

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228607056>

Using neural networks to forecast stock market prices

Article · January 1998

CITATIONS

167

READS

21,409

1 author:



[Ramon Lawrence](#)

University of British Columbia - Okanagan

108 PUBLICATIONS 1,401 CITATIONS

SEE PROFILE

Using Neural Networks to Forecast Stock Market Prices

Ramon Lawrence
Department of Computer Science
University of Manitoba
umlawren@cs.umanitoba.ca

December 12, 1997

Abstract

This paper is a survey on the application of neural networks in forecasting stock market prices. With their ability to discover patterns in nonlinear and chaotic systems, neural networks offer the ability to predict market directions more accurately than current techniques. Common market analysis techniques such as technical analysis, fundamental analysis, and regression are discussed and compared with neural network performance. Also, the Efficient Market Hypothesis (EMH) is presented and contrasted with chaos theory and neural networks. This paper refutes the EMH based on previous neural network work. Finally, future directions for applying neural networks to the financial markets are discussed.

1 Introduction

From the beginning of time it has been man's common goal to make his life easier. The prevailing notion in society is that wealth brings comfort and luxury, so it is not surprising that there has been so much work done on ways to predict the markets. Various technical, fundamental, and statistical indicators have been proposed and used with varying results. However, no one technique or combination of techniques has been successful enough to consistently "beat the market". With the development of neural networks, researchers and investors are hoping that the market mysteries can be unraveled. This paper is a survey of current market forecasting techniques with an emphasis on why they are insufficient and how neural networks have been used to improve upon them.

The paper is organized as follows. Section 2 provides the motivation for predicting stock market prices. Section 3 covers current analytical and computer methods used to forecast stock market prices. The majority of the work, in Section 4, details how neural networks have been designed to outperform current techniques.

Several example systems are discussed with a comparison of their performance with other techniques. The paper concludes with comments on possible future work in the area and some conclusions.

2 Motivation

There are several motivations for trying to predict stock market prices. The most basic of these is financial gain. Any system that can consistently pick winners and losers in the dynamic market place would make the owner of the system very wealthy. Thus, many individuals including researchers, investment professionals, and average investors are continually looking for this superior system which will yield them high returns.

There is a second motivation in the research and financial communities. It has been proposed in the Efficient Market Hypothesis (EMH) that markets are efficient in that opportunities for profit are discovered so quickly that they cease to be opportunities. The EMH effectively states that no system can continually beat the market because if this system becomes public, everyone will use it, thus negating its potential gain. There has been an ongoing debate about the validity of the EMH, and some researchers attempted to use neural networks to validate their claims. There has been no consensus on the EMH's validity, but many market observers tend to believe in its weaker forms, and thus are often unwilling to share proprietary investment systems.

Neural networks are used to predict stock market prices because they are able to learn nonlinear mappings between inputs and outputs. Contrary to the EMH, several researchers claim the stock market and other complex systems exhibit chaos. Chaos is a nonlinear deterministic process which only appears random because it can not be easily expressed. With the neural networks' ability to learn nonlinear, chaotic systems, it may be possible to outperform traditional analysis and other computer-based methods.

In addition to stock market prediction, neural networks have been trained to perform a variety of financial related tasks. There are experimental and commercial systems used for tracking commodity markets and futures, foreign exchange trading, financial planning, company stability, and bankruptcy prediction. Banks

use neural networks to scan credit and loan applications to estimate bankruptcy probabilities, while money managers can use neural networks to plan and construct profitable portfolios in real-time. As the application of neural networks in the financial area is so vast, this paper will focus on stock market prediction.

Finally, although neural networks are used primarily as an application tool in the financial environment, several research improvements have been made during their implementation. Notable improvements in network design and training and the application of theoretical techniques are demonstrated by the examination of several example systems.

