NEURAL NETWORK AS A FORECASTING TOOL FOR FINANCIAL DECISION-MAKING

ONUR GÖRGÜN

IŞIK UNIVERSITY
2008

NEURAL NETWORK AS A FORECASTING TOOL FOR FINANCIAL DECISION-MAKING

ONUR GÖRGÜN
B.S., Information Technologies, Işık University, 2005

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Information Technologies

IŞIK UNIVERSITY
2008

ISIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

NEURAL NETWORK AS A FORECASTING TOOL
FOR FINANCIAL DECISION-MAKING

ONUR GÖRGÜN

APPROVED BY:

Assoc. Prof. Nazım Ziya Perdahçı ____________________
(Thesis Supervisor)

Prof. Dr. Hasan Dağ ____________________

Dr. Özlem İnanç ____________________

APPROVAL DATE: 18 / 09 / 2008

NEURAL NETWORK AS A FORECASTING TOOL
FOR FINANCIAL DECISION-MAKING

Abstract

For the last decade, machine learning techniques have been applied to financial tasks such as portfolio management, risk assessment and stock market prediction. Among these techniques artificial neural network as a machine learning algorithm is the most widely used model. In stock market environment, multi layer perceptron with backpropagation model is dominant among others in stock market prediction. This study examines the forecasting power of multi layer perceptron models for predicting the direction of ISE 100 daily index value. The results show that multi layer perceptron has a promising power in predicting the stock market trend. However, it also shows that selection of input variables is dominant factor in stock market prediction to obtain accurate results.

FİNANSAL KARAR ALMADA ÖNGÖRÜ ARACI OLARAK SİNİR AĞI

Özet

Son on yılda makine öğrenimi yöntemleri portföy yönetimi, risk değerlendirmesi ve hisse senedi piyasası öngörme gibi finansal problemleri çözmede kullanılmaktadır. Bütün modeller içerisinde yapay sinir ağı ise en fazla uygulanan yöntem olarak görülmektedir. Hisse senedi piyasalarında hata geri yayma yöntemi ile eğitilmiş çok katmanlı algılayıcı baskın yapay sinir ağları modelidir. Bu çalışma çok katmanlı algılayıcıların İstanbul Menkul Kıymetler Borsası 100 endeksinin yönünün tahmininde ki gücünü incelemektedir. Sonuçlar çok katmanlı algılayıcının borsa piyasası tahmini konusunda gelecek vadeden bir yapı olduğunu ortaya koymaktadır. Ancak, doğru girdi değişkeni seçiminin isabetli tahmin yapma konusunda ne kadar etkili olduğu da vurgulanmaktadır.

Acknowledgements

There are many people who helped to make my years at the graduate school most valuable. First, I thank Nazım Ziya Perdahçı, my major professor and thesis supervisor. Having the opportunity to work with him over the years was intellectually rewarding and fulfilling.

I would like to thank Hasan Dağ and Özlem İnanç for their support and help to improve the thesis. Many thanks to my colleagues who helped me all through the years full of class work and their deep support. The last words of thanks go to my family. I thank my parents for their patience and encouragement, and endless support through this long journey.

I dedicate the thesis to my parents who always believed in me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Networks
ANOVA	Analysis of Variance
ARMA	Auto Regressive Moving Average
CBRT	Central Bank of the Republic of Turkey
EMH	Efficient Market Hypothesis
GFNN	Generalized Feedforward Neural Network
HR	Hit Rate
IGE	İstanbul Gold Exchange
ISE	İstanbul Stock Index
MA	Moving Average
MACD	Moving Average Convergence/Divergence
MAPE	Mean Absolute Percentage Error
MFI	Money Flow Index
MFNN	Multi Layer Feedforward Neural Network
MLP	Multi Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
ON	Overnight Interest Rate
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
SSE	Sum Squared Error
SO	Stochastic Oscillator

Chapter 1
Introduction

Stock market prediction is defined as acting to determine the future value of stock market. Determination can be done in two ways; predicting the future value of individual stock or stock market index value. The major consequence of successful stock market prediction is the possibility of gathering significant profit. On the other hand, there exists some hypothesizes stating the impossibility of predicting the stock market. However, existence of these hypothesizes motivate both researchers and individuals to work on forecasting models which are able to yield more accurate results.

Neural networks have been known as an efficient tool for non-parametric modeling of data where the output is non-linear function of inputs. The non-linear characteristic of financial data makes neural networks as an attractive alternative tool for forecasting researchers and practitioners. There are several distinguishing features that make neural networks attractive for financial prediction. First, neural networks are data-driven self-adaptive methods unlike traditional models and learn by examples. Second, a neural network can generalize. After training, it can even predict highly noisy data. Third, they are accepted as general approximators and can approximate continuous functions accurately. Finally, as opposed to traditional forecasting methods, they are non-linear.

This study examines and analyzes the use of neural networks as a forecasting tool in financial markets. Specifically, neural networks' ability to forecast the future trends in stock market is tested, and neural network performance is compared against traditional forecasting methods, linear and non-linear regression tools. The study builds upon studies discussed through the literature, and tries to build a roadmap to model the stock market environment using neural network approach. Modeling is based on the direction prediction capabilities of neural networks, and it is tested on ISE100 stock index value.

The objective is to examine the theory of artificial neural network and then develop a model that accurately predicts the direction of the stock market closing price for the next day using commercially available software packages. When the task is accomplished, hit rate of the model, based on the performance of model in prediction period, is calculated. After performance validation by comparing network results with regression analysis, benefits for the investor is determined.

In order to meet the objectives proposed we will look for answers for the following questions:

- For what reasons need for intelligent systems in financial forecasting arise?
- What is the theory behind artificial neural network?
- What are the issues to be considered during when building a neural network model for stock market prediction?
- Can neural networks accurately forecast a stock market index?
- Can neural networks make more accurate questions than multiple linear regression and non-linear regression techniques?

The strategy used while conducting research starts with learning the mathematical theory behind artificial neural networks. Next, research continues with the literature review on neural network studies conducted in financial area. Then, we try to illustrate the issues related to neural network modeling for prediction tasks. These issues are grouped into appropriate categories, data specific issues, structure specific issues and performance evaluation issues. With respect to the issues defined, we try to build our model in order to accomplish the prediction task.

The remaining portion of the thesis is broken into the following chapters: Neural Networks in Financial Tasks (Chapter 1), Predicting ISE100 Index with Neural Networks (Chapter 2), and Conclusions and Recommendations for Future Works. In Chapter 1, financial tasks addressed by neural networks are discussed. Main focus of this chapter is stock market prediction. While introducing stock market prediction, the factors motivating both individuals and researchers to build models yielding better results are also discussed. Chapter continues with an introduction to neural network theory and issues related to neural network modeling for prediction tasks. Chapter 2 includes an experiment conducted for Turkish stock market. In experiment, performance of neural network model and traditional regression models are discussed. Thesis concludes with comments on experiment and recommendations for future works.

Chapter 2
Neural Networks in Financial Tasks

2.1. Introduction

As a result of globalization in economics with the evolution of information technology, financial data are being generated in huge amounts and processed at high speeds. The main goal is to keep the track of business performance of companies (business perspective), monitoring and detecting changes in market (marketing perspective), and support financial decision-making (economics perspective). On the other hand, the huge and rapidly increasing size of these data makes it impossible for a human analyst to analyze with interesting information that will help in the decision making process. So, that leads to the need for automated approaches to analyze the data efficiently and effectively. The aim is to utilize the massive financial data to support businesses and individuals in financial decision-making and strategic planning, focusing on discovering knowledge from various electronic data repositories, both internal and external. Although the experts are able to detect patterns and correlations in financial data, unprecedented rate of data makes it impossible to screen and detect it immediately. Since machine learning techniques aim to discover hidden patterns and predict future trends in scientific researches, these models can lead to the achievement of this goal.

Artificial Neural Networks (ANN) is the most frequently used machine learning technique among data mining techniques in financial area. Moreover, ANNs are classified as flexible, autonomous and scalable tools [1]. They can contribute to solving financial problems such as stock market prediction, currency exchange rate, bank bankruptcies, understanding and managing financial risk, credit rating, loan management, bank customer profiling, and money laundering analyses.

