The Pennsylvania State University

The Graduate School

Department of Compute Science and Engineering

USE OF NEURAL NETWORKS IN FINANCIAL TRADING AND ASSET ALLOCATION

A Thesis in

Computer Science and Engineering

by

Manu Parbhakar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2008

The thesis of Manu Parbhakar was reviewed and approved* by the following:

Yuan Xie
Assistant Professor of Computer Science and Engineering
Thesis Advisor

Sencun Zhu
Assistant Professor of Computer Science and Engineering

Chao-Hsien Chu
Professor of Information Sciences and Technology

Mahmut Kandemir
Associate Professor of Computer Science and Engineering
Chair of Graduate Program

*Signatures are on file in the Graduate School

ABSTRACT

Recent research has been focused in using neural networks for developing trading strategies and also for tactical financial asset allocation. These strategies for both tactical asset allocation and financial trading in various time horizons are based on attributes of the historic time series data – volatility (std. deviation), returns, mean etc.

Further some of these trading strategies use technical indicators as inputs into the neural networks. Technical indicators are based on price action of the financial assets – like the opening, closing, high, low and volumes of any particular security. These indictors tend to capture the sentiment quotient in the financial markets and make profits from capturing that.

My paper uses a number of different technical indicators to develop trading strategies and conducts comparative analysis between various neural network architectures and technical indictors to see the best fit for trading market indices and individual stocks. Further the paper uses neural networks to do tactical asset allocation. Based on the historic correlation between the assets the neural network is trained to predict when to enter or exit a particular asset class. This decision is made periodically on monthly basis. Different methodologies are evaluated to find the optimal parameter to measure the accuracy of the neural model. Finally using paper money P/L is calculated in both strategies and evaluated against a default "buy and hold" strategy

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgment

I would like to thank Dr Xie, Dr Zhu, Dr Chu and Dr Haushalter for being part of my Master's defense academic committee.

Working for more than a year and a half with Dr Haushalter has been an enriching experience. He mentored me through this time and exposed me to various aspects of finance. In his lucid and gentle demeanor he has been an amazing mentor and guide.

Dr Xie has been steadfast in his support for my endeavors throughout my academic career here at Penn State.

Dr Zhu and Dr Chu, thanks you for agreeing to be part of my thesis defense committee.

I would not do justice if I don't thank Dr Tim Simins for the course he taught in Portfolio Management which initialized me on this path back in spring, 2007.I also thank Dr Lee Giles who encouraged me to combine my interests in computer science with that of finance.

It's been a rich and edifying experience I want to thank you all for making this possible.

Chapter 1

Introduction

Investment banks and hedge fund firms are using neural networks to uncover relationships in the markets and exploit them with real time trades. The idea is to strip human emotions such as fear and greed out of investing. Today computer-guided trading has reached levels undreamt of a decade ago. A third of all U.S. stock trades in 2006 were driven by automatic programs, or algorithms

The neural networks mainly find use in prediction in the financial sector. Since prediction is about finding relations between historic data and future prices or returns and neural networks can fit any kind of function and are shown to be universal approximators- they fit the bill perfectly.

This paper deals in two critical aspects of the market structure. The first one is tactical asset allocation -which is the distribution of capital among different asset classes such as stocks, bonds, money market, options, commodities, real estate and future contract. The importance of working with different asset classes has been researched in various academic papers. Assets have different characteristics and many of them are not correlated. It's this aspect that has been researched in [1], [2]. An investor benefits from asset allocation by lowering risk exposure through diversification of an investment portfolio [3], [4] among asset classes that may be uncorrelated or have a low correlation factor. In addition, portfolio volatility can be reduced. Risk in financial jargon is explained as std. deviation of the time series representing the financial asset

Tactical Asset Allocation is concerned with shorter periods of time, usually a week, month or quarter. This contrasts with Strategic asset allocation [5] that pertains with investment in a time horizon or a year or more. In tactical asset allocation decisions are made at the end of each time period whether to remain in the asset or exit.

The second aspect dealt in this paper is about devising a trading strategy. It derives inspiration from efforts to predict the market path in a small time window from the current data point. This has been a hot research topic for many years. [6], [7], [8], [9]. There are various thought in terms of the ability to profit from the equity market with prediction. The first school of thought believes that no investor can achieve above average trading advantage based on the historical and present information. This is advocated by major theories include the Random Walk Hypothesis and the Efficient Market Hypothesis [10]. The Random Walk Hypothesis [11] states that prices on the stock market wander in a purely random and unpredictable way. Each price change occurs without any influence by past prices. The Efficient Market Hypothesis states that the markets fully reflect all of the freely available information and prices are adjusted fully and immediately once new information becomes available. If this is true then there should not be any benefit from prediction, because the market will react and compensate for any action made from this available information.

The second school of thought to whom this paper belongs is that there are silos of inefficiencies in the market. In the actual market, the information flow is hierarchical. Some participants acquire information immediately while others later. Information flow is asymmetric has been proved in [12], [13]. The late comers react to market trends already established. Eventually however due to the efficiency of the markets, the returns follow a random walk. Further humans are influenced by emotions [16] and are not always logical e.g. we have a tendency to stick to our loosing positions and less likely to book profit from our winners. Taylor [17], Shiller [18], [19] provides compelling evidence against the random walk hypothesis and lay credence to the belief that the human emotions have considerable influence on the markets and thus offers encouragement for research into better market prediction.

After you have come across the notion that financial prediction can happen there are two big schools of thoughts on what factors to base prediction on. There are two schools of thoughts when you talk about financial73

prediction. The first ones called the “Fundamental Analysts”-a method of evaluating a security by attempting to measure its intrinsic value by examining related economic, financial and other qualitative and quantitative factors [20]. Fundamental analysts attempt to study everything that can affect the security's value, including macroeconomic factors (like the overall economy and industry conditions) and individually specific factors (like the financial condition and management of companies).

The other school is of technical analysts. Here securities are evaluated by analyzing statistics generated by market activity, such as past prices and volume. Technical analysts [21], [22], [23] do not attempt to measure a security's intrinsic value, but instead use charts and other tools to identify patterns that can suggest future activity. These rules are the result of years of experience and practical knowledge of the stock holders and brokers. Sayings such as “When the 10 day moving average crosses above the 30 day moving average and both averages are in upward direction it is time to buy” or “When the 10 day moving average crosses below the 30 day moving average and both moving averages are directed in the downward direction, it is the time to sell”, etc are applied widely on the trading floors on a daily basis.

Both technical and fundamental analysis lead to the fundamental point of identifying securities’ current price in hopes of figuring out what sort of position to take in that security. If the analysis suggests that the asset is under priced, then it’s a buying opportunity. Otherwise if the analysis suggests that the asset is over priced then it’s a selling/shorting opportunity.

