

CHAPTER 2

LITERATURE REVIEW

2.1 Artificial Neural Network

Artificial neural network (ANN), usually called Neural Network (NN), is an algorithm that was originally motivated by the goal of having machines that can mimic the brain. A neural network consists of an interconnected group of artificial neurons. They are physical cellular systems capable of obtaining, storing information, and using experiential knowledge. Like human brain, the ANN's knowledge comes from examples that they encounter. In human neural system, learning process includes the modifications to the synaptic connections between the neurons. In a similar way, ANNs adjust their structure based on output and input information that flows through the network during the learning phase.

Data processing procedure in any typical neural network has two major steps: the learning and application step. At the first step, a training database or historical price data is needed to train the networks. This dataset includes an input vector and a known output vector. Each one of the inputs and outputs are representing a node or neuron. In addition, there are one or more hidden layers. The objective of the learning phase is to adjust the weights of the connections between different layers or nodes. After setting up the learning samples, in an iterative approach a sample will be fed into the network and

the resulting outputs will be compared with the known outputs. If the result and the unknown output are not equal, changing the weights of the connections will be continued until the difference is minimized. After acquiring the desired convergence for the networks in the learning process, the validation dataset is applied to the network for the validating step (Shahkarami A. et al. 2014).

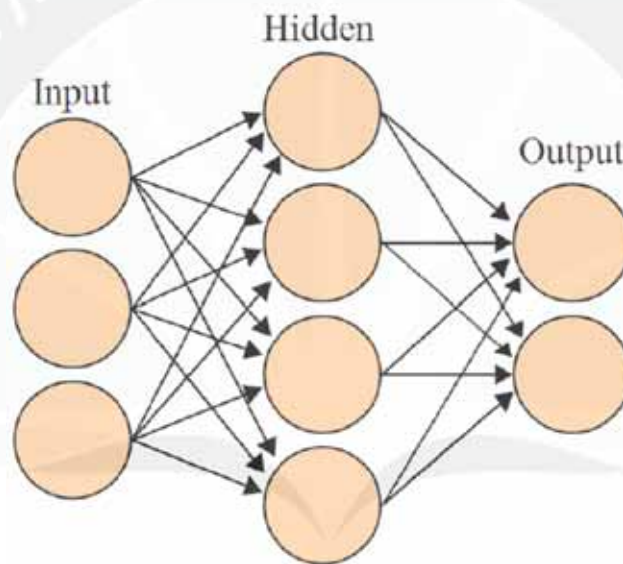


Fig. 1 An artificial neural network is an interconnected group of nodes.

Source : SPE International, Colorado, USA, 16–18April 2014.

2.2 Review of previous researches

Several economists advocate the application of neural networks to different fields in financial markets and economic growth methods of analysis (Kuan, C.M. and White, H. 1994). We focus the review of prior studies on prediction of financial market. Chen et al (2003) attempted to predict the trend of return

on the Taiwan Stock Exchange index. The probabilistic neural network (PNN) is used to forecast the trend of index return. Statistical performance of the PNN forecasts is compared with that of the generalized methods of moments (GMM) with Kalman filter and random walk. Empirical results showed that PNN demonstrate a stronger predictive power than the GMM–Kalman filter and the random walk prediction models. Kim (2003) used SVM to predict the direction of daily stock price change in the Korea composite stock price index (KOSPI). This study selected 12 technical indicators to create the initial attributes. The indicators are stochastic K%, stochastic D%, Slow %D, momentum, ROC, Williams' %R, A/D oscillator, disparity 5, disparity 10, OSCP, CCI and RSI. In addition, this study examined the feasibility of applying SVM in financial prediction by comparing it with back-propagation neural network (BPN) and case-based reasoning (CBR). Experimental results proved that SVM outperform BPN and CBR and provide a promising alternative for stock market prediction. Altay & Satman (2005) compared the forecasting performance artificial neural network and linear regression strategies in Istanbul Stock Exchange and got some evidence of statistical and financial outperform of ANN models. Kumar & Thenmozhi (2006) investigated the usefulness of ARIMA, ANN, SVM, and random forest regression models in predicting and trading the S&P CNX NIFTY Index return. The performance of the three nonlinear models and the linear model are measured statistically and financially via a trading experiment. The

empirical result suggested that the SVM model is able to outperform other models used in their study.

