Knowledge Artificial Neural Networks to Enhanced Parametric Option Pricing

Andreou C. Panayiotis[1], Charalambous Chris[2], Martzoukos H. Spiros[3*]

University of Cyprus

Department of Public and Business Administration

This Version: January 2006

JEL classification: G13, G14

Keywords: Option pricing, implied volatilities, implied parameters, artificial neural networks, optimization

Acknowledgements: This work has been partly funded by the Hermes European Center of Excellence in Computational Finance and Economics, and a University of Cyprus grant for research in ANN and Derivatives.

1,2,3 PhD Candidate, Professor of Management Science and Assistant Professor of Finance respectively
* Corresponding author:
Spiros H. Martzoukos, Assistant Professor of Finance
University of Cyprus, Dept. of Public and Business Administration
P.O. Box 20537, CY 1678 Nicosia – Cyprus
Fax: +357-22-89 24 60, Tel.: +357-22-89 24 74
email: baspiros@ucy.ac.cy

Abstract

In this paper we explore ways that alleviate problems of nonparametric (artificial neural networks) and parametric option pricing models by combining the two. The resulting knowledge enhanced network model is compared to standard artificial neural networks and to parametric models with several historical and implied parameters. Empirical results using S&P 500 index call options strongly support our approach.

Introduction

We propose a new approach in the empirical pricing of options by combining nonparametric Artificial Neural Network (ANN) methodology in conjunction to several established uses of parametric models with historical and implied parameters. This proposed method we call Knowledge (enhanced) Artificial Neural Networks (KANNs). It can be seen as a sophisticated ANN structure dedicated to option pricing and provides a nonparametric enhancement of parameter values used in the Parametric Option Pricing Models (POPMs).

The Black and Scholes model (BS) is an options pricing formula (Black and Scholes, 1973, see also Merton, 1973) that is built on a set of unrealistic assumptions and exhibits *systematic biases* like the *volatility smile* (i.e. Black and Scholes, 1975, Rubinstein, 1985, Bakshi et al., 1997, Andresen, 2002). BS has shown severe time endurance and is still widely used by practitioners since it generates quite accurate prices for a wide spectrum of European financial options. The post-BS financial engineering research came up with a variety of parametric option pricing models that relax several of the BS fundamental assumptions. Recent POPMs that incorporate stochastic volatility and jump risk factors (e.g. Bakshi et al., 1997, Bates, 1991 and 1996), mitigate much of the bias associated with the original BS. Nevertheless, none of these models has managed to generalize *all* of the BS assumptions, and provide results consistent with the observed market data. Besides the fact that the above-mentioned parametric models seem to perform better than the BS[1] they are *often too complex to implement, have poor out-of-sample pricing and hedging performance and have implausible and sometimes inconsistent implied parameters* (i.e. Bakshi et al., 1997).

For resolving this issue researchers have addressed attention to the use of *market-data driven* models such as ANNs that can be used for nonlinear regression. The key power provided by ANNs compared to other statistical techniques (like projection pursuit, generalized additive models, multivariate adaptive regression splines) is that they rely on fairly simple algorithms and the underlying form of the nonlinearity can be learned from training data. The

[1] Although the post-BS option-pricing models have managed to eliminate some of the BS biases in practice are very difficult to be implemented due to their complexity. According to Andersen et al., (2002), *"the option pricing formula associated with the Black and Scholes diffusion is routinely used to price European options, although it is known to produce systematic biases".*

models are extremely powerful, have nice theoretical properties (with respect to convergence), and apply well to a vast array of real-world applications (see Duda et al., 2001, for further details). Attempts in pricing options with ANNs have shown that these models are promising alternatives in respect to robust pricing accuracy. Contrary to the parametric option pricing models that rely on specific assumptions about the dynamic evolution of some state variables (like the underlying asset, the volatility, the interest rate, etc), ANNs involve no financial theory since the option's price is estimated inductively by using options transactions data.

ANNs are used to estimate directly the empirical options pricing function (thereinafter termed as the standard ANN approach). Evidence concerning the out-of-sample pricing performance is mixed. Hutchison et al. (1994) apply ANNs on market transactions of the S&P 500 futures call options from 1987 to 1991 to conclude that although the learning networks do not constitute a substitute for the more traditional BS formulas, they are more accurate and computationally more efficient alternatives when the underlying asset's price dynamics are unknown. Lajbcygier et al. (1996) that examines futures call options transacted on the Sydney Futures Exchange find that ANNs outperform the modified BS formula in a reduced data region, but they are not superior when all call option are considered (see also Lajbcygier, 2004 for extended results). Andreou et al. (2005) conjecture that similar results hold for a reduced and a full dataset for the S&P 500 index call options even for more general OPM that alleviate much of the BS biases, like the Corrado and Su model (hereinafter CS, see Corrado and Su, 1996), with both historical and implied parameters. Anders et al. (1998) as well as Garcia amd Gencay (2000), find the BS with historical volatility underperforms significantly the standard ANNs.

Of course, the application of ANNs for pricing of options has its own merits and limitations. First of all, Anders and Korn (1999) indicate that neural networks are usually applied in cases where there is lack of knowledge about an adequate functional form; so they are commonly interpreted as "black boxes" since they learn the empirical functions inductively from transactions data without embedding any information related to the problem under scrutiny. Second, in the absence of any kind of knowledge or prior information about the problem, ANNs need relative large amounts of training data to ensure an adequate accuracy.

The above demonstrate that we should explore ANN structures that are enhanced by knowledge from the POPMs. As supported by Lajbcygier (2004), the standard ANNs are very sensitive to the nonstationarity of input variables and this problem is exaggerated with the use of large training-validation-testing datasets; thus we should eventually use any knowledge together with ANNs in an attempt to minimize the size of datasets. The use of standard ANNs can deliver options prices that violate fundamental financial principles; for instance they can return negative option values or irrational Greek letters (these are the partial derivatives of the option with respect to a parametric model's structural parameters). Therefore we must combine ANNs and POPMs in a way that does not allow such problems to occur.

Conceptual Framework

In this study we extend the ANN structures that have been applied previously in options pricing, by using a knowledge enhancement methodology that allows an adjustment to some of the input variables to the parametric model. A significant feature of our methodology is that it allows a set of the input variables of the parametric model to be *jointly* determined by a neural network.

Such kind of knowledge oriented ANN models are more desirable than the standard ones for a variety of reasons. First, they will always return arbitrage-free and nonnegative option values and we thus expect them to exhibit reasonable pricing performance at the boundary option pricing areas, in both dense and sparse input areas. Similarly, it is also *certain* that KANN will always deliver theory consistent Greek letters.

Second, the proposed approach assures nonnegative implied state price densities in all cases. Herrmann and Narr (1997) show that standard ANNs return negative implied state price densities in state regions that available training options data do not contain any information about these regions.

Third, as conjectured by Wang and Zhang (1997), knowledge based ANN structures should not need large amount of training samples to exhibit a satisfactory performance in out of sample testing as opposed to the case of standard ANNs.

The proposed methodology can fit the options pricing field in many perspectives. It can be utilized in studies like the one of Bakshi et al. (1997)

(see also Eraker, 2004, Bates, 1996 and 2000, Corrado and Su, 1996 and 1997, Whaley, 1982, Lehar et al., 2002, etc) that examine the cross sectional pricing performance of alternative option pricing models. They rely on previous day's option prices to back out the required parameter values and then to use them as input to compute current day's model-based option prices. KANNs can also be considered as a generalization of studies that first employ some kind of methodology to estimate versions of time varying volatility that is either simultaneously or subsequently used with the BS model to price options. For instance, Dumas et al. (1998) estimate arbitrary Deterministic Volatility Functions of quadratic forms and examine how well they predict option prices. Kiesel (2002), elaborates on semi-parametric version of the Black and Scholes formula according to which the formula is used with a volatility function that is estimated inductively by using a three-dimensional Kernel Estimator. In another example, Adesi et al. (2005) derive relatively simple option pricing formulas in which the instantaneous variance is driven by a GARCH diffusion process.

