



UOIuBIH
ORSinBIH

Operations Research Society in
Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online: www.scjournal.com.ba



IUS Soft Computing
Research Group

A Nonparametric Approach to Pricing Options Learning Networks

Mehmet Can and Šadi Fadda

International University of Sarajevo, Faculty of Engineering and Natural Sciences, Hrasnicka Cesta 15, Ilidža
71210 Sarajevo, Bosnia and Herzegovina

Article Info

Article history:

Article received on March 2014

Keywords:

Option prices; Neural networks; ANN
applications in finance; Forecasting
option prices using ANN

Abstract

For practitioners of equity markets, option pricing is a major challenge during high volatility periods and Black-Scholes formula for option pricing is not the proper tool for very deep out-of-the-money options. The Black-Scholes pricing errors are larger in the deeper out-of-the-money options relative to the near the-money options, and it's mispricing worsens with increased volatility. Experts opinion is that the Black-Scholes model is not the proper pricing tool in high volatility situations especially for very deep out-of-the-money options. They also argue 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 option implied volatilities were decreasing monotonically as the call went deeper into out-of-the-money, while the put option implied volatilities were decreasing monotonically as the put went deeper into in-the-money. Since these findings cannot be explained by the Black-Scholes model and its variations, researchers searched for improved option pricing models. Feedforward networks provide 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. In this article a nonparametric method for estimating S&P 100 index option prices using artificial neural networks is presented. To show the value of artificial neural network pricing formulas, Black-Scholes option prices are compared with the network prices against market prices. To illustrate the practical relevance of the network pricing approach, it is applied to the pricing of S&P 100 index options from April 4, 2014 to April 9, 2014. On the five days data while Black-Scholes formula prices have a mean \$10.17 error for puts, and \$1.98 for calls, while neural network's error is less than \$5 for puts, and \$1 for calls.

1. INTRODUCTION

Much of the success and growth of the market for options and derivative securities may be traced to the much quoted articles by (Black-Scholes 1973) and (Merton, 1973), in which closed-form option pricing formulas were obtained through a dynamic hedging argument and arbitrage freeness condition. The well-known Black-Scholes and Merton pricing formulas have now been generalized, extended, and applied to such a array of securities and contexts that it is impossible to make an exhaustive list. Moreover, while closed-form expressions all available in many of these generalizations and extensions, pricing formulas may still be obtained numerically.

In each case, the derivation of the pricing formula via the hedging arbitrage approach, either analytically or numerically, depends intimately on the particular parametric form of the underlying asset's price dynamics $S(t)$. A misspecification of the stochastic process for $S(t)$ will lead to thematic pricing and hedging errors for derivative securities linked to. Therefore, the success or failure of the traditional approach to pricing and hedging derivative securities, which is called a parametric pricing method, is closely tied to the ability to capture the dynamics of the underlying asset's price process.

Therefore the failure of Black-Scholes pricing formulas in predicting correct option prices is due to the inappropriateness of the distributional assumptions behind the Black-Scholes model (Black-Scholes, 1973). These assumptions behind the Black-Scholes model have been investigated extensively. Black (Black, 1976) found out that in the early years of trading on the Chicago Board of Trade, implied volatilities tended to increase with increasing strike price. On the other hand Macbeth and Merville (Merville, 1979) revealed that the calculated with the implied volatility, Black-Scholes prices of at-or near-the-money options, are on average less than market prices for in-the-money call options. While they are on average greater than market prices for out-of the-money call options.

Moreover the extent to which the Black-Scholes model overprices an in-the-money option increases with the extent to which the option is in-the-money and decreases as time-to-maturity decreases. The extent to which it overprices an out-of-the-money option increases with the extent to which the option is out-of-the-money and decreases as time-to-maturity decreases.

It means that the implied volatilities are inversely related to the exercise price, and this fact is contrary to Black's (Black, 1976) results. According to Macbeth and Merville (Macbeth and Merville 1979), these results might be due to variable variance of the underlying distribution of asset returns. To Rubinstein (Rubinstein, 1985), strike price bias is statistically significant, but the direction of the bias changes from period to period. Since these findings cannot be explained by the Black-Scholes model and its variations, researchers tried to find new paradigms for

more efficient option pricing models (Lo et al, 1993; Merton 1976).

Black-Scholes pricing formula's appeal to practitioners often originates from its analytical simplicity to determine the price of a European call options c , and puts p on a non-dividend paying asset by

$$c = SN(d_1) - Ke^{-rT}N(d_2) \tag{1}$$

$$p = -SN(-d_1) + Ke^{-rT}N(-d_2) \tag{2}$$

with

$$d_1 = \left(\text{Ln}\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T \right) / \sigma\sqrt{T}, \text{ and} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

where N is the cumulative normal distribution, S is the price of the underlying security, K is the strike price, r is the prevailing risk-free interest rate, T is the time-to-maturity and σ is the volatility of the underlying asset. (1) - (3) do not contain neither preferences of individuals nor the preferences of the aggregate market (Hull, 1993).

