International Journal of Theoretical and Applied Finance
Vol. 3, No. 2 (2000) 183–204

PROFILING NEURAL NETWORKS FOR OPTION PRICING

A. CARELLI
Banca Intesa-Risk Management and Research,
Via Clerici 4, 20121 Milano, Italy

S. SILANI and F. STELLA
Dipartimento Di Informatica, Sistemistica e Comunicazione,
Viale Sarca 202, 20126 Milano, Italy

Received 22 October 1998

In recent years the problem of option pricing has received increasing interest from both financial institutions and academics. It is well known that conventional modeling techniques for option pricing have inherent, persistent and systematic biases which are mainly due to the assumption of constant volatility for prices associated with the underlying financial instrument. Nowadays, there is strong and increasing evidence that financial markets are far from being stationary and then, whenever dealing with option pricing, we have to take into account the market heteroschedasticity. A possible approach for dealing with non-constant volatility relies on the modeling of the basic characteristics named implied volatility. Unfortunately this task is extremely complex and parametric models are not available. In this paper the authors discuss how models from the class of Feedforward Neural Networks can be exploited for approaching the task of implied volatility modeling. In particular the paper shows how the main techniques from the nonlinear regression framework can be exploited when models from the class of Feedforward Neural Networks are used. Indeed, in such a case the paucity of data, which can be used for the network training, and the particular structure of Feedforward Neural Networks make the modeling task numerically complex. The authors discuss how the nonlinear regression technique named profile can be exploited for selecting the optimal network's structure and evaluating its numeical properties. To this end, a numerical procedure for empirical model building, in the case of Feedforward Neural Networks, has been developed. Results are evaluated through an ad-hoc procedure which utilizes the estimated implied volatility surface for pricing general contingent claims. Numerical experiments, in the case of the USD/DEM options, are presented and discussed.

1. Introduction

It is well known that conventional modeling techniques for option pricing have inherent, persistent and systematic biases which are mainly due to the assumption of constant volatility for prices associated with the underlying financial instrument. Nowadays, there is increasing evidence that financial markets are far from being stationary and then, whenever dealing with option pricing, we have to take into a proper account the *market heteroschedasticity*. A possible approach consists of developing a model for the characteristics named *implied volatility* and then to

use it within traditional scheme for option pricing. Unfortunately such an approach is extremely complex and parametric models for the implied volatility are not available.

Recently, nonparametric regression models have been investigated for different aspects concerning option pricing and some evidence exists that *Feedforward Neural Networks* (FNNs) are the most promising ones. The main property of FNNs is that they are universal approximators in the sense described by White [1, 2]. This property establishes that FNNs, with as many as one hidden layer, are capable to approximate, to any degree of accuracy, any mapping between independent variables and response variables as the size of the data set, describing the *Data Generating Process* (DGP), and the number of neurons of the hidden layer go to infinity. While today the limitation in the number of neurons is less keenly felt, due to technological enhancements and costs reduction, the problem of paucity of data is still the main issue. Indeed, in actual applications and in particular in the case of implied volatility modeling, data are difficult and expensive to get and for this reason it is important that the FNN model is *data-efficient*, i.e., it squeezes most information out of few data. Unfortunately, the universal approximation property, enjoyed by FNNs, is an asymptotic result of no help in actual problems in which, quite to the contrary, an increase in the number of parameters leads, too often, to overfitting and overparametrization. In such a framework the problem is twofold:

(i) to develop efficient algorithms for the network's structure selection,
(ii) to design procedures for assessing the model performances on new data.

Several theoretical and empirical approaches have been presented and discussed in the literature for approaching these problems [3–14] and a discussion about their main characteristics and limitations is out of the scope of this paper.

In this paper the authors investigate further the class of FNNs to the extent of modeling the implied volatility. In particular, a procedure for the network's structure selection is presented. This procedure relies on both graphical and numerical techniques from nonlinear regression and allows to control the network's complexity/flexibility in such a way to reduce the risk of overfitting. The main idea, behind the proposed approach, is to evaluate the usefulness of neurons, from the hidden layers, for providing a satisfactory approximation for the response variable.

The rest of the paper is organized as follows. In Sec. 2 the main definitions about FNNs are given together with their main properties. Section 3 introduces the numerical and graphical procedures, from the nonlinear regression framework, which are the basic tools for dealing with the problem of the network's structure selection. This section presents and discusses a numerical procedure for the network's structure selection in the case of single layer FNNs. Numerical experiments, for the USD/DEM option, are reported in Sec. 4. Furthermore, through this section, a computational scheme for pricing general contingent claim is described. Such a scheme is used to the extent of evaluating the effectiveness of the proposed approach for the network's structure selection. Finally, Sec. 5 is devoted to comments

and conclusions about the proposed approach for the implied volatility modeling through FNNs.

2. Feedforward Neural Networks

An FNN (Fig. 1) is a layered graphical structure consisting of computing units named *artificial neurons* (circles) and connections between neurons named *synapses* (directed links).

Artificial neurons, or simply neurons, of the *input layer* are associated with the independent variables, namely the *input variables*; while the *output neurons* are associated with the response variables, namely the *output variables*. Synapses are oriented connections linking the neurons from the input layer to the neurons of the hidden layer and the neurons from the hidden layer to the output neuron. This means that the information flows from the neurons of the lower layers to the neurons of the higher layers and cannot flow between the neurons of the same layer or between a higher layer to the neurons of the lower layers. The strength of the connection from the neuron i to the neuron j is determined by means of a real value w_{ij}, named *synapse's weight* or simply *weight*. Furthermore, each neuron j in the hidden layer, and eventually the output neuron, are associated with a real value b_j, named *neuron's bias* or *threshold*. Finally, the neurons from the hidden layer and eventually the output neuron are associated with a nonlinear function, named *transfer* or *activation function*.

Let us formally describe the structure of FNNs and how, given an input vector, the network's output is computed. To this end, consider the single layer FNN with K input variables and H hidden neurons depicted in Fig. 1 and let:

- $\mathbf{x} = (x_1, \ldots, x_K)$, be the input vector (associated with independent variables);
- $w_{ij}^{(1)}$, be the weight associated with the link from the i^{th} input to the j^{th} hidden neuron;

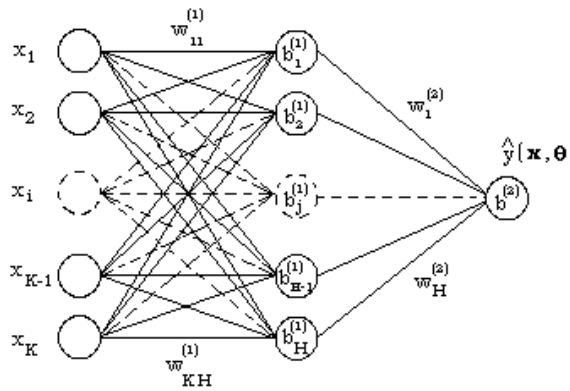


Fig. 1. Single layer Feedforward Neural Network.

- $b_j^{(1)}$, be the bias associated with the j^{th} neuron of the hidden layer;
- $w_j^{(2)}$, be the weight associated with the link from the j^{th} hidden neuron to the output neuron;
- $b^{(2)}$, be the bias associated with the output neuron.

Furthermore, let $\mathcal{G}^{(1)}(\cdot)$ and $\mathcal{G}^{(2)}(\cdot)$ be the activation functions associated with the hidden neurons and the output neuron. Typical choices include:

- *logistic* $\mathcal{G}(z) = \frac{1}{1+\exp(-z)}$,
- *hyperbolic tangent* $\mathcal{G}(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$.