In addition, the KANN structure we develop has some common characteristics with a variety of previous approached in the engineering area. For instance it is knowledge oriented as the ANN structures proposed by Wang and Zhang (1997) for microwave design, with some significant differences: *i)* they do not use a fully connected network as we do, *ii)* contrary to our approach, they use a relative more complex network structure with six different layers, *iii*) they employ the knowledge layer in a parallel fashion with their hidden layers where the error propagation is split into two paths, one through a knowledge layer and the other through a set of hidden layers each serving a specific purpose, and *iv*) the knowledge embedded in their network is in the form of empirical and semi-analytical functions. Our methodology also preserves some characteristics of the Neuromodeling Space Mapping Techniques of Bandler et al., (1999) (see also Sanchez, 2004) developed for microwave circuits design and optimization. The Space Mapping concept was initiated by Bandler et al. (1994); it establishes a mathematical link between the computational efficiency of coarse models with the accuracy of fine models. The coarse models are computationally very efficient with limited accuracy models while fine models are extremely accurate but very computer intensive. In the Neuromodeling Space Mapping concept, the purpose is to make the coarse model response as close as possible to the fine model response for all the training points considered. This is achieved by constructing a nonlinear multidimensional vector mapping function from fine to coarse input space using a typical ANN.

The data for this research come from two dominant world markets, the New York Stock Exchange (NYSE) for S&P 500 and the Chicago Board of Options Exchange (CBOE) for call option contracts, spanning a period from January 2002 to August 2004. Compared to previous literature in empirical options pricing, we examine more explanatory variables including historical and implied ones. Also, instead of constant maturity risk-free interest rate, we use nonlinear interpolation for extracting a continuous risk-free interest rate according to each option's time to maturity.

In this study we do not restrict our attention to parametric benchmarks that have limited flexibility in terms of the underlying diffusion process. For instance, BS assumes that log-relative returns are normally districted but empirical evidence does not support this principle. ANNs can allow more general probability density distributions (see Herrmann and Narr, 1997). Thus, it is imperative to compare ANNs with: *i*) the BS model with other versions of volatility measures besides the traditional historical ones, and *ii*) parametric models that allow for negative skewness and excess kurtosis for the underlying asset's log-relative returns; thus, as in Andreou et al. (2005) we will also consider the semi-parametric CS that can be considered as a flexible model that can proxy for many other more complex ones[2].

The Parametric Models Used

Below we reel off the different POPMs we exploited in this study. The first model examined is the Black and Scholes (1973) since is a benchmark and widely referenced model. The Black Scholes formula for European call options modified for dividend-paying underlying asset is:

$$c^{BS} = S e^{lT} N(d) - X e^{-rT} N(d - s\sqrt{T}) \quad (1)$$

where,

[2] Backus et al. (1997) conjecture that the CS formula exhibits good performance for pricing options when the underlying asset follows a jump-diffusion process (see also Jurczenko et al., 1997).

$$d = \frac{\ln(S/X) + (r - l)T + (s\sqrt{T})^2 / 2}{s\sqrt{T}} \quad (1.a)$$

$c^{BS} \equiv$ *premium paid for the European call option*; $S \equiv$ *spot price of the underlying asset*; $X \equiv$ *exercise price of the option*;[3] $r \equiv$ *continuously compounded risk free interest rate*; $l \equiv$ *continuous dividend yield paid by the underlying asset*; $T \equiv$ *time left until the option expiration date*[4]; $s^2 \equiv$ *yearly variance rate of return for the underlying asset*; $N(.) \equiv$ *the standard normal cumulative distribution* .

The need to use more advance POPMs is necessitated by the fact that exist a specific behavior of the BS implied volatility for various moneyness (the ratio of the underlying asset to strike price) and time to maturity levels. (see Bakshi et al., 1997) So, we use the Corrado and Su (1996) model that constitutes an extension of the BS model that accounts for additional skewness and kurtosis in stock returns in a heuristic manner. Corrado and Su, based their extension on a methodology employed earlier in 1982 by Jarrow and Rudd. Via subtle handlings of the Gram-Charlier series expansion of a normal density function they defined their model as (see also the correction in Brown and Robinson, 2002):

$$c^{CS} = c^{BS} + m_3 Q_3 + (m_4 - 3)Q_4 \quad (2)$$

where c^{BS} is the BS value for the European call option adjusted for dividends and,

$$Q_3 = \frac{1}{3!} S e^{-lT} s\sqrt{T}((2s\sqrt{T} - d)n(d) + s^2 T N(d)) \quad (2.a)$$

$$Q_4 = \frac{1}{4!} S e^{-lT} s\sqrt{T}((d^2 - 1 - 3s\sqrt{T}(d - s\sqrt{T}))n(d) + s^3 T^{3/2} N(d)) \quad (2.b)$$

[3] For the purposes of this study we use the following moneyness categories: *deep out the money* (DOTM) when S/X=0.90, *out the money* (OTM) when 0.90<S/X=0.95, *just out the money* (JOTM) when 0.95<S/X=0.99, *at the money* (ATM) when 0.99<S/X=1.01, *just in the money* (JITM) when 1.01<S/X=1.05, *in the money* (ITM) when 1.05<S/X=1.10, *deep in the money* (DITM) when S/X>1.10.

[4] In terms of time length, an option contract is classified as *short term maturity* when its maturity is less than 60 days, as *medium term maturity* when its maturity is between 60 and 180 days and as *long term maturity* when it has maturity longer than (or equal to) 180 days.

Q_3 and Q_4 represent the marginal effect of non-normal skewness and kurtosis, respectively in the option price whereas m_3 and m_4 correspond to coefficients of skewness and kurtosis. In the above expressions,

$$n(z) = \frac{1}{\sqrt{2p}} exp(-z^2 / 2) \qquad (2.c)$$

refers to the standard normal probability density function.

Greek letters are the partial derivatives of a call options with respect to its structural parameters. For the purpose of this study, we need the following Greek letters:

- *BS Vega*:

$$V^{BS} \equiv \frac{\partial c^{BS}}{\partial s} = Se^{-lT}\sqrt{T}n(d) \qquad (3)$$

- *CS Vega*:

$$\frac{\partial c^{CS}}{\partial s} = \frac{\partial c^{BS}}{\partial s} + \frac{1}{3!} m_3 \frac{\partial Q_3}{\partial s} + \frac{1}{4!}(m_4 - 3)\frac{\partial Q_4}{\partial s} \qquad (4)$$

where,

$$\frac{\partial Q_3}{\partial s} = Se^{-lT}\, n(d) s\sqrt{T}\left[3ds\sqrt{T} + 3d^2 + s\sqrt{T} + 3\right] - S\sqrt{T}d^3\, n(d) + 3Ss^2T^{3/2}N(d) \qquad (5)$$

$$\frac{\partial Q_4}{\partial s} = S\sqrt{T}\, d\, n(d)\left[-2d + d^3 - 4s\sqrt{T}\, d^2 + 6d\left(s\sqrt{T}\right)^2 - 4\left(s\sqrt{T}\right)^3\right] - S\sqrt{T}\, n(d) + 6Ss^2T^{3/2}\, n(d) + 4Ss^3T^2N(d) + S\, n(d)s^4T^{3/2} \qquad (6)$$

- CS *partial derivative of call with respect to skewness*:

$$\frac{\partial c^{CS}}{\partial m_3} = Q_3 \qquad (7)$$

- CS *partial derivative of call with respect to kurtosis*:

$$\frac{\partial c^{CS}}{\partial m4} = Q_4 \tag{8}$$

Neural networks

Multilayer Neural Networks are a flexible heuristic technique for doing statistical pattern recognition and for approximating highly nonlinear functions. A neural network is a collection of interconnected simple processing elements structured in successive layers and can be depicted as a network of links (termed as *synapses*) and nodes (termed as *neurons*) between layers. A typical feedforward neural network has an input layer, one or more hidden layers and an output layer. Each interconnection corresponds to a modifiable weight, which is adjusted according to the faced problem via an optimization, commonly termed as training algorithm. The particularity of ANNs relies on the fact that the neurons on each layer operate collectively and in a parallel manner on all input data.