3 Analytical Methods

Before the age of computers, people traded stocks and commodities primarily on intuition. As the level of investing and trading grew, people searched for tools and methods that would increase their gains while minimizing their risk. Statistics, technical analysis, fundamental analysis, and linear regression are all used to attempt to predict and benefit from the market's direction. None of these techniques has proven to be the consistently correct prediction tool that is desired, and many analysts argue about the usefulness of many of the approaches. However, these methods are presented as they are commonly used in practice and represent a base-level standard for which neural networks should outperform. Also, many of these techniques are used to preprocess raw data inputs, and their results are fed into neural networks as input.

3.1 Technical Analysis

The idea behind technical analysis is that share prices move in trends dictated by the constantly changing attitudes of investors in response to different forces. Using price, volume, and open interest statistics, the technical analyst uses charts to predict future stock movements. Technical analysis rests on the assumption that history repeats itself and that future market direction can be determined by examining past prices. Thus, technical analysis is controversial and contradicts the Efficient Market Hypothesis. However, it is used by

approximately 90% of the major stock traders[3]. Despite its widespread use, technical analysis is criticized because it is highly subjective. Different individuals can interpret charts in different manners.

Price charts are used to detect trends. Trends are assumed to be based on supply and demand issues which often have cyclical or noticeable patterns. There are a variety of technical indicators derived from chart analysis which can be formalized into trading rules or used as inputs to neural networks. Some technical indicator categories include filter indicators, momentum indicators, trend line analysis, cycle theory, volume indicators, wave analysis, and pattern analysis. Indicators may provide short or long term information, help identify trends or cycles in the market, or indicate the strength of the stock price using support and resistance levels.

An example of a technical indicator is the moving average. The moving average averages stock prices over a given length of time allowing trends to be more visible. Several trading rules have been developed which pertain to the moving average. For example, "when a closing price moves above a moving average a buy signal is generated." [3]. Unfortunately, these indicators often give false signals and lag the market. That is, since a moving average is a past estimate, a technical trader often misses a lot of the potential in the stock movement before the appropriate trading signal is generated. Thus, although technical analysis may yield insights into the market, its highly subjective nature and inherent time delay does not make it ideal for the fast, dynamic trading markets of today.

3.2 Fundamental Analysis

Fundamental analysis involves the in-depth analysis of a company's performance and profitability to determine its share price. By studying the overall economic conditions, the company's competition, and other factors, it is possible to determine expected returns and the intrinsic value of shares. This type of analysis assumes that a share's current (and future) price depends on its intrinsic value and anticipated return on investment. As new information is released pertaining to the company's status, the expected return on the

company's shares will change, which affects the stock price.

The advantages of fundamental analysis are its systematic approach and its ability to predict changes before they show up on the charts. Companies are compared with one another, and their growth prospects are related to the current economic environment. This allows the investor to become more familiar with the company. Unfortunately, it becomes harder to formalize all this knowledge for purposes of automation (with a neural network for example), and interpretation of this knowledge may be subjective. Also, it is hard to time the market using fundamental analysis. Although the outstanding information may warrant stock movement, the actual movement may be delayed due to unknown factors or until the rest of the market interprets the information in the same way. However, fundamental analysis is a superior method for long-term stability and growth. Basically, fundamental analysis assumes investors are 90% logical, examining their investments in detail, whereas technical analysis assumes investors are 90% psychological, reacting to changes in the market environment in predictable ways.

3.3 Traditional Time Series Forecasting

Time series forecasting analyzes past data and projects estimates of future data values. Basically, this method attempts to model a nonlinear function by a recurrence relation derived from past values. The recurrence relation can then be used to predict new values in the time series, which hopefully will be good approximations of the actual values. A detailed analysis and description of these models is beyond the scope of this paper. However, a short overview is presented as the results from these models are often compared with neural network performance.

There are two basic types of time series forecasting: univariate and multivariate. Univariate models, like Box-Jenkins, contain only one variable in the recurrence equation. Box-Jenkins is a complicated process of fitting data to appropriate model parameters. The equations used in the model contain past values of moving averages and prices. Box-Jenkins is good for short-term forecasting but requires a lot of data, and

it is a complicated process to determine the appropriate model equations and parameters.