2.2. Prediction of Stock Market

Stock market is defined as trading company shares, stocks, securities and its derivatives. As known, the stock market data is time-variant and has non-linear characteristic. Non-linear characteristic of stock market data makes classical linear regression tools, as a traditional forecasting tool, less efficient in order to make predictions. However, the stock market behavior can be predicted in a relatively acceptable precision. Since the desirable output of trading on stock market is maximum return of investment, investors should be informed on whether to buy or to sell stocks at appropriate times, should know what the right time to enter the market is, or on which financial instrument should invest. There are several motivations for trying to predict stock market behaviors. But the most common one is the “financial gain”. To obtain the financial gain many individuals, traders and researchers are continually looking for superior systems which will help them to obtain financial gain at high levels. The main advantage of neural networks is that they can approximate any nonlinear function to an arbitrary level of accuracy. Hence, they are called universal approximators [2]. Since neural networks can mine valuable information from mass historical data, it is commonly used in major financial tasks such as stock market prediction.

First motivation is the existence of Efficient Market Hypothesis (EMH). The EMH claims that no predictions system can continually beat the market. EMH states that, at any time, the price of a share captures all known information about the share [3]. All new information about the market occurs randomly. Since, the stock market prices perform a “random walk”, there is no such a system to beat the market. Despite the fact that EMH is untrue in practice, there is no conclusive evidence to reject the hypothesis. The situation in crisis periods contradicts the EMH, since most crises are based on investor fear. EMH can be true on totally ideal world where all information is distributed equally, but the market contains many privileged players with additional tools or information. It means that the market is open to manipulation.

On the other hand, some researchers state that stock market as a complex system exhibits chaos which is a non-linear deterministic process and occurs randomly.

Chaos theory assumes that one part of the process is deterministic and other part of the process is random. The deterministic process can be characterized using regression methods, and randomness can be characterized using statistical distributions. It can be considered as an attempt to show that order exists in randomness. As a result, the Chaos theory contradicts the EMH, since Chaos theory implies that stock market is chaotic.

Third, Technical Analysis is the idea that stock market moves in trends dictated by constantly changing attitudes of investors in response to different forces. The prediction is conducted by using price and volume indexes. Technical analysis based on an assumption that future price of stock market is determined by the past prices of stock market. In that case, technical analysis contradicts the EMH. Despite the fact that technical analysis is subjective, since it may be interpreted differently by different individuals, approximately most of the predictions are based on technical analysis in real world applications. In fact, the technical analysis proposes a new word to terminology, called as technical indicators. Technical indicators provide a closer look to the market. Although technical indicators may yield insights into the market, its subjective nature may lead to false signals and lag the market.

Fourth, Traditional Time-Series Forecasting is conducted by considering the past data and projecting future values. The most of the previously conducted researches is to capture relationships between available information and stock market returns relying on simple linear regression. There is no evidence to support the assumption that this relationship is perfectly linear. Moreover, some researchers suggested that non-linear analysis needs to be considered. In general, these processes can be used to model a non-linear function derived from past values, and the prediction results produces a base point which is used for comparing with proposed model's performance. Time series forecasting can be done in two ways: univariate and multivariate. Box-Jenkins methodology is one of the univariate time series forecasting. The appropriate model equations and parameters are hard to determine since it requires a lot of data. Hence, Box-Jenkins is a complicated process which is good for short-term prediction. Multivariate models are expanded version of univariate models. Regression Analysis is the most popular multivariate time-series forecasting model. But, the accuracy of regression analysis diminishes as the prediction period is getting longer.

All proposed models come with its own benefits and shortcomings. But there is a fact that the stock market is a chaotic environment, which can be predicted at some times while other times it walks randomly. The nature of such a system is more like human-beings, since the human nature is neither totally random and nor totally predictable. It is obvious that the stock market is a collection of millions of people, and they act in chaotic manner. It is impossible to predict the behavior of millions of people as it is to predict the behavior of one person.

In addition, all proposed systems may work best when all are combined and employed together. The major benefit of using artificial neural networks, as a subcategory of machine learning, then is for the network to learn how to use these methods effectively. So, this mechanism can be expected to learn the behavior of daily stock market index at an arbitrary rate.

2.3. Machine Learning

In general, machine learning, as a broad subfield of artificial intelligence, is described as a concept involving adaptive mechanism that enables computers to learn from experience and learn by example. In other words, machine learning is concerned with the design and implementation of techniques and algorithms that enables computer to learn. In addition, the learning capabilities may improve the learning performance of an intelligent system over time. At general level, there are two types of machine learning: deductive and inductive.

Deductive Learning: It is also called as deductive reasoning. Deductive learning is defined as a process of moving from given statements (premises), which are assumed to be true, to conclusions, which must be true if the premises are true.

Inductive Learning: Inductive learning, also known as inductive reasoning, is a process of reasoning in which the premises are believed to support conclusion but do not entail it.

Moreover, it can be described as generalization based on individual instances, and are used to extract rules and patterns out of large data sets automatically.

Machine learning is closely related to data mining and machine learning approaches could be an alternative in data mining processes. One of the most popular approaches to machine learning is artificial neural networks (ANN). The rest of this chapter is dedicated to investigation of neural networks.

2.3.1. What is Artificial Neural Network?

An artificial neural network can be defined as a model of reasoning based on the human brain. Human brain consists of densely interconnected set of nerve cells. Nerve cells, called as neurons, can be considered as main information processing units. Generally, human brain consists of nearly 10 billion neurons and 50 trillion connections between these neurons, called synapses. Although a neuron has a simple architecture and limited processing power, owing to the huge number of connections between neurons, human brain can perform its functions much faster by using multiple neurons simultaneously. This can be considered as the most tremendous processing power in existence today.

As stated before, a neuron has a very simple structure. It consists of a cell body, soma, a number of connection cables called dendrites, and a main connection body called axon. A neuron connects to other neurons by stretching out its dendrites through other neuron's axon. Figure 2.1 illustrates a neuron's structure.

Neural signals are propagated from neuron to neuron by electrochemical reactions. A neuron receives signals from other neurons through its dendrites, and transmits signals through its axon. The axon stretches eventually into branches. The connection point between neurons is called synapse. When the synapse is formed, chemical substances called neurotransmitters are released, and it causes electrical potential change in neuron body. When the electrical potential reaches a threshold, an electrical pulse is generated and sent through the neuron axon. The same pulse is spread into synapses. The most important feature of a neural network is that it exhibits plasticity[1]. Plasticity in neural network is described as long-term changes in the strength of the connections between the neurons. The long-term changes are made based on the information that which connection leads to the right answer, and which ones do not. As a conclusion, it can be said that human brain has the ability to learn by experience. The ease of learning of human brain, led to researches to emulate the human brain in computer, called Artificial Neural Network. As an emulation of human brain, an artificial neural network is thought to have the qualifications that biological neural network satisfies. Hence, an artificial neural network should learn, or generalize, when a sufficient number of examples are exposed.