Figure 1 depicts the network structure developed for the purposes of this study. As it will become apparent shortly, such structure can been seen as an extension of the two-layer network studied in Andreou et al. (2005) and elsewhere (e.g. Hutchison et al., 1994, Lajbcygier et al., 1996).

For our analysis, inputs are set up in feature vectors, $\tilde{x}_q = [x_{1q}, x_{2q} \ldots, x_{Nq}]$ for which there is an associated and known target, t_q, $q \equiv 1,2,\ldots,P$, where P is the number of the available sample feature vectors for a particular training sample. The network's outputs are obtained when the training patterns are presented as inputs at the input layer and after evaluating the signals at each node. To let the network learn the underlying relationship, its weights are adjusted in order to minimize the error between the network output and the desire target values.

The proposed network model under scrutiny has four layers. The first three are typical ANN layers: an *input* layer with N input variables, a *hidden* layer with H neurons, and a layer with M *output* neurons. For these three layers, each node is connected with all neurons in the previous and the forward layer. Each connection is associated with a *weight*, $w_{in}^{(1)}$, and a *bias*, $w_{i0}^{(1)}$, in the input layer ($i=1,2,\ldots,H$, $n=1,2,\ldots N$) and a *weight*, $w_{ji}^{(2)}$, and a *bias*,

$w_{j0}^{(2)}$, in the hidden layer ($j=1,2,\ldots,M$). Each neuron behaves as a summing vessel that computes the weighted sum of its inputs to form a scalar term and with the use of the transfer function it eventually works as a non-linear mapping junction for the forward layer. The part of the network that is outside the bold-dotted line in Figure 1 is a typical two-layer ANN with a single output that under proper treatment can be used for nonlinear regression. Such type of networks has been used previously to approximate the empirical options pricing function (e.g. Andreou et al., 2005, Hutchison et al., 1994). In our analysis, we employ this type of network for implementing the standard ANN results for comparison reasons with the KANNs.

The fourth layer, which hereafter will be termed as a *knowledge layer*, makes possible for a chosen parametric options pricing model to be an *inseparable* part of the network's structure. This is the innovative contribution of the model we develop since under this setting we can hypothesize that our network structure embeds knowledge from the parametric model during training. If we let X_S to denote the set of all input variables that are necessary for the parametric model to price options, then $X_{S1} \supseteq X_S$ should correspond to the *enhanced*[5] variables coming from the network's output layer and $X_{S2} \supset X_S$ those variables that are passed to the parametric model *exogenously*. It is obvious that $X_{S2} = X_S - X_{S1}$ and in the case that we choose to let all parametric model variables to be determined via the network, then $X_{S2} = \varnothing$. The definition of X_{S1} is basically a choice of the researcher and manifests the number of neurons at the output layer and the type of transfer function to be used at the knowledge layer.

According to Figure 1, the operation carried out for computing the final estimated output, y, in the case of a single endogenous variable ($X_{S1} = \{v_1\}$) is the following:

$$y = f_{PM}(v_1, X_{S2}) \tag{9}$$

and,

$$v_1 = f_{d_1}(d_1) \tag{10}$$

[5] We use the term "enhanced variable" to describe the number of variables that come as an output of the network and used as input to the parametric model.

where $f_{PM}(.,.)$ refers to the functional form of the parametric options pricing model, $f_{d_1}(.)$ is a smooth monotonically increasing transfer function and d_1 is simply the descaled valued of $y_1^{(2)}$. Eq. (9) can be interpreted as a nonlinear regression of y over the networks inputs if we allow an error term.

Computation of $y_1^{(2)}$ follows the functional form of a typical two-layer ANN similar with the one used by Andreou et al. (2005) and Hutchison et al. (1994):

$$y_1^{(2)} = f_M[w_{10}^{(2)} y_0 + \sum_{i=1}^{H} w_{1i}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_{sn})] \qquad (11)$$

where $f_M(.)$ and $f_H(.)$ are smooth monotonically increasing transfer functions associated with the output and hidden layer respectively and x_{sn}, n=1,2,…,N, is just the scaled value of the input x_n. The network's structure employs a scaling scheme for both the inputs and the enhanced variables. This is essential for the training of ANNs since it increases the effectiveness of the optimization algorithm and minimizes the significance of different dimensions of the input signals (see Haykin, 1999, and Bishop, 1995). We apply a standard *z-score* scaling: $\tilde{z} = (\tilde{x} - m)/s$, where $\tilde{x}$ is the vector of an input/enhanced variable, m is the mean and s the standard deviation of this vector.

[Figure 1, here]

For our case, the smooth monotonically increasing transfer functions is either the hyperbolic tangent sigmoid,

$$f(h) = a\left[\frac{e^{bh} - e^{-bh}}{e^{bh} + e^{-bh}}\right] \qquad (12)$$

the logistic,

$$f(g) = \frac{a}{1+e^{-bg}} \tag{13}$$

or the linear one,

$$f(x) = x \tag{14}$$

The differential of the above expressions can be expressed in a particularly simple form and equal to:

$$f'(h) = \frac{b}{a}\left(a^2 - f^2(h)\right) \tag{15}$$

for the hyperbolic tangent sigmoid,

$$f'(g) = bf(g)\left(1 - \frac{f(g)}{a}\right) \tag{16}$$

for the logistic and,

$$f'(x) = 1 \tag{17}$$

for the linear transfer function. In the above expressions, with $a, b \in \Re$ where a controls the output range and b the slope of the transfer function. As advised by Duda et al. (2001, pg.308), the overall range and slope are not important, because it is their relationship to parameters such as the learning rate and magnitudes of the inputs and targets that affect learning. According to Bishop (1995) (see also Duda et al., 2001), these transfer functions work well with ANNs. In the hidden layer we always use the standard hyperbolic tangent sigmoid transfer function (with a and b equal unity) for $f_H(.)$, while in the output layer we use a linear transfer function for $f_M(.)$ as this is necessitated by the scaling scheme we apply at the output layer.

The choice of the transfer function at the knowledge layer is dictated by the type of the parametric model we use and the kind of the enhanced variable(s) we choose to map via the network; thus it is possible for $f_{d1}(.)$, $f_{d2}(.),\ldots,f_{dM}(.)$ to be different depending on the case considered. This set of transfer functions are necessary during the implementation of the method in order to ensure that each of the enhanced variable value is in an acceptable range for use with the parametric model[6]. Table 1 – Panel A describes the different transfer functions we have used at the knowledge level for all cases considered. We use transfer functions that truncate implicitly the enhanced variable value range. For instance in the case of BS we do not allow volatility to be larger than 150% and for the case of CS, skewness is confined in the $[-15,15]$ range. The choice of the truncation point is not crucial for the implementation of the models as long as we allow the enhanced variables to vary into plausible ranges. This choice can be guided by empirical investigation. For example we rarely observe volatility to be above 150% or skewness to be below -15 or above 15 (e.g. Corrado and Su, 1997, Bates, 1991).