Multivariate models are univariate models expanded to "discover casual factors that affect the behavior of the data." [3] As the name suggests, these models contain more than one variable in their equations. Regression analysis is an multivariate model which has been frequently compared with neural networks. Overall, time series forecasting provides reasonable accuracy over short periods of time, but the accuracy of time series forecasting diminishes sharply as the length of prediction increases.

3.4 The Efficient Market Hypothesis

The Efficient Market Hypothesis (EMH) states that at any time, the price of a share fully captures all known information about the share. Since all known information is used optimally by market participants, price variations are random, as new information occurs randomly. Thus, share prices perform a "random walk", and it is not possible for an investor to beat the market.

Despite its rather strong statement that appears to be untrue in practice, there has been inconclusive evidence in rejecting the EMH. Different studies have concluded to accept or reject the EMH. Many of these studies used neural networks to justify their claims. However, since a neural network is only as good as it has been trained to be, it is hard to argue for acceptance or rejection of the hypothesis based solely on neural network performance. In practice, stock market crashes, such as the market crash in October 1987, contradict the EMH because they are not based on randomly occurring information, but arise in times of overwhelming investor fear.

The EMH is important because it contradicts all other forms of analysis. If it is impossible to beat the market, then technical, fundamental, or time series analysis should lead to no better performance than random guessing. The fact that many market participants can consistently beat the market is an indication that the EMH may not be true in practice. The EMH may be true in the ideal world with equal information distribution, but today's markets contain several privileged players who can outperform the market by using

inside information or other means.

3.5 Chaos Theory

A relatively new approach to modeling nonlinear dynamic systems like the stock market is chaos theory. Chaos theory analyzes a process under the assumption that part of the process is deterministic and part of the process is random. Chaos is a nonlinear process which appears to be random. Various theoretical tests have been developed to test if a system is chaotic (has chaos in its time series). Chaos theory is an attempt to show that order does exist in apparent randomness. By implying that the stock market is chaotic and not simply random, chaos theory contradicts the EMH.

In essence, a chaotic system is a combination of a deterministic and a random process. The deterministic process can be characterized using regression fitting, while the random process can be characterized by statistical parameters of a distribution function. Thus, using only deterministic or statistical techniques will not fully capture the nature of a chaotic system. A neural networks ability to capture both deterministic and random features makes it ideal for modeling chaotic systems.

3.6 Other Computer Techniques

Many other computer based techniques have been employed to forecast the stock market. They range from charting programs to sophisticated expert systems. Fuzzy logic has also been used.

Expert systems process knowledge sequentially and formulate it into rules. They can be used to formulate trading rules based on technical indicators. In this capacity, expert systems can be used in conjunction with neural networks to predict the market. In such a combined system, the neural network can perform its prediction, while the expert system could validate the prediction based on its well-known trading rules.

The advantage of expert systems is that they can explain how they derive their results. With neural networks, it is difficult to analyze the importance of input data and how the network derived its results.

However, neural networks are faster because they execute in parallel and are more fault tolerant.

The major problem with applying expert systems to the stock market is the difficulty in formulating knowledge of the markets because we ourselves do not completely understand them. Neural networks have an advantage over expert systems because they can extract rules without having them explicitly formalized. In a highly chaotic and only partially understood environment, such as the stock market, this is an important factor. It is hard to extract information from experts and formalize it in a way usable by expert systems. Expert systems are only good within their domain of knowledge and do not work well when there is missing or incomplete information. Neural networks handle dynamic data better and can generalize and make "educated guesses." Thus, neural networks are more suited to the stock market environment than expert systems.

3.7 Comparing the various models

In the wide variety of different modeling techniques presented so far, every technique has its own set of supporters and detractors and vastly differing benefits and shortcomings. The common goal in all the methods is predicting future market movements from past information. The assumptions made by each method dictate its performance and its application to the markets.

The EMH assumes that fully disseminated information results in an unpredictable random market. Thus, no analysis technique can consistently beat the market as others will use it, and its gains will be nullified. I believe that the EMH has some merit theoretically, but in real-world applications, it is painfully obvious that there is a uneven playing field. Some market participants have more information or tools which allow them to beat the market or even manipulate it. Thus, stock market prices are not simply a random walk, but are derived from a dynamic system with complexities too vast to be fully accounted for.

