REPUBLIC OF TURKEY
ÇUKUROVA UNIVERSITY
THE INSTITUTE OF SOCIAL SCIENCES
DEPARTMENT OF ECONOMETRICS

ESTIMATING SESSION RETURNS OF BORSA ISTANBUL 100 INDEX BY INTEGRATING MARKOV CHAINS AND ARTIFICIAL NEURAL NETWORK MODELS

Çiğdem KOŞAR

MASTER OF ARTS

ADANA / 2013

REPUBLIC OF TURKEY
ÇUKUROVA UNIVERSITY
THE INSTITUTE OF SOCIAL SCIENCES
DEPARTMENT OF ECONOMETRICS

ESTIMATING SESSION RETURNS OF BORSA ISTANBUL 100 INDEX BY INTEGRATING MARKOV CHAINS AND ARTIFICIAL NEURAL NETWORK MODELS

Çiğdem KOŞAR

Supervisor: Assoc. Prof. Dr. Süleyman Bilgin KILIÇ

MASTER OF ARTS

ADANA / 2013

To the Directorship of the Institute of Social Sciences of Çukurova University,

We certify that this thesis is satisfactory for the award of the degree of Master of Arts in the Department of Econometrics.

Head of Committee: Assoc. Prof. Dr. Süleyman Bilgin KILIÇ
(Supervisor)

Member of Examining Committee: Prof. Dr. Fatih CİN

Member of Examining Committee: Assist. Prof. Dr. Ersin KIRAL

I certify that this thesis conforms to the formal standards of the Institute of Social Sciences.…/……/ 2013

Prof. Dr. Azmi YALÇIN
Director of Institute

ABSTRACT

ESTIMATING SESSION RETURNS OF BORSA ISTANBUL 100 INDEX BY INTEGRATING MARKOV CHAINS AND ARTIFICIAL NEURAL NETWORK MODELS

Çiğdem KOŞAR

Master Thesis, Department of Econometrics
Supervisor: Assoc. Prof. Süleyman Bilgin KILIÇ
August 2013, 74 Pages

The objective of the study is to estimate the direction of next session return and research the investment opportunities of the Borsa Istanbul 100 (BIST 100) index when passing from one session to another. In this study, we first modeled session returns of BIST 100 index as the discrete state Markov chain process, and second we trained an Artificial Neural Network (ANN) model in order to estimate direction of session returns. At the end of the study we gave some information to the investors according to findings of research and gave some advices to other researchers about the future research perspectives.

Keywords: Artificial Neural Networks, Markov Chains, Conditional probability, BIST 100.

ÖZET

BORSA İSTANBUL 100 ENDEKSİNE AİT SEANS GETİRİLERİN MARKOV ZİNCİRLERİ VE YAPAY SİNİR AĞLARI MODELLERİYLE TAHMİN EDİLMESİ

Çiğdem KOŞAR

Yüksek Lisans Tezi, Ekonometri Anabilim Dalı
Danışman: Doç. Dr. Süleyman Bilgin KILIÇ
Ağustos 2013, 74 Sayfa

Çalışmanın amacı, bir seanstan diğerine geçiş esnasında gelecek seansın bir önceki seansa göre yönünü tahmin etmek ve Borsa İstanbul 100 (BİST 100) endeksine ait yatırım olanaklarını araştırmaktır. İlk olarak, BİST 100 endeksine ait seans getirileri Markov zincirleri süreciyle kategorilere ayrılmış ve daha sonra seans getirilerinin yönünü tahmin etmek amacıyla bir Yapay Sinir Ağları (YSA) modeli eğitilmiştir. Çalışmanın sonunda, araştırmanın bulgularına göre yatırımcılara bazı bilgiler verilmiş ve araştırmacılara bu alanda yapılabilecek benzer çalışmalar ile ilgili tavsiyelerde bulunulmuştur.

Anahtar Kelimeler: Yapay Sinir Ağları (YSA), Markov Zincirleri, Koşullu Olasılık, BİST 100.

ACKNOWLEDGEMENTS

First I would like to express my deepest and most sincere gratitude to my thesis supervisor Assoc. Prof. Dr. Süleyman Bilgin KILIÇ for his comments, support, professional advices and giving his valuable time during the preparation of this thesis.

I am grateful to my thesis committee members Prof. Dr. Fatih CİN and Assist Prof. Dr. Ersin KIRAL for their valuable review and comment on the thesis.

I would like to express my gratitude to our Head of Department Prof. Dr. H. Altan ÇABUK and also, i want to thank all other members of the Department of Econometrics for their sincerity and support.

I am very grateful to my mother Şefika KOŞAR and my father Ahmet KOŞAR for their innumerable contributions to my life, their endless love and prayers. I love them all very much.

Lastly, I am extremely grateful for all the encouragement I have received from, Erhan TAŞ, and I would like to express my gratitude for his very important and everlasting support, patience and love.

This thesis was supported by Çukurova University Scientific Research Fund under the project number İİBF2012YL8.

To my Older Sister, ÖZLEM KOŞAR

TABLE OF CONTENTS

CHAPTER III

MARKOV CHAINS

CHAPTER IV

THE SUMMARY OF EMPRICAL STUDIES

CHAPTER V

APPLICATION

CHAPTER VI

RESULTS AND FUTURE RESEARCH PERSPECTIVES

LIST OF ACRONYMS

ADALINE : Adaptive Linear Neuron
AI : Artificial Intelligence
ANN : Artificial Neural Network
ANNs : Artificial Neural Networks
ARCH : Autoregressive Conditional Heteroskedasticity
ARMA : Autoregressive Moving Average
ART : Adaptive Resonance Theory
BIST 100 : Borsa Istanbul 100
BSE : Bombay Stock Exchange
DJIA : Dow Jones Industrial Average
EMH : Efficient Market Hypothesis
FTM : Fuzzy Transformation Model
GA : Genetic Algorithm
GAFD : Genetic Algorithm Feature Discretization
GARCH : Generalized Autoregressive Conditional Heteroskedasticity
Gas : Genetic Algorithms
GTM : GA-based Feature Transformation Model
KOSPI : Korea Stock Price Index
LSE : London Stock Exchange
LVQ : Learning Vector Quantization
MAE : Mean Absolute Error
MCMC : Markov Chain Monte Carlo
MLP : Multi Layer Perceptron
MS-MA : Markov Switching- Moving Average
MSE : Mean Squared Error
NMSE : Normalized Mean Squared Error
NYSE : New York Stock Exchange
OLS : Ordinary Least Squares
PDPs : Parallel Distributed Processing Systems
PPN : Probabilistic Network
PR : Pattern Recognition
RMSE : Root Mean Squared Error

SHSE : Shangai Stock Exchange
SOM : Self Organizing Map
S&P 500 : Standard and Poor’s 500
SV : Stochastic Volatility
TL : Turkish Lira
TOPIX : Tokyo Stock Exchange Prices Index
UK : United Kingdom
US : United States
USD : United States Dollar

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

1.1. The Subject of Study

A stock market is a legal public entity that trades company stocks and other capital market instruments according some rules in confidence and stability.

The stock is each one of the equivalent parts of a corporation’s principal. People who buy stocks (shares) of a company become a shareholder of that company. At the same time, this means that, these people participate to the company’s profit and loss. For this reason, these shareholders want to estimate the direction of stock returns. But there is no way to determine the behavior of the stocks accurately because of the high volatility. Therefore, many researchers have done studies on prediction of the stock returns in many stock markets around the world.

Financial analysts use various methods to determine the stock return movements. The most important ones of these methods are Artificial Neural Networks (ANNs) and Markov chains.

One of the early studies on prediction the stock return movements with Markov chains was presented by Ryan (1973). He explained the relevance of the theory of Markov process to the analysis of stock return movements and showed that Markov theory is seen to be relevant to the analysis of stock returns in two ways.

Besides the studies on prediction the stock return movements with ANNs have been done about twenty three years. One of the first studies was presented by Kimoto, Asakawa, Yoda and Takeoka (1990) on Tokyo Stock Exchange.

In developing countries such as Turkey, it is very hard to estimate the stock market returns because of many kinds of speculative movements in markets. That’s why prudential estimations on the finance sector have become more frequent over the recent years in Turkey. Stock market has an important role in finance sector in our country (Karaatlı, Güngör, Demir, Kalaycı, p. 38).

In this thesis, we tried to estimate the direction of next session return and investment opportunities of the Borsa Istanbul (BIST) 100 index when passing from one session to another. We use Markov chain process to model session returns of BIST 100 and then, we trained an ANN model in order to estimate direction of session returns. At the end of the application Markov chains and ANNs gave effective results.

1.2. The Aim of Study

The objective of the study is to estimate the direction of next session return and research the investment opportunities of the Borsa Istanbul 100 (BIST 100) index when passing from one session to another. For this purpose, we tried to estimate the direction of session returns. We first modeled session returns of Borsa Istanbul 100 index as the discrete state Markov chain process, and second we trained an Artificial Neural Network (ANN) model. At the end of the study we gave some information to the investors according to findings of research and gave some advices to other researchers about the future research perspectives.

1.3. The Plan of Study

The study consists of six chapters including introduction and result parts.

The first chapter is introduction part and consists of the subject of study, the aim of study and the plan of study.

In the second chapter of the study, it was tried to explain Artificial Neural Networks with their description, general properties, advantages, disadvantages and using areas. Furthermore, historical development of ANNs was presented in this chapter and training system of ANNs was shown. Finally, classification of Artificial Neural Networks was examined under the three groups.

In the third chapter of the study, we gave some information about Markov Chains. We presented the transition probabilities of Markov chains and their classification of states as irreducible, recurrent, transient, absorbing, periodic and aperiodic states. Also, we examined the long run behavior of Markov chains.

In the fourth chapter of the study, it was given place to some empirical studies about Artificial Neural Networks and Markov chains on stock markets which were done in Turkey and other countries.

In the fifth chapter of the study first, we modeled session returns of Borsa Istanbul 100 index between the periods of January 04, 1988-April 04, 2012 as the discrete state Markov chain process, and second we trained an Artificial Neural Network (ANN) model so as to estimate direction of session returns.

Lastly, in the sixth chapter of the study, we gave valuable information to the investors about the direction of next session return and investment opportunities of the BIST 100 index when passing from one session to another.

CHAPTER II

ARTIFICIAL NEURAL NETWORK

2.1. Introduction to Artificial Neural Network

Under the science of artificial intelligence (AI), neural network is a research area that researchers show very intense interest. ANN includes works on learning of computers.

The problems that can not be created its formulation and can not be solved are found out with computers by way of heuristic methods. The studies which equip computers with these properties and lead the development of these abilities of theirs are known that artificial intelligence (Oztemel, 2012, p. 13).

AI is a sub-branch of computer science which uses information processing methods like data structures, algorithms, programming languages and techniques in order to save and operate information and it researches automation of behavior of human intelligence (Akpınar, 1994, p. 43).

Artificial Neural Network is one of several areas that results from artificial intelligence works and contributes to this works. In this section, we will present Artificial Neural Network in general and explain fundamental element of it.

2.1.1. Description and Main Task of Artificial Neural Network

An Artificial Neural Network is an information-processing system that has certain performance characteristics in common with biological neural networks. ANNs have been developed as generalizations of mathematical models of human cognition or neural biology, and the assumptions about the ANNs are that:

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which in a typical neural net, multiplies the signal transmitted.

4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A neural network is characterized by the first assumption its pattern of connections between the neurons, called its architecture; by the second assumption its method of determining the weights on the connections, called its training, or learning, algorithm and by the third assumption its activation function (Fausett, 1994, p. 3).

ANNs are computer systems that they can derive and create new information without any help via learning which one of the properties of human brain. It is very hard or not possible to make these abilities with traditional programming languages. Therefore, it can be said that ANN is a branch of computer science which is developed for events like can not be done or very difficult to programming (Oztemel, 2012, p. 29).

ANNs, also called parallel distributed processing systems (PDPs) and connectionist systems, are intended for modeling the organizational principles of the central nervous system, with the hope that the biologically inspired computing capabilities of the ANN will allow the cognitive and sensory tasks to be performed more easily and more satisfactorily than with conventional serial processor. (Base & Liang, 1996, p. 1).

Above, ANNs are defined different forms. There are several common points of these definitions. First of these, Artificial Neural Networks consist of artificial cells connected each other with hierarchically and can work parallel. These cells are called process elements. It is accepted that these cells connect one another and each connection has a value. The network, result of connecting with each other of process elements is called Artificial Neural Network. This network has certain performance characteristics in common with biological neural network (Oztemel, 2012, p. 30).

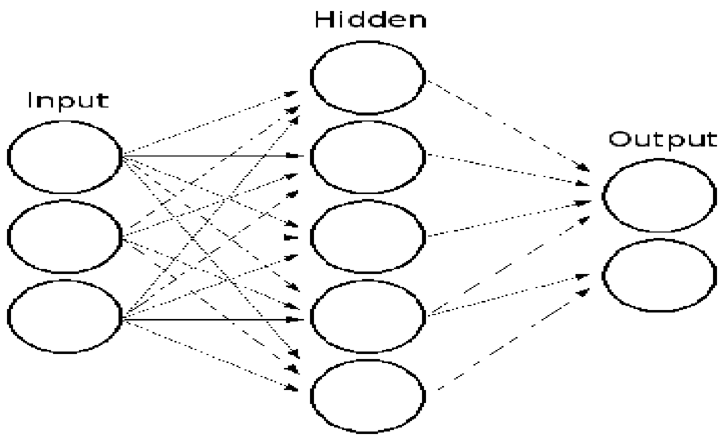


Figure 1. Example of an Artificial Neural Network
Source: http://psychology.wikia.com/wiki/Artificial_neural_network.htm. (Access Date: 4.12.2012).

Technically, determining an output set correspond to input set which is shown to it is main task of an ANN. For this, network is trained with examples of related events. So, network regains its ability to be able to generalize. With this generalization, output sets for similar events are determined. (Oztemel, 2012, p. 30)

2.1.2. General Properties of Artificial Neural Networks

- Artificial Neural Networks perform machine learning.
- Their programs and other programming methods that are known work style are alike.
- In ANNs, information is measured with values of network links and is saved in links. But in other programs' data are presented in data base or in program.
- ANNs learn through examples. It is necessary to determine examples about events, so ANNs can learn events. They are reached the ability to be able to generalize the events by using examples.
- It is necessary to educate and examine their performances firstly to run ANNs with confidence. Initially, examples must show to link so link can determine relations between the events.
- They can produce information about unprecedented examples.

- They can be used in events about sensation.
- They can do classification and association of pattern.
- ANNs can make pattern complete.
- They have abilities like self-organize and learning.
- They can work with deficient information.
- ANNs have fault tolerance.
- They can operate the information which is not proper and indefinite.
- They show graceful degradation. A network deteriorates slowly over time.
- They have distributed memory. In ANNs, the information spreads to the network.
- They only can work with numerical information. Symbolic information should be expressed in numerical information.

Through this information, ANNs are the most powerful problem-solving technique that can operate deficient, abnormal and indefinite information. (Oztemel, 2012, p. 31-33)

2.1.3. Advantages and Disadvantages of Artificial Neural Networks

ANNs are relatively new computational paradigm, so advantages, disadvantages, applications and relationships to traditional computing are not completely understood. Expectations for this area are high. Neural networks are particularly well suited for certain applications, especially trainable pattern association. The notion that artificial neural networks can solve all problems in automated reasoning, or even all mapping problems, is probably unrealistic.

Advantages

- Inherently massively parallel.
- May be fault tolerant because of parallelism.
- May be designed to be adaptive.
- Little need for extensive characterization of problem (other than through the training set).

Disadvantages

- No clear rules or design guidelines for arbitrary application.
- No general way to asset the internal operation of the network.
- Training may be difficult or impossible.
- Difficult to predict future network performance (generalization).

(Schalkoff, 1997, p. 10).

2.1.4. Where are Neural Nets Being Used?

The study of neural network is an extremely interdisciplinary field, both in the way of its development and in the way of its application. A brief sampling of some of the areas in which neural networks are currently being applied suggests the breadth of their applicability. The examples range from commercial successes to areas of active research that show promise for the future (Fausett, 1994, p. 7).

2.1.4.1. Signal Processing

There are many applications of neural networks in the general area of signal processing. One of these commercial applications is to suppress noise on a telephone line. The neural net used for this purpose is a form of Adaptive Linear Neuron (ADALINE) (Fausett, 1994, p. 7).

2.1.4.2. Pattern Recognition

There are important relations between certain aspects of neural computing and the field of pattern recognition. Pattern recognition (PR) applications take many forms. In some cases, there is an underlying and quantifiable statistical basis for the generation of patterns. Statistical (or decision-theoretical) PR assumes, as its name implies, that there is a statistical basis for the classification of algorithms. A set of characteristic measurements, denoted features, are extracted from the input data and each feature vector is assigned to one of classes. Features are assumed to be generated by a state of nature or class conditioned set of probability density functions. There are many examples of neural implementations of statistical PR algorithms for classification and many similarities in the operation of neural networks with statistical PR algorithms (Schalkoff, 1997, p. 25).

2.1.4.3. Medicine

One of many examples of the application of neural networks to medicine was developed by Anderson in the mid-1980s. The idea behind this application is to train an auto associative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case. After training, the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the “best” diagnosis and treatment (Fausett, 1994, p. 9).

2.1.4.4. Speech Production

One of the most widely known examples of a neural network approach to the problem of speech production is NET talk. In contrast to the need to construct rules and look-up tables for the exceptions, NET talk’s only requirement is a set of examples of the written input, together with the correct pronunciation for it. The written input includes both the letter that is currently being spoken and three letters before and after it (to provide a context). Additional symbols are used to indicate the end of a word or punctuation. The net is trained using the 1000 most common English words. After training, the net can read new words with very few errors; the errors that it does make are slight mispronunciations, and the intelligibility of the speech is quite good (Fausett, 1994, p. 9-10).

2.1.4.5. Speech Recognition

Progress is being made in the difficult area of speaker-independent recognition of speech. A number of useful systems now have a limited vocabulary or grammar or require retraining for different speakers. Several types of neural networks have been used for speech recognition, including multilayer nets or multilayer nets with recurrent connections (Fausett, 1994, p. 10).