[Table 1, here]

The training of any type of ANN model is a highly non-linear optimization process in which the network's weights are modified according to an error function. Below we describe the updating formulas for the training of our networks in the case that there is only one enhanced variable. The formulas for the general case of more enhanced variables are trivial. The error function between the estimated response y_q and the actual response t_q is defined as:

$$e_q(w) = y_q(w) - t_q \tag{18}$$

[6] For instance, if BS is the chosen parametric model and volatility is the enhanced variable, then our transfer function should be a logistic that allows only positive values whilst if the enhanced variable is the skewness of CS then the transfer function should be a hyperbolic tangent one that allows for both positive and negative values.

where, w is an ?-dimensional column vector containing the weights and biases given by: $w = [w_{10}^{(1)} \dots, w_{H0}^{(1)}, w_{11}^{(1)}, \dots, w_{HN}^{(1)}, w_{j0}^{(2)}, w_{j1}^{(2)}, \dots, w_{jH}^{(2)}]^T$. The traditional backpropagation[7] algorithm which is based on the gradient descent error is the most popular method for training the ANNs. It is shown in Charalambous (1992) that this training algorithm is often unable to converge *rapidly* to the optimal solution. So, in this paper we rely on the Levenberg-Marquardt algorithm (LM) which is much more efficient training method in terms of training time and convergence rate. According to LM, the weights and the biases of the network are updated in such a way so as to minimize the following sum of squares performance function:

$$F(w) = \sum_{q=1}^{P} e_q^2(w) \equiv \sum_{q=1}^{P} (y_q - t_q)^2 \tag{19}$$

Then, at each iteration t of the algorithm, the weights vector w is updates as follows:

$$w_{t+1} = w_t + \left[J^T(w_t) J(w_t) + m_t I\right]^{-1} J^T(w_t) e(w_t) \tag{20}$$

where, $J(w_t)$ is the $P \times ?$ Jacobian matrix of the *P-dimensional* output error column vector at t^{th} iteration, and is given by:

$$J(w) = \begin{bmatrix} \nabla e_1^T(w) \\ \vdots \\ \nabla e_P^T(w) \end{bmatrix} \tag{21}$$

In the above, I is $n \times n$ identity matrix, and μ_t is like a learning parameter that is automatically adjusted in each iteration in order to secure convergence (by assuring that the part in the square brackets of Eq. (20) is always nonsingular). Large values of μ_t lead to directions that approach the steepest descent, while small values lead to directions that approach the Gauss-

[7] The backpropagation algorithm is so-called because during training an error must be propagated from the output layer *back* to the hidden layer in order to perform the learning step (from Duda, 2001, pg. 292).

Newton algorithm. Further technical details about the implementation of LM can be found in Hagan and Menhaj (1994) and Hagan et al. (1996). Based on Eq. (20), the weights and biases update takes place in a batch mode and only when all input vectors have been presented to the network. Moreover, the network initialization technique proposed by Nguyen and Windrow (see Hagan et al., 1996) that generates initial weights and bias values for a nonlinear transfer function so that the active regions of the layer's neurons are distributed roughly evenly over the input space is employed accordingly.

The quantity $\nabla e_q(w)$ is the gradient vector of $e_q(w)$ with respect to the trainable parameter vector w. This quantity is computed in a similar fashion as with the case of the traditional backpropagation algorithm that is commonly used in the context of multilayer perceptron neural networks. Since the error function does not depend explicitly upon the network's weights, $\nabla e_q(w)$ is evaluated via the chain rule. Based on the neural network model depicted in Figure 1, the partial derivative of the error function in Eq. (18) with respect to the weight $w_{ji}^{(2)}$ at the hidden layer is:

$$\frac{\partial e_q}{\partial w_{ji}^{(2)}} = d_j^{(2)} y_i^{(1)} \tag{22}$$

and,

$$d_j^{(2)} = f'_{PM}(v_j) f'_{d_j}(d_j) s_j f'_M(y_j^{(2)}) \tag{23}$$

where $f'_M(y_j^{(2)})$ and $f'_{d_j}(.)$ are the differentials of the knowledge and the output neuron transfer function at points $y_j^{(2)}$ and d_j respectively (see Eqs. (15) and (17)), and s_j the standard deviation of the enhanced variable as used during scaling.

Quantity $f'_{PM}(v_j)$ is the partial derivative of the parametric model with respect to input v_j and makes our network model more dedicated to options pricing. We believe that this quantity is very important during the training of the ANN because it incorporates knowledge from a parametric model. All necessary Greek letters for the implementation of the different KANN models have been previously discussed in the parametric models section.

The partial derivative of the error function in Eq. (18) with respect to the weight $w_{in}^{(1)}$ at the input layer is:

$$\frac{\partial e_q}{\partial w_{in}^{(1)}} = d_i^{(1)} x_{sn} \tag{24}$$

where,

$$d_i^{(1)} = e_i^{(1)} f'_H(y_i^{(1)}) \tag{25}$$

$$e_i^{(1)} = \sum_{j=1}^{M} w_{ji}^{(2)} d_j^{(2)} \tag{26}$$

and x_{sn} is simply the z-score scaled value of x_n .

The optimal number of hidden neurons is chosen via a cross-validation procedure. Standard ANN structures and KANNs with 2 to 6 hidden neurons are trained, and the one that performs the best in the validation period is selected. Since the initial network weights affect the final network performance, for a specific number of hidden neurons, the network is initialized, trained and validated ten separate times. After defining the optimal network structure, its weights are frozen and its pricing capability is tested (out of sample) in a third separate *testing dataset* in order to verify the ANN ability to generalize to unseen data.

Data and Methodology

The data considered cover the period January 2002 to August 2004. The S&P 500 index call options are used because this option market is extremely liquid; one of the most popular index options traded in the CBOE and is the closest to the theoretical setting of the parametric models (Garcia and Gencay, 2000). All options data are purchased from CSI. For each trading day we have the available last transaction call price, c^{mrk} , along with the striking price, X , date of expiration, volume and open interest. Along with the index, we have collected a daily dividend yield, l, provided online by Datastream.

We used a chronological data partitioning via a rolling-forward procedure in order to have a better simulation of the actual options trading conditions. The data is divided into ten different *overlapping* training (*trn*) and validation (*vld*) sets, each followed by separate and *non-overlapping* testing (*tst*) set. Each *trn*, *vld* and *tst* period has 8, 4 and 2 month spanning period respectively. For instance, the first *trn* set covers the period January to August 2002, the first *vld* set covers the period September to December 2002, the first *tst* set covers the period January to February 2003, etc. The ten testing (out of sample) periods are non-overlapping and cover almost the last two years of the data-set. According to this splitting, the *trn* sets have about 10,000 datapoints, the *vld* about 5,000 datapoints and the *tst* about 2,500 datapoints.

For the needs of the analysis, we created an aggregate testing period (*agr*) with 25,750 datapoints by simply pooling together the pricing estimates of all ten *tst* periods. For *agr* and for the case of pricing accuracy, we compute and tabulate: the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Median Absolute Error (MdAE) and for conceptualizing each model's absolute error distribution, we compute the 5th Percentile of Absolute Error (P_5AE) and 95th Percentile of Absolute Error (P_{95}AE).

Filtering Rules

To create an informative dataset we mostly rely on the filtering rules adopted before by Bakshi et al. (1997) for empirical options pricing (see also Andreou et al., 2005). We first eliminate all observations that have zero trading volume since they do not represent actual trades. Second, we eliminate observations that violate either the lower or the upper arbitrage bounds. Third, we eliminate all options with less than six days to expiration to avoid extreme option prices that are observed due to potential illiquidity problems. In the same spirit, options with more than 260 trading days are also excluded. Fourth, price quotes of less than 0.5 index points are not included since it perceived that such options come from a different data generating process. Last, we demand at least four datapoints per maturity to secure that during the implied parameters extraction process, every maturity period is satisfactorily represented.

The final dataset used is still larger than previous studies. For instance Hutchison et al. (1994) have an average of 6,246 data points per sub-period. Lajbcygier et al. (1996) include 3,308 data points, and Schittenkopf and Dorffner (2001) include 33,633 data points.