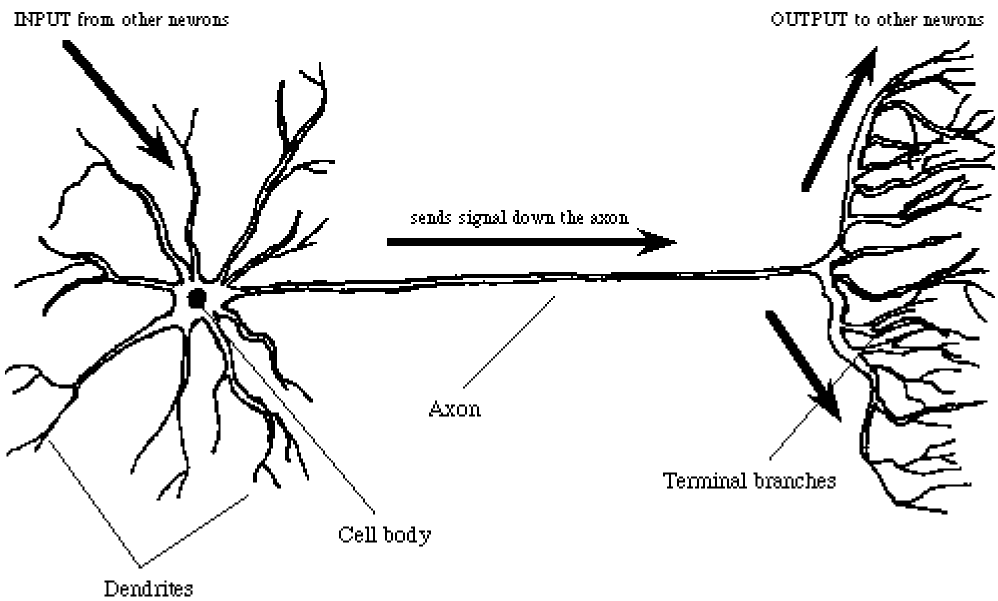

Figure 2.1 Structure of a neuron [2]

Artificial neural networks consist of a number of interconnected processors called artificial neurons. The neurons are connected to other artificial neurons by weighted links, and like neurons, an artificial neurons are connected to multiple artificial neurons. Each link has a numerical weight associated with it. These links pass output signals to other neurons through artificial neuron's outgoing connection. Outgoing signal is divided into branches, so an artificial neuron transmits the same signal

[1] In neuroscience, plasticity is a property of a neuron or synapse to change its internal parameters in response to its history

[2] Source: http://www.ccs.neu.edu/groups/honors-program/freshsem/19951996/cloder/neuron.html

throughout its outgoing connections. Figure 2.2 illustrates the connections described previously.

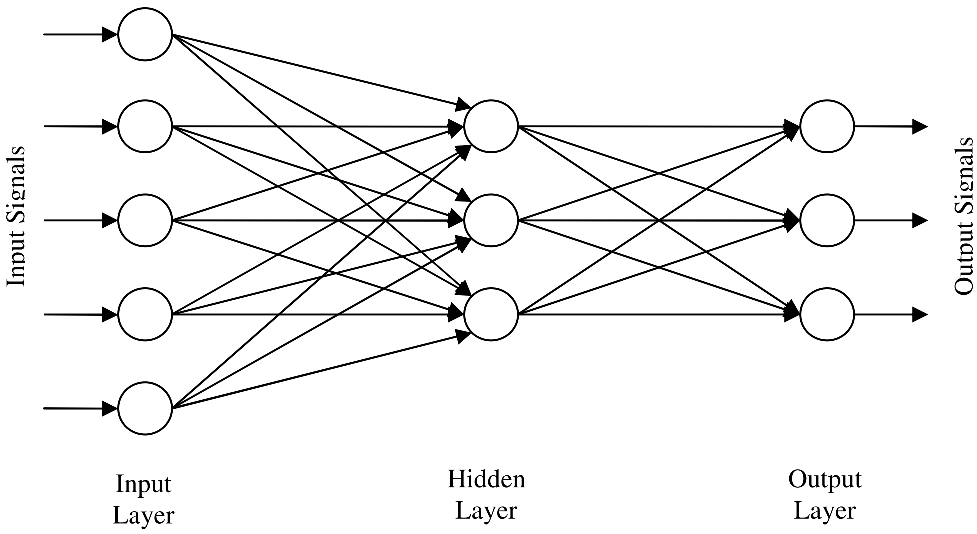

Figure 2.2 Typical neural network architecture

2.3.2. Artificial Neuron

An artificial neuron (Figure 2.3) is considered as basic processing unit of an artificial neural network. As stated previously, an artificial neuron receives several signals from its input links. These input links can be both in form of raw data or output from previous artificial neuron, and the generated output can be both the final solution to the problem or input to preceding artificial neuron. But how does an artificial neuron decide on its output? An artificial neuron must be activated to produce an output signal like biologic neuron does.

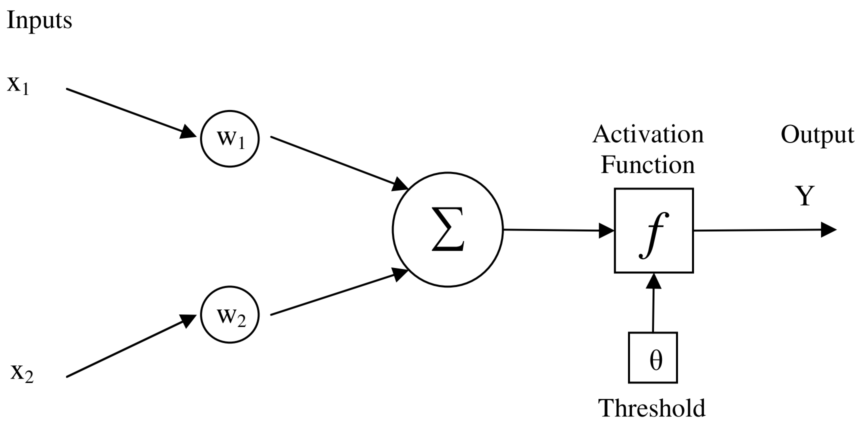

Figure 2.3. An artificial neuron

The artificial neuron computes the weighted sum of the input signals and compares the result with a predefined threshold value, θ, using an activation function.

$$\boldsymbol{X} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{w}_i \qquad (2.1)$$

where X is the net input to neuron, x_i is the value of input to neuron i, w_i is the weight of input to neuron i. The most commonly used activation functions are as follows:

Step Function

If the net input is greater than or equal to threshold value, neuron output is 1. But if the net input is less than threshold value neuron output is 0.

$$Y^{step} = \begin{cases} 1 \; if \; X \geq \theta \\ 0 \; if \; X < \theta \end{cases}$$

Sign Function

If the net input is greater than or equal to threshold value, neuron output is +1. But if the net input is less than threshold value neuron output is -1.

$$Y^{sign} = \begin{cases} +1 \; if \; X \geq \theta \\ -1 \; if \; X < \theta \end{cases}$$

Sigmoid Function

Sigmoid function is a form of logistic function. It takes any input between plus and minus infinity, and produces output in the range between 0 and 1.

$$Y^{sigmoid} = \frac{1}{1 + e^{-x}}$$

Hyperbolic Tangent

Hyperbolic tangent is another form of sigmoid function. It takes any input between plus and minus infinity, and produces output in the range between -1 and 1.

$$Y^{tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Linear Function

Linear function outputs an output equal to net input.

$$Y^{linear} = X$$

In general, step and sign functions are called as hard limiter, and commonly used in decision making neurons for classification. Sigmoid function is the mostly used in back-propagation algorithms and for approximation. In terms of statistics, a perceptron with linear activation function is a linear regression model, and a perceptron with logistic activation function is a logistic regression model [4].

As a milestone, we said that artificial neural networks learn by examples. In order to understand how this learning paradigm is solved, we have to start with the simplest neural network called perceptron. A perceptron applies a hard limiter function to net weighted input, and produces -1 and +1 based on the decision whether the weighted input is greater than or equal to, or less than threshold value. The aim of a perceptron is, basically, to classify inputs into two distinct classes. But a perceptron can only learn linearly separable functions. The point where the boundary between two classes is:

$$\sum_{i=1}^{n} x_i w_i - \theta = 0 \tag{2.2}$$

In terms of learning, a perceptron learns its classification task by making small adjustments in the weights to reduce the difference, called error, between actual and desired output. The error is given by

$$e(p) = Y_d(p) - Y(p)$$

where p refers to the pth iteration, called epoch. If the error is positive, we need to decrease $Y_{d,}$ so w_i. But if the error is negative we need to decrease $Y_{d,}$ so w_i. Thus, the following perceptron learning rule is established:

$$w_i(p+1) = w_i(p) + \alpha \times x_i(p) \times e(p) \tag{2.3}$$

where α is learning rate, a positive constant less than unity. Perceptron learning algorithm is defined as follows [5]:

Step 1: Initialization

Initialize weights $w_1, w_2, ..., w_n$ and threshold θ to random numbers [-0.5, +0.5].