2.1.4.6. Genetic Computing

Genetic computing offers a potentially important mechanism for the training of ANNs. This training is much broader in scope than the simple determination of weights; entire architectures may be searched via genetic algorithms (Schalkoff, 1997, p. 27).

2.1.4.7. Artificial Intelligence

Arguably, some (but presently not all) aspects of human intelligence may be emulated by computers. Significant scientific, engineering, and mathematical effort is being expended in capturing the architectural and aspects of intelligent behavior. To this end, the emerging technology of ANNs, or more generally, neural computing could play a major role (Schalkoff, 1997, p. 25-27).

2.1.5. The History of Artificial Neural Networks

The history of ANN started with people’s interest in neurobiology. It is necessary to divide the studies in the history. We can classify the studies before 1970 and after 1970. The year 1970 is a turning point in history of this science. After this time, many problems about this science were solved and new developments started so ANNs increased over again.

2.1.5.1. The Studies Before 1970

The progress of neurobiology has allowed researchers to build mathematical models of neurons to simulate neural behavior. This idea dates back to the early 1940s when one of the first abstract models of a neuron was introduced by McCulloch and Pitts (1943). They have shown that it is possible to formulate all kind of logical expression with artificial neural cells. Donald Hebb (1949) proposed a learning law named “Hebbian learning” that explained how a network of neurons learned. Other researchers pursued this notion through the next two decades, such as Rosenblatt (1958). Rosenblatt suggested the perceptron learning algorithm. This algorithm has led to a significant improvement in the history of artificial neural network. At about the same time, Widrow and Hoff developed an important variation of perceptron learning, known as the Widrow-Hoff rule. At the end of the 1960s, artificial neural networks entered the unproductive period because Minsky and Papert (1969) had pointed out theoretical limitations of single-layer neural network models in their landmark book Perceptrons (Fu, 1994, p. 4).

2.1.5.2. The Studies After 1970

Despite the negative atmosphere after the book named perceptrons and cutting off the financial supports, some researchers still continued their research and produced meaningful results. For example, Anderson and Grossberg did important work on psychological models. In 1972s, Kohonen and Anderson developed associative memory models. At the end of the 1970s, Fukushima presented the Neocognitron model to image and pattern recognition.

In the early 1980s, the neural network approach was resurrected. Hopfield (1982) introduced the idea of energy minimization in physics into neural networks. His influential paper endowed this technology with renewed momentum. Feldman and Ballard (1982) made the term "connectionist" popular. Sometimes, connectionism is also referred to as sub symbolic processes, which have become the study of cognitive and AI systems inspired by neural networks. Unlike symbolic AI, connectionism emphasizes the capability of learning and discovering representations. Insidiously, connectionism has become a common ground between traditional AI and neural network research.

In the middle 1980s, the book Parallel Distributed Processing by Rumelhart and McClelland (1986) generated great impacts on computer, cognitive, and biological sciences. Notably, the back propagation learning algorithm developed by Rumelhart, Hinton, and Williams (1986) offers a powerful solution to training a multilayer neural network and shattered the curse imposed on perceptrons. A spectacular success of this approach is demonstrated by the NET talk system developed by Sejnowski and Rosenberg (1987), a system that converts English text into highly intelligible speech. It is interesting to note, however, that the idea of back propagation had been developed by Werbos (1974) and Parker (1985) independently (Fu, 1994, p. 4-5).

Since 1987, artificial neural networks have discussed in various symposiums and conferences every year and new models and learning methods have presented.

2.2. Structure and Basic Elements of Artificial Neural Networks

2.2.1. Biological Nerve cells

In studying artificial neurons, it is helpful to first consider the biological origins of neural computing.

It is said that there are 10^{10} nerve cell and more than $6*10^{13}$ links of these cells in a human brain. With these properties, billions of nerve cells come together and generate nervous system. Artificial neural networks are developed using of these features of biological nerve cells (Oztemel, 2012, p. 46-48).

2.2.2. Artificial Nerve Cell (Element of the Process)

In artificial nerve cells firstly, the synapses of the biological neurons are modeled as weights. The synapse of the biological neuron interconnects the neural network and the main task of the synapse is to give the strength of the connection. In artificial nerve cells weights are numerical expressions. A negative weight represents an inhibitory connection; on the other hand a positive weight represents excitatory connections. Second, all inputs are summed together and modified by the weights. Lastly, an activation function controls the amplitude of the output. The range of output is usually between 0 and 1, or it could be -1 and 1.

This process is described in the figure:

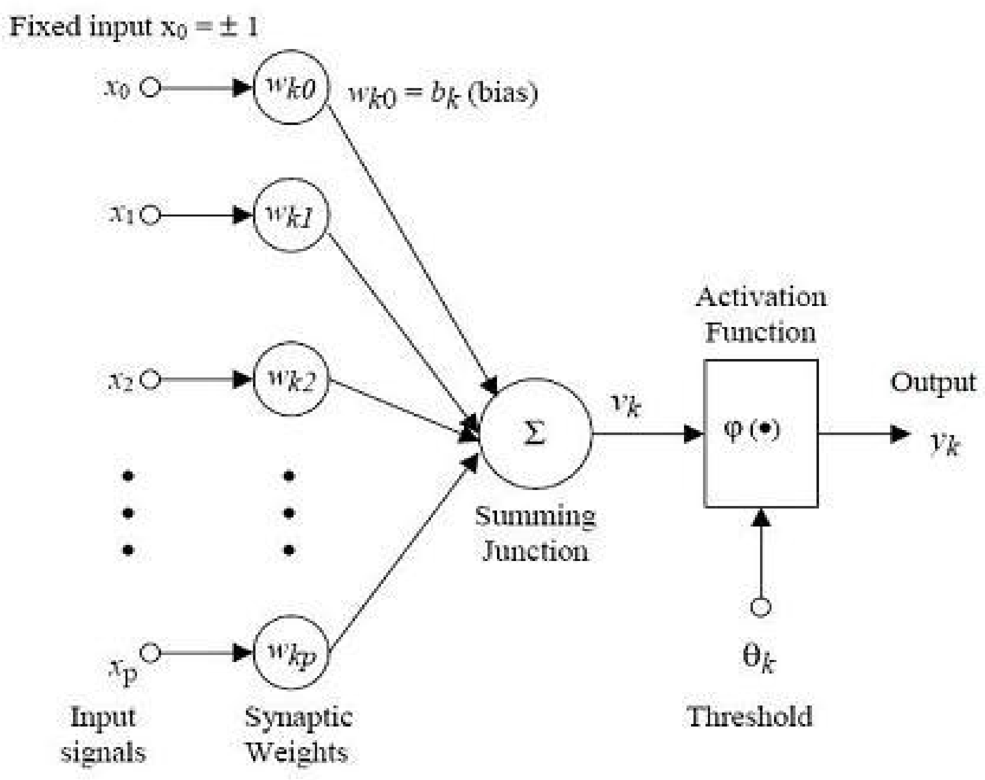


Figure 2. The Structure of an Artificial Neural Network

Source: http://www.learnartificialneuralnetworks.com/. (Access Date: 13.12.2012).

From this model the interval activity of the neuron can be shown as:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j$$

The output of the neuron, y_k, would therefore be the outcome of some activation function on the value of v_k.

Activation Function

The output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). Generally, there are three types of activation functions and they are denoted by Φ(.) .

Firstly, there is the Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(v)=\begin{cases}1 & \text{if } v\ge 0\\ 0 & \text{if } v<0\end{cases}$$

Secondly, there is the Piecewise-Linear function. This function again can take on the values of 0 or 1, but can also take on values between that depending on the amplification factor in a certain region of linear operation.

$$\varphi(v)=\begin{cases}1 & v\ge \frac{1}{2}\\ v & -\frac{1}{2}>v>\frac{1}{2}\\ 0 & v\le -\frac{1}{2}\end{cases}$$

Thirdly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function.

$$\varphi(v)=\tanh\left(\frac{v}{2}\right)=\frac{1-\exp(-v)}{1+\exp(-v)}$$

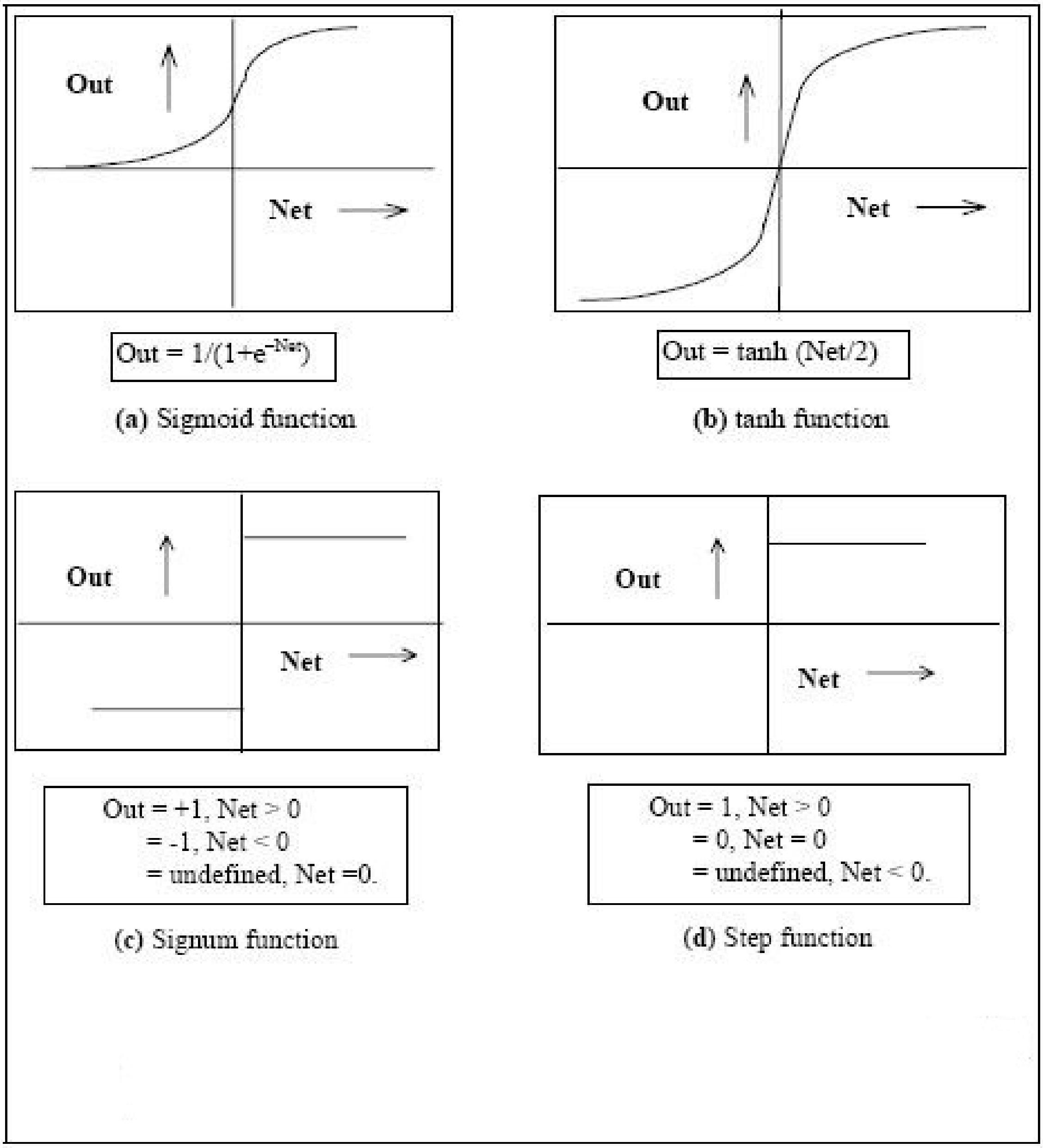


Figure 3. Common Non-Linear Functions Used for Synaptic Inhibitions. Soft Non-Linearity: (a) Sigmoid and (b) Tanh; Hard Non-linearity: (c) Signum and (d) Step.
Source: http://www.learnartificialneuralnetworks.com/. (Access Date: 13.12.2012).

2.2.3. The Working Principle of Artificial Neural Cell

To understand working principle of an artificial neural network easier, it is necessary to give an example. Information and weights that come to the element of process are assumed given in the figure. As shown in, element of process have 4 input values and 4 output values.

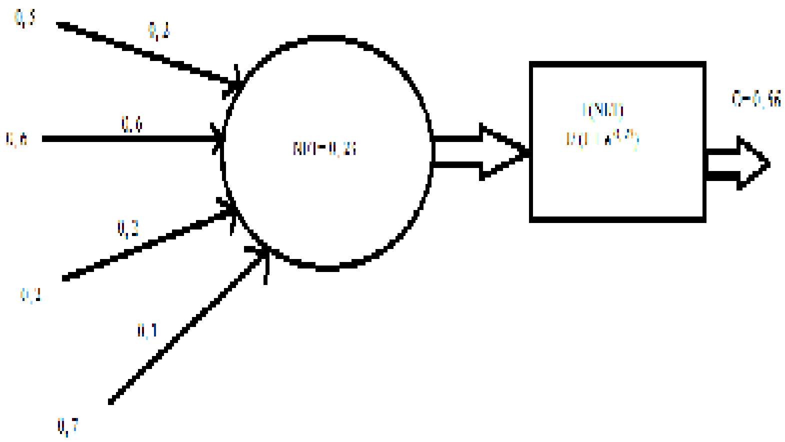


Figure 4. An Example about Working of Artificial Nerve Cell

NET information that comes to the cell is calculated by taking weighted sum as follows:

NET= 0, 5 * (-0, 2) + 0, 6 * 0, 6 + 0, 2 * 0, 2 + 0, 7 * (-0, 1)

NET= -0, 1 + 0, 36 + 0, 04 – 0, 07

NET= 0, 23

If output (O) of the cell is calculated according to its sigmoid function;

$O= 1 / (1 + e^{-0, 23}) = 0, 56$

After doing these procedures for all elements of process, it is understood that how a net produce outputs correspond to inputs (Oztemel, 2012, p. 52).

2.2.4. The Structure of Artificial Neural Network

The topology of a neural network refers to its framework as well as its interconnection scheme. The framework is often specified by the number of layers and the number of nodes per layer. The types of layers include:

- **The input layer:** The nodes in it are called input units, which encode the instance presented to the network for processing. For example, each input unit may be designated by an attribute value possessed by the instance.
- **The hidden layer:** The nodes in it are called hidden units, which are not directly observable and hence hidden. They provide nonlinearities for the network.

- **The output layer:** The nodes in it are called output units, which encode possible concepts (or values) to be assigned to the instance under consideration. For example, each output unit represents a class of objects.(Fu, 1994, p. 18-19)

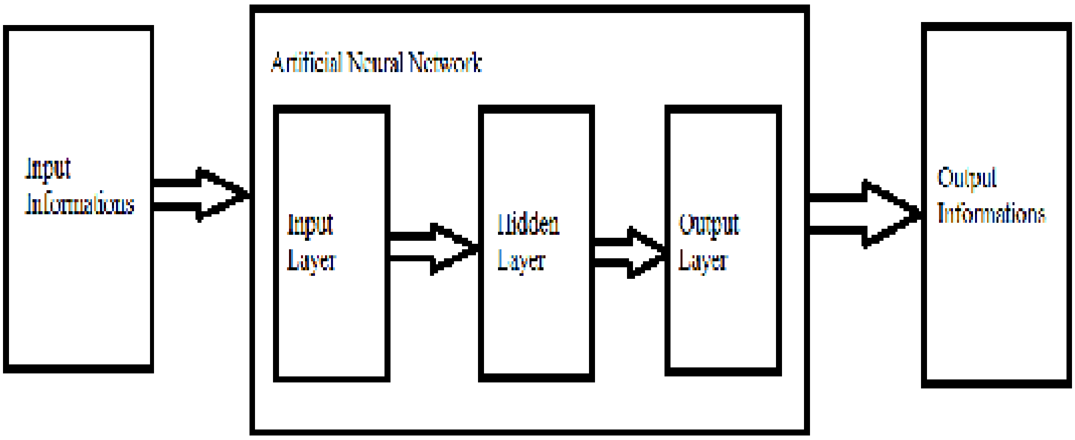


Figure 5. Relationships between Layers of Artificial Neural Networks

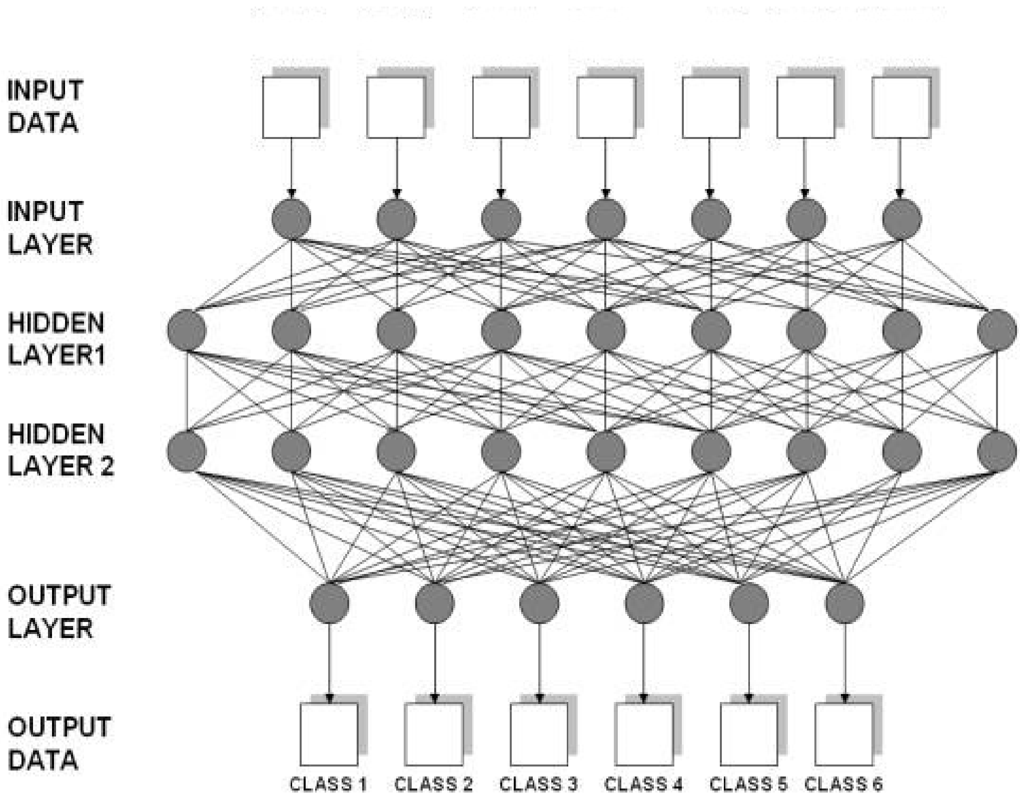


Figure 6. Example of an Artificial Neural Network with One Input Layer, Two Hidden Layers and One Output Layer

Source: http://wgbis.ces.iisc.ernet.in/energy/paper/TR-111/chapter3.htm. (Access Date: 13.12.2012).