Observed Structural Parameters

The moneyness ratio, S/X, is the basic input to be used with all network structures since it is highly related with the pricing bias associated with the POPMs. The moneyness ratio S/X is calculated and used like in Hutchison et al. (1994) (see also Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-lT})/C$ is preferred here since dividends are relevant. In addition, the time to maturity (T) is computed assuming 252 days in a year. Previous studies have used 90-day T-bill rates as approximation of the interest rate. In this study we use nonlinear cubic spline interpolation for matching each option contract with a continuous interest rate, r, that corresponds to the option's maturity. For this purpose, 1, 3, 6, and 12 months constant maturity T-bills rates (rates collected from the U.S. Federal Reserve Bank Statistical Releases) were taken into consideration.

Here we use several volatility measures with BS; as in Bakshi et al. (1997), we employ overall average implied parameter measures that are theory consistent. This is in contrast to several studies (e.g. Hutchison et al., 1994, Garcia and Gencay, 2000) where the assumption is made that ANNs are able to capture the associated volatility from data. For pricing reasons at time instant *t*, the implied structural parameters derived at day *t-1* are used together with all other needed information.

The 60-days volatility is a widely used historical estimate (see Hutchison et al., 1994, and Lajbcygier et al., 1997). This estimate is calculated using all past 60 log-relative index returns and is symbolized as s_{60}^{BS}. In addition, the VIX Volatility Index is an estimate that can be directly observed from the CBOE. It was originally developed by CBOE in 1993 as a measure of the volatility of the S&P 100 Index (currently termed VXO) but nowadays reflects a proxy for the volatility of S&P 500. VIX is calculated as a weighted average of S&P 500 option with an average time to maturity of 30 days and emphasis on at-the-money options. This volatility measure is symbolized as s_{vix}^{BS}.

Weighted Volatility Measures

To obtain the contract specific implied volatility (s^{imp}) of each option we use the widely applied Newton-Raphson. Afterwards, each of the implied volatilities is utilized to create daily weighted implied volatility forecasts to be used with the BS. The general formula we use is:

$$s(t) = \frac{\sum_{j=1}^{N_t} x_j s_j^{imp}}{\sum_{j=1}^{N_t} x_j} \tag{27}$$

where N_t refers to the number of different call option transaction datapoints available at each time instance t and x is a weighting factor. We apply various versions of x in this study. The most widely used weighted volatility measure is the one proposed by Chiras and Manaster (1978) in which is implied volatility is weighted by the price elasticity of the call option:

$$x = \frac{\partial c^{mrk}}{\partial s^{imp}} \frac{s^{imp}}{c^{mrk}} \tag{28}$$

This weighted volatility forecast will be termed as s_{cm}^{BS}. In addition, we also implement another weighting scheme in which x represents the percentage of the trading volume of each options. As documented by Day and Lewis (1988), at the money options which are the most sensitive to volatility changes concentrate the largest trading activity and out-of-the money options that might represent noisy trades concentrate the least trading activity. Volume weighted volatility will be termed as s_{vol}^{BS}.

Implied Volatility Measures

The methodology employed in this study for the implied parameter estimation is similar to that in previous studies that somehow adopt the Whaley's (1982) simultaneous equation procedure to minimize a price deviation function with respect to the unobserved parameters. As with Bates (1991), market option prices (c^{mrk}) are assumed to be the corresponding POPM prices (c^k, k=BS or CS) plus a random additive disturbance term (e_N^k, k=BS or CS):

$$c_N^{mrk} = c_N^k + e_N^k \tag{29}$$

where N refers to the number of different call option transaction datapoints available. To find optimal implied parameter values we solve an optimization problem that has the following form:

$$SSE(t) = \min_{q^k} \sum_{j=1}^{N_t} (x_j e_j^k)^2 \tag{30}$$

where t represents the time instance, $?^k$ the unknown parameters associated with a specific parametric options pricing model ($q^{BS} = \{s^{BS}\}$, $q^{CS} = \{s^{CS}, m_3, m_4\}$) and x is a weighing factor. The SSE is minimized via a non-linear least squares optimization based again on the Levenberg-Marquardt algorithm. To minimize the possibility to obtain implied parameters that correspond to a local minimum of the error surface with each model we use three different starting values for the unknown parameters based on reported average values for the S&P 500 (but for different periods) according to Bates (1991), Bakshi et al. (1997), and Corrado and Su (1996 and 1997).

The above approach is used daily to obtain two different sets of implied parameters for each parametric model. The first optimization is performed by including all available options data in order to obtain daily average implied parameters by assuming that x is unity; this is consistent with Bates (1991), Bates (1996), Bakshi et al. (1997), and Corrado and Su

(1996). These implied parameter measures will be termed as *implied-overall average* and for notation reasons will be denoted with *av*. Furthermore, the same procedure is done for each by assuming that x is the percentage of the daily trading volume for each option observation included in the optimization. As explained before, the motivation for this approach follows the study of Day and Lewis (1988). These implied parameter measures will be termed as *implied-overall volume average* and for notation reasons we will be denoted with *vav*.

For the optimization procedure above we have imposed two kind of constraints for practical reasons; nonnegative implied parameters are optimized using an exponential transformation, the skewness of CS[8] is allowed to vary in the range [-10, 5] whereas kurtosis is constrained to be less than 30 .

The two different implied BS volatility estimates will be symbolized as: s_j^{BS} , $j = \{av, vav\}$, whilst the two different sets of CS parameters as: $\{s_j^{CS}, m3_j^{CS}, m4_j^{CS}\}$. We should note that the pricing dynamics differ between BS and CS so we can expect s_j^{BS} and s_j^{CS} to differ for the same call contract.

Comparison of the Alternative Models

With the BS models we use as input S, X, T, l , r, and any of the following six volatility forecasts: s_{60}^{BS} , s_{vix}^{BS} , s_{av}^{BS} , s_{vav}^{BS} , s_{cm}^{BS} and s_{vol}^{BS} . We use the following notation when we refer to the parametric BS models: BS_{60} , BS_{vix} , BS_{av} , BS_{vav} , BS_{cm} , BS_{vol} . In a similar way there are two different CS models according to the implied parameters used: CS_{av} (with s_{av}^{CS} , $m3_{av}^{CS}$, $m4_{av}^{CS}$), and CS_{vav} (with s_{vav}^{CS}, $m3_{vav}^{CS}$, $m4_{vav}^{CS}$).

[8] If not somehow constrained, skewness and kurtosis can take implausible values due to model overfitting that will lead on enormous pricing errors on the next day (especially for deep in the money options). It is hard to believe that implied skewness and kurtosis can exceed these values (i.e. Bates, 1991) on most of the days. In our case these constraints were binding in less than 2% of the whole dataset and actually the fitting errors on these cases were in acceptable levels.

With the neural networks, we also use three standard input variables/parameters: $(Se^{-dT})/X$, T and r; in addition, the target variable is always the standardized market call price: c^{mrk}/X. The notation for the models depends on the parametric model considered. We use nBS_j, with *j*={*60*, *vix*, *av*, *vav*, *cm*, *vol*}, to denote the six standard ANNs that use as an additional input one of the BS volatilities: s_{60}^{BS}, s_{vix}^{BS}, s_{av}^{BS}, s_{vav}^{BS}, s_{cm}^{BS} and s_{vol}^{BS}. Similarly we use nCS_j, with *j*={*av*, *vav*}, to denote the two standard ANNs that use as additional input one set of the CS variables: either s_{av}^{CS}, $m3_{av}^{CS}$, $m4_{av}^{CS}$ or s_{vav}^{CS}, $m3_{vav}^{CS}$, $m4_{vav}^{CS}$. Moreover, we use kBS_j, with *j*={*60*, *vix*, *av*, *vav*, *cm*, *vol*}, to denote the six knowledge enhanced networks that use as an additional input variable the BS volatilities: s_{60}^{BS}, s_{vix}^{BS}, s_{av}^{BS}, s_{vav}^{BS}, s_{cm}^{BS} and s_{vol}^{BS}; for these models volatility is the only enhanced variable. In the same spirit we use kCS_j^{sig}, with *j*={*av*, *vav*}, to denote the two KANNs that use as additional inputs the CS variables: s_{av}^{CS}, $m3_{av}^{CS}$, $m4_{av}^{CS}$ and s_{vav}^{CS}, $m3_{vav}^{CS}$, $m4_{vav}^{CS}$ respectively, with volatility being the only enhanced variable. Finally, we use kCS_j^{all}, with *j*={*av*, *vav*}, to denote the two networks that use as additional inputs the CS variables: s_{av}^{CS}, $m3_{av}^{CS}$, $m4_{av}^{CS}$ and s_{vav}^{CS}, $m3_{vav}^{CS}$, $m4_{vav}^{CS}$ respectively, with volatility, skewness and kurtosis being the enhanced variables. Description of the network models can be found in Table 1 – Panel B.

