CHAPTER - III
CHAPTER - III
TRANSFORM TECHNIQUES

A. INTRODUCTION

In this chapter, the proposed simple and efficient techniques are described and analyzed. Wavelets are group of functions that divide the data into different frequency components. Wavelet transforms have many real world applications, including the compression of finger print images, analysis time series data.

Neural networks have been shown to learn complex relationships. It would be interesting to see if the networks can be trained to learn the non linear relationship underlying Black-Scholes type models [70]. The use of neural networks in finance is a recent phenomenon. They have been used to predict the economy, pick stocks, construct portfolios, spot insider dealings, and assess bond risks. Recently, a number of studies have looked at the ability of networks to learn Black-Scholes type models.

In daily life investors typically face inaccurate data since in many cases it is not possible or feasible to observe and to measure a phenomenon avoiding an arbitrary degree of accuracy. This impression leads to difficulties in managing and constructing models, especially in the case of complex problems. A fuzzy approach is a method able to simplify complexity by taking into account a reasonable amount of imprecision, vagueness and uncertainty.

B. DISCRETE WAVELET TRANSFROM & NEURAL NETWORKS

3.1. Finance Background, Futures and Options

A future contract is an obligation to either buy or sell a specific commodity known as the underlying – at an agreed price at some time in the future. Futures have an expiry time. At expiry the holder of the future must either buy or sell the underlying at the price specified in the futures contract. The futures market is a no net gain market. For each investment there will always be an equal and opposite
investment. This implies that for every dollar that one investor makes, another investor, who took the opposite trade makes an equal and opposite loss.

Stock market indices are designed to reflect over all price movements in a large number of equity shares (securities). The performance of an equity index is important because it represents the performance of a broadly diversified stock portfolio and gives insights into the broad market risk/return profile.

Index futures are contracts that commit the user to either buy (go long) or sell (go short) the stocks in the index at the currently determined market price at some point in the future. If the investor believes the index to be going down he should sell/go short. If, on the other hand, the investor believes that the index is going up he should buy/go long.

Whereas futures are the obligation to buy or sell the underlying at a particular price in the future, options are the right (not the obligation) to buy or sell the underlying at a particular price in the future. There are two types of options analogous to long/short futures. The right (not the obligation) to buy is a call. The right (not the obligation) to sell is a put. An investor who uses his right to buy or sell the underlying is said to have exercised the option and takes delivery of the underlying at the price specified in the option contract – the exercise or strike price. For a call option, if the underlying is greater than the strike price the option is said to be in-the-money. If the underlying price is less than the strike price the option is said to be out-of-the-money. If the underlying price is similar to the strike price the option is said to be at-the-money. An option, like a future, has a lifetime. The option has a price or value known as a premium. After an option expires its premium is worthless and the option cannot be exercised. An option on a future is the right but not the obligation to purchase the future at the strike price before the expiry.

One final distinction is between American and European style options. American style options can be exercised at any time prior to expiration, whilst European style options can only be exercised at expiry.
3.2. Discrete Wavelets in Option Pricing

As an improvement over what has been so far researched, the Wavelets are now being used for analyzing changes in financial marketing especially in option pricing. Wavelets by definition are small groups of function that are formed by dividing the data into different frequency distribution. Wavelet analysis provides an important tool for extracting information from financial market data with applications ranging from short term prediction to the testing of models and the calculation of variance in relation to specific time series [80].

3.3. The Hybrid-Artificial Neural Network Approach

Motivated by the good initial fit to the data provided by the modified Black model, we use a hybrid approach in which non-parametric regression techniques model the residuals between the option transaction prices and the modified Black model prices [46].

The fundamental advantage of non-parametric regression is that it makes very few assumptions about the unknown function to be estimated. Lajbcygier and Flitman [47] have compared artificial neural networks (ANN's) with a method from each of the general classes of non-parametric regression methods: global parametric methods (i.e. linear regression), local parametric methods (i.e. kernel regression) and adaptive computation methods (i.e. projection pursuit regression). ANN's were among the most accurate regression techniques compared.

The relationship between the option input variables (i.e. F / X, T - t^m, \sigma_{Dened}^n) and the residuals shows that there are persistent and systematic (weakly) nonlinear biases. Furthermore, weak interactions between the
input variables are shown to exist. ANN’s are eminently suitable for modeling such functions.

The hybrid model can be depicted mathematically as follows:

\[ f_{\text{hybrid}}(x) = f_{MB}(x) - f_{NN}(x) \quad \text{...3.1} \]

Hybrid neural networks of the form in equation 3.1 were shown to outperform hybrid linear models, for a similar data set, by a factor of two in Lajbcygier and Flitman [47].

Intraday call option transactions were considered from January 1993 to December 1993. The first half of the data, January through June, was used as an estimation set and the rest was reserved for out of sample testing.

A three layer, fifteen hidden unit neural network was estimated using back propagation with a 20% cross validation set used for network selection. What follows is an analysis of the ANN-hybrid output. The estimated hybrid option pricing model is shown in figure 3.2–3.4. They plot the output of the Modified Black hybrid ANN as a function of \( F/X \) and \( T-t \). Figure 3.2 is the ANN output surface for the standard deviation equal to 0.11 – the lowest standard deviation for the in-sample data, while figure 3.4 is the same surface for standard deviation equal to 0.28 – the highest standard deviation in the in-sample set.

In general, the surfaces are complicated, smooth and imply consistent mis-pricing in the conventional models of between 2 and -2 points (approximately $50 and -$50 per option respectively, if we assume a strike of \( X = 2000 \).
The most striking feature of the hybrid ANN output at all standard deviations is the ridge at $F/X \approx 1$. This divides the options into those that have positive and negative value relative to the conventional option-pricing model (see Table 3.1).