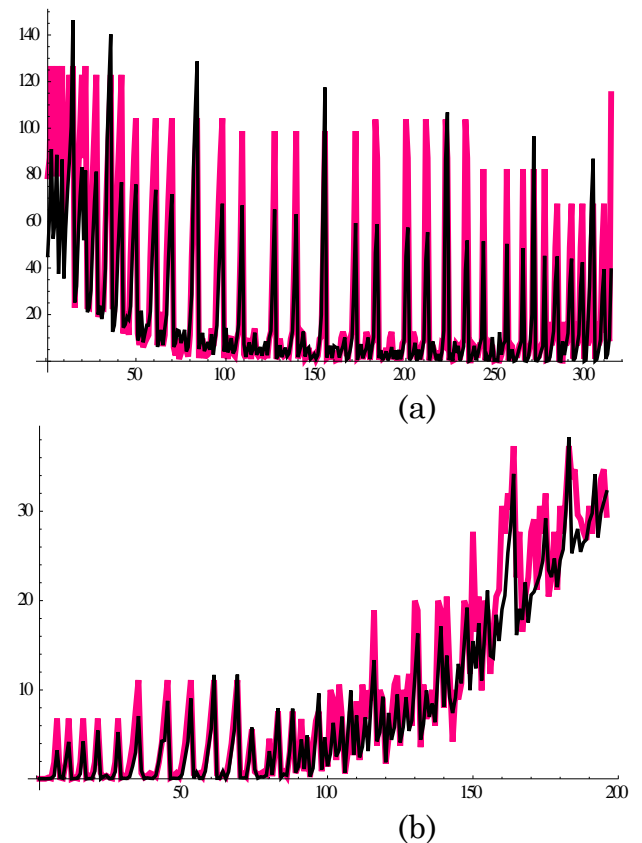


Figure 1. Option prices using Black-Scholes formula (black/dark), and market realization (red/light); put options (a), call options (b).

Table 1

Mean absolute differences between option prices computed using Black-Scholes formula and market realizations. Mostly puts are underpriced, and calls are overpriced by the Black-Scholes formula.

	Puts	Calls
MAE	\$10.17	\$1.98

Black-Scholes derivation has been mostly criticized for its distributional assumptions of the underlying security. Empirical 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. (Oldfield et al., 1977), Rosenfeld (Rosenfeld, 1980), and Ball and Torous (Ball, and Torous 1985) have fitted mixtures of continuous and jump processes to the stock price data. Black (Black, 1976), Beckers (Beckers, 1980), and Christie (Christie, 1982) document negative correlation between stock prices and volatility.

Schmalensee and Trippi (Schmalensee and Trippi, 1978) found that changes in implied volatilities are negatively correlated with changes in stock prices. Blattberg and Gonedes (Blattberg and Gonedes, 1974) conclude that volatility is a random process through time. Attempts to accommodate 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 (Bakshi, et. al, 1997) provide closed form solutions for valuing options under stochastic volatility and stochastic interest rates using Heston's (Heston, 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 (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 (Derman and Kani, 1994a,b), Dupire (Dupire, 1994) and Rubinstein (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 (Dumas et. al., 1998) report that the DVF option valuation model's fit is no better than an ad hoc procedure that merely smoothes 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

$$C = f(S, K, s, r, T) \quad (4)$$

where S is the price of the underlying asset, K is the strike price, s is the volatility of the underlying asset, r is the interest rate and T 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 (Lo, and Poggio, 1994) divide the function and its arguments by K and write the pricing function as follows:

$$\frac{c}{K} = f\left(\frac{s}{K}, 1, s, r, T\right) \quad (5)$$

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 is nonstationary while the variable S/K is stationary as strike prices bracket the underlying asset price process. This paper uses (5) as the nonparametric model for feedforward network estimation.

To relax the distributional assumptions of the Black-Scholes model, nonparametric valuation models are a natural extension. In this paper, the feedforward network models will be used for option pricing.

Indeed recently, a number of papers have used nonparametric methods to price options. Ghysels (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 (Gouriéroux et al., 1994) apply a Kernel M-estimator methodology to the option pricing problem by extending the Black-Scholes formulation. 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. Ait-Sahalia and Lo (Ait-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. Hutchinson, Lo and Poggio (Hutchinson et al., 1994) investigate several techniques for pricing and hedging options nonparametrically with radial basis functions, projection pursuit regression, and feedforward networks. Gençay and Garcia (Gençay and Garcia, 2000), Gençay and Salih (Gençay and Salih, 2001), (İltüz Samur, and Tekin Temur, 2009) 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 (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; foresee et al., 1997), early stopping, and bagging (Breiman, 1996) which we use here to estimate parsimonious models. Our results indicate that bagging is a robust network selection method 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 in Section 5.

2. Nonparametric Option Pricing

In this section, a nonparametric pricing model is presented. In this approach, the data is allowed to determine both the dynamics of $S(t)$ and its relation to the prices of derivative securities with minimal assumptions on $S(t)$. The primary economic variables that influence the derivative's price are taken as inputs, for example, rent fundamental asset price, strike price, time-to-maturity, volatility of underlying asset, and risk free interest rate, and the derivative price to be the output into which the learning network maps the inputs into outputs. When properly trained, the network "becomes" the derivative pricing formula, which may be used in the same way formulas obtained from the parametric pricing method are used: for pricing, delta-hedging, simulation exercises, and so on.

These network-based models have several important advantages the more traditional parametric models. First, since they do not rely on restrictive parametric assumptions such as lognormality or sample continuity, they are robust to the specification errors that troubles metric models. Second, they are adaptive and respond to structural changes in the data-generating processes in ways that parametric methods cannot. Finally, they are flexible enough to encompass a wide range of derivative securities and fundamental asset price dynamics, yet they are relatively simple to implement.