Step 2: Activation

Activate the perceptron by applying inputs $x_1, x_2, ..., x_n$ and desired output Y_d. Calculate the actual output Y at epoch p=1.

$$Y(p) = step\left[\sum_{i=1}^{n} x_i(p) w_i(p) - \theta\right] \tag{2.4}$$

Step 3: Weight training

Update the weights of perceptron

$$w_i(p+1) = w_i(p) + \Delta w_i(p) \tag{2.5}$$

where Δw_i is weight correction, called delta rule:

$$\Delta w_i(p) = w_i(p) + \alpha \times x_i(p) \times e(p) \tag{2.6}$$

Step 4: Iteration

Increase iteration p by one, loop back to step 2 and repeat the process until convergence.

Since perceptron can learn only linearly separable functions, researchers mathematically analyzed the perceptron learning, and came to a conclusion that single perceptron cannot make global generalizations even if number of layers is increased. So, this limitation holds true for multilayer neural networks. Although these findings discourage the further research on neural networks, different network architectures are proposed to cope with these problems.

2.3.3. Issues in Artificial Neural Network Modeling for Forecasting

Despite the strengths of ANNs, building a neural network model for a forecaster is not a simple task. The neural network model should be build on carefully and well-tuned, since the modeling will affect the prediction performance. Important issues on building a neural network model can be listed as determining the appropriate network structure, the number of layers, the number of nodes in each layer, selection of activation functions for each layer, the training algorithm, data transformation and normalization methods, training and test sets and performance measures [6]. This section will investigate the above mentioned issues. Fully-connected multilayer neural networks trained with back-propagation algorithm are the most widely used neural network structure in financial applications. Hence, this section is based on MLP.

2.3.4. Data Specific Issues

2.3.4.1. Selection of Input Variables

This step is the most fundamental one among others, because the most accurate financial predictions rely on models which are fed with appropriate data. Also, input variable selection purely depends on a clear understanding of the economical background. Basically, input variables can be divided into two categories: raw data and technical indicators. Raw input data can be thought as the first value what comes to mind such as raw material costs, etc. In case of stock market index prediction, raw data may address previous day's closing index, or session closing prices without any transformation. On the other hand technical indicators are derived ones which can be obtained during technical analysis (Section 1.2.). Technical indicators are derived from the previously selected raw input data. There are at least five commonly used technical indicators for neural network prediction task [7]:

- *Relative Strength Index (RSI):* RSI is a financial technical analysis oscillator showing price strength by comparing upward and downward close-to-close movements. The RSI is popular because it is relatively easy to interpret. It was developed by J. Welles Wilder. The term *relative strength* also refers to the strength of a security in relation to its sector or the overall market.
- *Money Flow Index (MFI):* Money flow in technical analysis is *typical price* multiplied by volume, a kind of approximation to the dollar value of a day's trading. MFI is an oscillator calculated over an N-day period, ranging from 0 to 100, showing *money flow* on up days as a percentage of the total of up and down days.
- *Moving Average (MA):* It can be used as a generic smoothing operation. MA is used to smooth out short-term fluctuations, thus highlighting longer-term trends or cycles.
- *Stochastic Oscillator (SO):* The stochastic oscillator is a momentum indicator used in technical analysis, introduced by George Lane in the 1950s, to compare the closing price of a commodity to its price range over a given time span.
- *Moving Average Convergence/Divergence (MACD):* A trend-following momentum indicator that shows the relationship between two moving averages of prices.

Moving average plays a more important role than other technical indicators, since its smoothing effect on trend prediction. They are used by researchers in stock market prediction and their success has been proven [8].

In terms of number of input variables, researchers have stated that using much more variable than needed is not a big problem, since sensitivity analysis will overcome this situation by choosing the most sensitive variables.

2.3.4.2. Data Preprocessing

Before the data is presented to the network, it has to be preprocessed appropriately. The form of data is so important, because, as is stated, the neural networks are pattern matchers and will process the data as presented. Data processing can be applied in different ways:

- *Data cleaning* is dealt with filling up missing values, smoothing noisy data, identifying outliers, and correct data inconsistencies.

- *Data normalization* is dealt with converting data into appropriate forms which neural network can deal with. The normalization process is applied in both directions: normalization of input data, and denormalization of output data. Four methods of normalization are described by [9]:

    - *Along Channel Normalization:* This method normalizes each input variable individually. A channel can be defined as a set of elements in the same position over all input vectors in training, validation and test sets. In addition, a channel can thought as an independent input variable.

    - *Across Channel Normalization:* It is done across all input elements in a data pattern.

    - *Mixed Channel Normalization:* This method proposes some kind of combination of along and across channel normalization.

    - *External Normalization:* All the training data are normalized into a specific range.

The choice of which normalization method is going to be used depends on the form of data presented into the network. For time-series forecasting problems, external normalization is appropriate. On the other hand, the normalization can be performed by forecaster or neural network software. It is obvious that using neural network software for normalization saves time, but it also limits the user on trying different normalization strategies. For each type of normalization approach stated above, the following formulae are frequently used:

- *Linear transformation to [0,1]*

$$x_n = \frac{(x_0 - x_{min})}{(x_{max} - x_{min})}$$

- *Linear transformation to [a,b]*

$$x_n = \frac{(b-a)(x_0 - x_{min})}{(x_{max} - x_{min}) + a}$$

- *Statistical normalization*

$$x_n = \frac{(x - \bar{x})}{s}$$

- *Simple normalization*

$$x_n = \frac{(x_0 - x_{min})}{(x_{max} - x_{min})}$$

where,

x_n : value after normalization

x_0 : value to be normalized (original data)

x_{min} : min value along the time-series

x_{max} : max value along the time-series

$\bar{x}$: arithmetic mean of time series

s : standard deviation of time series

a : lower limit for normalization

b : upper limit for normalization

There are several studies have been conducted about the effects of normalization for both classification and prediction problems, and [10] concluded that the normalization is beneficial in terms of Mean Square Error (MSE), but it slows down the training process.

Actually, the normalization function, sometimes referred as squashing function, decision depends on the activation function of corresponding node. This normalization is typically [0, 1] for sigmoid function and [-1, 1] for hyperbolic tangent function, since these functions produce outputs in [0, 1] and [-1, 1] for sigmoid and hyperbolic tangent respectively. On the other hand, observed output of activation function will correspond to the normalized range. To interpret the results, outputs must be denormalized to the original range, since performance analysis is done based on the denormalized data.

2.3.4.3. Sensitivity Analysis

Sensitivity analysis is used to find out which indicator is more sensitive to the outputs. A mentioned before, after sensitivity analysis, less sensitive indicators will be discarded from input data set. In neural networks models used for financial forecasting, the input time-series are believed to be in relationship with other time-series. Sometimes, this relationship is in form of correlation with many other series. So, using these correlated data together does not enhance forecasting abilities. Studies have shown that complex models are having difficulties on beating the simple models. Another frequently used technique for eliminating less sensitive variables is to perform training with different number of input variable. The aim is to perform successive training operations, and detect less sensitive variables based on the comparison results with other models, and to eliminate these variables. In other words, if there is no difference on the performance with or without these variables, these variables are labeled as less significance to the output, and can be discarded from input data set. As stated in [8], is might be expected a performance improvement by doing sensitivity analysis.

Overfitting, also known as overtraining is another major concern closely related with the number of input data presented to the neural network. Overfitting can be described as fitting a statistical model that has too many parameters or complexity of input data is inappropriate. In cases such as this, neural networks tend to memorize rather than generalizing. As a result of overfitting, performance of training increases while test performance becomes worse. In order to avoid overfitting, it is necessary to use additional techniques. The most frequently used technique is cross-validation for early-stopping [11].

Early-stopping concept is an efficient way of dealing with problem of overfitting as mentioned above. In early-stopping training set is divided into two sets: training set and validation set. After each sweep of training set, network is evaluated on the validation set. This process is named as cross validation. Cross validation can be applied in different forms, such as *k*-fold cross validation. Although, the performance of *k*-fold cross validation yields more accurate results [4], however, it is not commonly used in financial prediction.

