EXCHANGE RATE PREDICTION USING A MULTI LAYER FEED-FORWARD ARTIFICIAL NEURAL NETWORKS

BY

Adu Sakyi

(BSc. Mathematics)

A Thesis Submitted to the Department of Mathematics, Kwame Nkrumah University of Science and Technology in partial fulfillment of the requirement for the degree of

MASTER OF PHILISOPHY APPLIED MATHEMATICS

DEPARTMENT OF MATHEMATICS

College of Science

AUGUST, 2013

DEDICATION

This work is dedicated to my dear mum Mrs. Vida Antwiwaa

DECLARATION

I hereby declare that this submission is my own work towards the MPhil and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

ADU SAKYI

PG 6202511

20251036 ________________ ______________

Student Name, Index & ID Signature Date

Certified by:

DR. LORD MENSAH ________________ ________________

Supervisor's Name Signature Date

DR. JOSEPH ACKORA PRAH ________________ ________________

Supervisor's Name Signature Date

PROF. S. K. AMPONSAH ________________ ________________

Head of Dept. Name Signature Date

Table of Contents

Abstract

This research aimed at forecasting the Ghanaian Cedi – US Dolar rate with Treasury bill rates, money supply, consumer price index and inflation

With the aim of exploring the efficiency of Artificial Neural Network (ANN) which is an imitation of the Human Brain, a two (2) layer feedforward network using Levenberg – Marquardt Backpropagation Algorithm was used for the forecast and the results were measured by the Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and the Weighted Absolute Percentage Error (WAPE).

After careful and extensive training, validation and testing, the ANN produced MSE, RMSE, WAPE and an R-value of 0.0010, 0.0324, 2.30%, and 0.99634 respectively.

An Artificial Neural Network model was obtained which was compared with the traditional multiple regression model with the ANN model producing a prediction accuracy of 97.70% as compared to 76.79% of the Regression model.

Acknowledgements

I thank the almighty God for His directions and protection over me. Special thanks also go to my Supervisor's Dr. Lord Mensah and Dr. Joseph Ackora Prah for their support throughout the entire research.

I also thank Rev. Dr. William Obeng Denteh for his support and the whole Mathematics Department Knust for their mentoring.

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter focuses on the background of the study, problem statement, objectives of the study, the methodology, the justification of the study and the organization of the study.

1.2 Background of the study

Few human activities have been so exhaustively studied during the past fifty years, from so many angles and by so many different sorts of people, as has the buying and selling corporation securities. Forecasting the exchange rates and determining the market timing, when to buy and import, are the two main problems which most researchers and practitioners face in order to get high financial gain.

The value of international currencies traded daily exceeds one trillion dollars. In comparison the value of goods and services traded is expected to be less than 5 percent of that amount, Froot and Thaler (1990).

Neural networks, recently, have been used in forecasting and trading stocks as well as exchange rates. An-Sing Chena, Mark T. Leung, and Hazem Daouk (2003), attempt to model and predict the direction of return on market index of the Taiwan Stock Exchange, one of the fastest growing financial exchanges in developing Asian countries. The probabilistic neural network (PNN) is used to forecast the direction of index return after it has been trained with historical data. The good performance of the PNN suggests that the neural network models are useful in predicting the direction of index returns. Other research shows the ability of neural networks to predict the price movements of a stock

1.3 Problem Statement

Consequently the relationship between an exchange rate and its explanatory variables continues to be a subject of extensive research. Most studies use a regression model to estimate the relationship between a bilateral exchange rate and variables such as differences in interest rates, money supply, industrial production, inflation, etc. The models generally assume a linear functional relationship in the original variables or transformed variables as in a logarithmic model.

Some researchers have found that a random walk model best explained the exchange rate behavior, Meese and Rogoff (1983). Others such as Boughton (1987) concluded that models based on economic factors were more successful in predicting exchange rates. Most of these models, however, perform poorly in predicting exchange rates in the short run. Since foreign exchange trade is often based on short run factors. It is important to explore other means for modeling and predicting exchange rates.

This research attempts to use Artificial Neural Network (ANN) to predict Exchange rates. Historical data of Exchange Rates, Treasury Bills, Money Supply, Consumer Price Index and Inflation were used to evaluate the performance of this neural network.

1.4 Objectives

The main objective of the study was to examine the strength of Artificial Neural Network in predicting the Ghana Cedi – US Dollar Exchange Rate. The specific objectives were to:

1. Ascertain how Artificial Neural Network functions and its application to financial market.
2. Investigate whether Artificial Neural Network (ANN) could be used to generate a forecasting model that would help in predicting Exchange Rates giving Treasury Bills, Money Supply, Consumer Price Index and Inflation.
3. Compare forecast accuracy of the ANN with the traditional Regression model.

1.5 Justification

The significance of this research is to aid in predicting Exchange Rates which is a key determinant of how the Ghana Cedis fell against foreign currencies. For instance, financial institutions and individuals who engage in imports and exports will be able to do effective trading if they are able to forecast the rate at which the Ghana Cedis will be faring in future times.

The research is expected to obtain fundamental as well as highly developed principles to promote the significance of having suitable application of neural network in financial management of the financial institutions in Ghana and the individuals to forecast and predict Exchange rate trends. This will mainly help companies, government and individuals to forecast the trends in Exchange rates in order to see how successful they are in sustaining interest value. It will in turn assist investors, importers and exporters as well as the individual to know the Exchange Rate trends and the best investment package to depend on. It will also aid the management of companies to monitor the performance of the Ghana Cedis in the foreign exchange market.

There are many traditional methods that can be used in the prediction and forecasting of data. However it is very clear that most of the traditional methods are inherently linear, also nonlinear ones that exist tend out to be very difficult to solve with minimal prediction accuracies.

The utilization of neural network models lies in the fact that they can be used to infer a function from observations. One of the first successful applications of ANNs in forecasting is reported by Lapedes and Farber (1987). Using two deterministic chaotic time series generated by the logistic map and the Glass Mackey equation, they designed the feed forward neural networks that can accurately mimic and predict such dynamic nonlinear systems. Their results show that ANNs can be used for modeling and forecasting nonlinear time series with very high accuracy. From the scope of applications, it is clear that considerable work has been done in the field of ANN deployment. Nevertheless, the application of neural network in the foreign exchange market to predict the strength of the Ghana Cedis cannot be overemphasized.

1.6 Methodology

Following the works of Vincenzo Pacelli et al (2010), exchange rates can be predicted using artificial intelligence. This research seeks to predict Exchange Rates using feedforward Artificial Neural Network on data set from January 1997 to December 2010.

1.7 Organization of the study

The study is organized into five chapters. Chapter one focuses on the introduction which includes background to the study, research problem, the objectives, justification and organization of the study. Chapter two is devoted to the review of relevant literature on the application of Artificial Neural Network (ANN) in the financial markets and exchange rate, a literature on the Exchange Rates, Money Supply, Inflation, Consumer Price Index, and Treasury Bills. Chapter three explains the methodology including model specification, data sources and training mechanisms. Chapter four discusses empirical results. The summary, conclusions, recommendations as well as the limitation of the study and direction for future research are presented in chapter five.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter focuses on reviewed papers on application of neural network in financial and foreign exchange market. It also includes a brief description of Exchange rates, Treasury Bills, Money Supply, Consumer Price Index and Inflation.

2.2 Introduction

One of the most important features of a neural network is its ability to adapt to new environments. Therefore, learning algorithms are critical to the study of neural networks. There are many different types of Neural Networks, each of which has different strengths particular to their applications. The abilities of different networks can be related to their structure, dynamics and learning methods. Neural Networks offer improved performance over conventional technologies in areas which includes: Machine Vision, Robust Pattern Detection, Signal Filtering, Virtual Reality, Data Segmentation, Data Compression, Data Mining, Text Mining, Artificial Life, Adaptive Control, Optimization and Scheduling, Complex Mapping and more.

It is natural and informative to judge forecasts by their accuracy. However, actual and forecasted values will differ, even for very good forecasts. To take an extreme example, consider a zero mean white noise process. The optimal linear forecast under quadratic loss is simply zero, so the paths of forecasts and realizations will look different. These differences illustrate the inherent limits to predictability, even when using optimal forecasts. The extent of series predictability depends on how much information the past conveys regarding future values of this series; as a result, some processes are inherently easy to forecast, and others are difficult. In addition to being of interest to forecasters, predictability measures are potentially useful in empirical macroeconomics. Predictability provides a succinct measure of a key aspect of time series dynamics and is therefore useful for summarizing and comparing the behavior of economic series, as well as for assessing agreement between economic models and data.

Diebold and Kilian (2000), do not advocate comparing models to data *purely* on the basis of predictability, rather predictability simply provides an easily-digested summary distillation of certain important aspects of dynamics. Remarkably little attention has been paid to methods for measuring predictability.

Existing methods include those based on canonical correlations between past and future, and those based on comparing the innovation variance and unconditional variance of stationary series. Those methods, however, are inadequate in light of recent work stressing the possible presence of unit roots, rich and high-dimensional information sets non-quadratic and possibly even asymmetric loss functions and variations in forecast accuracy across horizons. It has been realized that the major forecasting methods used in the financial area are either technical or fundamental. Due to the fact that Exchange Rates are affected by many highly interrelated economic, political and even psychological factors, interactions among these indicators become very complex.

Therefore, it is generally very difficult to forecast the trend in the foreign exchange. Classical statistical techniques for forecasting reach their limitation in applications with nonlinearities in the data set. There are quite a number of applications of neural network and this thesis attempts to give references.

2.2.1 Foreign exchange background

The most basic needs in life involve cash. Whenever someone goes to the supermarket, puts gas on his car, or goes shopping, that person will get in some sort of transaction in which money will be present. It matters a lot when a person's budget is limited because of a particular currency weakness. A weak currency can make the purchasing power of GHC500 be worth 250 dollars in America. It will matter the more when that individual decides to go on travel to a country where his spending ability will be constrained because of an unfavorable exchange rate. Imagine a U.S. traveller who decides to exchange his entire travel budget at his European country of destination. He believes it will be fine to do it since the euro is used in that country and its value (euro) will be stable for the duration of his trip. Taking this for granted could be seen as acting unwisely and naively. Exchanging your travel money at the last moment could be deemed as unsafe because your exchange options will be reduced considerably. While your money is exchanged, the advantageous or disadvantageous exchange rate fluctuations that have happened since you left your home country are not taken into account. These fluctuations are mostly discarded for exchange rate purposes for the simple reason that banks post a rate on a daily basis not on an hourly basis. This means that the daily rate posted is not a real rate and this may reduce your travel money if the rate has become more beneficial for you. The best advice for a U.S. traveller going overseas would be to plan his travel well in advance, do a little bit of research about the exchange rates on the country of his destination and exchange small amounts of money at different times over the entire planning process.

Monitoring exchange rates is of strategic importance not only for worldwide travellers, but also for export/import merchants. For instance, a Ghanaian merchant whose Ghanaian currency is strengthening as the days pass by will see how his products will be less competitive abroad. A strong Ghanaian currency will decrease the demand for Ghanaian products overseas. But at the same time, imported products from abroad will be cheaper, and Ghanaian local products will lose competitiveness in the domestic market to them. This situation will boost imports at the expense of exports. The other way around is the result of a weak Ghanaian currency. A weak Ghana Cedis will decrease the cost of and increase the demand for Ghanaian products overseas. Imported products will be more expensive because Ghanaian importers will have to pay more for a unit of foreign currency. This will cause Ghanaian demand for overseas products to decrease. Imported products will lose competitiveness to locally produced Ghanaian commodities. This situation will cause exports to soar and imports to shrink. Because of these and many other situations, such as a Ghanaian Ph.D. candidate financing his studies at a renowned business school in Europe through Ghana Cedis student loans, it is good to have a clear understanding of how the exchange rate mechanism works, and the precautions needed to avoid a lose-lose situation in the foreign exchange market. If we want to have a comprehensible picture of the exchange rate market, we need to understand the forces that cause these fluctuations and the mechanisms available to counter them.

2.2.2 Economic implications of exchange rates

Exchange rates serve a variety of purposes in the global business world. For instance, by helping in the translation and conversion of foreign currency, exchange rates expedite global commerce, the flow of products and services internationally. They also serve as economic indicators. For instance, a strong exchange rate indicates a growing economy and political stability for a particular country. On the contrary, a weak exchange rate may indicate economic recession. Foreign exchange reserves can be used by politicians to exert an influence on currency rates and manage the economy. This happens when government officials use foreign exchange reserves to repurchase domestic currency and strengthen its value. Exchange rates can also be used as strategic instruments. This can be seen when dealing with currency derivatives such as options and futures contracts. Futures and forward markets are the commonplace setting where transactions with futures, forwards and options take place. In these markets, exchange rates are set for specified periods of time in order to manage risks. These unwanted risks can sometimes create financial havoc for currency traders in foreign markets.

2.2.3 Factors influencing exchange rates

Exchange rates are influenced by a variety of factors some of which are difficult to control and manage, Lowery (2008). As for example, the demand and supply for money. Since exchange rates are based on the relative price of two national currencies, they are determined by the relative supplies and demands for these currencies. These relative supplies and demands for money are in turn determined by the plans of private businesses and government policies. These policies have a direct impact on the economy of a nation. Other economic factors which exert an influence on exchange rates are productivity, equity flow, hedging activities, interest rate differentials, inflation, growth domestic product (GDP), current account balances (CAB) and trans-oceanic economic policies. An interest rate differential usually reflects exchange rate expectations, Khor and Rojas-Suarez, (1991). The widely known purchasing power parity theory (PPP) links exchange rates to inflation.

Surprisingly, in the European Union there is another variable which measures GDP growth on an aggregate level. This variable, which is called the Economic Sentiment Indicator (ESI), is part of the joint harmonized EU programme of business and consumer surveys , European Commission DG ECFIN (2009). Business and consumer surveys reveal the opinion and expectations of financial and economic experts in the area about the current trend of the different sectors of the economy: industry, services, construction, retail trade and consumers. These business and consumer surveys provide important quantitative and qualitative information about the economic health, short term forecasts and economic research for the euro-area. ESI, which is mostly of a qualitative nature, reflects the monthly judgments, anticipations, perceptions and expectations concerning diverse facets of economic activity in the different sectors of the economy. Mehrotra and Rautava (2007) claim that business sentiment indicators are useful in forecasting developments in the Chinese economy because they tend to transmit useful information about the current and future state of affairs of economic activity in the country. Nilsson and Tapasanun (2009) support the idea that sentiment indicators play a big role in the foreign exchange market. They claim that the EUR/USD exchange rate rose as euro zone economic sentiment indicator increased for the first time since May 2007. As risk sentiment improved, the US dollar and the yen fell against their opposites. The sterling rose on expectations of a UK economic recovery by the end of the year. Likewise, positive feelings for a global economic recovery and higher commodity prices made the Canadian and Australian dollars rise again. Rodriguez (2009) states that several sentiment indicators have a significant role in the US dollar/Euro exchange rate. On a theoretical level, financial relationships have been suggested between foreign exchange markets and sentiment indicators. Deans (2010) and Lawrence (2011) state that it is the traders' expectations in the foreign exchange market which cause the constant buying and selling of currencies. Some of these expectations are in turn related to political and psychological factors. Political factors may include the monetary policy in place, government intervention or manipulation, a country's relative economic exposure and the political stability of a country. There is an implicit assumption among traders which states that even psychological factors exert an influence on market participants engaged in exchange rate transactions. Psychological factors could include risk avoidance, market anticipation, greed, speculative pressures and future expectations, Hill (2004). Due to the vast array of external and internal forces encompassing foreign exchange markets, the possibility of making erroneous exchange rate forecasts is more possible than ever. Traders often try to predict exchange rates with a high degree of reliability, but most of the times those forecasts are far from being true. A high-quality forecast would allow traders and market participants to make more-informed decisions, Wang (2008). Regrettably, in the financial world this is not the case. The next section will explain the difficulty in forecasting exchange rates.

2.2.4 **Complexity of exchange rates**

Exchange rates are highly volatile which makes them very difficult to predict over short to medium time horizons, Cheong et al (2011). Because of this volatility, an estimation of the future value assets and liabilities denominated in foreign currency is very hard to predict. Some researchers have proposed the idea that exchange rates and financial assets behave alike. This means that the price movements of a financial asset are determined by changes in expectations about future economic variables, rather than by changes in current ones, Wang (2008). What this implies is that the real contribution of average exchange rate models is not their capacity to forecast currency values, but their ability to forecast economic fundamentals such as trade balances, money supply, national income and other key variables. Other researchers challenge the idea that economic fundamentals help predict currency values and attribute changes in exchange rates to random luck , Meese and Rogoff (1983). They claim that the performance of a random walk model is just as good as the performance of a model based on economic fundamental variables.