2.2.5. Operation of Artificial Neural Network

As shown in the figure, general principle of artificial neural networks can be explained as to generate an output set corresponds to an input set. For this, it is necessary to train the network. Firstly, the examples that shown to the network are turned into vector. This vector is shown to network and the network produces the output vector which is necessary for this vector. The input vectors are can be numerical values that show days of week, gray tones of a picture, weekly or daily stock market values of a security in stock market. Similarly output vectors show the class of input vectors. Imagination of input and output vectors is determined by the person who improved the network (Oztemel, 2012, p. 54).

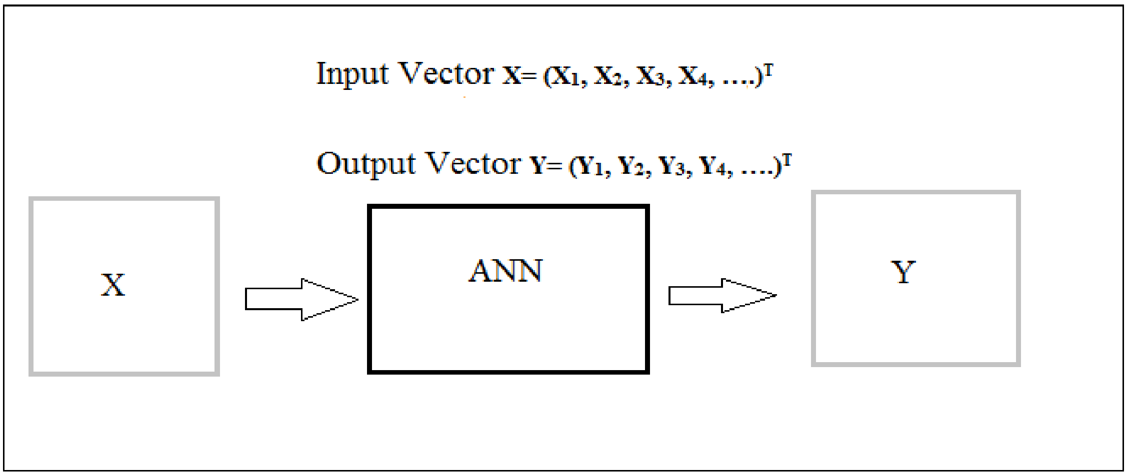


Figure 7. Relationship between Output and Input

2.2.6. Information and Intelligence in Artificial Neural Networks

According to Oztemel (2012, p. 56) information of Artificial Neural Networks is hidden in weight values of links. In other words, intelligence of Artificial Neural Networks is hidden in weight values of links. If weighted values of the network are extremely accurate, the performance of the net is high. Weight values spread over the whole network so a single weight value does not make a sense. For this reason, even if some weight values lost for some reasons, the network maintains its working. And this is the most important property of the Artificial Neural Networks.

2.2.7. The most commonly Used Models of Artificial Neural Networks

In artificial neural networks, topology, sum function and activation function, learning strategy and used learning rules determine the model of the network. The most commonly used models in practical life are:

- Sensors
- Multi-layer sensors
- Vector quantization models (LVQ)
- Self-organizing models (SOM)
- Adaptive resonance theory models (ART)
- Hopfield networks
- Counter propagation networks
- Neocognitron networks
- Boltzman machine
- Probabilistic networks (PPN)
- Elman networks
- Radial-based networks (Oztemel, 2012, p. 56)

2.3. Classification of Artificial Neural Networks

We can examine artificial neural networks in three different categories according to arrays of neurons, time and variations of computation that are done so as to arrange the weights of neurons.

2.3.1. Classification According to the Structure of Artificial Neural Network

Artificial Neural Networks are divided into feed forward and back propagation according to connection forms of neurons.

2.3.1.1. Feed Forward Networks

In feed forward networks, layer connections to only a layer which come after it. Firstly, information that come to the artificial neural network pass from input layer, and than respectively pass from hidden layers and output layer. Multi Layer Perceptron (MLP) and Learning Vector Quantization (LVQ) networks are examples of Feed Forward Networks.

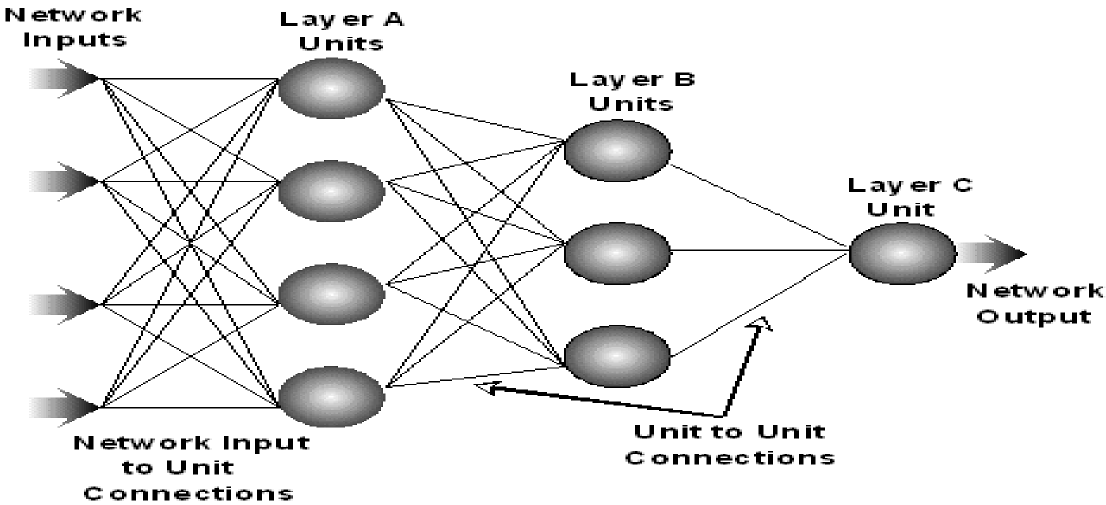
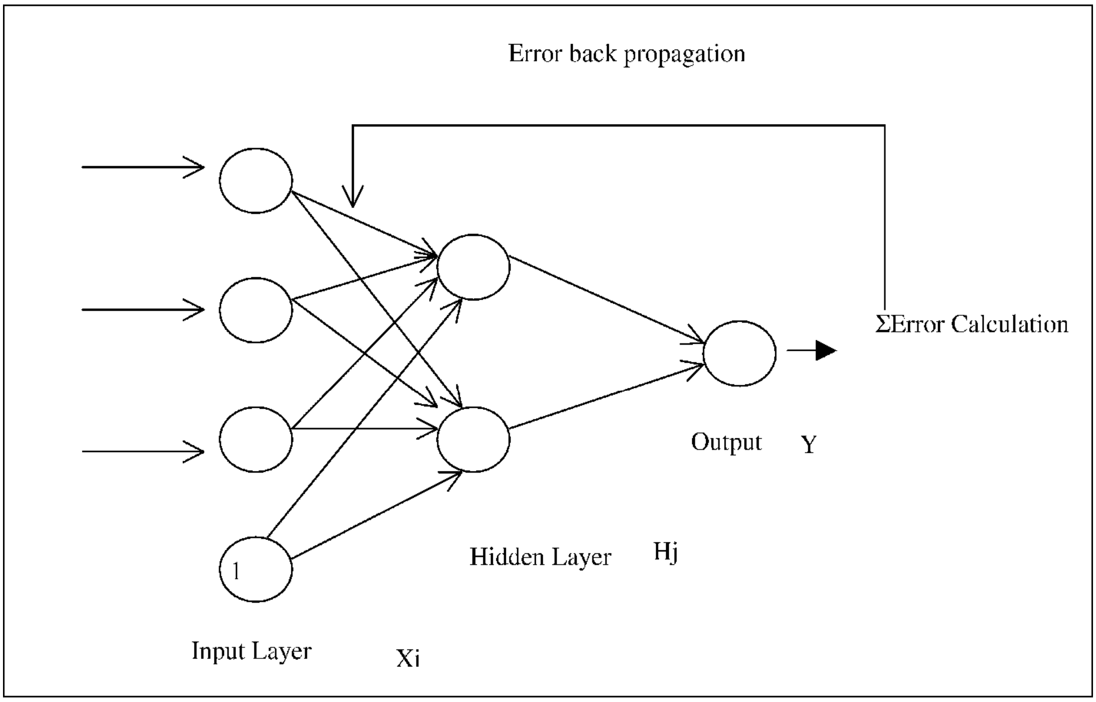


Figure 8. A Three-Layer Feed-Forward Neural Network.
Source: Plummer, 2000.

2.3.1.2. Back Propagation Networks

On the contrary to feed forward neural network, output of a neuron is not given only layer of a neuron that comes after it as an input. It can be connected to another neuron which is in previous layer or in its own layer.



Notes: The weight connecting node *i* in the input layer to node *j* in the hidden layer is denoted by *Wji*, and the weight connecting node *j* to the output node is represented by *Vj*

Figure 9. Feed Forward, Back Propagation Neural Network.

Source: Lisboa, Nagar & Nageim, 2007, p.240-253

2.3.2. Classification According to the Training Algorithms of Artificial Neural Network

We can examine the artificial neural networks in three different categories as supervised, unsupervised and reinforcement training.

2.3.2.1. Supervised Training

During supervised training output values are given for input values that given to network. The network updates its own values in order to present the wanted outputs. As computing the errors between outputs of the network and expected outputs, new weights of the network are designed according to these errors. Delta Rule, Generalized Delta Rule and Back Propagation algorithm are examples of Supervised Training.

2.3.2.2. Unsupervised Training

In this kind of training, only inputs are given to the network during the learning. No target output information is given. The network presents its own rules to classify each example according to information that given at the beginning. The network completes its learning process by designing the weight values. Adaptive Resonance Theory (ART) and Self Organizing Map (SOM) training algorithms are the examples of Unsupervised Training.

2.3.2.3. Reinforcement Training

Reinforcement training algorithm doesn't need the output information. Instead of this, a score or degree which present whether the output is true or not are reported. The network arranges itself according to this information again. Boltzmann rule is an example of Reinforcement Training.

2.3.3. Classification According to the Training Rules of Artificial Neural Network

Artificial neural networks are separated as on-line and off-line learning according to training rules.

2.3.3.1. On-line Training

These systems that are trained according to this rule, working in the real time, they both fulfill their functions and maintain their training. Adaptive Resonance Theory (ART) and Kohonen Training Rule are the examples of on-line training (Oztemel, 2012, p. 26).

2.3.3.2. Off-line Training

These systems based on this rule are trained on the examples before using. During using these systems are not trained. Delta Training Rule is the example of this kind of training (Oztemel, 2012, p. 26).

CHAPTER III

MARKOV CHAINS

3.1. Stochastic Process

Stochastic process can be defined a collection of random variables $\{X_t\}$ where index $t \epsilon$ T. Often T is the set of nonnegative integers, representing time increments. The variable Xt is meant to represent a measurable characteristic, or point of interest (Or, 1986, p.2.1)

3.2. Markov Process

The occurrence of a future state in a Markov process depends on the immediately preceding state and only on it (Taha, 2003, p. 694).

If $t_0 < t_1 < \ldots < t_n$ (n=0,1,2,….) indicates points in time, the family of random variables $\{\xi_{tn}\}$ is a Markov process if it has the following Markovian property:

$$P\{\xi_{tn}= X_n \mid \xi_{tn-1}= X_{n-1}, \ldots\ldots, \xi_{t0}= X_0 \}= P\{\xi_{tn}= X_n \mid \xi_{tn-1}= X_{n-1}\} \quad (3.1)$$

For $\xi_{t0}, \xi_{t1}, \xi_{t2}, \ldots\ldots, \xi_{tn}$.

The probability $P_{xn-1}, X_n = P\{\xi_{tn}= X_n \mid \xi_{tn-1}= X_{n-1}\}$ is called the transition probability. It represents the conditional probability of the system being in X_n at t_n, given it was in X_{n-1} at t_{n-1} (with X representing the states and t the time). This probability is also referred to as the one-step transition because it describes the system between t_{n-1} and t_n. An m-step transition probability is thus defined by

$$P_{xn}, X_{n+m} = P\{\xi_{tn+m}= X_{n+m} \mid \xi_{tn}= X_n\} \quad (3.2)$$

(Idolor, 2010, p.63).

3.3. Markov Chains

A Markov Chain is a special kind of stochastic process where the outcome of an experiment depends only on the outcome of a previous experiment. In other words, the next state of the system depends only on the current state and not on the previous state.

Stochastic processes are of interest for describing the behavior of a system evolving over a period of time (Nambi, Subha, Vasanthi, 2011, p. 73).

A stochastic process $\{X_t\}$ is said to have the Markovian property if

$$P\{X_{t+1}= j \mid X_0= k_0, X_1= k_1,\ldots, X_{t-1}= k_{t-1}, X_t= i\}= P\{X_{t+1}= j \mid X_t= i\} \quad (3.3)$$

for t=0,1,2,.. and every sequence i, j, k_0 ,k_1 ,…., k_{t-1} (Hillier, Lieberman, 2005, p. 734). This is saying that the conditional probability of any future event, given any past events and the present state $X_t = i$, is independent of the past events and depends solely upon the present state.

3.4. Transition Probabilities

The conditional probabilities $P\{X_{t+1}= j \mid X_t= i\}$ are called (one step) transition probabilities.

If the transition probabilities are such that for each i and j,

$$P\{X_{t+1}= j \mid X_t= i\}= P\{X_1= j \mid X_0= i\} \quad (3.4)$$

for all t=1,2,…. Then they are set to be stationary and are denoted by p_{ij}. Stationary transition probabilities indicate that transition probabilities do not change over time.

The existence of stationary (one step) transition probabilities also shows that, for each i, j, and n=0, 1, 2, ….

$$P\{X_{t+n}= j \mid X_t= i\}= P\{X_n= j \mid X_0= i\} \quad (3.5)$$

for all t=0,1,2,….. These conditional probabilities are called n-step transition probabilities.

Stationary transition probabilities can be simply noted as:

$$p_{ij} = P\{X_{t+1}= j \mid X_t= i\},$$

$$p_{ij}^{(n)}= P\{X_{t+n}= j \mid X_t= i\}. \quad (3.6)$$

(Hillier, Lieberman, 2005, p.734).

3.4.1. Chapman- Kolmogorov Equations

Chapman- Kolmogorov equations provide a method for computing these n- step transition probabilities:

$$p_{ij}^{(n)} = \sum_{k=0}^{M} p_{ik}^{(m)} p_{kj}^{(n-m)} \quad (3.7)$$

for all i= 0, 1, ………, M

$j = 0, 1, \ldots\ldots\ldots, M$

and any $m = 1, 2, \ldots\ldots\ldots, n-1$

$n = m+1, m+2, \ldots\ldots$

These equations are used to point out that when we go from one steady state to another in n- steps, the process will be in some other state after exactly m (less than n). Hence, $p_{ik}^{(m)} \, p_{kj}^{(n-m)}$ is just the conditional probability that, given a starting point of state i, the process goes to state k after m steps and then to state j in n-m steps.

Therefore, by summing up these conditional probabilities over all possible k must yield $p_{ij}^{(n)}$. For m= 1, the expression is:

$$p_{ij}^{(n)} = \sum_{k=0}^{M} p_{ik} . p_{kj}^{(n-1)} \qquad (3.8)$$

For m= n-1 the expression is as follows:

$$p_{ij}^{(n)} = \sum_{k=0}^{M} p_{ik}^{(n-1)} . p_{kj} \qquad (3.9)$$

for all states i and j.

As can be seen from the expressions, n- step transition probabilities to be obtained from the one- step transition probabilities recursively (Hillier, Lieberman, 2005, p. 739- 740)

The elements of a higher transition matrix $\|p_{ij}^{(n)}\|$ can be obtained directly by matrix multiplication. Thus,

$$\|p_{ij}^{(2)}\| = \|p_{ij}\| \, \|p_{ij}\| = P^2 \qquad (3.10)$$

$$\|p_{ij}^{(3)}\| = \|p_{ij}^{(2)}\| \, \|p_{ij}\| = P^3 \qquad (3.11)$$

In general

$$\|p_{ij}^{(n)}\| = P^{n-1} \, P = P^n \qquad (3.12)$$

Therefore, if the probabilities are defined in vector form as

$$a^{(n)} = (a_1^{(n)} \, a_2^{(n)} \, a_3^{(n)} , \ldots. \,) \qquad (3.13)$$

Then

$$a^{(n)} = a^{(0)} \, P^n \qquad (3.14)$$

(Taha, 2003, p. 695).

3.4.2. Transition Matrix

The conditional probabilities for a stochastic process can be organized into an n-step transition matrix (Mitra, Riggieri, 2011, p. 8). Such a matrix is of the form

$$P^{(n)} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdot & \cdot & \cdot & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdot & \cdot & \cdot & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdot & \cdot & \cdot & p_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{n1} & p_{n2} & p_{n3} & \cdot & \cdot & \cdot & p_{nn} \end{bmatrix}$$

A transition matrix shows the transition probabilities in a particular column and row as the transition from the row state to the column state (Mitra, Riggieri, 2011, p.8). The matrix $P^{(n)}$ is called a homogeneous transition matrix because all the transition probabilities p_{ij} are fixed and independent of time (Taha, 2003, p.694). Since transition matrices are comprised of conditional probabilities, each entry of a transition matrix is nonnegative and less than 1. This is expressed as follows:

$$\sum_j p_{ij} = 1, \text{ for all i} \qquad (3.15)$$

$$P_{ij} \geq 0, \text{ for all i and j} \qquad (3.16)$$

Each row of the overall stochastic process and each entry within each row is a conditional probability for the process to be in that state.

