Degree of Mispricing with the Black-Scholes Model and Nonparametric Cures

Ramazan Gençay* Aslihan Salih†

February 2001

Abstract

Black-Scholes pricing errors are larger in the deeper out-of-the-money options relative to the near out-of-the-money options, and mispricing worsens with increased volatility. Our results indicate that the Black-Scholes model is not the proper pricing tool in high volatility situations especially for very deep out-of-the-money options. Feedforward network provides more accurate pricing estimates for the deeper out-of-the money options and handles pricing during high volatility with considerably lower errors for out-of-the-money call and put options. This could be invaluable information for practitioners as option pricing is a major challenge during high volatility periods.

Keywords: Option Pricing, Nonparametric Methods, Feedforward Networks, Bayesian Regularization, Early Stopping, Bagging

*Ramazan Gençay, Department of Economics, University of Windsor, Windsor Ontario, N9B 3P4, Canada. Ramazan Gençay thanks the Social Sciences and Humanities Research Council of Canada and the Natural Sciences and Engineering Research Council of Canada for financial support.

†Faculty of Business Administration, Bilkent University, Bilkent 06533, Ankara, Turkey.

1. Introduction

The violations of the distributional assumptions behind the Black-Scholes (1973) model have been investigated extensively. Black (1976) documents that in the early years of trading on the Chicago Board of Trade, implied volatilities tended to increase with increasing strike price. Macbeth and Merville (1979) reports that the Black-Scholes prices, calculated with the implied volatility of at- or near-the-money options, are on average less (greater) than market prices for in-the-money (out-of-the-money) call options. Moreover the extent to which the Black-Scholes model underprices (overprices) an in-the-money (out-of-the-money) option increases with the extent to which the option is in-the-money (out-of-the-money) and decreases as time-to-maturity decreases. This bias acts as if the implied volatilities were inversely related to the exercise price and contrary to Black's (1976) results. Macbeth and Merville propose that these results might be due to nonstationary variance of the underlying distribution of asset returns. Rubinstein (1985) states that strike price bias is statistically significant, but the direction of the bias changes from period to period. Dumas, Fleming and Whaley (1998) argues that prior to the 1987 crash, volatilities were symmetric around zero moneyness, with in-the-money and out-of-the-money having higher implied volatilities than at-the-money options. However, after the crash, the call (put) option implied volatilities were decreasing monotonically as the call (put) went deeper into out-of-the-money (in-the-money). Since these findings cannot be explained by the Black-Scholes model and its variations, researchers searched for improved option pricing models.[1]

Nonparametric valuation models are a natural extension as it is easier to relax the distributional assumptions. In this paper, we investigate the robustness of the feedforward network models when pricing deeper out-of-the money options relative to the near out-of-the-money options. Our findings indicate that the Black-Scholes pricing errors are larger in the deeper out-of-the money options relative to the near-the-money options. Furthermore, Black-Scholes mispricing worsens with increased volatility. Feedforward networks provide more accurate pricing estimates for the deeper out-of-the money options and handles pricing during high volatility with con-

[1]Bakshi, Cao and Chen (1997) review the parametric option pricing alternatives and empirically compare the pricing performance of five different parametric models including the Black-Scholes model for S&P 500 index options. Sarwar and Krehbiel (2000) examine the out-of-sample pricing performance, bias of the stochastic volatility and modified Black-Scholes option pricing models for European currency call options.

siderably lower errors for out-of-the-money call and put options. This could be invaluable information for practitioners, as option pricing is a major challenge during high volatility periods and our findings confirm that Black-Scholes is not the proper tool for very deep out-of-the-money options.

Recently, a number of papers have used nonparametric methods to price options. Ghysels et al. (1997) provide a survey of this literature. Two papers appeal to financial theory to complement a strictly nonparametric approach. Gouriéroux, Monfort and Tenreiro (1994) apply a Kernel M-estimator methodology to the option pricing problem by extending the Black-Scholes formulation.[2] In doing so, they recognize that the Black-Scholes formula is not strictly valid, but that its shape can still be useful to recover a pricing formula more in line with observed data. Aït-Sahalia and Lo (1998) use kernel estimation techniques for the option pricing function and point out that several of the partial derivatives of the option pricing function are of special interest such as the well-known delta of the option.[3] Hutchinson, Lo and Poggio (1994) investigate several techniques for pricing and hedging options nonparametrically with radial basis functions, projection pursuit regression, and feedforward networks. Gençay and Garcia (2000) demonstrate that feedforward networks with hints can be used successfully to estimate a pricing formula for options, with good out-of-sample pricing performance. Gençay and Qi (2001) utilize bagging and Bayesian regularization methods to improve the generalization performance of feedforward networks for option pricing models.

One of the most important issues in the feedforward network estimation is to construct an estimated network with desirable generalization properties. Several methods have been suggested to prevent overfitting and to improve generalization in neural networks. These include information-based criteria such as Schwarz Information Criteria, Bayesian regularization (MacKay 1992), early stopping, and bagging[4] (Breiman 1996) which we use here to estimate parsimonious models. Our results indicate that bagging and Bayesian regularization are robust network selection methods with desirable generalization properties.

Section 2 discusses the nonparametric approach to option pricing. Section 3 describes data set. Empirical findings are presented in Section 4. We conclude afterwards.

[2]Aït-Sahalia and Lo (1998) also use the same semiparametric approach, along with their purely nonparametric approach.

[3]The first derivative of the option pricing formula with respect to the stock price.

[4]Bagging is the acronym for *b*ootstrap *agg*regat*ing*.

2. Nonparametric Option Pricing

Black-Scholes pricing formula's appeal to practitioners often originates from its analytical simplicity to determine the price of a European option on a non-dividend paying asset by

$$C_t = S_t N(d_1) - K e^{-r\tau} N(d_2) \tag{1}$$

with $d_1 = [\ln(S_t/K) + (r + 0.5\sigma^2)T]/(\sigma\sqrt{\tau})$, $d_2 = d_1 - \sigma\sqrt{\tau}$ where N is the cumulative normal distribution, S_t is the price of the underlying security, K is the exercise price, r is the prevailing risk-free interest rate, τ is the time-to-maturity and σ is the volatility of the underlying asset. Equation 1 contains neither preferences of individuals nor the preferences of the aggregate market.

Black-Scholes derivation has been mostly criticized for its distributional assumptions of the underlying security. Empiricial studies of stock price find too many outliers for a simple constant variance log-normal distribution (Merton 1976). Alternative explanations have been suggested by many researchers. Oldfield et al. (1977), Rosenfeld (1980), and Ball and Torous (1985) have fitted mixtures of continuous and jump processes to the stock price data. Black (1976), Beckers (1980), and Christie (1982) document negative correlation between stock prices and volatility. Schmalensee and Trippi (1978) found that changes in implied volatilities are negatively correlated with changes in stock prices. Blattberg and Gonedes (1974) conclude that volatility is a random process through time. Attempts to accomodate stochastic volatility and stochastic interest rates within the framework of Black-Scholes analysis have been complicated by the complexity of the estimation of the market price of risk. Bakshi, Cao and Chen (1997) provide closed form solutions for valuing options under stochastic volatility and stochastic interest rates using Heston's (1993) Fourier inversion method to calculate volatility and interest rate market risk premiums. Their results document that stochastic volatility and stochastic interest rate models are structurally misspecified. However adding the stochastic volatility feature to the Black-Scholes model improves out-of-sample pricing and hedging performance of the model. In a later paper Sarwar and Krehbiel (2000) report that the Black-Scholes model calculated with daily revised implied volatilities performs as well as the stochastic volatility model for European currency call options. Derman and Kani (1994a,b), Dupire (1994) and Rubinstein (1994) develop a deterministic volatility function (DVF) option valuation model in an attempt to exactly explain the observed cross-section of option prices.

However, Dumas, Fleming and Whaley (1998) report that the DVF option valuation model's fit is no better than an ad hoc procedure that merely smooths Black-Scholes implied volatilities across exercise prices and time-to-maturity.

Nonparametric valuation models are a natural extension as it is easier to relax the distributional assumptions. A natural nonparametric function for pricing a European call option on a non-dividend paying asset will relate the price of the option to the set of variables which characterize the option

$$C_t = f(S_t, K, \sigma_t, r_t, \tau) \tag{2}$$

where S_t is the price of the underlying asset, K is the strike price, σ_t is the volatility of the underlying asset, r_t is the interest rate and τ is the time-to-maturity. It is generally more difficult to estimate a function nonparametrically when the number of input variables is large. To reduce the number of inputs, Hutchinson, Lo and Poggio (1994) divide the function and its arguments by K and write the pricing function as follows:

$$\frac{C_t}{K} = f(\frac{S_t}{K}, 1, \sigma_t, r_t, \tau). \tag{3}$$

This form assumes the homogeneity of degree one in the asset price and the strike price of the pricing function f. Another technical reason for dividing by the strike price is that the process S_t is nonstationary while the variable S_t/K is stationary as strike prices bracket the underlying asset price process.[5] This paper uses Equation 3 as the nonparametric model for feedforward network estimation.

2.1. Feedforward Networks

An artificial neural network is a parallel distributed statistical model made up of simple data processing units, which process information in currently available data, and makes generalizations for future events. Although it is common to use neural network models in a time series context, it can also be used with problems pertaining to cross-section environments.