Another reason why economic models find it so difficult to predict exchange rates is that there is no apparent connection between exchange rates and economic fundamentals, Wang (2008). This could be attributed to the intrinsic limitations of the economic models themselves or to a model's misspecifications. The coefficients of an economic model try to specify the relationship between exchange rates and its fundamentals. However, these parameters are estimated based on historical data while the predictive power of these models is based on their ability to forecast currency values from new data. Meese and Rogoff (1983) claim these parameters may be different over time. This doesn't mean that economic fundamentals need to be discarded as predictors of exchange rates, but rather that other combinations of economic fundamentals and forecasting methods (econometric and time series models) have to be taken into account when predicting exchange rates.

2.2.5 **Modeling Exchange Rates**

There are two well-known approaches to forecast foreign exchange rates. The first one is known as the fundamental approach and the second one is known as the technical approach, Tsay (2005). Table 2.1 on the next page shows the main principles of each approach.

The Fundamental Analysis	The Technical Analysis
Belief in economics	Broad economic data only
Macro and micro economics	Price charts hold all the information
Analysis of financial reports	Analysis of price patterns
Belief in management accounts	Interpretation of indicators

Table 2 .1: Comparison between fundamental and technical analysis

The fundamental approach is based on fundamental economic variables which influence currency predictions. The following fundamental economic variables, most of which are taken from economic models, are of importance in foreign exchange markets: trade balance, inflation rates, unemployment, interest rates, GNP, productivity index and consumption, among others Johnston and Scott (1997). Structural (equilibrium) models dominate fundamental forecasts. These models are then modified to incorporate the statistical characteristics of the data, and the knowledge and skill of the forecasters. The structural models are used to equilibrate exchange rates. Once the exchange rates are equilibrated, projections or trading signals can be produced. Bjornland and Hungnes (2006) did a comparative study of the forecasting performance of a structural exchange rate model which unites the PPP condition with the interest rate differential in the long run against some alternative exchange rate models. They are able to show that the interest rate differential is very important in the long run when predicting exchange rate behavior. A forecasting model with its respective forecasting equation is the first product of the fundamental approach, Tsay (2005). This model is built on a single theory, for example the PPP, a combination of theories or on the specialized knowledge and skills of a forecaster. The very first thing this forecaster does is to collect the data needed to estimate the forecasting equation. This forecasting equation is then evaluated using statistical techniques or some other measures. Once the model has been statistically approved, the next step will be the generation of forecasts. These forecasts will be finally evaluated to determine the reliability and accuracy of the model. One important thing to mention is that once the financial data is collected, the forecaster will divide it in 2 sets. The first set is commonly known as the training or estimation period. The second set is known as the validation period. The training period, as its name implies, is used for training the model and selecting its parameters. The validation period is used to test your model for future generation of forecasts. At this last stage, the forecaster decides if the model's forecasting performance is deemed acceptable for out-of-sample forecasting. Out-of-sample forecasting means using today's information to forecast the future path of exchange rates. On the other hand, in-sample forecasting means using today's information to forecast today's spot rate. The fitted values estimated in a model are all in-sample forecasts. Going back to the second approach in exchange rate forecasting, the technical approach is mostly built on price information, Tsay (2005). The word 'technical' means that the financial reasoning employed in the model does not make use of fundamental economic variables, but of extrapolations of past price trends. A technical trader looks for repetitive price patterns, trends or turning points. Buy or sell signals are produced from these turning points. The most often used technical models are based on filters, momentum indicators or moving averages. A filter model produces by signals when an exchange rate rises a specified number of percentage points above its most recent trough, and sell signals when an exchange rate falls a specified number of percentage points below the previous peak. Momentum models measure the healthiness of an asset by looking at the change in velocity with which the asset's price rises or falls. If the price rises very quickly, a buy signal is produced. A moving average model is a specified average of past prices. The trader specifies the number of days he wants the moving average to be. Moving averages smooth-out the noise and price variations of financial data, thus allowing traders to identify trends present in the data and make financial gains out of them. Moving averages are of importance in foreign exchange markets, Rosenberg (2003). Neely and Weller (2011) maintains that more complex forms of moving averages, which are responsible for excess returns, are more present than ever. Harris and Yilmaz (2008) analyze moving average rules and momentum trading strategies in order to determine which one produces higher returns and higher Sharpe ratios for exchange rates. Lento (2008) examines the ability of moving averages to forecast security returns of exchange rates. The results are quite mixed suggesting that moving averages are able to outperform the random walk model at some instances only and that moving averages provide valuable information for predicitng the holding period returns ten days after a buy or sell signal is generated. Neely, Weller and Dittmar (1997) show that technical trading rules provide excess returns for six exchange rates studied, and that technical trading rules are able to identify patterns in the data which are not detected by standard statistical techniques. For their ability in detecting price trends in foreign exchange data, this thesis makes use of moving averages as inputs to the forecasting technique selected.

2.3 Reviewed Papers on Application of Neural Network in Financial Market

There has been quite a number of works done on artificial neural network with its application to financial market as well as foreign exchange market and some are presented in subsequent sessions.

2.3.1 Forecasting the 30-year U.S. Treasury Bond with a System of Neural Networks"

Wei Cheng, et al (1996) paper discussed a forecasting model based on a system of artificial neural networks (ANN) which was used to predict the direction of the 30-Year U.S. Treasury Bond on a weekly basis. At the close of Friday's market, the 23-variable database is updated with the latest information and the data pre-processed for input to the prediction system. Thirty-two feed-forward neural networks are trained on the new data and then individually recalled to predict the following Friday's market direction. These results are then used as input to a decision model that ultimately determines the final prediction. Forecasting began in 1989, with live trades commencing in 1990. The average accuracy of the buy prediction was 67% over a five-year period, with an average annualized return on investment (ROI) of 17.3%. This compares with an ROI of 13.9% for the Lehman Brothers T-Bond Index. Their paper described the methods used for data selection, training and testing, the basic system architecture, and how the decision model improved the total system accuracy as compared to individual networks

2.3.2 Time Series Prediction and Neural Networks

Frank R. J., *et al* (2000) began by discussing Neural Network approaches to time series prediction, and the need to find the appropriate sample rate and then identified an appropriately sized input window. They proposed that, work in neural networks on forecasting future developments of the time series is from past values '*x*' up to the current time. Formally this could be stated as: find a function $f{:}R^N$ ----R such as to obtain an estimate of *x* at time t + d, from the N time steps back from time t, so that:

$$x(t+d) = f(x(t), x(t\ 1), K, x(t\ N\ 1)) \qquad 2.1$$

$$x(t+d) = f(\mathbf{y}(t)) \qquad 2.2$$

where **y**(t) is the N- array vector of lagged x values

Normally d will be one, so that f will be forecasting the next value of x.

They trained a feedforward neural network with 120 hidden units, using conjugate gradient error minimization. The embedding dimension, the size of the input layer, is increased from 1 unit to 9 units. The data is split into a training set of 1200 vectors and test set of 715 vectors. Each network configuration is trained 10 times with different random starting points. It can be inferred from Frank R. J., *et al* (2001) findings that the results they obtained by testing the model with the data set suggested that the embedding theorem and the false nearest neighbour method can provide useful heuristics for use in the design of neural networks for time series prediction. With two of the data sets they examined here, the predicted embedding size corresponded with a network configuration that performed well, with economical resources. On the other hand, in their research and with reference to their data sets they did not observe an overlarge embedding size to have a deleterious effect on the network. The tree ring data, however, showed that conclusions must be treated with caution, since poor predictive results were produced whatever the window size.

2.3.3 Traffic Trends Analysis using Neural Networks

According to Edwards T., *et al* (1997), one aspect of time series analysis involves forecasting the value of a variable from a series of observations of that variable up to a particular time. They suppose that observations are available at discrete, equispaced historical time intervals, z[T], z[T-t], z[T-2t], z[T-3t]…, with time interval t . The observations {z(T), ... z[t - (N-1)t]} constitute a "window" and N is referred to as the window size. The aim is to forecast the value of z at some later time, z_f(T+L), where L is the lead-time and is an integral multiple of t . More formally the objective is to obtain a forecast function $z_f(T+L)$ which minimizes the mean square of the deviations $z_f(T+L) - z(T+L)$ for each lead time L.

They proposed that the simplest way to estimate z using a neural net is to use a feedforward net with one input for each member of the window and one output for $z_f(N+ t)$. An obvious extension of this model is to have multiple outputs corresponding to multiple lead times. These models are often called sliding window nets. As such, a more sophisticated predictor may be produced by buffering either the hidden units or the output units and recurrently adding these activations to the input vector. These predictors are particularly useful when the data is inherently noisy. In the work reported here they used a single hidden layer feedforward network with a sliding window input, trained with a scaled conjugate gradient algorithm. They concluded that, Neural Networks provide a useful tool for time series prediction in the telecoms domain. Critical to the performance of the predictor is the selection of an appropriate window size for the data which needed to be modeled. The nearest neighbor algorithm, which they described, has been tested empirically and shown to be a valuable technique for allowing definition of this window size from analysis of the data.

2.3.4 Neural Networks, Financial Trading and the Efficient Markets Hypothesis

Skabar and Cloete(2002) here described a methodology by which neural networks can be trained indirectly, using a genetic algorithm based weight optimization procedure, to determine buy and sell points for financial commodities traded on a stock exchange. In order to test the significance of the returns achieved using this methodology, the returns on four financial price series were compared with returns achieved on random walk data derived from each of these series using a bootstrapping procedure. These bootstrapped samples contain the same distribution of daily returns as the original series, but lack any serial dependence which is present in the original. The results indicated that on the Dow Jones Industrial Average Index, the return achieved over a four year out of sample period were significantly greater than that which would be expected had the price series been random. This lends support to the claim that some financial time series are not entirely random, and that – contrary to the predictions of the efficient markets hypothesis – a trading strategy based solely on historical price data can be used to achieve returns better than those achieved using a buy-and-hold strategy.

2.3.5 An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks

Walczak (1998) makes it clear that evidence has been presented that contradicts the current financial neural network development heuristic, which implies that greater quantities of training data is necessary to produce better-quality forecasting models. A new time series model, termed the Time-Series Recency Effect, has been demonstrated to work consistently across neural network models for six different currency exchange time series. The Time-Series Recency Effect claims that model building data that is nearer in time to the out-of-sample values to be forecast produces more accurate forecasting models.

The empirical results discussed in this article show that frequently a smaller quantity of training data will produce a better-performing back propagation neural network model of a financial time series. Other problematic issues related to the development of the best possible neural network model such as selection of input variables or selection of the neural network training algorithm and the design of the neural network architecture are not discussed in the literature. The prudent financial time series neural network developer will realize that these other factors will affect the neural network model's performance and should utilize existing guidelines to solve these issues. He then pointed out that for financial time series two years of training data is frequently all that is required to produce optimal forecasting accuracy. He further proposed that future research can continue to provide evidence for the Time-Series Recency Effect by examining the effect of training set size for additional financial time series (e.g., any other stock or commodity and any other index value). The Time-Series Recency Effect may not be limited only to financial time series, and evidence from nonfinancial time series domain neural network implementations already indicates that smaller quantities of more recent modeling data are capable of producing high-performance forecasting models. Additionally, the Time-Series Recency Effect has been demonstrated with neural network models trained using back propagation. In summary, it has been noted that neural network systems incur costs from training data. This cost is not only financial, but also impacts the development time and effort. Empirical evidence demonstrates that frequently only one or two years of training data will produce the "best" performing back propagation trained neural network forecasting models. The proposed methodology for identifying the minimum necessary training set size for optimal performance enables neural network researchers and implementers to develop the highest-quality financial time series forecasting models in the shortest amount of time and at the lowest cost.

2.3.6 Neural Network and Equity Forecasting

Jingao and Hean-Lee (1995) carried out a research on the performance of several back propagation neural networks applied to the prediction of Kuala Lumpur Stock Exchange Composite Index (KLSE) on stock market index. The delayed index levels and some technical indicators were used as the inputs of neural networks, while the current index level was used as output. With the prediction, significant paper profits were obtained for a chosen testing period of 304 trading days in 1990/91. Besides, the experiments showed that useful prediction could be made without the use of extensive market data or knowledge. They pointed out four challenges beyond the choice of either technical or fundamental data for using neural network to forecast the stock prices. First, the inputs and outputs of the neural networks have to be determined and preprocessed.

Second, the types of neural networks and the activation functions for each node have to be chosen. Third, the neural network architecture based on the experiment with different models has to be determined. Finally, different measures to evaluate the quality of trained neural networks for forecasting have to be experimented with.

2.3.7 Forecasting Foreign Exchange Rates Using Recurrent Neural Networks

In order to forecast foreign exchange rates by Tenti (1996), he proposed the use of recurrent neural network in which activity patterns pass through the network more than once before generating an output. This can learn extremely complex temporal sequences. They made a comparison of three architectures in terms of prediction accuracy of futures forecast for Deutche mark currency. They then devised and optimized a trading currency by taking into account transaction costs, shown for the different architectures. With respect to the derivation of the model for recurrent neural network, the use of an exponential trace memory which acts on the series of input $x(1)$, …, $x(t)$, creating a state representation as $[x_1(t), x_2(t), . . ., x_i(t)]$. Their model equation for the forecast value $x_i(t)$ is given as $x_i(t)=(1-b_i)x_1(t)+ b_i x_1(t-1)$.

This model is then implemented by Tenti with three different recurrent neural network architecture of which the first architecture has one recurrent neurode, the second with two recurrent neurode and the third with three recurrent neurrode. This therefore, allow for incremental calculations of the forecast value.

The forecast formulated by the three versions of the recurrent neural networks was just the initial part of the trading strategy according to Tenti. He further transformed the predictions into market actions obtained by specifying a set of rules to buy and sell currency futures.

Although, it is quite significant that Recurrent Neural Networks can yield good results due to the rough repetition of similar patterns present in foreign exchange and other time series forecasting. It is critical to mention that Recurrent Neural Networks require substantially more connections, and more memory in simulation, than standard neural networks. These subtle sequences cannot provide beneficial forecastability.

2.3.8 Artificial Neural network model for forecasting foreign exchange rate

Adewole, Akinwale and Akintomide (2011) developed an Artificial Neural Network Foreign Exchange Rate Forecasting Model (AFERFM) for foreign exchange rate forecasting to correct the ineffectiveness of some models to handle uncertainty and instability nature of foreign exchange data. The design was divided into two phases, namely: training and forecasting. In the training phase, back propagation algorithm was used to train the foreign exchange rates and learn how to approximate input. Sigmoid Activation Function (SAF) was used to transform the input into a standard range [0, 1]. The learning weights were randomly assigned in the range [-0.1, 0.1] to obtain the output consistent with the training. SAF was depicted using a hyperbolic tangent in order to increase the learning rate and make learning efficient. Feed forward Network was used to improve the efficiency of the back propagation. Multilayer Perceptron Network was designed for forecasting. The design was implemented using matlab7.6 and visual studio because of their supports for implementing forecasting system. The system was tested using mean square error and standard deviation with learning rate of 0.10, an input layer, 3 hidden layers and an output layer. According to them, the best known related work, Hidden Markov Foreign Exchange Rate Forecasting Model (HFERFM) showed an accuracy of 69.9% as against 81.2% accuracy of AFERFM. This shows that the new approach provided an improved technique for carrying out foreign exchange rate forecasting.

2.4 The Foreign Exchange

This entry provides the official value of a country's monetary unit at a given date or over a given period of time, as expressed in units of local currency per US dollar and as determined by international market forces or official fiat.

Exchange rate prices are expressed in various ways:

- **Spot Exchange Rate -** the spot rate is the actual exchange rate for a currency at current market prices. This is determined by the FOREX market on a minute-by-minute basis on the basis of the flow of supply and demand for any one particular currency.
- **Forward Exchange Rate -** a forward rate involves the delivery of currency at some time in the future at an agreed rate. Companies wanting to reduce the risk of exchange rate uncertainty by buying their currency ‘forward’ on the market often use this.

- **Bi-lateral Exchange Rate** - this is simply the rate at which one currency can be traded against another. Examples include:

    - $/DM, Sterling/US Dollar, $/YEN or Sterling/Euro

- **Effective Exchange Rate Index (EER)** - the EER is a weighted index of sterling's value against a basket of international currencies the weights used are determined by the proportion of trade between the UK and each country
- **Real Exchange Rate** - this measure is the ratio of domestic price indices between two countries. A rise in the real exchange rate implies a worsening of international competitiveness for a country.