Of course, all these advantages do not come without some cost: The nonparametric pricing method is highly data-intensive, requiring large quantities of historical prices to obtain a sufficiently well-trained network. Therefore, such an approach would be inappropriate for thinly traded derivatives, or newly created derivatives that have no similar counterparts among existing securities. Also, if the fundamental asset's price dynamics are well-understood and an analytical expression for the derivative's price is available under these dynamics, the parametric formula will almost always dominate the network formula in pricing and hedging accuracy. Nevertheless, these conditions occur rarely enough that there may still be great

practical value in constructing derivative pricing formulas by learning networks.

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 (Ng, and Lippman 1991).

Amongst nonlinear methods, neural network is one of the most recent techniques used in nonlinear modelling. This is partly due to some modeling problems encountered in the early stage of development within the neural networks field. In the earlier literature, the statistical properties of neural networks estimators and their approximation capabilities were questionable (Friedman et al., 1981). 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 (Grosi et al., 1990, 1992, 1993). The results in Cybenko (Cybenko, 1989), Funahashi (Funahashi, 1989), Hornik et al. (Hornik et al., 1989, 1990), and Hornik (Hornik, 1989, 1991) indicate that feedforward networks with sufficiently many hidden units and properly adjusted parameters can approximate an arbitrary function arbitrarily well (Cybenko, 1981; Diaconis et al., 1984). Hornik et al. (Hornik et al., 1990) and Hornik (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 modeling (Barron, 1991, 1998). In options pricing, for instance, Hutchinson et al. (Hutchinson et al., 1994) and Garcia and Gençay (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 (Garcia and Gençay, 2000) and Gençay and Qi (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 networks are located is called the output layer (Parker, 1995). 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 (Haykin 1999).

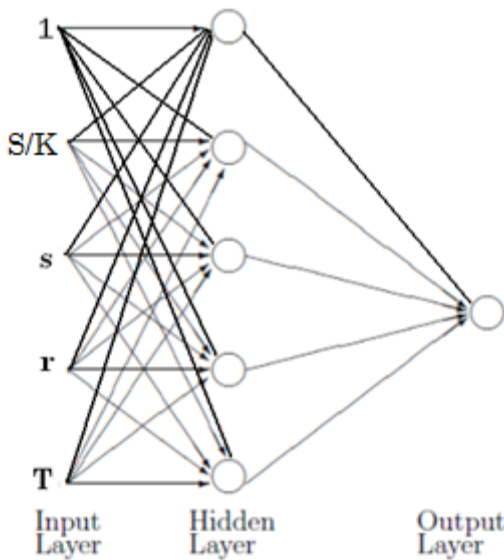


Figure 2: A five input and five hidden neuron in a single hidden layer feedforward network

An example of a single hidden layer feedforward network is presented in Figure 2. This figure demonstrates a five input and five hidden neuron in a single hidden layer feedforward network, where $x = (S/K, s, r, T)$ are the inputs at time t ;

$$W_{1i} = (w_{1i1}, w_{1i2}, w_{1i3}, w_{1i4}, w_{1i5}), i = 1, \dots, 5 \quad (6)$$

synaptic weights, that are the parameters of the first activation function and

$$W_2 = (w_{21}, w_{22}, w_{23}, w_{24}, w_{25}) \quad (7)$$

are the synaptic weights of the second activation function. First weights in all sets are intercepts and the others are slope parameters.

The numbers created at the five hidden neurons are

$$v_i = g(w_{1i1}) + g\left(w_{1i2} \frac{S}{K}\right) + g(w_{1i3}r) + g(w_{1i4}s)$$

$$+ g(w_{1i5}T), \quad i = 1, \dots, 5 \quad (8)$$

Where the activation function $g(z)$ is

$$g(z) = Tanh(z). \quad (9)$$

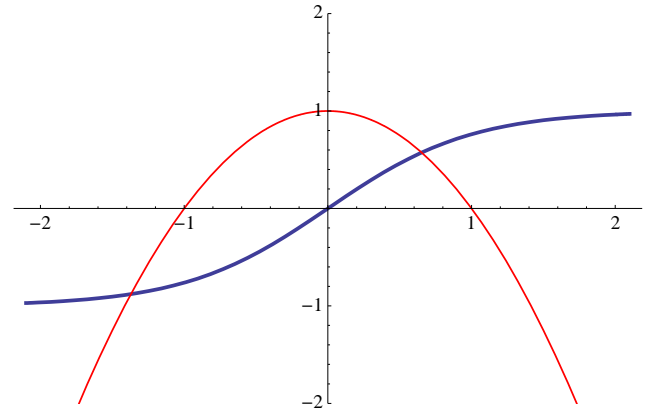


Figure 3: Activation function $g(z) = Tanh(z)$ (thick), and its derivative $g'(z) = (1 + z) (1 - z)$ (thin).

If the synaptic weights are tuned properly, the number accumulated at the output node will be the call option price corresponding the input set of data:

$$y = w_{21}g(v_1) + w_{22}g(v_2) + w_{23}g(v_3) + w_{24}g(v_4) + w_{25}g(v_5) \quad (10)$$

The underlying functional form $f(x, \theta)$ is a network output which depends on the inputs and the network parameters. The x 's

$$x = \left(1, \frac{S}{K}, r, s, T\right) \quad (11)$$

here represents a vector of all inputs at time t and the symbol θ represents the vector of parameters, W_1 's and W_2 '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 (Niyogi, and Grosi, 1994; Poggio, and Grosi, 1990).

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 sigum 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 linear, sigmoidal or hyperbolic tangent activation functions can be used in an output layer as seen in Figure 3.

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 five inputs and five hidden units is presented in Figure 2.