Let the vector $\boldsymbol{\theta}$ be defined as follows:

$$\boldsymbol{\theta} = (w_{11}^{(1)}, \ldots, w_{1H}^{(1)}, \ldots, w_{K1}^{(1)}, \ldots, w_{KH}^{(1)}, b_1^{(1)}, \ldots, b_H^{(1)}, w_1^{(2)}, \ldots, w_H^{(2)}, b^{(2)}) . \tag{2.1}$$

and assume that the activation functions $\mathcal{G}^{(1)}(\cdot)$ and $\mathcal{G}^{(2)}(\cdot)$ are given, then the network in Fig. 1 computes the following function of the input vector $\mathbf{x}$:

$$\hat{y}(\mathbf{x}, \boldsymbol{\theta}) = \mathcal{G}^{(2)} \left(\sum_{j=1}^{H} \left(w_j^{(2)} \cdot \mathcal{G}^{(1)} \left(\sum_{i=1}^{K} w_{ij}^{(1)} \cdot x_i - b_j^{(1)} \right) \right) - b^{(2)} \right) . \tag{2.2}$$

In the rest of the paper $\phi(|\mathbf{x}|, H)$ represents the structure of a single layer FNN having the elements of the vector $\mathbf{x}$ as inputs, H hidden neurons and a single output neuron, analogously $\phi(|\mathbf{x}|, H^{(1)}, H^{(2)})$ represents the structure of an FNN consisting of $H^{(1)}$ and $H^{(2)}$ neurons in the first and second hidden layer.

Let us focus the attention to the problem of the network's structure selection. Informally, given a set of experimental observations, this problem consists of building an FNN model such that it will be capable to describe the main features of the DGP. In order to solve this problem, we have to answer the following questions:

(i) How many hidden layers are needed?
(ii) How many neurons for each hidden layer are needed?

The above questions are not trivial and a great deal has been devoted to them in the specialized literature.

Answering the above questions requires the solution of the so-called *training problem*. This problem can be formally introduced as follows. Let $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ be the data set, where $\mathbf{X} \in \Re^N \times \Re^K$ is called the input matrix, $\mathbf{Y} \in \Re^N$ is called the output vector and $N \in \mathbf{N}$ is the number of observations. Notice that the rows $\mathbf{x}_n,\ n = 1, 2, \ldots, N$ of the matrix $\mathbf{X}$ are vectors in the input space $\Re^K$; while the elements $y_n \in \Re,\ n = 1, 2, \ldots, N$ of the $\mathbf{Y}$ vector represent observations of the dependent variable. Then, for any given network's structure, the training problem consists of finding the parameter vector $\hat{\boldsymbol{\theta}}$ which solves the following nonlinear least squares problem:

$$E(\hat{\boldsymbol{\theta}}) = \min_{\theta} \sum_{n=1}^{N} (\hat{y}(\mathbf{x}_n, \boldsymbol{\theta}) - y_n)^2 . \tag{2.3}$$

Problem (2.3), in the contest of FNNs, is usually solved by means of algorithms such as *backpropagation* [14–17] or its main variant named *momentum learning* [18]. Also numerical methods from nonlinear regression as *Gauss–Newton*, *Newton–Raphson* and *Levenberg–Marquardt* [19, 20] can be used.

3. Profiling and Network's Structure Selection

As mentioned, the problem of the network's structure selection requires the solution of several training problems on different FNNs and so to cope with the nonlinear regression analysis. For this reason, the formal nonlinear regression framework, i.e. the nonlinear regression model and the consequent notation are hereafter introduced. To this end, let $\mathbf{D}$, $\mathbf{X}$ and $\mathbf{Y}$ be defined as in the previous section. Then, according to [21], a nonlinear regression model can be written as follows:

$$y_n = f(\mathbf{x}_n, \boldsymbol{\theta}) + z_n \tag{3.1}$$

where $f(\cdot, \cdot)$ refers to the "*expectation function*", $\boldsymbol{\theta} \in \Re^P$ represents the "*parameter vector*" and z_n is a "*disturbance term*".

The nonlinear model (3.1) differs from the linear one in the sense that at least one of the derivatives of the expectation function $f(\cdot, \cdot)$ with respect to the parameters $\boldsymbol{\theta}$ depends on at least one of the parameters. When analyzing a given data set $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, we can create the "*expectation surface*", i.e. the N-vector $\boldsymbol{\eta}(\boldsymbol{\theta})$ having the n^{th} element given by

$$\boldsymbol{\eta}_n(\boldsymbol{\theta}) = f(\mathbf{x}_n, \boldsymbol{\theta}) \quad n = 1, \ldots, N \tag{3.2}$$

and write the nonlinear regression model (3.1) as

$$\mathbf{Y} = \boldsymbol{\eta}(\boldsymbol{\theta}) + \mathbf{Z} \tag{3.3}$$

where $\mathbf{Z}$ is normally distributed, i.e.

$$E[\mathbf{Z}] = 0\,, \quad \text{Var}[\mathbf{Z}] = E[\mathbf{Z}\mathbf{Z}^T] = \sigma^2 \mathbf{I}\,. \tag{3.4}$$

For any given data set $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, solving the problem (2.3) requires the determination of the expectation surface $\boldsymbol{\eta}(\boldsymbol{\theta})$.

This task is made up of two main components:

(i) to specify the nonlinear regression model $f(\cdot, \cdot)$;
(ii) to select the vector $\boldsymbol{\theta}$ such that the Sum of Squared Errors (SSE):

$$S(\boldsymbol{\theta}) = \|\mathbf{Y} - \eta(\boldsymbol{\theta})\|_2^2 \tag{3.5}$$

is minimized, where $\|\cdot\|_2$ is the L_2 norm.

Once an estimate $\hat{\boldsymbol{\theta}}$ for the parameter vector $\boldsymbol{\theta}$ has been determined it is helpful to summarize the estimation situation. To this end, we can compute "*likelihood regions for parameters*". Unfortunately, for the model (3.1), analytic expressions are not available. This can be dealt with by means of two approaches:

(i) to use approximate likelihood regions;
(ii) to compute exact likelihood regions.

The first approach allows to compute the "*approximate joint likelihood regions*" as

$$(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \hat{\mathbf{V}}^T \hat{\mathbf{V}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \leq P s^2 F(P, N-P; \alpha) \tag{3.6}$$

where:

- $\hat{\mathbf{V}}$, is the $N \times P$ derivative matrix having elements $\{v_{np} = \frac{\partial f(\mathbf{x}_n, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_p} \mid_{\hat{\boldsymbol{\theta}}}\}$;
- s^2, is defined as $s^2 = \frac{S(\hat{\boldsymbol{\theta}})}{N-P}$;
- $F(P, N-P; \alpha)$, represents the upper α quantile for the Fisher's F distribution with P and $N-P$ degrees of freedom.

This approach provides approximate likelihood regions which are easy to calculate for any number of parameters. Unfortunately it can be extremely misleading [21–24], because it is based on only one approximation (at $\hat{\boldsymbol{\theta}}$), and therefore could drive to draw wrong conclusions about the significance of the model's parameters.

