# Application of Neural Networks in Finance and Investment Decisions

Prof. A K Mittal[1]


## Abstract

Artificial Neural networks are revolutionizing almost all decision making process in financial and investment area. Leading financial firms worldwide are making use of neural networks to tackle difficult tasks requiring the detection of data patterns which elude conventional analytical techniques. Neural networks are already being used to trade the stock, currency and commodity markets, to forecast the economy and to analyze credit risk. Neural networks can draw conclusions from incomplete data, recognize patterns as they unfold in real time and forecast the future. They can even learn from past mistakes and improve decision making accurately. ANN is a non-linear, statistical computational model, having adaptive system that changes its structure based on external or internal information that flows through the network during the processing phase. They are used to model complex relationships between inputs and outputs and find successive patterns. The novelty of neural network lies in their ability to discover non- linear relationship in the input data set without a priori assumption of the knowledge of relation between the input and output. . With the correct implementation, ANNs can be used naturally in online learning and large data set applications.


**Keywords**: Computational Model, trained neural networks, adaptive learning, multi-layered neural networks

## Literature Review

The Traditional Time Series Prediction analyzes historic data and attempts to approximate future values of a time series as a linear combination of these historic data. In econometrics there are two basic types of time series forecasting: univariate (simple regression) and multivariate (multivariate regression). These types of regression models are the most common tools used in econometrics to predict time series. The way they are applied in practice is that firstly a set of factors that influence (or more specific is assumed that influence) the series under prediction is

[1] Professor, GJIMT, Mohali. E-mail address: mittal.ak768@gmail.com Mobile no. - 9872254768

formed. These factors are the explanatory variables of the prediction model. Regression models have been used to predict stock market time series.

**Machine Learning Methods:** Several methods for inductive learning have been developed over a period of time, under the label of machine learning. All these methods use a set of samples to generate an underling function that generated the data. The aim is to draw conclusions from these samples in such way that when unseen data are presented to a model it is possible to infer the value of desired variable from these data.

Eugene Fama (1970) “Efficient Capital Markets”, the efficient market hypothesis (EMH) states that stock prices reflect full available information and no opportunities to earn extra profit. Another theory related to EMH is Random Walk Theory which states that all future prices do not follow any trend or pattern and are random departure from previous prices.

Economists have established an opposite theory called ‘Inefficient market Hypothesis’ (IMH). IMH states that markets are not always efficient and not always in random walk (Pan Heping, 2003)

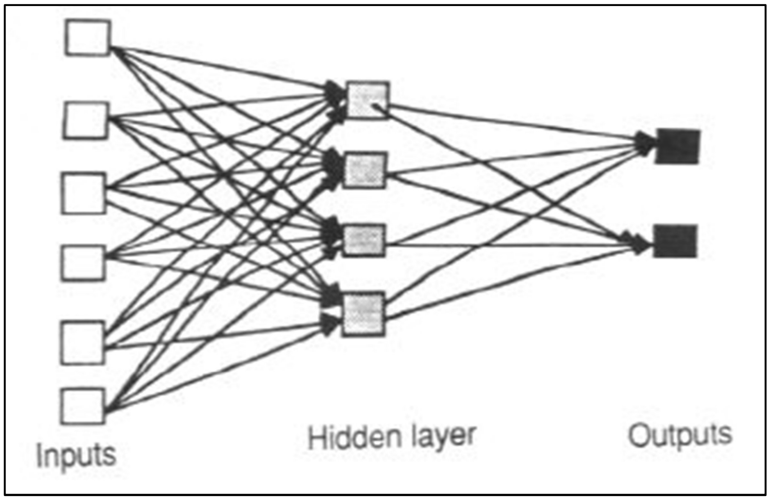
Many researchers and practitioners have proposed many models using various fundamental, technical and analytical techniques for prediction purpose. Fundamental analysis involves the in depth analysis of the changes of the stock prices in terms of exogenous macroeconomic variables. Technical analysis is based on prevailing price, volume, open interest and statistical charts to predict future stock movements.

The analytical technique uses time series forecasting tools. The past data of the prediction variable is analyzed and modeled to capture the patterns of the historic changes in the variable. These models are then used to forecast the future prices. Time series mostly uses linear methods like moving averages, exponential smoothing, and time series regression. One of the most popular linear method is the ’Autoregressive Integrated Moving Average’ (ARIMA) model by Box and Jenkins.