The Artificial Neural Networks (ANNs) which are a connection of simple processing elements. Every connection of neural network has a weight attached to it [24]. The backpropagation algorithm has emerged as one of the most widely used learning procedures for multi-layer networks. The typical backpropagation algorithm has an input layer, some hidden layers and an output layer. The units in the network are connected in feedforward manner, from the input layer to the output layer. The weights of the connections have been given an initial value. The error between the predicted output value and the actual value is backpropagated through the network for updating the weights. This is called supervised learning procedure that attempts to minimize the error between the desired and the predicted outputs.

In this paper the neural networks have been used for predicting returns of assets into the future. As explained above this is based on the belief, today's prices have a bearing on those in the future and that the neural networks can simulate almost all non linear functions. This has led them to be used in the financial industry where they have been used widely to predict stock and bond prices -mapping historic data and trends to future price movements.

Past relationships are discovered through study and observation [25]. The basic idea of forecasting is to find an approximation of mapping between the input and output data in order to discover the implicit rules governing the observed movements. For instance, the forecasting of stock prices can be described in this way. Assume that I_u represents today's price, I_v represents the price after ten days. If the prediction of a stock price after ten days could be obtained using today's stock price, then there should be a functional mapping to I_v, where $I_v = \Gamma_i(I_u)$. Using all (I_u, I_v) pairs of historical data, a general function $\Gamma()$ which consists of $\Gamma_i()$ could be obtained, that is $v = \Gamma(u)$. More generally, $u^{->}$ which consists of more information in today's price could be used in function $\Gamma()$. As Neural Networks (NNs) are universal approximators, we can find a NN simulating this $\Gamma()$ function. The trained network is then used to predict the movements for the NN based financial forecasting has been explored for about a decade.

Many research papers are published on various international journals and conferences proceedings. Some companies and institutions are also claiming or marketing the so called advanced forecasting tools or models. Some research results of financial forecasting found in references for instance, stock trading system, stock forecasting [26, 27, 29], foreign exchange rates forecasting [28], option prices [30], advertising and sales volumes .

1.1 Contributions of the paper

One of the novelties of this paper is the depth of the data. Sourcing data is one of the tougher jobs in any computational finance paper and many of the research papers lacks the depth and the data points required for proper training of the neural networks. For example, [1] uses 90 data points for training and 10 data points for testing. Compared to this we used different granularity of data for the three different experiments. For experimenting with individual securities, I used (Google here).Listed in 2004 on the NASDAQ; the results are based on around 1200 data points. For defining trading strategies for the indices the data is based from 1981for the Dow Jones Industrial Average (DJIA) and since inception in 1999 for the NASDAQ. The data points are weekly and around 1700 in total. For the asset allocation, I used more than 4000 weekly data points from 1980 for the Dollar Index, the S&P 500 and the 1 year Treasury yields.

Also the data is from some of the most liquid stock exchanges in the world like the NYSE, NASDAQ. This allows for a more robust treatment as compared to research done in much smaller and less liquid exchanges like those from the developing countries KLCI [26], [31], and [32]

Secondly in this paper we analyze the entire spectrum of technical indicators right from the trend catchers to trend followers. There is a more thorough analysis as compared with other papers [33] that uses just a few selective technical indictors or [34], [35] where only a single indicator like the "time lag" and "bull flag" is used as inputs for price prediction by neural networks

Thirdly, this paper does a more practical analysis of the asset allocation. Returns in financial markets are always qualified by their associated risk. This contrasts from [1] where only the S&P 500 is chosen as the benchmark. Instead we have used an optimized portfolio of the three assets based on the industry wide accepted Modern Portfolio Theory (MPT) as the benchmark. This approach is more robust and more in line with industry practices

So this paper attempts to

1. Design a neural network based agent to trade securities/indices based on a combination of different technical indicators.

2. Evaluate the technical indicators and neural network architectures best suited for trading individual assets and indices

3. Design a neural network based agent to do tactical allocation between three assets- the S&P 500, the Dollar Index and the one-year Treasury constant maturity index.

Chapter 2

Technical Indicators and Data Preprocessing

There are three major steps in the neural network based forecasting proposed in research: preprocessing, architecture and post processing. In preprocessing information that could be used as inputs and outputs of the neural networks is collected. These data are first normalized or scaled in order to reduce the fluctuations and noise. In architecture, a variety of neural network models that could be used to capture the relationships between the inputs and the outputs are built. Different models and configurations using different training, validation and forecasting data sets are experimented. The best models are then selected for use in forecasting based on such measures as out-of-sample hit rates. Sensitivity analysis is then used to find the most influential variables fed to the neural networks. Finally, in post processing, different trading strategies are applied to the forecasting results to maximize the capacity of the neural network prediction.

We first discuss how the various technical indicators and other inputs were evaluated from the daily price and returns data of the financial assets. After analyzing inputs for trading in indices/stocks we see how the inputs were prepared for asset allocation.

2.1 Inputs for Trading Securities and Indices

Technical indicators are based out of various exchange related aspects of a financial asset like its price action, the volumes traded, the highs and the lows it charted over the observed phase. It attempts to use past stock price and volume information to predict future price movements. The technical analyst believes that there are recurring patterns in the market behavior that are predictable. They use such tools as charting patterns, technical indicators and specialized techniques like Gann lines, Elliot waves and Fibonacci series. They are further divided into the trend catchers (like moving averages, exponential moving averages, Momentum etc) and the trend changers like the (RSI, MACD Oscillators etc).Indicators are derived from price and trading volume time series. In most cases, there are five attributes of the time series representing a single share or market index. These five attributes of the time series are open price, close price, highest price, lowest price and trading volume. Analysts monitor changes of these numbers to decide their trading. An example of trading rule such as “When the 10-day moving average crosses above the 30-day moving average and both moving averages are in an upward direction it is the time to buy"; “When the 10-day moving average crosses below the 30-day moving average and both moving averages are directed downward it is time to sell", etc. are some of the rules used on the trading floor.

Other classification of Technical Indicators is whether they predict the continuation of a trend or a change in the trend i.e. tracing of inflexion points in the price behavior

The following technical indicators were selected as per our needs of technical analysis.

The RSI (Relative Strength Index) compares the magnitude of a stock's recent gains to the magnitude of its recent losses and turns that information into a number that ranges from 0 to 100. It takes a single parameter, the number of time periods to use in the calculation

RSI =100 – 100/1+RS

Average Gain= (Total Gains/n)

Average Loss= (Total Losses/n)

First RS = (Average Gain / Average Loss)

n = number of RSI periods

Moving Averages are one of the most popular and easy to use tools available to the technical analyst. They smooth a data series and make it easier to spot trends, something that is especially helpful in volatile markets. They also form the building blocks for many other technical indicators and overlays. A simple moving average is formed by computing the average (mean) price of a security over a specified number f periods.