The second approach consists of directly computing the "*exact joint likelihood regions*", namely the set of all values $\boldsymbol{\theta}$ such that

$$S(\boldsymbol{\theta}) \leq S(\hat{\boldsymbol{\theta}}) \left[1 + \frac{P}{N-P} F(P, N-P; \alpha) \right] . \tag{3.7}$$

Together with the joint likelihood region, we can develope "*marginal likelihood intervals*" for each model's parameter θ_p, $p = 1, 2, \ldots, P$. This can be done according to what presented and discussed in [21, 22]. In order to define the marginal likelihood interval for the parameter θ_p, we first have to define some useful quantities.

Definition 3.1 [21]. Let $\boldsymbol{\theta}_{-\mathbf{p}} = (\theta_1, \ldots, \theta_{p-1}, \theta_{p+1}, \ldots, \theta_P)$ and the vector $\tilde{\boldsymbol{\theta}}_{-p} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_{p-1}, \tilde{\theta}_{p+1}, \ldots, \tilde{\theta}_P)$ be the least squares estimate of $\boldsymbol{\theta}_{-p}$ conditional on θ_p, $(\theta_p, \tilde{\boldsymbol{\theta}}_{-p})$ be the vector with elements $(\tilde{\theta}_1, \ldots, \tilde{\theta}_{p-1}, \theta_p, \tilde{\theta}_{p+1}, \ldots, \tilde{\theta}_P)$, then the profile sum of squares function $\tilde{S}(\theta_p)$ for the parameter θ_p is defined as

$$\tilde{S}(\theta_p) = \min_{\boldsymbol{\theta}_{-p}} S((\theta_p, \boldsymbol{\theta}_{-p})) = S((\theta_p, \tilde{\boldsymbol{\theta}}_{-p})) . \tag{3.8}$$

Definition 3.2 [21]. For any parameter θ_p, of the model (3.1), the profile t function $\tau(\theta_p)$ is defined as follows:

$$\tau(\theta_p) = \text{sign}(\theta_p - \hat{\theta}_p) \frac{\sqrt{\tilde{S}(\theta_p) - S(\hat{\boldsymbol{\theta}})}}{s} . \tag{3.9}$$

Definition 3.3 [21]. For any parameter θ_p, of the model (3.1), its α marginal likelihood interval is the set of all θ such that

$$-t \left[N-P; \left(1 - \frac{(1-\alpha)}{2} \right) \right] \leq \tau(\theta_p) \leq t \left[N-P; \left(1 - \frac{(1-\alpha)}{2} \right) \right] \tag{3.10}$$

where $t[N-P;(1-\frac{(1-\alpha)}{2})]$, represents the upper $(1-\frac{(1-\alpha)}{2})$ quantile for the Student's t distribution with $N-P$ degrees of freedom.

The importance of the *profile t function*, in the case of FNNs, is twofold:

- plots of the *profile t function* provide *exact likelihood intervals* for parameters,
- plots of the *profile t function* reveal how nonlinear the parameter is.

Suppose the model were linear. Then, a plot of $\tau(\theta_p)$ versus θ_p would be a straight line. In particular, a plot of $\tau(\theta_p)$ versus the *studentized parameter* $\delta(\theta_p)=\frac{\theta_p-\hat{\theta}_p}{se(\hat{\theta}_p)}$, would be a straight line through the origin with unit slope. Indeed, in a linear model the following equality holds [21]:

$$\delta(\theta_p)=\frac{\theta_p-\hat{\theta}_p}{se(\hat{\theta}_p)}=\text{sign}(\theta_p-\hat{\theta}_p)\frac{\sqrt{\tilde{S}(\theta_p)-S(\hat{\boldsymbol{\theta}})}}{s}=\tau(\theta_p)\,. \tag{3.11}$$

On the other hand, in the case when a nonlinear model is considered, the equality (3.11) does not hold, thus a plot of $\tau(\theta_p)$ versus $\delta(\theta_p)$ will be curved. The amount of curvature carries information about the degree of nonlinearity for the model's parameter θ_p.

Let us now clarify how Definitions 3.2 and 3.3 can be exploited to approach the problem of the network's structure selection. To this end, consider the elementary computing unit of FNNs, i.e. the artificial neuron. (Figure 2) and notice that it is characterized by the following set of free parameters:

- $\mathbf{W}_{\text{in}}=\{w_{sj}\mid s=1,2,\ldots,\ S\}$, input weights,
- $\mathbf{W}_{\text{out}}=\{w_{jr}\mid r=1,2,\ldots,\ R\}$, output weights,
- b_j, bias or threshold.

Therefore its particular structure dictates that, for any given network, it will not be able to contribute to the output value if both the following conditions are met:

(i) all its input weights $w_{sj}\in\mathbf{W}_{\text{in}}$ are equal to zero,
(ii) all its output weights $w_{jr}\in\mathbf{W}_{\text{out}}$ are equal to zero.

In the case when the previous conditions are satisfied, the artificial neuron will have no effect on the network's output, i.e. the neuron will be useless to the extent of

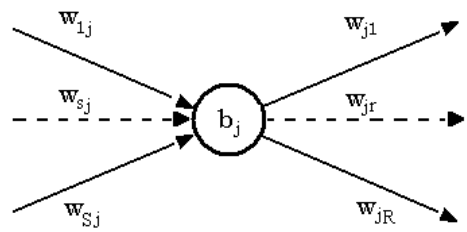


Fig. 2. Artificial neuron.

modeling the unknown relationship between the independent variables (network's input) and the response variable (network's output). However, when solving the training problem for FNNs, it will be very unlikely that some of the neuron's weights and/or its bias will be exactly equal to zero. Then, in order to evaluate the neuron usefulness, we need to provide some procedure for deciding whether or not a given neuron's parameter is equal to zero. This can be done by means of standard statistical tools as likelihood regions or equivalently hypothesis testing. In order to establish whether or not a given neuron is useful we introduce the following definition of α-non rejectable neuron.

Definition 3.4. Let $\mathbf{W}_{\text{in}} = \{w_{sj} \mid s = 1, \ldots, S\}$ and $\mathbf{W}_{\text{out}} = \{w_{jr} \mid r = 1, \ldots, R\}$ denote respectively the set of input and output weights for a given neuron. Then, for any fixed level of confidence α, the neuron is said to be α-non rejectable if at least one of the following conditions holds:

$$\underline{0} \notin JLI(\mathbf{W}_{\text{in}}, \alpha) \tag{3.12}$$

$$\underline{0} \notin JLI(\mathbf{W}_{\text{out}}, \alpha) \tag{3.13}$$

where $JLI(\cdot, \alpha)$ represents the α level joint likelihood region.

Let us now discuss the meaning and usefulness of Definition 3.4 for approaching the problem of the network's structure selection. On the above definition, we shall build a tool for deciding whether a given artificial neuron is useful for approximating the unknown mapping between the input variables and the network's output. Indeed, we can state what follows:

- If either condition (3.12) or condition (3.13) are satisfied then there is no statistical evidence, at the α level of significance, that the neuron will be useless to the extent of approximating the response function.
- If neither condition (3.12) nor condition (3.13) are satisfied, then there is no statistical evidence, at the α level of significance, that the neuron will be useful to the extent of approximating the response function.

As can be noticed from Definition 3.4, establishing whether or not a given neuron is *α-non rejectable* requires to perform joint hypothesis testing in the case when dependent parameters are present. Then, to ensure that the real confidence level for the test of hypothesis is at least α, we have to use appropriate procedures such as Bonferroni's intervals [25].

Furthermore, having fixed the confidence level α, the significance of any given neuron tells us nothing about the overall significance of the network. Indeed, the neurons are far to be mutually independent.