Amongst nonlinear methods, neural networks is one of the most recent techniques used in nonlinear modelling. This is partly due to some modelling problems encountered in the early stage of development within the neural networks field. In the earlier

[5]This point is emphasized in Ghysels et al. (1997).

literature, the statistical properties of neural networks estimators and their approximation capabilities were questionable. For example, there was no guidance in terms of how to choose the number of neurons and their configurations in a given layer and how to decide the number of hidden layers in a given network. Recent developments in the neural network literature, however, have provided the theoretical foundations for the universality of feedforward networks as function approximators. The results in Cybenko (1989), Funahashi (1989), Hornik et al. (1989,1990), and Hornik (1991) indicate that feedforward networks with sufficiently many hidden units and properly adjusted parameters can approximate an arbitrary function arbitrarily well. Hornik et al. (1990) and Hornik (1991) further show that the feedforward networks can also approximate the derivatives of an arbitrary function.

The universal approximation property in which both the unknown function and its derivatives can be uncovered from data is an important result theoretically and has immediate implications for financial and economic modelling. In options pricing, for instance, Hutchinson et al. (1994) and Garcia and Gençay (2000) demonstrate that feedforward networks can be used successfully to estimate a pricing formula for options, with good out-of-sample pricing and delta-hedging performance. In the option pricing framework, it is crucial to approximate both the function and the derivatives of the function accurately as the derivatives of the option pricing formula are the risk management tools (e.g. delta, gamma of an option). A small function approximation error may lead to larger errors in the derivatives of the function and therefore poorly approximated risk management tools. Garcia and Gençay (2000) and Gençay and Qi (2001) show that feedforward networks provide great enhancements over the parametric econometric tools in terms of providing more accurate pricing and hedging performances.

In a feedforward network model, the neurons (activation functions) are organized in layers. The layer which contains the inputs is called the input layer. Similarly, the layer where the output(s) of the network are located is called the output layer. There can be a number of layers between the input and the output layers. These layers, because they are kept between the input and the output layers, are called the hidden layers. Depending upon the network complexity or the nature of the studied problem, there can be a number of hidden layers in a neural network model. A single layer feedforward network has only one hidden layer whereas a multilayer feedforward network would have several hidden layers.

An example of a single layer feedforward network is presented in Figure 1. This

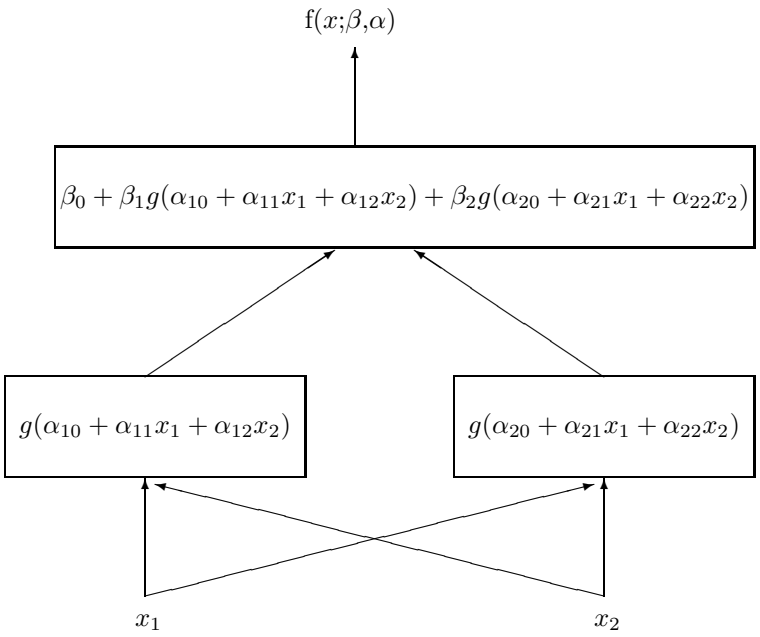


Figure 1: A two input and two hidden unit single layer feedforward network.

figure demonstrates a two input single layer feedforward network where $\mathbf{x}_t = (x_{1t}, x_{2t})$ are the inputs at t time; α_{10}, α_{11}, α_{12} are the parameters of the first activation function and α_{20}, α_{21}, α_{22} are the parameters of the second activation function. β_0, β_1, β_2 are the intercept and the slope parameters. The underlying functional form $f(\mathbf{x}_t, \theta)$ is a network output which depends on the inputs and the network parameters. The $\mathbf{x}_t$ here represents a vector of all inputs at time t and the symbol θ represents the vector of parameters, α's and β's. Often, f is termed to be the network output function. This example demonstrates that a simple feedforward network model can easily be seen as a nonlinear flexible regression model which can be estimated with the standard optimization tools used in econometrics.

A further variation of this example would be to restrict the output to a binary response. This can be achieved by assigning a threshold or signum type activation function between the hidden and the output layers. If the output is needed to be restricted to a certain interval and can take any value within this interval, the piecewise

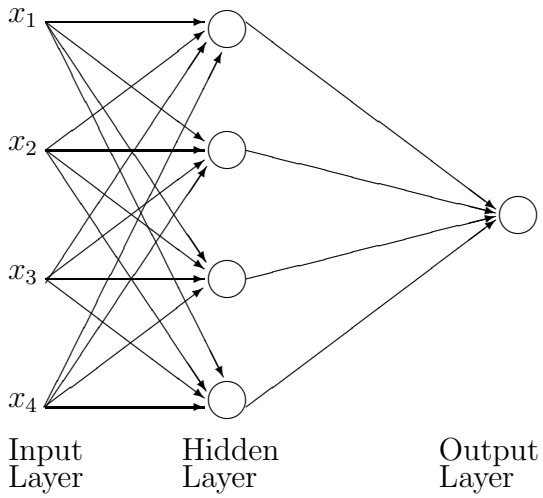


Figure 2: A single layer feedforward network with four inputs, four hidden units with a single output unit.

linear, sigmoidal or hyperbolic tangent activation functions can be used in an output layer.

The architecture of a neural network model determines the exact nature of the function f. Different types of network architectures would lead to different types of functions. An example of a single layer feedforward network with four inputs and four hidden units is presented in Figure 2. An example of a two-layer feedforward network with six inputs, four hidden units in the first hidden layer and two hidden units in the second layer is presented in Figure 3. In both figures, there is a single output unit in the output layer.

As pointed out earlier, even a *single* layer feedforward network with sufficiently many hidden units and properly adjusted parameters can theoretically approximate an arbitrary function arbitrarily well. Although these are important theoretical results which establish the universal approximation capabilities of feedforward networks, they may have limited practical implications. One element of the theoretical universal approximation results is the requirement of sufficiently many activation functions in a single hidden layer. In practice, the number of activation functions (or hidden

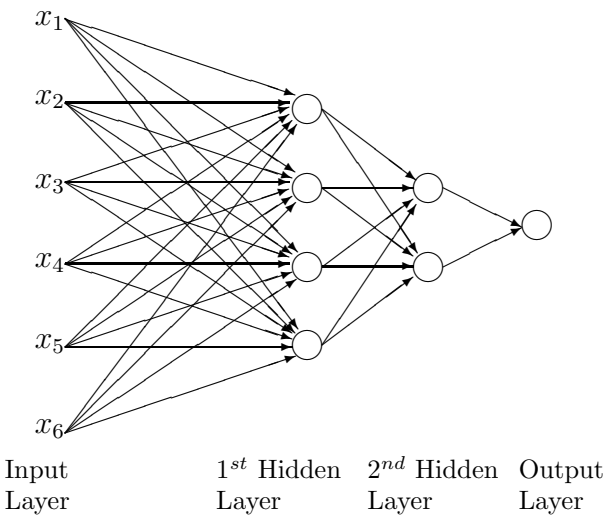


Figure 3: A two layer feedforward network with six inputs, six hidden units with a single output unit.

units) used in a network is constrained by the available degrees of freedom,[6] which is controlled by the data length and the total number of parameters of the network. Therefore, a sufficiently *large* number of hidden units in a single layer may not be feasible in certain problems such as macroeconomic data where there may only be two or three decades of annual observations available.

Let $\mathbf{x}_t$ and y_t be the input (regressors) and the target (regressand) vectors with dimensions $1 \times n$ and $1 \times w$ with t indicating the time index.[7] The observations for a sample size N are denoted by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ and $y_1, y_2, \ldots, y_N$. Given inputs $\mathbf{x}_t = (x_{1,t}, \ldots, x_{n,t})$, a single layer feedforward network regression model with q hidden units is written as

$$\begin{aligned} y_t &= s\left(\beta_0 + \sum_{i=1}^{q} \beta_i h_{i,t}\right) + \epsilon_t, \\ h_{i,t} &= g\left(\alpha_{i0} + \sum_{j=1}^{n} \alpha_{ij} x_{j,t}\right) \end{aligned} \tag{4}$$

[6]The degrees of freedom is the number of independent unrestricted random variables constituting a statistic.

[7]For simplicity, we will assume that $w = 1$ here.

for $i = 1, \ldots, q$ or

$$\begin{aligned} y_t &= s\left[\beta_0 + \sum_{i=1}^{q} \beta_i g\left(\alpha_{i0} + \sum_{j=1}^{n} \alpha_{ij} x_{j,t}\right)\right] + \epsilon_t \\ &= f(x_t, \theta) + \epsilon_t, \end{aligned} \tag{5}$$

where s and g are known activation functions; ϵ_t is an error term distributed with zero mean and variance σ_t^2 and the parameters to be estimated are $\theta = (\beta_0, \ldots, \beta_q, \alpha_1, \ldots, \alpha_q)'$ and $\alpha_j = (\alpha_{j,0}, \ldots, \alpha_{j,n})$. The range of the output values of the feedforward network model is controlled by s such that if the output takes discrete values, then s can be chosen to be a threshold function, piecewise linear function or a signum function. If the range of the output function is not restricted to a particular interval, then it can simply be set to an identity function, where $s(x) = x$. In a typical neural network model, s is normally an identity function.