2.4.1 Treasury Bills

Treasury Bills are short-term debt instruments (securities) issued by the Government of Ghana with a maturity of less than one year. They commonly have maturities of 91 days and 182 days. Treasury Bills are issued at discount from par, which means that rather than making fixed interest payments like conventional bonds, income is earned by the difference between the face value and the price of the security. Treasury bills are quoted for purchase and sale in the secondary market on an annualized discount percentage, or basis.

General calculation for the discount yield for **Treasury bills** is

$$\textit{Discount}\ \text{Yield}(\%) = \frac{\textit{face}\ \text{value - purchase price}}{\textit{face}\ \text{value}} \times \frac{360}{\textit{days}\ \text{till maturity}} \times 100\%$$

Face value, simply put, is the stated value of an investment. For stocks, face value is the par value, or original price, of the stock. For bonds and other debts, face value is the principal amount of the debt. Market value, on the other hand, is the price at which buyers and sellers reach agreement in secondary markets such as stock exchanges and debt-purchase agreements. The market value of an investment can deviate considerably from its face value. Investments with market values higher than their face values have appreciated in value enough to have earned a profit over their original value, and the opposite holds true as well.

2.4.2 Inflation

In economics, **inflation** is a rise in the general level of prices of goods and services in an economy over a period of time. When the general price level rises, each unit of currency buys fewer goods and services. Consequently, inflation also reflects an erosion in the purchasing power of money – a loss of real value in the internal medium of exchange and unit of account within the economy. A chief measure of price inflation is the inflation rate, the annualized percentage change in a general price index (normally the Consumer Price Index) over time.

Inflation's effects on an economy are various and can be simultaneously positive and negative. Negative effects of inflation include an increase in the opportunity cost of holding money, uncertainty over future inflation which may discourage investment and savings, and if inflation is rapid enough, shortages of goods as consumers begin hoarding out of concern that prices will increase in the future. Positive effects include ensuring that central banks can adjust real interest rates (intended to mitigate recessions), and encouraging investment in non-monetary capital projects.

Economists generally agree that high rates of inflation and hyperinflation are caused by an excessive growth of the money supply. Views on which factors determine low to moderate rates of inflation are more varied. Low or moderate inflation may be attributed to fluctuations in real demand for goods and services, or changes in available supplies such as during scarcities, as well as to growth in the money supply. However, the consensus view is that a long sustained period of inflation is caused by money supply growing faster than the rate of economic growth.

Today, most economists favor a low and steady rate of inflation. Low (as opposed to zero or negative) inflation reduces the severity of economic recessions by enabling the labor market to adjust more quickly in a downturn, and reduces the risk that a liquidity trap prevents monetary policy from stabilizing the economy. The task of keeping the rate of inflation low and stable is usually given to monetary authorities. Generally, these monetary authorities are the central banks that control monetary policy through the setting of interest rates, through open market operations, and through the setting of banking reserve requirements.

2.4.3 Money Supply

The Bank of Ghana defines money supply as demand deposits at the banks (both Primary and secondary) plus currency in the hands of the general public. According to Wikipedia **Monetary policy** is the process by which the monetary authority of a country control the supply of money, often targeting a rate of interest for the purpose of promoting economic growth and stability. The official goals usually include relatively stable prices and low unemployment. Monetary theory provides insight into how to craft optimal monetary policy. It is referred to as either being expansionary or contractionary, where an expansionary policy increases the total supply of money in the economy more rapidly than usual, and contractionary policy expands the money supply more slowly than usual or even shrinks it. Expansionary policy is traditionally used to try to combat unemployment in a recession by lowering interest rates in the hope that easy credit will entice businesses into expanding. Contractionary policy is intended to slow inflation in order to avoid the resulting distortions and deterioration of asset values.

Monetary policy differs from fiscal policy, which refers to taxation, government spending, and associated borrowing.

2.4.4 Consumer Price Index

A Consumer Price Index (CPI) measures changes in the price level of consumer goods and services purchased by households. The CPI in the United States is defined by the Bureau of Labor Statistics as "a measure of the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services."

The CPI is a statistical estimate constructed using the prices of a sample of representative items whose prices are collected periodically. Sub-indexes and sub-sub-indexes are computed for different categories and sub-categories of goods and services, being combined to produce the overall index with weights reflecting their shares in the total of the consumer expenditures covered by the index. It is one of several price indices calculated by most national statistical agencies. The annual percentage change in a CPI is used as a measure of inflation. A CPI can be used to index (i.e., adjust for the effect of inflation) the real value of wages, salaries, pensions, for regulating prices and for deflating monetary magnitudes to show changes in real values.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter gives an in-depth description of ANN, tries to illustrate the connection between ANN and the Human brain. It basically presents the brain behind this research.

3.2 The Human Brain

Human Brain is said to have a highly complex, non – linear and parallel computation with structural constituents known as Neurons. There are massively parallel neurons in the human brain, about 10 billion neurons and 60 trillions of interconnection. We can only try to mimic very small part of the human brain using computers and electronic networks.

On the matter of memory, there is no comparison. Neural networks are potentially faster and more accurate than humans.

Many studies suggest that humans may use less than 10 percent of their brains' potential power. While this anecdotal evidence has not been scientifically proven, it is one of the many mysteries of the human brain. Some scientists state that human memory cells are located in certain areas of the brain. Others state that memory is distributed throughout the brain and there is no specific memory location. Of course, nothing is clear.

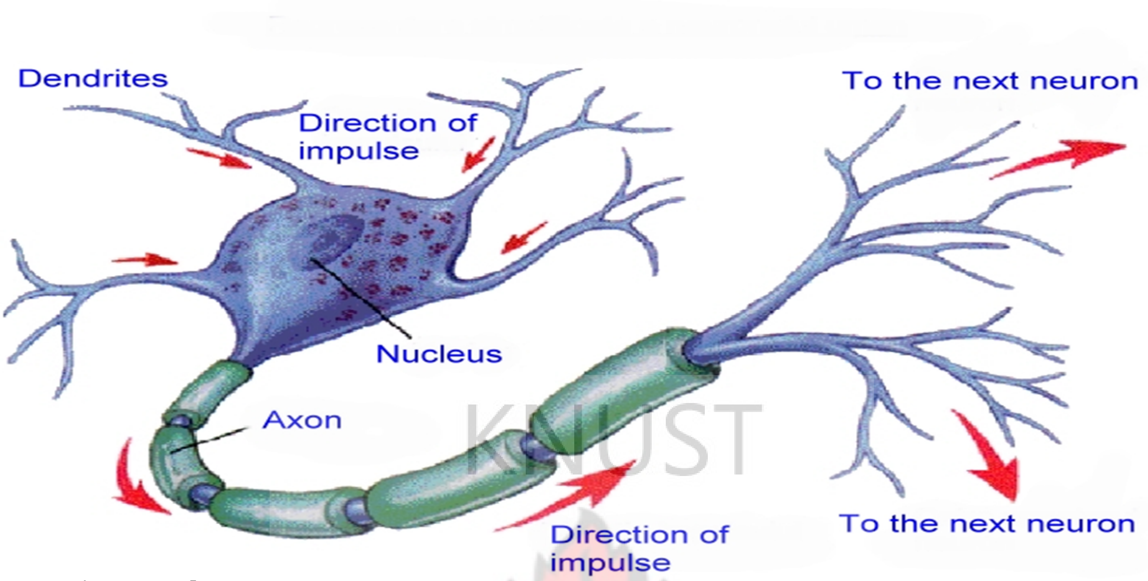


Fig 3.1: The Human Neuron

Human Behavior

When a child is born, what does the child know? To our knowledge, the child knows only how to cry. The child probably does not know its parents. When the child grows, the step by step learning process begins. First, the child learns to drink milk. Then the child learns to identify its parents. Every time a child learns something, it is encoded into some portion of the brain.

Yet there is a difference in the way the information is stored in brain. Some information or instances are "hard-coded" within the brain. As a result, we never forget certain things. For example, once we learn to swim, we never forget swimming. Though it appears normal to say that we know swimming, there is a mystery behind this. Why are we unable to forget the swimming? The reason might be that when we are fully trained to swim, it is hard-coded into our brain. There are many examples of unforgettable information. Another example is once we learn that 1 + 1 equals 2, we never forget that fact. Why? The reason is that it is completely learned.

These examples demonstrate that we can learn, understand and remember certain things completely, partially and sometimes not at all. Depending on our capacity for learning, the information is stored in our brain. Whatever is incompletely learned will lose its strength and not be retained in our brain. So, if we do not practice what we learned, we start to forget. Consequently, by practice or training, we can hard-code some selected things into our brains. Naturally, we cannot become expert in all areas. For example, it is difficult for one person to learn all of civil, computer, mechanical and electrical engineering along with medicine. We choose our subject areas based on our subjective interests. Even if you learn computer engineering, there are several areas within computer science. We cannot learn all areas and become an expert on everything in computer science. We choose one area and become inquisitive in that area searching for extreme interest. Finally, when we prove that we know much of that area, we are regarded as an expert in that area.

Many neuroscientists believe that learning stimulates new dendrite connections between neurons. Greater usage of the brain through learning and stimulation creates greater dendrite connectivity. Thus, as we learn more and more, we become more intelligent. Wisdom is not created through genetics. Wisdom and knowledge are based on how we learn and how we practice what we learned.

Neural Network Behavior

Now let us compare this human activity with neural networks. Whenever we create a new neural network, it is like giving birth to a child. After giving birth, we start to train the network. Not surprisingly, we may have created the neural network for certain applications or purposes. Here, the difference between childbirth and neural networks is obvious; first, we decide why we need a neural net and create it. Childbirth results are random in nature. When a child is born we do not know where the child will concentrate its studies through life. We leave it in the hands of the child and its parents. Naturally, parents play an extremely important role in child development and this is similar to the person creating a neural network. In the same way that a child becomes an expert in an area, we train the neural networks to become expert in an area. Once we establish an automatic learning mechanism in neural networks, it practices and starts to learn on its own and does its work as expected. Once it is proven that the neural network is doing its intended job correctly, we call it an "expert" and it operates according to its own decisions and judgment.

In our daily life, in many instances we have already transferred decision-making processes to computers. For example, say you attempt to purchase a product using a credit card over the Internet. For some reason, the billing address does not match the mailing address; it may be due to missing letters or misspelled words or other reasons. Although you are the correct person using a valid credit card, the purchase does not go through because the seller's computer does not allow transactions with a mismatch in the address. Based on this computer verification, the seller decides not to process your request. Although instances such as this happen daily in our lives, we tend to forget the computer's role in the decision.

Comparison

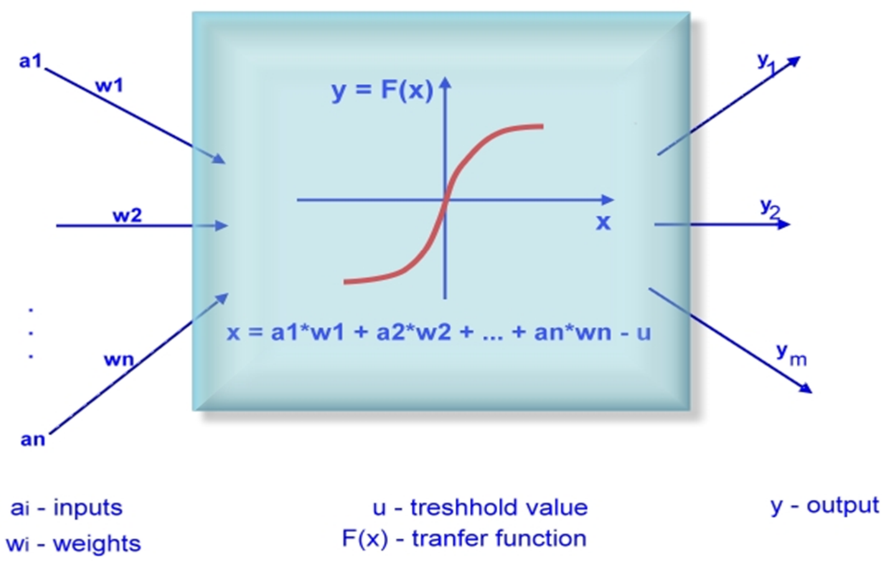
Now the question remains, what is the difference between human and neural networks? Both can learn and become expert in an area and both are mortal. The main difference is, humans can forget but neural networks cannot. Once fully trained, a neural net will not forget. Whatever a neural network learns is hard-coded and becomes permanent. A human's knowledge is volatile and may not become permanent. There are several factors that cause our brain cells to die and if they do, the information that is stored in that part is lost and we start to forget.

The other difference is accuracy. Once a particular application or process is automated through a neural network, the results are repeatable and accurate. Whether the process is replicated one thousand times or one million times, the results will be the same and will be as accurate as calculated the first time. Human beings are not like that. The first 10 processes may be accurate, but later we may start to make mistakes in the process. Another key difference is speed. Neural networks can be hardware or software. It is obvious that neural networks are much faster than humans in processing data and information.

Conclusion

Neural networking promises to provide computer science breakthroughs that rival anything we have yet witnessed. Once neural networks are trained properly, they can replace many human functions in targeted areas.

Fig 3.2: The Artificial Neuron



3.3 Artificial Neural Networks (ANN)

ANNs offer a computational approach that is quite different from conventional digital computation. Digital computers operate sequentially and can do arithmetic computation extremely fast. Biological neurons in the human brain are extremely slow devices and are capable of performing a tremendous amount of computation tasks necessary to do everyday complex tasks, commonsense reasoning, and dealing with fuzzy situations. The underlining reason is that, unlike a conventional computer, the brain contains a huge number of neurons, information processing elements of the biological nervous system, acting in parallel. ANNs are thus a parallel, distributed information processing structure consisting of processing elements interconnected via unidirectional signal channels called connection weights. Although modeled after biological neurons, ANNs are much simplified and bear only superficial resemblance. Some of the major attributes of ANNs are:

- It exploits non – linearity in the essence that there exist a simple non – linear equation connecting inputs to outputs.
- Input – output mapping: ANN's can undergo a learning process in which the inputs are fed with an idea of what the expected output is going to be. If the expected outputs are quite different from the actual output, the parameters in the system can be adjusted such that for a giving set of inputs we can obtain the output that is closer to the expected output. This might not be achieved directly, hence there is a continuous adjustment of the parameters such that the difference between the actual and expected is small.
- Adaptivity: the free parameters can be adapted to changes in the surrounding environment.

- Evidential Response: Gives response with confidence levels and decisions with a measure of confidence.
- Fault Tolerance: Cases where a particular connection is not functioning, the network still works.
- VLSI implementation: Using the Very Large Scale Integrated Circuit, it is possible to integrate a large number of neurons together.
- Neurobiological analogy: Motivated by the biological neuron networks.

3.3.1 Individual Neuron

The individual processing unit in ANNs receives input from other sources or output signals of other units and produces an output as shown in figure 3.3. The input signals (x_i) are multiplied with weights (w_{ji}) of connection strength between the sending unit "*i*" and receiving unit "*j*". The sum of the weighted inputs is passed through activation function. The output may be used as an input to the neighboring units or units as the next layer.

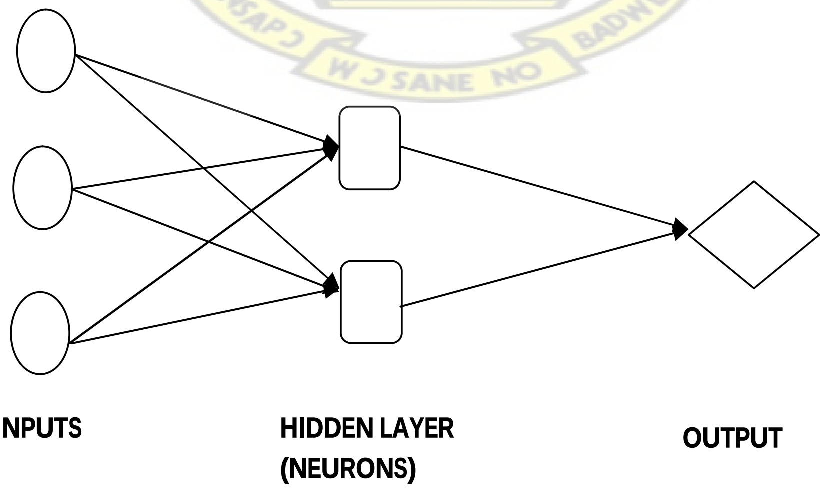


Fig 3.3 Schematic Diagram of a Neural Network

3.3.2 Adaptation of Neural Network

An interesting property of neural networks is their ability to learn. Most newly programmed neural networks are not able to perform their task with the desired accuracy at once. Usually a network behavior is adapted in a *learning or training process*. During this process the network is iteratively provided with a set of input patterns together with the corresponding output patterns until it produces the desired output. This set of input patterns and corresponding output patterns is called a *training set*. While training, the network may change the values of its parameters according to the applied *learning rule*.