This is consistent with both the Rubinstein [69] and Derman and Kani [19] studies of the S & P 500 CBOE futures options. Rubinstein, [69] conjectures that this bias is caused by investors' fear of a repeat of the 1987 crash. The shape of the hybrid surface is almost identical to the deviations noted by Corrado and Miller [16] for the S&P 500. This is quite remarkable given the different markets.

Table 3.1 Positive (+) value above conventional model, (-) value below conventional model.

<table>
<thead>
<tr>
<th></th>
<th>Short Maturity (0–0.15)</th>
<th>Long Maturity (0.15 – 0.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the Money (0.9 –1)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Out of the Money (1 – 1.1)</td>
<td>+ / –</td>
<td>–</td>
</tr>
</tbody>
</table>
Figure 3.2: Hybrid ANN output, standard deviation $\sigma = 0.11$

Figure 3.3: Hybrid ANN output, standard deviation $\sigma = 0.2$

Figure 3.4: Hybrid ANN output, standard deviation $\sigma = 0.28$
For low standard deviation (see figure 3.2), out of the money short time to maturity options are valued more highly than the conventional model. This low standard deviation effect, has not been emphasized in prior studies.

It is interesting to compare and contrast figure 3.2 and figure 3.4 to ascertain the interaction between the surface variables: \( \frac{F}{X} \), \( T - t \) and \( \sigma \). No large differences in the surfaces exist, but there are five subtle changes. Firstly, the top flat region in figure 3.2 has extended and moved forward. Secondly, the top flat region in figure 3.2 has shifted up. Thirdly, the bottom of the surface near \( T - t \) equal to zero has moved up. Fourthly, the region between the flat top and the steep wall on the right of the surface is smoother. Finally, the dip at \( \frac{F}{X} \) equal to 1.02 and \( T - t \) equal to 0.07 in figure 3.2 has become shallower in figure 3.4.

3.4. Trading Strategies Based on Bootstrap Confidence Intervals

The large majority of the research in option-pricing involves finding a model that fits the empirical data. Very little research has been done on generating the confidence intervals of the option-pricing model. The confidence intervals will allow both choosing between option pricing models and deciding when a trade should be executed.

Due to neural networks nonlinearity and structural complexity, classical statistical theory provides little help in estimating confidence limits. Chryssolouris [15] requires unrealistic and strong assumptions (normal errors) to estimate confidence limits for neural networks. In this work, confidence intervals for option pricing models are generated by bootstrap methods.
Figure 3.5: Out-of-the-money option price premiums versus $F/X$: The Modified Black pricing model falls within the confidence intervals of the "hybrid" model. The option pricing parameters are $T-t = 0.1$, $\sigma = 0.15$, $X = 2000$.

Figure 3.6: At-the-money option price premium versus $F/X$: In some regions such as $F/X = 0.98$ the two models are distinguishable. The option pricing parameters are $T-t = 0.1$, $\sigma = 0.15$, $X = 2000$. 
Given a total of $n$ options in the data-set, $i$ bootstrap data sets are generated. Bootstrap data sets $L_i = \{(c_j^{(i)}, x_j), j = 1, ..., n\}$ are generated by

$$c_j^{(i)} = \hat{f}_{\text{hybrid}}(x) + e_j^{(i)}$$

where $e_j^{(i)}$ are drawn randomly with replacement from the empirical distribution $p(e) = n^{-1} \sum \delta (e - \hat{e}_i)$, and $\hat{e}_i$ are the observed residuals from initial hybrid model fit. This is known as a “bootstrap residual approach” Tibshirani [76]. Predictors $\hat{f}^{(i)}(x)$ are estimated on the bootstrap data sets, $L_i$ in the same manner as the hybrid predictor. The bootstrap assumption for confidence intervals is

$$(f(x) - \hat{f}_{\text{hybrid}}(x))^2 \approx \frac{1}{N-1} \sum_{i=1}^{N} (\hat{f}_{\text{hybrid}}(x) - \hat{f}^{(i)}(x))^2$$...

where $f(x)$ is the true function, and $N$ is the number of bootstrap data series simulated, in this case 30. Tibshirani [76] used $N = 20$, he argues this is a lower limit on the number of bootstrap replications but necessary due to the complicated ANN model. In figure 3.5 confidence intervals computed in equation 3.2 and centered at the hybrid predictor are shown for +/- one standard deviation. Bootstrap and bagging predictors are also plotted in figure 3.5. The width of the confidence intervals varies over the input space. In the region of at-the-money options, figure 3.6 confidence intervals are much tighter than for deep out of the money options. The modified Black predictor often falls outside the confidence intervals. In these regions, confidence can be placed in the hybrid predictors.

Bootstrap confidence intervals allow the identification of option prices, which both appear profitable and are outside the range of model uncertainty. Since the confidence intervals vary over the input space of the model trading positions will be confined to areas of greater certainty.
Identification of profitable trades is not the only use for better option pricing models. The process of limiting exposure of a financial position to changes in underlying assets is known as hedging and is determined by the option pricing model. Hedges are incorporated into the option trading strategy by buying a position in the underlying futures equal to $-\frac{\partial f_{\text{hybrid}}(x)}{\partial F}$, known as the “delta”, of the option position which allows a small change in the option price to be offset by a change in the future price.
Typical delta surfaces are shown in figures 3.7–3.10 for both the hybrid neural network and the modified Black model. The two models yield a slightly different delta, which implies that different hedging strategies will be employed. The hybrid delta surfaces are not nearly as smooth as the modified Black delta surfaces. Wrinkles in the delta surface are especially evident for delta approximately equal to half. Furthermore, there exists for $T - t$ very small and $F/X$ close to one a negative delta value. This is unrealistic. This is one drawback of using a neural network derived delta. It is negative whereas the Modified Block model always has a positive delta. It does not seem likely that an ideal model would have a negative delta, so this appears to be an artefact due to a limited amount of training data.