Given the network structure in Equation 5 and the chosen functional forms for s and g, a major empirical issue in the neural networks is to estimate the unknown parameters θ with a sample of data values. A recursive estimation methodology, which is called *backpropogation* is such a method to estimate the underlying parameter vector θ from data.[8] In backpropogation, the starting point is a random weight θ vector that is updated[9] according to

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \eta \nabla f(\mathbf{x}_t, \hat{\theta}_t) \left[y_t - f(\mathbf{x}_t, \hat{\theta}_t)\right], \tag{6}$$

where $\nabla f(\mathbf{x}_t, \hat{\theta})$ is the (column) gradient vector of f with respect to $\hat{\theta}$ and η is the parameter which controls the learning rate. This estimation procedure is characterized by the recursive updating of estimated parameters. The parameter updates are carried out in response to the size of the error which is measured by $y_t - f(\mathbf{x}_t, \hat{\theta})$. By imposing appropriate conditions on the learning rate and functional forms of s and g, White (1989) derives the statistical properties for this estimator. He shows that the backpropagation estimator asymptotically converges to an estimator which locally minimizes the expected squared error loss. Backpropogation and nonlinear regression can be seen as alternative statistical methods to solve the least squares problem.

[8] A more detailed discussion of backpropogation can be found in Haykin (1999) and White (1992).

[9] The hat symbol denotes the estimated parameter from data.

Compared to nonlinear least squares, backpropogation fails to make efficient use of the information in the underlying data.

These recursive estimation techniques are important for large samples and real time applications since they allow for adaptive estimation. However, recursive estimation techniques do not fully utilize the information in the data sample. White (1989) further shows that the recursive estimator is not as efficient as the nonlinear least squares (NLS) estimator. One important aspect of the backpropogation methods is the choice of the learning rate η. The inefficiency of the backpropogation originates from keeping the learning rate constant in an environment where the influence of random movements in x_t are not accounted for in y_t. This would lead the parameter vector $\hat{\theta}$ to fluctuate indefinitely. A minimum requirement is to drive the learning rate gradually to zero to achieve convergence. In fact, White (1989) demonstrates that η_t has to be chosen not as a vanishing scalar but as a gradually vanishing matrix of a very specific form. These arguments on learning rates are only valid if the environment is not changing over time (stationary environment). If the environment is evolving (nonstationary environment), a gradually vanishing learning rate may fail and a constant learning rate may be more suitable (see White,1989).

This paper uses the NLS estimator which minimizes

$$\min_\theta \; L(\theta) = \sum_{t=1}^{N} [y_t - f(\mathbf{x}_t, \theta)]^2. \tag{7}$$

Here, the goal is to choose the parameter vector θ such that the sum of squared errors are minimized as much as possible. Since the function f is nonlinear (a neural network model) and it is a nonlinear function of θ, this procedure is named as nonlinear least squares or nonlinear regression. This is a straightforward multivariate minimization problem. Conjugant gradient routines studied in Gençay and Dechert (1992) work very well for this problem. In Gallant and White (1992), it is shown that the least squares method can consistently estimate a function and its derivatives from a feedforward network model, provided that the number of hidden units increases with the size of the data set. This would mean that a larger number of data points would require a larger number of hidden units to avoid overfitting in noisy environments.

2.2. Network Selection

2.2.1. Information Theoretic Criteria

The specification of a feedforward network model requires the choice of the type of inputs, the number of hidden units, the number of hidden layers and the connection structure between the inputs and the output layers. The common choice for this specification design is to adopt the model-selection approach. Information based criteria such as the Schwarz Information Criterion (SIC) and the Akaike Information Criterion (AIC) are used widely. The SIC is computed by (Schwarz, 1978)

$$\text{SIC} = \log\left[\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2\right] + \frac{w}{N}\log(N) \tag{8}$$

where w is the number of parameters in the model and N is the number of observations. The model with the smallest SIC is the preferred model. The first term in the SIC criterion is the mean squared error (MSE)

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \tag{9}$$

where y_t is the target variable at time t and $\hat{y}_t$ is the estimated network output at time t. The second term in SIC indicates that the simple estimation model with fewer number of parameters is better if both models give the same MSE's. When two models have the same number of parameters, the comparison of SIC is the same as the comparison of the mean squared errors.

The AIC is computed by (Akaike 1973,1974)

$$\text{AIC} = \log\left[\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2\right] + \frac{2}{w}N \tag{10}$$

where w is the number of parameters and N is the number of observations.

Swanson and White (1995) report that the SIC fails to select sufficiently parsimonious models in terms of being a reliable guide to the out-of-sample performance. Since the SIC imposes the more severe penalty than the AIC, the results with AIC would lead to poorer out-of-sample predictions.

2.2.2. Bayesian Regularization

To design a network which generalizes outside of the training data, MacKay (1992) proposes a method to constrain the size of the network parameters through the so-called regularization. With regularization, the objective function becomes

$$F = \gamma E_D + (1 - \gamma) E_\theta \tag{11}$$

where E_D is the sum of the squared errors, E_θ is the sum of squares of the network parameters, and γ is the performance ratio, the magnitude of which dictates the emphasis of the training. If γ is very large, then the training algorithm will produce small errors. But if γ is very small, then training will emphasize parameter size reduction at the expense of network errors, thus producing a smoother network response.

The optimal regularization parameter γ can be determined by the Bayesian techniques.[10] In the Bayesian framework the weights of the network are considered random variables. Let $D = (y, \mathbf{x})$ represent the data set, θ represent the vector of network parameters, and M be the particular neural network model used. With the data set D, the density function for the weights can be updated according to the Bayes' rule

$$P(\theta|D, \gamma, M) = \frac{P(D|\theta, \gamma, M) P(\theta|\gamma, M)}{P(D|\gamma, M)} \tag{12}$$

where $P(\theta|\gamma, M)$ is the prior density, which represents our knowledge of the weights before any data is collected, $P(D|\theta, \gamma, M)$ is the likelihood function, which is the probability of the data occurring given the weights θ. $P(D|\gamma, M)$ is a normalization factor, which guarantees that the total probability is 1. If we assume that the noise and the prior distribution for the weights are both Gaussian, the probability densities can be written as

$$P(D|\theta, \gamma, M) = (\pi/\gamma)^{-N/2} e^{-\gamma E_D} \tag{13}$$

and

$$P(\theta|\gamma, M) = [\pi/(1 - \gamma)]^{-L/2} e^{-(1-\gamma) E_\theta} \tag{14}$$

where L is the total number of parameters in the neural network model. Substituting Equation 14 into Equation 12, we obtain

[10] MacKay (1992) and Foresee and Hagan (1997) have detailed studies on this issue.

$$P(\theta|D,\gamma,M) = Z_F(\gamma)e^{-F(\theta)} \ . \tag{15}$$

In the Bayesian framework, the optimal weights should maximize the posterior probability $P(\theta|D,\gamma,M)$, which is equivalent to minimizing the regularized objective function given in Equation 11.

The performance ratio can also be optimized by applying the Bayes' rule,

$$P(\gamma|D,M) = \frac{P(D|\gamma,M)P(\gamma|M)}{P(D|M)} \ . \tag{16}$$

Assuming a uniform prior density $P(\gamma|M)$ for the regularization parameter γ, the maximization of the posterior is achieved by maximizing the likelihood function $P(D|\gamma,M)$. Since all probabilities have a Gaussian form, the normalization factor can be expressed as

$$P(D|\gamma,M) = (\pi/\gamma)^{-N/2}[\pi/(1-\gamma)]^{-L/2}Z_F(\gamma) \ . \tag{17}$$

Assuming that the objective function has a quadratic shape in a small area surrounding a minimum point, we can expand $F(\theta)$ around the minimum point of the posterior density θ^*, where the gradient is zero. Solving for the normalizing constant yields

$$Z_F \approx (2\pi)^{L/2}(\det((H^*)^{-1}))^{1/2}e^{-F(\theta^*)} \tag{18}$$

where $H = \gamma \bigtriangledown^2 E_D + (1-\gamma) \bigtriangledown^2 E_\theta$ is the Hessian matrix of the objective function. Substituting Equation 18 into Equation 17, we can solve for the optimal value of γ at the minimum point. This is done by taking the derivative with respect to the log of Equation 17 and setting it equal to zero.

The Bayesian optimization of the regularization parameters requires the computation of the Hessian matrix of $F(\theta)$ at the minimum point θ^*. Foresee and Hagan (1997) propose using the Gauss-Newton approximation to Hessian matrix, which is readily available if the Levenberg-Marquardt optimization algorithm is used to locate the minimum point. The additional computation required of the regularization is thus minimal.

2.2.3. Early Stopping

With a goal to obtain a model with desirable generalization properties, it is difficult to decide when it is best to stop training by just looking at the learning curve for

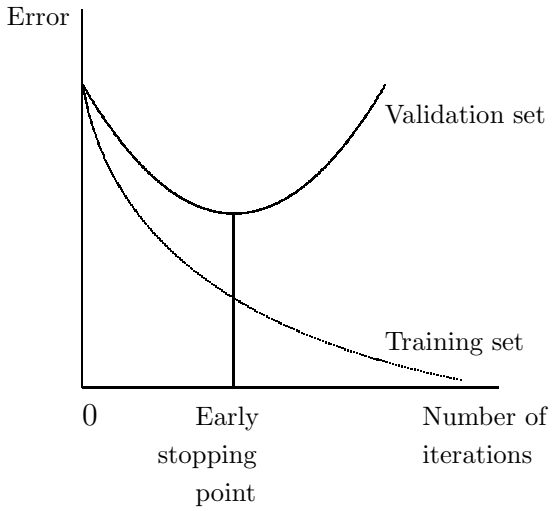


Figure 4: Early stopping method. The validation error will normally decrease during the initial phase of training, as does the error on the training set. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise. In the method of early stopping, when the validation error increases for a specified number of iterations, the training is stopped, and the weights at the minimum of the validation error are returned.

training by itself. It is possible to overfit the training data if the training session is not stopped at the right point.