Another problem consists of deciding the most complex FNN which should be trained. A possible approach, for answering this question, is as follows. Assume the data set consists of N observations, K independent variables and one response variable, then a reasonable approach will be to impose that the maximum number of network's parameters should be strictly less than N. In the case of single layer

FNNs, the maximum number of hidden neurons is

$$H_{\max} = \left[\frac{N-1}{(K+2)} \right] \tag{3.14}$$

while in the case of two layers FNNs, when the first layer consists of $H^{(1)}$ neurons, the maximum number of neurons for the second hidden layer is

$$H^{(2)}_{\max} = \left[\frac{N - (H^{(1)} \cdot (K+1) + 1)}{(H^{(1)} + 2)} \right] . \tag{3.15}$$

Finally, concerning the choice of a suitable value for the maximum number of network's free parameters $P_{\max}$, some help is readily available from a time honored statistical tradition. Indeed, a reasonable ratio of the number of observations over the maximum number of free parameters should range from 3 to 5 in order to obtain good statistical properties. In the light of such consideration, we can determine the maximum number of hidden neurons, for both the first and the second hidden layer by replacing N with $P_{\max}$ in (3.14) and (3.15).

Let us conclude this section by introducing a procedure for the network's structure selection which exploits Definition 3.4 in the case when single layer FNNs are considered.

Procedure. *Single Layer Network's Structure Selection*

(i) *Initialization.* Let $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ be the set of the available data points. Fix the desired level of confidence α, the maximum number of neurons $H_{\max}$ according to (3.14), where N has been replaced by $P_{\max}$, and initialize the set of candidate networks to the empty set, i.e. set $\Phi^{\mathbf{D}}_{\alpha} = \{\emptyset\}$. Furthermore, set the current number of hidden neurons to zero, i.e. $H = 0$.

(ii) *New Iteration.* Increase the number of hidden neurons, i.e. set $H = H + 1$. If $H > H_{\max}$, then go to Step (vi). Otherwise, initialize the parameter vector $\boldsymbol{\theta}$ with random values, and go to Step (iii).

(iii) *Network Training.* Solve the Problem (2.3) for the network $\phi(|\mathbf{x}|, H)$, record the solution in the vector $\hat{\boldsymbol{\theta}}$ and go to Step (iv).

(iv) *Network Profiling.* For each free parameter $\theta_p \in \hat{\boldsymbol{\theta}}$ associated with the network $\phi(|\mathbf{x}|, H)$, compute the corresponding profile t function and go to Step (v).

(v) *α-Hidden Neuron Checking.* If all the hidden neurons for the network $\phi(|\mathbf{x}|, H)$ are *α-non rejectable* then add $\phi(|\mathbf{x}|, H)$ to the set of the candidate networks, i.e. set $\Phi^{\mathbf{D}}_{\alpha} = \Phi^{\mathbf{D}}_{\alpha} \cup \phi(|\mathbf{x}|, H)$ and go to Step (ii). Otherwise, discard $\phi(|\mathbf{x}|, H)$, and go to Step (ii).

(vi) *Network Selection.* Among the FNNs belonging to the set $\Phi^{\mathbf{D}}_{\alpha}$, select the network $\phi^*(|\mathbf{x}|, H)$ having the smallest SSE (3.5).

(vii) *Network Criticism.* Analyze the residuals obtained through $\phi^*(|\mathbf{x}|, H)$. If statistical evidence exists that some of the assumptions (3.4) are violated, then remove the current network from the set of the candidate networks, i.e. $\Phi^{\mathbf{D}}_{\alpha} = \Phi^{\mathbf{D}}_{\alpha} \setminus \phi^*(|\mathbf{x}|, H)$, and go to Step (vi). Otherwise retain the current network for prediction purposes.

A similar argument allows to define a general procedure for the network's structure selection in the case when two layer FNNs are considered.

Let us discuss the main characteristics of the procedure just introduced.

(i) The proposed approach utilizes all the available data; indeed, by using Definition 3.4, it is possible to evaluate the extent to which a neuron is added and whether it is useful. This allows to deal with situations in which the amount of the available data is small and no further data points can be collected.

(ii) The procedure could also reveal that the quality of the data points at hand is poor. In such a case the overall approach can be fruitfully exploited for suggesting which new data points should be collected in order to improve the results of the modeling task. Notice that this could also mean that, by using only the available data points, we are not able to give any answers concerning the main characteristics of the DGP. This should not be taken as a failure, but carries valuable information and protect us against misunderstandings.

(iii) The sequence of tests and inspections to which the model is submitted can increase or decrease the confidence in its capability to have good performances on new cases.

(iv) When dealing with models of increasing complexity it will be more and more difficult to find good approximations for the optimal solution by means of iterative methods. In such cases, it will be useful to check the particular nature of the found solution, i.e. how the SSE (3.5) behaves around the given approximation of the optimal solution.

(v) Once the candidate model has been validated confidence bands for the expectation function can be provided.

(vi) Last but not least, a well-known criteria like the Occam razor is naturally embedded in the procedure we have presented. Indeed, neurons are not to be added without necessity.

The above numerical procedure can be jointly used with the graphical summaries for the network's parameters (*Profile t function*). Indeed, the visual inspection of the *profile t function* for the network's parameters can reveal how nonlinear the estimation situation is.

Graphical summaries carry useful information for evaluating which are the most important input variables to the extent of modeling the response variable. Furthermore they can be very useful for finding physical explanations for the phenomena under study.

4. Implied Volatility and Option Pricing

In this section the problem of approximating the implied volatility surface via FNNs is addressed and analyzed. To this end, numerical experiments, related to the case of the USD/DEM option, are presented and discussed.

The performance of the FNN, used for approximating the implied volatility, is assessed through an ad-hoc procedure. This procedure has been developed in a

previous work, jointly with a different learning algorithm for FNNs (see [26, 27] for details). It represents a new approach for contingent claim pricing, in which no parametric modelling for the market is supposed to exist, but prices are directly *made* by the market itself.

The basic idea is to avoid an *a priori* hypothetical model for the market, but to directly get all the required information from the market prices of the traded options, and to use such information for building a price process satisfying the following conditions:

(i) smiles compatibility,
(ii) model completeness.

More precisely, given the option prices for the entire set of strikes and maturities, the objective is to determine a risk-neutral process for the security S of the form:

$$\frac{dS}{S} = r(t)dt + \sigma(S,t)dW \tag{4.1}$$

where $r(t)$ represents the cost of carry and the local volatility $\sigma(S,t)$ is a deterministic function of both the time and the underlying level price.

If the spot price follows a diffusion price of this type, then the model is complete and the option prices can be computed via a risk-neutral valuation principle.

In order to develop a pricing system, based on the market data, we have to overcome the following problems:

(i) the available prices represent a restricted subset of strikes and maturities,
(ii) it is necessary to pass from the prices set to the local volatility surface $\sigma(S,t)$,
(iii) price computation must be done via a discrete scheme able to model the process (4.1).

A possible procedure for dealing with these problems consists of the following 5 steps (Fig. 3):

(i) computation of the implied volatility surface starting from the market prices,
(ii) computation of the option prices surface starting from the implied volatility surface,
(iii) computation of the local volatility surface starting from the option prices surface,
(iv) mapping of the local volatility surface on a discrete model,
(v) computation of the option prices and comparison with the original ones.

Let us now briefly discuss the main steps of such a pricing system.