3.4.3. State Transition Diagrams

A convenient and useful method to visualize the state of Markov Chains when they have stationary transition probabilities and a finite number of states is through the use of a state transition diagram. In such diagrams, each state of a Markov chain is drawn as a numbered of node and the conditional probability of moving from one state to another is drawn by connecting the nodes with an edge and labeling the edge with the numbered probability (Mitra, Riggieri, 2011, p. 8- 9).

Let’s consider the following Markov Chain:

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/3 & 1/3 & 1/3 \\ 2/6 & 3/6 & 1/6 \end{pmatrix} \end{array}$$

This chain is illustrated graphically in Figure 10 as follows:

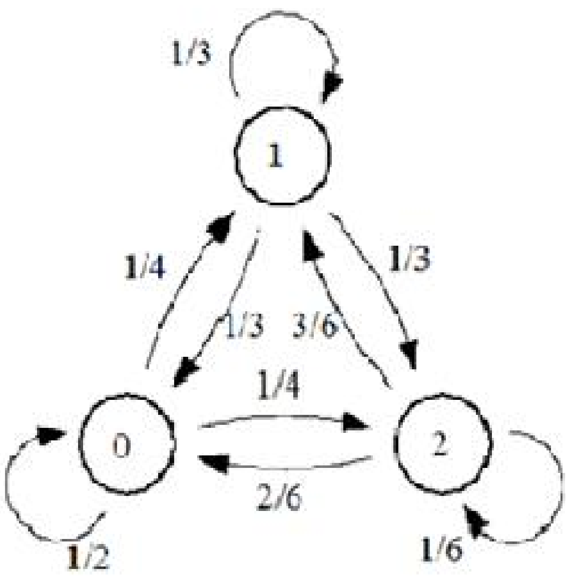


Figure 10. Example of the States of a Markov Chain.

3.5. Classification of States in Markov Chains

Markov Chains are long run stochastic processes. For this reason we need a systematic procedure that will predict the long- run behavior of the system. In such a case it is very necessary to classify, or categorize, the varying types of states.

Given two states i and j, a state j is reachable from state i if there is a path leading from i to j ($P_{ij}^{(n)} > 0$ for some $n \geq 0$) (Winston, 2004, p. 931).

If a state j is reachable from a state i and i is reachable from state j then states i and j are said to communicate with one another (Mitra, Riggieri, 2011, p. 9).

3.5.1. Irreducible States

In a Markov Chain, every state communicates with itself (since $P_{ii}^{(0)} = P\{X_0 = i \mid X_0 = i\} = 1$), and if a state i communicates with another state j, then state j communicates with state i. If state i communicates with state j and state j communicates with state l, then state i communicates with state l.

As a result of these three properties, Markov Chains can be placed into classes, which have states that only communicate with one another. If every state of a Markov Chain communicates with every other state in the chain, if there is only one class all the states communicate, the Markov Chain is said to be irreducible (Hillier, Lieberman, 2005, p. 742-743).

3.5.2. Recurrent States and Transient States

If a state j is reachable from i, but i is not reachable from j (provided that $j \neq i$), then i is considered a transient state. Consequently, if there is a transient state and the process visits this state, there exists a positive probability that the process will move to another state and will never return to the original state. Thus, a transient state will be visited only a finite number of times (Hillier, Lieberman, 2005, p. 743).

A state is said to be a recurrent state if, upon entering this state, the process definitely will return to this state again. Consequently, a state is recurrent if and only if it is not transient. A recurrent state will be visited infinitely often if the process continues forever (Hillier, Lieberman, 2005, p. 743).

3.5.3. Closed Set and Absorbing States

In a Markov process, a set K of states is said to be closed if the system, once in one of the states of K, will remain there indefinitely.

A special example of a closed set is a single state j with transition probability $p_{jj}= 1$. In this case j is called an absorbing state (Taha, 2003, p. 696).

In other words, as explained in 3.5.2 if a stochastic process, such as a Markov chain, enters a state, and will definitely returns to it, the state is said to be recurrent. Thus, recurrent states cannot be transient; however, they can be absorbing (Mitra, Riggieri, 2011, p. 11).

3.5.4. Periodic and Ergodic States

A state j is said to be periodic with period k > 1 if k is the smallest number such that all paths leading from state j back to state j have a length and that is a multiple of k. If a recurrent state is not periodic, it is referred to as aperiodic.

According to Winston (2004), if all states in a Markov chain are recurrent, aperiodic, and communicate with each other, the chain is said to be ergodic.

3.6. Properties of Markov Chains in the Long Run

3.6.1. Steady State Probability

It is very important to understand the steady-state probabilities, which can be used to describe the long- run behavior of a Markov chain. While calculating the n- step transition probabilities, if the value of n is large enough, Markov chain will indicate the feature of steady state. In other words after the n- step transition probabilities for a Markov chain, every row of the matrix have identical entries, thus the probability that the system is in each state does not depend on the initial state of the system. So, after a large number of transitions, the probability that the process will be in each state k, and this probability no longer depends on the initial state. This can be summarized as:

For any irreducible ergodic Markov Chain, $\lim_{n\to\infty} P_{ij}^{(n)}$ exists and is independent of i.

Moreover,

$$\lim_{n\to\infty} P_{ij}^{(n)} = p_j > 0 \qquad (3.17)$$

Where the p_j uniquely satisfy the following steady- state equations:

$$p_j = \sum_{i=0}^{M} p_j \, p_{ij} \text{ , for } j=0, 1, 2, \ldots, M, \qquad (3.18)$$

$$\sum_{j=0}^{M} p_j = 1. \qquad (3.19)$$

(Hillier, Lieberman, 2005, p. 744-745).

The p_j is called the steady- state probabilities of the Markov chain. As shown in the above, after a large number of transitions the probability of finding the process in a certain state, such as j tends to the value of p_j, which is independent of the probability distribution of the initial state (Mitra, Riggieri, 2011, p. 12).

According to Hillier and Lieberman (2005), it is important to remark that the steady- state probability does not imply that the process settles down into one state. Conversely, the system continues to make transitions from state to state, and at any step n the transition probability from state i to state j is still state p_{ij}.

The p_j is also known as stationary probabilities. If the initial probability of being in state j is given by p_j for all j, then the probability of finding the process in state j at time n=1, 2,…. is also given by p_j, or

$$P\{X_n = j\} = p_j \tag{3.20}$$

(Hillier, Lieberman, 2005, p. 745).

3.6.2. Expected Average Cost per Unit Time

In many applications, there is a cost (or other penalty function) associated with a Markov Chain in a state X_t and it is important what happens to that cost in the long run (Hillier, Lieberman, 2005, p. 747)

Suppose that a cost, penalty (or reward) C (X_t) is incurred when the process is in state X_t at time t, for t=0, 1, 2, 3,…. . Then C (X_t) is a random variable taking one of the values C(0), C(1), C(2),……….., C(M). Total cost after n transitions is

$$\sum_{t=1}^{n} C(X_t) \tag{3.21}$$

(Or, 1986, p. 2.33)

According to Hillier and Lieberman (2005), it makes more sense to talk about cost per unit time rather than total cost in many cases. For example, suppose one would like to compare two inventory policies. One of them covers a five year horizon and the other one covers an eight year horizon. In such a case, it is meaningless to talk about total cost. Instead, it is a good way to compare average, yearly cost. For that purpose we can define average cost as follows:

$$AC_n = \frac{1}{n}\sum_{t=1}^{n} C(X_t) \tag{3.22}$$

The expected average cost over n periods given Markov chain is in state i initially:

$$E_i(AC_n) = E_i \left[\frac{1}{n} \sum_{t=1}^{n} C(X_t) \right] = \frac{1}{n} \sum_{t=1}^{n} E_i(C(X_t)) \quad (3.23)$$

$$\frac{1}{n} \sum_{t=1}^{n} (\sum_{j=0}^{M} C(j) P\left\{X_t = j / X_0 = i\right\} \quad (3.24)$$

$$\frac{1}{n} \sum_{t=1}^{n} \sum_{j=0}^{M} C(j) P^t(i, j) \quad (3.25)$$

$$\sum_{j=0}^{M} C(j) \left[\frac{1}{n} \sum_{t=1}^{n} P^t(i, j) \right] \quad (3.26)$$

It can be shown that for any irreducible Markov chain with positive recurrent states

$$\text{Lim } \frac{1}{n} \sum_{t=1}^{n} P^t(i, j) = p_j \quad (3.27)$$

Then

$$E_i\,(AC_n) = \sum_{j=0}^{M} C(j) p\,(j) \quad (3.28)$$

(Or, 1986, p. 2.34).

3.7. First Passage Time

n step transition probabilities (from state i to state j) also make probability statements about the number of transitions made by the process in going from state i to state j for the first time. This amount of time is named the first passage time in going from state i to state j when i is not equal to j. When i = j, it is not the first passage time; it is only the number of transitions that goes from one state and back to itself. In this situation, the first passage time is named as the recurrence time for state i (Hillier, Lieberman, 2005, p. 750).

Generally, the first passage time is random variable. These probabilities associated with them depend upon the transition probabilities of the process. We can denote $f_{ij}^{(n)}$ as the probability that the first passage time from state i to state j is equal to n. For n > 1, this first passage time is n if the first transition is from state i to some state k, k is not equal to j, and then the first passage time from state k to state j is n – 1. So these probabilities satisfy the following recursive relationships:

$$f_{ij}^{(1)} = p_{ij}^{(1)} = p_{ij}, \quad (3.29)$$

$$f_{ij}^{(2)} = \sum_{k \neq j} p_{ik} f_{kj}^{(1)} , \tag{3.30}$$

. .

. .

$$f_{ij}^{(n)} = \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)} . \tag{3.31}$$

There is a problem with these relationships. The summations of all these relationships can be strictly less than 1 and this shows that a process initially in state i may never actually reach state j. This means that these relationships are not the best measure of first passage times. Because it is very long to calculate f_{ij} for all n, it is relatively simple to obtain the expected first passage time from state i to state j. If we denote the expectation of the first passage time by m_{ij}, it is indicated as follows:

$$m_{ij} = \begin{cases} \infty \text{ if } \sum_{n=1}^{\infty} f_{ij}^{(n)} < 1 \\ \sum_{n=1}^{\infty} n f_{ij}^{(n)} \text{ if } \sum_{n=1}^{\infty} f_{ij}^{(n)} = 1 \end{cases} \tag{3.32}$$

Whenever,

$$\sum_{n=1}^{\infty} f_{ij}^{(n)} = 1, \tag{3.33}$$

$$m_{ij} = 1 + \sum_{k \neq j} p_{ik} m_{kj} . \tag{3.34}$$

This equation implies that the first transition from state i can be to either state j or to some other state k. If it is to state j, then the first passage time is 1. Instead of it, if the first transition is to some state k (k is not equal to j), in this case the probability is shown by p_{ik}, conditional expected first passage time from state i to state j is 1+ m_{kj} . When we combine these new facts, and we summarize over all possibilities for the initial transition, leads us directly to this equation (Hillier, Lieberman, 2005, p. 751).

3.8. First Recurrence Time

When the process is in state i at first, it may return to state i for the first time at the nth step, for each $n \geq 1$. The number of steps before the process returns to i is called the first recurrence time (Taha, 2003, p. 697).

For the case of m_{ij} where i = j, m_{ii} is the expected number of transitions until the process returns to the first state i, and so is called the expected recurrence time for state i. After obtaining the steady- state probabilities (p_0, p_1, p_2, …… ,p_M), this expected recurrence times can be calculated as

$$m_{ii} = \frac{1}{p_i}, \qquad \text{for i= 0, 1, ….., M} \qquad (3.35)$$

(Hillier, Lieberman, 2005, p. 752).

CHAPTER IV

THE SUMMARY OF EMPRICAL STUDIES

4.1. Some Studies about Markov Chains on Stock Markets

Bhargava and Flietz (1973), in their study empirical results for the vector process model suggest that price movements appear to be described by a first- or higher-order non-stationary Markov chain. Tests also indicate that the vector-process Markov chain is heterogeneous. Empirical results for the individual-process Markov chain model suggest that an individual stock has a short-term memory with respect to daily price relatives.

Chen and So (2006) proposed a threshold heteroscedastic model which integrates threshold nonlinearity and GARCH-type conditional variance for modeling mean and volatility asymmetries in financial markets. They showed higher average volatility and more persistent volatility when bad news arrives.

Greyserman, Jones and Strawderman (2006) contributed to portfolio selection methodology using a Bayesian forecast of the distribution of returns by stochastic approximation. The daily stock market indices for 11 different countries over the period 1975-2002 were used in this comparison of DU and MV using different data models. The countries included US, UK, Canada, Belgium, Australia, France, Japan, Austria, Spain, Germany and Hong Kong. The indices were compiled and provided by Morgan Stanley Capital International the data were partitioned into five periods of five consecutive years. This allowed a comparison of the means, standard deviation, within – country serial correlations and between – country correlations. They carried out a numerical optimization procedure to maximize expected utility using the Markov chain Monte Carlo MCMC samples from the posterior predictive distribution. They showed that this model resulted in an extra 1.5 percentage points per year in additional portfolio performance which was quite a significant empirical result. This approach applied to a large class of utility functions and models for market returns.

Griffin and Steel (2006) used the algorithm was based on Markov chain Monte Carlo (MCMC) methods and they used a series representation of Lévy processes. Their application to stock price data showed the models performed very well, even in the face of data with rapid changes, especially if a superposition of processes with different risk premiums and a leverage effect was used.

Guidolin and Timmermann (2007) characterized equilibrium asset prices under adaptive, rational and Bayesian learning schemes in a model where dividends evolve on a binomial lattice. They investigated restrictions on prior beliefs under which Bayesian and rational learning lead to identical prices and show how the results can be generalized to more complex settings where dividends follow either multi-state i.i.d. distributions or multi-state Markov chains.

Hamilton and Susmel (1994) examined the U.S. weekly stock returns, allowing the parameters of an ARCH process to come from one of several different regimes, with transitions between regimes governed by an unobserved Markov chain. They estimated models with two to four regimes in which the latent innovations come from Gaussian and Student *t* distributions.

Idolor (2010) presented a method of Markovian analysis of security prices. He used eight securities randomly selected from banking sector of the Nigerian Stock Exchange for a period spanning January 4th 2005, to June 30th 2008; and defined a set of three states (rise, drop, stable) for the process in terms of a Markovian framework. The findings showed that the Markov Chains model did not give a precise prediction of the direction in which prices were headed in the short run in addition, it was shown that under a fairly general partial adjustment Markov Chain model of stock price determination, price movement which did not display the random walk characteristics may be interpreted as purely following a Markov process, while a set of hypotheses and standard methods were suggested for drawing statistical inferences in time on the randomness of stock prices when Markov Chains are applied.

Idolor (2011) used a method of Markovian analysis of the long run prospects of security prices in Nigeria. This study examined five securities randomly selected from the banking January 4th 2005, to June 30th 2008 and defined a set of three states (rise, drop and stable) for the process in terms of a Markov framework and finally it was found that price levels are likely to remain relatively stable in the long run irrespective of the current down turn of prices and global economic meltdown.

Mills and Jordanov (2003) studied on the predictability of size portfolio return using a new database constructed from the London Stock Exchange for the period 1985-1995. The size portfolios were derived from DATASTREAM Database on-line. The underlying data set contained the constituents of the FTSE actuaries all share index except Financials and Investment Trusts, i.e., General Industrials, Consumer goods, Services, Mineral Extraction and Utilities. The sample was restricted to those shares available at the end of 1995 and the number of firms was further restricted to those with positive market-to-book values and barrowing ratios. The returns showed evidence of nonlinearity and non-normality, so conventional autocorrelation analysis was supplemented by the use of a Markov chain technique. They found evidence that the returns of the two smallest and four largest were predictable in the direction suggested by the bubbles and fads alternatives to the random walk hypothesis.

Kanas (2003) studied on the US stock market and considered annual stock price and dividend series over the period 1872-1999. In the end of the study he suggested that Markov regime switching model is the most preferable non-linear empirical extension of the present-value model for out of sample stock return forecasting.

Kılıç (2005) used Markov chain methodology to test whether or not the daily returns of the BIST 100 index follows a martingale (random walk) process. He examined daily values of BIST 100 index which were obtained from electronic data delivery system of Central Bank of Turkey (http://tcmbf40.tcmb.gov.tr/cbt-uk.html) The index value covered 4234 workdays of 17 years for the period 23.10.1987-02.11.2004. Result of this study hold the weak-from EMH that at any given time, stock prices fully reflect all the available historical information. Under a random walk, historical data on prices and volume have no value in predicting future stock return.

King and Zhang (2008) presented a MCMC algorithm to estimate parameters and latent stochastic processes in the asymmetric stochastic volatility (SV) model, in which the Box-Cox transformation of the squared volatility follows an autoregressive Gaussian distribution and the marginal density of asset returns has heavy-tails. They found that when their model and its competing models were applied to daily returns of another five stock indices, in terms of SV models, the Box-Cox transformation of squared volatility is strongly favored against the log-transformation for the five data sets.

Lin, Tsai and Wang (2009) proposed a hidden Markov switching moving average model (MS-MA model) to extend the moving average model when the dynamic process of stock returns is predictable. They showed that the dynamic process of stock returns exhibits MS-MA property, meaning the moving averages of stock returns are correlated.

Liu (2011) examined a continuous- time intertemporal consumption and portfolio choice problem under ambiguity, where expected returns of a risky asset follow a hidden Markov chain. He found that continues Bayesian revisions under incomplete information generate ambiguity-driven hedging demands that mitigate intertemporal hedging demands.

Los (1998) analyzed the weekly pricing in the six Asian stock markets, as measured by their respective market indices in the past ten years, from July 1986 to June 1996. The weekly data for the Jakarte Stock exchange range actually from January 1988 to June 1996, and the data for the stock exchange of Thailand from July 1987 to June 1996. At the end of the study, the indices of the six Asian stock markets in the region do not pass all the efficiency tests. For these stock markets the random walk hypothesis of market pricing was rejected.