Table 2 demonstrates descriptive statistics of the data used with all models exploited in this study. Tables 3 and 4 exhibit the performance of all models considered in terms of RMSE, MAE and RMeSE, P_5AE, P_{95}AE for the aggregate period (*agr*). Since all types of ANNs are effectively optimized in respect to sums of squares, (see Eq. 19), the out of sample pricing performance should be similarly based on RMSE and in a lesser degree on the other measures.

[Tables 2, 3, 4, here]

We first concentrate our attention to Panel A of Tables 3 and 4 for the parametric BS and CS models. BS_{av} is the best performing model within the alternative BS parametric alternatives. From the BS class results, we view that BS_{vav}, which utilizes a volatility forecast that is estimated with volume information, is the second best performing model and still inferior to BS_{av} in all performance measures considered. Likewise, CS_{av} outperforms CS_{vav}. In addition, CS_{av} (that captures better fat tails and asymmetry in log-returns) is the best performing model with the POPMs with RMSE equal to 3.21.

The pricing results of Table 3-Panel B show that the standard ANNs do not exhibit superior performance compared to the BS or CS parametric alternatives, although the overall results are mixed. We see that BS_{av} is superior compared to nBS_{av}. Likewise, CS_{av} is superior to nCS_{av}.

When we take all results into account, always the KANNs outperform significantly in all performance measures, both the respective parametric and standard ANNs. kBS_{cm} is the best performing within the KANN-BS class, followed by kBS_{av}, and kCS_{av}^{all} the best performing within the KANN-CS class. Overall we see that the ratio of the BS models to their KANN equivalents is between 1.23 and 1.61 whilst the corresponding figures for the CS case are 1.10 and 1.17. Another noticeable observation relates to the improvement in the A_{95}PE. This performance measure indicates a degree of confidence about large mispricings. Someone should feel more confident by using kBS_{av} (kCS_{av}^{all}) with A_{95}PE equal to 6.66 (5.97) compared to BS_{av} (CS_{av}) with A_{95}PE equal to 8.19 (6.61).

Overall the knowledge enhanced kCS_{av}^{all} (and kCS_{av}^{sig}) are the best performing models among all considered in this study. Despite this, given the relative simplicity of BS and its wide applicability (see Andersen et al., 2002), the BS based KANN models remain an excellent alternative.

Summary and Conclusions

In this study we combine the ANN models that have been applied previously in options pricing, with parametric models. Our approach allows a set of the input variables of the parametric model to be jointly determined by the neural network. The knowledge oriented ANN models proposed by this study have many desirable properties compared to standard ANNs like arbitrage-free and nonnegative option values, theory consistent Greek letter etc. In general, this methodology is proposed as a way to eliminate some of the deficiencies of the modern parametric options models and the standard ANNs.

We compare the proposed methodology with standard ANNs and with the Black and Scholes and the Corrado and Su models. For pricing performance analysis we use the S&P 500 index call options, with both historical and several overall average implied parameters for the period January 2002 to August 2004.

The results obtained strongly support the proposed methodology. Specifically, we find that the increase in the pricing accuracy of KANN-BS over the standard BS models is between 23% and 61%. The increase of the pricing accuracy of the KANN-CS over the CS model is between 10% and 17%. Compared to the standard ANNs, the increase in pricing accuracy is even more significant.

Literature

Adesi, B. G., Rasmussen, H., and Ravanelli, C. (2005). An Option Pricing Formula for the GARCH Diffusion Model. *Computational Statsistics & Data Analysis*, vol. 49, pg. 287-310.

Anders, U., Korn, O., and Schmitt, C. (1998). Improving the Pricing of Options: A Neural Network Approach. *Journal of forecasting*, 17, 369-388.

Andersen, T.G., Benzoni, L., and Lund, J. (2002) "An Empirical Investigation of Continuous-Time Equity Return Models", *Journal of Finance*, vol. 57, no. 3, pp. 1239-1276.

Andreou, P.C., Charalambous, C. and Martzoukos, S.H. (2005). Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters. To be published in the *European Journal of Operational Research*.

Backus, D., Foresi, S., Li, K., and Wu, L. (1997). Accounting for Biases in Black-Scholes. *Working Paper*, Stern School of Business.

Bakshi, G., Cao, C., and Chen, Z. (1997) "Empirical Performance of Alternative Options Pricing Models", *Journal of Finance*, vol. 52, no. 5, pp. 2003-2049.

Bandler, W. J, Biernacki, M. R., Chen, H. S., Grobenly, A., P, and Hemmers, H. R. (1994). Space Mapping Technique for Electromagnetic Optimization. *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 12, pg. 2536-2544.

Bandler, W. J, Ismail, A. M., Sanchez E.R. J., and Zhang, J. Q (1999). Neuromodeling of Microwave Circuits Exploiting Space-Mapping Technology. *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, pg. 2417-2427.

Bates, D. S. (1991) "The Crash of '87: Was it Expected? The Evidence From Options Markets," *Journal of Finance*, vol. 46, no. 3, pp. 1009-1044.

Bates, D. S. (1996) "Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options" *The Review of Financial Studies*, vol. 9, no. 1, pp. 69-107.

Black, F. and Scholes, M. (1973) "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, vol. 81, pp. 637-654.

Brown, C. and Robinson, D. (2002) "Skewness and Kurtosis Implied by Option Prices: A Correction", *Journal of Financial Research*, vol. 25, no. 2, 279-282.

Charalambous, C., (1992) "A Conjugate Gradient Algorithm for Efficient Trading of Artificial Neural Networks", IEE Proceedings – G, 139, pp. 301-310.

Cont, R., and Fonseca, J. (2002) "Dynamics of Implied Volatility Surfaces" *Quantitative Finance*, vol. 2, pp.45-60.

Corrado, C. J., and Su, T. (1996) "Skewness and Kurtosis in S&P 500 Index Returns Implied by Option Prices", *Journal of Financial Research*, vol. 19, no. 2, pp. 175-192.

Corrado, C. J., and Su, T. (1997) "Implied Volatility Skews and and Stock Index Skewness and Kurtosis Implied by S&P 500 Index Options", *Journal of Derivates*, vol. 4, pp. 8-19.

Cybenko, G., (1989) "Approximation by Superpositions of a Sigmoidal Function", *Mathematics of Control, Signal and Systems* 2, pp. 303-314.

Duda, O. R., Hart, E. P., and Stork, G. D. (2001). *Pattern Classification*. 2nd Edition, John Wiley and Sons, Inc.

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

Eraker, B. (2004). Do Stock Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices. *Journal of Finance*, vol. LIX, no. 3, pg. 1367-1403.

Garcia, R. and Gencay, R. (2000) "Pricing and Hedging Derivative Securities With Neural Networks and a Homogeneity Hint," *Journal of Econometrics*, vol. 94, pp. 93-115.

Hagan, M. T. and Menhaj, M. (1994) "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989-993.