However, stock market returns are not linear most of the time as the residual variance between the predicted return and actual return is quite high. The existence of the non-linearity of financial market is propounded by many researchers (Abhyankar et al., 1997). The non-linear models such as Autoregressive Conditional Heteroskedasticity (ARCH) and General Autoregressive Conditional Heteroskedasticity (GARCH) are most widely used in modeling financial time series that exhibit time-varying volatility clustering, i.e. periods of swings followed by periods of relative calm.

**Introduction**

In last few years there has been advancement in application of 'Artificial Neural Network' (ANN) in forecasting market indices. A neural network is a massively parallel distributed processor made up of simple processor unit which has a natural propensity for storing experiential knowledge and making it available for use. ANN is a non-linear, statistical computational model, having adaptive system that changes its structure based on external or internal information that flows through the network during the processing phase. They are used to model complex relationships between inputs and outputs and find successive patterns.




Network connectivity in General

**Network layers**

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of **"output**" units.

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

Software implementation of a neural network can be made with their advantages and disadvantages.

**Advantages:**

- A neural network can perform tasks that a linear program cannot do.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.

**Disadvantages:**

- The neural network needs training to operate.
- The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
- Requires high processing time for large neural networks.

**Trained neural networks?**

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" information system to analyze and predict market behavior.

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self-organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel for use in real time operations

**Neural networks versus conventional computers**

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks.

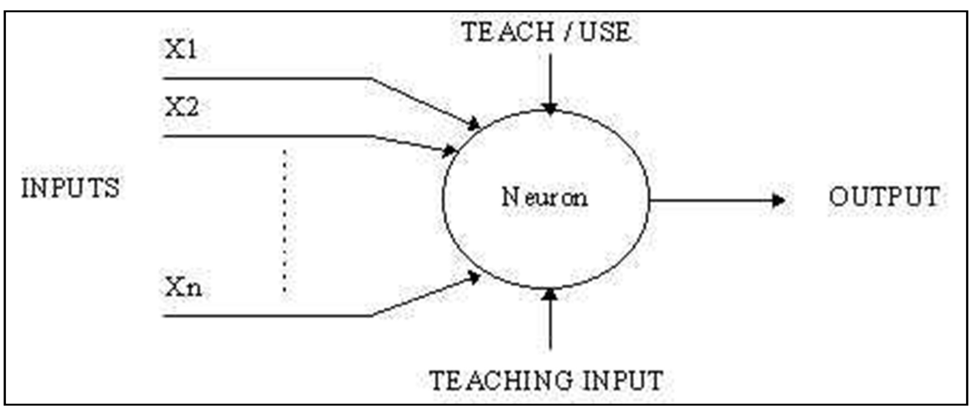
**Taxonomy of Neural Networks**

There are two phases in neural information processing. They are the learning phase and the retrieving phase. In the training phase, a training data set is used to determine the weight parameters that define the neural model. This trained neural model will be used later in the retrieving phase to process real test patterns and yield classification results.

- **Retrieving Phase**: Various nonlinear systems have been proposed for retrieving desired or stored patterns. The results can be either computed in one shot or updated iteratively based on the retrieving dynamics equations. The final neuron values represent the desired output to be retrieved.
- **Learning Phase**: A salient feature of neural networks is their learning ability. They learn by adaptively updating the synaptic weights that characterize the strength of the connections. The weights are updated according to the information extracted from new training patterns.

## A simple neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.




A simple neuron

**Models**

Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function $f : X \to Y$ or a distribution over $X$ or both $X$ and $Y$, but sometimes models also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a *class* of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).

**Network function**

The word network in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. The most basic system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" which are used to manipulate the data in the calculations.