If an investor does not believe in the EMH, the other models offer a variety of possibilities. Technical analysis assumes history repeats itself and noticeable patterns can be discerned in investor behavior by

examining charts. Fundamental analysis helps the long-term investor measure intrinsic value of shares and their future direction by assuming investors make rational investment decisions. Statistical and regression techniques attempt to formulate past behavior in recurrent equations to predict future values. Finally, chaos theory states that the apparent randomness of the market is just nonlinear dynamics too complex to be fully understood.

So what model is the right one? There is no right model. Each model has its own benefits and shortcomings. I feel that the market is a chaotic system. It may be predictable at times, while at other times it appears totally random. The reason for this is that human beings are neither totally predictable nor totally random. Although it is nearly impossible to determine a person's reaction to information or situations, there are always some basic trends in behavior as well as some random elements. The market is a collection of millions of people acting in a chaotic manner. It is as impossible to predict the behavior of a million people as it is to predict the behavior of one person. Investors are neither mostly psychological as predicted by technical analysis, nor logical as predicted by fundamental analysis. Our approach and view on the world varies daily in a manner that we do not even fully understand, so it follows that the stock market behaves in similar ways.

In conclusion, these methods work best when employed together. The major benefit of using a neural network then is for the network to learn how to use these methods in combination effectively, and hopefully learn how the market behaves as a factor of our collective consciousness.

4 Application of Neural Networks to Market Prediction

4.1 Overview

The ability of neural networks to discover nonlinear relationships in input data makes them ideal for modeling nonlinear dynamic systems such as the stock market. Various neural network configurations have

been developed to model the stock market. Commonly, these systems are created in order to determine the validity of the EMH or to compare them with statistical methods such as regression. Often these networks use raw data and derived data from technical and fundamental analysis discussed previously.

This section will overview the use of neural networks in financial markets including a discussion of their inputs, outputs, and network organization. Also, any interesting applications of theoretical techniques will be examined. For example, many networks prune redundant nodes to enhance their performance. The networks are examined in three main areas:

1. Network environment and training data
2. Network organization
3. Network performance

4.2 Training a Neural Network

A neural network must be trained on some input data. The two major problems in implementing this training discussed in the following sections are:

1. Defining the set of input to be used (the learning environment)
2. Deciding on an algorithm to train the network

4.2.1 The Learning Environment

One of the most important factors in constructing a neural network is deciding on what the network will learn. The goal of most of these networks is to decide when to buy or sell securities based on previous market indicators. The challenge is determining which indicators and input data will be used, and gathering enough training data to train the system appropriately.

The input data may be raw data on volume, price, or daily change, but it may also include derived data such as technical indicators (moving average, trend-line indicators, etc.) or fundamental indicators (intrinsic share value, economic environment, etc.). One neural network system[13] used phrases out of the

president's report to shareholders as input. The input data should allow the neural network to generalize market behavior while containing limited redundant data.

A comprehensive example neural network system, henceforth called the JSE-system[3], modeled the performance of the Johannesburg Stock Exchange. This system had 63 indicators, from a variety of categories, in an attempt to get an overall view of the market environment by using raw data and derived indicators. The 63 input data values can be divided into the following classes with the number of indicators in each class in parenthesis:

1. fundamental(3) - volume, yield, price/earnings
2. technical(17) - moving averages, volume trends, etc.
3. JSE indices(20) - market indices for various sectors: gold, metals, etc.
4. international indices(9) - DJIA, etc.
5. gold price/foreign exchange rates(3)
6. interest rates(4)
7. economic statistics(7) - exports, imports, etc.

The JSE-system normalized all data to the range $[-1,1]$. Normalizing data is a common feature in all systems as neural networks generally use input data in the range $[0,1]$ or $[-1,1]$. Scaling some input data may be a difficult task especially if the data is not numeric in nature.