Maskawa (2003) studied a multivariate Markov chain model as a stochastic model of the price changes of portfolios in the framework. Their modeling of the transition matrix extremely reduced the number of independent parameters. That is very import for practical prediction, because the number of the independent parameters for Markov chain model immediately becomes on astronomical number when we consider large portfolios.

McQueen and Thorley (1991) used a Markov chain model to test the random walk hypothesis of stock prices. They showed that annual real returns exhibit significant non random walk behaviors in the postwar period in New York Stock Exchange (NSE).

Mitra and Riggieri (2011) used the stochastic processes called Markov Chains. They sought out to predict the immediate future stock prices for a given company. They found the moving averages for the data and the grouped them into four different states of results. They did Markov Chain calculations to the data to create a 4x4 transitional probability matrix. Using this transition matrix they solved a system of equations and found four steady states that were variables that represented the probability that a stock price for a given day would fall into one of the four states. According to Mitra and Riggieri, when they use this information they can apply their actual data to these equations and predict the next stock prices for the near future. They were able to successfully predict the next few days of stock prices using this method.

Nelson and Startz (1989) used a Markov model of heteroskedasticity, risk and learning in the stock market. The estimates indicate that agents are consistently surprised by high variance periods, so there is a negative correlation between movements in volatility and in excess returns.

Ryan (1973) explained the relevance of the theory of Markov process to the analysis of stock price movements and show that Markov theory is seen to be relevant to the analysis of stock price in two ways: "1. As a useful tool for marking probabilistic statements about future stock price levels. In this role it constitutes. On alternative to the more traditional regression forecasting techniques to which it is, in many ways superior. 2. As an extension of the random walk hypothesis."

Tsionas (2000) illustrated the computation of marginal likelihoods and Bayes factors when Markov Chain Monte Carlo had been used to produce draws from a model's posterior distribution. Models included a normal finite mixture, a GARCH and a Student *t* -model as alternative models for the Standard and Poor's stock returns.

Vasanthi, Subha and Nambi (2011) In this study the First Order Markov Chain Model was applied to indices of various stock exchanges round the globe Indices from markets like the American stock markets (DJIA, S&P 500), European Markets (FTSE, FTSH), Australian markets (AUSTA ^ DRD), China (SSE ^), South East Asian markets (Hong Seng), Pakistan (KSE), India (BSE, NSE) etc. were chosen for the study. All major stock market indices representing popular investment destinations were included in the study. The results of the trend prediction using Markov Chain analysis were compared with the results obtained through traditional trend prediction tools. The prediction of the trend using Markov Chain Model was done using short term (one year data), medium term (3 year data) and long term (5 year data) and the results were compared. The results of the study showed that majority of the time, Markov model out perform the traditional trend prediction methods.

4.2. Some Studies about Artificial Neural Network on Stock Markets

Aygören, Sarıtaş and Moralı (2012) thought that it is very important to estimate stock market price for developing countries such as Turkey. In this study they used some variables which were relevant to BIST 100 index and tried to estimate with traditional time series, numerical search models and artificial neural networks. For the model, BIST 100, golden price, interest rate, dual transactions between banks TL and USD daily closing price data were used as independent variables. The total number of daily values was 3779. In the model, ARMA (p, q) models for traditional time series, Newton Method for numerical search models and backpropagation algorithm for artificial neural networks were developed for comparison. According to model results, artificial neural networks had more performance than traditional time series and numerical search models.

Avcı and Çinko (2008) analyzed the efficiency of artificial neural network in predicting the daily returns of six developing countries. These countries are five new members of European Community and a candidate country. For this purpose, they used criteria of performance as mean squared error (MSE), normalized mean squared error (NMSE), mean absolute error (MAE) and accuracy of trend. The findings show that these criteria of performance are not adequate to determine the profitable models. In many times, developed ANN models provide a return more than market but no ANN model can outclass to market and any other ANN models consistently.

Cao, Leggio and Schniederjans (2005) did a predicting the Chinese stock market. They used artificial neural networks to predict stock price movements (i.e., price returns) for firms traded on the Shanghai stock exchange. The data of this study covers the time period of January 1, 1999 through December 31, 2002. They used data for 367 public corporations traded on the Shanghai Stock Exchange (SHSE). The data consist of daily closing prices getting from SinoFin (www.sinofin.net) and quarterly book value and common shares outstanding getting from a manual review of the data from the annual reports of each firm. They compared the predictive power using linear models from financial forecasting literature and univariate and multivariate neural network models. For that purpose, they used hypotheses testing as Paired sample t- test and Diebold and Mariano test and arrived some conclusions. Their results show that neural networks outperform the linear models compared. These results are statistically significant across sample firms and indicate neural networks are useful tool for stock price prediction in emerging markets, like China.

Diler (2003) tried to predict the movement of BIST 100 index for the next day. For this purpose, he used artificial neural networks method which was very popular in financial field in the world. For this kind of analysis, the most used method which is error backpropagation method and the shape of this method which is enhanced by momentum were preferred in this work. In the model nine technical indicators were used. Daily values of these indicators were gotten from the beginning of 1990, to November 2003. In the model, one day lagged values technical indicators were used. There are seven inputs and one output in this work. At the end of this work, it is seen that error backpropagation method estimated the direction of next day of BIST 100 index as 60, 81 %.

Dutta, Jha, Lha and Mohan (2006) worked on the modeling of the Indian stock market (price index) data using artificial neural network. They researched the efficiency of artificial neural network (ANN) in modeling the Bombay Stock Exchange (BSE) SENSEX weekly closing values. They developed two networks with three hidden layers as ANN1 and ANN2. ANN1 takes as its inputs the weekly closing value, 52 week moving average of the weekly closing SENSEX values, 5- week moving average of the same, and the 10- week Oscillator for the past 200 weeks. ANN2 takes as its inputs the weekly closing value, 52 week moving average of the weekly closing SENSEX values, 5- week moving average of the same and the 5- week volatility for the past 200 weeks. Both the neural networks are trained using data for 250 weeks starting January 1997. They used the networks to predict the weekly closing SENSEX values for the two-year period which begin January 2002 and choose the root mean square error (RMSE) and mean absolute error (MAE) as indicators to perform networks. Consequently, they found that the performance of ANN1 is better than that of ANN2 for predicting the weekly closing values of the BSE SENSEX.

Fernandez- Rodriguez, Gonzales- Martel and Sosvilla- Rivero (1999) investigated the profitability of a simple technical trading rule based on artificial neural networks. They applied this investment strategy to the General Index of Madrid Stock Market. In absence of trading costs, the technical trading rule is always superior to a buy- and- hold strategy for both “bear” market and “stable” market episodes. In their work, on the other hand, they found that the buy- and- hold strategy generates higher returns than the trading rule based on ANN only for a subperiod presenting upwards trend, “bull” markets.

Freisleben (1992) evaluate the performance of backpropagation neural networks to solve the problem of predicting stock market prices. The training data used in his experiment is not only based on stock market prices, but also includes a variety of other economical factors. Several simulation results illustrate the prediction quality obtained from backpropagation neural network.

Kanas (2001) measured the performance of monthly aggregate stock returns for Dow Jones and FT. He chose a sample and compared the forecasting performance of a linear and an artificial neural network (ANN) model of monthly aggregate stock returns. He did this study to examine whether forecasts from the ANN model are preferable to forecast form, the linear model in terms of directional accuracy as well as forecast encompassing. On the basis of directional accuracy, he analyzed whether one model is better at predicting the direction of change of stock returns. For this purpose, he used the Peseron and Timmerman (1992) testing procedure. On the basis of the forecast encompassing criterion, he analyzed whether the forecast errors of one model can be explained by the forecast of the other model. For this purpose he used the Clements and Hendry (1998) testing procedure. In this paper, monthly data for aggregate stock returns, trading volume and dividends for two countries, namely the United Kingdom (UK) and the United States (US) were used. The stock index for the UK was the FT all share index (FT) and the Dow Jones Industrial Average DJ for the US. Trading volume was the number of shares traded on the London Stock Exchange (LSE), and on the New York Stock Exchange (NYSE). The dividend series for each country was the dividend index constructed by DataStream. The period extended from January 1980 to June 1999, with a total of 234 observations for each series. The period from January 1980 to December 1994 was treated as the "trading" in- simple period for ANN and the estimation of the linear model. The subsequent period from January 1995 to June 1999 was the "testing" out-of-sample period. At the end of the study, both models performed badly in terms of predicting the directional change of the two indices. The ANN forecasts could explain the forecast errors for both indices. So, the ANN forecasts are preferable to linear forecasts, indicating that the inclusion of nonlinear terms in the relation between stock returns and fundamentals is important in out- of sample forecasting.

Kim and Han (2000) investigated genetic algorithms (GAs) approach to feature discretization and the determination of connection weights for artificial neural networks (ANNs) to predict the stock price index. In this study GA was employed not only to improve the learning algorithm, but also to reduce the complexity in feature space. The data of this study is technical indicators and the direction of change in the daily Korea stock price index (KOSPI). The total number of samples is 2928 trading days, from January 1989 through December 1998. The direction of daily change in the stock price index was categorized as "0" or "1". "0" means that the today's index is higher than the next day's index and "1" means that the next day's index is higher than today's index. They select 12 technical indicators as feature subsets by the review of prior researches. This study proposed GA approach to feature discretization (GAFD). GAFD is classified as an exogenous, global, hard and non- parameterized discretization method. This study proposed a new hybrid GA and ANN to mitigate the above limitations. They concluded that GAFD reduces the dimensionality of the feature space then enhances the generalizability of the classifier from the empirical results.

Kim and Lee (2004) used artificial neural networks and compared a feature transformation using a genetic algorithm (GA) with two conventional methods of ANNs. They incorporated GA to improve the learning and generalizability of ANNs for stock market prediction. Daily predictions were conducted and prediction accuracy was measured. For ANNs, three feature transformation methods were compared. These three transformation model (LTM), fuzzy transformation model (FTM) and GA- based feature transformation model (GTM). This study tested the statistical significance of the difference. The research data used in this study consist of technical indicators and the direction of change in the daily Korea composite stock price index (KOSPI). The total number of samples came from 2348 trading days from January 1991 to December 1998. The results showed that GA- based feature transformation model more successful than the other two feature transformation models. The GTM discretized the original continuous data according to optimal or near- optimal thresholds. They concluded that the GTM reduce the dimensionality of the feature space and then enhanced the generalizability of the classifier from the empirical results.

Kutlu and Badur (2009) used artificial neural networks to predict Istanbul Stock Exchange index value. The data of the work were gotten from Central Bank of Turkey and other stock markets. The total number of daily values was 1270 trading days, from 2. 7. 2001 to 13. 7. 2006. At the end of the study, the results showed that feedforward artificial neural networks can also be used to model ISE market index value successfully.

Mizuno, Kosaka, Yajima and Komoda (1998) proposed a neural network model for technical analysis of stock market and its learning method for improving the prediction capability. In this work, neural network was applied to technical analysis as a buying and selling timing prediction system for Tokyo Stock Exchange Prices Index (TOPIX) was presented. The dataset of the work were gotten from technical indexes of TOPIX. Typical technical indexes are an average of the prices over certain past period, deviation of price from moving average, psychological line, relative strength index. Each of these indexes was normalized into 0 to 1 to form an input pattern to the neural network model. As output data, they defined three patterns: buying signal, selling signal, and no-change. This paper proposed a learning method that contributed to improving prediction accuracy of other, more important categories. In the method, the numbers of learning samples were controlled by using information about the importance of each category. Experimental simulations applied to practical data demonstrated that the prediction system generates buying and selling signals at more proper timings on the whole and made higher profit compared with that yielded by a single use of each technical index.

O'Connor and Madden (2005) researched the effectiveness of using external indicators to predict movements of Dow Jones Industrial Average index (DJIA). Two of these external indicators are commodity prices and currency exchange rates. The data of this work comprised of daily opening and closing values of the DJIA index and corresponding values for a range of external indicators. When they chose the external indicators, they cared whether an indicator is likely to have a significant influence on the movement of the index. The relevance of the chosen external indicators was determined experimentally by adding them as inputs to the neural network models. They assessed whether these indicators improved the performance. The dataset also includes technical indicators derived from the DJIA spot values, specifically the daily gradient of the DJIA, calculated as (Closing- Opening)/Opening, and 10 days and 30 days moving averages of opening values. The performance of each technique was evaluated using different domain- specific metrics to predict movements of DJIA. They described the evaluation procedure, involving the use of trading simulations to determine practical value of predictive models, and comparison with simple benchmarks that respond to underlying market growth. They presented experiments, basing trading decisions on a neural network trained on a range of external indicators resulted in a return on investment of 23, 5 % per annum, during a period when the DJIA index grew by 13, 03 % per annum.

Olson and Mossman (2003) compared neural network forecasts of one- year-ahead Canadian stock returns. The used forecasts were obtained by using ordinary least squares (OLS) and Logistic Regression (logit) techniques. The data used in this study were obtained from the Canadian Compustat annual reports for the 18 year period from 1976 to 1993. They constructed a dataset of 61 annual accounting ratios for 2352 Canadian companies. The most recent six years of data were rolled forward each year to forecast annual returns for 1983- 1993. Their results indicated that backpropagation neural networks outperform the best regression alternatives for both point estimation and in classifying firms expected to have either high or low returns.

Tektaş and Karataş (2004) discussed artificial neural network as one of the recent methods in business decision making. They researched that whether the ANN is suitable to apply to finance or not. For this reason, they did an application about Turkey. In this study, the stock prices of seven companies in Istanbul Stock Exchange (ISE) were tried to estimate. Initially, ANN method was applied to the daily and weekly prices and using the daily values had more performance than using the weekly values. In the second step, daily prices were predicted using ANN and linear regression. ANN showed higher prediction performance; imply that ANN can be considered as an alternative financial analysis tool.

CHAPTER V

APPLICATION

5.1. The Data

The sample data covers 10357 session closing values of the Borsa Istanbul 100 (BIST 100) index between the period of January 04, 1988-April 04, 2012. The data were obtained from the electronic data delivery system of the Borsa Istanbul (http://borsaistanbul.com/).

The daily session returns (R_t) are computed as a percentage change of the BIST 100 index session closing values (P_t) as;

$$R_t = (P_t - P_{t-1})/P_{t-1} \qquad (5.1)$$

Here, t represents the sessions (t=1…10356).

The average (expected) return (m_R) is calculated as approximately 0,0011 and standard deviation (s_R) as 0,0206 for the period considered. The standard deviation is extremely high in comparison to expected return (approximately 19 times of the expected return). So, the BIST 100 index return exhibits high dispersion, implying an enormous risk for the investors.

5.2. The Methodology

Kılıç (2013, May) integrated Artificial Neural Network Models by Markov Chain Process to estimate the movement direction of Turkish Lira/US Dollar exchange rate returns. We used this method for session returns of BIST 100 index to predict the direction of next session return. Excell IF ve SPSS 16 programmes were used in application.

5.2.1. Modeling the Returns as the Discrete Categorical States of Markov Chain Process

In this thesis, the session returns are supposed to be a stochastic process with sixthteen discrete state spaces and eighteen states $\{S_1, S_2, S_3, ..., S_{17}, S_{18}\}$ with Markov chain that conditional probability of any next future return state (S_j^t) depends only on the present return of the state (S_i^{t-1}) and is independent any other states of the past returns $P\left(S_j^t \middle| S_i^{t-1}\right)$.

As stated previously, objective of modeling the returns as discrete categorical states is to calculate probability of positive/non positive return for the next session given the present session. And then these probabilities are used as inputs to the ANN models for training process for prediction of next session's return direction. Therefore, we transformed returns into sixthteen equal discrete categorical intervals of states, from high loss (negative return) to positive high return. 95 % of session returns are in the 2 standard deviation distance (2σ) from average. So we computed 2 standard deviations of session returns and separated into sixthteen equal parts between -2σ and 2σ. It might be less than sixthteen equal discrete intervals of states but it leads to reduction of sensitivity of process. On the other hand it might be more than sixthteen equal discrete intervals of states but this brings more computational effort.

In Table 9 of Appendix 1 the total number of return transitions, occurring from the present session to the next session, from states S_i to S_j, was calculated for the whole period considered. As shown in table 9, the number of transitions from state 1 to state 9 is 22; the number of transitions from state 10 to state 17 is 8. If we look at total numbers, we see that the total number of transitions from state 6 to all other states is 478. Similarly, we can see that the total number of transitions from state 18 to all other states is 318. At the same time, we can observe that BIST session returns are most in state 8, 9, 10 and 11 in Table 9 for the whole period considered. So we can say that BIST session returns are most between -0, 0103 and 0, 0103 (neither too positive high return nor too high loss).

We can easily compute the one step (one session) conditional transition probability matrix $P\left(S_j^t \middle| S_i^{t-1}\right)$ from Table 9 of Appendix 1, from state i to state j by dividing the row elements by row total. This matrix is given in Table 10 of Appendix 2.

Here, when the return in state S_1 in the present session, conditional probability of it will be going to S_{17} in the next session is $P(S_{17}^{t} \mid S_{1}^{t-1}) =$ 0,019. Similarly, conditional probability of passing from state S_9 to S_{13} is $P(S_{13}^{t} \mid S_{9}^{t-1}) =$ 0,052. As can be seen in Table 10, each row total of this one step conditional probability matrix equals to 1.

From Table 10 of Appendix 2 probability of non-positive return in the next session (step) given the present state ***i*** can be calculated by equation (5.2).

$$P(R_t \leq 0) \mid S_i^{t-1} = \sum_{j=1}^{9} P\left[(S_{ij}^{t}) \mid S_i^{t-1} \right] \tag{5.2}$$

And probability of positive return in the next session (step) given the present state ***i*** can be calculated by equation (5.3).

$$P(R_t > 0) \mid S_i^{t-1} = 1 - P\left[(R_t \leq 0) \mid S_i^{t-1}) \right] \tag{5.3}$$

Table 1 gives each of the states, return range, percentage of occurrence the states, and probability of positive return in the next session (step) given the present state for the period considered.