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 units) used in a network is constrained by the available degrees of freedom, 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 x and y be the input (regressors) and the target (regressand) vectors with dimensions $1 \times n$ and $1 \times m$. The observations for a sample size N are denoted by $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ and $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$. Given inputs $\mathbf{x} = (x_1, x_2, \dots, x_n)$, a single layer feedforward network regression model with q hidden units is written as

$$\mathbf{y} = F(\mathbf{w}_{21} + \sum_{i=2}^q \mathbf{w}_{2i} \mathbf{h}_i) \quad (12)$$

$$\mathbf{h}_i = G(\mathbf{w}_{1i0} + \sum_{j=1}^n \mathbf{w}_{1ij} \mathbf{x}_j), \quad i = 1, \dots, q. \quad (13)$$

or

$$\mathbf{y} = F\left(\mathbf{w}_{21} + \sum_{i=2}^q \mathbf{w}_{2i} G\left(\mathbf{w}_{1i1} + \sum_{j=2}^n \mathbf{w}_{1ij} \mathbf{x}_j\right)\right) = f(\mathbf{x}, \boldsymbol{\theta}) \quad (14)$$

where F, G are known activation functions; and the parameters to be estimated are

$$\mathbf{W}_{1i} = (w_{1i1}, w_{1i2}, w_{1i3}, w_{1i4}, \dots, w_{1in}), \quad i = 1, \dots, 5 \quad (15)$$

and

$$\mathbf{W}_2 = (w_{21}, w_{22}, w_{23}, w_{24}, w_{2q}) \quad (16)$$

The range of the output values of the feedforward network model is controlled by F such that if the output takes discrete values, then F 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 $F(x) = x$. In a typical neural network model, F is normally an identity function.

Given the network structure in (16) and the chosen functional forms for F and G , a major empirical issue in the neural networks is to estimate the unknown parameters $\boldsymbol{\theta}$ with a sample of data values. A recursive estimation methodology, which is called back propagation, is such a method to estimate the underlying parameter vector $\boldsymbol{\theta}$ from data (Rumelhart et al., 1986).

In back propagation, the starting point is a random weight $\boldsymbol{\theta}$ vector that is updated according to

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta} + \eta \nabla f(\mathbf{x}, \boldsymbol{\theta})(y - f(\mathbf{x}, \boldsymbol{\theta})) \quad (17)$$

where $\nabla f(\mathbf{x}, \boldsymbol{\theta})$ is the column gradient vector of f with respect to $\boldsymbol{\theta}$, 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 $y - f(\mathbf{x}, \boldsymbol{\theta})$.

By imposing appropriate conditions on the learning rate and functional forms of F and G , White (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. Backpropagation and nonlinear regression can be seen as alternative statistical methods to solve the least squares problem (Broomhead et al., 1989; Chen 1991).

Compared to nonlinear least squares, back propagation 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 (White, 1989) further shows that the recursive estimator is not as efficient as the nonlinear least squares (NLS) estimator. One important aspect of the Backpropagation methods is the choice of the learning rate η . The inefficiency of the Backpropagation originates from keeping the learning rate constant in an environment where the influences of random movements in x are not accounted for in y . This would lead the parameter vector $\boldsymbol{\theta}$ to fluctuate indefinitely. A minimum requirement is to drive the learning rate gradually to zero to achieve convergence (Moody, and Parker, 1989).

In fact, White (White, 1989) demonstrates that η 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 (White, 1989).

In this paper the nonlinear least square estimator (NLS) is used. This estimator minimizes the cost function

$$L(\theta) = \sum_{t=1}^N (y_t - f(x_t, \theta))^2 \tag{18}$$

Here, the goal is to choose the parameter vector θ such that the sum of squared errors is 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 (Gençay and Dechert, 1992) work very well for this problem. In Gallant and White (Gallant and White, 1992), it is shown that the least squares method can consistently estimate a function and its derivatives from a feed-forward 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 over fitting in noisy environments.

2.2. Network Selection

To decide about the network architecture, networks with several hidden layers and several hidden neurons in each hidden layer are tried. It is seen that the complexity/accuracy balance is with five hidden neurons in one hidden layer.

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

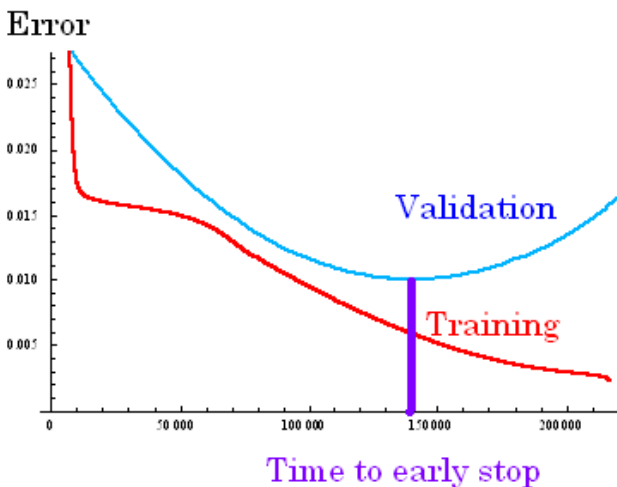


Figure 4: Early stopping method. The validation error will normally decrease during the initial phase of training, as does the error on the training set. However, when the

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.

The onset of over fitting 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.