The onset of overfitting can be detected through cross-validation in which the available data are divided into training, validation, and prediction (testing) subsets. The training subset is used for computing the gradient and updating the network weights. The error on the validation set is monitored during the training session. The validation error will normally decrease during the initial phase of training (see Figure 4), as does the error on the training set. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise. In the method of early stopping, when the validation error starts to increase after a number of iterations, the training is stopped, and the weights at the minimum of the validation error are returned for the optimum network complexity.

2.2.4. Bagging

In bagging (or bootstrap aggregating), multiple versions of a predictor are generated and they are used to get an aggregated predictor. The multiple versions are formed

by making bootstrap replicates of the training set and using these as new training sets. When predicting a numerical outcome, the aggregation takes the average over the multiple versions that are generated from bootstrapping. According to Breiman (1996), both theoretical and empirical evidence suggests that bagging can greatly improve the forecasting performance of a good but unstable model where a small change in the training data can result in large changes in a model.

Let L represent the training set that consists of data $\{(y_t, \mathbf{x}_t), t = 1, ..., N_L\}$, where N_L is the number of observations in the training set. Let a neural network model be fitted to the training set and this generates a predictor $f(\mathbf{x}_t, L)$, e.g., if the input is $\mathbf{x}_t$, y_t is predicted by $f(\mathbf{x}_t, L)$. Now, suppose we have a sequence of training sets $\{L_k, k = 1, ..., K\}$ each consisting of N_L independent observations from the same underlying distribution as L. We can use the $\{L_k\}$ to get a better predictor than the single learning set predictor $f(\mathbf{x}_t, L)$ by working with the sequence of predictors $\{f(\mathbf{x}_t, L_k)\}$. An obvious procedure is to replace $f(\mathbf{x}_t, L)$ by the average of $f(\mathbf{x}_t, L_k)$ over k, i.e., by $f_A(\mathbf{x}_t) = \sum_{k=1}^{K} f(\mathbf{x}_t, L_k)$. However, usually there is only a single training set L without the luxury of replicates of L. In this case, repeated bootstrap[11] samples $L^{(b)} = \{(y_t^{(b)}, \mathbf{x}_t^{(b)}), t = 1, ..., N_L\}$ can be drawn from $L = \{(y_t, \mathbf{x}_t), t = 1, ..., N_L\}$. Each $\{(y_t^{(b)}, \mathbf{x}_t^{(b)})\}$ is a random pick from the original training set $\{(y_t, \mathbf{x}_t), t = 1, ..., N_L\}$ with replacement. The bootstrap samples $L^{(b)}$ are used to form predictors $\{f(\mathbf{x}_t, L^{(b)}\}$. The bagging predictor f_B can thus be calculated as

$$f_B(\mathbf{x}_t) = \sum_{b=1}^{B} f(\mathbf{x}_t, L^{(b)}) \tag{19}$$

where B represents the total number of bootstrap replicates of the training set.

We slightly modify the bagging procedure of Breiman (1996). First, the available data are divided into the training, validation, and prediction subsets. Second, a bootstrap sample is selected from the training set. The bootstrap sample is then used to train the feedforward network with 1 to 10 hidden layer units. The validation set is used to select the best feedforward network that has the optimal number of hidden layer units, and the best model is used to generate one set of prediction on the testing set. This is repeated 25 times giving 25 sets of predictions ($B = 25$). Third, the bagging prediction is the average across the 25 sets of predictions, and the prediction error is computed as the difference between the actual and the bagging prediction values.

[11] Different bootstrap procedures can be implemented according to the nature of the data.

3. Data Description

The data are daily S&P 500 index[12] options obtained from the Chicago Board of Exchange for the period January 1988 to October 1993. The S&P 500 index option market is extremely liquid and it is one of the most active options markets in the United States. This market is the closest to the theoretical setting of the Black-Scholes model. The option contracts on this index trade on the Chicago Board Options Exchange and mature on the Saturday following the third Friday in the expiration month. They are actively European style options, and the settlements are always in cash. S&P 500 index options are very popular among institutional investors as portfolio insurance instruments. For each option written on the S&P 500 index, the data set contains the date of the transaction, expiration month, closing market price of the option, put-call identifier, exercise price, daily S&P 500 closing index, the number of days to maturity, daily S&P 500 returns, dividend yields and the interest rate at the maturity of the option.

In constructing the data used in the estimation, options with zero volume are not used. Put-call parity checks are done to eliminate erroneous prices, therefore a put (call) is only included if there is a call (put) with the same exercise price trading at that particular date. For Black-Scholes price calculations, historical volatilities are calculated using the daily S&P 500 returns. If an option has less than 22 days to expiration, historical volatility is calculated using the last 22 days daily returns. If an option has more than 22 days to maturity then the historical volatility is calculated using the historical returns that match the exact number of days to maturity.[13]

For each year, the sample is split into three parts: first half of the year (training period), third quarter (validation period) and fourth quarter (prediction period). One

[12] S&P 500 Stock index represents the market value of all outstanding common shares of 500 firms selected by Standard and Poor's.

[13] Black-Scholes prices are calculated with the Merton (1973) option pricing formula which incorporates continuous dividend yield adjustment $C_t = S_t e^{-q\tau} N(d_1) - K e^{-r\tau} N(d_2)$ where $d_1 = [\ln(S_t/K) + ((r-q) + 0.5\sigma^2)T]/\sigma\sqrt{\tau}$, $d_2 = d_1 - \sigma\sqrt{\tau}$ and q is a dividend yield.

One can actually be more picky about the data and the method used in the Black-Scholes price calculation such as using the actual dividend stream of the S&P 500 index instead of the Merton's continous dividend adjustment, or using high frequecy data for synchronous prices of options and the underlying index, or using the implied volatilities instead of historical volatility. However we use the exact same data for the feedforward networks to price the options and hence comparisons between parametric and nonparametric methods are fair. The degree of mispricing induced by the data in both methods is beyond the scope of current research.

possible drawback of such a setup is that we will always evaluate the predictive ability of our networks on the last quarter of the year. The advantage is that it will facilitate comparison between years. We estimate networks with 1 to 10 hidden units over half of the data points for a particular year, the training sample. Next, we choose the network in each family that gives the best mean square prediction error over half of the remaining data points in the sample, called the validation sample. Finally, we assess the prediction performance (MSPE) of the best model chosen in the previous step for the models from the four methods over the last quarter of data, the prediction sample.

4. Empirical Findings

The network pricing performance measure is the mean squared prediction error (MSPE) in the prediction sample. Results are presented in Table 1. For each year, we report the average mean squared prediction errors (MSPE) of ten estimations for each family of networks, along with the average number of hidden units selected,[14] standard deviations of ten estimations and the p-values of the Diebold-Mariano (1995) statistics. A ratio of the Black-Scholes model's MSPE relative to that of the neural network models is also reported.

Table 1 indicates that all model selection methods provide substantially smaller MSPE's relative to the Black-Scholes model. For 1988, the improvements in the MSPE's are in the order of 40 percent for the feedforward network models over the Black-Scholes model. For 1989-1993, the improvements in the MSPE's vary between 80-83% in favour of the feedforward networks when compared to the MSPE's of the Black-Scholes model. Between the model selection methods, Bayesian regularization (BR) and bagging (BA) methods outperform the SIC and early stopping (ES) methods. ES does not affect the pricing accuracy. For 1988, 1989 and 1991, the BR method provides the best pricing performance in the prediction sample. In 1990, 1992 and 1993, the BA method is the best performing model selection method in terms of best prediction performance.

The Diebold-Mariano (DM) test measures the loss differential of the mean squared prediction errors between the feedforward network models. For 1988, the p-value of

[14]To control for the potential uncertainty in the relative performance that might be caused by different random seeds, the training starts with the same set of initial random weights for all model selection methods.

the DM test is 1.6% for the BR model. In 1990 and 1993, the DM is less than one percent for BA and it is approximately 1% for BR in 1991. For these years, the p-values indicate statistically significant differentials for the MSPE's of the BA and BR methods when compared to the SIC-based networks. Although the SIC methodology is commonly used in feedforward network selection, our results indicate that more robust networks can be estimated with the Bayesian regularization and bagging methods.

In an attempt to explore the complexity of the problem for the option pricing models, the market/exercise price (C/K) is plotted against time-to-maturity (τ) in Figure 5 for out-of-the-money call options. Figure 5 illustrates three cases, namely, the deepest out-of-the-money call options ($S/K < 0.95$), deeper out-of-the-money call options ($S/K \geq 0.95$ & $S/K < 0.97$) and the near out-of-the-money ($S/K \geq 0.97$ & $S/K < 0.99$) call options. As Figure 5 illustrates there is a positive, but quite noisy, relationship between C/K and τ for the near out-of-the-money call options. As it is moved to deeper out-of-the-money call options, the relationship becomes noisier with apparent outliers, and there is hardly any obvious functional relationship for the deepest out-of-the money options. This figure illustrates the difficulty of estimating the price of out-of-the-money call options due to the nature of the outliers and the noise in the empirical data.

Figure 6 depicts the relationship between the market and Black-Scholes prices. The first observation is that the Black-Scholes prices are biased estimates of the market prices. For the deepest out-of-the-money options, the Black-Scholes prices overestimate market prices whereas market prices are underestimated for the deeper and the near out-of-the money options. In particular, the performance of the Black-Scholes model in explaining the observed market prices is quite poor for the deepest out-of-the-money options.