The layers network through the mathematics of the system algorithms. The network function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is

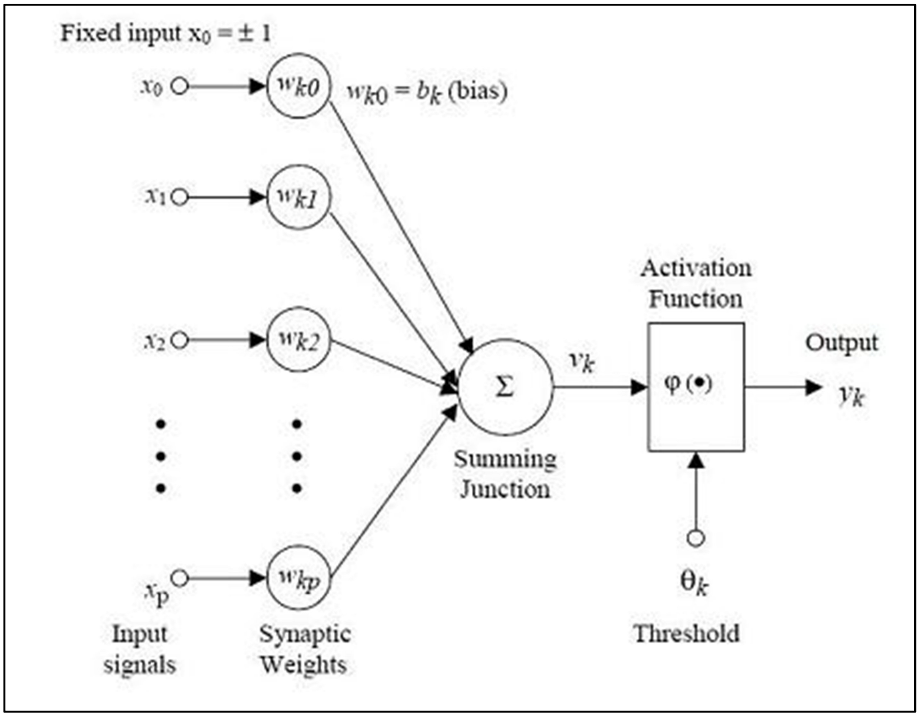
Nonlinear weighted sum, where $$f(x) = K\left(\sum_i w_i g_i(x)\right),$$

$K$ is some predefined function. It will be convenient for the following to refer to a collection of functions $g_i$ as simply a vector $g = (g_1, g_2, \dots, g_n)$.

**The Mathematical Model**

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections. The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination.

Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1. Mathematically, this process is described in the figure




The network shown above is a Multi-Layer Neural Network. The network architecture comprises of input layer with input neurons, two hidden layers with hidden neurons in each of the hidden layers and the output layer. The hidden layer of the neural network captures the data patterns and characteristics, and establishes a complex dynamic nonlinear relationship between the input and the output variables.

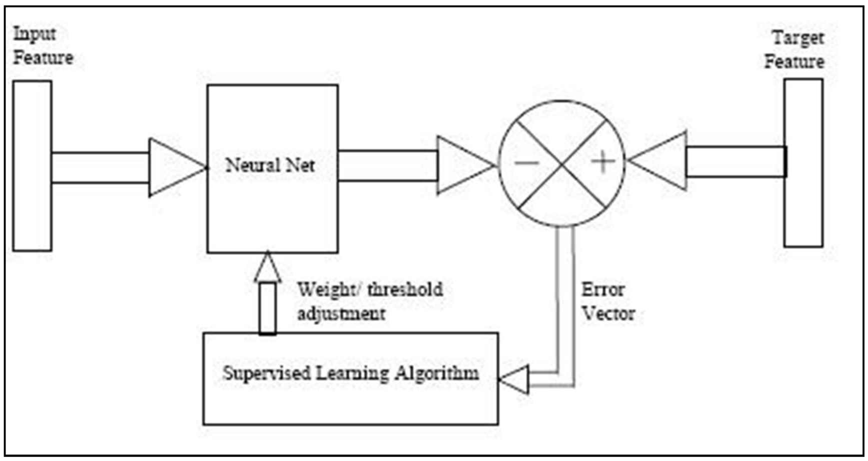
Neural networks are data driven models. The novelty of neural network lies in their ability to discover non- linear relationship in the input data set without a priori assumption of the knowledge of relation between the input and output. The greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism which ‘learns’ from observed data. With the correct implementation, ANNs can be used naturally in online learning and large data set applications.

**Training of artificial neural networks**

A **neural network** has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to **'train' the neural network** by feeding it teaching patterns and letting it change its weights according to some learning rule.