Other neural network systems use similar types of input data. Simpler systems may use only past share prices[10] or chart information[4]. A system developed by Yoon[13] based its input on the types and frequencies of key phrases used in the president's report to shareholders. Bergerson[2] created a system that traded commodities by training it on human designed chart information rather than raw data. Such directed training has the advantage of focusing the neural network to learn specific features that are already known as well as reducing learning time. Finally, the self-organizing system built by Wilson[12] used a combination of technical, adaptive (based on limited support functions), and statistical indicators as inputs.

It is interesting to note that although the final JSE-system was trained with all 63 inputs, the analysis showed that many of the inputs were unnecessary. The authors used cross-validation techniques and

sensitivity analysis to discard 20 input values (and reduce the number of hidden nodes from 21 to 14) with negligible effect on system performance. Such pruning techniques are very important because they reduce the network size which speeds up recall and training times. As the number of inputs to the network may be very large, pruning techniques are especially useful. Examining the weight matrix for very large or very small weights may also help in detecting useless inputs.

Determining the proper input data is the first step in training the network. The second step is presenting the input data in a way that allows the network to learn properly without overtraining. Various training procedures have been developed to train these networks.

4.2.2 Network Training

Training a network involves presenting input patterns in a way so that the system minimizes its error and improves its performance. The training algorithm may vary depending on the network architecture, but the most common training algorithm used when designing financial neural networks is the backpropagation algorithm. This section describes some of the training techniques and their associated challenges in some implemented systems.

The most common network architecture for financial neural networks is a multilayer feedforward network trained using backpropagation. Backpropagation is the process of backpropagating errors through the system from the output layer towards the input layer during training. Backpropagation is necessary because hidden units have no training target value that can be used, so they must be trained based on errors from previous layers. The output layer is the only layer which has a target value for which to compare. As the errors are backpropagated through the nodes, the connection weights are changed. Training occurs until the errors in the weights are sufficiently small to be accepted. It is interesting to note that the type of activation function used in the neural network nodes can be a factor on what data is being learned. According to Klimasauskas[6], the sigmoid function works best when learning about average behavior,

while the hyperbolic tangent (\tanh) function works best when learning deviation from the average.

The major problem in training a neural network is deciding when to stop training. Since the ability to generalize is fundamental for these networks to predict future stock prices, overtraining is a serious problem. Overtraining occurs when the system memorizes patterns and thus loses the ability to generalize. It is an important factor in these prediction systems as their primary use is to predict (or generalize) on input data that it has never seen. Overtraining can occur by having too many hidden nodes or training for too many time periods (epochs). Poor results in papers are often blamed on overtraining. However, overtraining can be prevented by performing test and train procedures or cross-validation.

The test and train procedure involves training the network on most of the patterns (usually around 90%) and then testing the network on the remaining patterns. The network's performance on the test set is a good indication of its ability to generalize and handle data it has not been trained on. If the performance on the test set is poor, the network configuration or learning parameters can be changed. The network is then retrained until its performance is satisfactory. Cross-validation is similar to test and train except the input data is divided into k sets. The system is trained on $k-1$ sets and tested on the remaining set k times (using a different test set each time). Application of these procedures should minimize overtraining, provide an estimate on network error, and determine the optimal network configuration. With these procedures, it is a weak statement to say the network's performance was poor because of overtraining, as overtraining can be controlled by the experimenter.

The amount of training data is also important. Ideally, it is desirable to have as much training data as the system can be feasibly trained with. The JSE-system was trained on 455 patterns and tested on 51 patterns. Each pattern corresponded to the input values on a particular day. A system designed to analyze IBM stock returns[11] was trained on 1000 days of data and tested on 500 days of data. A similar system designed to predict Tokyo stocks[5] learned daily data over 33 months. Some systems were even trained on data spanning over 50 years. It is desirable to have a lot of data available, as some patterns may not be detectable

in small data sets. However, it is often difficult to obtain a lot of data with complete and correct values. As well, training on large volumes of historical data is computationally and time intensive and may result in the network learning undesirable information in the data set. For example, stock market data is time-dependent. Sufficient data should be presented so that the neural network can capture most of the trends, but very old data may lead the network to learn patterns or factors that are no longer important or valuable.