Table 1

States, Return Ranges, Percentage of Occurrence of States, and Probability of Positive Return in the Next Step

Present states (S_i^{t-1})	**Return Range**	**% of occurrence of states**	**Probability of positive return for next step given the preset state** $P(R_t \geq 0)\|S_i^{t-1}$
S_1	$R_t \leq$ -0,0412	0,0247	0,3438
S_2	-0,0412 < $R_t \leq$ -0,0361	0,0085	0,3708
S_3	-0,0361< $R_t \leq$ -0,0309	0,0142	0,4054
S_4	-0,0309 < $R_t \leq$ -0,0258	0,0181	0,4574
S_5	-0,0258 < $R_t \leq$ -0,0206	0,0295	0,4379
S_6	-0,0206 < $R_t \leq$ -0,0155	0,0461	0,4623
S_7	-0,0155 < $R_t \leq$ -0,0103	0,0769	0,4642
S_8	-0,0103 < $R_t \leq$ -0,0052	0,1128	0,4919
S_9	-0,0052 < $R_t \leq$ 0,0000	0,1466	0,5221
S_{10}	0,0000 < $R_t \leq$ 0,0052	0,1432	0,5280
S_{11}	0,0052 < $R_t \leq$ 0,0103	0,1228	0,5550
S_{12}	0,0103< $R_t \leq$ 0,0155	0,0852	0,5606
S_{13}	0,0155< $R_t \leq$ 0,0206	0,0551	0,5674
S_{14}	0,0206 < $R_t \leq$ 0,0258	0,0349	0,6271
S_{15}	0,0258 < $R_t \leq$ 0,0309	0,0231	0,5708
S_{16}	0,0309 < $R_t \leq$ 0,0361	0,0152	0,6582
S_{17}	0,0361 < $R_t \leq$ 0,0412	0,0113	0,5847
S_{18}	R_t > 0,0412	0,0307	0,6321

As stated in Table 1, probability of positive return for next step given the present state S_9,…., S_{18} (positive session returns) is higher than the other present states (S_1,…..S_8). This shows that BIST session returns do not give amazing results.

% of occurrence of states was calculated by dividing row totals by the sum of the row totals from Table 9 in Table 1. If we look at the results, we see that the high values of % of occurrence of states belong to S_8, S_9, S_{10} and S_{11}.

5.2.2. Training of the ANN Models

By using the information (probabilities of positive or non positive return in the next step given the present state *i*) provided by Markov chain process in the previous section, we trained three ANN models; Model (0), Model (1) and Model (0, 1) after performing so many experiments.

Table 2

Inputs and Outputs of the Trained Models

	Inputs	Output
Model (0)	$R_{t-1} \le 0$ $P(R_t \le 0)\|S_i^{t-1}$	$R_t \le 0$ $R_t > 0$
Model (1)	$R_{t-1} > 0$ $P(R_t > 0)\|S_i^{t-1}$	$R_t \le 0$ $R_t > 0$
Model (0,1)	R_{t-1} $P(R_t > 0)\|S_i^{t-1}$	$R_t \le 0$ $R_t > 0$

Inputs of the trained models are given in Table 2. Inputs of the Model (0) are non- positive return of the present state and probability of non-positive return in the next state given the present state. Inputs of the Model (1) are positive return of the present state and probability of positive return in the next state given the present state. Inputs of the Model (0, 1) are present state return (either positive or non-positive) and probability of positive return in the next session given the present state i.

Outputs of the three models are the same; prediction of either non-positive or positive returns for the next state and defined by following function:

$$y_i^{'} = \begin{cases} 0, & if\ R_t \leq 0 \\ 1, & if\ R_t > 1 \end{cases} \tag{5.4}$$

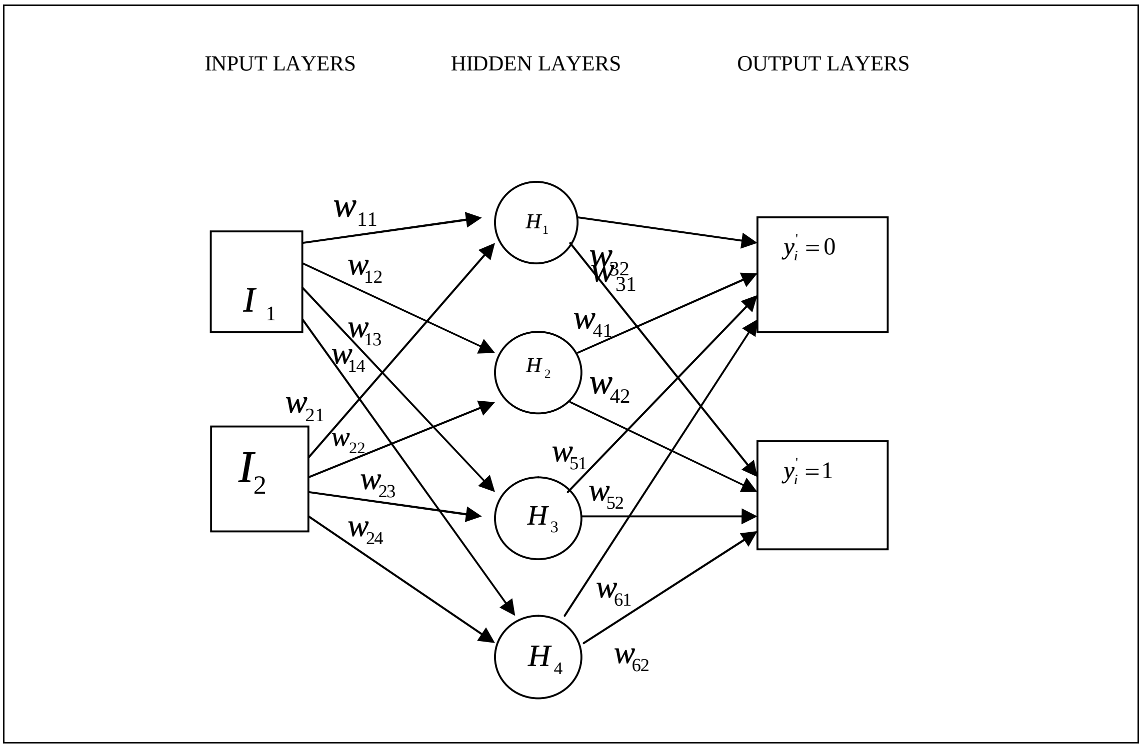


Figure 11. Network Diagram of the Estimated ANN Model (0)

The trained Model (0) has an architecture that is given in Figure 11. The ANN model consists of two input (one of the inputs (I_1) is the probability of non-positive return in the next state and other one (I_2) is the non-positive return of present state) and two output nodes in the input and output layer, and one hidden layer with four nodes between input and output layer. By using gradient descent multilayer perceptron procedure the ANN model can be trained as follows;

Hidden node H_1 contains the following sigmoid function:

$$H_1 = 1/(1 + e^{-z_1}) \quad , \quad z_1 = w_{11} I_1 + w_{21} I_2 \tag{5.5}$$

The other hidden node H_2 contains the following function:

$$H_2 = 1/(1 + e^{-z_2}) \quad , \quad z_2 = w_{12} I_1 + w_{22} I_2 \tag{5.6}$$

The hidden node H_3 contains the following function:

$$H_3 = 1/(1 + e^{-z_3}) \quad , \quad z_3 = w_{13} I_1 + w_{23} I_2 \tag{5.7}$$

The hidden node H_4 contains the following function:

$$H_4 = 1/(1 + e^{-Z_4}) \quad , \quad z_4 = w_{14} I_1 + w_{24} I_2 \tag{5.8}$$

Input of the output node ($y_i^{'} = 0$) is the output from the four hidden nodes H_1, H_2, H_3 and H_4 which are weighted by w_{31}, w_{41}, w_{51} and w_{61};

$$y_i^{'} = w_{31}\left(1/e^{-H_1}\right) + w_{41}\left(1/e^{-H_2}\right) + w_{51}\left(1/e^{-H_3}\right) + w_{61}\left(1/e^{-H_4}\right) \tag{5.9}$$

Similarly, input of the output node ($y_i^{'} = 1$) is the output from the three hidden nodes H_1, H_2, H_3 and H_4 which are weighted by w_{32}, w_{42}, w_{52} and w_{62};

$$y_i^{'} = w_{32}\left(1/e^{-H_1}\right) + w_{42}\left(1/e^{-H_2}\right) + w_{52}\left(1/e^{-H_3}\right) + w_{62}\left(1/e^{-H_4}\right) \tag{5.10}$$

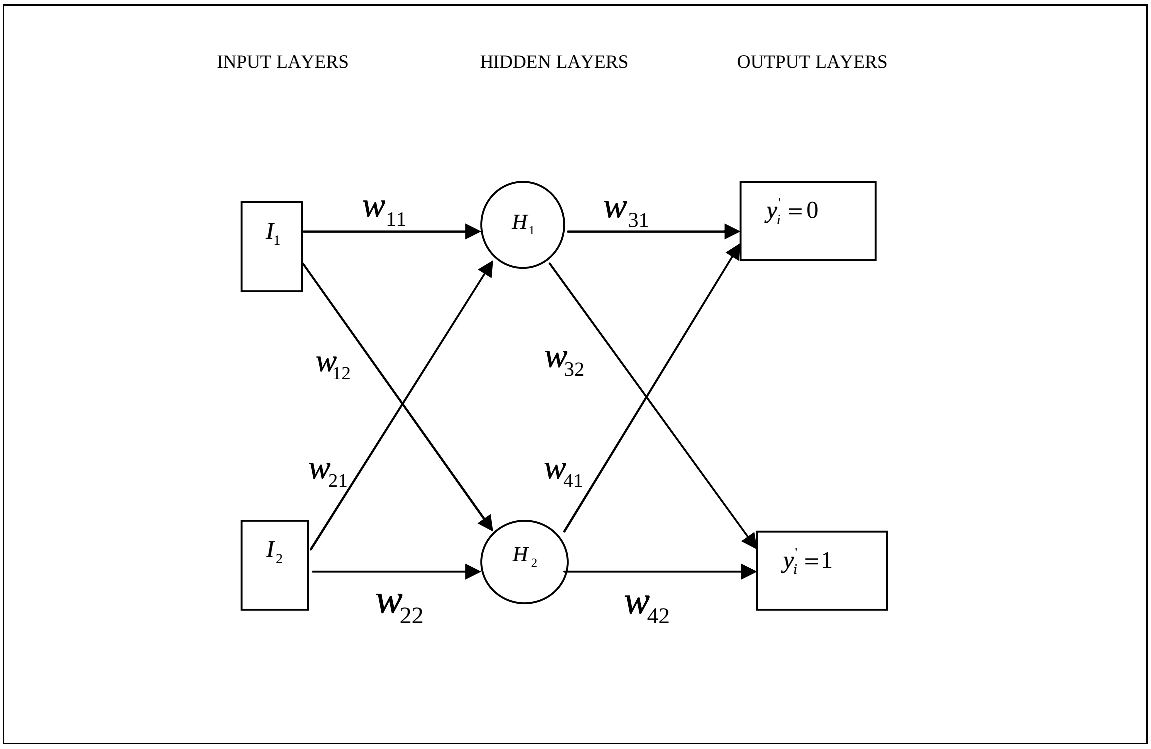


Figure 12. Network Diagram of the Estimated ANN Model (1)

The trained Model (1) has an architecture that is given in Figure 12. The ANN model consists of two input (one of the inputs (I_1) is the probability of non-positive return in the next state and other one (I_2) is the non-positive return of present state) and two output nodes in the input and output layer, and one hidden layer with two nodes between input and output layer. By using gradient descent multilayer perceptron procedure the ANN model can be trained as follows;

Hidden node H_1 contains the following sigmoid function:

$$H_1 = 1/(1 + e^{-Z_1}) \quad , \quad z_1 = w_{11} I_1 + w_{21} I_2 \tag{5.11}$$

Hidden node H_2 contains the following sigmoid function:

$$H_2 = 1/(1+e^{-z_2}) \quad , \quad z_2 = w_{12} I_1 + w_{22} I_2 \tag{5.12}$$

Input of the output node ($y_i^{'}=0$) is the output from the two hidden nodes H_1 and H_2 which are weighted by w_{31} and w_{41};

$$y'_i = w_{31}\left(1/e^{-H_1}\right) + w_{41}\left(1/e^{-H_2}\right) \tag{5.13}$$

Similarly, input of the output node ($y_i^{'}=1$) is the output from the two hidden nodes H_1 and H_2 which are weighted by w_{32} and w_{42};

$$y'_i = w_{32}\left(1/e^{-H_1}\right) + w_{42}\left(1/e^{-H_2}\right) \tag{5.14}$$

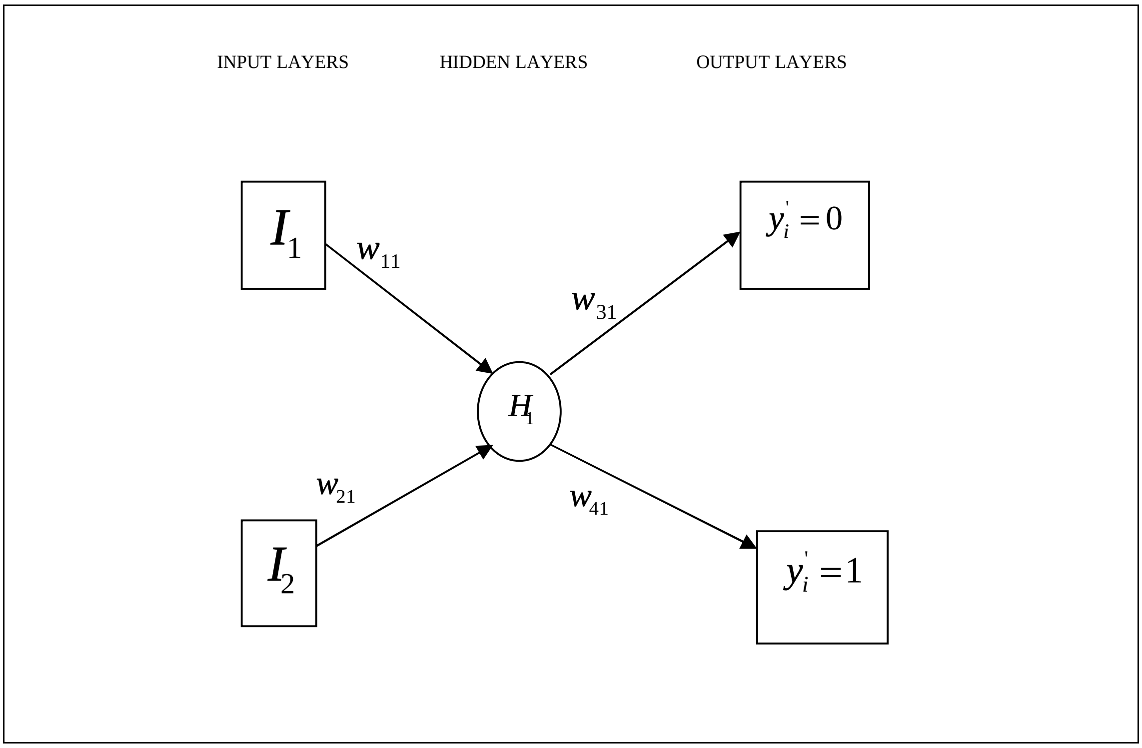


Figure 13. Network Diagram of the Estimated ANN Model (0, 1)

The trained Model (0, 1) has an architecture that is given in Figure 13. The ANN model consists of two input (one of the inputs (I_1) is the probability of non-positive return in the next state and other one (I_2) is the non-positive return of present state) and two output nodes in the input and output layer, and one hidden layer with one node between input and output layer. By using gradient descent multilayer perceptron procedure the ANN model can be trained as follows;

Hidden node H_1 contains the following sigmoid function:

$$H_1 = 1/(1+e^{-z_1}) \quad , \quad z_1 = w_{11} I_1 + w_{21} I_2 \tag{5.15}$$

Input of the output node ($y_i' = 0$) is the output from the one hidden node H_1 which is weighted by w_{31};

$$y'_i = w_{31} \left(1/e^{-H_1} \right) \tag{5.16}$$

Similarly, input of the output node ($y_i' = 1$) is the output from the one hidden node H_1 which is weighted by w_{41};

$$y'_i = w_{41} \left(1/e^{-H_1} \right) \tag{5.17}$$

Therefore, the prediction error is the difference between the actual direction and predicted direction;

$$e_i = y_i - y_i' \tag{5.18}$$

Here, y_i is the actual (observed) direction. The total prediction error (E) can be written as a function of ANN weights (w_{ij});

$$E(w) = \sum_{i=1}^{n} e_i^2 \tag{5.19}$$

Gradient vector of the total prediction error function can be written as;

$$\nabla E(w) = (\partial E / \partial w) \tag{5.20}$$

ANN weights can be adjusted by the gradient descent method;

$$w_{new} = w_{old} + a \, (\partial E / \partial w) \Big|_{w_{old}} \tag{5.21}$$

Here, α = learning parameter (0≤ α≤1). Iteration eventually terminates at a local minimum when $w_{new} \cong w_{old}$.

Table 3

Adjusted Weights for ANN Model (0)

Parameter Estimates

Predictor		Predicted					
		Hidden Layer 1				Output Layer	
		H_1	H_2	H_3	H_4	$y_i^{'}=0$	$y_i^{'}=1$
Input Layer	I_1	-,784	,235	,240	,012		
	I_2	-,534	,997	-,321	,438		
Hidden Layer 1	H_1					-,786	,301
	H_2					-,259	,565
	H_3					-,778	-,036
	H_4					,087	,341

After the training process, an adjusted connection weight for Model (0) is given in Table 3. For example, for Model (0) connection weights between input node I_1 and hidden layer nodes H_1, H_2, H_3 and H_4 (w_{11}, w_{12}, w_{13} *and* w_{14}) are -0.784, 0.235, 0.240 and 0.012 respectively. Similarly connection weights between input node I_2 and hidden layer nodes H_1, H_2, H_3 and H_4 (w_{21}, w_{22}, w_{23} *and* w_{24}) are -0.534, 0.997, 0.321 and 0.438 respectively. Weight between hidden layer node H_1 and output node $y_i^{'}=0$ (w_{31}) is -0.786 and weight between hidden layer node H_1 and output node $y_i^{'}=1$ (w_{32}) is 0.301 respectively. Weight between hidden layer node H_2 and output node $y_i^{'}=0$ (w_{41}) is -0.259 and weight between hidden layer node H_2 and output node $y_i^{'}=1$ (w_{42}) is 0.565 respectively. Weight between hidden layer node H_3 and output node $y_i^{'}=0$ (w_{51}) is -0.778 and weight between hidden layer node H_3 and output node $y_i^{'}=1$ (w_{52}) is -0.036 respectively. Weight between hidden layer node H_4 and output node $y_i^{'}=0$ (w_{61}) is 0.087 and weight between hidden layer node H_4 and output node $y_i^{'}=1$ (w_{62}) is 0.341 respectively.