In Figure 7, the relationship between the market and the feedforward network prices is presented. The feedforward networks largely eliminate the overestimation bias observed in Figure 6 with the Black-Scholes model. The estimated deepest out-of-the-money call options are centered around the 45-degree line with substantially less and smaller outliers. The performance for the deeper and the near out-of-the-money call options are also substantially improved without any evidence of underestimation bias and lack of outliers. The comparison of Figures 6 and 7 indicates that the feedforward network corrects under and overestimation bias of the Black-Scholes model successfully. Since both feedforward network and the Black-Scholes model use identi-

cal inputs, the gain from the feedforward network originates from flexible functional form at which Black-Scholes model may be constraining the data unnecessarily. The findings of the figures above corraborate the results in Table 1 where the MSPE's of the feedforward networks when compared to the MSPE's of the Black-Scholes model provide 40-80 percent gains across years.

To investigate the effect of volatility on the mispricing, Figures 8 and 9 analyse the relationship between pricing errors and volatility for the Black-Scholes and feedforward network models, respectively. An ideal model should have ball shaped pricing errors centered around zero at all volatility levels. For extreme volatilities, a desirable pattern is symmetric pricing errors centered around zero. Investigations of Figure 8 provide a number of insights for the performance of the Black-Scholes model. For the deepest out-of-the-money options ($S/K < 0.95$), there are large positive pricing errors for high volatility levels (volatility levels between 0.25 and 1) and pricing errors are hardly symmetric around zero. For low volatility levels, there is a ball-type pricing error around zero although there are a large number of negative errors for volatilities between 0.10 and 0.20. For the deeper out-of-the-money options ($S/K \geq 0.95$ & $S/K < 0.97$), the performance of the Black-Scholes model is more satisfactory, although large positive pricing errors at higher volatility levels and negatively skewed pricing errors at low volatility levels remain. The performance for the near out-of-the-money call options ($S/K \geq 0.97$ & $S/K < 0.99$) is similar to that of the deeper out-of-the-money options. In Figure 9, the study is done between the pricing errors of the feedforward model and the underlying volatility. The results with the deepest out-of-the-money options is the most striking. Large positive pricing errors which were quite dominant in the Black-Scholes model are now largely eliminated such that pricing errors are centered around zero at high volatility levels. The negatively biased pricing errors at the low levels of volatility are also largely corrected. The examinations of the deeper out-of-the money and near out-of-the money options also indicate clear pricing error patterns centered around zero for all levels of volatilities. These figures reveal that when results are examined from the volatility window, a number of results emerge. In particular, feedforward network provides lower bias in terms of the pricing performance relative to the Black-Scholes model; Black-Scholes mispricing worsens with increasing volatility and feedforward networks handle pricing during high volatility with considerably lower errors for out-of-the-money calls. This could be invaluable information for practitioners as option pricing is a major challenge during high volatility periods and our findings confirm that Black-Scholes

is not the proper pricing tool for very deep out-of-the-money options.

Further investigation is carried out between time-to-maturity and pricing errors. An ideal model should exhibit pricing errors centered around zero at all levels of time-to-maturity. Figure 10 illustrates that there are large positive pricing errors at all levels of time-to-maturity for the deepest out-of-the-money call options estimated with the Black-Scholes model. For the deeper out-of-the-money calls and the near out-of-the-money calls, there are negatively slanted pricing errors towards higher levels of time-to-maturity with large positive pricing errors remaining. In Figure 11, feedforward network pricing errors are plotted against the time-to-maturity. Feedforward networks successfully eliminate large positive pricing errors which were dominating in Figure 10. For all levels of the out-of-the-money calls, the pricing errors are centered around zero with no evidence of bias in either direction and lack of outliers. This other view of the data provides further support for our earlier findings that feedforward networks are invaluable tools for pricing options, in particular for the deepest out-of-the-money calls.

We have also repeated the same study for the put options. Similar findings prevail between feedforward networks and the Black-Scholes model where the deterioration in the Black-Scholes model is the largest for the deepest out-of-the-money put options. Large systematic pricing errors are also present at the deeper out-of-the-money and the near out-of-the-money put options for the Black-Scholes model. Feedforward networks successfully correct for the over and under estimation pricing bias of the Black-Scholes model.

5. Conclusions

For the deepest out-of-the-money options, the Black-Scholes prices overestimate market prices whereas market prices are underestimated for the deeper and near out-of-the money options. In particular, the performance of the Black-Scholes model in explaining the observed market prices is quite poor for the deepest out-of-the-money options. The feedforward networks largely eliminate the overestimation bias observed in the Black-Scholes model. The estimated deepest out-of-the-money call options exhibit substantially less and smaller outliers. The performance of the deeper and the near out-of-the-money call options are also substantially improved without any evidence of underestimation bias and lack of outliers.

To investigate the effect of volatility on the mispricing, we analyse the relation-

ship between pricing errors and volatility. An ideal model should have ball shaped pricing errors centered around zero for all volatility levels. For extreme volatilities, a desirable pattern is symmetric pricing errors centered around zero. For the deepest out-of-the-money options, there are large positive pricing errors for high volatility levels and pricing errors are hardly symmetric around zero for the Black-Scholes model. Feedforward network models successfully eliminate large positive pricing errors which are quite dominant in the Black-Scholes model for the deepest out-of-the-money options. The examinations of the deeper out-of-the money and near out-of-the money options also indicate clear pricing error patterns centered around zero for all levels of volatilities. Overall findings indicate that Black-Scholes mispricing worsens with increasing volatility and feedforward networks handle pricing during high volatility with considerably lower errors for out-of-the-money call and put options. This could be invaluable information for practitioners as option pricing is a major challenge during high volatility periods.

Table 1

Out-of-Sample Mean Square Prediction Errors of the S&P 500 Call Options

(1988, Total Sample: 3434, Validation Sample: 1642, Prediction Sample: 1479)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.8044 (7)	0.7321 (7)	0.8044 (7)	0.7591 (7)	1.2905
σ	0.1084	0.0631	0.1085	0.0601	
DM		0.0155		0.0867	
Ratio	0.6233	0.5673	0.6233	0.5882	

(1989, Total Sample: 3052, Validation Sample: 1565, Prediction Sample: 1515)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.4093 (9)	0.3991 (8)	0.4093 (9)	0.4003 (7)	1.8912
σ	0.0041	0.0069	0.0041	0.0025	
DM		0.8116		0.9734	
Ratio	0.2164	0.2110	0.2164	0.2117	

(1990, Total Sample: 3605, Validation Sample: 2075, Prediction Sample: 2166)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.6003 (7)	0.5911 (6)	0.6003 (7)	0.5789 (7)	3.2905
σ	0.0089	0.0091	0.0089	0.0028	
DM		0.9545		0.0010	
Ratio	0.1824	0.1796	0.1824	0.1759	

(1991, Total Sample: 4481, Validation Sample: 1922, Prediction Sample: 2061)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.3651 (8)	0.3457 (9)	0.3651 (8)	0.3478 (7)	1.7110
σ	0.0310	0.0058	0.0310	0.0065	
DM		0.0082		0.0289	
Ratio	0.2134	0.2020	0.2134	0.2033	

(1992, Total Sample: 4374, Validation Sample: 1922, Prediction Sample: 1848)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.1398 (6)	0.1426 (6)	0.1398 (6)	0.1394 (7)	1.2110
σ	0.0064	0.0023	0.0064	0.0019	
DM		0.8723		0.1912	
Ratio	0.1154	0.1178	0.1154	0.1151	

(1993, Total Sample: 4214, Validation Sample: 1973, Prediction Sample: 2030)

Statistics	SIC	BR	ES	BA	BS
$\bar{x}$	0.0551 (8)	0.0573 (8)	0.0551 (7)	0.0485 (7)	0.7309
σ	0.0041	0.0043	0.0041	0.0008	
DM		0.8824		0.0003	
Ratio	0.0754	0.0784	0.0754	0.0664	

Notes: This table presents the out-of-sample mean square prediction error (MSPE) performance of feedforward networks and the Black-Scholes model for call option prices from the SP500 call options. SIC, BR, ES, BA and BS refer to Schwarz Information Criteria, Bayesian regularization, early stopping, bagging and Black-Scholes model, respectively. The table reports the average ($\bar{x}$) of the ten MSPEs corresponding to ten networks estimated from different seeds. The average number of hidden units of the ten runs are reported between parentheses next to the average MSPEs. σ is the standard deviation of the ten MSPEs of the estimated networks. The *Ratio* is the ratio of MSPEs of the Black-Scholes model and the feedforward network models. DM refers to p-values of the Diebold and Mariano (1995) test for a mean loss differential. This test statistic is distributed standard normal in large samples. All DM test statistics are calculated from the loss differential of the mean square prediction errors between the feedforward network models. MSPE reported figures have been multiplied by 10^4.

References

[1] Aït-Sahalia, Y. and A. Lo (1998), Nonparametric Estimation of State-Price Densities Implicit in Financial Asset Prices, *Journal of Finance*, 53, 499-547.

[2] Akaike, H. (1973), Information Theory and an Extension of the Maximum Likelihood Principle, in B. N. Petrov and E. Csaki, eds, *Proceedings of the 2nd International Symposium on Information Theory, Akademia Kiado, Budapest*, 267–281.

[3] Akaike, H. (1974), A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*, 19, 716–723.

[4] Bakshi, G., C. Cao, and Z. Chen (1997), Empirical Performance of Alternative Option Pricing Models, *Journal of Finance*, 52, 2003-2049.

[5] Ball, A.C., and W. Torous (1985), On Jumps in Common Stock Prices and Their Impact on Call Option Pricing, *Journal of Finance*, 40, 155-174.