Hagan, M., Demuth, H. and Beale, M. (1996) *Neural Network Design*, PWS Publishing Company.

Haykin, S. (1999). *Neural Netwroks – A Comprehensive Foundation*, 2nd Ed., Prentice Hall.

Herrman, R., and Narr, A. (1997). Neural Networks and the Valuation of Derivaties – Some Insights Into the Implied Pricing Mechanism of German Stock Index Options. University of Karlsruhe, Germany, Discussion Paper 2002.

Hutchison, J. M., Lo, A. W. and Poggio, T. (1994) "A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks", *Journal of Finance*, vol. 49, no. 3, pp. 851-889.

Jurczenko, E., Maillet, B., and Negrea, B. (2002). Revisited Muti-moment Aproximate Option Pricing Models: A General Comparison (Part I). *Working Paper*, LSE-FM.

Kiesel, R. (2002). Nonparametric Statistical Methods and the Pricing of Derivatives Securities. *Journal of applied Mathematical and Decision sciences*, 6(1), pg. 1-22.

Lajbcygier, P. (2004). Improving Option Pricing With the Product constrained Hybrid Neural Network, *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, pg.465-476.

Lajbcygier P., Boek C., Palaniswami M. and Flitman A. (1996) "Comparing Conventional and Artificial Neural Network Models For the Pricing of Options on Futures" *Neurovest Journal*, vol. 4, no. 5, pp. 16-24.

Lajbcygier, P., Flitman, A., Swan, A. and Hyndman, R. (1997) "The Pricing and Trading of Options Using a Hybrid Neural Network Model With Historical Volatility", *Neurovest Journal*, vol. 5, no. 1 pp. 27-41.

Lehar, A., Scheicher, M., and Schittenkopf., C. (2002). GARCH Vs. Stochastic Volatility: Option Pricing and Risk Management. *Journal of Banking and Finance*, 26, pg. 323-345.

Merton, R.C. (1973) "Theory of Rational Option Pricing," *Bell Journal of Economics and Management Science*, vol. 4, pp. 141-183.

Rubinstein, M. (1985) "Nonparametric Tests of Alternative Option Pricing Models Using All Reported Trades and Quotes on the 30 Most Active CBOE Option Classes From August 23, 1976 Through August 31, 1978", *The Journal of Finance*, vol. XL, pp. 455-480.

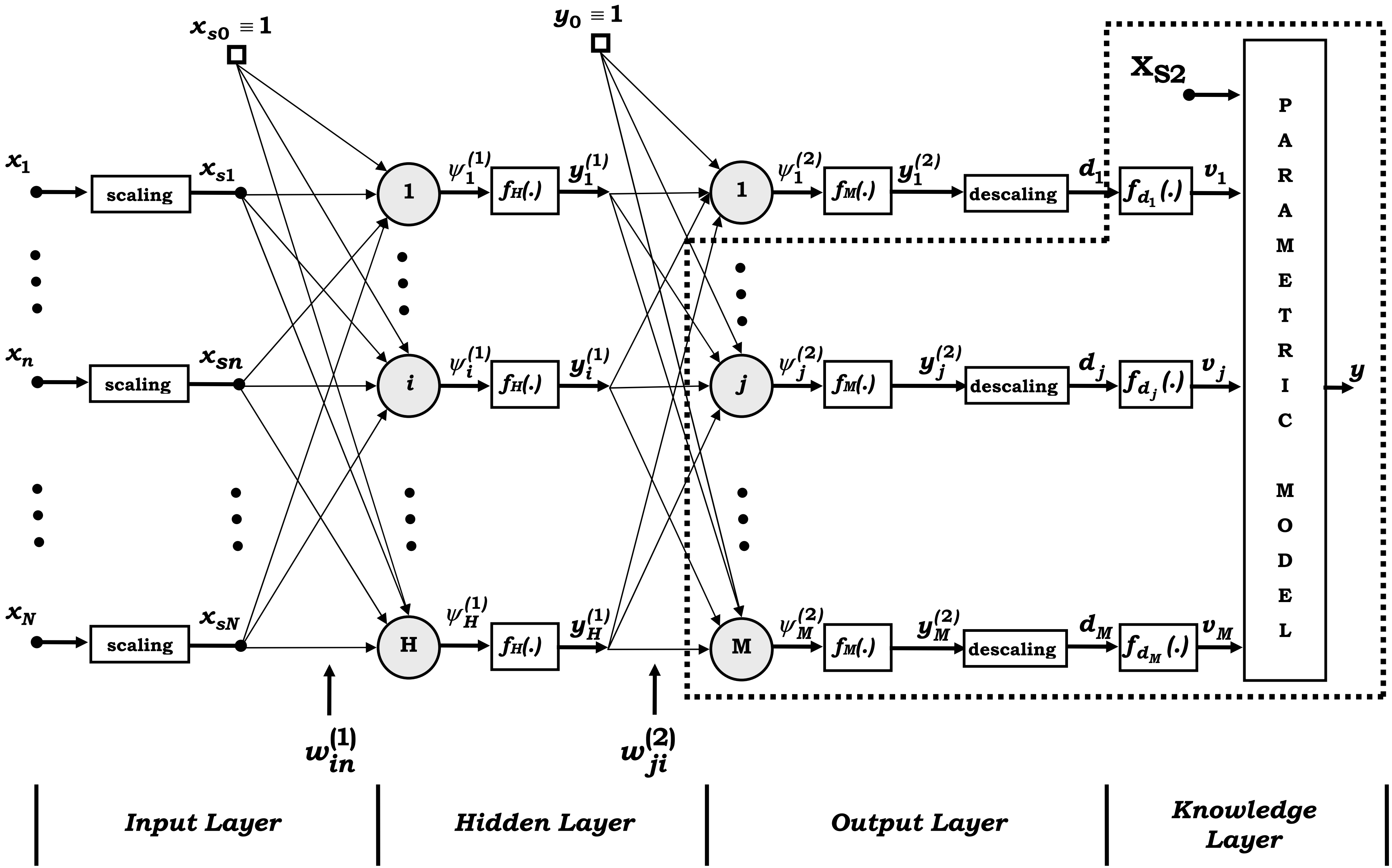
Sanchez., E.R.J. (2004). EM-Based Optimization of Microwave Circuits Using Artificial Neural Networks: The State of the Art. *IEEE Transactions on Microwave Theory and Techniques*, vol.52, no. 1, pg. 420-435.

Schittenkopf C. and Dorffner, G. (2001) "Risk-Neutral Density Extraction from Option Prices: Improved Pricing with Micture Density Networks", *IEEE Transactions on Neural Networks* vol. 12, no. 4, pp. 716-725.

Wang, F., and Zhang, J.Q. (1997). Knowledge-Based Neural Models for Microwave Design. *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pg. 2333-2343.

Watson, P. and Gupta, K.C. (1996) “EM-ANN Models for Microstript Vias and Interconnected in Multilayer Circuits”, IEEE Trans., Microwave Theory and Techniques, pp. 2495-2503.
Whaley, R.E. (1982) “Valuation of American Call Options on Dividend-Paying Stocks”, *Journal of Financial Economics*, vol. 10, pp. 29-58.