(i) The first step is an extremely complex modeling task. Indeed, parametric models are not available and the data paucity increases further its complexity. The approach proposed through this paper is concerned with this modeling task which is addressed through FNNs. Unfortunately the success of FNNs strongly depends from the choice of the appropriate network's structure. This problem is recognized as very complex and the data paucity increases its

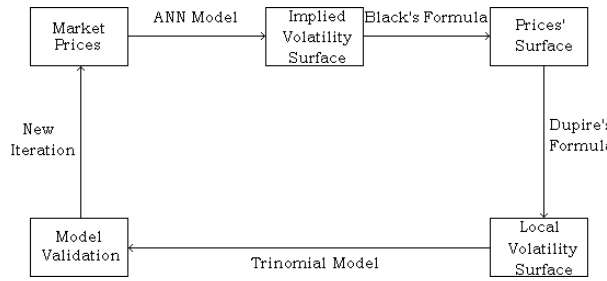


Fig. 3. Pricing system outline.

complexity. Furthermore, data paucity makes the classical algorithms for network's structure selection extremely difficult and expensive to apply.

(ii) The second step is realized using the celebrated Black's formula [28].

(iii) The third step is one of the most critical of the entire procedure. It is the phase concerned with the computation of the local volatility surface starting from the option prices' surface. The solution to this problem has been inspired by the work of Dupire [29, 30], Derman and Kani [31] and Derman, Kani and Zou [32]. Their works are motivated by the observation that, whenever the markets are complete and the price process is Markovian, the price process conditional distribution (for any fixed time) and the local volatility surface can be computed as explicit functions of the gradient of the price function with respect to the strike and the time to maturity.

(iv) The fourth step, concerning the modeling of the resulting process described by (4.1), has been solved using a trinomial scheme which allows to assign to each node of the tree a different local volatility value according to its time and spot level [33].

(v) Finally the comparison between the original option prices and the ones resulting from the overall procedure is accomplished in order to evaluate the usefulness of the overall approach and in particular represents a measure of the precision which can be obtained through FNNs modeling.

The pricing system must satisfy the following two requirements:

(i) to "price" correctly (all) the traded instruments in the same asset class,
(ii) to satisfy "*stylized facts*" (beliefs),

and is used throughout this section not only as the basic tool for option pricing but also for validating further the approach introduced in the previous section for the network's structure selection.

To this end, the problem of option pricing for the USD/DEM option is considered. In particular a set of 25 USD/DEM OTC (Table 1), where the options correspond to 20-delta and 25-delta USD/DEM puts and calls and 50-delta calls, is considered.

Table 1. USD/DEM option market prices.

Maturity	Type	Strike	Bid	Offer	Mid	IVOL
	Call	1.5421	0.0064	0.0076	0.0070	14.9
	Call	1.5310	0.0086	0.0100	0.0093	14.8
30 days	Call	1.4872	0.0230	0.0238	0.0234	14.0
	Put	1.4479	0.0085	0.0098	0.0092	14.2
	Put	1.4371	0.0063	0.0074	0.0069	14.4
	Call	1.5621	0.0086	0.0102	0.0094	14.4
	Call	1.5469	0.0116	0.0135	0.0126	14.5
60 days	Call	1.4866	0.0313	0.0325	0.0319	13.8
	Put	1.4312	0.0118	0.0137	0.0128	14.0
	Put	1.4178	0.0087	0.0113	0.0100	14.2
	Call	1.5764	0.0101	0.0122	0.0112	14.1
	Call	1.5580	0.0137	0.0160	0.0149	14.1
90 days	Call	1.4856	0.0370	0.0385	0.0378	13.5
	Put	1.4197	0.0141	0.0164	0.0153	13.6
	Put	1.4038	0.0104	0.0124	0.0114	13.6
	Call	1.6025	0.0129	0.0152	0.0141	13.1
	Call	1.5779	0.0175	0.0207	0.0191	13.1
180 days	Call	1.4823	0.0494	0.0515	0.0505	13.1
	Put	1.3902	0.0200	0.0232	0.0216	13.7
	Put	1.3682	0.0147	0.0176	0.0162	13.7
	Call	1.6297	0.0156	0.0190	0.0173	13.3
	Call	1.5988	0.0211	0.0250	0.0226	13.2
270 days	Call	1.4793	0.0586	0.0609	0.0598	13.0
	Put	1.3710	0.0234	0.0273	0.0254	13.2
	Put	1.3455	0.0173	0.0206	0.0190	13.2

The implied volatility corresponding to the mid-market price is displayed in the last column (IVOL).

Prices computed using the pricing system are compared with the market ones and the estimated volatility surfaces, computed using neural networks and Dupire's formula, are analyzed in order to verify that their shape satisfy the market behavior.

From Table 1, we can describe the data set of examples $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ in the following way:

- $\mathbf{x}_{(1)}$ "*time to maturity*",
- $\mathbf{x}_{(2)}$ "*strike price*",
- $\mathbf{y}_n$, "*implied volatility*".

In order to accomplish the implied volatility modeling task we used all the available data points, i.e. *25* observations.

Table 2. *α-rejectable neurons* for single and two hidden layers networks.

Network	ELR	ALR	$S(\theta)$
$\phi(\mid \mathbf{x} \mid, \mathbf{1})$	none	none	$1.77 \cdot 10^{-4}$
$\phi(\mid \mathbf{x} \mid, \mathbf{2})$	none	all	$4.90 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{3})$	h_1	all	$4.21 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{4})$	h_4	all	$3.84 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{5})$	h_4, h_5	all	$48.74 \cdot 10^{-6}$
$\phi(\mid \mathbf{x} \mid, \mathbf{1}, \mathbf{1})$	none	all	$2.26 \cdot 10^{-4}$
$\phi(\mid \mathbf{x} \mid, \mathbf{2}, \mathbf{1})$	none	all	$4.93 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{3}, \mathbf{1})$	$h_1^{(1)}, h_2^{(1)}$	all	$3.33 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{2}, \mathbf{2})$	$h_1^{(1)}, h_2^{(1)}$	all	$1.82 \cdot 10^{-5}$
$\phi(\mid \mathbf{x} \mid, \mathbf{3}, \mathbf{2})$	$h_1^{(1)}, h_2^{(1)}, h_1^{(2)}$	all	$1.09 \cdot 10^{-5}$

Then, accordingly to what presented for the network's structure selection, we set the maximum number of hidden neurons for the single layer FNN to five, i.e. $H = 5$ while for two layer FNN we set the maximum number of hidden neurons for the first layer to three, i.e. $H_{\max}^{(1)} = 3$.

Then, applying Steps 1 to 6 from the above described procedure for the network's structure selection, with confidence level of $\alpha = 0.99$, we obtain the set of the candidate networks $\Phi_{0.99}^{\mathbf{D}}$ (Table 2) for both cases when approximated likelihood regions (ALR) and exact likelihood regions (ELR) are used.

From Table 2 we can read off the set of the candidate networks. In particular we can notice that whether the approximated likelihood regions are used this set is $\Phi_{0.99}^{\mathbf{D},a} = \{\phi(|\mathbf{x}|, 1)\}$, while in the case when the exact likelihood regions are used it is $\Phi_{0.99}^{\mathbf{D},e} = \{\phi(|\mathbf{x}|, 1), \phi(|\mathbf{x}|, 2), \phi(|\mathbf{x}|, 1, 1), \phi(|\mathbf{x}|, 2, 1)\}$.

Let us cope with this discrepancy. To this end, accordingly to Step 7, the first network which should be inspected is $\phi(|\mathbf{x}|, 2)$. For this network, in the case when the approximated likelihood regions are used, all the neurons are rejected. To explain this discrepancy, let us analyze the *profile t plots* for the parameters associated with the neurons h_1 and h_2.

