can also be used as transformation. Sampling and Filtering can also be done to create a more uniform distribution.

3.1.4 Training, Testing and Validation

We divide our time series data into three sets called Training, Testing and Validation sets. Training set is the major part of the sample data which we use for artificial neural network model development and to learn the pattern present in data. Testing set is about 30 percent of training set and, is used for evaluating the forecasting and generalization ability of the developed model. Validation is used to avoid over fitting problems. It also determines the stopping point of the training process.

3.2 Neural Network Paradigms

Building a neural network forecaster for a particular forecasting problem is a non trivial task. Critical decision is to determine appropriate architecture such as number of layers, number of nodes in each layer, number of arcs which interconnect with nodes and selection of activation function for hidden and output nodes.

Number of input nodes

Number of input nodes corresponds to the number of variables in input vector used to forecast future values. In case of time series data, number of input nodes correspond to number of lagged observations. There should be medium number of inputs. Too few or too many input nodes can affect either learning or prediction capability of network.

Number of hidden layer and nodes

It allows neural network to detect the feature to capture pattern in data. It is also used for performing complicated nonlinear mapping between input and output variable. Generally, we use only one hidden layer in forecasting problem. If number of hidden nodes are high with one hidden layer, then we can take two hidden layers with less number of hidden nodes. Maximum of the network problem can be solved within two hidden layers. Number of hidden layers and nodes are decided by experiment and by trial and error method.

Number of output nodes

There are two types of forecasting horizon- one step ahead (only one output node) and multi step ahead. Multistep forecast can be made in two ways. First one is iterative forecasting in which forecasted value iteratively used as input for next forecast. There is only one output node needed for iterative forecasting. Second one is direct method in which network has several output nodes to directly forecast each step in future.

Activation function

It determines input and output relationship between nodes. Any differentiable function can be used as activation function but generally bounded, monotonically increasing and differentiable functions are chosen. Logistic function is the most widely used activation function.

Training algorithm

Neural network training is an unconstrained non-linear minimization problem in which weights are being set iteratively to minimize sum of square error. Here,we are using back-propagation algorithm for the training of BSE data. The process is described in the steps below:

1. Load input and target data.
2. Normalize input and target data.
3. Assign number of hidden neurons in hidden layer.
4. Find the size of Input and Output Vectors.
5. Initialize the weight matrices with random weights.
6. Set the count to zero to know the number of iterations.
7. Train the neural network for error-value function.
8. If error-value is greater than threshold value, train the network again.
9. Change the weight matrix by adding error-value effect on them.

3.3 Forecasting of BSE Data

We have taken Bombay Stock Exchange (BSE) daily closing data from January 2012 to December 2013. We first find out return of the BSE close data. For the training we have taken returns of closing prices as input and next day return as target. We have used one hidden layer and two hidden nodes. If we increase number of hidden layers, complexity of algorithm will increase. We have used sigmoid logistic function as activation function. Plot for mean square error is given below. Finally, error has been minimized up to 1.75.

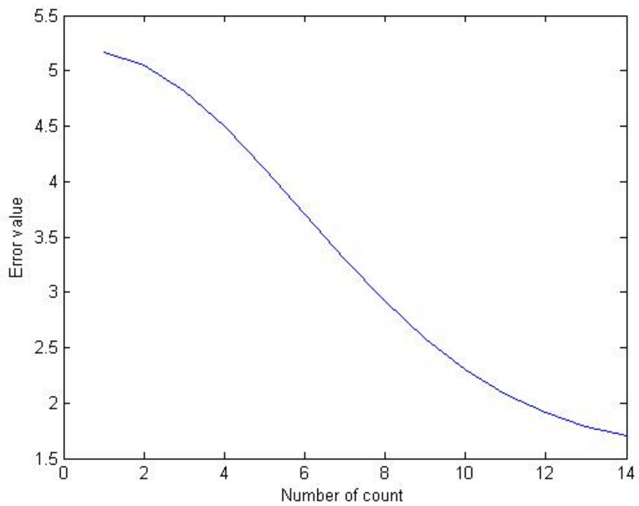

Figure 3.1: Mean squre error plot of BSE daily close data from Jan 2012 to Dec 2013.