In the trading strategies employed below, a hedge in the futures position is incorporated with each option position. This allows the profitability of the strategy to be stressed, instead of the variability of the underlying. The point is that a better hedge will lead to less volatile results.

In table 3.2, the profitability of various trading strategies is shown. All trading strategies are based on taking a position on options that are one point beyond the confidence limits of the hybrid model and simultaneously employing a one-time hedge. One index point is a reasonable approximation for the costs associated with crossing the bid-ask spread and exchange costs associated with undertaking the option transaction Gilmore [32]. Since all the options expire on the same date, only a single equity for each model is quoted. The confidence intervals performed as hoped. As one begins to trade outside the region of uncertainty, dramatic improvements in the Sharpe ratio begin and stay. The Sharpe ratio is the standard measure of trading performance, it is the (equity per trade / standard deviation) of returns and is a useful metric because it penalizes risky strategies. This is why a trading strategy based on the Hybrid + 3 sigma is comparable to the Hybrid + 2 sigma which make more equity per trade.
Table 3.2 Trading profitability of the Hybrid and Modified Black based strategies. Incorporating confidence intervals allows the Hybrid model performance to increase by nearly a factor of 10. Note the poor performance of all Modified Black based strategies.

<table>
<thead>
<tr>
<th></th>
<th># of Trades</th>
<th>Equity / # of Trades</th>
<th>Var.</th>
<th>Sharpe Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid</td>
<td>449</td>
<td>17.69</td>
<td>382</td>
<td>0.0463</td>
</tr>
<tr>
<td>Hybrid + sigma</td>
<td>240</td>
<td>10.00</td>
<td>289</td>
<td>0.0346</td>
</tr>
<tr>
<td>Hybrid + 2 sigma</td>
<td>109</td>
<td>61.17</td>
<td>205</td>
<td>0.2984</td>
</tr>
<tr>
<td>Hybrid + 3 sigma</td>
<td>51</td>
<td>48.66</td>
<td>159</td>
<td>0.3060</td>
</tr>
<tr>
<td>Modified Black</td>
<td>478</td>
<td>-7.24</td>
<td>405</td>
<td>-0.0179</td>
</tr>
<tr>
<td>Modified Black + Sigma</td>
<td>275</td>
<td>-41.38</td>
<td>278</td>
<td>-0.1289</td>
</tr>
<tr>
<td>Modified Black + 2 Sigma</td>
<td>142</td>
<td>-3.45</td>
<td>142</td>
<td>-0.0147</td>
</tr>
<tr>
<td>Modified Black + 3 Sigma</td>
<td>68</td>
<td>-75.63</td>
<td>68</td>
<td>-0.4170</td>
</tr>
</tbody>
</table>

There is an eight fold improvement in Sharpe ratio performance between the Hybrid + sigma and Hybrid + 2 sigma bands. The two sigma standard error bands capture most of the trading opportunities associated with the hybrid model, which explains why the Sharpe ratio performance does not improve dramatically for Hybrid + 3 sigma. Note the failure of the standard Modified Black strategy.

3.5. Estimation of Bias Using Bootstrap Techniques

The large majority of option pricing research involves finding a model that fits the data. Very little research has been done on generating bias estimates for new models. The bias of an estimator $\theta$ is the difference between the expected value of the estimator and the true value of the parameter $\text{Bias}(\theta) = \theta - E(\theta)$. If $\theta$ is the hybrid ANN the Bias (Hybrid ANN) = $f_{\text{hybrid}} - f_{\text{bag}}$. The bias estimate is useful because it can show the regions of input space in which the bias becomes serious. In these regions, the estimator is poor and an alternative estimator may be considered.
The bootstrap estimate of bias for a hybrid neural network as a function of $F/X$, time to maturity, and a low implied volatility is shown in Figure 3.11. For this particular implied volatility, the neural network is showing a large bias for in-the-money options that are near maturity. In this region, there is a relatively sparse amount of training data because options are typically written out-of-the-money. This region is also at the edge of the training set because of its nearness to expiry and is showing some of the poor generalization that often occurs at the edge of the data with nonparametric regression techniques. This observation motivates further work which shall utilize the inherent option pricing model boundary conditions and utilize a novel ANN architecture so to constrain the ANN at the option pricing boundary conditions Lajbcygier [48].

C. NEURAL NETWORK TECHNIQUES

The basic concept in a neural network is the neuron. A neuron receives inputs from each of a set of other units provides inputs $x = (x_1, x_2, ..., x_j)$ and output $y = \phi\left(\sum_{i=1}^{j} a_i x_i + c\right)$. The mapping $\phi(.)$ is called the activation mapping and $a_i$ is called connection weights and $c$ is called bias. Here a very simple single hidden layer neural network which consists of $k$ hidden neurons and 1 output neuron is

Figure 3.11: Estimate of the bias using bootstrap techniques at standard deviation $\sigma = 0.11$. 
used. For each of the $k$ hidden neurons, the activation function (logistic squasher) $G(u) = \frac{\exp(u)}{1 + \exp(u)}$ is used, and the input will be the one dimensional factor $x$.