2.2.2. Bootstrap Aggregating

In 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 (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

$$L = \{(y_i, x_i), i = 1, \dots, N_L\}, \tag{19}$$

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 $f(x_i, L)$, e.g., if the input is (y_i, x_i) is predicted by $f(x_i, L)$. Now, suppose we have a sequence of training sets $\{L_j, j = 1, \dots, J\}$ each consisting of N_L independent observations from the same underlying distribution as L . We can use the $\{L_j\}$ to get a better predictor than the single learning set predictor $f(x_i, L)$ by working with the sequence of predictors $\{f(x_i, L_j)\}$.

An obvious procedure is to replace $f(x_i, L)$ by the average of $\{f(x_i, L_j)\}$. over J , i.e.,
by

$$f(\mathbf{x}_i) = \frac{1}{J} \sum_{j=1}^J f(\mathbf{x}_i, L_j) \quad (20)$$

However, usually there is only a single training set L without the luxury of replicates of L . In this case, repeated bootstrap samples can be drawn from

$$L = \{(y_i, \mathbf{x}_i), i = 1, \dots, N_L\},$$

Each $\{(y, \mathbf{x})\}$ is a random pick from the original training set L with replacement. The bootstrap samples $L^{(b)}$ are used to form predictors $\{f(\mathbf{x}_i, L^{(b)})\}$. The bagging predictor f_B can thus be calculated as

$$f_B(\mathbf{x}_i) = \frac{1}{B} \sum_{b=1}^B f(\mathbf{x}_i, L^{(b)}) \quad (21)$$

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

We slightly modify the bagging procedure of Breiman (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 4 hidden layers. The validation set is used to select the best feedforward network that has the optimal number of hidden layers, and the best model is used to generate one set of prediction on the testing set.

For put options whole data set consists of 314 data. 100 of them reserved as the testing data. From the remaining 214, 7 samples of size 100 is chosen randomly with replacement.

For calls data set consists of 196 data. 70 of them reserved as the testing data. From the remaining 126, 7 samples of size 70 is chosen randomly with replacement.

This gave seven predictions ($B = 7$). Third, the bagging prediction is the average across the seven sets of predictions, and the prediction error is computed as the absolute difference between the actual and the bagging prediction values.

3. Data Description

The data are daily S&P 100 index five days obtained from the Chicago Board of Exchange for the period April 3-8 2014. 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 100 index options are very popular among

institutional investors as portfolio insurance instruments. For each option written on the S&P 100 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 100 closing index, the number of days to maturity, daily S&P 100 returns, dividend yields and the interest rate at the maturity of the option.

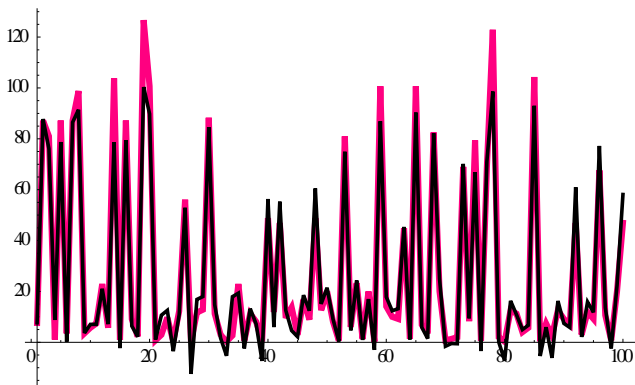
In constructing the data used in the estimation, options with zero volume are not used. Put-call parity checks are done to eliminate erroneous prices, therefore a put is only included if there is a call with the same exercise price trading at that particular date, while a call is only included if there is a put with the same exercise price. For Black-Scholes price calculations, historical volatilities are calculated using the daily S&P 100 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.

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 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.

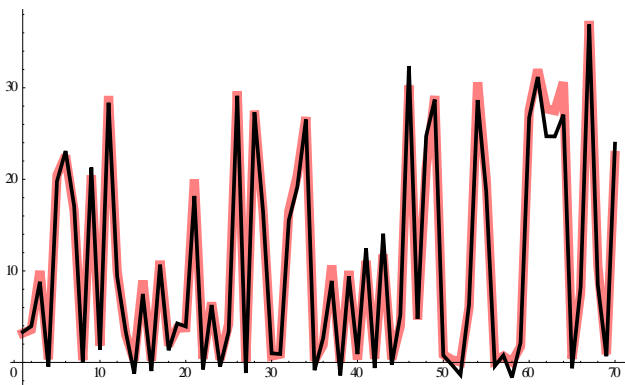
4. Empirical Findings

Variables $\mathbf{x} = \left(\frac{S}{K}, r, s, T\right)$ are entered into artificial neural networks in two different versions of the volatility \mathbf{s} . a) Implied volatility computed from Black-Scholes formula, b) Implied volatility computed by the use of ANN's (Can, and Fadda, 2014), c) Historical volatility.

Figure 5. and Table 2. depicts the relationship between the prices computed by artificial neural networks trained with implied volatility, and market prices. The first observation is that the ANN prices are biased estimates of the market prices. ANN prices underestimate market prices. But, compared to the Black-Scholes prices in Figure 1., the mispricing is halved when ANN's are used.