The amount of data the system can process is practically limited by computer processing speeds. As the dimensionality of the input space tends to be large (eg. 63 for the JSE-system), training the neural network is a very computationally expensive procedure. Also, the number of hidden nodes tends to be large, which further slows training. Although computational speed is a diminishing issue in today's environment, many of the studies on neural networks were performed during times where computational resources were not as readily available. These performance restrictions partially explain some of the initially poor results. For example, the IBM stock modeling system[11] was trained for over 30 hours using backpropagation without converging on a 4 MIPS machine. The author had to settle for a single-layer feedforward network which yielded poor results. Another 3-32-16-1 backpropagation system trained on UK stocks[8] took 30,000 iterations over 3 to 4 days to train. Thus, backpropagation methods often yield good results, but training them is complicated and computationally intensive.

A variation on the backpropagation algorithm which is more computationally efficient was proposed by Kimoto *et al.*[5] when developing a system to predict Tokyo stock prices. They called it supplementary learning. In supplementary learning, the weights are updated based on the sum of all errors over all patterns (batch updating). Each output node in the system has an associated error threshold (tolerance), and errors are only backpropagated if they exceed this tolerance. This procedure has the effect of only changing the units in error, which makes the process faster and able to handle larger amounts of data. The system also changes the learning constants automatically based on the amount of error data being backpropagated.

The Toyko stock prediction system employed moving simulation during training. In moving simulation,

prediction is done while moving the target learning and prediction periods. For example, initially the system is trained on data from January, tested on data for February, then used to predict data for March. In the next iteration, the system trains on the February data, tests on the actual March data, and then predicts data for April. In this way, the system is continually updated based on the most recent data and its performance is improved.

Finally, there are a variety of network architectures used for financial neural networks which are not trained using backpropagation. There are different algorithms for some recurrent architectures, modular networks, and genetic algorithms which cannot be adequately covered here. Regardless of the training algorithm used, all prediction systems are very sensitive to overtraining, so techniques like cross-validation should be used to determine the system error.

4.3 Network Organizations

As mentioned before, the most common network architecture used is the backpropagation network. However, stock market prediction networks have also been implemented using genetic algorithms, recurrent networks, and modular networks. This section discusses some of the network architectures used and their effect on performance.

Backpropagation networks are the most commonly used network because they offer good generalization abilities and are relatively straightforward to implement. Although it may be difficult to determine the optimal network configuration and network parameters, these networks offer very good performance when trained appropriately.

The JSE-system was a backpropagation network designed using a genetic algorithm. The genetic algorithm allowed the automated design of the neural network, and determined that the optimal network configuration was one hidden layer with 21 nodes. Genetic algorithms are especially useful where the input dimensionality is large. They allowed the network developers to automate network configuration

without relying on heuristics or trial-and-error. The Tokyo stock prediction system[5] was a modular neural network consisting of 4 backpropagation networks trained on different data items. Many other stock market prediction systems are also based on the backpropagation network[13, 8, 10, 9, 1].

Recurrent network architectures are the second most commonly implemented architecture. The motivation behind using recurrence is that pricing patterns may repeat in time. A network which remembers previous inputs or feedbacks previous outputs may have greater success in determining these time dependent patterns. There are a variety of such networks which may have recurrent connections between layers, or remember previous outputs and use them as new inputs to the system (increases input space dimensionality). The performance of these networks are quite good. A recurrent network model was used in [4].

A self-organizing system was also developed by Wilson[12] to predict stock prices. The self-organizing network was designed to construct a nonlinear chaotic model of stock prices from volume and price data. Features in the data were automatically extracted and classified by the system. The benefit in using a self-organizing neural network is it reduces the number of features (hidden nodes) required for pattern classification, and the network organization is developed automatically during training. Wilson used two self-organizing neural networks in tandem; one selected and detected features of the data, while the other performed pattern classification. Overfitting and difficulties in training were still problems in this organization.