Hyup Roh (2007) introduces hybrid models with neural networks and time series model for forecasting the volatility of stock price index in two vision points: deviation and direction and the results showed that ANN-time series models can increase the predictive power for the perspective of deviation and direction accuracy. His research experimental results showed that the proposed hybrid NN-EGARCH model could be improved in forecasting volatilities of stock price index time series.

Adebiyi Ayodele A. et al. (2009) presented a hybridized approach which combines the use of the variables of technical and fundamental analysis of stock market indicators for daily stock price prediction. The study used three-layer (one hidden layer) multilayer perceptron models (a feedforward neural network model) trained with backpropagation algorithm. The best outputs of the two approaches (hybridized and technical analysis) are compared. Empirical results showed that the accuracy level of the hybridized approach is better than the technical analysis approach. Liao & Wang (2010) applied a Stochastic Time Effective Neural Networks in predicting China global index and their study results showed that the mentioned model outperform the regression model. Kara et al (2011) compared neural networks performance and SVM in predicting the movement of stock price index in Istanbul Stock Exchange. The input variables in suggested models include technical indicators such as CCI, MACD, LW R%, etc. The results revealed that neural

networks work better in prediction than SVM technique. Zhou Wang et al (2011), propose a new model to predict the Shanghai stock price. They used Wavelet De-noising-based Back propagation (WDBP) neural network. For demonstrating superiority new model in predicting, the results of it is compared with Back Propagation neural network and the total results showed that the WDBP model for forecasting index is better than BP model.

Putra and Kosala (2011) try to predict intraday trading Signals at IDX they used technical indicators - the Price Channel Indicator, the Adaptive Moving Averages, the Relative Strength Index, the Stochastic Oscillator, the Moving Average Convergence-Divergence, the Moving Averages Crossovers and the Commodity Channel Index. The result of their experiments showed that the model performs better than the naïve strategy. Also Veri and Baba (2013) forecasting the next closing price at IDX, they used opening price, highest price, lowest price, closing price and volume of shares sold as experimental variables. The result showed that the most appropriate network architecture is 5-2-1 with dividing the data into two parts, with 40 training data with 95% accuracy of data and 20 test data with 85% accuracy of data.

2.3 Learning Paradigms in ANNs

The ability to learn is a peculiar feature pertaining to intelligent systems, biological or otherwise. In artificial systems, learning (or training) is viewed as the process of updating the internal representation of the system in response to external stimuli so that it can perform a specific task. This

includes modifying the network architecture, which involves adjusting the weights of the links, pruning or creating some connection links, and/ or changing the firing rules of the individual neurons.

ANN approach learning has demonstrated their capability in financial modelling and prediction as the network is presented with training examples, similar to the way we learn from experience. In this paper, a three-layered feed-forward ANN model was structured to predict stock price index movement is given in Fig. 2. This ANN model consists of an input layer, a hidden layer and an output layer, each of which is connected to the other. At least one neuron would be employed in each layer of the ANN model. Inputs for the network were twelve technical indicators which were represented by twelve neurons in the input layer. Each neuron (unit) in the network is able to receive input signals, to process them and to send an output signal. Each neuron is connected at least with one neuron, and each connection is evaluated by a real number, called the weight coefficient, that reflects the degree of importance of the given connection in the neural network (Daniel et al. 1997).

The error between the predicted output value and the actual value is back-propagated through the network for the updating of the weights. This method is proven highly successful in training of multi-layered neural networks. The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and

is used to adjust the connections between the layers, thus improving performance.