Figure **2-1**

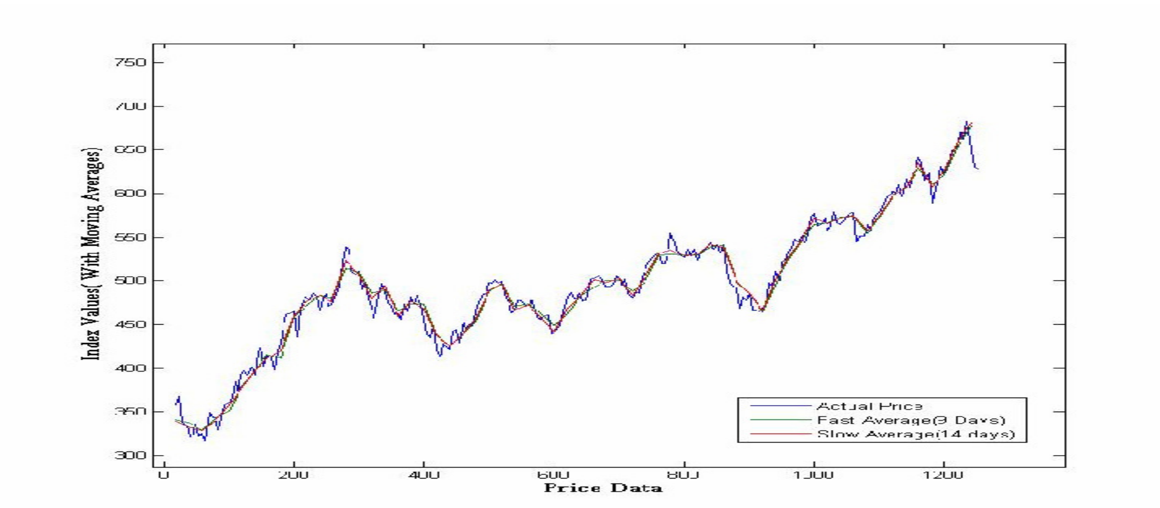

Figure **2-1**: Moving Averages

Exponential Moving Averages can be specified in two ways - as a percent-based EMA or as a period-based EMA. A percent-based EMA has a percentage as its single parameter while a period-based EMA has a parameter that represents the duration of the EMA.

$EMA_{current}$ = $Price_{current}$ - $EMA_{prev.}$ x Multiplier + EMA_{prev}.

While the multiplier is calculated as 2 / (Time periods + 1)

The **Moving Average Convergence/Divergence (MACD)** is one of the simplest and most reliable indicators available. MACD uses moving averages, which are lagging indicators, to include some trend-following characteristics. These lagging indicators are turned into a momentum oscillator by subtracting the longer moving average from the shorter moving average. The resulting plot forms a line that oscillates above and below zero, without any upper or lower limits

$$MACD = EMA_{12\ day} - EMA_{26\ day} / EMA_{26\ day}$$

EMA= Exponential moving average

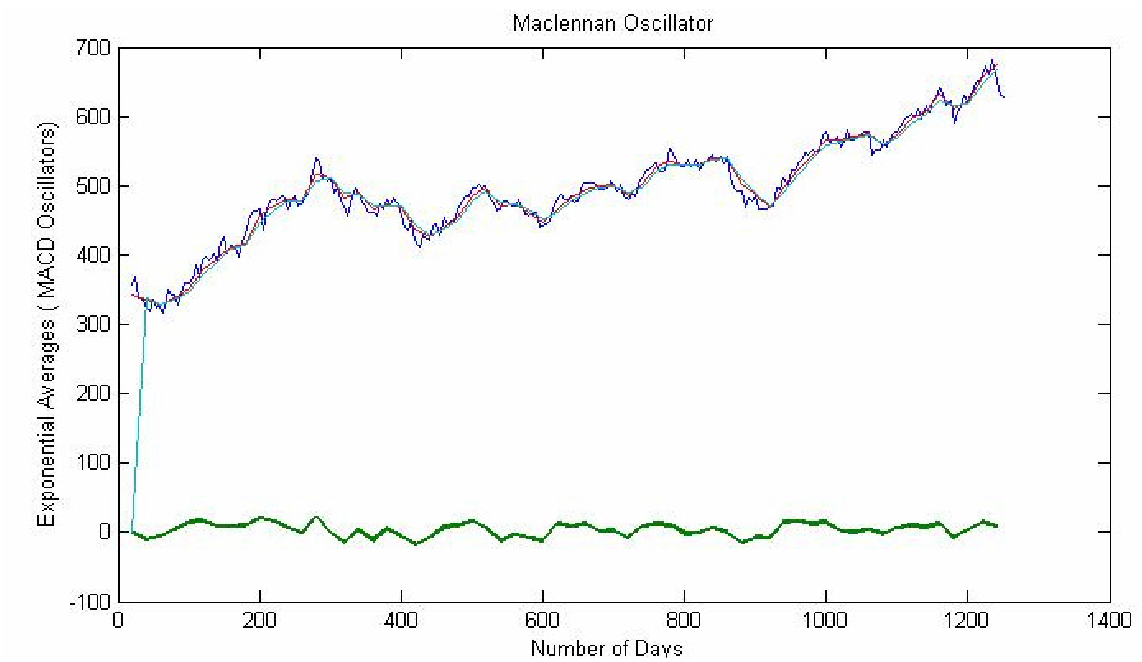

Figure **2-2**: Maclennan Oscillator

The **Stochastic Oscillator** is a momentum indicator that shows the location of the current close relative to the high/low range over a set number of periods. Closing levels that are consistently near the top of the range indicate accumulation (buying pressure) and those near the bottom of the range indicate distribution (selling pressure).

$\%k = CCP - L_9 / (H_9 - L_9)$

Where L_9 is the lowest low of the 9 days

H_9 is the highest high of the last 9 days

CCP is current closing point

Momentum is an another indicator bases on the price action and accounts for the market interest building in either direction

$M = CCP - OCP$

Where CCP is the current closing price

OCP is the old closing price for a predetermined period (5 days)

The **Moving Average of Stochastic** is a momentum or price velocity indicator. Buy signals are generally reinforced when the crossover occurs in the 10-15% ranges, and sells in the 85-90% range

$\%D = H_3/L_3\ 100$

Where H_3 is three day sum of $(CCP - L_9)$

L_3 is the three day sum of $(H_9 - L_9)$

Apart from these technical indicators the inputs to the neural networks are cuts and triggers that indicate whether to buy or the sell positions in the assets

Cut based on moving averages is when the fast moving average (9 bars) cuts the slow moving average (14 bars) from below (above) we get a buy (sell) signal