Figure 1



x_{s0} ≡ 1
y_0 ≡ 1
X_{S2}
x_1
x_n
x_N
scaling
scaling
scaling
x_{s1}
x_{sn}
x_{sN}
1
i
H
fH(.)
fH(.)
fH(.)
1
j
M
fM(.)
fM(.)
fM(.)
descaling
descaling
descaling
d_1
d_j
d_M
v_1
v_j
v_M
PARAMETRIC MODEL
y
w_{in}^{(1)}
w_{ji}^{(2)}
Input Layer
Hidden Layer
Output Layer
Knowledge Layer

Tables

Model	Enhanced Variable	Transfer Function	Parameter Values (a,b)
BS	Volatility	Logistic	(1.5,1)
CS	Volatility	Logistic	(1.5,1)
CS	Skewness	Tangent	(15,0.15)
CS	Kurtosis	Logistic	(30,0.20)

Panel A: *Transfer functions used with enhanced variables*

Model	Input Variables	Enhanced Variable(s)
nBS_{60}	$(Se^{-lT})/X$, T , r , s_{60}^{BS}	none
nBS_{vix}	$(Se^{-lT})/X$, T , r , s_{vix}^{BS}	none
nBS_{av}	$(Se^{-lT})/X$, T , r , s_{av}^{BS}	none
nBS_{vav}	$(Se^{-lT})/X$, T , r , s_{vav}^{BS}	none
nBS_{cm}	$(Se^{-lT})/X$, T , r , s_{cm}^{BS}	none
nBS_{vol}	$(Se^{-lT})/X$, T , r , s_{vol}^{BS}	none
nCS_{av}	$(Se^{-lT})/X$, T , r , s_{av}^{CS} , $m3_{av}^{CS}$, $m4_{av}^{CS}$	none
nCS_{vav}	$(Se^{-lT})/X$, T , r , s_{vav}^{CS} , $m3_{vav}^{CS}$, $m4_{vav}^{CS}$	none
kBS_{60}	$(Se^{-lT})/X$, T , r , s_{60}^{BS}	Volatility
kBS_{vix}	$(Se^{-lT})/X$, T , r , s_{vix}^{BS}	Volatility
kBS_{av}	$(Se^{-lT})/X$, T , r , s_{av}^{BS}	Volatility
kBS_{vav}	$(Se^{-lT})/X$, T , r , s_{vav}^{BS}	Volatility
kBS_{cm}	$(Se^{-lT})/X$, T , r , s_{cm}^{BS}	Volatility
kBS_{vol}	$(Se^{-lT})/X$, T , r , s_{vol}^{BS}	Volatility
kCS_{av}	$(Se^{-lT})/X$, T , r , s_{av}^{CS} , $m3_{av}^{CS}$, $m4_{av}^{CS}$	Volatility, skewness, kurtosis
kCS_{vav}	$(Se^{-lT})/X$, T , r , s_{vav}^{CS} , $m3_{vav}^{CS}$, $m4_{vav}^{CS}$	Volatility, skewness, kurtosis

Panel B: *Description on all ANN based models*

Table 1: Network characteristics

	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM
S/X	**≤0.90**	**0.90-0.95**	**0.95-0.99**	**0.99-1.01**	**1.01-1.05**	**1.05-1.10**	**>1.10**
Short Term Options <60 Days							
c^{mrk}	1.81	3.46	8.83	21.25	38.81	74.44	156.09
Volume	591	995	1274	1766	554	303	213
σ^{BS}_{60}	0.20	0.16	0.14	0.14	0.15	0.15	0.16
σ^{BS}_{vix}	0.28	0.22	0.19	0.19	0.19	0.20	0.21
σ^{BS}_{av}	0.23	0.20	0.18	0.17	0.18	0.18	0.19
σ^{BS}_{vav}	0.24	0.19	0.17	0.17	0.17	0.18	0.18
σ^{BS}_{cm}	0.23	0.18	0.16	0.16	0.16	0.17	0.17
σ^{BS}_{vol}	0.24	0.20	0.17	0.17	0.17	0.18	0.19
# obs	378	1696	3868	2237	2843	1332	857
Medium Term Options 60-180 Days							
c^{mrk}	4.09	10.40	23.76	39.93	55.62	86.72	176.02
Volume	287	565	650	1553	494	143	221
σ^{BS}_{60}	0.18	0.15	0.15	0.15	0.15	0.15	0.15
σ^{BS}_{vix}	0.23	0.19	0.19	0.19	0.19	0.19	0.20
σ^{BS}_{av}	0.21	0.18	0.18	0.18	0.18	0.18	0.18
σ^{BS}_{vav}	0.21	0.17	0.17	0.17	0.17	0.17	0.18
σ^{BS}_{cm}	0.19	0.16	0.16	0.16	0.17	0.16	0.17
σ^{BS}_{vol}	0.21	0.18	0.17	0.17	0.18	0.17	0.18
# obs	1165	1759	1682	980	1111	750	713
Long Term Options ≥ 180 Days							
c^{mrk}	10.21	27.24	45.51	61.12	76.52	102.84	175.32
Volume	343	478	341	471	200	110	185
σ^{BS}_{60}	0.18	0.16	0.15	0.15	0.16	0.15	0.16
σ^{BS}_{vix}	0.23	0.21	0.20	0.20	0.20	0.19	0.20
σ^{BS}_{av}	0.20	0.19	0.18	0.18	0.18	0.18	0.18
σ^{BS}_{vav}	0.20	0.18	0.18	0.17	0.18	0.17	0.18
σ^{BS}_{cm}	0.19	0.17	0.17	0.17	0.17	0.16	0.17
σ^{BS}_{vol}	0.21	0.19	0.18	0.18	0.18	0.17	0.18
# obs	1080	734	660	429	417	258	353

Table 2: Sample descriptive statistics

	BS_{60}	BS_{vix}	BS_{av}	BS_{vav}	BS_{cm}	BS_{vol}
RMSE	7.11	5.54	3.98	4.47	4.68	6.24
MAE	5.01	4.00	2.88	3.01	3.05	3.49
MeAE	3.40	3.01	2.17	1.87	1.67	1.90
P_5AE	0.22	0.38	0.21	0.17	0.12	0.15
P_{95}AE	15.59	11.43	8.19	10.02	10.53	11.63

Panel A: *Parametric Black and Scholes Pricing Performance*

	nBS_{60}	nBS_{vix}	nBS_{av}	nBS_{vav}	nBS_{cm}	nBS_{vol}
RMSE	7.89	5.12	4.54	4.78	4.56	5.45
MAE	5.82	3.28	3.02	3.26	2.77	3.56
MeAE	4.77	2.17	2.16	2.25	1.74	2.34
P_5AE	0.47	0.22	0.23	0.22	0.18	0.23
P_{95}AE	14.84	9.86	8.64	9.46	9.12	10.56

Panel B: *Standard ANNs Pricing Performance*

	kBS_{60}	kBS_{vix}	kBS_{av}	kBS_{vav}	kBS_{cm}	kBS_{vol}
RMSE	5.18	3.44	3.24	3.55	3.21	4.15
MAE	3.61	2.36	2.27	2.48	2.19	2.74
MeAE	2.41	1.62	1.64	1.75	1.51	1.90
P_5AE	0.20	0.12	0.13	0.13	0.11	0.15
P_{95}AE	11.38	7.23	6.66	7.42	6.59	8.48

Panel C: *KANNs Pricing Performance*

Table 3: Black and Scholes based parametric and network models

	CS_{av}	CS_{vav}
RMSE	3.21	3.88
MAE	2.22	2.51
MeAE	1.53	1.48
P_5AE	0.13	0.12
P_{95}AE	6.61	8.55

Panel A: *Parametric Corrado and Su Pricing Performance*

	nCS_{av}	nCS_{vav}
RMSE	4.79	5.50
MAE	3.23	3.37
MeAE	2.26	2.13
P_5AE	0.24	0.21
P_{95}AE	9.32	10.39

Panel B: *Standard ANNs Pricing Performance*

	kCS_{av}^{sig}	kCS_{vav}^{sig}	kCS_{av}^{all}	kCS_{vav}^{all}
RMSE	2.95	3.46	2.92	3.32
MAE	2.00	2.49	2.01	2.29
MeAE	1.35	1.80	1.40	1.60
P_5AE	0.13	0.15	0.12	0.15
P_{95}AE	6.08	7.07	5.97	6.71

Panel C: *KANNs Pricing Performance*

Table 4: Corrado and Su based parametric and network models