The purpose of training a neural network on a certain task depends on important assumption. After the training phase the neural network is assumed to perform its task satisfactory on previously unencountered input patterns: the training is useful only if the knowledge gained from training patterns generalizes to other input patterns.

Therefore it is important for the training set to be representative for all input patterns on which the network will perform its task.

Two conditions have to be fulfilled regarding the representativeness of training patterns:

- The training patterns must belong to the class of patterns which the network is expected to process.
- The training pattern must be selected from input space according to the distribution in which all input patterns occur in it. A network cannot be expected to predict correctly when it is trained on a training set with too many outliers.

3.3.3 Computation of units

McCulloch and Pitts proposed a simple model of a neuron as a binary threshold unit. Specifically, the neuron model computes a weighted sum $h_i = \sum_j w_{ij} V_j$ of its inputs from other units, and produces a one or a zero.

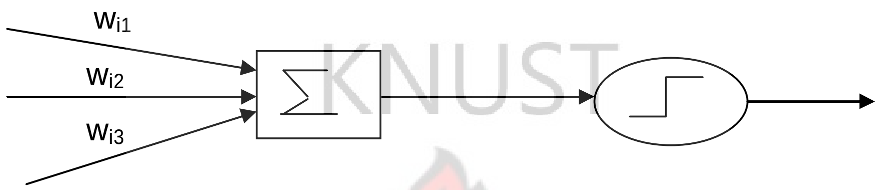


Fig 3.4: Model of the McCulloch – Pitts neuron

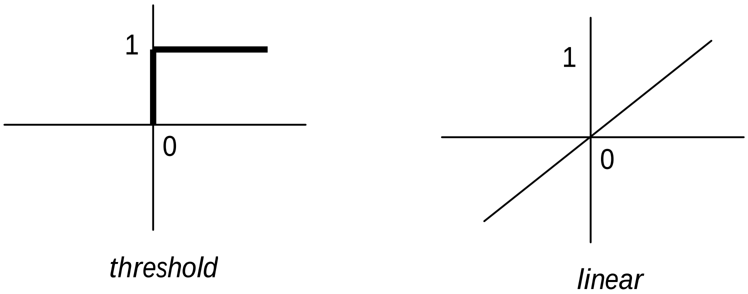
This depends on the value of the sum, whether it is above or below a certain threshold. The networks in this project are built from these neurons, which are connected through unidirectional links. Each link has a value, the weight of the link. Each unit produces a continuous valued activation V_i. The activation of a unit is $V_i = f(h_i) = f(\sum_j w_{ij} V_j)$ where *f(h)* denotes the activation or transfer function.

Now we can make a classification of units based on their transfer function:

- *Threshold* units incorporate a threshold function. The activation functions are restricted to 1 and 0, as stated before. For optimization techniques on an error – or cost – function a continuous differentiable transfer function is desirable. These units have continuous valued activations.
- *Linear* units use a function *f(h)=h.*

- *Non – Linear* units are most commonly used in gradient descent learning. These units mostly have sigmoid transfer functions, such as *f(h)=tanh(h)* or $f(h) = \frac{1}{1+e^{-h}}$

The three different transfer functions are shown in fig 3.5



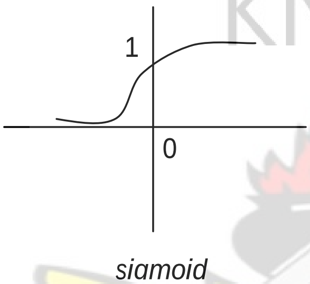


Fig 3.5: Three different transfer functions

3.4 Types of Artificial Neural Networks

There are various types of Artificial Neural Networks; an artificial neural network is a computational simulation of a biological neural network. These models mimic the real life behavior of neurons and the electrical messages they produce between input (such as from the eyes or nerve endings in the hand), processing by the brain and the final output from the brain (such as reacting to light or from sensing touch or heat). There are other ANNs which are adaptive systems used to model things such as environments and population (Wikipedia).

3.4.1 Feed forward Neural Networks

A feedforward neural network is an interconnection of perceptrons in which data and calculations flow in a single direction, from the input data to the outputs.

The neurons in each layer feed their output forward to the next layer until we get the final output from the neural network. There can be any number of hidden layers within a feedforward network. The number of neurons can be completely arbitrary.

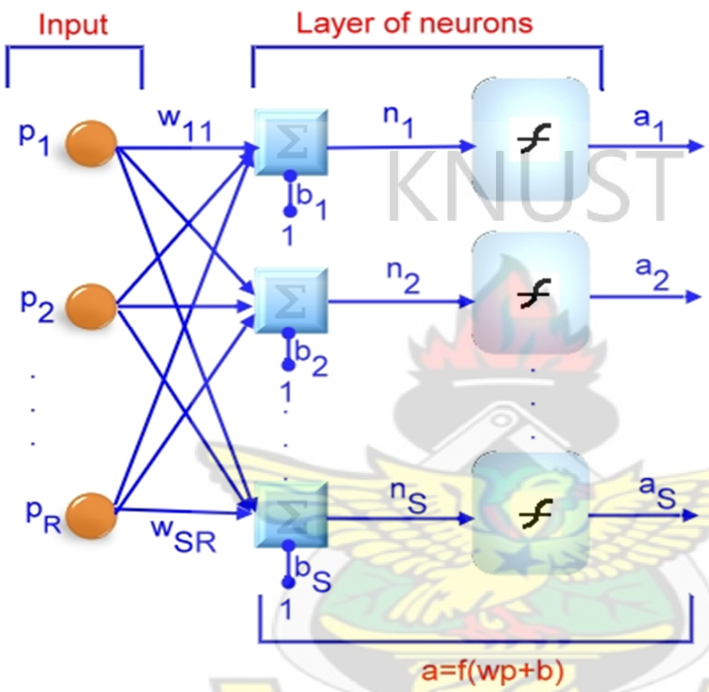


Fig 3.6: A Simple Feed forward Network

3.4.2 Radial Basis Function (RBF) Network

Radial basis functions are powerful techniques for interpolation in multidimensional space. A RBF is a function which has built into a distance criterion with respect to a center. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in multi-layer perceptrons. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian.

RBF networks have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptrons. This is because the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Linearity ensures that the error surface is quadratic and therefore has a single easily found minimum. In regression problems, this can be found in one matrix operation. In classification problems the fixed non-linearity introduced by the sigmoid output function is most efficiently dealt with using iteratively re-weighted least squares.

3.4.3 Recurrent Neural Networks (RNNs)

Contrary to feedforward networks, recurrent neural networks (RNNs) are models with bi-directional data flow. While a feedforward network propagates data linearly from input to output, RNNs also propagate data from later processing stages to earlier stages. RNNs can be used as general sequence processors.

3.4.4 Simple Recurrent Networks (SRNs)

The topology of SRNs is similar to that of feed – forward networks, but there are additional feedback connections that make the network suitable for the use on time series. A SRN typically consists of one input, one context, one hidden, and one output layer. The context layer receives feedback from a higher layer or from itself and it provides the network with information about previous activations of certain units. However, the weights are set to fixed values, and only the weights on the forward connections are trained using for example the standard back – propagation algorithm.

3.4.5 Fully recurrent networks

The difference between SRN's and fully recurrent networks is that a fully recurrent network includes direct or indirect loops of connections, and the weights on these connections can be learned. Example of such networks are Hopfield and Boltzman networks.

3.5 Neural Network Architecture and Training

Training is the procedure by which the Neural Network learns and understands the relationship between the input and output variables. Learning in biological systems may be considered as modifications made to the weights (synaptic connections) that exist between the neurons. Learning or training in an Artificial Neural Network is brought about by introduction of the network to a validated set of input / output data where the training algorithm iteratively adjusts the synaptic connection weights. These connection weights store the information learned by the network and are necessary to solve specific problems during the testing phase of the network validation process. Neural Networks are characterized by the following properties:

- The pattern of connections between the various network layers (Network types).
- Number of neurons in each layer (complexity).
- Learning algorithm.
- Neuron activation functions.

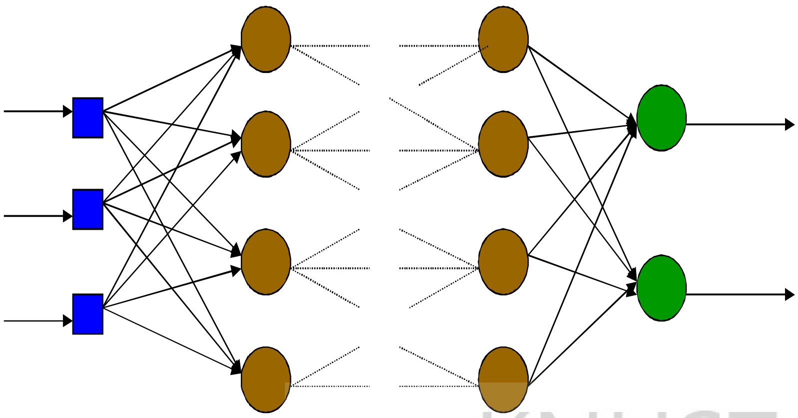
Generally speaking, a Neural Network is a set of connected input and output units where each connection has a weight associated with it. The learning phase involves the network's ability to adjust the weights so as to be able to correctly forecast or classify the output target of a given set of input data.

Given the numerous types of Neural Network architectures that have been developed in the literature, the important types of Neural Networks often used for forecasting and classification problems are discussed.

3.5.1 Multilayer Perceptrons

Multilayer Perceptions are layered Feed Forward networks classically trained with static back propagation algorithms. These networks are extensively used in countless applications requiring static pattern classification. Their key property is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train relatively slowly, and require comparatively a large amount of training data sets.

To be able to solve nonlinearly separable problems, a number of neurons are connected in layers to build a multilayer perceptron. Each of the perceptrons is used to identify small linearly separable sections of the inputs. Outputs of the perceptrons are combined into another perceptron to produce the final output. In a multilayer perceptron, the neurons are arranged into an input layer, an output layer and one or more hidden layers. The learning rule for the multilayer perceptron is known as "the generalized delta rule" or the "Backpropagation rule".



Input Layer **Hidden Layers** **Output Layers**

Fig 3.7: Multilayer Perceptron Architecture

As a humans needs to learn and adapt to changes so do the network also. For the purpose of the study, the following training mechanisms will be discussed.

3.5.2 Gradient Descent Algorithm

Given a differentiable scalar field $f(x)$ and an initial guess x_1 gradient descent iteratively moves the guess toward lower values of f by taking steps in the direction of the negative gradient $-\nabla f(x)$. Locally, the negated gradient is the steepest descent direction, i.e., the direction that x would need to move in order to decrease f the fastest. The algorithm typically converges to a local minimum, but may rarely reach a saddle point, or not move at all if x_1 lies at a local maximum.

For networks having differentiable activation functions, there exist a powerful and computationally efficient method, called error Backpropagation for finding the derivatives of an error function with respect to the weights and biases in the network. Gradient Descent algorithm is the most commonly used error Backpropagation method. If s is the network designed by taking the scalar product of the connections (weights) and the stimulus then we define,

$$s = \underline{w}^T \underline{x}$$

Given a training set of input – target pairs of the from

$$H = \{(\underline{x}^1, t^1), (\underline{x}^2, t^2), \ldots, (\underline{x}^n, t^n)\}$$

Now, defining a single pair error measure based on $(\mathrm{t}^p, \mathrm{O}^p)$, where O^p is the output obtained from $\underline{i}^p$, we have

$$(\mathrm{e}^p)^2 = \frac{1}{2}(t^p - O^p)$$

The basic idea is to compute $\dfrac{d(e^p)^2}{d\underline{w}}$, we use this quantity to adjust $\underline{w}$. Let for the k^{th} element of $\underline{w}$, i.e we have

$$\frac{d(e^p)^2}{dw_k} = \frac{d(e^p)^2}{dO^p} \cdot \frac{d\ \mathrm{s}}{dw_k}$$
$$= -(t^p - O^p)x_k^p$$

Which is the gradient of the pattern error with respect to weight w_k and forms the basis of the gradient descent training algorithm.

Specifically, we assign the weight correction, Δw_k such that

$$\Delta w_k = -\alpha \frac{d(e)^2}{dw_k}$$

Where α is referred to as the learning weight, it controls the rate of falling.

Hence the weights are updated using

$$w_k^{j+1} = w_k^j + \Delta w_k$$

3.5.3 The Generalized Delta Rule

Since we are now using units with nonlinear activation functions, we have to generalise the delta rule:

The activation is a differentiable function of the total input, given by $O_k^p = F(s_k^p)$ in which $s_k^p = \sum_j w_{jk} x_j^p$. To get the correct generalization of the delta rule we must set $\Delta_p w_{jk} = -\alpha \frac{\partial e^p}{\partial w_{jk}}$ where w_{jk} represents input weight from *jth* input to the *kth* neuron.

The error measure e^p is defined as the total quadratic error from pattern *p* at the output units:

$e^p = \frac{1}{2}\sum_{o=1}^{N_o}(t_o^p - O_o^p)^2$, where t_o^p is the desired output for unit *o* when pattern *p* is clamped.

We further set $E = \sum_p e^p$ as the sum squared error.

We can write $\frac{\partial e^p}{\partial w_{jk}} = \frac{\partial e^p}{\partial s_k^p}\frac{\partial s_k^p}{\partial w_{jk}}$, by the equation $s_k^p = \sum_j w_{jk} x_j^p$ we see that the second factor is $\frac{\partial s_k^p}{\partial w_{jk}} = x_j^p$. If we define $\delta_k^p = -\frac{\partial e^p}{\partial s_k^p}$, we will get an update rule which is given by $\Delta_p w_{jk} = \alpha \delta_k^p x_j^p$. Next we try to figure out what δ_k^p should be for each unit *k* in the network.

Using chain rule, $\delta_k^p = -\frac{\partial e^p}{\partial s_k^p} = -\frac{\partial e^p}{\partial O_k^p}\frac{\partial O_k^p}{\partial s_k^p}$, from $O_k^p = F(s_k^p)$ we can say that $\frac{\partial O_k^p}{\partial s_k^p} = F^{'}(s_k^p)$ hence we get for any output unit *o*

$$\delta_o^p = (t_o^p - O_o^p)F_o^{'}(s_o^p) \qquad (3.1)$$

Secondly, if k is not an output unit but a hidden unit $k = h$, we do not readily know the contribution of the unit to the output error of the network. However, the error measure can be written as a function of the net inputs from hidden to output layer; $e^p = e^p(s_1^p, s_2^p, ..., s_j^p)$ and we use the chain rule to write

$$\frac{\partial e^p}{\partial x_h^p} = \sum_{o=1}^{N_o}\frac{\partial e^p}{\partial s_o^p}\frac{\partial s_o^p}{\partial x_h^p} = \sum_{o=1}^{N_o}\frac{\partial e^p}{\partial s_o^p}\frac{\partial}{\partial x_h^p}\sum_{j=1}^{N_h} w_{ko}x_j^p = \sum_{o=1}^{N_o}\frac{\partial e^p}{\partial s_o^p}w_{ho} = -\sum_{o=1}^{N_o}\delta_o^p w_{ho}$$

Substituting this in the equation $\delta_k^p = -\frac{\partial e^p}{\partial s_k^p} = -\frac{\partial e^p}{\partial O_k^p}\frac{\partial O_k^p}{\partial s_k^p}$ yields

$$\delta_h^p = F^{'}(s_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho} \qquad (3.2)$$

Equations (3.1) and (3.2) give a recursive procedure for computing the δ's for all units in the network, which are then used to compute the weight changes.

This procedure constitutes the generalized delta rule for a feed-forward network of non-linear units.