Figure **2-3**

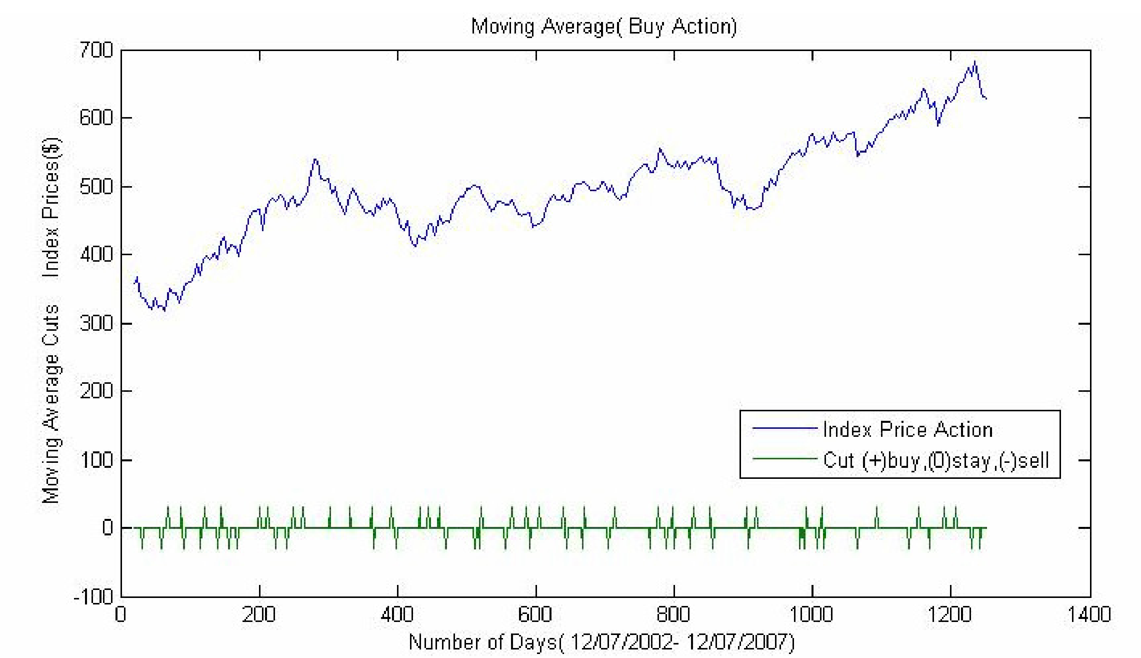

Figure **2-3**: Cut generated with moving averages

Triggers is when the oscillators vary from their normal behavior and move to the oversell/overbuy conditions. In stochastic oscillators Buy signals are generally reinforced when the crossover occurs in the 10-15% ranges, and sells in the 85-90% range. We have similar parameters for the RSI indicators

2. 2 Inputs for tactical asset allocation

For asset allocation we calculated the cumulative normal distribution. We wish to calculate the probability of asset A giving a higher return than asset B. Referring to paper [1] we can calculate this probability as a function of risk premium, mean risk premium and standard deviation of risk premium. Where risk premium is calculated as

Risk Premium of Asset A over asset B is given by

R(AB) = E(A)-E(B)

E(A) = expected return of asset A

E(B) = expected return of Asset B

From this we can calculate

$P(A>B) = CND\ [r_{AB}, U_{AB}, \sigma^2_{AB}]$

U_{AB} = Mean Risk Premium

σ^2_{AB} = Standard deviation of Risk Premium

CND = Cumulative Normal Distribution

Looking at historic data the P(A>B) gives us the probability of asset A delivering a higher return than asset B. We evaluate this probability at the end of each trading window and based on this probability we make allocation for the next window.

The TAA portfolio can then be represented by the vector:

TAA -> (X_S, X_B, X_T)

Where

X_S = Allocation to stocks

X_B = Allocation to bonds

X_T = Allocation to Dollar Index

After these probabilities have been calculated, they are fed as samples to the NN to train and on the input side we feed the excess returns between the three asset classes as inputs. i.e. the excess return of S&P 500 over Treasury, Excess returns of S&P500 over Dollar Index, and excess returns of Dollar Index over Treasury. These three combined together form the inputs to the neural networks.r

In the testing phase the neural network estimates its probability on each asset generating excess returns and this probability is compared with the original one found from the CND analysis.

Chapter 3

Neural Network Architecture and Input Data

A neural network is a collection of interconnected simple processing elements. Every connection of neural network has a weight attached to it. The back propagation algorithm has emerged as one of the most widely used learning procedures for multi-layer networks. The typical back propagation neural network usually has an input layer, some hidden layers and an output layer. Figure 3-2 shows a one-hidden layer neural network. The units in the network are connected in a feed forward manner, from the input layer to the output layer. The weights of connections have been given initial values. The error between the predicted output value and the actual value is back propagated through the network for the updating of the weights as shown in Figure 3-1.

This is a supervised learning procedure that attempts to minimize the error between the desired and the predicted outputs. The output value for a unit j is given by the following function:

$\theta_j=G(\sum W_{ij}X_i-\theta_j)$ where $G(z)=\tanh(z)=1-e^{-z}/1+e^{-z}$

G()is the activation function used for time series prediction

Figure **3-1**

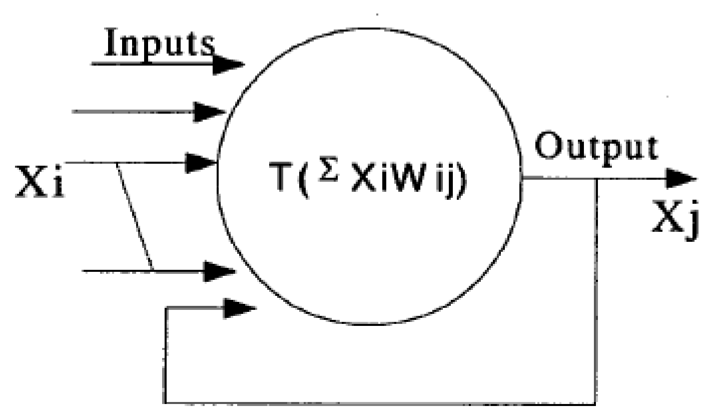

Figure **3-1**: Feedback and Back Propagation is central to NN Architecture

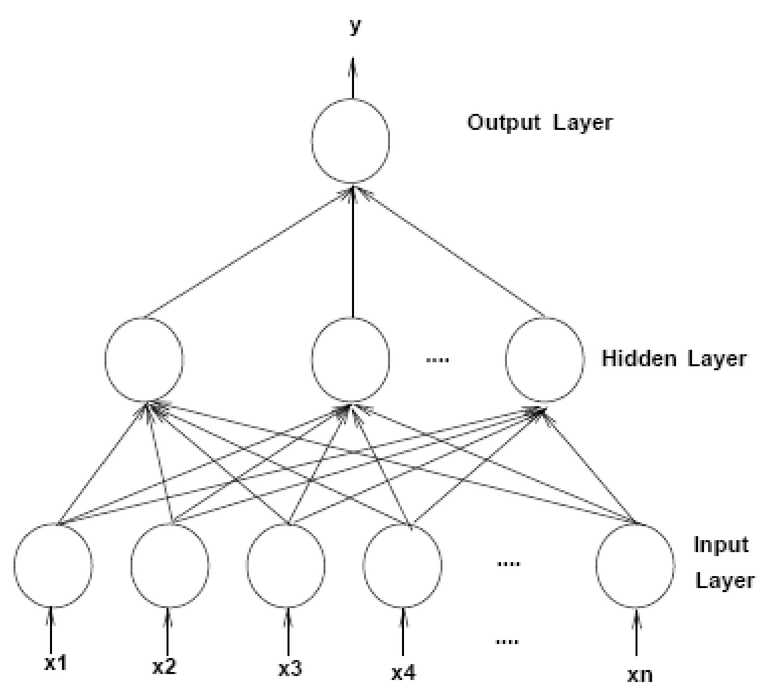

Figure **3-2**: Architecture of Neural Networks

Figure **3-2**

3.1 Training Process

Historical data is divided into three parts- training, validating and testing sets. Different splits of data into the three were analyzed. In the big picture a model is considered good if the error for out of sample testing is the lowest compared with other models. If the trained model is the best one for validation and also the best one for testing, one can assume that it is a good model for future forecasting.