On top of this hidden layer, we have an output neuron which has the activation mapping as the identity mapping, and the output of the $k$ hidden neurons as its input. In all, as illustrated in figure 3.12, we use a neural network mapping as follows,

$$f_N(x) = \sum_{j=1}^{k} a_j G(b_j x + c_j); \quad \text{where} \quad G(u) = \frac{\exp(u)}{1 + \exp(u)}.$$  

...3.3

![Neural Network Diagram]

**Figure 3.12: The Neural Networks**

### 3.6. Neural Network Latent Factor Filter

Denote the neural networks approximation as $f_N(\theta, x_i)$ where $\theta$ denotes the vector of parameters. If we know the sample path of the underlying factor $x$, an easy way to estimate the value of $\theta$ is to do a nonlinear regression minimizing the
distance between the true values and the approximated values as in Gallant and White [29]. It is unfortunate in this case that the sample path of the underlying factor is latent and has to be inferred from the price changes. And we impose two equations to make the whole estimation doable,

\[ p_t = f_N(\theta, x_t) + \eta_t \quad \text{with} \quad \eta_t \sim N(0, \sigma^2_\eta), \quad \ldots \tag{3.4} \]

\[ \Delta p_t = f_{Nx}(\theta, x_t) \varepsilon_t + \frac{1}{2} f_{Nxx}(\theta, x_t) \sigma^2_\varepsilon \quad \text{where} \quad x_t = \sum_{i=1}^{t-1} \varepsilon_i, \quad \ldots \tag{3.5} \]

where \( f_{Nx} \) and \( f_{Nxx} \) are the first and second derivatives of function \( f \) respectively. We could easily see what the first equation tries to do is to match the level of the price with the artificial neural network model. And instead of minimizing the \( L^2 \) distance as in Gallant and White [29], we impose a probability structure on the error as a normal random variable with \( \sigma^2_\eta \). The second equation can be seen as a discrete version of the Ito's Lemma and plays the role of stochastic filter for the sample path of \( x \). As we have discussed already, \( x \) is assumed to be Brownian motion. Its increments \( \varepsilon_t \) has a law as \( N(0, \sigma^2_\varepsilon) \), where \( \sigma^2_\varepsilon \) could be thought as the time slipped by across consecutive observations according to the clock of \( x \). We call the above model as the Neural Network Latent Factor Filter (NNLFF).

3.7. Training Rules

Given a neural network model, we need to specify the learning rule, a recurrence algorithm in which the weights are modified as the data are processed. The whole process of learning is also called the training of neural networks, or in other words, the nonlinear estimation of the weights.

The maximum likelihood principle is used to estimate the NNLFF. It specifies a training rule as follows,

\[ \hat{\theta} = \arg\max_\theta L(\theta; \{p_t\}_{t=1}^T) \]
NNLFF gives us both the functional form of the $f$ and also gives us the estimated fundamental factor, both of them are major concerns of empirical finance and could be quite meaningful. We can certainly extend the above procedure even further by using multi-dimensional neural nets to track down the major economic factors driving the stock market.

3.8. The Training of the Neural Network

The data set we use is the S&P 500 daily series, from June 1\textsuperscript{st}, 1982 to June 1\textsuperscript{st}, 1987, a total of 1265 observations. Although longer time series are available, we use time series of this length. Because we believe any longer term times series may not offer more in terms of capturing the risk involved in the near future, rather it is very likely that the risk will be exaggerated by long term regime switching which are not relevant since here we are only concerned about the short term option. We also want to note that the price on June 1\textsuperscript{st}, 1987 is 289.83 and we are calculating the option price at a time of pre-crash period.

Out of many possibilities Liu [50], we end up picking three units logistic squasher neural networks. The parameters of this neural networks function will be \{ $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$ \}, where $b_1$ is normalized to be $-1$ and $x$ is normalized as starting from point 0. The two other parameters we use are $\sigma^2_n, \sigma^2_\varepsilon$, the measurement error of the stock price and the internal clock of the Brownian motion. We will use the simplex method to do the maximization which is implemented as in NMSIMP of GQOPT.
Table 3.3 The estimated Neural Networks

<table>
<thead>
<tr>
<th>Model</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$b_2$</th>
<th>$b_3$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$\sigma_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNLFF</td>
<td>5.1d-4</td>
<td>1.5d4</td>
<td>-6.3d3</td>
<td>-3.4d-4</td>
<td>8.9d-4</td>
<td>-2.6</td>
<td>-1.4</td>
<td>-2.4d-1</td>
<td>1.00</td>
</tr>
<tr>
<td>($k=3$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNLFF</td>
<td>-1.4d2</td>
<td>4.4d2</td>
<td>-4.8d-2</td>
<td></td>
<td>-1.8</td>
<td>-1.8</td>
<td></td>
<td></td>
<td>1.02</td>
</tr>
<tr>
<td>($k=2$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3 gives us the estimated values of the coefficients. We note the estimated $\sigma_\eta$ (not reported) is very small, which means the neural networks match the data quite well and very little of the variability has been resorted to the measurement error. It means also there is very little error to use the equation 3.5 as the filter to filter out the underlying Brownian motion. From the upper right panel
of figure 3.13, we may see that there is positive correlation between consecutive measurement errors.

Since we set our goal as tracking down both the underlying fundamental factor and the functional form of $f$, we should appraise our estimation along these two dimensions. In figure 3.13, we draw the picture of the fitted stock price and the estimated conditional volatility and the underlying factor with NNLFF. And as we can see, most of the dynamics of the price series is captured by the dynamics of the underlying factor and the conditional volatility is relatively constant. In figure 3.14, we show the neural net function with domain as $[-90, 10]$. As we can see, the function $f$ is quite linear except in the case of Neuron #1 which contains a jump, but since the magnitude of Neuron #1 is quite small, the jump has very little effect on the level of the price. The derivative of the function could be seen from figure 3.15. Notice the derivative function seems increasing and convex. Also interestingly, despite the level of Neuron #2 and Neuron #3 looks linear, it is convex and concave respectively, this could be seen clearly from their derivatives. The point where the conditional volatility begins to drop roughly corresponds to a point in time around February 10th, 1986. From the picture, NNLFF suggests that prior to February 10th, 1986, the conditional volatility $f_{nx}$ is quite constant, and after that date or so, we could see a sharp decrease in the conditional risk. It is worth commenting that this date corresponds well with the timing of the drop of oil price and the resolution of the oil crisis.
Figure 3.14: NNLFF $k = 3$ function