The exact likelihood interval, for each parameter θ_p, can be computed by means of its *profile t plot*. Indeed, the *profile t plot* gives a set of values for $\delta(\theta_p)$, $\tilde{\boldsymbol{\theta}}_{-p}$ and $\tau(\theta_p)$ which allows to plot $\tau(\theta_p)$ versus $\delta(\theta_p)$. Then, we can compute the exact likelihood region, for each parameter θ_p, according to the following formula:

$$\theta_p = \hat{\theta}_p + (\delta(\theta_p) \cdot se(\hat{\theta}_p)) \tag{4.2}$$

- The neuron h_1 is associated with the weights $w_{11}^{(1)}, w_{21}^{(1)}$ and $w_1^{(2)}$ corresponding respectively to θ_1, θ_3 and θ_7 in the $\boldsymbol{\theta}$ parameter space. The corresponding *profile t plots* are reported in Fig. 4 in which we have included a straight line with slope 1 (solid), corresponding to the linear case, axes in scales of $\delta(\theta_p)$ and nominal confidence level, which make easy to read off likelihood intervals. Dashed lines show the 100 (0.99)% marginal likelihood interval.

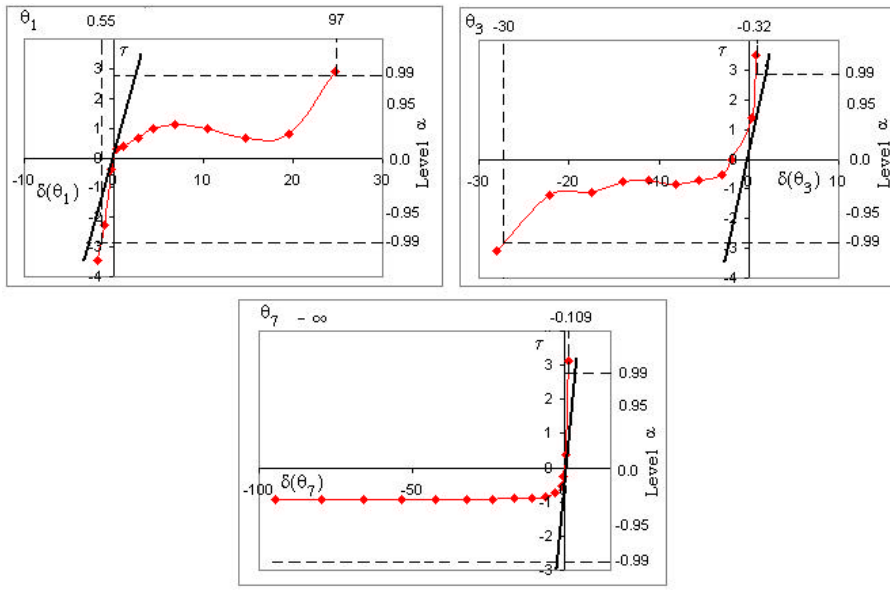


Fig. 4. *Profile t plots* for θ_1, θ_3 and θ_7 (neuron h_1).

Table 3. Approximated and exact likelihood regions for the neuron h_1.

ALR	ELR
$JLI(\{\theta_1,\theta_3\},0.99) = \begin{bmatrix} -1.39, & 5 \\ -5.12, & 1.9 \end{bmatrix}$	$JLI(\{\theta_1,\theta_3\},0.99) = \begin{bmatrix} 0.55, & 97 \\ -30, & -0.32 \end{bmatrix}$
$MLI(\{\theta_7\},0.99) = [-4.95, 1.95]$	$MLI(\{\theta_7\},0.99) = [-\infty, -0.109]$

The analysis of the plots reported in Fig. 4 shows severe nonlinearity associated with these parameters. Indeed, the profiles are badly curved. In particular, the *profile t plot* for parameter θ_7 tends to asymptote and the likelihood interval does not close on the left side.

The 100 (0.99)% exact likelihood intervals are very different from their linear approximations (Table 3).

Thus, applying Definition 3.4 to the neuron h_1, in the case when the ELR are used h_1 is *α-non rejectable*. While in the case when the ALR are used it is *α-rejectable*.

- The neuron h_2 is associated with the weights $w_{12}^{(1)}, w_{22}^{(1)}$ and $w_2^{(2)}$ corresponding respectively to θ_2, θ_4 and θ_8 in the $\boldsymbol{\theta}$ parameter space. The *profile t plots* are reported in Fig. 5. As for the neuron h_1, the linear approximation is inadequate to the extent of properly summarizing the estimation situation. Indeed, the profiles are badly curved. In particular, the *profile t plots* for the parameters θ_4 and θ_8 tend to asymptote and the likelihood intervals do not close.

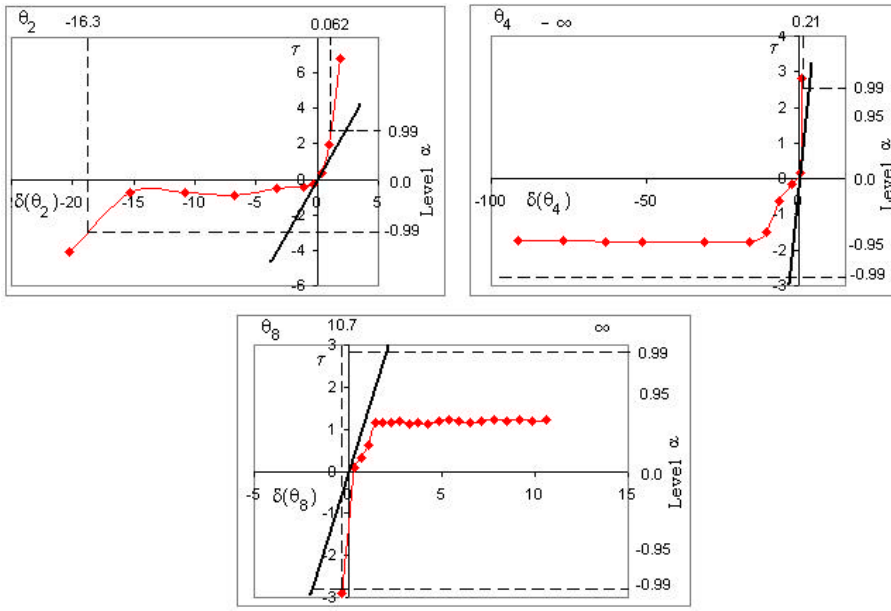


Fig. 5. *Profile t plots* for θ_2, θ_4 and θ_8 (neuron h_2).

Table 4. Approximated and exact likelihood regions for the neuron h_2.

ALR	ELR
$JLI(\{\theta_2,\theta_4\},0.99) = \begin{bmatrix} -3.68, & 2.28 \\ -8.35, & 5.45 \end{bmatrix}$	$JJLI(\{\theta_2,\theta_4\},0.99) = \begin{bmatrix} -16.3, & 0.062 \\ -\infty, & 0.21 \end{bmatrix}$
$MLI(\{\theta_8\},0.99) = [-1.36, 1.38]$	$MLI(\{\theta_8\},0.99) = [10.7, +\infty]$

Therefore also for the neuron h_2 some evidence exists that its parameters are extremely nonlinear. Indeed, the exact likelihood intervals for the parameters associated with the neuron h_2 are extremely different from their linear approximations as can be noticed from Table 4.