2.3.4.4. Training, Validation and Test Samples

As a starting point, traditional regression forecasting models methods are based on the assumption that pattern discovered from historical data will hold in future. These models use all available data to build such a forecasting model. However, these models also may not yield acceptable results. On the other hand, when neural network is trained, test data is separated from training data, called as out-of-sample-data. Then, this out-of-sample data is used to test the performance of the model. As stated in previous section, validation set is used to avoid overfitting. Selection of the validation set depends on several factors such as the problem characteristics, the data type and size of available data.

It is critical to have both training and validation sets representative of the population, because, inappropriate selection of training and data sets will affect the performance of neural network [12]. But, what will be the strategy of correct sizes of training, validation and testing data sets. Actually, there is no certain strategy; however, literature provides some guidance in selecting appropriate samples. Most of the studies conducted suggest 70%-20%-10% rule for data set selection [13].

Another important factor is sample size. In any statistical approach, sample size is closely related to required accuracy of the problem. In other words, larger sample size is required in order to determine the hidden relationship between the input and output. With a large enough sample size, ANNs can model complex structures. In comparison with linear models, such as Box-Jenkins, artificial neural networks can yield more accurate results with more data. In case of modeling complex structures, neural networks can benefit from large data sets than linear models.

2.3.5. Neural Network Architecture Specific Issues

This phase is also known as model construction phase. During model construction, a forecaster has to deal with some questions such as: Which neural network structure will be used? What will be number of inputs and outputs? How many hidden layer will be included in the neural network structure? Which activation functions will be applied in each layer? The answers to these questions are basically problem-dependent, although there are different approaches proposed by researchers.

2.3.5.1. Network Architecture

Selection of appropriate network structure is problem-dependent actually. Hence there is unfortunately, no certain strategy for which network structure performs well with any given problem. Therefore, the only guideline is to investigate the literature and to try which one is going to be the best fit for problem in hand. Another decision criterion is the capabilities of the neural network library or software in hand. If the forecaster could develop his own neural network library or software, this limiting factor would be eliminated, but it is obvious that it is a time consuming process and reliability of development should be carefully tested.

Given the various types of neural network architectures that have been developed by researchers, three types of artificial neural network structures are widely used by forecasters in stock market prediction: Multi Layer Perceptron (MLP), Recurrent Neural Networks (RNN), and Generalized Feed forward Neural Networks (GFNN).
MLPs are multilayer feed-forward neural networks which are commonly used in financial forecasting due to their generalization capabilities. In general, a typical

multilayer perceptron (MLP) is described as feed forward neural network consisting of one input, one output and one or more hidden layers, each layer with its own activation function. The neurons located in the same layer use same activation function. The logic of including a hidden layer into neural network structure is to detect feature that both input and output layer cannot. The detection operation is performed by using computational neurons in hidden layer. MLP with one hidden layer and with non-linear function (Figure 2.2) is enough to represent continuous functions [2]. With two hidden layers, a MLP represents even discontinuous functions. The major drawback of MLP is the difficulty to observe the hidden-layer outputs. That explains why the MLP is considered as black-box. A typical MLP contains neurons in a range between 10 and 1000. The number of hidden layers and hidden neurons can be increased, but every hidden layer, and so hidden neurons, may increase the computational burden.

As stated previously, a typical MLP uses back-propagation as learning algorithm. Back-propagation learning is similar to perceptron learning. A training set of input patterns is presented to the network. The network computes the output, and if there is an error, the weights are adjusted to reduce this error. The direction of adjustment is from output layer to hidden layer, and from hidden layer to input layer. In other words, the error calculated is propagated backwards through the network from output layer to input layer. Figure 2.4 illustrates how the back-propagation is performed conceptually.

The indices i, j and k refer to neurons in input, hidden and output layers, w_{ij} and w_{jk} refer to weights between input and hidden, and hidden and output layers respectively. At output layer, error is calculated as

$$e_k(p) = Y_{d,k}(p) - Y_k(p) \tag{2.7}$$

where $Y_{d,k}$ is the desired output of output neurons at iteration p. The weight adjustment, Eq. (1.3), is transformed into

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \tag{2.8}$$

2.3.5.2. Number of Nodes in Each Layer

Number of input nodes maps the number of input variables, both raw data and technical indicators, for forecasting of future values. However, in the literature, there is no consistent conclusion determining the number of input node. However, having too many input variables may cause neural network to learn-by-example not to generalize [14]. In my opinion, number of input nodes is the most crucial decision for a forecaster. Insufficient number of variables may cause the loss of important data about the complex system, so affects prediction accuracy precisely. Using more input nodes is not a big deal, since sensitivity analysis deals with the less sensitive data.

Number of hidden layers plays an important role for successful predictions applications of neural networks in financial areas, and it is also stated that hidden features is detected by the hidden nodes. These hidden nodes can perform complicated nonlinear mapping between input and output variables somehow, and it is clear that a neural network without hidden layer and output nodes with linear activation function acts as a linear predictor. Like deciding on neural network structure, there is no direct rule to decide on number of hidden layers. The most common way in determining the number of hidden layers and the number of hidden nodes is via trial-and-error.

In a single hidden layer structure, more hidden nodes are required for modeling the complex problems such as financial forecasting. As a result, this situation will burden in term of computational time. Moreover, there exist several guidelines on deciding the number of hidden layers. These are $2n+1$, $2n$, n, and $n/2$ where n is the number of input variables. In addition to trial-and-error approach, some researchers have built multiple models with different number of hidden nodes, and have compared their performance [11].

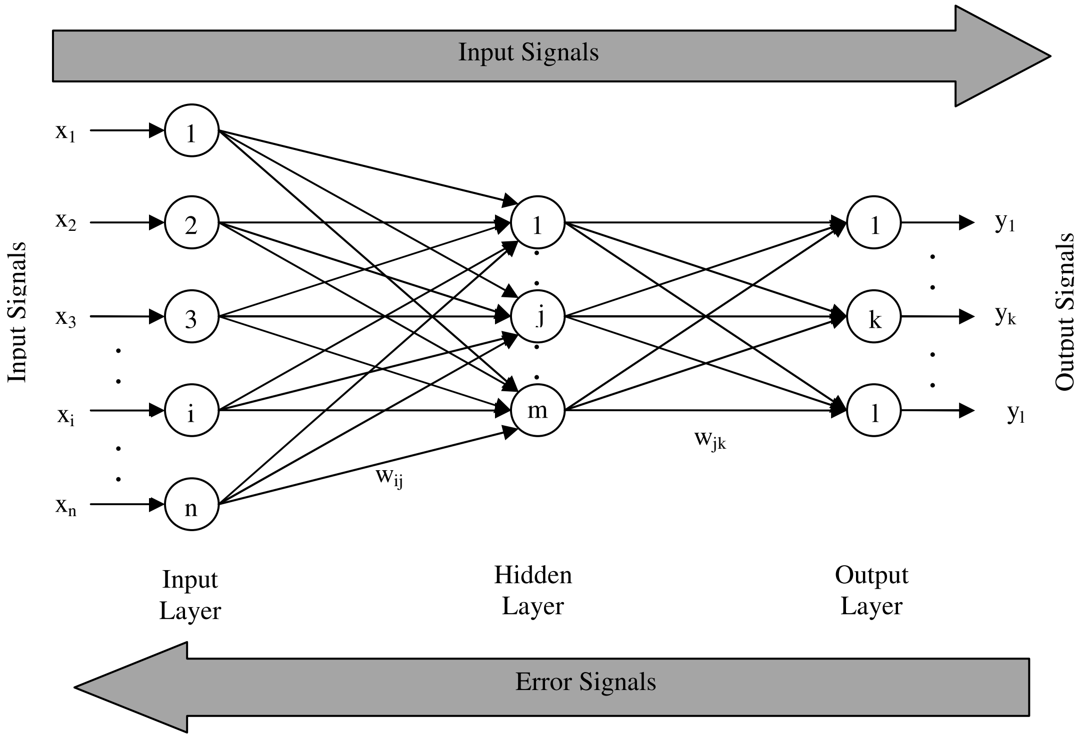

Figure 2.4 Multi layer perceptron with back-propagation

In contrary, neural networks with small number of hidden nodes performs well, and yield relatively accurate results in stock market prediction [5, 11]. Hence, this contradiction between the theoretical and experimental approaches strengthens the trial-and-error hypothesis.