(a)



(b)

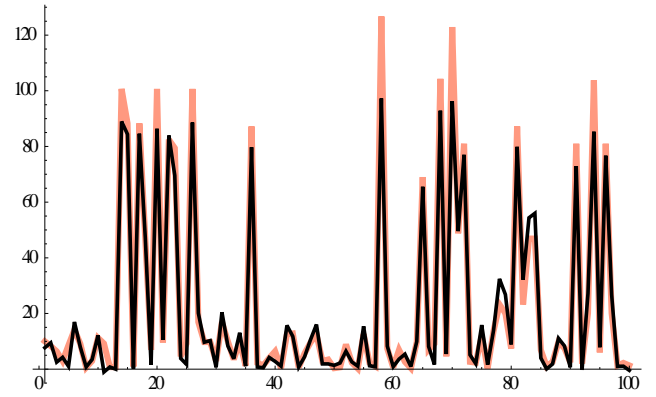
Figure 5. Option prices using ANN's with implied volatility (black), and market realization (red); call options (a), put options (b).

Table 2

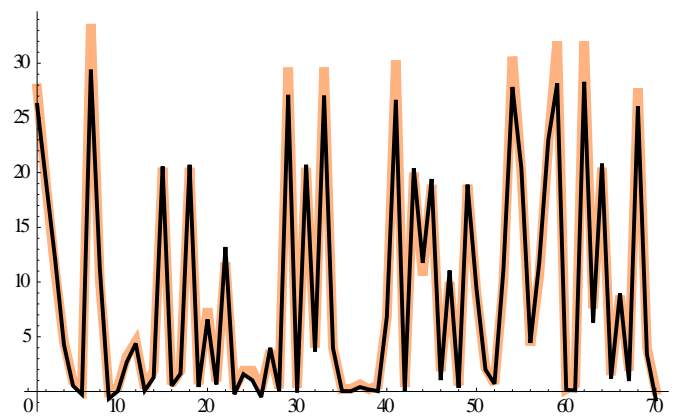
Mean absolute differences (MAE) between option prices computed using ANN with implied volatility and market realizations and between option prices computed using Black-Scholes price formula and market realizations

	Puts	Calls
ANN	\$4.90	\$0.95
B-S	\$10.17	\$1.98

When ANN implied volatilities are used, there is some gain, although it is not as noticeable as in case of implied volatilities. Figure 6. and Table 3. depicts the relationship between the prices computed by artificial neural networks trained with ANN computed volatility, and market prices. ANN prices underestimate market prices. But, compared to the Black-Scholes prices in Figure 1., the mispricing is dramatically reduced when ANN's are used together with volatilities computed by ANN's.



(a)



(b)

Figure 6. Option prices using ANN's with ANN volatility (black), and market realization (red); call options (a), put options (b).

Table 3

Mean absolute differences (MAE) between option prices computed using ANN with ANN volatility and market realizations and between option prices computed using Black-Scholes price formula and market realizations

	Puts	Calls
AN Networks	\$3.70	\$0.72
Black-Scholes	\$10.17	\$1.98

When historical volatilities are used, there is some gain, although it is not as noticeable as in case of implied volatilities. Figure 7. and Table 4. depicts the relationship between the prices computed by artificial neural networks trained with historical volatility, and market prices. ANN prices underestimate market prices, but, compared to the Black-Scholes prices in Figure 1., in put options, the mispricing is reduced, although less than the case of implied volatilities. When ANN's are used with historical volatilities the mispricing for call options worsens.

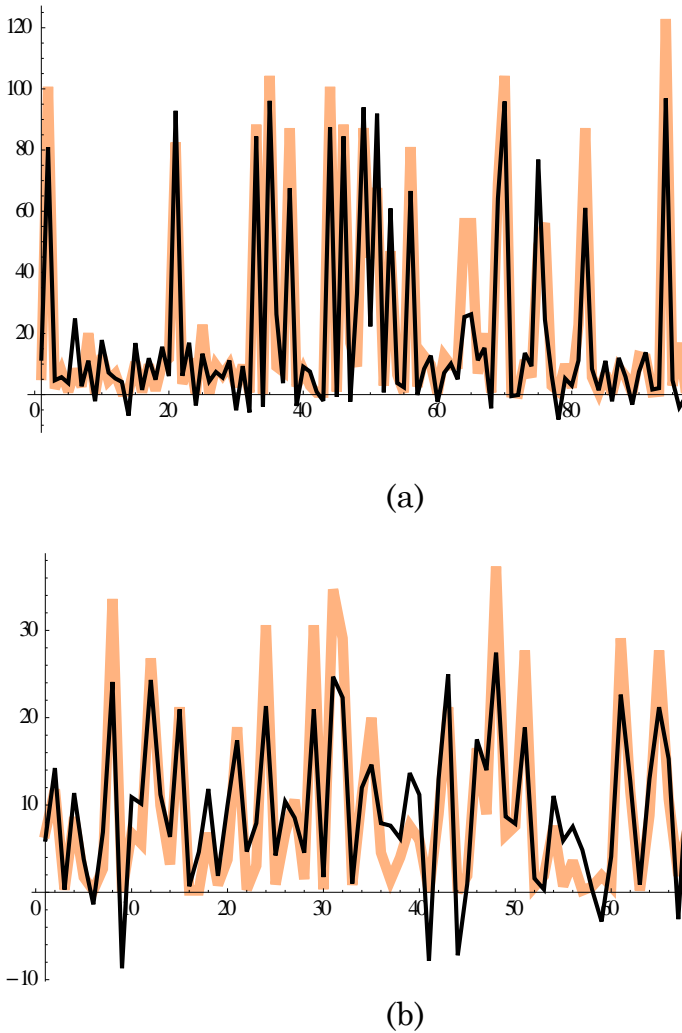


Figure 7. Option prices using ANN's with historical volatility (black), and market realization (red); call options (a), put options (b).

Table 4
Mean absolute differences (MAE) between option prices computed using ANN with historical volatility and market realizations.