Thus, applying Definition 3.4 to the neuron h_2, in the case when the ELR are used h_2 is α-*non rejectable*, while in the case when the ALR are used it is α-*rejectable*.

In such a situation the correct approach is to use the exact likelihood regions which drive to draw the right conclusions about the significance of the model's parameters.

Once we realize that the set of models which should be analyzed is obtained through the exact likelihood regions, we are left with the problem of checking the assumptions (3.4).

Thus we have to analyze the residual distribution resulting from the model $\phi(|\mathbf{x}|, 2)$.

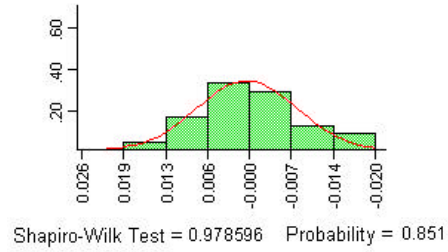


Fig. 6. Histogram plot of the $\phi(|\mathbf{x}|, 2)$ residuals.

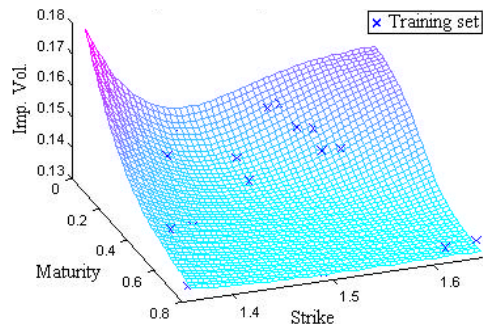


Fig. 7. Implied volatility surface.

In order to accomplish this task we can use graphical procedures and/or tests of normality as Shapiro–Wilk or Kolmogorov–Smirnoff [25].

In particular, the Shapiro–Wilk test statistic has a value of 0.978596 resulting in a probability of 0.8516.

Furthermore, from the histogram plot (Fig. 6), we conclude that there is no statistical evidence that the residual distribution differs significantly from the gaussian one.

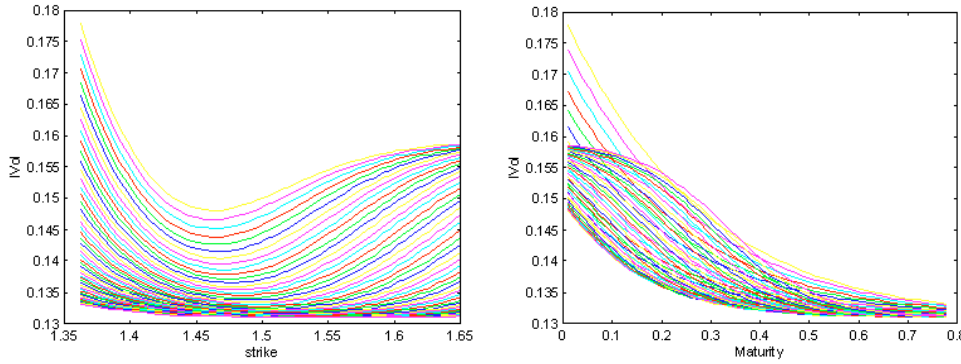
This suggests that the model $\phi(|\mathbf{x}|, 2)$ can be used for predicting new cases from the DGP.

In Fig. 7 the implied volatility surface obtained by means of $\phi(|\mathbf{x}|, 2)$ is reported.

It is interesting to analyze some sections of the obtained implied volatility surface (Fig. 8).

The first section (Fig. 8(a)) is concerned with the so-called strike structure. In such a case, for any fixed expiration, the implied volatility varies with the strike level. As can be seen the implied volatility increases for decreasing strikes, i.e. the out-of-the-money puts trade at higher implied volatility than the out-of-the-money calls.

The second section (Fig. 8(b)) represents the term structure. For any fixed strike level, the implied volatility varies with the time to expiration. We can observe how the short-term implied volatility exceeds the long-term implied volatility.



(a) (b)

Fig. 8. Strike and term structures. (a) Strike structure and (b) Term structure.

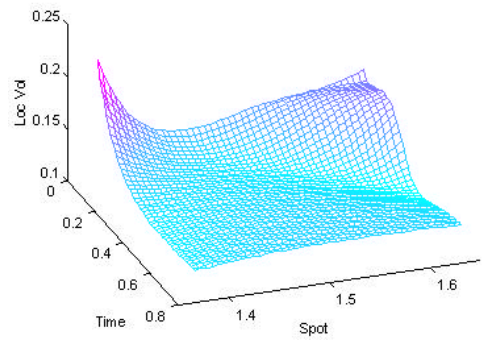


Fig. 9. Local volatility surface.

According to what was presented concerning the computational scheme for option pricing, and starting from the implied volatility surface, the local volatility surface has been computed using the Dupire's formula.

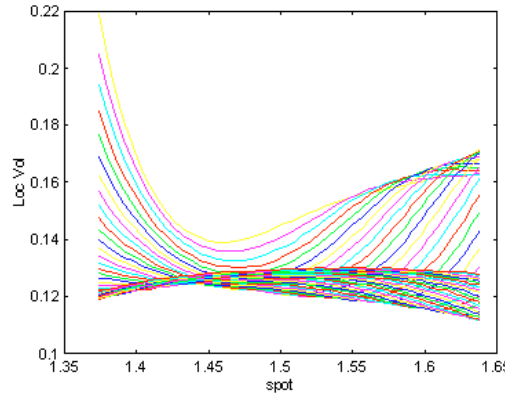
An important transformation is performed through this step: The implied volatility function $\sigma_{\text{imp}}(K, T)$, function of the strike price K and of the time to maturity T, is used to compute the corresponding local volatility function $\sigma_{\text{loc}}(S, t)$, function of the future underlying level S and of the time t.

The local volatility surface, obtained for the USD/DEM market, is reported in Fig. 9.

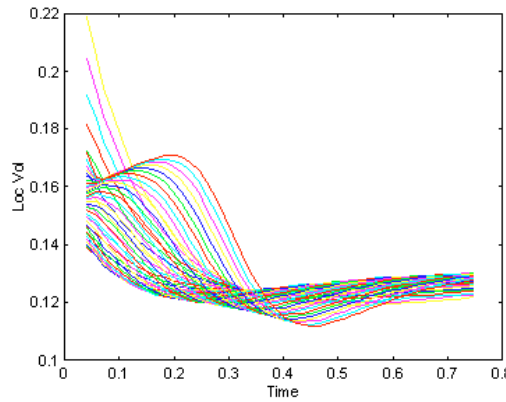
Let us consider two sections of the local volatility surface (Fig. 10).

The first section (Fig. 10(a)), which is respect to the spot price, describes the market's consensus for the future local volatility with respect to the spot price changes.

The second section (Fig. 10(b)) shows the shape of the volatility function for fixed values of the spot price.



(a)



(b)

Fig. 10. Spot and time structures. (a) Sport structure and (b) Time structure.

Finally, in order to validate the proposed approach, we have to first compute the option prices for the original set of strikes and maturities (see Table 1) and second to perform a comparison of such values with the real ones. In order to cope with the first task, i.e. the computation of the option prices for the original set of strikes and maturities, we utilize a trinomial tree capable to model the diffusion process of the form

$$\frac{dS}{S} = \sigma_{\text{loc}}(S,t) + \mu(S,t)$$

for which the trend $\mu(s,t)$ and the volatility $\sigma_{\text{loc}}(s,t)$ explicitly depend from the underlying level S and time t. The results of the second task, i.e. the comparison between the computed and the original price values, are reported in Table 5.