[6] Beckers, S. (1980), The Constant Elasticity of Variance Model and its Implications for Option Pricing, *Journal of Finance*, 35, 661-673.

[7] Black, F., (1976), Studies of Stock Price Volatility Changes, *Proceedings of the 1976 Meetings of the American Statistical Association*, 177-181.

[8] Black, F., and M. S. Scholes (1973), The Pricing of Options and Corporate Liabilities, *Journal of Political Economy*, 81, 637-659.

[9] Blattberg, R., and N. Gonedes (1974), A Comparison of Stable and Student Distribution of Statistical Models for Stock Prices, *Journal of Business*, 47, 244-280.

[10] Breiman, L. (1996), Bagging Predictors, *Machine Learning*, 24, 123-140.

[11] Christie, A. (1982), The Stochastic Behavior of Common Stock Variances: Value, Leverage, and Interest Rate Effects, *Journal of Financial Economics*, 10, 407-432.

[12] Cybenko, G. (1989), Approximation by Superposition of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 2, 303–314.

[13] Derman, E., and I. Kani (1994a), The Volatility Smile and its Implied Tree, *Quantitative Strategies Research Notes*, Goldman Sachs, New York.

[14] Derman, E., and I. Kani (1994b), Riding on the Smile, *Risk*, 7, 32-39.

[15] Diebold, F., and R. Mariano (1995), Comparing Predictive Accuracy. *Journal of Business and Economic Statistics* 13, 253-263.

[16] Dumas, B., J. Fleming, and R. E. Whaley (1998), Implied Volatility Functions: Empirical Tests, *Journal of Finance*, 53, 2059-2106.

[17] Dupire, B. (1994), Pricing with a Smile, *Risk*, 7, 18-20.

[18] Foresee, F. D., and M. T. Hagan (1997), Gauss-Newton Approximation to Bayesian Learning, *Proceedings of IEEE International Conference on Neural networks*, 3, 1930–1935.

[19] Funahashi, K. (1989), On the Approximate Realization of Continuous Mappings by Neural Networks, *Neural Networks*, 2, 183–192.

[20] Gallant, A. R., and H. White (1992), On Learning the Derivatives of an Unknown Mapping with Multilayer Feedforward Networks, *Neural Networks*, 5, 129–138.

[21] Garcia, R., and R. Gençay (2000), Pricing and Hedging Derivative Securities with Neural Networks and a Homogeneity Hint, *Journal of Econometrics*, 94, 93–115.

[22] Gençay, R., and W. D. Dechert. (1992), An Algorithm for the n Lyapunov Exponents of an n-Dimensional Unknown Dynamical System, *Physica D*, 59, 142–157.

[23] Gençay, R. and M. Qi (2001), Pricing and Hedging Derivative Securities with Neural Networks and Bayesian Regularization, Early Stopping and Bagging, *IEEE Transactions on Neural Networks*, forthcoming.

[24] Ghysels, E., V. Patilea, E. Renault, and O. Torrès (1997), Nonparametric Methods and Option Pricing, *Statistics and Finance*, D. Hand and S. Jacka (eds.), Edward Arnold, London, Ch. 13, 261-282.

[25] Gouriéroux, C., A. Monfort and C. Tenreiro (1994), Kernel M-Estimators: Nonparametric Diagnostics for Structural Models, Working Paper 9405, CEPREMAP, Paris.

[26] Haykin, S. (1999), *Neural Networks*, Prentice Hall, New Jersey, Second edition.

[27] Heston, S. (1993), A Closed Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency options, *Review of Financial Studies*, 6, 327-343.

[28] Hornik, K. (1991), Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, 4, 251–257.

[29] Hornik, K., Stinchcombe, M., and H. White (1989), Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, 359–366.

[30] Hornik, K., Stinchcombe, M., and H. White (1990), Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks. *Neural Networks*, 3, 551–560.

[31] Hutchinson, J. M., Lo, A., and A. W. Poggio (1994), A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks, *Journal of Finance*, 31, 851–889.

[32] Macbeth, J. D., and L. J. Merville (1979), An Empiricial Examination of Black-Scholes Call Option Pricing Model, *Journal of Finance*, 34, 1173-1186.

[33] MacKay, D. J. C. (1992), Bayesian Interpolation, *Neural Computation* 4, 415-447.

[34] Merton, R. C. (1973), Theory of Rational Option Pricing, *Bell Journal of Economics and Management Science*, 4, 141-182.

[35] Merton, R. C. (1976), Option Pricing When Underlying Stock Returns are Discontinous, *Journal of Financial Economics*, 3, 125-144.

[36] Oldfield, G. S., R. J. Rogalski, and R. A. Jarrow (1977), An Autoregressive Jump Process for Common Stock Returns, *Journal of Financial Economics*, 5, 389-418.

[37] Rosenfeld, E. (1980), Stochastic Processes of Common Stock Returns: An Empiricial Investigation, Ph.D. Dissertation, MIT.

[38] Rubinstein, M. (1985), Nonparametric Tests of Alternative Option Pricing Models Using All Reported Trades and Quotes on the Thirty Most Active CBOE Option Classes from August 23, 1976 Through August 3, 1978, *Journal of Finance*, 40, 455-480.

[39] Rubinstein, M. (1994), Implied Binomial Trees, *Journal of Finance*, 49, 771-818.

[40] Sarwar, G., and T. Krehbiel (2000), Empirical Performance of Alternative Pricing Models of Currency Options, *Journal of Futures Markets*, 20, 265-291.

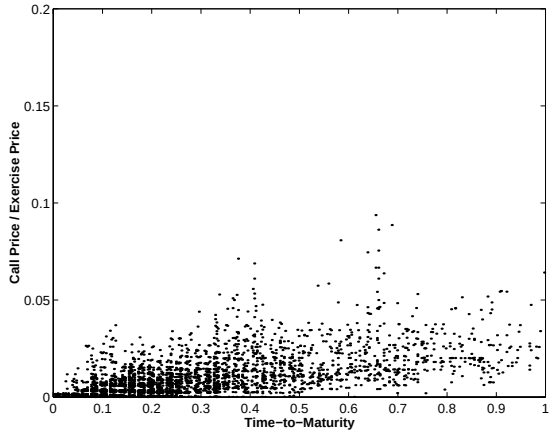
[41] Schmalensee, R., and R. R. Trippi (1978), Common Stock Volatility Expectations Implied by Option Premia, *Journal of Finance*, 33, 129-147.

[42] Schwarz, G. (1978), Estimating the Dimension of a Model, *Annals of Statistics*, 6, 461–464.

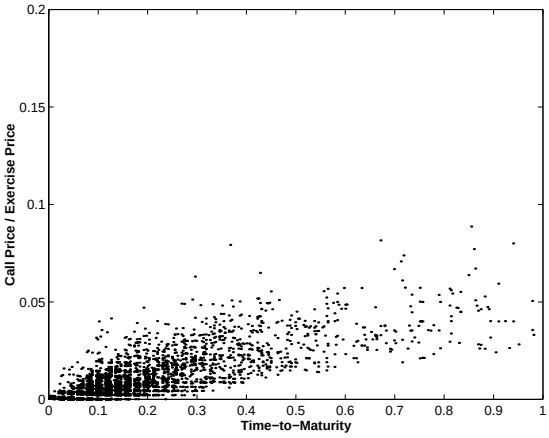
[43] Swanson, N., and H. White (1995), A Model-Selection Approach to Assessing the Information in the Term Structure Using Linear Models and Artificial Neural Networks, *Journal of Business and Economic Statistics*, 13, 265–275.

[44] White, H. (1989), Some Asymptotic Results for Learning in Single Hidden-Layer Feedforward Network Models, *Journal of the American Statistical Association*, 94, 1003–1013.

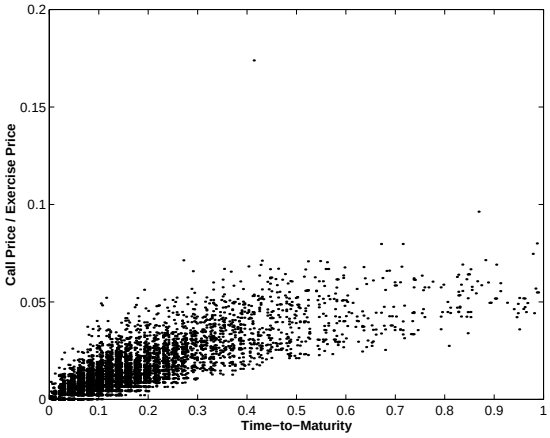
[45] White, H. (1992), *Artificial Neural Networks: Approximation and Learning*, Blackwell, Cambridge.