The neural network is trained by training data set to find the general pattern of inputs and outputs. To avoid over fitting, the hold out validation set is used as cross-validation and the "best" model is picked up. This model is then used as a forecasting model applied to out-of-sample testing data set. The data are chosen and segregated in time order. In other words, data of earlier period is used for training, the data for later period are used for validation and the data for the latest period is used for testing. This method may have some recency problems.

The data feeding into the networks is as follows

Table **3-1**

Table **3-1**:Trading based on Technical Indictors(Stock and Index Trading)

Moving Averages (Fast)
Simple Return
Moving Average (Slow)
Cuts
Returns (Daily)
MACD Oscillator
Exponential Moving Averages/Other Osc.

Table **3-2**

Table **3-2**: Input Data for Tactical Asset allocation

S&P 500 Excess Returns
Dollar Index Excess
One-Year Treasury Constant Maturity Index Yields

In theory a Neural Network can fit any kind of function and data could be built. They have been shown to be universal approximators and their functions mathematically. The main consideration when building a suitable neural network for the financial application is to make a tradeoff between convergence and generalization. It is important not to have too many nodes in the hidden neural network, to learn by example only and not generalize

We played around with various architectures both considering

1. Number of Hidden Layers
2. Number of Neurons in each Hidden Layer

In our case through the use of customizable Neural Networks we were able to control various aspects of the technical analysis. We performed experiments with different neural network architecture in combination with various input parameters to fine tune the output to match the actual training data with greater accuracy.

3.2 Description of Data

Now here is a brief description of the various data indices. All the data points have the following five attributes open, close, high, low, volumes. The periodicity of the data was different for the two experiments. For trading in securities we used daily stock data like Google here. For trading in indices we used weekly data from 1993 for the DJIA and from inception in 1999 for the NASDAQ. Finally for asset allocation we used weekly data from 1980 to 2007.

The US Dollar Index (USDX) is an index or measure of the value of the United States dollar relative to a basket of foreign currencies. It is a weighted geometric mean of the dollar's value compared to the Euro (EUR), Japanese yen (JPY), Pound sterling (GBP), Canadian dollar (CAD), Swedish krona (SEK) and Swiss franc (CHF).

The S&P 500 is a stock market index containing the stocks of 500 Large-Cap corporations, most of which are American. All of the stocks in the index are those of large publicly held companies and trade on the two largest US stock markets, the New York Stock Exchange and NASDAQ. After the Dow Jones Industrial Average, the S&P 500 is the most widely watched index of large-cap US stocks. It is considered to be a bellwether for the US economy and is a component of the Index of Leading Indicators. It is often quoted using the symbol SPX or INX

The one-Year Treasury Constant Maturity index published by the Federal Reserve Board based on the average yield of a range of Treasury securities, all adjusted to the equivalent of a one-year maturity. Yields on Treasury securities at constant maturity are determined by the U.S. Treasury from the daily yield curve. That is based on the closing market-bid yields on actively traded Treasury securities in the over-the-counter market.

The NASDAQ is the largest electronic screen-based equity securities trading market in the United States. It lists more than 3200 companies and hosts most of the top technology firms in the world. It's a proxy for the health of the technology sector of the US.

Besides the data points described we have the daily price data on financial securities like Google.

All data was sourced from the either the Bloomberg Terminal or the Trade Station Platform in the Trading Room at the Smeal School of Business.

Chapter 4

Evaluation and Results

In this chapter we first discuss the parameters we selected to benchmark our results and the reasons for selecting them. In the second part we see we discuss the results for trading in individual securities (Google here) and then move on to results on trading in indices. In the third part of the chapter we look at our experiments in “Tactical Asset Allocation”.

4.1 Methods of Evaluation

In this section we discuss the various strategies that we could use to evaluate the success of the predictions from our Neural Network model. [26] Summarizes very lucidly the major evaluation method. I am listing the top three of them and then we look at why the “Grad” is the best choice in evaluating success of neural networks in financial trading

We first calculated the NMSE (Normalized Root mean Squared Error) which is given by

NMSE= $Sum^2_{Actual\ Value-\ Predicted\ Value}$/ Actual Value- $Mean^2_{Actual\ Data}$

Where

Actual value- is the value at a tick

Predicted Value- is the value predicted using the Neural Net at the same data point

The second evaluation methodology is that of Signs. This is given by

Sign = $\sum S_k / N$

Where

$S_k = 1 \quad X_k * \hat{X}_k > 0$

$= 1 \quad X_k = \hat{X}_k = 0$

$= 0$ otherwise

Here Sign represents the correctness of signs after normalization.

The third evaluation strategy that we will use and which can nicely quantify the directional changes can be expressed as Gradients

Grad= Sum (G_k)/N

Where

$G_k = 1$ if (Predicted $_{(t+1)}$-Actual $_{(t)}$ =0 && Actual $_{(t+1)}$-Actual $_{(t)}$ =0)

$= 1$ if (Actual $_{(t+1)}$-Actual $_{(t)}$ * Predicted $_{(t+1)}$-Actual $_{(t)}$ > 0)

$= 0$ otherwise

Now the NMSE seems as the natural choice in evaluating neural network prediction, however a prediction that closely follows the trend of the actual target would also result in a low NMSE.

Also in trading we are interested in if the predicted price path trends in the same way as the original rather than predicting absolute values. This is why we'll be choosing the "Grad" as an indicator of our performance.

4.2 Implementation Details

The entire tests for the works and experiments have been conducted majority with the MATLAB Neural Utility. As described in Chapter 2, the data set was divided into three portions for training, validation and then out of sample testing of the final prediction.

The primary processing on the data- that is converting the tick data into technical indicators or CND for asset allocation was done in MATLAB. After that we used the ANN library of MATLAB to run neural networks with the inputs that we designed.

We used different granularity of data for the three different experiments. For the first issue of trading in particular assets I chose the daily closing price of the asset (Google here).Listed in 2004 on the NASDAQ, the results are based on around 1200 data points. For defining trading strategies for the indices the data is based from 1981for the Dow Jones Industrial Average (DJIA) and since inception in 1999 for the NASDAQ. The data points are weekly and around 1700 data points in total. For the asset allocation, I have used weekly data points from 1980 for the Dollar Index, the S&P 500 and the 1 year Treasury yields.