Figure 3.15: NNLFF $k = 3$ derivative
3.9. Option Pricing with NNLFF

The option values we compute are the prices of the S&P 500 index call options traded on CBOE (Chicago Board Options Exchange) on June 1st, 1987. We focus ourselves on the easiest among all, the one month options. We focus on the one month options since in this case the options are the closest to the European option and we do not need to worry about paying dividends. Another advantage of the near term options is that we do not need to take into account the variability of interest rate. These options will mature on July 18th, and we have a maturity of 33 days (working days). We use the three months treasury bill rate on June 1st as the constant discount rate, which is an annual rate of 4.83 per cent. In the calculation, a day is divided into 20 time intervals as an attempt to simulate the continuous sample path implied by the continuous diffusion process. So we have a total of $20 \times 33 = 660$ periods in our simulation. The expectation in the following equation

$$ w(t) = B(t)E_t \left( \frac{T}{T} \exp \left( \int_t^T \gamma dx - \frac{1}{2} \int_t^T \gamma^2 dt \right) h \right), $$

is computed in a Monte Carlo way as an average of 5000 samples.

In table 3.3, we show the forecasts of option prices (we do not report the table for put option). In the first row, we give out the striking price, the second row we list the market price of the corresponding option, the third row we show the Black-Scholes option price calculated in the standard way, the fourth row we have the forecasts of option prices with the NNLFF approach ($k = 3$). From Hull and White [40], when there is a positive correlation between conditional volatility and level, the Black-Scholes formula tends to over price those deep-in-the-money options and this is exactly what we see in the table.
Table 3.3 Comparison of forecasts of call option prices

<table>
<thead>
<tr>
<th>Striking</th>
<th>Market</th>
<th>B-S</th>
<th>NNLFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>215</td>
<td>75.53</td>
<td>76.14</td>
<td>74.45</td>
</tr>
<tr>
<td>220</td>
<td>70.25</td>
<td>71.17</td>
<td>69.52</td>
</tr>
<tr>
<td>225</td>
<td>65.12</td>
<td>66.20</td>
<td>64.51</td>
</tr>
<tr>
<td>260</td>
<td>30.75</td>
<td>31.50</td>
<td>29.97</td>
</tr>
<tr>
<td>270</td>
<td>20.50</td>
<td>21.95</td>
<td>20.07</td>
</tr>
<tr>
<td>275</td>
<td>15.25</td>
<td>17.49</td>
<td>15.47</td>
</tr>
<tr>
<td>280</td>
<td>11.50</td>
<td>13.38</td>
<td>11.16</td>
</tr>
<tr>
<td>285</td>
<td>7.75</td>
<td>9.78</td>
<td>7.70</td>
</tr>
<tr>
<td>290</td>
<td>4.75</td>
<td>6.81</td>
<td>4.77</td>
</tr>
<tr>
<td>295</td>
<td>2.87</td>
<td>4.51</td>
<td>2.77</td>
</tr>
<tr>
<td>300</td>
<td>1.44</td>
<td>2.82</td>
<td>1.40</td>
</tr>
<tr>
<td>305</td>
<td>0.75</td>
<td>1.67</td>
<td>0.69</td>
</tr>
<tr>
<td>310</td>
<td>0.31</td>
<td>0.93</td>
<td>0.24</td>
</tr>
<tr>
<td>315</td>
<td>0.06</td>
<td>0.51</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 3.4 Comparison of option pricing methods

<table>
<thead>
<tr>
<th>Norm</th>
<th>B-S&lt;sup&gt;c&lt;/sup&gt;</th>
<th>NNLFF&lt;sup&gt;c&lt;/sup&gt;</th>
<th>B-S&lt;sup&gt;p&lt;/sup&gt;</th>
<th>NNLFF&lt;sup&gt;p&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell^1$</td>
<td>18.63</td>
<td>4.54</td>
<td>3.36</td>
<td>2.90</td>
</tr>
<tr>
<td>$\ell^2$</td>
<td>29.09</td>
<td>3.05</td>
<td>1.29</td>
<td>1.50</td>
</tr>
<tr>
<td>$\ell^{\infty}$</td>
<td>2.24</td>
<td>1.08</td>
<td>0.70</td>
<td>0.88</td>
</tr>
</tbody>
</table>

In table 3.4 we use different standards to compare the different pricing methods. The standards we use include $\ell^1$, $\ell^2$ and $\ell^{\infty}$ norm difference between the forecasted value of option price and the market price for all options on the
market with different strike price. We evaluate the performance of our procedures in both the case of call option and put option, which is denoted with superscript $c$ and $p$ respectively. And as we could see from table 3.4, NNLFF certainly is the best. And for 14 different striking prices, the $\ell^2$ norm of the difference between the forecasted value and the market value is only a tenth of that of the Black Scholes.

3.10. Experimental Design

We use a back propagation neural network system to explore how well it learns the nonlinear form underlying the Black-Scholes type models. We generate two types of outputs. Firstly, we find how well the network learns to yield call option price given data on four of the five variables on which the call price depends in option pricing models: price of underlying asset, its volatility, time to maturity and risk free rate. Then we find the volatility of the underlying asset given call option price as one of the inputs along with the other four dependent variables. We compare network yielded call price (NCP) with market call price (MCP). We compare volatility implied by the network (NIV) with implied volatility that we found using Black [9] commodity option pricing model. To do the two types of comparisons, we do standard $t$-tests and find mean absolute errors (MAE) and root mean squared errors (RMSE). Our data period is from March 1983 to June 1995.