	Puts	Calls
AN Networks & hist. volatility	\$7.60	\$3.99
Black-Scholes & hist. volatility	\$10.17	\$1.98

5. Conclusions

To investigate the effect of volatility on the mispricing, we analyze the relationship between pricing errors and volatility. When prices are computed by artificial neural networks trained with implied volatility, Black-Scholes

misprices are halved. When ANN computed volatilities are used, mispricing levels are dramatically reduced for both puts, and calls comparable with the case of implied volatilities. When ANN's are used with historical volatilities the mispricing is reduced for puts, but it is worsened for call options as seen in Table 5. This fact shows that to get more acceptable prices for options, first new approaches to detect the volatility of the underlying assets is needed.

Table 5
Mean absolute differences (MAE) between option prices computed using ANN with implied volatility and market realizations and between option prices computed using Black-Scholes price formula and market realizations

	Puts	Calls
AN Networks & ANN volatility	\$3.70	\$0.72
AN Networks & implied volatility	\$4.90	\$0.95
AN Networks & historical volatility	\$7.60	\$3.99
Black-Scholes & historical volatility	\$10.17	\$1.98

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.

Although parametric derivative pricing formulas are preferred when they are available, our results show that nonparametric learning-network alternatives can be useful substitutes when parametric methods fail. While our findings are promising, we cannot yet claim that our approach will be successful in general. For simplicity, our simulations have focused only on the Black-Scholes model, and our application has focused only on a single instrument and time period, S&100 futures options for April 3-8 2014. In particular, there are a bunch of parametric derivative pricing models, as well as many practical extensions of these models that may improve their performance on any particular data set. We hope to provide a more comprehensive analysis of these alternatives in the near future.

However, we do believe there is reason to be cautiously optimistic about our general approach, with a number of promising directions for future research. Perhaps the most pressing item on this agenda is the specification of additional inputs, inputs that are not readily captured by parametric models such as the return on the market, general market volatility, and other measures of business conditions. A related issue is the incorporation of the predictability of the underlying asset's return, and cross-predictability among several correlated assets (Lo and Wang 1993). This may involve the construction of a factor

model of the underlying asset's return and volatility processes.

Other research directions are motivated by the need for proper statistical inference in the specification of learning networks. First, we require some method of matching the network architecture; number of nonlinear units, number of centers, type of basis functions, to the specific dataset at hand in some optimal and, preferably, automatic fashion.

Second, the relation between sample size and approximation error should be explored, either analytically or through additional Monte Carlo simulation experiments. Perhaps some data-dependent metric can be constructed, such as the model prediction error that can provide real-time estimates of approximation errors in much the same way that standard errors may be obtained for typical statistical estimators.

And finally, the need for better performance measures is clear. While typical measures of goodness-of-fit such as R^2 do offer some guidance for model selection, they are only incomplete measures of performance. Moreover, the notion of degrees of freedom is no longer well-defined for nonlinear models, and this has implications for all statistical measures of fit.

References

- Aït-Sahalia, Y. and A. Lo (1998), Nonparametric Estimation of State-Price Densities Implicit in Financial Asset Prices, *Journal of Finance*, 53, 499-547.
- 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.
- Akaike, H. (1974), A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*, 19, 716-723.
- Bakshi, G., C. Cao, and Z. Chen (1997), *Empirical Performance*
- 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.
- Barron, A.R. (1991) Universal approximation bounds for superpositions of a sigmoidal function. Technical Report 58, Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, Marc.
- Barron, A.R., and R.L. Barron (1988) Statistical learning networks: a unifying view. In *20th Symposium on the Interface: Computing Science and Statistics*, pages 192-203.
- Beckers, S. (1980), The Constant Elasticity of Variance Model and its Implications for Option Pricing, *Journal of Finance*, 35, 661-673.
- Black, F., (1976), Studies of Stock Price Volatility Changes, *Proceedings of the 1976 Meetings of the American Statistical Association*, 177-181.
- Black, F., and M. S. Scholes (1973), The Pricing of Options and Corporate Liabilities, *Journal of Political Economy*, 81, 637-659.
- 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.
- Breiman, L. (1996), Bagging Predictors, *Machine Learning*, 24, 123-140.
- Broomhead, D.S., and D. Lowe (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2: 321 - 355.
- Chen, H., (1991) Estimation of a projection-pursuit type regression model. *Ann. Statistics*, 19:142-157.
- Christie, A. (1982), The Stochastic Behavior of Common Stock Variances: Value, Leverage, and Interest Rate Effects, *Journal of Financial Economics*, 10, 407-432.
- Cybenko, G. (1989), Approximation by Superposition of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 2, 303-314.
- Cybenko, O., (1988) Approximation by superpositions of a sigmoidal function. Technical Report 856, University of Illinois, Dept. of Electrical and Computer Engineering.
- Derman, E., and I. Kani (1994a), The Volatility Smile and its Implied Tree, *Quantitative Strategies Research Notes*, Goldman Sachs, New York.
- Derman, E., and I. Kani (1994b), Riding on the Smile, *Risk*, 7, 32-39.
- Diaconis, P., and M. Shahshahani (1984) On nonlinear functions of linear combinations. *SIAM J. Sci. Stat. Comput.*, 5(1):175 -191.
- Diebold, F., and R. Mariano (1995), Comparing Predictive Accuracy. *Journal of Business and Economic Statistics* 13, 253-263.
- Donoho, D.L., and I. Johnstone (1989) Projection-based approximation and a duality with kernel methods. *Ann. Stat.*, 17:58-106.
- Dumas, B., J. Fleming, and R. E. Whaley (1998), Implied Volatility Functions: Empirical Tests, *Journal of Finance*, 53, 2059-2106.
- Dupire, B. (1994), Pricing with a Smile, *Risk*, 7, 18-20.
- 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.