(a) S/K < 0.95

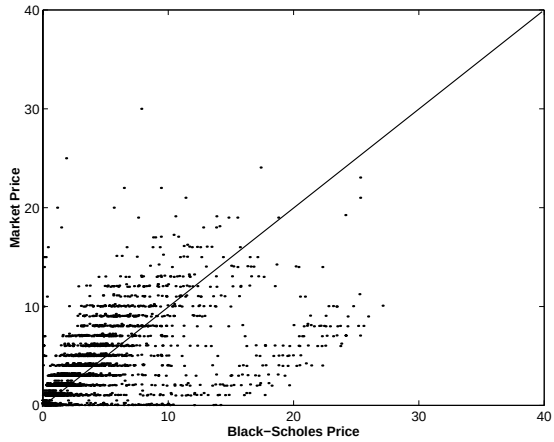


(b) S/K ≥ 0.95 & S/K < 0.97

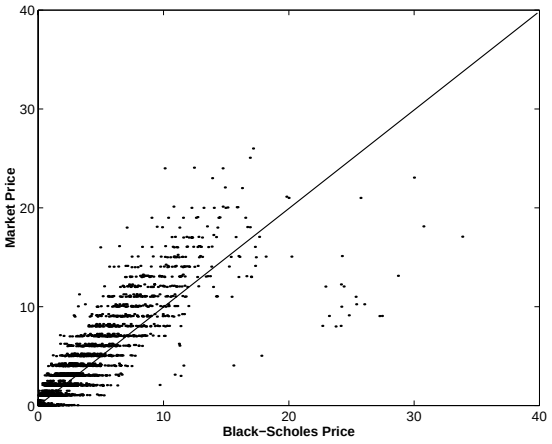


(c) S/K ≥ 0.97 & S/K < 0.99

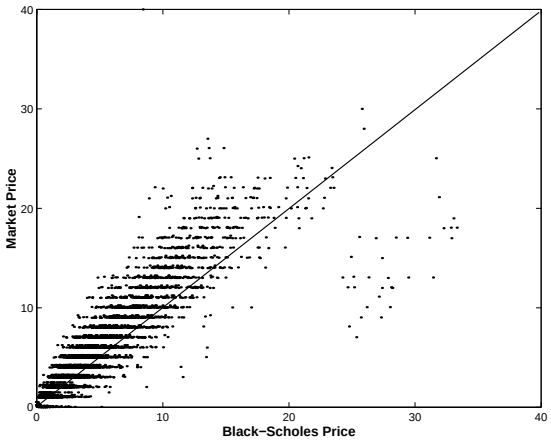
Figure 5: **S&P 500 Call Prices versus Time-to-Maturity**. The market/exercise price (C/K) is plotted against time-to-maturity (τ) for out-of-the-money call options. There are three cases, namely, the deepest out-of-the-money call options ($S/K < 0.95$), deeper out-of-the-money call options ($S/K \geq 0.95$ & $S/K < 0.97$) and the near out-of-the-money ($S/K \geq 0.97$ & $S/K < 0.99$) call options. There is a positive, but quite noisy, relationship between C/K and τ for the near out-of-the-money call options. As it is moved to deeper out-of-the-money call options, the relationship becomes noisier with apparent outliers, and there is hardly any obvious functional relationship for the deepest out-of-the money options. This figure illustrates the difficulty of estimating the price of out-of-the-money call options due to the nature of the outliers and the noise in the empirical data.



(a) S/K < 0.95

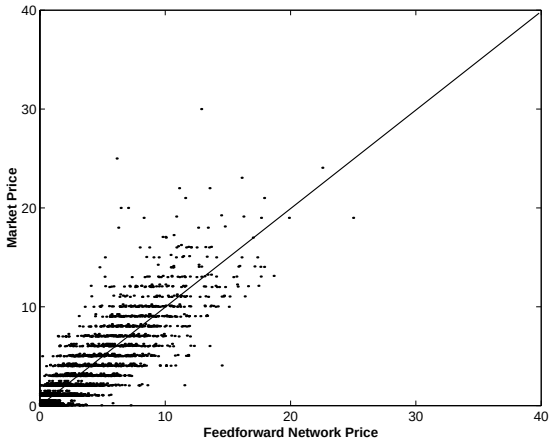


(b) S/K ≥ 0.95 & S/K < 0.97

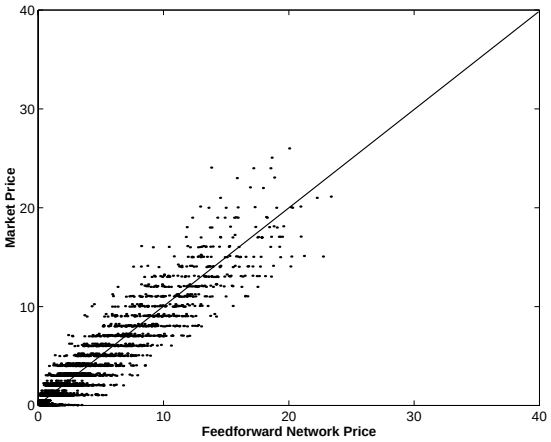


(c) S/K ≥ 0.97 & S/K < 0.99

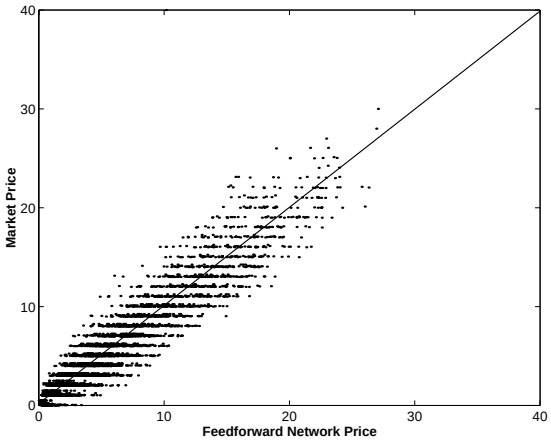
Figure 6: **S&P 500 Call Prices versus Black-Scholes Call Price.** This figure depicts the relationship between the market and Black-Scholes prices. The first observation is that the Black-Scholes prices are biased estimates of the market prices. For the deepest out-of-the-money options, the Black-Scholes prices overestimate market prices whereas market prices are underestimated for the deeper and the near out-of-the money options. In particular, the performance of the Black-Scholes model in explaining the observed market prices is quite poor for the deepest out-of-the-money options.



(a) S/K < 0.95

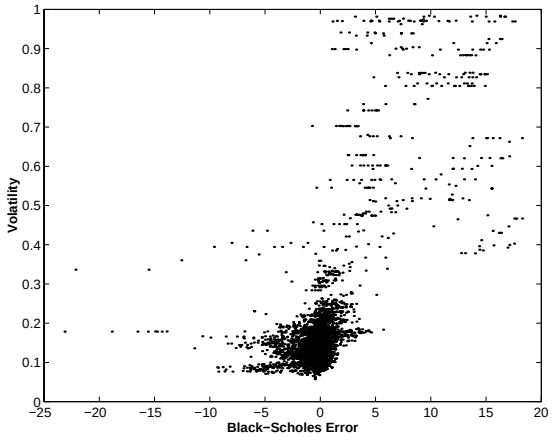


(b) S/K ≥ 0.95 & S/K < 0.97

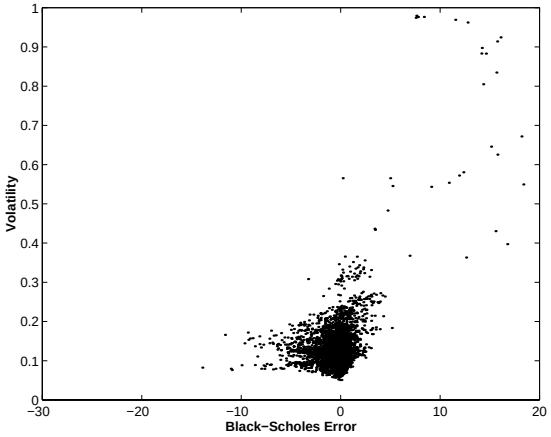


(c) S/K ≥ 0.97 & S/K < 0.99

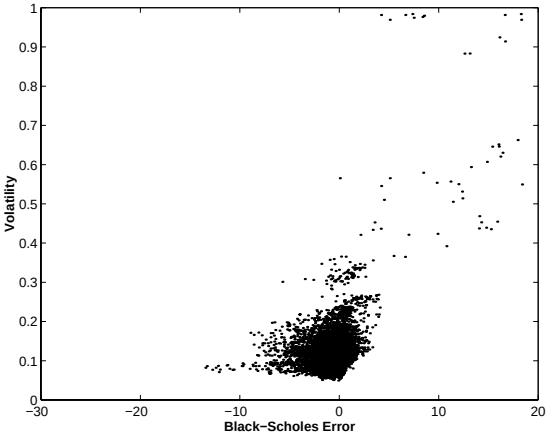
Figure 7: **S&P 500 Call Prices versus Feedforward Network Call Price.** The relationship between the market and the feedforward network prices is presented. The feedforward networks largely eliminate the overestimation bias observed in Figure 6 with the Black-Scholes model. The estimated deepest out-of-the-money call options are centered around the 45-degree line with substantially less and smaller outliers. The performance for the deeper and the near out-of-the-money call options are also substantially improved without any evidence of underestimation bias and lack of outliers. The comparison of Figures 6 and 7 indicates that the feedforward network corrects under and overestimation bias of the Black-Scholes model successfully. Since both feedforward network and the Black-Scholes model use identical inputs, the gain from the feedforward network originates from flexible functional form at which Black-Scholes model may be constraining the data unnecessarily.