We use the Black [9] model to extract volatilities implied in the prices of call options on S&P 500 Index futures contracts. The Black model is modified to get implied standard deviation (ISD) per trading days as opposed to calendar days which is the basis for most studies. The modified Black model for a call option $C$, is given by:

$$C = e^{-rT_c} \times [F \times N(d_1) - K \times N(d_2)]$$

where

$$...3.6$$
\( F \) = futures price;

\( K \) = strike price;

\( r \) = risk-free interest rate;

\( T_c \) = calendar days to maturity of options;

\( d_1 = \ln \left( \frac{F}{K} \right) + \frac{1}{2} T_c \sigma^2 / (\sigma \sqrt{T_c}) \);

\( d_2 = d_1 - \sigma \sqrt{T_c} \);

\( T \) = trading days until maturity of options;

\( \sigma \) = standard deviation of percentage changes in the futures price;

\( N(.) \) = standard cumulative normal probability function.

The Newton-Raphson search technique yields the standard deviation implied by a call option. If the futures and options markets are efficient and the model is properly specified, then the standard deviation implied by the market price of the call option will represent the market consensus forecast of expected volatility of the futures contract over the remaining life of the option.

We use data on nearest to maturity S&P 500 Index futures and options on futures contract classes from March 1983 to June 1995. This period gives us 49 futures and futures options series spread over 49 quarters (12\( \frac{3}{4} \)) years. From the data of each series we find network yielded call price (NCP), network implied volatilities (NIV), and implied volatilities using Black model (OIV) on 9 days. NCPs are compared to market call prices (MCPs). NIVs are compared to OIVs.

NCPs, NIV and OIVs found on three of the days form longer horizons (57, 56 and 55 days to maturity of a series), three are for intermediate horizons (37, 36 and 35 days to maturity), and three are for shorter horizons (17, 16 and 15 days to maturity). (Futures options on the nearest futures series typically has about sixty to sixty-three days.)
On each of these nine days of a series, we use data on all call options on the nearest futures contracts traded on the day. Using the Black model, we extract implied volatilities for all futures options traded on the nine days in the life of a series. For the 49 series of options and underlying futures contracts, we get 4,131 observations involving 441 days (49 series × 9 days per series). This gave us plenty of observations for training the neural network and obtaining forecasts.

Using S&P 500 Index futures options contracts rather than S&P 500 Index options contracts has certain advantages. Feinstein [27] mentions the following advantages:

- The S&P 500 index futures and its options contracts are heavily traded in nearby pits at the Chicago Mercantile Exchange. This creates condition for simultaneous trading. Since most of the trades take place towards the end of the day, the settlement prices reported in the newspapers virtually represent actual simultaneous prices for the two contracts.
- As Kawaller, Koch and Koch [44] mention, stock index reflects the effect of last traded prices of stocks comprising the index. This may not reflect the effect of recent trades. But in valuing stock index futures, all information about the included stocks is available. The index itself may not reflect all available information.
- In pricing options on index futures, we do not need to subtract expected dividends from stock prices as in pricing index options. The futures contract incorporates market assessment on payout over the life of the futures contract.

Randolph [67] point out that the use of futures contracts rather than the spot index help in bypassing the variance estimation problems caused by autocorrelation in the index returns.

We generate call prices from neural network. We use a three-layered back propagation network. We have five input variables to generate call price. So the
network has 5 neurons in the first layer, five in the hidden layer, and one neuron for the single output that we want. We create a file with data on the following five variables: trading days to maturity, calendar days to maturity, futures price, strike price, and interest rate. (Recall that in finding implied volatilities using the Black model we modify it to get implied volatilities per trading days as opposed to calendar days; so for consistency we include data for both of them as inputs in the neural network.) The file has data on market call price on the last column.

The network trains by using data on the five input variables, producing an output comparing with the data in the last column (the desired call price). We do not use values on volatility as one of the inputs though in options pricing models it is one of the inputs in finding call price. It would be interesting to see how close to the market call price the network generates output when the network is not fed the volatility parameter. As a next step we want to see what improvement is achieved by the network when implied volatility values from the Black model are fed along with the values of other variables mentioned above.

The output we get from the network termed network implied call price (NCP). Before we can get output from the network it is trained with 102 sets of observations out of 4,131 observations in all. It is fed with the first set of data on the five input categories; it produces an output, compares with the desired call price \( C_i \) -- the market call price that goes with the five values, finds the error and feeds the magnitude of the error backwards. As the error is fed back from the output layer to the second and the first layers through the various neurons, the weights associated with each neuron is reset. Then the initial values are again fed through the network. A new output is produced. It is compared with the desired call price \( C_i \), the new error is calculated and fed back to the network and weights again reset. This feed forward and back propagation of errors is done 500 times on each set of data. Then the network is trained on the second data set 500 times. In this way the
network is trained on 102 data sets one after another - in each case, 500 times. Then the network is set to yield call price based on 103rd data set. In this data set the strike price is just-out-of-the-money (futures price minus strike price is closest to zero but negative). This forms our first observation. In this case, the futures and corresponding futures options contracts are 57 days to maturity.

Then the network trains with 1st data set to 107th data set and produces the second call price observation based on input data for 108th data set (again, in this data set, the strike price is just-out-of-the-money). This gives an observation with 56 days to maturity of futures and futures options contracts. Then the network is trained with 1st to 113th data set and then it produces call price observation with 114th data set in which case the strike price is again just-out-of-the-money. This serves as an observation for 55 days to maturity. In this way we get call price observations on 37, 36, 35, 17, 16 and 15 days to maturity of futures and futures options contracts. In each case we use the data set in which the strike price is just-out-of-the-money. We get a total of 47 call price observations for each of the nine maturity horizons. We use 102 data sets covering 2 futures and futures options contract classes. We have data on a total of 49 contract classes.