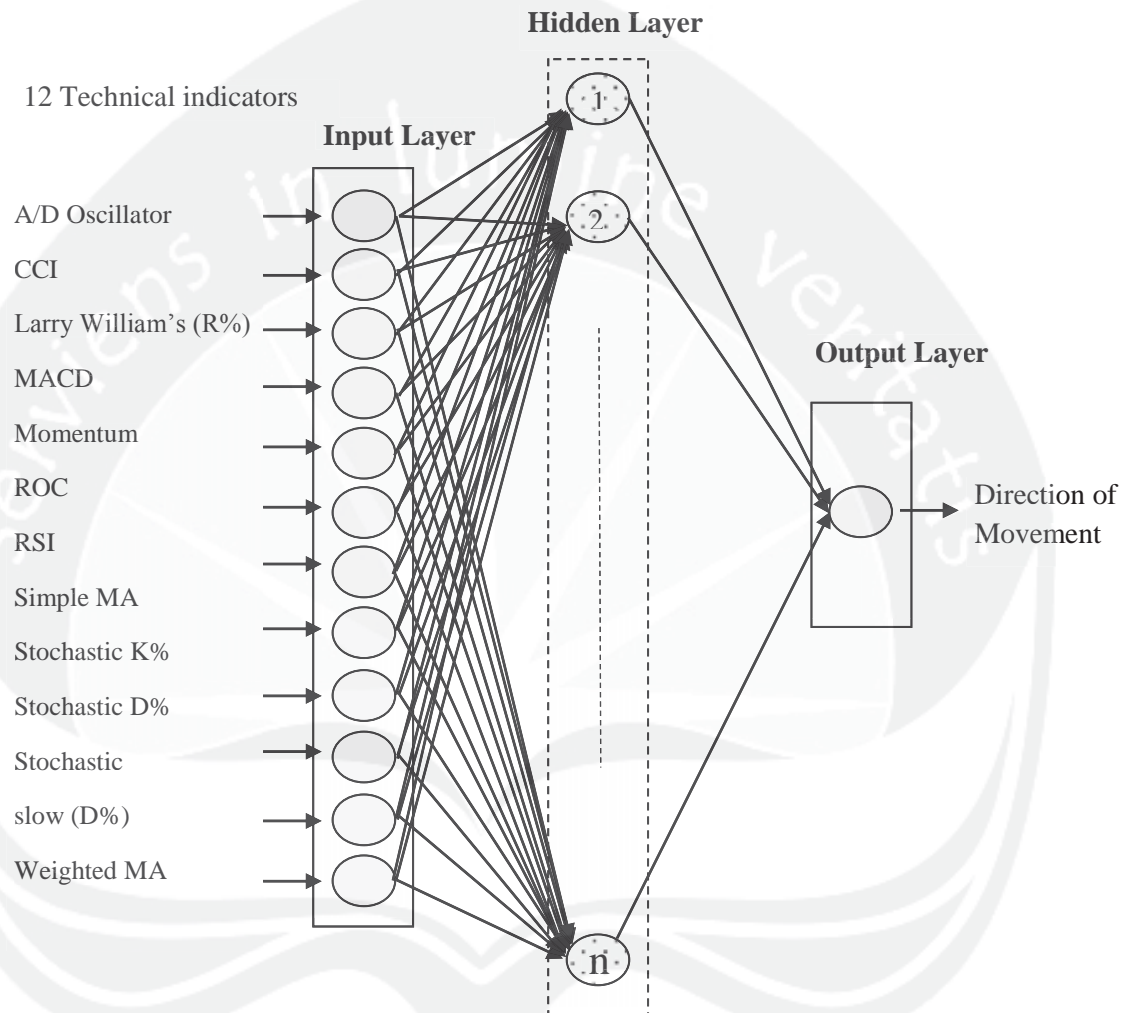


Fig. 2. A Neural network with three-layer feed forward

Source: Y. Kara et al. / Expert Systems with Applications 38 (2011)

5311–5319

This a supervised learning procedure that attempts to minimise the error between the desired and the predicted outputs. If the error of the validation

patterns increases, the network tends to be over adapted and the training should be stopped.