- Friedman, J.H., and W. Stuetzle (1981) Projection pursuit regression. *Journal of the American Statistical Association, Theory and Methods Section*, 76(376), December.
- Funahashi, K. (1989), On the Approximate Realization of Continuous Mappings by Neural Networks, *Neural Networks*, 2, 183–192.
- Gallant, A., and H. White (1992) On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Networks*, 5:128-138.
- Garcia, R., and R. Gençay (2000), Pricing and Hedging Derivative Securities with Neural Networks and a Homogeneity Hint, *Journal of Econometrics*, 94, 93–115.
- 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.
- 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.
- Gençay, R., and A. Salih (2001) Degree of Mispricing with the Black-Scholes Model and Nonparametric Cures, Technical Report.
- Ghysels, E., V. Patilea, E. Renault, and O. Torr s (1997), Nonparametric Methods and Option Pricing, *Statistics and Finance, D. Hand and S. Jacka (eds.)*, Edward Arnold, London, Ch. 13, 261-282.
- Girosi, F., and G. Anzellotti (1992) Rates of convergence of approximation by translates. A.I. Memo 1288, Massachusetts Institute of Technology Artificial Intelligence Laboratory.
- Girosi, F., M. Jones, and T. Poggio (1993) Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. *Artificial Intelligence Memo 1430*, Massachusetts Institute of Technology.
- Girosi, F., and T. Poggio (1990) Networks and the best approximation property. *Biological Cybernetics*, 63:169 - 176.
- Gouri roux, C., A. Monfort and C. Tenreiro (1994), Kernel M-Estimators: Non-parametric Diagnostics for Structural Models, Working Paper 9405, CEPREMAP, Paris.
- Haykin, S. (1999), *Neural Networks*, Prentice Hall, New Jersey, Second edition.
- 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.
- Hornik, K. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359-366.
- Hornik, K., M. Stinchcombe, and H. White (1990) Universal approximation of an unknown mapping and its derivatives. *Neural Networks*, 3:551-560.
- Hornik, K. (1991), Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, 4, 251–257.
- Hornik, K., Stinchcombe, M., and H. White (1989), Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, 359–366.
- 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.
- Hull, J. C. (1993) *Options, Futures, and Other Derivative Securities*. Prentice-Hall, Englewood Cliffs, New Jersey, 2nd edition.
- 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.
- Ilt zer Samur, Z. and G. Tekin Temur (2009) The Use of Artificial Neural Network in Option Pricing: The Case of S&P 100 Index Options, *World Academy of Science, Engineering and Technology* 54.
- Lo, A. and J. Wang (1993) Implementing option pricing models when asset returns are predictable. *Research Program in Computational Finance Working Paper RPCF-1001-93*, MIT Sloan School of Management.
- Macbeth, J. D., and L. J. Merville (1979), An Empirical Examination of Black-Scholes Call Option Pricing Model, *Journal of Finance*, 34, 1173-1186.
- MacKay, D. J. C. (1992), Bayesian Interpolation, *Neural Computation* 4, 415-447.
- Maruyama, M., F. Girosi, and T. Poggio (1991) A connection between GRBF and MLP. *Artificial Intelligence Memo 1291*, Massachusetts Institute of Technology.
- Merton, R. C. (1973), Theory of Rational Option Pricing, *Bell Journal of Economics and Management Science*, 4, 141-182.
- Merton, R. C. (1976), Option Pricing When Underlying Stock Returns are Discontinuous, *Journal of Financial Economics*, 3, 125-144.
- Moody, J., and C. Darken (1989) Fast learning in networks of locally tuned processing units. *Neural Computation*, 1(2):281-294.
- Ng, K. and R. Lippman (1991) A comparative study of the practical characteristics of neural network and conventional pattern classifiers. In R. Lippman, J. Moody, and D. Touretsky, editors, *Advances in Neural Information Processing Systems* J. Morgan-Kaufman.
- Niyogi, P., and F. Girosi (1994) On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. A.I. Memo 1467, MIT Artificial Intelligence Laboratory.

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.

Parker, D.B. (1985) Learning logic. Technical Report 47, Center for Computational Research in Economics and Management Science, MIT, April.

Poggio, T., and F. Girosi (1990) Networks for approximation and learning. *Proceedings of IEEE*, 78(9):1481—1497.

Powell, M. J. D. (1987) Radial basis functions for multivariable interpolation: A review. In J. C. Mason and M. C. Cox, editors, *Algorithms for Approximation*, pages 143-167. Clarendon Press, Oxford.

Reinsch, C. H. (1967) Smoothing by spline functions. *Numer Math*, 10:177-183.

Robbins H., and S. Monro (1951) A stochastic approximation model. *Annals of Mathematical Statistics*, 22:400-407.

Rosenfeld, E. (1980), *Stochastic Processes of Common Stock Returns: An Empirical Investigation*, Ph.D. Dissertation, MIT.

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.

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

Rumelhart, D.E., G.E. Hinton, and R.J. Williams (1986) Learning internal representation by error propagation. In D.E. Rumelhart and IL. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1: Foundations, chapter 8. MIT Press, Cambridge, MA, 1986.

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

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

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

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.

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

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