3.5.4 Understanding and Working with Back-propagation

The equations derived in the previous section may be mathematically correct, but what do they actually mean? Is there a way of understanding back-propagation other than reciting the necessary equations? The answer is, of course, yes. In fact, the whole back-propagation process is intuitively very clear. What happens in the above equations is the following. When a learning pattern is clamped, the activation values are propagated to the output units, and the actual network output is compared with the desired output values, we usually end up with an error in each of the output units. Let's call this error e_o for a particular output unit o. We have to bring e_o to zero The simplest method to do this is the greedy method: we strive to change the connections in the neural network in such a way that, next time around, the error e_o will be zero for this particular pattern. We know from the delta rule that, in order to reduce an error, we have to adapt its incoming weights according to $\Delta w_{ho} = (t_o - O_o)x_h$.

That's step one but this alone is not enough; when we only apply this rule, the weights from input to hidden units are never changed, and we do not have the full representational power of the feed-forward network as promised by the universal approximation theorem. In order to adapt the weights from input to hidden units, we again want to apply the delta rule. In this case, however, we do not have a value for δ for the hidden units. This is solved by the chain rule which does the following: distribute the error of an output unit to all the hidden units which is connected to it, weighted by this connection. Differently put, a hidden unit h receives a delta from each output unit equal to the delta of that output unit weighted with (multiplied by) the weight of the connection between those units.

The application of the generalised delta rule thus involves two phases: During the first phase the input x is presented and propagated forward through the network to compute the output values O_o^p for each output unit. This output is compared with its desired value t_o, resulting in an error signal δ_o^p for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and appropriate weight changes are calculated. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and appropriate weight changes are calculated.

The Weight adjustments with sigmoid activation function are illustrated below:

- The weight of a connection is adjusted by an amount proportional to the product of an error signal δ, on the unit k receiving the input and the output of the unit j sending this signal along the connection: $\Delta_p w_{jk} = \alpha \delta_k^p x_j^p$
- If the unit is an output unit, the error signal is given by $\delta_o^p = (t_o^p - O_o^p) F^{'}(s_o^p)$. Take as the activation function F the 'sigmoid' function as defined $O^p = F(s^p) = \dfrac{1}{1+e^{-s^p}}$. In this case the derivative is equal to

$$\begin{aligned} F^{'}(s^p) &= \frac{\partial}{\partial s^p} \frac{1}{1+e^{-s^p}} \\ &= \frac{1}{(1+e^{-s^p})^2}(-e^{-s^p}) \\ &= \frac{1}{(1+e^{-s^p})} \frac{(-e^{-s^p})}{(1+e^{-s^p})} \\ &= O^p(1-O^p) \end{aligned}$$

Such that the error signal for an output unit can be written as $\delta_o^p = (t_o^p - O_o^p)O_o^p(1-O_o^p)$.

- The error signal for a hidden unit is determined recursively in terms of error signals of the units to which it directly connects and the weights of those connections. For the sigmoid activation function:

$$\delta_h^p = F'(s_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho} = O_h^p(1-O_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho}$$

3.5.5 Learning rate and momentum

The learning procedure requires that the change in weight is proportional to $\frac{\partial e^p}{\partial w_{jk}}$. True gradient descent requires that infinitesimal steps are taken. The constant of proportionality is the learning rate . For practical purposes we choose a learning rate that is as large as possible without leading to oscillation. One way to avoid oscillation at large is to make the change in weight dependent of the past weight change by adding a momentum term: $\Delta w_{jk}(t+1) = \alpha\delta_k^p x_j^p + \Delta w_{jk}(t)$, where t indexes the presentation number and F is a constant which determines the effect of the previous weight change. Although, theoretically, the back-propagation algorithm performs gradient descent on the total error only if the weights are adjusted after the full set of learning patterns has been presented, more often than not the learning rule is applied to each pattern separately, i.e., a pattern p is applied, e^p is calculated, and the weights are adapted (p = 1, 2,..... P). There exists empirical indication that this results in faster convergence. Care has to be taken, however, with the order in which the patterns are taught. For example, when using the same sequence over and over again the network may become focused on the rest few patterns. This problem can be overcome by using a permuted training method.

3.5.6 Summary of the Backpropagation Algorithm

1. Run the network with the first set of input data set to obtain the network output.
2. For each output node compute $\delta_o^p = (t_o^p - O_o^p)O_o^p(1 - O_o^p)$.
3. For each hidden node compute $\delta_h^p = O_h^p(1 - O_h^p)\sum_{o=1}^{N_o} \delta_o^p w_{ho}$
4. Update the weights as follows $\Delta w_{jk}(t+1) = \alpha\delta_k^p x_j^p + \Delta w_{jk}(t)$

3.6 Learning Mechanisms

The actual choice of ANN is to mimic what goes on in the in the human brain. As the dendrites in the human brain has to adjust their strength until a desired response is reached, so does the adjustment of weights also occurs in ANN. The adjustments of weights are what constitute the learning process hence making the learning process an iterative process.

Learning processes can either be supervised or unsupervised. Supervised learning is the learning process in which changes in a network's weights and biases are due to the intervention of any external teacher. The teacher typically provides output targets. Learning process in which changes in a network's weights and biases are not due to the intervention of any external teacher is said to be Unsupervised. Commonly changes are a function of the current network input vectors, output vectors, and previous weights and biases. The five (5) basic learning rules are

- Error – Correction Learning
- Memory Based Learning
- Hebbian Learning
- Competitive Learning

- Boltzmann Learning

For the purpose of this study I will discuss the Error – Correction Learning.

In the error – correction learning, the error between the target and the network output is computed. Suppose our target is denoted *d*, and the network output by *y*, the error *e* between them is given by

$$e = d - y$$

Iteratively it can be represented as

$$e(n)=d(n)-y(n)$$

where n is the discrete time step. Ideally this error should be very small hence we represent the instantaneous error as

$$Minimise\ \mathrm{e(n)}=\frac{1}{2}\sum_{k} e_k^2(n)$$

Where k is each output neuron. After the error has been computed the weights can be updated by

$$\Delta w_{jk}(n) = \eta e_k(n) x_j(n)$$

Where x_i is the input vector and η is the learning rate which is chosen arbitrary. The updated weight will now be given by

$$w_{jk}(\mathrm{n}+1) = \mathrm{w}_{jk}(n) + \Delta\, \mathrm{w}_{jk}(n)$$

The error – correction learning is practically used in supervised learning.

3.7 The Levenberg – Marquardt Back- propagation Algorithm

The Levenberg–Marquardt algorithm blends the steepest descent method and the Gauss–Newton algorithm. Fortunately, it inherits the speed advantage of the Gauss–Newton algorithm and the stability of the steepest descent method. It's more robust than the Gauss–Newton algorithm, because in many cases it can converge well even if the error surface is much more complex than the quadratic situation.

Although the Levenberg–Marquardt algorithm tends to be a bit slower than Gauss–Newton algorithm (in convergent situation), it converges much faster than the steepest descent method. The basic idea of the Levenberg–Marquardt algorithm is that it performs a combined training process: around the area with complex curvature, the Levenberg–Marquardt algorithm switches to the steepest descent algorithm, until the local curvature is proper to make a quadratic approximation; then it approximately becomes the Gauss–Newton algorithm, which can speed up the convergence significantly, Hao Yu and Bogdan (2010).

According to Syed el at (2007), the Levenberg-Marquardt algorithm was designed to approach second order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares that is

$$F(\underset{\sim}{\mathrm{w}}) = \frac{1}{2}e^T e = \frac{1}{2}e^T(w)e(w) = \frac{1}{2}\sum_i^k \sum_j^p (O_{ij} - T_{ij})^2$$

where $\underset{\sim}{W} = [w_1, w_2.w_3, ..., w_N]^t$ consists of all weights of the network, the function of sum of squared errors is defined as $F(W) = \dfrac{1}{2}e^t e$

Newton's method for minimizing objective function is generated using well known recurrence formula $W_{i+1} = W_i - H^{-1}\nabla F(W)$ and $\nabla F(W)$ is the gradient of $F(W)$ then the Hessian matrix

$$H=\begin{bmatrix} \frac{\partial^2 E}{\partial w_1^{\ 2}} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^{\ 2}} & \cdots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_N^{\ 2}} \end{bmatrix}$$

can be approximated as $H \approx J^T J$ and the gradient can be computed as $g = J^T \underset{\sim}{e}$ where the Jacobian matrix J contains the first derivatives of the network errors with respect to weights and biases, and $\underline{e}$ is a vector of network errors. The Gauss – Newton update formula can be

$$W_{i+1} = W_i - (J_i^T J_i)^{-1} J_i^T \varepsilon_i$$

and

$$J=\begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & . \ . & \frac{\partial e_{11}}{\partial w_N} \\ \frac{\partial e_{21}}{\partial w_1} & \frac{\partial e_{21}}{\partial w_2} & . \ . & \frac{\partial e_{21}}{\partial w_N} \\ . & . & . \ . & . \\ . & . & . \ . & . \\ \frac{\partial e_{k1}}{\partial w_1} & \frac{\partial e_{k1}}{\partial w_2} & . \ . & \frac{\partial e_{k1}}{\partial w_N} \\ . & . & . \ . & . \\ . & . & . \ . & . \\ \frac{\partial e_{1p}}{\partial w_1} & \frac{\partial e_{1p}}{\partial w_2} & . \ . & \frac{\partial e_{1p}}{\partial w_N} \\ \frac{\partial e_{2p}}{\partial w_1} & \frac{\partial e_{2p}}{\partial w_2} & . \ . & \frac{\partial e_{2p}}{\partial w_1} \\ \frac{\partial e_{kp}}{\partial w_1} & \frac{\partial e_{kp}}{\partial w_2} & . \ . & \frac{\partial e_{kp}}{\partial w_N} \end{bmatrix}$$

Where $(J^T J)$ is a positive definite, but if it is not, then, we make some perturbation into it that will control the probability of being non positive define.

Thus, $H \approx J^T J + \lambda I$

$$W_{i+1} = W_i - (J_i^T J_i + \lambda I)^{-1} J_i^T \varepsilon_i \quad \text{\_\_\_\_\_\_\_\_\_\_\_\_}(3.6)$$

Where the quantity λ is called the learning parameter (adaptive value), it ensures that $J^T J$ is positive definite. It is always possible to choose λ sufficiently large enough to ensure a descent step. The learning parameter is decreased as the iterative process approaches to a minimum.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

This chapter presents data acquisition and preparation. The chapter focuses on the creation of the network architectures, the activation functions and the algorithms used to obtain the prediction.

4.2 Data Acquisition and Preparation

The thesis is aimed at looking at how best exchange rates can be predicted from inflation, money supply, consumer price index and money supply using Artificial Neural Network, hence data was acquired from the Research Department of Bank of Ghana, Kumasi. The monthly dataset spanned from January, 1997 to December 2010.

The study aims at forecasting Exchange rates using Inflation, Money Supply, Consumer price Index and Treasury Bills, hence the predictor variables (input variables) are Inflation, Money Supply, Consumer price Index and Treasury Bills, with Exchange rates being the target values.

In multilayer networks, sigmoid transfer functions are generally used in the hidden layers. These functions become essentially saturated when the net input is greater than $e^{-3} \cong 0.05$. When this happens at the beginning of the training process, the gradients will be very small, and the network training will be very slow. To avoid this, the data needs to be preprocessed, the data was normalized using the min – max normalization which rescales the data into the range of -1 and 1. Thus transforming the dataset in a way to enable the network perform efficiently. The min – max normalization is performed using the equation

$$\bar{x}_i = \left[\left(\frac{x_i - x_{\min}}{x_{\max} - x_{\min}}\right) * (1-(-1)) \right] + (-1) \qquad \text{\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_}(4.1)$$

Where $X = (x_1, x_2, x_3, ..., x_n)$ and $\bar{x}_i$ is the new data point after normalization.

The matlab code used for this is *mapminmax(X)*.