We compare the network implied call prices (NCP) on 57 days to maturity with market call price (MCP) on those days. Similarly, we compare NCPs for the other horizons with corresponding MCPs. The results of the tests of the differences in the means of market call prices (MCP) and network implied call prices for various days to maturity of the options are presented in table 3.5.

We next create a file with data on the following six input variables: trading days to maturity, calendar days to maturity, futures price, strike price, market call price and interest rate. This time the network is trained with data on these six variables to produce the futures price volatility implied by these variables. We call these network implied volatilities (NIV). In this case, we specify six neurons in the
first layer, six in the hidden layer, and one in the output layer for the single output we want. The network is trained with 102 data sets and volatility output is produced with 103rd data set as done for call price outputs. The other outputs are produced as in the case of call price outputs. Then we compare the network implied volatilities (NIV) on 57 days to maturity with option-model implied volatilities (OIV) on those days. Similarly, we compare NIVs for the other horizons with corresponding OIVs. The results of the tests of the differences in the means of option-model implied volatilities and network implied volatilities on various days to maturity of the options are presented in table 3.6. Mean absolute errors and root mean squared errors of network implied volatilities compared to option-model implied volatilities are also presented.

Table 3.5 Results of tests of differences in the means of market call prices and neural network implied call prices for options on S&P 500 Index futures (nearest contract from March 1983 to June 1993) for various days to maturity of options.

<table>
<thead>
<tr>
<th>Days to maturity</th>
<th>Means of</th>
<th>Std. Dev. of</th>
<th>t-test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Market Call Price</td>
<td>Neural Implied Call Price</td>
<td>Market Call Price</td>
</tr>
<tr>
<td>57</td>
<td>7.71</td>
<td>7.54</td>
<td>3.15</td>
</tr>
<tr>
<td>56</td>
<td>7.83</td>
<td>7.65</td>
<td>3.17</td>
</tr>
<tr>
<td>55</td>
<td>7.94</td>
<td>7.73</td>
<td>3.06</td>
</tr>
<tr>
<td>37</td>
<td>6.54</td>
<td>6.43</td>
<td>2.81</td>
</tr>
<tr>
<td>36</td>
<td>6.67</td>
<td>6.35</td>
<td>3.78</td>
</tr>
<tr>
<td>35</td>
<td>6.30</td>
<td>6.13</td>
<td>3.24</td>
</tr>
<tr>
<td>17</td>
<td>4.50</td>
<td>3.94</td>
<td>2.16</td>
</tr>
<tr>
<td>16</td>
<td>4.45</td>
<td>3.78</td>
<td>1.94</td>
</tr>
<tr>
<td>15</td>
<td>4.47</td>
<td>3.45</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Note: Differences are significant at 7% and 0% levels, respectively.
Table 3.6 Results of tests of differences in the means of option model implied volatilities (OIV) and neutral network implied volatilities (NIV) of options on S&P 500 Index futures on various days to maturities; no significant differences are found; also, mean absolute errors (MAE) and root mean squared errors (RMSE) of NIV compared to OIV are shown.

<table>
<thead>
<tr>
<th>Days to maturity</th>
<th>Means of Std. Dev. of</th>
<th>Std. Dev. of</th>
<th>t-test</th>
<th>t-test</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OIV</td>
<td>OIV</td>
<td>NIV</td>
<td>NIV</td>
<td>t-stat</td>
<td>p-value</td>
</tr>
<tr>
<td>57</td>
<td>0.0093</td>
<td>0.0021</td>
<td>0.0029</td>
<td>-0.23</td>
<td>0.41</td>
<td>0.00126</td>
</tr>
<tr>
<td>56</td>
<td>0.0095</td>
<td>0.0026</td>
<td>0.0028</td>
<td>-0.08</td>
<td>0.47</td>
<td>0.00109</td>
</tr>
<tr>
<td>55</td>
<td>0.0093</td>
<td>0.0024</td>
<td>0.0027</td>
<td>-0.43</td>
<td>0.33</td>
<td>0.00087</td>
</tr>
<tr>
<td>37</td>
<td>0.0094</td>
<td>0.0042</td>
<td>0.0056</td>
<td>-0.91</td>
<td>0.18</td>
<td>0.00177</td>
</tr>
<tr>
<td>36</td>
<td>0.0099</td>
<td>0.0044</td>
<td>0.0053</td>
<td>-0.27</td>
<td>0.40</td>
<td>0.00105</td>
</tr>
<tr>
<td>35</td>
<td>0.0095</td>
<td>0.0036</td>
<td>0.0046</td>
<td>-0.59</td>
<td>0.28</td>
<td>0.00116</td>
</tr>
<tr>
<td>17</td>
<td>0.0088</td>
<td>0.0019</td>
<td>0.0029</td>
<td>-0.72</td>
<td>0.24</td>
<td>0.00130</td>
</tr>
<tr>
<td>16</td>
<td>0.0088</td>
<td>0.0020</td>
<td>0.0028</td>
<td>-0.46</td>
<td>0.32</td>
<td>0.00098</td>
</tr>
<tr>
<td>15</td>
<td>0.0090</td>
<td>0.0037</td>
<td>0.0030</td>
<td>-0.15</td>
<td>0.44</td>
<td>0.00108</td>
</tr>
</tbody>
</table>

3.11. Analysis of Results

Table 3.5 shows the means and standard deviations of market call prices and network implied call prices on the nine selected days to maturity. The table shows the results of $t$-tests for differences in the means. The test is based on assumption of two independent samples when sigma’s are unknown (paired test). Network implied call prices are in all cases lower than market call prices. But there is no significant difference in the two types of prices for seven of the nine maturity classes: 57, 56, 55, 37, 36, 35 and 17 days to maturity. The differences are significant for the 16-day maturity class ($p$-value = 0.07) and the 15-day maturity class ($p$-value = 0.00). That implies, when it comes to shorter horizons, the network fails to generate output close to market call price. It may be that the market gets choppier as future options contracts approach maturity which creates noise that the network is not able to filter out. It may be because the network does not have enough information to generate efficient output. Recall that the input variables based on which the network learns to produce efficient output do not include futures price volatility. We intend to add volatility to the list of inputs and then
train the network to generate call price. But given that the network can generate call prices not significantly different from market call prices without the advantage of volatility data is a testament to efficiency of neural networks.