(a) S/K < 0.95

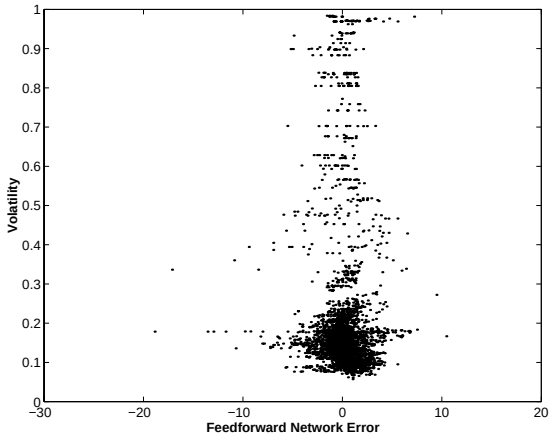


(b) S/K ≥ 0.95 & S/K < 0.97

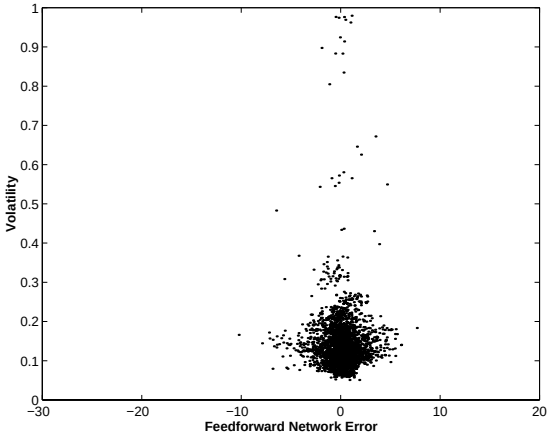


(c) S/K ≥ 0.97 & S/K < 0.99

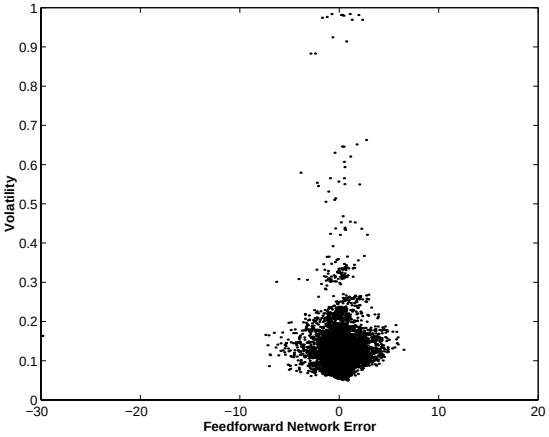
Figure 8: **S&P 500 Call Volatility versus Black-Scholes Pricing Error.** This figure investigates the effect of volatility on the mispricing. Black-Scholes error is the difference between Black-Scholes price and the market price. An ideal model should have ball shaped pricing errors centered around zero at all volatility levels. For extreme volatilities, a desirable pattern is symmetric pricing errors centered around zero. For the deepest out-of-the-money options ($S/K < 0.95$), there are large positive pricing errors for high volatility levels (volatility levels between 0.25 and 1) and pricing errors are hardly symmetric around zero. For low volatility levels, there is a ball-type pricing error around zero although there are a large number of negative errors for volatilities between 0.10 and 0.20. For the deeper out-of-the-money options ($S/K \geq 0.95$ & $S/K < 0.97$), the performance of the Black-Scholes model is more satisfactory, although large positive pricing errors at higher volatility levels and negatively skewed pricing errors at low volatility levels remain. The performance for the near out-of-the-money call options ($S/K \geq 0.97$ & $S/K < 0.99$) is similar to that of the deeper out-of-the-money options.



(a) S/K < 0.95

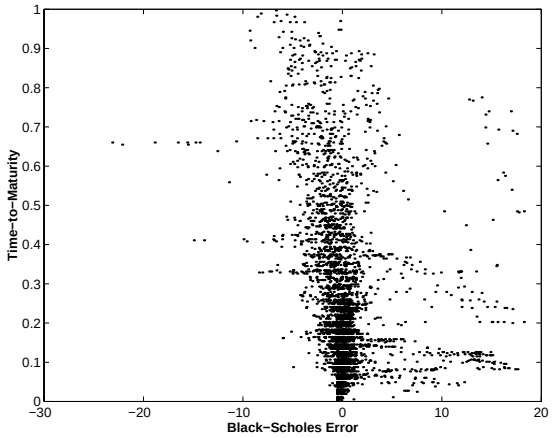


(b) S/K ≥ 0.95 & S/K < 0.97

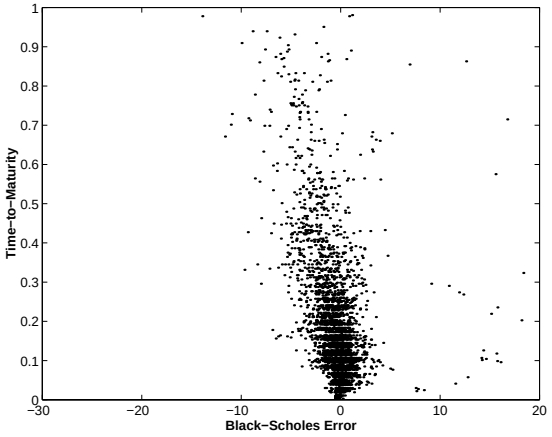


(c) S/K ≥ 0.97 & S/K < 0.99

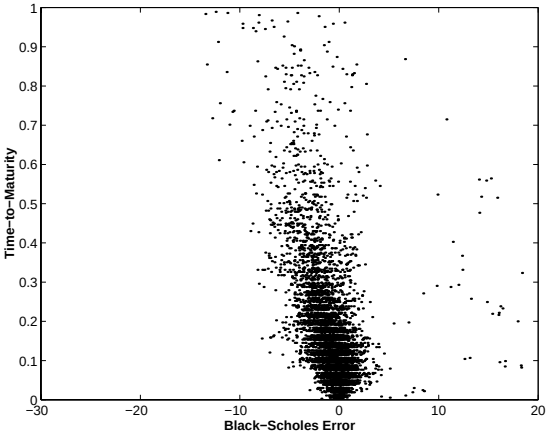
Figure 9: **S&P 500 Call Volatility versus Feedforward Network Pricing Error.** This figure investigates the effect of volatility on the mispricing. Feedforward network error is the difference between the estimated feedforward network price and the market price. An ideal model should have ball shaped pricing errors centered around zero at all volatility levels. For extreme volatilities, a desirable pattern is symmetric pricing errors centered around zero. The results with the deepest out-of-the-money options is the most striking. Large positive pricing errors which were quite dominant in the Black-Scholes model are now largely eliminated such that pricing errors are centered around zero at high volatility levels. The negatively biased pricing errors at the low levels of volatility are also largely corrected. The examinations of the deeper out-of-the money and near out-of-the money options also indicate clear pricing error patterns centered around zero for all levels of volatilities. These figures reveal that when results are examined from the volatility window, a number of results emerge. In particular, feedforward network provides lower bias in terms of the pricing performance relative to the Black-Scholes model.



(a) S/K < 0.95

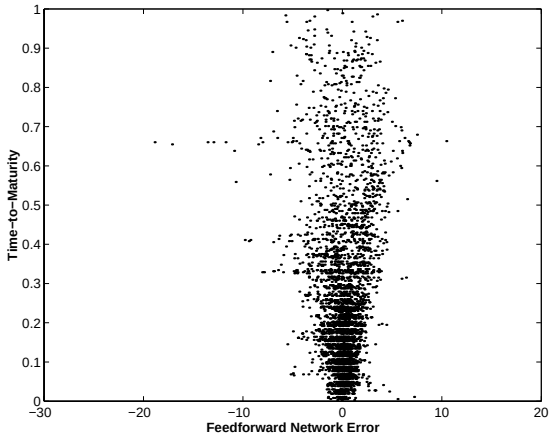


(b) S/K ≥ 0.95 & S/K < 0.97

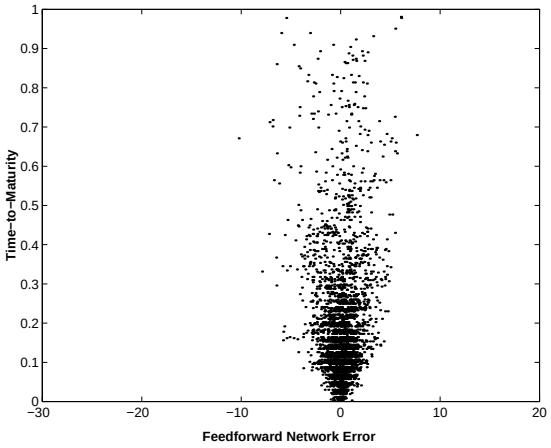


(c) S/K ≥ 0.97 & S/K < 0.99

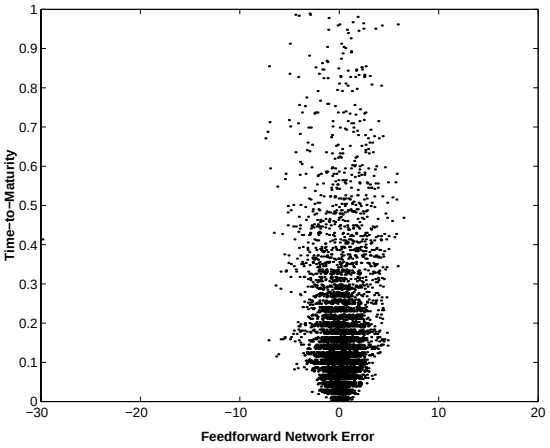
Figure 10: **Time-to-Maturity versus Black-Scholes Error.** An ideal model should exhibit pricing errors centered around zero at all levels of time-to-maturity. This figure illustrates that there are large positive pricing errors at all levels of time-to-maturity for the deepest out-of-the-money call options estimated with the Black-Scholes model. For the deeper out-of-the-money calls and the near out-of-the-money calls, there are negatively slanted pricing errors towards higher levels of time-to-maturity with large positive pricing errors remaining.



(a) S/K < 0.95



(b) S/K ≥ 0.95 & S/K < 0.97



(c) S/K ≥ 0.97 & S/K < 0.99

Figure 11: **Time-to-Maturity versus Feedforward Network Error.** An ideal model should exhibit pricing errors centered around zero at all levels of time-to-maturity. Feedforward networks successfully eliminate large positive pricing errors which were dominating in Figure 10. For all levels of the out-of-the-money calls, the pricing errors are centered around zero with no evidence of bias in either direction and lack of outliers. This other view of the data provides further support for our earlier findings that feedforward networks are invaluable tools for pricing options, in particular for the deepest out-of-the-money calls.