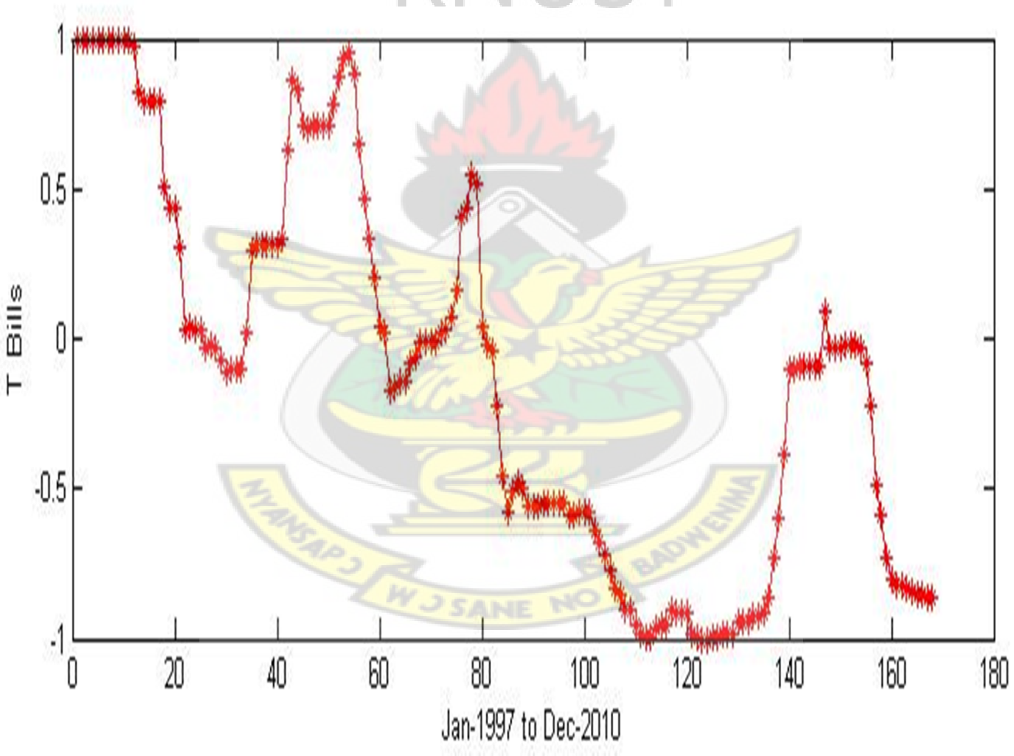


Fig 4.1a: Normalized Treasury Bills values over 168 months

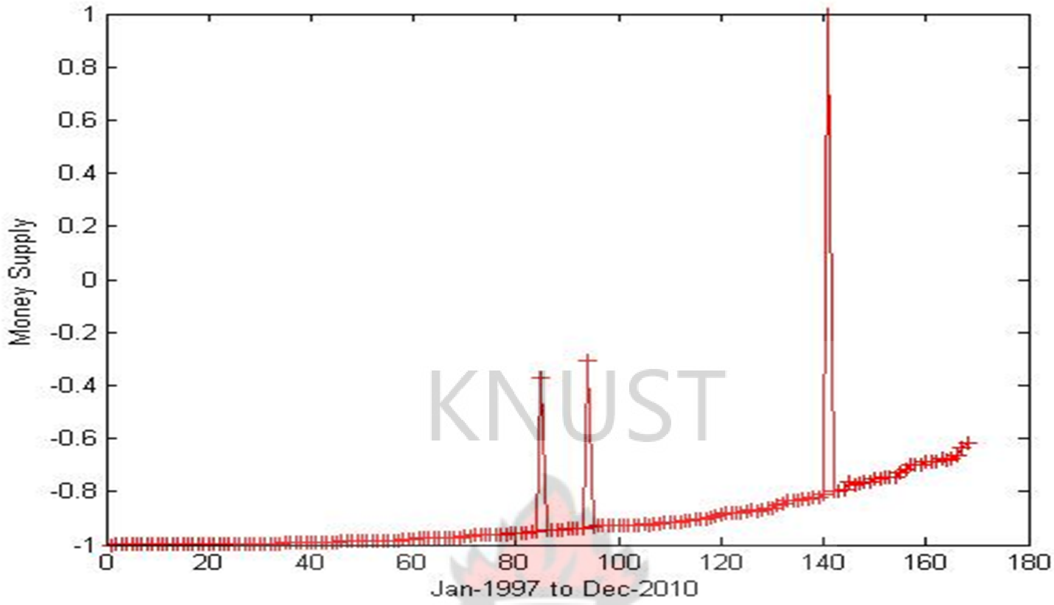


Fig 4.1b: Money Supply values over 168 months

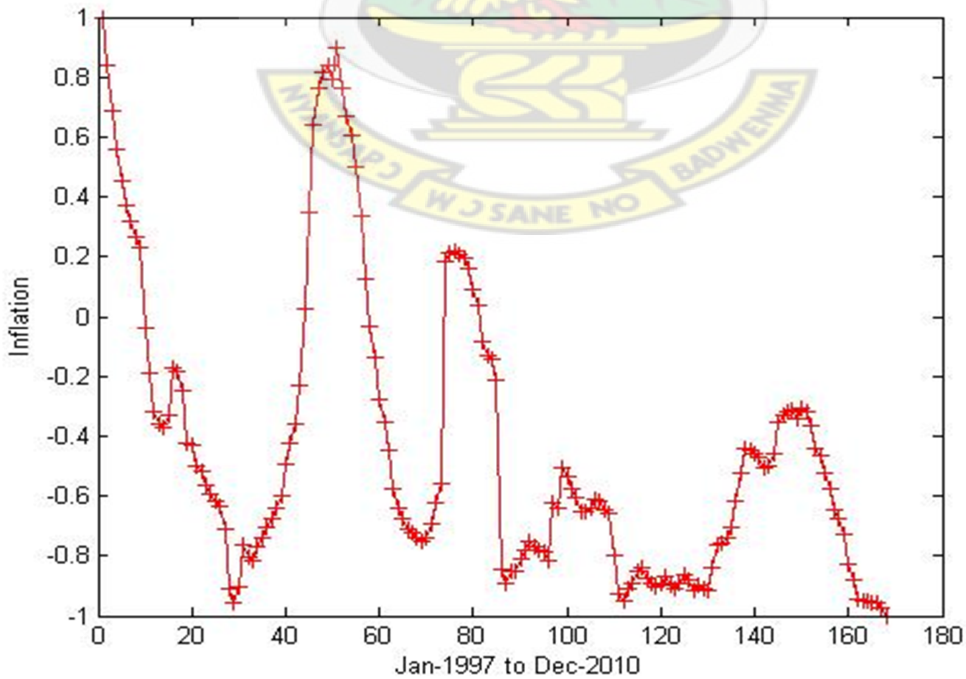


Fig 4.1c: Normalized Inflation values over 168 months

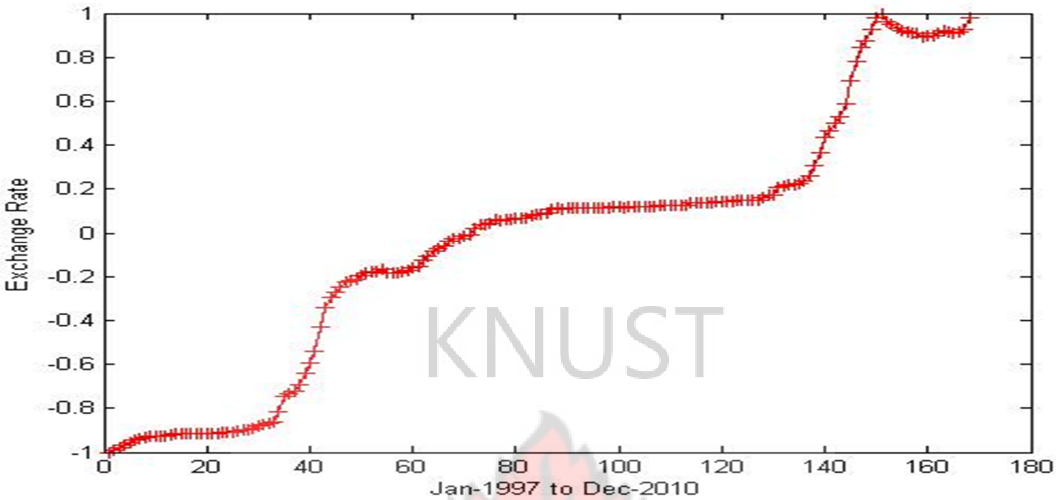


Fig 4.1d: Normalized Exchange Rate values over 168 months

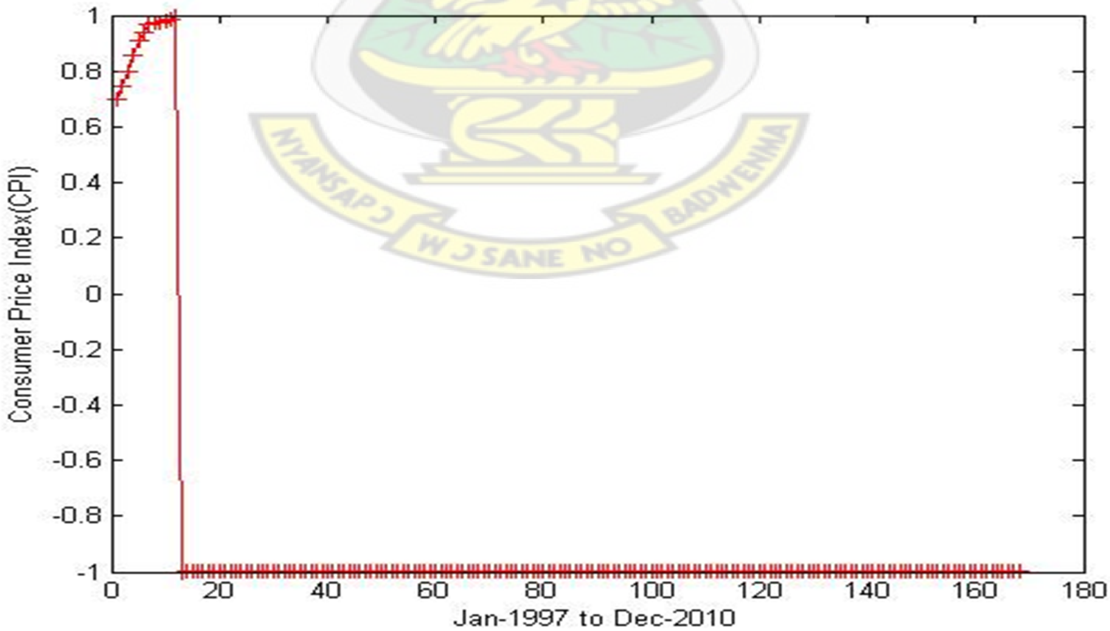


Fig 4.1e: Normalized Consumer Price Index values over 168 months

The figures above shows the trend of the data values after the min – max normalization procedure. Problems such as unknown data points and constant rows need to be checked but in my case such problems were not uncounted. From fig 4.1b it can be noticed that during January 2004, October 2004, September 2008 there was a significant increase in Money Supply relative to the surrounding months hence the spikes in the graph. From fig 4.1e, we could see a sharp decrease in CPI from December 97 against January 1998.

4.3 The Network Architecture

A two (2) layer feedforward network with backpropagation was used for the study. This consists of a hidden layer with twenty (20) input neurons chosen and an output layer with a single (1) neuron. Since there is no good theory surrounding the maximum number of neurons allowed in the hidden layer, hence the neurons in the hidden layer was varied until a minimum Mean Squared Error (MSE) was obtained.

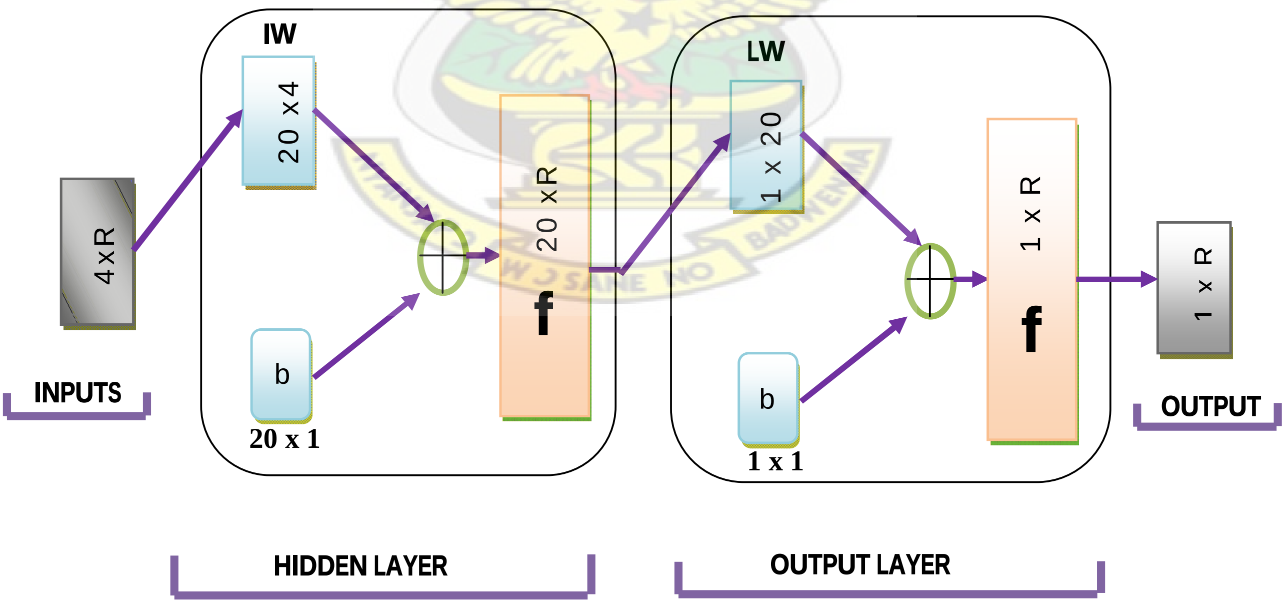


Fig 4.2: A two (2) layer feedforward network

Where ***R*** is the number of data points in each input set, ***b*** is the bias and ***f* is** the activation function

4.3.1 Data Division

The monthly data obtained from the Bank of Ghana consists of 168 data points. The data was divided into three (3) subsets at random thus 70%, 15% and 15%, namely the training set, validation set and the test set respectively. Out of the 168 data points, 118 was used for training, 25 for validation and 25 for testing.

4.3.2 Activation function

The activation function also called the transfer function. It determines the relationship between inputs and outputs of a node and a network. According to Chen and Chen (1995) any continuously differentiable function can be used as an activation function. For the purpose of this study two activation functions were used namely:

- $Tansig(n) = \frac{2}{1+e^{-2n}} - 1$ which is equivalent to the $\tanh(n) = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$ just that it runs faster than the MATLAB implementation of *tanh*. This was the activation function used in the hidden layer. It produces responses between -1 and 1.
- For the output layer a linear transfer function $Purelin(n) = n$ was used.

4.4 Network Training

After creating the network, the network training can now be done. For the purpose of this work Levenberg – Marquardt Backpropagation Algorithm is used. The application of Levenberg-Marquardt to neural network training is has been shown to be the fastest method for training moderate sized feed-forward neural networks (up to several hundred weights). It also has an efficient implementation in MATLAB software, since the solution of the matrix equation is a built-in function, so its attributes become even more pronounced in a MATLAB environment, Kuryati et al (2012). It is a combination of the gradient descent and Newton's method. As mentioned earlier, 118 of the data points were used for training (computing the gradient and updating the network weights and biases), the error on the validation set 25 data points were studied and the corresponding weight and biases are saved at the minimum of the validation set error. The test set error is used to compare the model with different models.

4.4.1 Network Performance

There are several methods available to evaluate forecast performance. The commonly used forecast accuracy measures are; Mean Error (ME), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) etc, Veerachai et al (2012). The performance measure used in this research is the MSE and RMSE, the linear correlation coefficient (r) as well as the Weighted Absolute Percentage Error (WAPE) was also calculated to measure the strength and the direction of the linear relationship between the output and target values. The underlining equations are as follows:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(T_k - O_k)^2 \qquad (4.2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(T_k - O_k)^2} \qquad (4.3)$$

$$WAPE = \frac{\sum |T_k - O_k|}{\sum T_k} \qquad (4.4)$$

$$r = \frac{n\sum T_k O_k - \left(\sum T_k\right)\left(\sum O_k\right)}{\sqrt{n\left(\sum T_k^{\ 2}\right) - \left(\sum T_k\right)^2}\sqrt{n\left(\sum O_k^{\ 2}\right) - \left(\sum O_k\right)^2}} \qquad \text{\_\_\_\_\_\_\_\_\_\_\_(4.5)}$$

Where T_k denotes the target data and O_k denotes the output data. The smaller the MSE, RMSE, WAPE, the better the prediction. The correlation coefficient *r* value of 1 indicate closed relationship while 0 is a random relationship and -1 a negative relationship.

4.5 Interpretation and Discussion of Results

For the implementation of the network, the data set collected from Bank of Ghana Research Department, Kumasi is used, Levenberg – Marquardt back propagation algorithm is used for training the network. Training automatically stops when generalization stops improving, as indicated by an increase in the Mean Squared Error (MSE) of the validation samples. The MSE is the average squared difference between output and target. Lower values are better while zero means no error. Regression R analysis is performed to measure the correlation between output and targets.

The performance of the proposed network when trained with Levenberg-Marquardt Backpropagation algorithm using Matlab R2012a, is shown in Figure 4.3

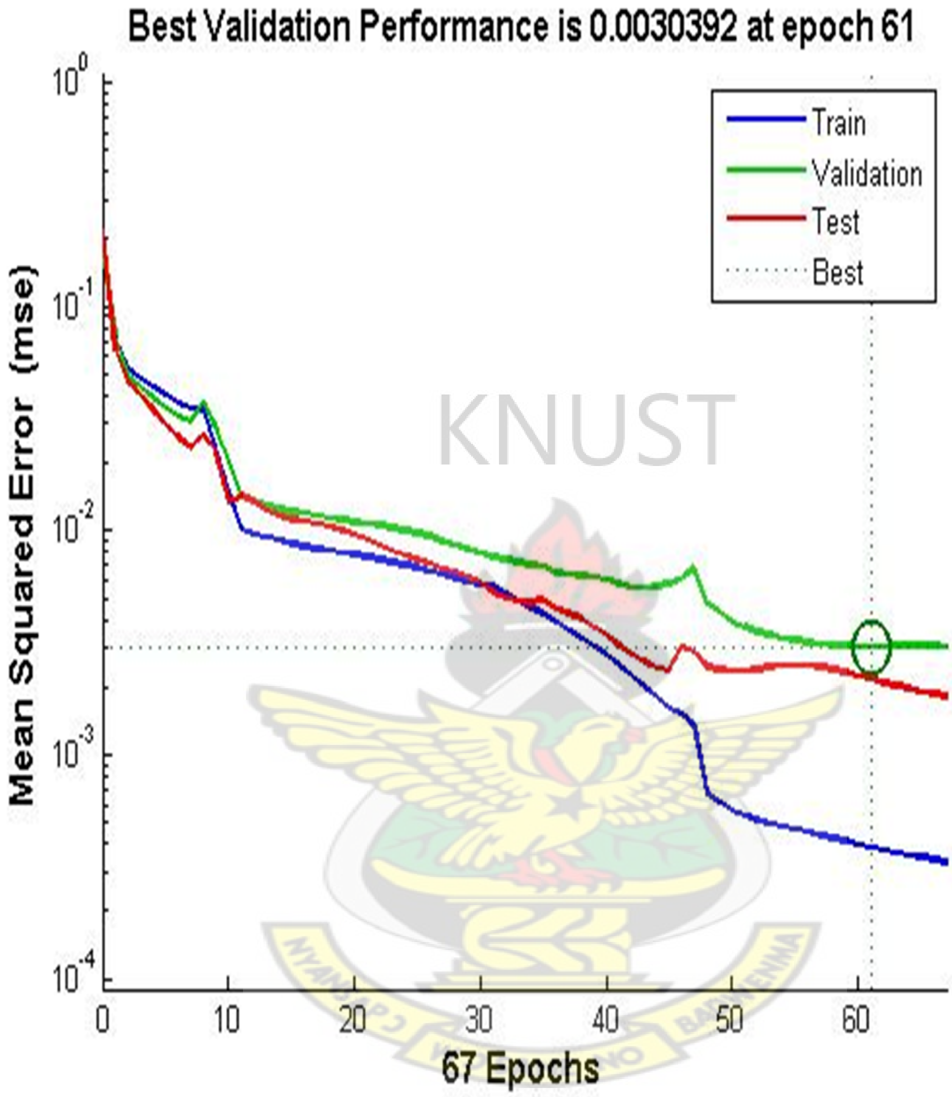


Figure 4.3: Performance of Levenberg – Marquardt Backpropagation Algorithm

From Figure 4.3, as indicated by the legend, the blue, green and red line indicates the trend in MSE associated with the network training, validation and testing respectively. Training the network means learning the relationship between the inputs and targets presented to the network. This involved computing the MSE at every point of the iteration (epochs) that is the difference between the target data and the output produced by the network as well as updating the network weights until a minimum MSE is obtained. As weights are presented to the training set at every epoch the MSE of the target – output on the training set is computed. The MSE of the validation set is also computed at every epochs and training stops when generalization stops as indicated by a rise in the MSE of the validation set.