An interesting hybrid network architecture was developed in [2], which combined a neural network with an expert system. The neural network was used to predict future stock prices and generate trading signals. The expert system used its management rules and formulated trading techniques to validate the neural network output. If the output violated known principles, the expert system could veto the neural network output, but would not generate an output of its own. This architecture has potential because it combines the nonlinear prediction of neural networks with the rule-based knowledge of expert systems. Thus, the combination of the two systems offers superior knowledge and performance.

There is no one correct network organization. Each network architecture has its own benefits and drawbacks. Backpropagation networks are common because they offer good performance, but are often difficult to train and configure. Recurrent networks offer some benefits over backpropagation networks because their "memory feature" can be used to extract time dependencies in the data, and thus enhance prediction. More complicated models may be useful to reduce error or network configuration problems, but are often more complex to train and analyze.

4.4 Network Performance

A network's performance is often measured on how well the system predicts market direction. Ideally, the system should predict market direction better than current methods with less error. Some neural networks have been trained to test the EMH. If a neural network can outperform the market consistently or predict its direction with reasonable accuracy, the validity of the EMH is questionable. Other neural networks were developed to outperform current statistical and regression techniques.

Many of the first neural networks used for predicting stock prices were to validate the EMH. The EMH, in its weakest form, states that a stock's direction cannot be determined from its past price. Several contradictory studies were done to determine the validity of this statement. The JSE-system demonstrated superior performance and was able to predict market direction, so it refuted the EMH. A backpropagation network[10] which only used past share prices as input had some predictive ability, which refutes the weak form of the EMH. In contrast, an earlier study on IBM stock movement[11] did not find evidence against the EMH. However, the network used was a single-layer feedforward network, which does not have a lot of generalization power. Overall, neural networks are able to partially predict share prices, thus refuting the EMH.

Neural networks have also been compared to statistical and regression techniques. The JSE-system was able to predict market movement correctly 92% of the time, while Box-Jenkins only performed at a

60% rate. Other systems[8, 1, 9] have constructed neural networks that consistently outperform regression techniques such as multiple linear regression (MLR). Also, the neural network proposed in [13] predicted price trends correctly 91% of the time as compared to 74% using multiple discriminant analysis (MDA). Thus, neural networks also consistently outperform statistical and regression techniques.

The ultimate goal is for neural networks to outperform the market or index averages. The Tokyo stock trading system[5] outperformed the buy-and-hold strategy and the Tokyo index. The self-organizing model[12] constructed model portfolios with higher returns and less volatility (risk) than the S&P 500 index. As well, most of these systems process large amounts of data on many different stocks much faster than human operators. Thus, a neural network can examine more market positions or charts than experienced traders.

Finally, it is debatable on what neural network architecture offers the best performance. McCluskey in his master's thesis[7] compared many different network architectures including backpropagation, recurrent models, and genetic algorithms. The networks were designed to predict S&P 500 index prices several weeks in advance. The networks were trained using 5-fold cross-validation using two different data sets. One data set spanned the years 1928-1993, while the other data set contained only data from 1979-1993. The neural networks all beat the buy-and-hold strategy and the S&P 500 index, but often failed to beat a hand-coded optimization performed using the same input data.

Some of the results were surprising. For example, during the test period the S&P 500 index increased 10%, the backpropagation network returned 20%, a single-layer network returned 21%, and the hand-coded method returned 20%. It is unusual that a single-layer network with its limited ability to generalize would outperform backpropagation. McCluskey attributed this irregularity to overtraining. Recurrent networks returned 30%, genetic algorithms yielded 23%, while cascade networks returned 22%. Cascade networks add hidden units one at a time, train several units in each step, then discard all the hidden units except for the one most correlated with the error. A large performance increase resulted by adding "windowing".

Windowing is the process of remembering previous inputs of the time series and using them as inputs to calculate the current prediction. Recurrent networks returned 50% using windows, and cascade networks returned 51%. Thus, the use of recurrence and remembering past inputs appears to be useful in forecasting the stock market.