The most typical activation function used in neural networks is the logistic sigmoid transfer function. This function converts an input value to an output ranging from 0 to 1. The effect of the threshold weights is to shift the curve right or left, thereby making the output value higher or lower, depending on the sign of the threshold weight. The output values of the units are modulated by the connection weights, either magnified if the connection weight is positive and greater than 1.0, or being diminished if the connection weight is between 0.0 and 1.0. If the connection weight is negative or (value < 0) then tomorrow close price value $<$ than today's price (loss). If (value > 0.5) then then tomorrow close price value $>$ than today's price (profit). As shown in Fig. 2, the data flows from the input layer through zero, one, or more succeeding hidden layers and then to the output layer. The back-propagation (BP) algorithm is a generalisation of the delta rule that works for networks with hidden layers. It is by far the most popular and most widely used learning algorithm by ANN researchers. Its popularity is due to its simplicity in design and implementation. The idea is to train a network by propagating the output errors backward through the layers. The errors serve to evaluate the derivatives of the error function with respect to the weights, which can then be adjusted. It involves a two stage learning process using two passes: a forward pass and a backward pass. The basic back propagation algorithm consists of three steps (Fig. 2). Although, the most

commercial back propagation tools provide the most impact on the neural network training time and performance. The output value for a unit is given by the following Equation:

$$\begin{aligned}
 \mathbf{y} &= f(h_j) \\
 &= \left(\sum_{i=1}^n \mathbf{w}_{ij} \mathbf{x}_i, \theta_j \right) \\
 &= \begin{cases} \mathbf{1} & \mathbf{w}_{ij} \mathbf{x}_i \geq \theta \\ \mathbf{0} & \mathbf{w}_{ij} \mathbf{x}_i < \theta \end{cases} \quad (i = 1, 2, \dots, n) \quad (1)
 \end{aligned}$$

Where \mathbf{y} the output value is computed from set of input patterns, \mathbf{X}_i of i^{th} unit in a previous layer, \mathbf{W}_{ij} is the weight on the connection from the neuron i^{th} to j , θ_j is the threshold value of the threshold function \mathbf{f} , and n is the number of units in the previous layer. The function $f(x)$ is a sigmoid hyperbolic tangent function (Barndorff-Nielsen et al. 1993)

$$\mathbf{f}(x) = \tanh(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad \text{Threshold: } \mathbf{f}(x) = \begin{cases} 0 & x < 0,1 \\ 1 & \text{if otherwise} \end{cases} \quad (2)$$

where $f(x)$ is the threshold function remains the most commonly applied in ANN models due to the activation function for time series prediction in back-propagation (Najeb Masoud, 2014):

$$\mathbf{y} = \mathbf{f} \left(\sum_{i=1}^n \mathbf{w}_i \mathbf{x}_i - u \right) = \left(\sum_{i=1}^n \mathbf{w}_i \mathbf{x}_i \right) \quad (3)$$

Once the output has been calculated, it can be passed to another neuron (or group of neurons) or sampled by the external environment. In terms of the weight change, $\Delta \mathbf{w}_{ij}$, the formula equation is given as:

$$\Delta \mathbf{w}_{ij} = \eta \delta_j \mathbf{x}_i \quad (4)$$

where η is the learning rate ($0 < \eta < 1$), δ_j is the error at neuron j , \mathbf{x}_i is an input vector and \mathbf{w}_i the weights vector. This rule of IDX can also be rewritten as:

$$\Delta \mathbf{w}_i = -\eta (\mathbf{t}_i - \mathbf{x}_i \mathbf{w}_i) \mathbf{x}_i \quad (5)$$

Although a high learning rate, η , will speed up training (because of the large step) by changing the weight vector, \mathbf{w} , significantly from one phase to another. According to Wythoff BJ. (1993) suggests that $\eta \in [0.1, 1.0]$.