4.3 Results for Trading in Individual Securities

Having chosen the Grad as the basis of evaluation, we used the Google daily price data and evaluated various architectures. The architectures varied in their number of hidden layers and number of neuron in each layer. The table below lists the results from these. The numbers in the parenthesis lists the number of neurons in each layer. The best accuracy was achieved with 20 neuron single layer architecture.

Table **4-1**

Table **4-1**:Prediction accuracy with different neural architecture

Architecture (Number of Layers)	1	2	3	4
Neurons in layers	(20) 76%	(20, 10) 68%	(20,10,10) 64%	(20,10,10,10) 72%
Neurons in layers	(10) 73%	(10, 10) 73%	(10,10,10) 73%	(10,10,10,10) 72%
Neurons in layers	(5) 73%	(10, 20) 69%	(10,10,20) 69%	(10,10,10,20) 72%

We also saw that changing the architecture has relatively less influence on the final results. The analysis of technical indicators is in the next section. For the analysis here we used the moving averages, the exponential moving averages, the RSI, MACD and the stochastic indicators. The figure below represents the actual (blue) and the predicted value (green) with1-hidden layer (20 Neuron) architecture

Figure **4-1**

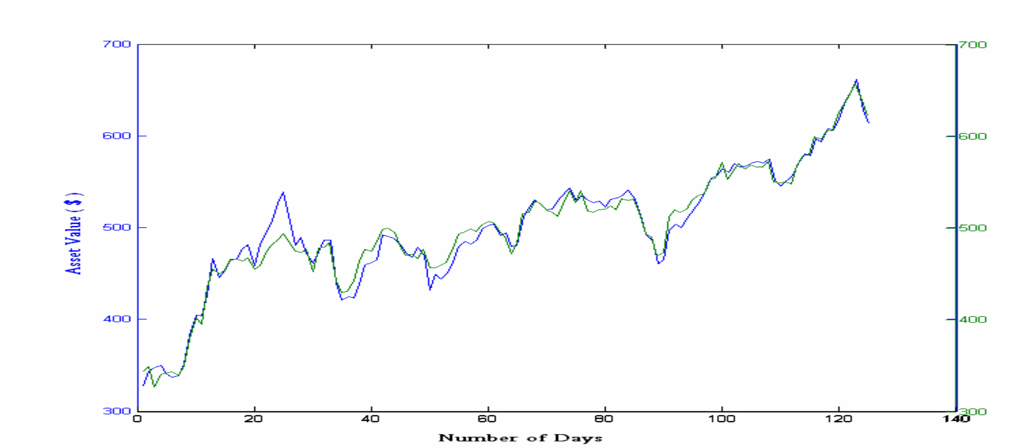

Figure **4-1**: Prediction on Google – (Green) Predicted Values, Blue(Actual Values)

As you can see we have a got a 76% amount of accuracy in predicting the trends. Now we will try to monetize this high accuracy albeit in paper money only. In the next section we see three different strategies and see how much money could have had made with this kind of accuracy.

4.3.1 Profit Loss using paper trading

We not try to monetize the high accuracy that we achieved in prediction. We evaluate the predicted model, by trading on them. Assume that a certain amount of seed money and we invest in the following three ways

There are three kinds of trading strategies that were evaluated

Strategy1: If ($Predicted_{t+1}$-$Predicted_{t}$>0) &If ($Predicted_{t}$-$Predicted_{t-1}$>0) then buy and similarly on the sell side

Strategy2: The generic "Buy and Hold"

Strategy3: Look ahead. Look ahead into the prediction window and sell at the highest point and buy at the lowest point

Discussion on Strategy 3

The neural network has been able to predict the values with an accuracy of 76%, up to a window of 7- 9 ticks in the future. Now depending on the input data these ticks can have different granularity from second tick to end of day /week. In our analysis we are looking at end of day data. So we can predict prices to an accuracy of around 76%, 7-9 days in advance. This is high number in the highly volatile environment of the stock markets

The idea is to inspect the predicted window and look for the lowest point and for the highest point. Sell your holdings at the highest point and buy it at the lowest point. For example in figure 4-2 in the first diagram you sell first and then buy later while in the second diagram you buy at the lowest point. This is better than trading at each time window as it avoids the compulsion of trading in local maxima and minima.

Figure **4-2**

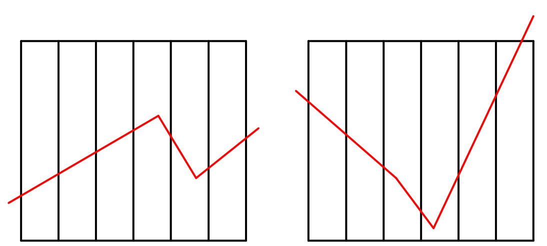

Figure **4-2**: Strategy 3 troughs and curves

We change the frequency of trade and indicate the requirement of trade only at the inflection points instead of the intervals, we will be able to gain much more theoretically.

How did we do-?

Below is the profit/loss analysis and returns that the paper trades generated. In the table we have a set of three trading strategies for three different levels of slippage. Even with taking a slippage factor of 2% per trade against the industry standard of the 0.5% we outperform the buy and hold. Here slippage refers to

Table **4-2** Table **4-2**: P/L using paper trading with various slippage

Return on Investment	Buy and Hold	Sell and Next Highest Bar	Look Ahead
Slippage Level (0.5% /trade)	2.23%	1.20%	5.67%
Slippage Levels (2%/trade)	2.23%	0.70%	4.43%
Slippage Levels (1%/trade)	2.23%	0.90%	5.20%

4.4 Results for Trading of Indices

In this section we list how we fared while trading in indices i.e. the NASDAQ and the DJIA. The focus in this section apart from the architecture will be on the selections of the inputs to the neural networks. The tables below list the performance of the neural network and the degree of accuracy achieved. An interesting finding of the study was that the introduction of

MACD Oscillators and RSI (Relative Strength Index) severely deteriorates the prediction quality of the system.

Table **4-3**

Table **4-3**: NASDAQ Weekly data from Dec, 1999 to June, 2008.Where FMA-Fast Moving Average (9ticks), SMA-Slow Moving Average (14ticks), RSI- Relative Strength Index, MACD Osc. - McClellan Oscillator, ExpAverage-Exponential Moving Average (9 ticks), Time Lag- Time series delayed by 7 data ticks

Input Parameters Number of Neurons	Time Lag(9), FMA(9), SMA(14)	Time Lag (9), FMA(9), SMA (14) & MACD Osc.	Time Lag(9), FA(9), SMA(14), ExpAverage(14)	Time Lag(9, FMA(9), SMA(14) & RSI	FMA(9),SMA(14) ExpAverage(14)	FMA(9), SMA(14), ExpAverage(14), Cut
10	13.0%	41.0%	41.0%	20.0%	71.0%	78.0%
20	76.08%	41.0%	80.4%	20.0%	73.0%	76.0%
50	73.00%	41.0%	69.0%	20.0%	71.0%	76.0%
100	80.42%	41.0%	76.0%	20.0%	76.0%	84.0%

MACD and RSI are discrete and indicate over bought or over sold states The possibility that discrete inputs were affecting the system adversely was abated when we got the best results by including Cut listed in the last column. The cut another discrete indicator has values of -1, 0 or 1 indicating sell, hold or buy signals.