From Table 5 it is immediate to verify how the pricing system correctly prices the considered options. Indeed, the computed values are very close to the market prices, and always within the bid-ask spread, reflecting the market uncertainty.

Table 5. USD/DEM option market prices comparison.

Maturity	Type	Strike	Bid	Offer	Mid	Computed
	Call	1.5421	0.0064	0.0076	0.0070	0.0071
	Call	1.5310	0.0086	0.0100	0.0093	0.0091
30 days	Call	1.4872	0.0230	0.0238	0.0234	0.0238
	Put	1.4479	0.0085	0.0098	0.0092	0.0093
	Put	1.4371	0.0063	0.0074	0.0069	0.0070
	Call	1.5621	0.0086	0.0102	0.0094	0.0098
	Call	1.5469	0.0116	0.0135	0.0126	0.0125
60 days	Call	1.4866	0.0313	0.0325	0.0319	0.0321
	Put	1.4312	0.0118	0.0137	0.0128	0.0129
	Put	1.4178	0.0087	0.0113	0.0100	0.0099
	Call	1.5764	0.0101	0.0122	0.0112	0.0115
	Call	1.5580	0.0137	0.0160	0.0149	0.0148
90 days	Call	1.4856	0.0370	0.0385	0.0378	0.0382
	Put	1.4197	0.0141	0.0164	0.0153	0.0156
	Put	1.4038	0.0104	0.0124	0.0114	0.0119
	Call	1.6025	0.0129	0.0152	0.0141	0.0150
	Call	1.5779	0.0175	0.0207	0.0191	0.0198
180 days	Call	1.4823	0.0494	0.0515	0.0505	0.0514
	Put	1.3902	0.0200	0.0232	0.0216	0.0206
	Put	1.3682	0.0147	0.0176	0.0162	0.0152
	Call	1.6297	0.0156	0.0190	0.0173	0.0170
	Call	1.5988	0.0211	0.0250	0.0226	0.0229
270 days	Call	1.4793	0.0586	0.0609	0.0598	0.0603
	Put	1.3710	0.0234	0.0273	0.0254	0.0256
	Put	1.3455	0.0173	0.0206	0.0190	0.0188

5. Conclusions

This paper shows two main opportunities.

The first one is the particular appealing of FNNs modeling in the case of option pricing and in particular for modeling the implied volatility surface in the case when appropriate procedures are adopted. More precisely the paper puts emphasis on realizing that the FNNs training is a nonlinear least squares problem and then the main tools from nonlinear regression analysis should be used. This reccomendation is particularly important expecially in the case when few data points are available which is exactly the case of the implied volatility modeling we considered. Furthermore the *profile t function* reveals how nonlinear the estimation situation is and then could be used for evaluating the importance for any explanatory variable.

The second opportunity is represented by the fact that it is possible to pass from a set of liquid option prices to a pricing system by means of which we can value other derivatives whose prices are not readily available from the market, i.e. illiquid European options, American options and Exotic options, secure in the knowledge that the system is valuing all the instruments consistently with the market. The proposed pricing system can be used for pricing Barrier options, where the probability of striking the barrier is sensitive to the shape of the smile, for creating static hedge portfolios for Exotic options or for generating Monte Carlo distributions for valuing path-dependent options.

References

[1] H. White, *Learning in artificial neural networks: a statistical perspective*, Neural Computation **1**(4) (1989) 425–464.

[2] H. White, *Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings*, Neural Networks **3**(5) (1990) 535–549.

[3] D. J. C. MacKay, *Bayesian interpolation*, Neural Computation **4**(3) (1992) 415–447.

[4] R. Reed, *Pruning algorithms - a survey*, IEEE Transactions on Neural Networks **4**(5) (1993) 740–747.

[5] G. Thimm and E. Fiesler, *Evaluating pruning methods*, Proc. Int. Sym. on Artificial Neural Networks, 1995.

[6] B. Hassibi and D. G. Stork, *Second derivatives for network pruning: optimal brain surgeon*, in *NIPS5* (1993) 164–171.

[7] E. Mayoraz and F. Aviolat, *Constructive training methods for feedforward neural networks with binary weights*, Technical report, Rutcor, New Brunswick, NJ (1996).

[8] M. Stone, *Cross-validatory choice and assessment of statistical predictions*, J. Royal Stat. Soc. **B36**(1) (1974) 111–147.

[9] G. Wahba and S. Wold, *A completely automatic French curve: fitting spline functions by cross-validation*, Communications in Statistics **A4**(1) (1975) 1–17.

[10] T. Poggio and F. Girosi, *Networks for approximation and learning*, in *Proc. IEEE* **78** (1990) 1481–1497.

[11] J. Rissanen, *Modeling by shortest data description*, Automatica **14** (1996) 465–471.

[12] Y. S. Abu-Mostafa, *The Vapnik–Chervonenkis dimension: information versus complexity in learning*, Neural Computation **1**(3) (1989) 312–317.

[13] R. M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, Springer Verlag, Berlin (1996).

[14] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon-Press, Oxford (1995).

[15] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge Univ. Press, Cambridge (1996).

[16] A. A. Gaivoronski, *Convergence properties of backpropagation for neural nets via theory of stochastic gradient methods, Part I*, in *Optimization and Software* (1994).

[17] J. McClelland, D. Rummelhart and the PDP Research Group, *Parallel Distributed Processing, Vol. 1 and 2*, The MIT Press, Cambrige (1986).

[18] A. K. Rigler, W. T. Zink, D. L. Alkon, T. P. Volg and J. K. Mangis, *Accelerating the convergence of the backpropagation method*, Biological Cybernetic **59** (1988) 257–263.

[19] J. J. McKeown, *Specialised versus general-purpose algorithms for minimizing functions that are sums of squared terms*, Mathematical Programming **9** (1975) 57–68.

[20] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Massachussets (1995).
[21] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and its Applications*, Wiley, New York (1988).
[22] D. G. Watts, *Parameter estimates from nonlinear models*, Methods in Enzymology **240** (1994) 23–36.
[23] D. A. Ratkowsky, *Nonlinear Regression Modeling: A Unified Practical Approach*, Marcel Dekker, Inc., New York (1983).
[24] D. A. Ratkowsky, *Handbook of Nonlinear Regression Models*, Marcel Dekker, Inc., New York (1989).
[25] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, John Wiley and Sons, New York (1987).
[26] A. Carelli, *Entropia e reti neurali per il pricing di opzioni*, PhD thesis, Univ. of Bergamo (1998).
[27] M. Avellaneda, A. Carelli and F. Stella, *A Bayesian Approach for Constructing implied volatility surfaces through neural networks*, to appear in J. Computational Finance (2000).
[28] F. Black, *The pricing of options and corporate liabilities*, J. Political Economy **81** (1973) 637–654.
[29] B. Dupire, *Model art*, RISK **6**(9) (September, 1993) 118–124.
[30] B. Dupire, *Pricing with a smile*, RISK **7**(1) (January, 1994) 18–20.
[31] E. Derman and I. Kani, *The volatility smile and its implied tree*, Technical report, Goldman Sachs (1994).
[32] E. Derman, I. Kani and J. Z. Zou, *The local volatility surface-unlocking the information in index option prices*, Technical report, Goldman Sachs (1995).
[33] M. Avellaneda and A. Paras, *Managing the volatility risk of portfolios of derivative securities: the Lagrangian uncertain volatility model*, Appl. Math. Finance **3** (1996) 21–52.