Number of output nodes is the easiest one to determine as it is directly related to the problem in hand. In stock market prediction, the predicted index value is returned; hence the number of output is set to one.

2.3.5.3. Activation Function

As discussed in (Section 1.3.2), activation function of a node defines the output of that node given input variables. In theory there is reasonable number of proposed activation functions, but in practice for financial forecasting there are three well-performed activation functions (Section 1.2); sigmoid (logistic) function, hyperbolic tangent function, and linear function.

Sigmoid function and hyperbolic tangent are the most popular choices among these three for forecasting problems. In general, different nodes in different layers or same layers may have different activation functions, but all nodes in same layer use same activation function. Selection of activation function simply depends on the problem on hand. If the problem is non-linear regression problem, hence the processing elements are utilized with non-linear activation functions (stock market prediction, hyperbolic tangent or sigmoid).

2.3.5.4. Training Algorithm

Network training is described as non-linear minimization problem by reducing weights to minimize the MSE or Sum Squared Error (SSE). Despite the fact, that there is no training algorithm which offers maximum optimization. However, literature shows some evidence. The researches state that back-propagation algorithm has become a standard for financial markets [14].

Backpropagation is extended form of perceptron learning rule. In perceptron learning rule, delta element is calculated using the input signal x_i. But in MLP, we cannot use x_i. Instead of x_i, we will use output of hidden neuron, y_j. The weight correction for output layer neurons is described as follows:

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p) \tag{2.9}$$

where δ_k(p) is the error gradient. Since the aim of learning algorithm is to minimize the error gradient, by taking partial derivatives, is used to reach the local minima. As mentioned before, the most widely used activation function in MLP is a form of logistic function, called sigmoid function, since to get the derivative of sigmoid function is simple. The gradient error for neuron k is

$$\delta_k(p) = \frac{\partial y_k(p)}{\partial X_k(p)} \times e_k(p) \tag{2.10}$$

where y_k is the output of neuron k, and X_k is the net input to neuron k. The activation function is selected as sigmoid function. Thus, we obtain:

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \tag{2.11}$$

where

$$y_k(p) = \frac{1}{1 + exp[-X_k(p)]} \tag{2.12}$$

In case of hidden layer weight correction, the perceptron learning algorithm is modified again. The delta rule for the hidden layer is

$$\Delta w_{ij}(p) = \alpha \ \times x_i(p) \times \delta_j(p) \tag{2.13}$$

where $\delta_j(p)$ refers to error gradient for hidden layer neuron

$$\delta_j(p) = y_k(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) w_{jk}(p) \tag{2.14}$$

where l is the number of hidden neurons at hidden layer

$$y_j(p) = \frac{1}{1 + e^{-X_j(p)}}$$

$$X_j(p) = \sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j$$

Next, derived back-propagation training algorithm can be described as

Step 1: Initialization

Sell all the weights and threshold values in network to small random numbers.

Step 2: Activation

Start activation feeding network with input values $x_1, x_2, \ldots, x_n$ and applying desired outputs $y_{d,1}(p), y_{d,2}(p), \ldots, y_{d,n}(p)$.

i. Calculate actual outputs of neurons in the hidden layer.

$$y_j(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j\right]$$

ii. Calculate actual outputs of neurons in the output layer.

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}(p) - \theta_k\right]$$

Step 3: Weight training

Propagate the errors calculated at output layer backwards.

i. Calculate error gradient in the output layer

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p)$$

where

$$e_k(p) = Y_{d,k}(p) - Y_k(p)$$

calculate weight corrections

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$$

update the weights in the output layer

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

ii. Calculate error gradient in the hidden layer

$$\delta_j(p) = y_k(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) w_{jk}(p)$$

calculate weight corrections

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$$

update the weights in the hidden layer

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

Step 4: Iteration

Increase iteration *p* by one and loop back to step 2, until the minimum error criterion is satisfied.

The error criterion mentioned at step 4, is a useful performance indicator for neural network. Since, the aim of neural network training algorithm is to minimize the error rate; it is the point where to stop training for neural network. One of commonly used error rate indicator is sum squared error (SSE). SSE is given by the formula,

$$SSE = \sum_{i=1}^{n} \left(y_d(p) - y_p\right)^2 \tag{2.15}$$

One of the major drawbacks of back-propagation is expensive training phase. The calculations in the hidden layer require more time. Hence, the in practical applications, it is infeasible to use a pure back-propagation algorithm. In order to accelerate the calculations, a derivative of sigmoid activation function, called hyperbolic tangent, is represented,

$$Y^{tanh} = \frac{e^{-x} - e^{-x}}{e^{-x} + e^{-x}} \tag{2.16}$$

A momentum term is proposed in the delta rule of Eq.2.9 to accelerate the training. [15]

$$\Delta w_{jk}(p) = \beta \times \Delta w_{jk}(p-1) + \alpha \times y_j(p) \times \delta_k(p) \tag{2.17}$$

where β is a positive number between ($0 \leq \beta < 1$). Thus, we obtain a generalized delta rule in Eq.2.17. When $\beta=0$ we obtain Eq.2.9. Observations made on momentum term showed that momentum term has a stabilizing effect on back-propagation algorithm. When it is applied to the delta rule, number of iterations, epochs, decreases moderately.

Many different ANN models have been proposed, but most influential models are Hopfield Networks (Recurrent Neural Networks), Generalized Regression Neural

Networks (GRNN), Probabilistic Neural Networks and Kohonen's self organizing networks. In the rest of this work, our focus will be on the Multilayer Feed Forward Neural Networks. The MFNNs are used in a variety of problem especially in forecasting and prediction with relatively good performance.

2.3.5.5. Performance Measures

Although there are many performance measures for neural networks, the most important measure of performance is the prediction accuracy in training data. This accuracy measurement is usually defined as prediction error, described as the difference between actual and predicted values. There are various numbers of measures, also called error functions, of accuracy for financial prediction with their advantages and disadvantages.

- *Sum of Squared Error (SSE):* SSE is the most widely used measure of the differences between values predicted and the actual values before estimation. These individual differences are then aggregated into a single measure of prediction performance. SSE is defined as follows:

$$SSE = \sum (y_d - y)^2$$

- *Mean Squared Error (MSE):* MSE is actually some form of SSE, but with one difference that sum of errors is divided by number of observations, named as degrees of freedom.

$$MSE = \frac{\Sigma (y_d - y)^2}{N}$$

- *Root Mean Squared Error (RMSE):* RMSE is defined as

$$RMSE = \sqrt{MSE}$$

- *Mean Absolute Percentage Error (MAPE):* MAPE is measure of accuracy in a time series value in statistics, specifically trending. It usually expresses accuracy as a percentage. Hence, the network performance is widely expressed in terms of MAPE.

$$MAPE = \frac{1}{N}\sum\left|\frac{y_d - y}{y_d}\right|(100)$$

where,

y_d : desired output

y : actual output

N : number of error terms (or number of observations)

- *Hit Rate (HR):* Hit rate is described as the percentage of successfully predicted samples in proportion to all data reserved as prediction set. HR is a measurement used for prediction of stock market index value.

As mentioned before, MSE and MAPE are the most frequently used performance measure and stopping criteria for neural network training. Since each error function come with its own advantages and disadvantages, using a single measure in a particular problem is not the best way to judge the performance of neural network. Performance issues are going to be discussed in "Validation of Neural Network Performance" section in details.

2.3.6. Performance Validation Issues

Up to this point, we have discussed the general structure of neural networks and criteria to be considered implementing a NN model for prediction tasks. In addition, we have also tried to illustrate how neural networks have served well for prediction tasks focusing on stock market. The studies represented here and others in literature not mentioned here illustrates NNs potential in financial area. On the other hand, some researchers argue NNs' potential and suggest that stronger evidence is necessary. To end up the confliction, evaluation criteria of how NNs contributes improvement in the accuracy of prediction in financial area have been proposed [16]. These evaluation criteria are called effectiveness of validation.