Table 4

Adjusted Weights for ANN Model (1)

Parameter Estimates

Predictor		Predicted			
		Hidden Layer 1		Output Layer	
		H_1	H_2	$y_i' = 0$	$y_i' = 1$
Input Layer	I_1	,085	,406		
	I_2	-,071	,212		
Hidden Layer 1	H_1			,016	-,247
	H_2			-,176	,349

After the training process, an adjusted connection weight for Model (1) is given in Table 4. For example, for Model (1) connection weights between input node I_1 and hidden layer nodes H_1 and H_2 (w_{11} and w_{12}) are 0.085and 0.406 respectively. Similarly connection weights between input node I_2 and hidden layer nodes H_1 and H_2 (w_{21} and w_{22}) are -0.071 and 0.212 respectively. Weight between hidden layer node H_1 and output node $y_i' = 0$ (w_{31}) is 0.016 and weight between hidden layer node H_1 and output node $y_i' = 1$ (w_{32}) is -0.247 respectively. Weight between hidden layer node H_2 and output node $y_i' = 0$ (w_{41}) is -0.176 and weight between hidden layer node H_2 and output node $y_i' = 1$ (w_{42}) is 0.349 respectively.

Table 5

Adjusted Weights for ANN Model (0, 1)

Parameter Estimates

Predictor		Predicted		
		Hidden Layer 1	Output Layer	
		H_1	$y_i^{'} = 0$	$y_i^{'} = 1$
Input Layer				
	I_1	-,758		
	I_2	-,669		
Hidden Layer 1				
	H_1		,142	-,354

After the training process, an adjusted connection weight for Model (0, 1) is given in Table 5. For example, for Model (0, 1) connection weight between input node I_1 and hidden layer node H_1 (w_{11}) is -0.758 respectively. Similarly connection weight between input node I_2 and hidden layer node H_1 (w_{21}) is -0.669 respectively. Weight between hidden layer node H_1 and output node $y_i^{'} = 0$ (w_{31}) is 0.142 and weight between hidden layer node H_1 and output node $y_i^{'} = 1$ (w_{32}) is -0.354 respectively.

Table 6

Classification Achievement of Estimated ANN Model (0)

Classification

Sample	Observed	Predicted		
		D≤0	D>1	Percent Correct
Training	D≤0	770	510	60,2%
	D>1	563	610	52,0%
	Overall Percent	54,3%	45,7%	56,3%
Testing	D≤0	792	498	61,4%
	D>1	584	592	50,3%
	Overall Percent	55,8%	44,2%	56,1%

Table 6 gives observed and predicted classification results of direction of session returns of BIST by the estimated ANN Model (0). Classification result is given for the training and testing sample. In the study the sample data was randomly divided two equal groups as training and testing sample; training sample was used to train (estimate) the model, testing sample was used to evaluate the model in terms of the classification achievements. In Table 6 for the testing sample we can see that if the present return is non-positive, the Model (0) predicts next session non positive direction ($R_t \leq 0$) as 61.4% correctly and next session positive direction ($R_t > 0$) as 50.3%.

Table 7

Classification Achievement of Estimated ANN Model (1)

Classification

Sample	Observed	Predicted D≤0	Predicted D>1	Predicted Percent Correct
Training	D≤0	84	1089	7,2%
	D>1	73	1426	95,1%
	Overall Percent	5,9%	94,1%	56,5%
Testing	D≤0	84	1093	7,1%
	D>1	80	1508	95,0%
	Overall Percent	5,9%	94,1%	57,6%

Table 7 gives observed and predicted classification results of direction of session returns of BIST by the estimated ANN Model (1). Classification result is given for the training and testing sample. In the study the sample data was randomly divided two equal groups as training and testing sample; training sample was used to train (estimate) the model, testing sample was used to evaluate the model in terms of the classification achievements. In Table 7 for the testing sample we can see that if the present return is positive, the Model (1) predicts next day positive direction (R_t>0) as 95% correctly. However, if the present return is positive, the Model (1) does not accurately predicts negative direction (7.1%).

Table 8

Classification Achievement of Estimated ANN Model (0, 1)

Classification

Sample	Observed	Predicted D≤0	Predicted D>1	Predicted Percent Correct
Training	D≤0	1416	1093	56,4%
	D>1	1234	1444	53,9%
	Overall Percent	51,1%	48,9%	55,1%
Testing	D≤0	1426	985	59,1%
	D>1	1226	1532	55,5%
	Overall Percent	51,3%	48,7%	57,2%

Table 8 gives observed and predicted classification results of direction of session returns of BIST by the estimated ANN Model (0, 1). Classification result is given for the training and testing sample. In the study the sample data was randomly divided two equal groups as training and testing sample; training sample was used to train (estimate) the model, testing sample was used to evaluate the model in terms of the classification achievements. In Table 8 for the testing sample we can see that if the present return is either positive or non-positive the Model (0, 1) correctly predicts next session non-positive return 59.1% and next session positive return 55.5%.

In order to eliminate these unreliable predictions and to make more accurate prediction, the three models can be integrated together for the prediction process.

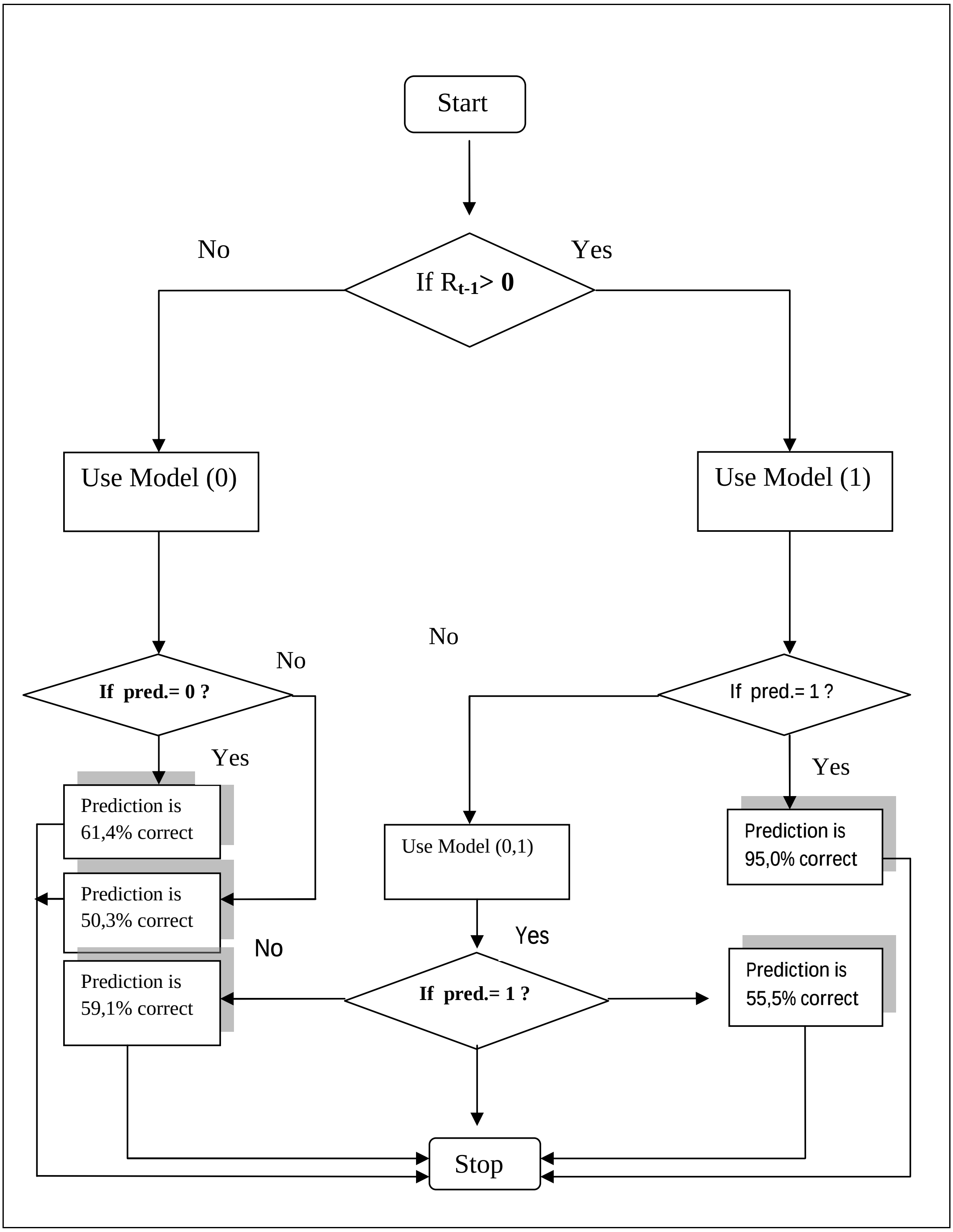


Figure 14. Flowchart of Using the Three Combined Models for Prediction

Flowchart of integrated use of these models in prediction process is given in Figure 14. Here, if the present return is positive, the Model (1) should be used. If the Model (1) predicts positive return, prediction is 95% correct, stop the prediction. If the Model (1) predicts non-positive return do not use model (1); use model (0, 1). If Model

(0, 1) predicts positive return, prediction is 55.5% correct, stop prediction. If Model (0, 1) predicts non-positive direction it is 59.1% correct, stop prediction.

If the present return is non-positive, the Model (0) should be used. If the Model (0) predicts non-positive return, prediction is 61.4% correct, stop the prediction. If the Model (0) predicts positive return it is 59.1% correct, stop the prediction.

Therefore, as an average integrated use of the three models provides 64.26% correct prediction for the direction of returns. This means that if an investor determines his/her session buying-selling strategy according to result of the prediction of the integrated models, his/her session investment strategy will be profitable 64.26% of the time in the long run.

CHAPTER VI

RESULT AND FUTURE RESEARCH PERSPECTIVES

6.1. Result and Future Research Perspectives

In developing countries such as Turkey, it is very hard to estimate the stock market returns because of many kinds of speculative movements in markets. That's why financial analysts use various methods to determine the stock return movements. The most important ones of these methods are Artificial Neural Networks (ANNs) and Markov chains. For this reason we tried to estimate the direction of session returns of Borsa Istanbul 100 Index by integrating Markov chains and Artificial Neural Network models.

The objective of the study is to estimate the direction of next session return and research the investment opportunities of the Borsa Istanbul 100 (BIST 100) index when passing from one session to another. So, we used 10357 session closing values of the Borsa Istanbul 100 (BIST 100) index between the period of January 04, 1988-April 04, 2012. The data were obtained from the electronic data delivery system of the Borsa Istanbul (http://borsaistanbul.com/).

In this study, we first computed the daily session returns (R_t) as a percentage change of the BIST 100 index session closing values (P_t) in (5.1) and found average (expected) return (m_R) as approximately 0.0011 and standard deviation (s_R) as 0.0206 for the period considered. Because the standard deviation is extremely high in comparison to expected return, we presented that BIST 100 index return exhibits high dispersion and this is enormous risk for the investors.

After computing the session returns, we supposed session returns to be a stochastic process with sixthteen discrete state spaces and eighteen states $\{S_1, S_2, S_3, ..., S_{17}, S_{18}\}$ with Markov chain so as to calculate probability of positive/ non-positive return for the next session given the present session. In Table 9 of Appendix 1, the total number of return transitions, occurring from the present session to the next session, from state S_i to S_j, was calculated for the whole period considered. And than we computed the one step (one session) conditional transition probability matrix from

Table 9 of Appendix 1, from state i to state j by dividing the row elements by row total in Table 10 of Appendix 2.

Probability of non-positive and positive returns were calculated in equations (5.2) and (5.3) from Table 10 of Appendix 2. And after these steps, states, return ranges, percentage of occurrence of states and probability of positive return in the next step were given in Table 1.

By using the probabilities of positive or non-positive return in the next step given the present state i provided by Markov chain process, we trained three ANN models as Model (0), Model (1) and Model (0, 1). The inputs and outputs for Model (0), Model (1) and Model (0, 1) were given in Table 2. Inputs of the Model (0) are non-positive return of the present state and probability of non-positive return in the next state given the present state. Inputs of the Model (1) are positive return of the present state and probability of positive return in the next state given the present state. Inputs of the Model (0, 1) are present state return (either positive or non-positive) and probability of positive return in the next day given the present state i. Outputs of the three models are the same; prediction of either non-positive or positive returns for the next state and defined by the function (5.4).

And then these three models' network diagrams were presented in Figure 11, 12 and 13. The ANN model (0) consists of two input and two output nodes in the input and output layer, and one hidden layer with four nodes between input and output layer. The ANN model (1) consists of two input and two output nodes in the input and output layer, and one hidden layer with two nodes between input and output layer. The ANN model (0, 1) consists of two input and two output nodes in the input and output layer, and one hidden layer with one node between input and output layer. By using gradient descent multilayer perceptron procedure the three ANN models trained as function (5.5),……., (5.21). After the training process, adjusted connection weights for Model (0), Model (1) and Model (0, 1) were given in Table 3, 4 and 5. We presented the observed and predicted classification results of direction of session returns of BIST for the estimated ANN Model (0) in Table 6. Classification results were given for the training and testing samples. In the study the sample data was randomly divided two equal groups as training and testing sample; training sample was used to train (estimate) the models, testing sample was used to evaluate the models in terms of the classification achievements. We found that if the present return is non-positive, the Model (0) predicts next session non positive direction ($R_t \leq 0$) as 61,4% correctly and next session positive direction (R_t>0) as 50.3%. We presented the observed and predicted classification results of direction of session returns of BIST for the estimated ANN Model (1) in Table 7. If the present return is positive, the Model (1) predicts next day positive direction (R_t>0) as 95% correctly. However, if the present return is positive, the Model (1) does not accurately predicts negative direction (7.1%). We presented the observed and predicted classification results of direction of session returns of BIST for the estimated ANN Model (0, 1) in Table 8. If the present return is either positive or non-positive the Model (0, 1) correctly predicts next session non-positive return 59.1% and next session positive return 55.5%.

Lastly, we integrated the three models together for the prediction process in order to eliminate these unreliable predictions and to make more accurate prediction. Flowchart of integrated use of these models in prediction process was given in Figure 14. Here, if the present return is positive, the Model (1) should be used. If the Model (1) predicts positive return, prediction is 95% correct, stop the prediction. If the Model (1) predicts non-positive return do not use model (1); use model (0, 1). If Model (0, 1) predicts positive direction it is prediction 55.5% correct, stop prediction. If Model (0, 1) predicts non-positive direction it is 59.1% correct stop prediction. If the present return is non-positive, the Model (0) should be used. If the Model (0) predicts non-positive return, prediction is 61.4% correct stop the prediction. If the Model (0) predicts positive return it is 59.1% correct, stop the prediction.

Therefore, as an average integrated use of the three models provides 64.26% correct prediction for the direction of returns. This means that if an investor determines his/her session buying-selling strategy according to result of the prediction of the integrated models, his/her session investment strategy will be profitable 64.26% of the time in the long run.

Similar further analysis can be performed for the returns of individual common stocks and other investment instruments such as gold and other foreign exchange returns.

We used session returns in this study. Further similar analysis can also be performed by considering returns of smaller time intervals, such as an intraday hourly change. Therefore, using small time intervals may provide more information.

BIBLIOGRAPHY

Akpınar, H. (1994). Yapay Sinir Ağları, Gelişimi ve Yapılarının İncelenmesi. *İ.Ü. İşletme Fakültesi Dergisi*, 1, 41-78.

Anderson, J.A. (1972). A Simple Neural Network Generation on Interactive Memory. *Mathematical Biosciences*, 14, 197-220.

Avcı, E. and Cinko, M. (2008). Endeks Getirilerinin Yapay Sinir Ağları Modelleri ile Tahmin Edilmesi: Gelişmekte Olan Avrupa Borsaları Uygulaması. *Iktisat, Isletme ve Finans*, 23(266), 2008, 114-137.

Aygören, H., Sarıtaş, H. and Moralı, T. (2012). İMKB 100 Endeksinin Yapay Sinir Ağları ve Newton Nümerik Arama Modelleri ile Tahmini. *International Jounal of Alanya Faculty of Business*, 4(1), 2012, 73-88.

Base, N. K. and Liang, P. (1996). *Neural Network Fundamentals with Graphs, Algorithms and Applications*. McGraw-Hill, New York.

Berry, M. J. A. and Linoff, G.S. (2004). *Data Mining Techniques for Marketing, Sales, and Customer Relationship Management*. Wiley Publishing, United States of America.

Bhargava, T.N. and Flietz, B.D. (1973). The Behavior of Stock-Price Relatives-A Markovian Anaysis. *Operations Research*, 21(6), (November-December, 1973), 1183-1199.

Canbaş, S., Düzakın, H. and Kılıç S.B. (2002). Fundamental and Macroeconomic Information for Common Stock Valuation: The Turkish Case. *Yapı Kredi Economic Review*, 13(1).

Cao, Q., Leggio, K.B. and Schniederjans M. J. (2005). A Comparison Between Fama and French's Model and Artifical Neural Networks in Predicting the Chinese Stock Market. *Computers & Operations Research*, 32(10), October 2005, 2499-2512.

Chen, C.W.S. and So, M. K. P. (2006). On a Thereshold Heteroscedastic Model. *International Journal of Forecasting*, 22(1), January-March 2006, 73-89.

Dharu and Khadayar. http://laurier.vsb.bc.ca/studentp/Dharu/Facts.html (Access Date: 10.12.2012).

Diler A. İ. (2003). İMKB Ulusal-100 Endeksinin Yönünün Yapay Sinir Ağları Hata Geriye Yayma Yöntemi ile Tahmin Edilmesi. *İMKB Dergisi*, 7(25-26), January-July 2003, 66-81.

Driffill, J. and Sola, M. (1998). Intrinsic Bubbles and Regime-Switching. *Journal of Monetary Economics*, 42(2), 357.