The results of tests of differences in the means of option model implied volatilities and neural network implied volatilities for selected days to maturities are given in table 3.6. The outputs from NN this time for all maturity classes are not significantly different from Black model implied volatilities. Test values are the highest for the 37-day maturity class: $p$-value = 0.18, mean absolute error = 0.0017 and root mean squared error = 0.0046. Network implied volatilities have higher means than Black model implied volatilities for all 9 maturity classes. This is consistent with outcomes for call prices from the network. The means of call prices are lower than market call prices for all maturity classes.

Neural networks have proved to be promising in various applications in finance. Results from this study show that the networks can have promising application in option pricing – in finding option price as well as implied volatility.

D. FUZZIFICATION

3.12. Fuzzy Approach in Option Pricing

Traditional modeling techniques do not capture the nature of complex systems especially when humans are involved. Fuzzy logic provides effective tools for dealing with such complex systems. Fuzzy logic, long proven in engineering and scientific applications, can now help in business dramatically and improve the way the decisions are made in respect of time forecasting for stock market strategy.
3.13. Approximate Reasoning

The ultimate goal of fuzzy logic is to form the theoretical foundation for reasoning about imprecise propositions; such reasoning has been referred to as approximate reasoning Zadeh [87, 88]. Approximate reasoning is analogous to classical logic for reasoning with precise propositions, and hence is an extension of classical propositional calculus that deals with partial truths.

Suppose we have a rule-based format to represent fuzzy information. These rules are expressed in conventional antecedent-consequent form, such as

Rule 1: IF x is A, THEN y is B,

where A and B represent fuzzy propositions (sets). Now suppose we introduce a new antecedent, say A', and we consider the following rule:

Rule 2: IF x is A', THEN y is B'.

From information derived from Rule 1, is it possible to derive the consequent in Rule 2, B'? The answer is yes, and the procedure is fuzzy composition. The consequent B' can be found from the composition operation, B' = A' ° R.

3.14. Generation of Membership Functions Using a Neural Network

We consider here a method by which fuzzy membership functions may be created for fuzzy classes of an input data set Takagi and Hayashi [72]. We select a number of input data values and divide them into a training data set and a checking data set. The training data set is used to train the neural network. Let us consider an input training data set as shown in figure 3.16a. Table 3.6 shows the coordinate values of the different data points considered (e.g., crosses in figure 3.16a). The data points are expressed with two coordinates each, since the data shown in figure 3.16a represent a two-dimensional problem. The data points are first divided into different classes (figure 3.16a) by conventional clustering techniques.
A single data point

\[
\begin{array}{c|c}
\text{Data points} & \text{R}^1 \\
1 & 1 \\
2 & 0 \\
\ldots & 0 \\
14 & 1 \\
\hline
X_1 & 0.7 \\
X_2 & 0.8 \\
\end{array}
\]

Figure 3.16: Using a neural network to determine membership functions Takagi and Hayshi [72]
Table 3.6 Variables describing the data points to be used as a training data set

<table>
<thead>
<tr>
<th>Data Point</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0.05</td>
<td>0.09</td>
<td>0.12</td>
<td>0.15</td>
<td>0.20</td>
<td>0.75</td>
<td>0.80</td>
<td>0.82</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.02</td>
<td>0.11</td>
<td>0.20</td>
<td>0.22</td>
<td>0.25</td>
<td>0.75</td>
<td>0.83</td>
<td>0.80</td>
<td>0.89</td>
<td>0.89</td>
</tr>
</tbody>
</table>

As shown in figure 3.16a data points have been divided into three regions, or classes, $R_1$, $R_2$, and $R_3$. Let us consider data point 1, which has input coordinate values of $x_1 = 0.7$ and $x_2 = 0.8$ (figure 3.16d). As this is in regions $R_2$, we assign to it a complete membership of one in class $R_2$ and zero membership in classes $R_1$ and $R_3$ (figure 3.16f). Similarly, the other data points are assigned membership values of unity for the classes they belong to initially. A neural network is created (figures 3.16b, e, h) that uses the data point marked 1 and the corresponding membership values in different classes for training itself to simulate the relationship between coordinate locations and the membership values. Figure 3.16c represents the output of the neural network which classifies data points into one of the three regions. The neural network then uses the next set of data values (e.g., point 2) and membership values to train itself further as seen in figure 3.16d. This repetitive process is continued until the neural network can simulate the entire set of input–output (coordinate location – membership value) values. The performance of the neural network is then checked using checking data set. Once the neural network is ready, its final version (figure 3.16h) can be used to determine the membership values (function) of any input data (figure 3.16g) in the different regions (figure 3.16i).

Notice that the points shown in the table in figure 3.16i are actually the membership values in each region for the data point shown in figure 3.16g. A complete mapping of the membership of different data points in the different fuzzy classes can be derived to determine the overlap of the different classes (the hatched portion in figure 3.16c shows the overlap of the three fuzzy classes).