From Figure 4.3 it is observed that the best validation performance 0.0030392 at epoch 61 is obtained. This means that the best weight combinations were obtained at that point. The corresponding MSE of the training and test set are indicated in the table below.

	Sample	MSE
Training	118	3.8616e-4
Validation	25	3.03916e-3
Testing	25	2.17134e-3

Table 4.1

The trend in the computation of gradients, adaptive value *mu* (learning parameter λ) and validation checks are shown in figure 4.4

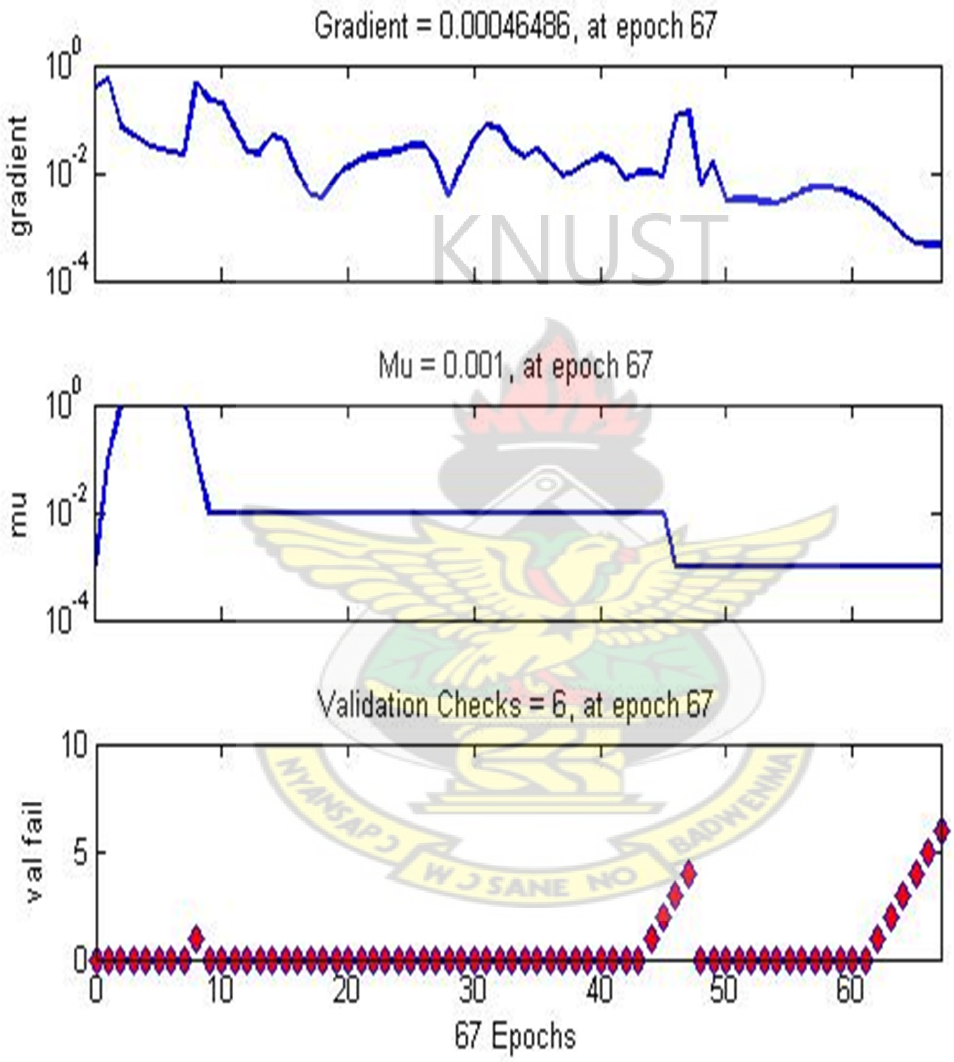


Figure 4.4: Network Training State

Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached (chosen as 100).
- The maximum amount of time is exceeded (chosen as infinite)
- Performance is minimized to the goal.
- The performance gradient falls below min_grad.(chosen as 1e-10)
- mu exceeds mu_max.(chosen as 1e10).
- Validation performance has increased more than max_fail times since the last time it decreased (maximum chosen as five(5) times).

The initial adaptive value was chosen as 0.001, as shown in Figure 4.4, the value is decreased after each successful step (reduction in MSE) by a decrease factor of 0.1 and multiplied by 10 whenever a step would increase the performance function (MSE).

It can be seen from the figure that training was stopped at epoch 67 because the validation failed more than five (5) times giving the best performance at epoch 61, at which the learning parameter was 0.001 and the gradient 0.00046486 at epoch 67.

The Regression plot shown in Figure 4.5 shows a perfect correlation between the output and the targets.

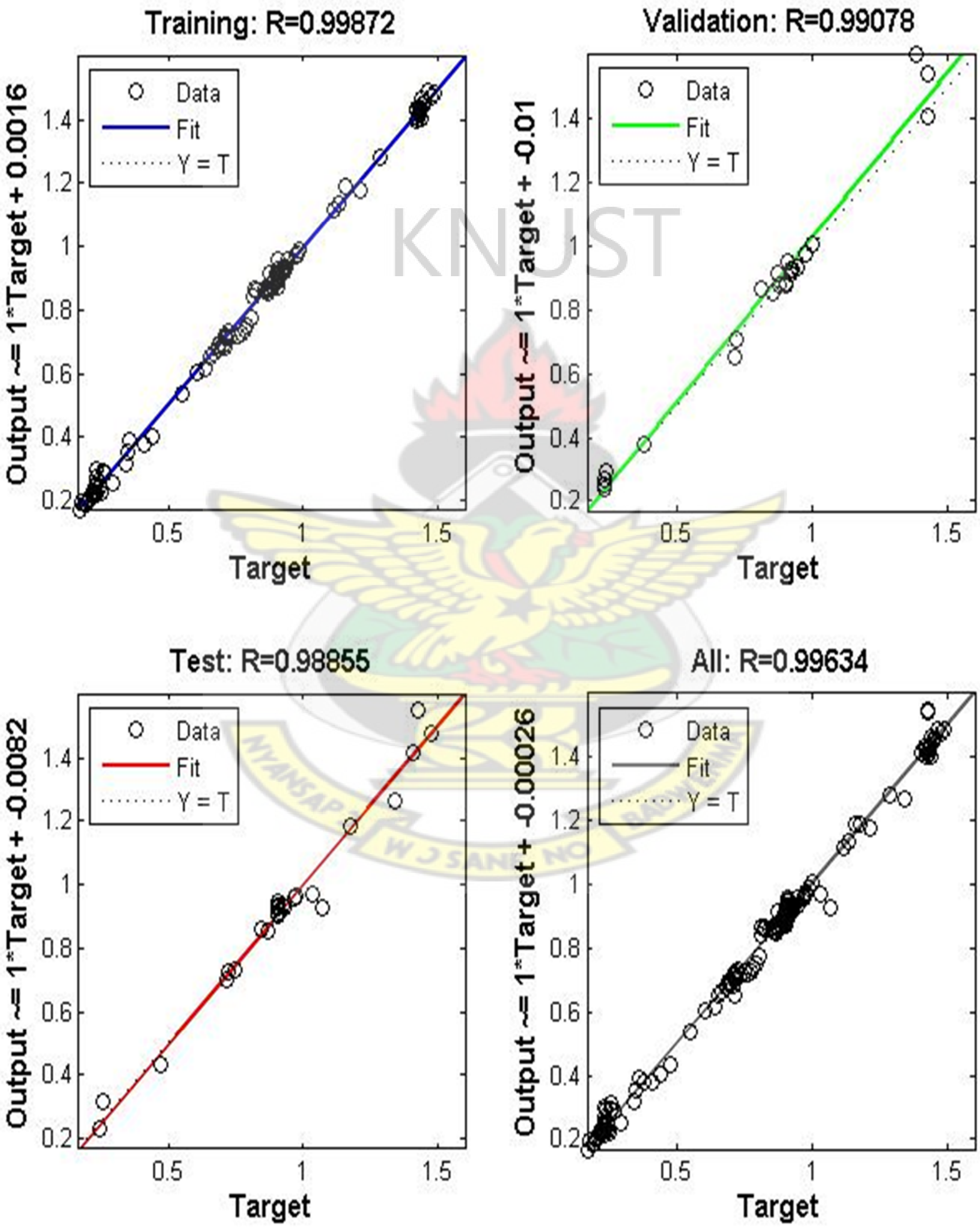


Figure 4.5: Network Training Regression

It is observed that the value of R is closest to 1 indicating accurate prediction thus the predicted values(output) and the target values are positively and closely correlated.

	Sample	R
Training	118	9.98717e-1
Validation	25	9.90778e-1
Testing	25	9.88550e-1

Table 4.2

With the overall R value given by 0.99634.

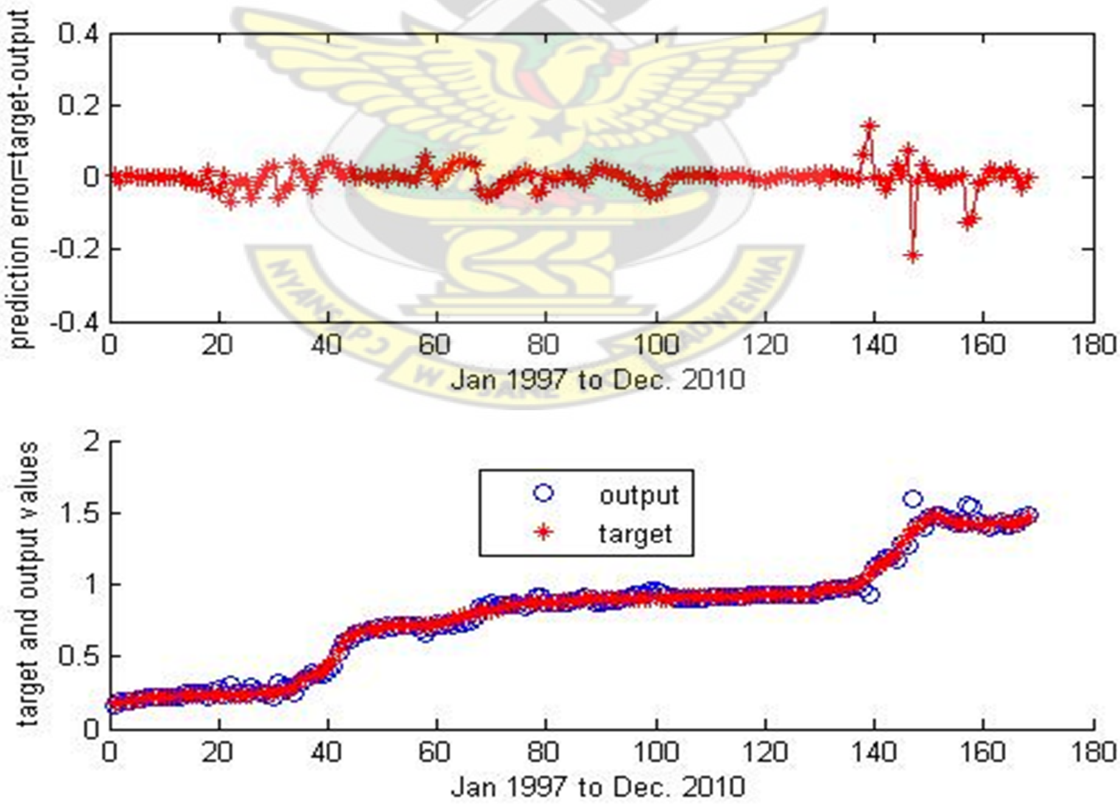


Figure 4.7: Subplot of Prediction error and a scatter diagram of Target and Output values

From figure 4.7 it can be seen that the minimum prediction error was -0.2162 at 147 month (Mar-2009) and the maximum 0.1405 at 139month (July -1998) but most of the errors are concentrated around the zero(0) line.

The scatter plot in Figure 4.7 depicts a close relation between the target and output values with an R value of 0.99634 as stated earlier.

The table below presents the overall MSE, RMSE, WAPE and the Prediction Accuracy.

Overall Prediction	
MSE	0.0010
RMSE	0.0324
WAPE	2.30%
Prediction Accuracy	97.70%

Table 4.3: Overall prediction performance

From Table 4.3, it can be seen that the overall prediction error is 2.3%, this gives the prediction accuracy of the network as 97.7%, hence it's a good network. The MSE and RMSE are also very small which is good for prediction purposes.

In summary, the Multi – layer feedforward network consists 2 layers with 20 neurons in the hidden layer and a single neuron in the output layer. With a tangent hyperbolic function in the hidden layer and a Linear function in the output layer. The number of neurons determines the dimension of the associated weight matrix.

Hence the associated weights in the hidden layer is a 20 x 4 matrix and a 20 x 1 Biases matrix

$$W^{hidden} = \begin{bmatrix} -1.4405 & -1.6495 & -1.8199 & 0.7352 \\ -1.1835 & -1.0694 & 1.9846 & -1.6903 \\ 1.9359 & -2.5990 & 2.3403 & -1.5298 \\ 0.1234 & 1.0238 & 1.8955 & -1.8912 \\ 0.4008 & 5.2491 & -1.7387 & -0.6306 \\ 2.4775 & 1.2674 & -5.9265 & 3.6620 \\ 1.1028 & -3.4549 & -1.0396 & 1.0022 \\ -0.0193 & 2.5751 & -0.1992 & 1.7522 \\ -1.6216 & 10.8901 & -5.4604 & 1.6031 \\ 0.3533 & 1.3310 & 1.8576 & 1.8519 \\ 0.1593 & -1.9292 & -1.5545 & -1.6702 \\ -2.6817 & 0.0471 & -3.6887 & 2.7453 \\ 1.8108 & 1.2412 & 0.5828 & 3.1422 \\ 2.0003 & 0.4178 & 1.1292 & 2.3832 \\ -1.4901 & -1.9743 & -1.5772 & 1.3330 \\ 2.4433 & -0.6506 & -2.2383 & 0.6013 \\ -1.4297 & 1.9519 & -2.2434 & -2.8982 \\ 1.4086 & -0.4976 & -1.4637 & 1.7572 \\ -2.2684 & 1.0550 & -0.0348 & -1.2236 \\ -0.4015 & -1.4813 & 2.3975 & 1.0213 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 2.9874 \\ 2.2781 \\ -1.8577 \\ -2.0270 \\ 1.7542 \\ -2.1966 \\ -1.3947 \\ -0.5767 \\ 7.2189 \\ -0.1904 \\ 0.2763 \\ -0.7297 \\ 0.9794 \\ 0.9240 \\ -1.6692 \\ 2.4811 \\ -2.1845 \\ 2.4781 \\ -2.3900 \\ -2.9181 \end{bmatrix}$$

Whiles the weights in the output layer is 1 x 20 matrix and a 1x1 Bias matrix

$$W^{output} = \begin{bmatrix} -0.1128 \\ 0.2166 \\ 2.2895 \\ 0.7863 \\ 2.1300 \\ -1.2140 \\ 2.3632 \\ 1.7749 \\ 7.4032 \\ -0.1247 \\ 0.1647 \\ -3.3143 \\ 0.8515 \\ -0.9615 \\ 0.3166 \\ 0.5613 \\ -1.4454 \\ 0.1219 \\ 0.8168 \\ 0.7556 \end{bmatrix}^T \qquad B_2 = \begin{bmatrix} 0.5011 \end{bmatrix}$$

The model for the network can be represented as

$$y^h{}_j = \tanh\left[(W^{hidden} X_j + B_1)\right]$$
$$y_j = W^{output} y_j^h + B_2$$

Where X_j denotes the Input vector (T-bills, Money Supply, CPI and Inflation), B is the bias, y_j^h is the output resulting from the hidden layer and y_j is the predicted output (Exchange Rate). In actual sense a matlab script has been generated to perform these computations much easier.

4.6 Model Comparison with the Multiple Regression Model

A multiple regression was performed on the data with the aim of comparing the results with the Feedforward Network model obtained.