5 Future Work

Using neural networks to forecast stock market prices will be a continuing area of research as researchers and investors strive to outperform the market, with the ultimate goal of bettering their returns. It is unlikely that new theoretical ideas will come out of this applied work. However, interesting results and validation of theories will occur as neural networks are applied to more complicated problems. For example, network pruning and training optimization are two very important research topics which impact the implementation of financial neural networks. Financial neural networks must be trained to learn the data and generalize, while being prevented from overtraining and memorizing the data. Also, due to their large number of inputs, network pruning is important to remove redundant input nodes and speed-up training and recall.

The major research thrust in this area should be determining better network architectures. The commonly used backpropagation network offers good performance, but this performance could be improved by using recurrence or reusing past inputs and outputs. The architecture combining neural networks and expert systems shows potential. Currently, implemented neural networks have shown that the Efficient Market Hypothesis does not hold in practice, and that stock markets are probably chaotic systems. Until we more fully understand the dynamics behind such chaotic systems, the best we can hope for is to model them as accurately as possible. Neural networks appear to be the best modeling method currently available as they capture nonlinearities in the system without human intervention. Continued work on improving neural network performance may lead to more insights in the chaotic nature of the systems they model. However, it is unlikely a neural network will ever be the perfect prediction device that is desired because the factors

in a large dynamic system, like the stock market, are too complex to be understood for a long time.

6 Conclusion

This paper surveyed the application of neural networks to financial systems. It demonstrated how neural networks have been used to test the Efficient Market Hypothesis and how they outperform statistical and regression techniques in forecasting share prices. Although neural networks are not perfect in their prediction, they outperform all other methods and provide hope that one day we can more fully understand dynamic, chaotic systems such as the stock market.

References

- [1] Dirk Emma Baestaens and Willem Max van den Bergh. Tracking the Amsterdam stock index using neural networks. In *Neural Networks in the Capital Markets*, chapter 10, pages 149–162. John Wiley and Sons, 1995.
- [2] K. Bergerson and D. Wunsch. A commodity trading model based on a neural network-expert system hybrid. In *Neural Networks in Finance and Investing*, chapter 23, pages 403–410. Probus Publishing Company, 1993.
- [3] Robert J. Van Eyden. *The Application of Neural Networks in the Forecasting of Share Prices*. Finance and Technology Publishing, 1996.
- [4] K. Kamijo and T. Tanigawa. Stock price pattern recognition: A recurrent neural network approach. In *Neural Networks in Finance and Investing*, chapter 21, pages 357–370. Probus Publishing Company, 1993.

- [5] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 1–6, 1990.
- [6] C. Klimasauskas. Applying neural networks. In *Neural Networks in Finance and Investing*, chapter 3, pages 47–72. Probus Publishing Company, 1993.
- [7] P. G McCluskey. Feedforward and recurrent neural networks and genetic programs for stock market and time series forecasting. Technical Report CS-93-36, Brown University, September 93.
- [8] Apostolos-Paul Refenes, A.D. Zaprani, and G. Francis. Modelling stock returns in the framework of APT: A comparative study with regression models. In *Neural Networks in the Capital Markets*, chapter 7, pages 101–126. John Wiley and Sons, 1995.
- [9] Manfred Steiner and Hans-Georg Wittkemper. Neural networks as an alternative stock market model. In *Neural Networks in the Capital Markets*, chapter 9, pages 137–148. John Wiley and Sons, 1995.
- [10] G. Tsibouris and M. Zeidenberg. Testing the Efficient Markets Hypothesis with gradient descent algorithms. In *Neural Networks in the Capital Markets*, chapter 8, pages 127–136. John Wiley and Sons, 1995.
- [11] H. White. Economic prediction using neural networks: The case of IBM daily stock returns. In *Neural Networks in Finance and Investing*, chapter 18, pages 315–328. Probus Publishing Company, 1993.
- [12] C. L. Wilson. Self-organizing neural network system for trading common stocks. In *Proc. ICNN'94, Int. Conf. on Neural Networks*, pages 3651–3654, Piscataway, NJ, 1994. IEEE Service Center.
- [13] Y. Yoon and G. Swales. Predicting stock price performance: A neural network approach. In *Neural Networks in Finance and Investing*, chapter 19, pages 329–342. Probus Publishing Company, 1993.