The weekly data from inception of the NASDAQ in 1999 was tested till present and we got the best result with an accuracy of 84% with one layer 100 Neuron architecture. Next we tested the Dow Jones Industrial Average (DJIA) from 1993 to present and we got the highest accuracy of around 88%. The table for DJIA is listed below. Around 10 % of the data points are used for out of sample testing, i.e. approximately 2 years of monthly data

Table 4-4

Table **4-4** : Dow Jones Industrial Average from July, 1993 to June, 2008. Where FMA-Fast Moving Average (9ticks), SMA-Slow Moving Average (14ticks), RSI- Relative Strength Index, MACD Osc. - McClellan Oscillator, ExpAverage-Exponential Moving Average (9 ticks), Time Lag- Time series delayed by 7 data ticks

Input Parameters Number of Neurons	Time Lag(9), FMA(9), SMA(14)	Time Lag (9), FMA(9), SMA (14) & MACD Osc.	Time Lag(9), FMA(9), SMA(14), ExpAverage(14)	Time Lag(9, FMA(9), SMA(14) & RSI	FMA(9),SMA(14) ExpAverage(14)	FMA(9), SMA(14), ExpAverage(14), Cut
10	61.84%	30.0%	75.0%	15%	73%	80.26%
20	72.36%	30.0%	88%	15%	82%	84%
50	65%	30.0%	73%	14%	78.9%	69%
100	72.36%	30.0%	81.58%	11%	80.26%	82%

The results prove that the prediction of NN is generic and not restricted to a particular security. It gives high accuracy for both the technology index and the industrial index. We can run similar paper trading tests as in the earlier section and monetize this high accurate prediction.

4.5 Results on Tactical Asset Allocation

To test asset allocation between three different assets we use monthly data from Nov, 2007. After calculating standard deviations and means data from January, 1986 is fed into the neural network. The 257 monthly data points are divided into training, validation and testing data. Approximately 10% of this data from Nov, 2005 to Nov, 2007 is used for testing

We compare the network's performance against the benchmark which in this case we assume to be the returns of equally weighted portfolio of all three assets. In table 3, we highlight the monthly returns for the three different assets- the S&P 500, the Dollar Index and the 1year Treasury Yield in the testing period. The highlighted (yellow) cells in each row signify the asset class that gave the highest return for that particular month.

Table 4 shows the accumulated returns of the S&P 500 in the 1st two columns. The results of the neural network model are depicted in the 3rd and the 4th column. The 5th column (Asset Chosen) tells the asset class chosen by the neural network for a particular month.

The neural network invests on the basis of the Cumulative Normal Distribution. After being trained with the excess returns of the assets as inputs and the probability derived from the CND as outputs, the neural network predicts its own set of

Table **4-5**

Table **4-5**: Returns for the assets in the testing period

Dates	S&P Index	Dollar Index	1 Year Treasury
Nov,2005	3.5%	1.7%	-0.2%
Dec, 2005	-0.1%	-0.4%	-0.7%
Jan,2006	2.5%	-2.5%	2.7%
Feb,2006	0.0%	1.3%	3.8%
Mar,2006	1.1%	-0.4%	3.4%
Apr,2006	1.2%	-4.1%	1.1%
May,2006	-3.1%	-1.6%	3.3%
June,2006	0.0%	0.5%	1.8%
July,2006	0.5%	0.2%	-4.1%
Aug,2006	2.1%	-0.3%	-4.4%
Sept,2006	2.4%	1.1%	-1.9%
Oct,2006	3.1%	-0.8%	0.0%
Nov,2006	1.6%	-2.8%	-2.4%
Dec,2006	1.3%	0.8%	4.9%
Jan,2007	1.4%	1.1%	2.5%
Feb,2007	-2.2%	-1.2%	-6.1%
Mar,2007	1.0%	-0.8%	-0.7%
Apr,2007	4.2%	-1.8%	-0.1%
May,2007	3.2%	1.0%	7.3%
June,2007	-1.8%	-0.5%	0.3%
July,2007	-3.3%	-1.4%	-8.1%
Aug,2007	1.3%	0.0%	-8.3%
Sept,2007	3.5%	-3.9%	-3.1%
Oct,2007	1.5%	-1.6%	-2.5%
Nov,2007	-4.5%	-0.4%	-26.5%

If the asset chosen by the neural network is the same as that with the highest return for the particular month then that cell is painted green and the asset class with the highest return is listed

Table **4-6**

Table **4-6**: The Systems Performance vs Benchmark

S&P	Accumulated	ANN	Accum	Asset
3.5%	3.5%	3.46%	3.46%	S&P
-0.1%	3.4%	-0.69%	2.77%	Tres
2.5%	5.9%	2.51%	5.29%	S&P
0.0%	5.9%	3.80%	9.09%	Tres
1.1%	7.0%	1.10%	10.19%	S&P
1.2%	8.2%	1.21%	11.40%	S&P
-3.1%	5.1%	3.27%	14.67%	Tres
0.0%	5.1%	1.83%	16.50%	Tres
0.5%	5.6%	-4.12%	12.38%	Tres
2.1%	7.7%	2.11%	14.49%	S&P
2.4%	10.1%	2.43%	16.91%	S&P
3.1%	13.2%	3.10%	20.01%	S&P
1.6%	14.9%	1.63%	21.65%	S&P
1.3%	16.1%	4.87%	26.52%	Tres
1.4%	17.5%	2.46%	28.98%	Tres
-2.2%	15.3%	-2.21%	26.77%	S&P
1.0%	16.3%	0.99%	27.77%	S&P
4.2%	20.5%	-0.13%	27.63%	Tres
3.2%	23.8%	1.04%	28.68%	DYX
-1.8%	22.0%	-0.46%	28.21%	DYX
-3.3%	18.7%	-1.41%	26.80%	DYX
1.3%	20.0%	0.03%	26.83%	DYX
3.5%	23.5%	3.52%	30.34%	S&P
1.5%	25.0%	1.47%	31.81%	S&P
-4.5%	20.5%	-0.44%	31.38%	DYX

In case the asset chosen is not the same as with the highest return then the cell is painted red and the asset selected by the neural network is listed in the cell. In the testing period the neural network accurately predicted the highest return in 17 months out of the 24 months being tested on. This gave us an accuracy of 71%.

We compared the returns of the neural networks with three bench marks. The first one is the equally weighted portfolio of all three assets. The second one is the optimized portfolio derived from the modern portfolio theory. The last bench mark is the S&P 500

The accumulated returns from the neural network beat all the three benchmarks handsomely.

The ANN beat the MPT Optimized portfolio where we got weight of 18.3 % (S&P), 39.7% (DYX) and 58.48 %(Treasury) by upwards of 60 percent or more than 6000 Basis points

The ANN beat the equally weighted portfolio where each asset has a 33 % weight in it by more than 42 percent or 4200 basis points

The ANN beat the S&P by 1088 basis points. (1 percent =100 basis points). In more than seven occasions when the S&P gave below average returns the neural network moved into another asset class to deliver higher returns.