Effectiveness of validation looks for an answer to the question, what do these techniques contribute to individual as forecasters? The study also represents three guidelines which can be applicable to any NN models. Three criteria of effectiveness of validation are as follows:

- *Comparison with well-accepted models*: Predictions from proposed model should be as good as well-accepted reference models. If it does not perform as accurate as these models, it cannot be argued that model can contribute the knowledge about the trend. Some well-accepted models widely used in this step are multiple linear regression, non-linear regression, moving averages, and Box-Jenkins (ARMA) model.

- *Use of out-of-sample performance*: Comparisons of forecasts should be based on *ex ante* (out-of-sample) performance. The data used in training phase and data to be used in test phase. Like real world situation, the trained network should not see the future values to be predicted.

- *Use of a reasonable sample size*: Sample size used in prediction should be large enough for effective learning.

Chapter 3

Predicting ISE100 Index with Neural Networks

3.1. Introduction

As stated in previous chapter, prediction of stock market is an important issue in financial area and ANNs have been widely used in stock market prediction with an acceptable accuracy. While these studies have focused on individual stocks, in Turkey studies examined majored on stock market index level estimation and predicting the direction of stock market index value. This experiment examines the forecasting power of ANNs for Istanbul Stock Exchange 100 Index (ISE100) in predicting the stock index direction, since sign of the direction gives more information and guide investors to direct their funds to more profitable investment instruments. Relying on the performance of MLPs trained with back-propagation, experiment is built on this structure. Network performance is validated by comparing neural network results with multiple linear regression and non-linear regression models. For neural network implementation "NeuroIntelligence" (by Alyuda Software Co.) package, proven software owing to its great visualization and automated data transformation capabilities, and for non-linear regression analysis "DataFit" (Oakdale Engineering) software are used.

3.2. Stock Market Applications of ANNs in Turkey

Although ANNs are most widely used in predicting financial failures in Turkey, there is reasonable number of applications in stock market prediction. One of the recent models has been built for ISE index value prediction [10]. Study uses an MLP with back-propagation as network structure. The data for represented to the model, historical data between July, 2001 and February, 2003, seems to be relatively small.

Organization of the data is based on 90%-10% rule, 90% as training set, 10% as testing set and no data set for validation purposes. In terms of performance issues, in addition to validation with well-known models such as Moving Averages (MA), researchers build three different models with 1, 2 and 4 hidden layers respectively, and included these models to comparison. Hence, they have a chance to compare the performance of two models with different number of hidden layer and processing units. Performance of the models is pretty good. All of three models outperform the moving average approach in level estimation task. While three models yield outputs with a MAPE (%) at a rate of 1.62, 1.65 and 1.70 respectively for models with 1, 2 and 4 hidden layer, MA performs with a MAPE (%) at a rate of 2.17 and 3.03. The study has showed that MLP trained with back-propagation outperformed MA approach. Moreover, performance of MFNN diminishes as the number of hidden layers increases.

A prediction model is presented for stock market index level estimation based on MLP trained with conjugate gradient algorithm [11]. Historical data used in this study corresponds to the period between 1996 and 2005, and is presented to the network based on 70%-20%-10% strategy, 70% for training set, 20% for validation set, and 10% for testing set. This study also builds different network structures with different number of processing units, but validation with well-known models is not included. Hence, it cannot be said that the proposed model performs better than traditional regression models. Instead of it, performance validation is performed by comparing these three models. Research also includes a sensitivity analysis and displays the effects of it in terms of higher financial gain when compared with the performance of network before sensitivity analysis.

MLP performance is investigated for stock market level estimation in crisis periods between 1 February, 2001 and 28 February, 2001 in Turkey [4]. Study achieves a good chance of obtaining a 73.68% accuracy rate at predicting the direction of stock market index value, even in crisis periods.

Among other studies examines, MLP models trained with back-propagation algorithm are used to estimate the direction of stock market index value, and study shows that direction of ISE100 stock index could be predicted at a rate of 60.81%

[12]. Research is also conducted on building a MLP model for ISE30 level estimation and predicting the direction of ISE30 stock index value [13]. After comparing NN results with the results of regression analysis, validation shows while MLP outperforming regression analysis in estimation of direction, regression analysis performs better than MFNN in level estimation.

Results of the studies discussed above shows MLPs performs pretty well in stock market index level estimation, and well predicting the direction of stock market index value in a highly volatile environment, such as ISE.

3.3. Data Set

Experimental data use in this experiment is gathered from Central Bank of the Republic of Turkey (CBRT). Data consists of previous day's ISE100 trading volume, previous day's gold market index, previous day's USD/YTL exchange rate. Financial data used in experiment corresponds to the period between 02.01.2002 and 31.07.2008 (Figure 3.1). There are inconsistencies in data since trading days for ISE and İstanbul Gold Exchange (IGE). So, the tuples with missing values were filled up with the mean of time-series. Therefore, we had 1,661 tuples which may be considered enough in size, represented by 3 input variables (Table 3.1).

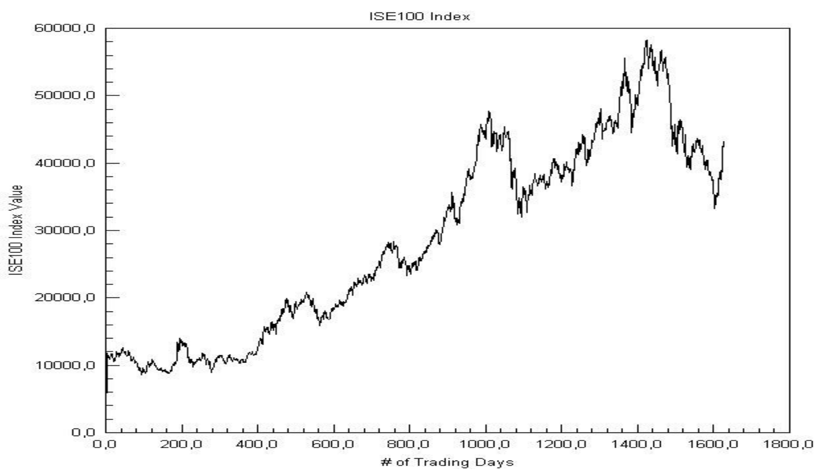

Figure 3.1 ISE100 index plot diagram (January 2002 –August 2008)

Table 3.1 List of input variables

1. Trade volume for ISE100 index
2. Previous day's USD/TYL exchange rate
3. Previous day's gold index value

Next, all raw input data is normalized into a range [-1, 1] as owing to the nature of hyperbolic tangent function. Finally, data sets are partitioned into training set, validation set and testing set based on 70%-20%-10% partitioning strategy except the data of 23 trading days, reserved for prediction period (Table 3.2).

Table 3.2 Number of sample data for daily data.

	2002-2008
Sample	
Training (70%)	1149
Validation (20%)	325
Testing (10%)	164
Prediction	23

3.4. Network Structure

Due to its prediction capabilities in stock market prediction discussed in previous sections, neural network structure is selected as a 3-layer MLP, and back-propagation is used as learning algorithm. An MLP with 1 hidden layer, 7 processing (hidden) elements, and 1 output element is utilized. Sigmoid function is selected as activation function for neurons in hidden and output layer. Network parameters such as learning rate and momentum term are used in an adjustable manner, as discussed in "Activation Function" section, but initial parameters, learning rate and momentum term, were selected as 0.2 and 0.3 respectively, relying to previous experiments. MSE, value of 0.0002 (the point there is no improvement in error correction), is also used as stopping criterion.

3.5. Empirical Findings

Even the stopping criteria is assigned initially, our neural network structure has not reached this point. However, due to the nature of selected experimental data, training stopped at the point where the network error improvement gets diminished instead of stopping criteria (MSE=.0002). Even, the neural network performance is better in testing and validation periods, prediction results are not as expected. The hit rate in percentage (number successful predictions/number of samples to be predicted, in percentage) is determined as 54.45%.

Known statistical methods are also applied to the experimental data for performance validation issues. We have tried to model the data for, 3 independent and one dependent variable, using multiple linear regression and non-linear regression approaches. According to the results gathered from these approaches, the direction prediction percentage is at 42.01% for multiple linear regression (Eq.3.1) and 52.01% for non-linear regression (Eq.3.2). The equations used in regression are as follows.

$$y = aX_1 + bX_2 + cX_3 \tag{3.1}$$

$$y = aX_1^2 + bX_2^2 + cX_3^2 + dX_1X_2 + eX_1X_3 + fX_2X_3 + gX_1X_2X_3 \tag{3.2}$$

The results presented here are the findings for the prediction periods and they are shown in Table 3.3.