Dutta, G., Jha, P., Laha, A.K. and Mohan, N. (2006). Artifical Neural Network Models for Forecasting Stock Price Index in the Bombay Stock Exchange. *Journal of Emerging Market Finance*, May 2006, 283.

Energy and Wetlands Research Group. http://wgbis.ces.iisc.ernet.in/energy/paper/TR-111/chapter3.htm (Access Date: 13.12.2012).

Eugene F.F. (1965). Random Walks in Stock Market Prices. *Financial Analysts Journal*, September / October.

Fausett, L. (1994). *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*. Prentice-Hall, New Jersey.

Feldman, J.A. and Ballard, D.H. (1982). Connectionist Models and Their Properties. *Cognitive Science*, 6, 205-254.

Fernandez-Rodriguez, F., Gonzales-Martel, C. and Sosvilla-Rivero, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid Stock Market. *Economics Letters*, 69(1), October 2000, 89-94.

Freisleben, B. (1992). Stock Market Prediction with Backpropagation Networks. *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems Lecture Notes in Computer Science*, 604, 1992, 451-460.

Fu L. (1994). *Neural Networks in Computer Intelligence*. McGraw-Hill, Singapore

Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by shift in position, *Biological Cybernetics*, 36, 193-202.

Goonatilake, R. and Herath, S. (2007). The Volatility of the Stock Market and News. *International Research Journal of Finance and Economics*, 11, 53-65.

Greyserman , A., Jones, D. H. and Strawderman, W. E. (2006). Portfolio Selection Using Hierarchical Bayesian Analysis and MCMC Methods. *Journal of Banking & Finance*, 30(2), February 2006, 669-678.

Griffin, J. E. and Steel, M. F. J. (2006). Inference with Non-Gaussian Ornstein-Uhlenbeck Processes for Stochastic Volatility. *Journal of Econometrics*, 134(2), October 2006, 605-644.

Grossberg, S.A. (1973). Contour Enhancement, Short Term Memory, and Consultancies in Reverberating Neural Networks. *Studies in Applied Mathematics*, 52(3), 213-257.

Guidolin, M. and Timmermann, A. (2007). Properties of Equilibrium Asset Prices under Alternative Learning Schemes. *Journal of Economic Dynamics and Control*, 31(1), January 2007, 161-217.

Hamilton J. D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrics*, 57, 357-384.

Hamilton J. D. and Susmel, R. (1994). Autoregressive Conditional Heteroscedasticty and Changes in Regime. *Journal of Econometrics*, 64(1-2), September-October 1994, 307-333.

Hebb, D.O. (1949). *The Organization of Behavior*. New York, Wiley, Chapter 4, 60-78.

Hiller, F. and Lieberman, G.J. (2005). *Introduction to Operations Research*. McGraw Hill, New York.

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National. *Academy of Sciences*, 79, 2554-2558

Idolor, E. J. (2010). Security Prices as Markov Process. *International Research Journal of Finance and Economics*, ISSN 1450-2887, 59.

Idolor, E. J. (2011). The Long Run Prospect of Stocks in the Nigerian Capital Market: A Markovian Analysis. *JORIND*, 9, June 2011, 388-400.

Kanas, A. (2001). Neural Network Linear Forecasts for Stock Returns. *International Journal of Finance & Economics*, 6(3), July 2001, 245-254.

Kanas, A. (2003). Non-linear Forecasts of Stock Returns. *Journal of Forecasting*, 22(4), 299.

Karaatlı, M., Güngör, İ., Demir, Y. and Kalaycı, Ş. http://www.bjmer.net/Makaleler/1857451686_38-48%20meltem-ibrahim-yusuf-%c5%9feref.PDF (Access Date: 22.07.2013)

Kılıç, S. B. (1997). İMKB'de Zayıf Etkinlik ve Rassal Yürüyüş". *III. Ulusal Ekonometri ve İstatistik Sempozyumu Dergisi*, 29-30 Mayıs, 1997, Uludağ Üniversitesi, Bursa.

Kılıç, S. B. (2005) . Test of the Weak Form Efficent Market Hypothesis for the Istanbul Stock Exchange by Markov Chains Methodology. *University of Social Sciences of Cukurova*, 14(1), 333-342.

Kılıç, S. B. (2013, May). Integrating Artificial Neural Network Models by Markov Chain Process: Forecasting the Movement Direction of Turkish Lira/US Dollar Exchange Rate Returns. *XIV th International Symposium on Econometrics, Operation Research and Statistics 2013.*

Kiff, J. http://psychology.wikia.com/wiki/Artificial_neural_network, (Access Date: 4.12.2012).

Kim, K. and Han, I. (2000). Genetic Algorithms Approach to Feature Discretization in Artifical Neural Networks for the Prediction of Stock Price Index. *Expert Systems with Applications*, 19(2), August 2000, 125-132.

Kim, K. and Lee, W.B. (2004). Stock Market Prediction Using Artificial Neural Networks with Optimal Feature Transformation. *Neural Comput & Applic*, 13, 2004, 255-260.

Kimoto, T., Asakawa, K.,Yoda, M. and Takeoka, M. (1990).Stock Market Prediction System with Modular Neural Network. *Proceedings of the International Joint Conference on Neural Networks*, 1-6.

Kohonen, T. (1972). Correlation matrix memories. *IEEE Transaction on Computers*, C-21(4), 353-359.

Kutlu, B. and Badur, B. (2009). Yapay Sinir Ağları ile Borsa Endeksi Tahmini. *Yönetim*, 63, July 2009, 25-40.

Lin, S. K., Wang , S.Y and Tsai, P. L. (2009). Application of Hidden Markov Switching Moving Average Model in the Stock Markets: Theory and Emprical Evidence. *International Review of Economics & Finance*, 18(2), March 2009, 306-317.

Lisboa, P.J.G., Nagar, R. and Nageim, H.A., (2007) "Comparison of neural network and binary logistic regression methods in conceptual design of tall steel buildings", *Construction Innovation: Information, Process, Management*, 7(3), 240 – 253.

Liu, H. (2011). Dyanamic Portfolio Choice Under Ambiguity and Regime Switching Mean Returns. *Journal of Economic Dynamics and Control*, 35(4), April 2011, 623-640.

Los, C. A. (1998). Nonparametric Efficiency Testing of Asian Stock Markets Using Weekly Data. *Yale School of Management's Economics Research Network*, 1-30.

McClelland, J. L. and Rumelhart, D.E. (1986). Parallel Distributed Processing, Explorations in the Microstructure of Cognitio, 2, Psychological and biological models. *MIT Press*, Cambridge, MA.

McCulloch, W.S. and Pitts, W.A. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Buttetin of Mathematics and Biophysics*, 5, 115-133.

McGraw-Hill Companies (2010), http://www.education.com/study-help/article/structure-function-neuron/, (Access Date: 10.12.2012).

McQueen, G., and Thorley, S. , (1991). Are Stock Returns Predictable? A Test Using Markov Chains. *Journal of Finance*, 46(1), 239.

Mills, T.C. and Jordanov, J.V. , (2003). The Size Effect and the Random Walk Hypothesis: Evidence from the London Stock Exchange Using Markov Chains. *Applied Financial Economics*, 13.

Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT Press, Cambridge, MA.

Mitra S.S. and Riggieri M.J. (2011). "Predicting Stock Prices". http://www.wpi.edu/ Jaunuary 2011.

Mizuno, H., Kosaka, M., Yajima, H. and Komoda, N. (1998). Application of Neural Network to Technical Analysis of Stock Market Prediction. http://sic.ici.ro/sic1998_2/art03.html, (Access Date: 11.03.2013).

Muradoğlu, G. and Metin K. , (1997). "Efficiency of the Turkish Stock Exchange with Respect to Monetary Variables: A Cointegration Analysis". *European Journal of Operational Research*, 90(3), 566-575.

Neal, R.M. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical Report CRG-TR-93-1. *Department of Computer Science*, University of Toronto.

Nelson, C. R., Startz, R. and Turner, C. M. (1989). A Markov Model of Heteroscedastisity Risk and Learning in the Stock market. *Journal of Financial Economics*, 25, 1989, 3-22.

NeuroAI (2012). http://www.learnartificialneuralnetworks.com/ (Access Date: 13.12.2012).

O'Connor, N. and Madden, M. G. (2005). A Neural Network Approach to Predicting Stock Exchange Movements Using External Factors. *Knowledge-Based Systems*, 19(5), February 2006, 371-378.

Olson, D. and Mossman, C. (2003). Neural Network Forecasts of Canadian Stock Returns Using Accounting Ratios. *International Journal of Forecasting*, 19(3), July-September 2003, 453-465.

Or, İ. (1986). *Introduction Stochastic Models in Operation Research.* Boğaziçi University Publications, Istanbul.

Öztemel, E. (2012). *Yapay Sinir Ağları*. Papatya Publications, Istanbul.

Parker, D.B. (1985). Learning-logic. *M.I.T. Cen. Computational Res. Economics Management Sci*. Cambridge, MA.

Radford M.N. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods, Technical Report CRG-TR-93-1. *Department of Computer Science*, University of Toronto, 25 September 1993.

Rosenblatt, F. (1958). *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Psychoanalytic Review, 65, 386-408

Rumelhart, D.E., Hinton, D.E. and Williams, R.J. (1986). Learning Representation by Backpropagation errors. *Nature*, 323(9), 533-536.

Plummer, A.E. (2000). *Time Series Forecasting with Feed Forward Neural Networks: Guidelines and Limitations.* Master of Science in Computer Science, University of Wyoming, Wyoming.

Ryan, T.M. (1973). Security Prices as Markov Processes. *Journal of Financial & Quantitative Analysis*, 8(1), 17- 20.

Schalkoff, R. J. (1997). *Artificial Neural Networks*. McGraw-Hill, New York.

Sejnowski, T.J and Rosenberg, C.R. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, 1, 145-168.

Taha, A.H. (2003). *Operations Research: An Introduction.* Pearson Education, New Jersey.

Takaki, H. (2004). A Discrete-Time Model of High-Frequency Stock Return. *Quantitative Finance*, 4(2), 140.

Tektaş, A. and Karatas, A. (2004). Yapay Sinir Ağları ve Finans Alanına Uygulanması: Hisse Senedi Fiyat Tahminleri. *İktisadi ve Idari Bilimler Dergisi*, 18(3-4), September 2004, 337-349.

Tsionas, E.G. (2000). Bayesian Model Comparison by Markov Chain Simulation: Illustration Using Stock Market Data. *Research in Economics*, 54(4), December 2000, 403-416.

Turner, C.M., Startz, R. and Nelson, C. R. (1989). A Markov Model of Heteroscedasticity Risk and Learning in the Stock Market. *Jourrnal of Financial Economics*, 25, 3-22.

University of Colorado at Boulder, http://www.teachengineering.org/view_lesson.php?url=collection/cub_/lessons/cub_human/cub_human_lesson06.xml, (Access Date: 10.12.2012).

Vasanthi, S., Subha, M.V. and Nambi, T. (2011). "An Emprical Study on Stock Index Trend Prediction Using Markov Chain Analysis". *JBFSIR*, 1(1), April 2011, 72-91.

Widrow, B. and Hoff, M.E. (1960). Adaptive Switching Circuits. *WESTCON Convention*, Record Part IV, 1960, 96-104.

Winston, L.W. (2004). *Operations Research Applications and Algorithms*. Thomson Learning, Canada.

Zhang, X. and King, M.L. (2008). Box-Cox Stochastic Volatility Models with Heavy-Tails and Correlated Errors. *Journal of Emprical Finance*, 15(3), June 2008, 549-566.

LIST OF APPENDICES

Appendix 1

Table 9

Number of Occurrence of the Transitions from States S_i to S_j

		STATE- t																		Total
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
STATE t-1	1	43	12	9	16	14	17	19	16	22	14	15	16	6	9	3	5	5	15	256
	2	7	2	4	1	11	5	10	5	11	9	3	3	2	5	3	1	2	5	89
	3	5	4	6	6	13	10	11	17	16	12	16	9	5	3	5	2	2	6	148
	4	8	4	4	6	15	15	13	20	17	25	23	7	9	7	4	4	1	6	188
	5	3	8	10	7	13	24	35	36	36	32	21	23	16	10	10	2	5	15	306
	6	15	6	11	13	13	36	39	60	64	63	46	39	24	13	12	8	4	12	478
	7	18	3	15	18	28	33	82	104	126	114	88	62	37	16	14	10	10	19	797
	8	17	7	10	19	37	55	102	170	177	170	145	88	67	39	23	13	9	21	1169
	9	17	4	12	20	34	67	114	164	294	280	207	115	80	49	26	9	7	20	1519
	10	17	7	15	18	24	55	133	193	238	259	207	139	71	36	24	24	8	15	1483
	11	20	6	12	15	27	54	76	153	203	198	192	127	69	51	30	13	7	19	1272
	12	11	9	10	16	23	40	51	92	136	124	127	94	58	31	25	9	11	16	883
	13	11	5	11	9	14	19	43	52	83	73	73	52	37	30	15	9	8	27	571
	14	11	2	7	6	12	15	18	29	35	45	42	43	23	17	14	11	12	20	362
	15	12	1	1	3	12	8	13	21	32	21	22	21	28	18	6	4	6	11	240
	16	5	4	3	2	4	7	10	10	9	16	18	18	11	6	9	9	3	14	158
	17	9	1	2	1	3	5	12	10	6	10	11	15	6	4	5	4	3	11	118
	18	27	4	6	12	9	13	16	17	13	18	16	12	22	18	13	21	15	66	318
Total		256	89	148	188	306	478	797	1169	1518	1483	1272	883	571	362	241	158	118	318	10355

Appendix 2

Table 10

One Step Conditional Transition Probability Matrix from S_i to S_j

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	*Total*
S_1	0,168	0,046	0,035	0,062	0,054	0,066	0,074	0,062	0,085	0,054	0,058	0,062	0,023	0,035	0,011	0,019	0,019	0,058	1,000
S_2	0,078	0,022	0,044	0,011	0,123	0,056	0,112	0,056	0,123	0,101	0,033	0,033	0,022	0,056	0,033	0,011	0,022	0,056	1,000
S_3	0,033	0,027	0,040	0,040	0,087	0,067	0,074	0,114	0,108	0,081	0,108	0,060	0,033	0,020	0,033	0,013	0,013	0,040	1,000
S_4	0,042	0,021	0,021	0,031	0,079	0,079	0,069	0,106	0,090	0,133	0,122	0,037	0,047	0,037	0,021	0,021	0,005	0,031	1,000
S_5	0,009	0,026	0,032	0,022	0,042	0,078	0,114	0,117	0,117	0,104	0,068	0,075	0,052	0,032	0,032	0,006	0,016	0,049	1,000
S_6	0,031	0,012	0,023	0,027	0,027	0,075	0,081	0,125	0,133	0,131	0,096	0,081	0,050	0,027	0,025	0,016	0,008	0,025	1,000
S_7	0,022	0,003	0,018	0,022	0,035	0,041	0,102	0,130	0,158	0,143	0,110	0,077	0,046	0,020	0,017	0,012	0,012	0,023	1,000
S_8	0,014	0,006	0,008	0,016	0,031	0,047	0,087	0,145	0,151	0,145	0,124	0,075	0,057	0,033	0,019	0,011	0,007	0,018	1,000
S_9	0,011	0,002	0,007	0,013	0,022	0,044	0,075	0,108	0,193	0,184	0,136	0,075	0,052	0,032	0,017	0,005	0,004	0,013	1,000
S_{10}	0,011	0,004	0,010	0,012	0,016	0,037	0,089	0,130	0,160	0,174	0,139	0,093	0,047	0,024	0,016	0,016	0,005	0,010	1,000
S_{11}	0,015	0,004	0,009	0,011	0,021	0,042	0,059	0,120	0,159	0,155	0,150	0,099	0,054	0,040	0,023	0,010	0,005	0,014	1,000
S_{12}	0,012	0,010	0,011	0,018	0,026	0,045	0,057	0,104	0,154	0,140	0,143	0,106	0,065	0,035	0,028	0,010	0,012	0,018	1,000
S_{13}	0,019	0,008	0,019	0,015	0,024	0,033	0,075	0,091	0,145	0,127	0,127	0,091	0,064	0,052	0,026	0,015	0,014	0,047	1,000
S_{14}	0,030	0,005	0,019	0,016	0,033	0,041	0,049	0,080	0,096	0,124	0,116	0,118	0,063	0,047	0,038	0,030	0,033	0,055	1,000
S_{15}	0,050	0,004	0,004	0,012	0,050	0,033	0,054	0,087	0,133	0,087	0,091	0,087	0,116	0,075	0,025	0,016	0,025	0,045	1,000
S_{16}	0,031	0,025	0,019	0,012	0,025	0,044	0,063	0,063	0,057	0,101	0,113	0,113	0,069	0,038	0,057	0,057	0,019	0,088	1,000
S_{17}	0,076	0,008	0,016	0,008	0,025	0,042	0,101	0,084	0,050	0,084	0,093	0,127	0,050	0,033	0,042	0,033	0,025	0,093	1,000
S_{18}	0,084	0,012	0,018	0,037	0,028	0,040	0,050	0,053	0,040	0,056	0,050	0,037	0,069	0,056	0,040	0,066	0,047	0,207	1,000

CURRICULUM VITAE

PERSONEL INFORMATIONS

Surname, Name : KOŞAR, Çiğdem

Date of Birth : 21.09.1987

Place of Birth : SEYHAN/ ADANA

Marital Status : Single

Address : Çukurova University, Faculty of Economics and Administrative Sciences, I. Block, Department of Econometrics, Office No:312, 01330, Sarıçam, ADANA.

Email Address : ckosar@cu.edu.tr

Phone Number : (0322) 3387254-166

EDUCATIONAL STATUS

2010-2013 (MS) : Çukurova University Social Sciences Enstitute, Department of Econometrics , Adana

2006-2010 (BS) : Çukurova University Faculty of Arts and Sciences, Department of Mathematics, Adana

2002-2005 : High school, Adana Ticaret Odası Anatolian High School, Adana

WORK EXPERIENCE

2012- Present : Research Assistant, Çukurova University, Faculty of Economics and Administrative Sciences, Deparment of Econometrics, Adana.