The predictor variables were chosen in the same order as the ANN's input.

$$x_1 = Treasury\ Bills$$
$$x_2 = Money\ Supply$$
$$x_3 = CPI$$
$$x_4 = Inflation$$

With the output $y = Exchange$ Rates

The model obtained from the Multiple Regression Model is

$$y = 1.0241 - 0.0256x_1 + 2.0994e^{-4}x_2 - 3.5254e^{-6}x_3 + 0.0182x_4$$

The table below compares the performance measure of the Regression Model with the Artificial Neural Network(ANN) Model.

	Multiple Regression Model	ANN Model
MSE	0.0598	0.0010
RMSE	0.2446	0.0324
WAPE	23.21%	2.30%
Prediction Accuracy	76.79%	97.70%
R- value	0.7581	0.99634

Table 4.4: Models Comparison

From Table 4.4 it is obvious and conclusive that the ANN model performs better than the Multiple Regression model.

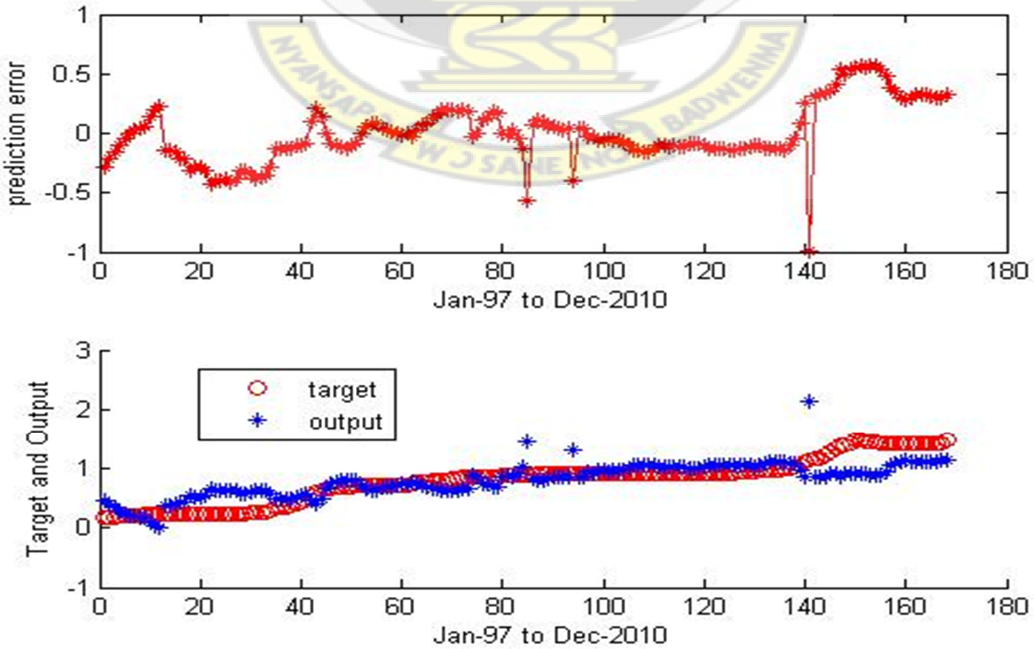


Figure4.8: Subplot of Prediction Error and Target – Output trend of the Multiple Regression Model

CHAPTER 5

CONCLUSION

5.1 Overview

This is the conclusive chapter of the research, the chapter maps the findings to the objectives of the study, proposes future researches that can be carried out on the study as well as recommendations on effective use of Artificial Neural Network.

5.2 Summary of Findings and Conclusion

The pertinent view, in economic literature, that exchange rates follow a random walk, has been dismissed by recent empirical work. There is now strong evidence that exchange rate returns are not independent of past changes. This is because a review of various practical applications and papers written on applications of neural network in financial market succeeds by producing very significantly close predictions. It has been realized that one of the most important features of a neural network is its ability to adapt to new environments. There are many different types of Neural Networks, each of which has different strengths particular to their applications. Our application of Neural Network in Financial uses a feedforward network with back propagation to forecast the Ghana Cedis – US Dolar rate.

It is natural and informative to judge forecasts by their accuracy. However, actual and forecasted values will differ, even for very good forecasts. In our forecast, the predictions gave results which are considered as very significantly close marginal differences. The forecast values of the US Dollar exchange rate produced differences in the range of 0.000051 and 0.2162. This result is obtained from several tests performed by using different input based on past data from Bank of Ghana, the T-bill rate, money supply, CPI and Inflation. The ANN model was compared with a Multiple Regression model developed. Upon comparison the mean squared error of the predicted values of the ANN model is 0.0010 whereas that of the regression model was 0.0598. The ANN model was found to be 97.70% accurate with the accuracy of the Regression model being 76.79%. Hence it is conclusive that the ANN model outperforms the Regression Model.

It can therefore, be concluded that application of neural network – feed forward neural network Levenberg Marquardt Backpropagation with 20 neurons in financial market is a good model for forecasting Ghana's Exchange Rates.

5.3 Recommendations

The application of Neural Network in Financial Market is very broad. Regardless of the fact that a lot has been done in this area by many researchers, it is no means exhaustive. I will therefore recommend that

- Young researchers should focus on ANN prediction approach since it can be used in various areas to produce an increase in forecast accuracy.
- There is no literature so far on the maximum choice of the number of neurons hence it's a prospective research area.
- It may be worthwhile to extend the predictor variables of the exchange rate to include Micro Economic Indicators to improve the accuracy of the forecast.

5.4 References

Adewole A. Philip, Akinwale A. Taofiki, Akintomide A. Bidemi (2011). Artificial Neural Network Model for Forecasting Foreign Exchange Rate. http://www.wcsit.org/pub/2011/april/Artificial%20Neural%20Network%20Model%20for%20Forecasting%20Foreign%20Exchange%20Rate.pdf Accessed May 14,2012

Bjørnland, H. and Hungnes, H. (2006). The importance of interest rates for forecasting exchange rate. Journal of Forecasting, 25:209-221.

Cheong, C., Kim, Y.-J. And Yoon, S.-M. 2011. Can we predict exchange rate movements at short horizons? Journal of Forecasting. doi: 10.1002/for.1236.

Chen, An-Sing, Mark T. Leung, and Hazem Daouk (2003), Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan stock index, *Computers and Operations Research* 30, 901-923.

Deans, M. 2010. Will consumer confidence boost help pound sterling? Economy news, Moneycorp. http://www.economy-news.co.uk/index.php Accessed 2012 November 15.

Diebold F. X. and Kilian L (2000), “Unit root tests are useful for selecting forecast model”. Journal of Buisiness and Economic statistics,18,265 – 273.

European Commission Directorate-General for Economic and Financial Affairs. 2009. The joint Harmonized EU programme of business and consumer surveys. http://ec.europa.eu/economy_finance/db_indicators/surveys/index_en.htm. Accessed 2012 May 22.

Edwards T., Tansley D.S.W., Frank R. J., Davey N (1997)., Traffic Trends Analysis using Neural Networks; Northern Telecom (Nortel) Limited Harlow, Essex, UK, CM17 9NA, Faculty of Information Sciences University of Hertfordshire Hatfield, Herts., UK, AL10 9AB.

Frank R. J., Davey N., Hunt S. P.(2001), Time Series Prediction and Neural Networks; Department of Computer Science, University of Hertfordshire, Hatfield, UK.

Harris, R. and Yilmaz, F. 2008. A momentum trading strategy based on the low frequency component of the exchange rate. Paper number: 08/04, University of Exeter, XFi Centre for Finance and Investment, Exeter, UK

Hill, C. 2004. International Business: Competing in the Global Marketplace. Boston, Massachusetts: McGraw-Hill College

Johnston, K. And Scott, E. 1997. Exchange rate probability distributions and fundamental variables. Journal of Financial and Strategic Decisions, 10: 27-36.

Jingao Y. and Hean-Lee P (1995)., Neural Network and Equity Forecasting, National University of Singapore

Kenneth A. Froot and Richard H. Thaler (1990). Anomalies; Foreign Exchange.

Khor, H. and Rojas-Suarez. L. 1991. Interest rates in Mexico: The role of exchange rate expectations and international creditworthiness. IMF Working Paper. http://www.cgdev.org/doc/experts/Int_Rates_in_Mex.pdf Accessed 2012 November 12.

Kuryati Kipli, Mohd Saufee Muhammed, Sh. Masniah Wan Masra,Nurdiani Z., Dayang A. (2012). Performance of Levenberg – Marquardt Backpropagation for full reference Hybrid Image Quality Metrics.

Lapedes A. and Farber R. (1987). Nonlinear signal processing using neural networks.

Lento, C and Gradojevic, N. 2007. The profitability of technical trading rules: a combined signal approach. Journal of Applied Business Research, 23:13-28.

Lowery, A. 2008. What makes currencies strong or weak? This is money. http://www.thisismoney.co.uk/money/markets/article-1619151/What-makescurrencies-strong-or-weak.html Accessed 2012 November 12.

Lawrence, C. 2011. Sterling drops after consumer confidence slump. http://www.exchangerates.org.uk/news/215/sterling-slumps.html. Accessed 2012 November 13.

Mehrotra, A. and Rautava, J. 2007. Do sentiment indicators help to assess and predict actual developments of the Chinese economy? Bank of Finland, Institute for Economies in Transition.

Meese, RA. and Rogoff, K. 1983. Empirical exchange rate models of the seventies: do they fit out of sample? Journal of International Economics, 14 (February): 3-24.

Neely, C and Weller, P. 2011. Technical analysis in the foreign exchange market. Working Paper 2011-001B, Working Paper Series, Research Division, Federal Reserve Bank of St.Louis.

Neely, C., Weller, P. And Dittmar, R. 1997. Is technical analysis in the foreign exchange market profitable? A genetic programming approach. Working Paper 96-006C, Working Paper Series, Research Division, Federal Reserve Bank of St.Louis.

Nilsson, H. and Tapasanun,W. 2009. Dollar and yen fall as risk sentiment improves. http://www.globicus.com/index.php?option=com_content&view=article&id=69:dollarand-yen-fall-as-risk-sentiment-improves&catid=38:globescenarios&Itemid=63. Accessed 2012 November 14.

Rodriguez, D. 2009. Forex trading forecast: US Dollar to weaken before further rallies http://www.dailyfx.com/story/strategy_pieces/fxcm_speculative_sentiment_index/Forex_Trading_Forecast__US_Dollar_1219327504086.html. Accessed 2012 November 14

Rosenberg, M. 2003. Exchange Rate Determination: Models and Strategies for Exchange Rate Forecasting. New York, New York: McGraw-Hill Library of Investment and Finance.

Skabar A. & Cloete I.(2002), Neural Networks, Financial Trading and the Efficient Markets Hypothesis.

Tsay, R. 2005. Analysis of financial time series. Hoboken, New Jersey: John Wiley & Sons

http://aquila.iseg.utl.pt/aquila/getFile.do?method=getFile&fileId=260090&contentContextPath_PATH=/unidade/ss/lateral/previous-editions/2012---13th-edition/downloads&_request_checksum_=420b0ef8768a47931a88ce1dcd4d3685212ea898 Accessed March 7, 2012

Tenti P. (1996), Forecasting Foreign Exchange Rates Using Recurrent Neural Networks, Tenti Financial Management s.a., Lugano, Switzerland.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.9855&rep=rep1&type=pdf

Accessed March 12, 2012

Vincenzo Pacelli (2010), Intelligent learning systems in Banking and Finance.

*http://*www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/jilsa.2011

Accessed Oct 4, 2012

Veeraachai Gosasang, Watcharavee Chandraprakaikul, Supaporn Kiattisin (2012). A Comparison of Statistical technique and neural networks forecasting techniques for container throughput in Thailand.

http://www.icms.polyu.edu.hk/ifspa**2012**/Papers/L24.pdf Accessed Dec 12, 2012

Walczak, S.; Tahai, A.; and Karim, K (1998). Improved cash flows using neural network models for forecasting foreign exchange rates. In P. Siegel, K. Omer, A. deKorvin, and A. Zebda (eds.), Applications of Fuzzy Sets and the Theory of Evidence to Accounting II. Stamford, CN: JAI Press, pp. 293–310.

Wang, J. 2008. Why are exchange rates is difficult to predict? Economic Letter— Insights from the Federal Reserve Bank of Dallas, 3.

http://www.summer.yale.edu/sites/default/files/ECON%20S476_1. Accessed Nov 6,2012

Wei Cheng, Lorry Wagner, Chien0hua Lin (1996), Forecasting the 30 – year U.S Treasury Bond with a system of Neural Networks.

http://www. citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.1844

Accessed Nov 6,2012

5.5 Appendix

Table 5.1: Predicted values of the ANN model.

0.1694	0.2275	0.4325	0.7146	0.879	0.9445	0.9279	1.1367	1.4003
0.1944	0.3002	0.5352	0.7289	0.8792	0.9266	0.9252	1.1899	1.4096
0.1847	0.2521	0.6061	0.7215	0.8792	0.9091	0.9223	1.1861	1.4351
0.1904	0.2485	0.6185	0.7286	0.8727	0.9025	0.9234	1.1773	1.4165
0.2014	0.2496	0.6546	0.7358	0.8865	0.902	0.9281	1.2787	1.406
0.2089	0.2963	0.6647	0.7507	0.8963	0.9048	0.9297	1.266	1.429
0.2196	0.2618	0.6752	0.7732	0.9161	0.9068	0.9299	1.5994	1.4638
0.2206	0.2432	0.6895	0.8434	0.8996	0.9052	0.9325	1.4166	1.476
0.2208	0.2276	0.6971	0.865	0.8734	0.906	0.934	1.4027	
0.2216	0.224	0.6912	0.8674	0.8794	0.9087	0.9548	1.4721	
0.2253	0.3134	0.716	0.8605	0.8874	0.9115	0.9575	1.4862	
0.2223	0.2922	0.7028	0.8623	0.8925	0.91	0.9616	1.4867	
0.2209	0.2885	0.713	0.8568	0.8936	0.9131	0.9762	1.4557	
0.2416	0.2544	0.7233	0.8667	0.9044	0.9162	0.9719	1.4484	
0.2414	0.3186	0.7189	0.857	0.9127	0.9162	0.9755	1.4314	
0.2463	0.3526	0.7152	0.8508	0.9283	0.9168	0.987	1.4216	
0.2505	0.3921	0.6819	0.8568	0.9318	0.9223	1.0062	1.5495	
0.218	0.3801	0.654	0.9131	0.9262	0.9258	0.9682	1.5385	
0.2671	0.3778	0.7076	0.9126	0.9565	0.9308	0.9287	1.4336	
0.2728	0.4021	0.7343	0.8677	0.951	0.9347	1.117	1.4273	

Table 5.2: Predicted values of the Regression Model

0.4564	0.5466	0.5608	0.7293	0.8836	0.9471	1.0573	2.1263	1.1194
0.3972	0.6567	0.4551	0.7824	0.8511	0.9648	1.0543	0.8495	1.1006
0.3382	0.638	0.3994	0.7331	0.9217	0.965	1.0522	0.8564	1.1093
0.2886	0.6333	0.4942	0.7084	1.0153	0.9827	1.0588	0.8719	1.1104
0.2478	0.6257	0.65	0.6975	1.4511	1.0122	1.0693	0.9228	1.1154
0.2178	0.645	0.7477	0.659	0.8191	1.0453	1.0664	0.9251	1.1266
0.1955	0.6207	0.7839	0.6482	0.7912	1.0557	1.0515	0.85	1.1415
0.1784	0.5604	0.8031	0.6187	0.8115	1.0685	1.0598	0.9131	1.1488
0.1672	0.5617	0.811	0.618	0.8403	1.0665	1.0582	0.9037	
0.0806	0.5944	0.7969	0.6215	0.8528	1.0446	1.0374	0.9194	
0.0309	0.6375	0.7988	0.6379	0.8637	1.0188	1.0673	0.9147	
-0.0047	0.6244	0.7136	0.6497	0.8697	1.0143	1.0975	0.9038	
0.3708	0.6201	0.6623	0.6673	0.8649	1.0299	1.101	0.8781	
0.3775	0.5826	0.6335	0.8902	1.3018	1.0237	1.1071	0.8741	
0.3902	0.4755	0.6261	0.8614	0.8624	1.0353	1.1113	0.8873	
0.441	0.4792	0.6741	0.7576	0.8561	1.0401	1.1249	0.9427	
0.4375	0.4888	0.6868	0.741	0.937	1.0075	1.0988	1.0447	
0.542	0.4997	0.6953	0.6927	0.9275	1.0089	1.0723	1.078	
0.5134	0.5125	0.718	0.6951	0.9702	1.0081	0.9799	1.1187	
0.513	0.5469	0.741	0.8713	0.9597	1.0166	0.8603	1.1288	