Table 3.2 Hit rate (%) results gathered from all models.

Model	Hit Rate (Direction)
MFNN (3-7-1)	54.45
Non-Linear Regression	52.01
Multiple Linear Regression	42.01

A second experiment is also conducted to increase the prediction accuracy of neural network. Original data set is expanded by including InterBank Overnight Interest Rate (ON) data as an alternative financial investment instrument. Neural network is trained with this new data set, and prediction is performed for July 1, 2008 – July 31, 2008 period. However ON data changes infrequently. In other words, ON data is same for long periods (e.g. for period between Jan 1, 2002 and Feb 19, 2008, ON is 59%). For this reason, prediction results have shown that including ON data does not affect the prediction capabilities of our neural network structure. Hit rate determined by MLP remains at 54.45%.

On the other hand, our new data set cannot be solved traditional non-linear regression methods. Moreover, it is solved by linear regression model given by Eq.3.3.

$$y = aX_1 + bX_2 + cX_3 + dX_4 \tag{3.3}$$

F value in ANOVA results (Table 3.4) indicates that linear regression is far from representing the trend.

Table 3.4 ANOVA results for linear regression.

Source	**DF**	**Sum of Squares**	**Mean Square**	**F Ratio**	**Prob (F)**
Regression	4	282460870094,429	70615217523,6072	1424,412372	0
Error	1657	82145744979,3673	49574981,8825391		
Total	1661	364606615073,796			

3.6. Conclusions for Experiments

These experiments have been designed to predict the direction of the ISE100 index using neural networks. In this manner, a multi layer perceptron trained with back-propagation is utilized and trained using two different data sets. In addition, proven traditional forecasting models are utilized in order to validate the results gathered from neural network.

The remarkable finding of both experiment, is that the performance of neural network is better than traditional forecasting methods for our experimental data, but not at expected level (inconsistent with literature). But as illustrated in [17], neural networks equipped with sigmoid function, or its derivative, behaves as non-linear regression models. Hence, the performance of NNs could be expected as same as the non-linear regression when the input variable selection is made based on limited financial background.

However, even NN performance is not at expected level in this experiment, it has still shown that a neural network performs better than both linear and non-linear regression methods for multivariate time series problems with non-linear structure.

4. Conclusions and Recommendations for Future Work

This study has been designed to investigate the forecasting abilities of artificial neural networks in stock market. The factors making the stock market prediction is a hot topic, are investigated. Next, mathematical and statistical model behind the neural network theory is given, and issues related with neural network design for stock market prediction are listed with reasons. This literature review has also revealed that, MLP with sigmoid activation function and trained with backpropagation is dominant neural network structure among other neural network models owing its generalization capabilities.

In this manner, an MLP model is built to investigate the prediction performance of neural networks on Istanbul Stock Exchange 100 (ISE100) index in terms of accurate direction prediction. The proposed network is trained with backpropagation. Prediction results show that MLP performs somewhat better that traditional time-series forecasting methods such as multiple linear regression and non-linear regression. The most important finding is that, while neural network outperforms better than multiple linear regression, its performance is slightly better than non-linear regression. Although the performance of proposed network should be at least as traditional methods is satisfied, the yielding results are not as expected.

Probability of 0.54 for prediction of trend in stock market should be considered carefully. This result should be interpreted after a sophisticated awareness analysis. Even the rate of probability is lower; it may be a winning probability in some cases. However, proposed system performance cannot be compared with any other proposed system, even if they are proposed as predictors for the same financial market. Moreover, this study emphasizes importance of input selection process. It is obvious that this process requires good economical background, since it directly influences the performance of neural network model.

In addition, in case of stock market prediction, linear and non-linear regression models should be supplied with lagged market data. The reason is that investor would like to take these short term indicators into consideration for making investment decision. Their contribution to the performance improvement for financial prediction is highlighted in studies [11, 18]. Including these time-series into our data set may yield better prediction results and this issue will be considered for future developments. Moreover, 70% of investors in Turkish stock market are foreign investors. According to this information, input data set should also include the variables such as situation of other financial markets and foreign exchange rates considered by foreign investors. The new data set including these variables may be expected to perform better. The future emphasis should also be placed on these variables.

References

[1] Zhang, D., Zhou, L., "Discovering Golden Nuggets: Data Mining in Financial Application", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol.34, no.4, pp.513-522, Nov. 2004.

[2] Hornik, K. Stinchcombe, M., White, H., *Multilayer Feed Forward Networks Are Universal Approximators*, Neural Networks, vol. 2, no. 5, pp. 359-366, 1989.

[3] Lawrence, R., *Using Neural Networks to Forecast Stock Market Prices*, Department of Computer Science, University of Manitoba, December 12, 1997.

[4] Akel, V., Bayramoglu, M. F., "Financial Predicting with Neural Networks In Crisis Periods: An Evidence from ISE 100 Index", *International Symposium on International Capital Flows and Emerging Markets*, April 24-27, 2008, Balikesir, Bandırma.

[5] Negnevitsky, Michael, *Artificial Intelligence: A Guide to Intelligent Systems*, 2nd Edition, Pearson Education, England, 2005.

[6] Azoff, E. M., 1994, *Neural Network Time Series Forecasting of Financial Market,* John Wiley and Sons, Chichester.

[7] Zhang, G., Patuwo, B. E., Hu, M. Y., "Forecasting with Artificial Neural Networks: The State of Art", 1998, *International Journal of Forecasting*, 14, 35-62.

[8] Dutta, G., Jha, P., Laha, A. K., Mohan, N., "Artificial Neural Network Models for Forecasting Stock Price Index In The Bombay Stock Exchange", *Journal of Emerging Market Finance*, 5:3, (2006)

[9] Shanker, M., Hu, M.Y., Hung, M.S., 1996, *Effect of Data Standardization on Neural Network Training*, Omega 24 (4), 385-397.

[10] Egeli, B., Özturan, M., Badur, B. "Stock Market Prediction Using Artificial Neural Networks", Proceedings, *3rd Hawaii International Conference on Business*, USA, June 18-21, 2003.

[11] Avcı, E., “Forecasting Daily and Sessional Returns of the ISE-100 Index with Neural Network Models”, *Dogus University Journal*, 8(2) 2007, 128-142.

[12] Diler, A. İ., “İMKB Ulusal 100 Endeksinin Yönünün Yapay Sinir Ağları: Hata Geriye Yayma Yöntemi ile Tahmin Edilmesi”, *İMKB Dergisi*, 7, 65–81.

[13] Altay, E., Satman, M. H. (2005), “Stock Market Forecasting: Artificial Neural Networks and Linear Regression Comparison in an Emerging Market”, *Journal of Financial Management and Analysis*, 18 (2), 18-33.

[14] Baum, E. B., Hassler, D., *What Size Net Gives Valid Generalization?*, Neural Computation, 1, (1989), 151-160.

[15] Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, (Vol. 1), pp. 318-362, Cambridge: The MIT Press, 318-362.

[16] Adya, M., Collopy, F., “How Effective Are Neural Networks at Forecasting and Prediction? A review and evaluation”, *Journal of Forecasting*, 17, 481-495, (1998).

[17] Sarle, W. S., *Neural Networks and Statistical Models,* Proceedings of the Nineteenth Annual SAS Users Group International Conference, April, 1994.

[18] Zorin, A., Borisov, A., “Modelling Riga Stock Exchange Index Using Neural Networks”, Proceedings of the international conference *Traditional And Innovations In Sustainable Development Of Society*, Rezekne, Latvia, February 28 - March 2, pp.312-320, 2002.

Curriculum Vitae

Onur Görgün was born on 14 December 1980, in İstanbul. He received his BS degree in Information Technologies in 2005 from Işık University. He has been a research assistant at the department of Information Technologies of Işık University from 2005. His research interests include database management systems, data mining and machine learning.