We can categorize the learning situations in two distinct sorts. These are:

- **Supervised learning** or Associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the neural network (self-supervised).




- **Unsupervised learning** or Self-organization in which an output unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.
- **Reinforcement Learning** This type of learning may be considered as an intermediate form of the above two types of learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self organizing neural learning may be categorized under this type of learning.

**The Learning Process**

The memorization of patterns and the subsequent response of the network can be categorized into two general paradigms:

**Associative mapping** in which the network learns to produce a particular pattern on the set of input units whenever another particular pattern is applied on the set of input units. The associative mapping can generally be broken down into two mechanisms:

**Auto-associative**: an input pattern is associated with itself and the states of input and output units coincide. This is used to provide pattern completion, i.e. to produce a pattern whenever a portion of it or a distorted pattern is presented. In the second case, the network actually stores pairs of patterns building an association between two sets of patterns.

**Hetero-association**: It is related to two recall mechanisms:

a. Nearest-neighbor recall, where the output pattern produced corresponds to the input pattern stored, which is closest to the pattern presented, and
b. Interpolative recall, where the output pattern is a similarity dependent interpolation of the patterns stored corresponding to the pattern presented. Yet another paradigm, which is a variant associative mapping is classification, i.e. when there is a fixed set of categories into which the input patterns are to be classified.

**Regularity detection:** in which units learn to respond to particular properties of the input patterns. Whereas in associative mapping the network stores the relationships among patterns, in regularity detection the response of each unit has a particular 'meaning'. This type of learning mechanism is essential for feature discovery and knowledge representation.

Every neural network possesses knowledge which is contained in the values of the connections weights. Modifying the knowledge stored in the network as a function of experience implies a learning rule for changing the values of the weights.

**Applications of neural networks**

Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- Sales forecasting
- Industrial process control
- Customer research
- Data validation
- Debt risk assessment
- Target marketing
- Business failure prediction
- Assess Bond and mortgage risk
- Predict bankruptcy
- Implement investment strategies

ANN are also used in the following specific applications: recognition of speakers in communications; recovery of telecommunications from faulty software; undersea mine detection; texture analysis; three-dimensional object recognition; hand-written word recognition; and facial recognition.

**Conclusion:**

The computing world has a lot of scope for deploying neural networks in variety of applications. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response which is due to their parallel architecture.

Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.

## References

Abhyankar, A., Copeland, L. S., & Wong, W. (1997). Uncovering nonlinear structure in real-time stock market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. Journal of Business &Economic Statistics, 15, 1–14.

Desai, V. S., &Bharati, R., (1998). A comparison of linear regression and neural network methods forpredicting excess returns on large stocks. Annals of Operations Research, 78, 127–163.

Engle, RF, (1982), Auto regression Conditional Heteroscedasticity Estimates of the variance of UK Inflation, Econometrica, 50, 987-1008.

Fama, Eugene (1970), “Efficient Capital Markets: A Review of Theory and Empirical Work,” Journal of Finance.

Fernandez-Rodriguez, F., Gonzalez-Martel, C., & Sosvilla-Rivero, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. Economic Letters, 69, 89–94.

Huang Wei., Wang Shouyang., Yu Lean., Bao1 Yukun., and Wang Lin., (2006) A New Computational Method of Input Selection for Stock Market Forecasting with Neural Networks, Lecture Notes In Computer Science.

Jung-Hua Wang, and Jia-Yann Leu. Stock market trend prediction using ARIMA-based neural networks, 1996., IEEE International Conference on Volume 4, Issue , 3-6 Jun 1996 Page(s):2160 –2165 vol.4.

Kim, Kyoung-Jae. Artificial neural networks with feature transformation based on domain knowledge for the prediction of stock index futures Intelligent Systems in Accounting, Finance & Management, Vol. 12, Issue 3, Pages 167 – 176.

Leigh, W.Paz, M., & Purvis, R. (2002). An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index. Omega-International Journal of Management Science, 30, 69–76.

Pan H. P. (2003), A joint review of Technical and Quantitative Analysis of Financial Markets Towards a Unified Science of Intelligent Finance, Paper for the 2003 Hawaii International Conference on Statistics and Related Fields.