Chapter 5

Discussion and Future Work

The paper selects the technical indicators that are most suited to trading in financial securities index. It creates a system of technical indicators and neural networks that can efficiently trade indices and individual securities and generate consistent higher returns than the benchmark “buy and hold” policy. The paper proves this for the Dow Jones Industrial Average, the NASDAQ and the technology company Google. The second part of the paper provides a comprehensive proof that the neural networks can be used in asset allocation. They can efficient select/deselect financial assets and deliver higher returns than holding any particular individual asset class albeit with accompanying risk. The paper proves that between the S&P 500, the Dollar Index and the 1 year Treasury

In the future work on this project we can take the individual risk into consideration by allocating assets according to the risk appetite of an individual. Also VaR (Value at Risk) can be incorporated so that it will give a better picture of the risk undertaken both while trading and asset allocation in this paper.

The input data used for the system can be more discrete so that the trading window can be made flexible. I.e. we can use only tick data as input and then can combine these data points for various granularities (10 min, hour, week, month etc) and design trading strategies for various time windows. Have strategies in any time window to solve "when to enter" paradigm. This can be automated and depending on the most profitability trade in any window he desires. Also with the smaller granularity we can train the neural networks better.

Another scope of improvement in the results can be by use of genetic algorithms. In the experiments in the thesis the technical indicators used have been the generic industry wide technical parameters. It may help to design a new class of technical indicators with genetic algorithms more tailored to the asset class under. Also the genetic algorithms can help to combine the various class of technical indicators like the price trenders (moving averages); price oscillators (MACD, RSI); and volume oriented indicators (open interest).Besides that the genetic algorithms can also be used to select the optimal parameters in these technical indicators i.e. how long should be the window in the selection process be for a fast or slow moving average, 9 or 14 ticks etc

Different financial products can be used in the allocating assets from the ones used. Exposure to commodities, real estate and emerging markets, CDOs etc would be the next logical extension. Diversifying into alternate assets generally results in higher returns on lesser risk per capita. Acquiring good and high granular data from the developing countries is a challenge as is the getting real estate, CDO data. Most of this data is proprietary and with very limited public access.

The project can now be ported to a dynamic real time environment where it can be connected to exchange and make trades in real time. During offline hours it should be able to run optimization algorithms as in optimizing neural network architecture, selecting technical indicators and then making trades when the system comes online. So the next logical step is to design such an interface to work with APIs of financial broker firms. A first attempt was made to run the optimization code with APIs of Interactive Brokers but there were issues in retrieving the data to a database and then running optimization algorithms on them

References

[1] C. August Casas - Tactical Asset Allocation: An Artificial Neural Network Based Model IEEE Explore 2001

[2] Nicolas Chapados-Cost Functions and Model Combination for VaR-Based Asset Allocation Using Neural Networks IEEE Transactions on Neural Networks, VOL. 12, NO. 4, JULY 2001

[3] Gerard and De Santis -International Asset Pricing and Portfolio Diversification with Time-Varying Risk. Journal of Finance Dec 1997

[4] Klemkosky and Martin- The Effect of Market Risk on Portfolio Diversification. The Journal of Finance Mar., 1975

[5] Brennan, Schwartz and Lagnadod - Strategic Asset Allocation. Journal of Economics and Control June 1997

[6] Wolfers, Zitzewitz –Prediction Markets NBER Working Paper May, 2004

[7] Kyoung-jae Kim-Financial time series forecasting using support vector machines Elsevier March, 2003

[8] Daijin and Chulhyun- Forecasting time series with genetic fuzzy predictor ensemble Fuzzy Systems IEEE Transaction Nov, 1997

[9] Schreiber, Wolfers, Pennock and Brian Galebach- Prediction Markets: Does Money Matter? Electronic Markets Sept, 2004

[10] Malkiel-The Efficient Market Hypothesis and Its Critics Journal of Economics Winter, 2003

[11] Malkiel BG - A Random Walk Down Wall Street 2000

[12] Johnson and Acharya- Insider trading in credit derivatives. Elsevier May, 2006

[13] Robe and Fishe- The impact of illegal insider trading in dealer and specialist markets: evidence from a natural experiment. Journal of Finance Aug, 2003

[14]Markowitz H M. Portfolio selection: efficient diversification of investments. New York: Wiley, 1959.

[15]Solnik, B.H. Note on the Validity of the Random Walk for European Stock Prices, Journal of Finance, Dec 1973

[16]Lo and Repin - The Psychophysiology of Real-Time Financial Risk Processing, Journal of Cognitive Neuroscience April, 2002

[17] Taylor, S. - Modeling Financial Time Series, Wiley and Sons, 1986

[18] Shiller - From Efficient Market Theory to Behavioral Finance, SSRN 2002

[19] Vasiliou, Eriotis and Papathanasiou- Incorporating Technical Analysis into Behavioral Finance: A Field experiment in the Large Capitalization Firms of the Athens Stock Exchange EuroJournal, 2008

[20] Abarbanell and Bushee- Fundamental Analysis, Future Earnings, and Stock Prices Journal of Accounting Research Spring 1997

[21] Neftci- Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of "Technical Analysis". The Journal of Business, Oct 1991

[22] Jennings and Brown- On technical analysis. Review of Financial Studies, 1989

[23] LeBaron, Brock, Lakonishok- Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. The Journal of Finance Dec, 1992

[24] Hassoun, M.H. Fundamentals of Artificial Neural Networks, Proceedings of the IEEE, Jun 1996

[25] Yao and Tan- Guidelines for Financial Forecasting with Neural Networks, NUS 2002

[26] Poh, Yao and Tan- Neural Networks for Technical Analysis: A study for KLCI, International Journal of Theoretical and Applied Finance 1999

[27] Skabar & Cloet- Neural Networks, Financial Trading and the Efficient Markets Hypothesis, SIT Germany

[28] Poa, Jasic and Yao- Foreign Exchange Rates forecasting with Neural Networks, NUS, Singapore

[29] Manu and Talal, Use of Neural Networks in Financial Trading 2007

[30] Chris and Spiros- Option pricing and trading with artificial neural networks and advanced parametric models with implied parameters, IEEE Joint Conference on Neural Networks, 2004

[31] Pan, Yearwood- Predicting the Australian Stock Market Index Using Neural Networks Exploiting Dynamical Swings and Intermarket Influences. 2003

[32] Daouk and Chen, Application of neural networks to an emerging financial Market: forecasting and trading the Taiwan Stock Index, 2002

[33] N. W. Tan, Incorporating Artificial Neural Networks into a Rule-Based Financial Trading System, 1993

[34] Yao and Tan, A case study on using neural networks to perform technical forecasting of Forex, 2000

[35] Purvis and Leigh, Forecasting the NYSE composite index with technical Analysis, pattern recognizer, neural network, and genetic algorithm, 2001