MATHEMATICAL BIOSCIENCES
AND ENGINEERING
Volume **11**, Number **2**, April **2014**

doi:10.3934/mbe.2014.11.331

pp. **331–342**

DESIGNING NEURAL NETWORKS FOR MODELING BIOLOGICAL DATA: A STATISTICAL PERSPECTIVE

Michele La Rocca and Cira Perna

Department of Economics and Statistics – University of Salerno
Via Giovanni Paolo II, 132. 84084 Fisciano (SA), Italy

Abstract. In this paper, we propose a strategy for the selection of the hidden layer size in feedforward neural network models. The procedure herein presented is based on comparison of different models in terms of their out of sample predictive ability, for a specified loss function. To overcome the problem of data snooping, we extend the scheme based on the use of the reality check with modifications apt to compare nested models. Some applications of the proposed procedure to simulated and real data sets show that it allows to select parsimonious neural network models with the highest predictive accuracy.

1. **Introduction.** It is widely accepted that complex structures, in time and in space, exist in biological data and that non-linear models, both parametric and non-parametric, can effectively be used to reveal these patterns. In this context, artificial neural networks are one of the most popular artificial learning tools due to their ability to accurately represent the complex, non linear behaviour of relatively poorly understood processes without any a priori knowledge of input and output relationships.

The growing interest in neural networks, with respect to other non parametric techniques, is due to their versatility which comes from the high capability of providing, under quite general conditions, an arbitrarily accurate approximation to an unknown target function of interest. Barron [2] obtained a deterministic approximation rate for a class of single hidden layer feedforward neural networks with r hidden units and sigmoid activation functions when the target function satisfies certain smoothness conditions. Hornik et al. [16] extended Barron's result to a class of neural networks with possibly non-sigmoid activation approximating the target function and its derivatives simultaneously. More recently, Makovoz [23] and Chen and White [5] obtained an improved degree of approximation of Barron's neural networks with sigmoid activation function. Moreover, neural networks do not suffer for the so-called 'curse of dimensionality'. Theoretically, they are expected to perform better than other approximation methods since the approximation form is not so sensitive to the increasing data space dimension, at least within the confines of particular classes of functions.

The reliability of using neural networks in practice has been affirmed in many different applications ranging from pattern recognition [3], in chromatographic spectra [15, 4], and expression profiles [36, 18], to functional analyses of genomic and proteomic sequences [6] to QSAR models [10, 1].

2010 *Mathematics Subject Classification.* Primary: 62G08, 62H15; Secondary: 62F40, 92B20.
Key words and phrases. Neural networks, predictive ability, multiple testing, bootstrap.

Many studies [24, 40, 32] agree that one of the main difficulties in applying neural networks in biotechnology is the choice of an adequate model. The problem can be faced referring to the many proposals based on trial and error procedures, adequately combined with pruning of the network graph, based on weight selection or classical information criteria. Alternatively, the neural network model building strategy could be faced in a statistical perspective, relating it to the classical model selection approach. In the case of the single hidden feedforward neural network class, model selection basically involves the choice of the number and type of input neurons and the number of neurons in the hidden layer. In our opinion, a model building process should highlight the different role of these two types of neurons. The input neurons are related to the explanatory variables and, as a consequence, are useful for the identification and interpretation of the model. Therefore, although many techniques have been proposed in the literature, their selection should be addressed focusing on statistical test procedures for variable selection in regression models [19, 20] in order to give information explicitly on the 'relevance' of the variable to the model. On the contrary, the hidden layer size takes into account the trade-off between estimation bias and variability and so it is related to the complexity of the model. Of course, the selection of this parameter plays an important role since under-parametrized (over-parametrized) models can lead to heavy consequences on underfitting (overfitting) and, consequently, poor modeling performance or reduced ex-post forecast accuracy. Generally, it is chosen according to one of the information criteria available in the statistical literature even if many statistical studies [31, 26] agree on the failure of these measures in choosing the best forecasting model.

In this paper, we propose a strategy in which the hidden layer size is selected by comparing models in terms of their out of sample predictive ability, for a specified loss function. In this context, since a given set of data is used more than once for inference and model selection, there is the possibility that any satisfactory results may simply be due to chance rather than to the model itself (a problem known as 'data snooping'). To overcome this problem, we extend the procedure based on the use of the reality check proposed in White [39] with the modification for nested models as proposed in Clark and McCracken [7, 8].

The paper is organized as follows. In the next section, we describe the structure of the data generating process and the neural network model. In section 3 we discuss the proposed test procedure for model selection. Numerical examples on simulated data and a moderate Monte Carlo experiment are reported in section 4 while in section 5 two applications to biological data are discussed. Some final remarks close the paper.

2. Neural network models.

Let the observed data be the realization of a sequence $\left\{\mathbf{Z}_i = \left(Y_i, \mathbf{X}_i^T\right)^T\right\}$ of independent identically distributed (iid) random vectors of order $(d+1)$, with $i \in \mathbb{N}$.

The random variables Y_i represent targets (in the neural network jargon) and it is usually of interest the probabilistic relationship with the variables $\mathbf{X}_i$, described by the conditional distribution of the random variable $Y_i \,|\mathbf{X}_i$. Certain aspects of this probability law are relevant in interpreting what is the modeling role of artificial neural network models. If $\mathbb{E}\,(Y_i) < \infty$, then $\mathbb{E}\,(Y_i \,|\mathbf{X}_i\,) = g\,(\mathbf{X}_i)$ and we can write

$$Y_i = g\,(\mathbf{X}_i) + \varepsilon_i \tag{1}$$

where $\varepsilon_i \equiv Y_i - g(\mathbf{X}_i)$ and $g : \mathbb{R}^d \to \mathbb{R}$ is a measurable function. Clearly, by construction the error term ε_i is such that $\mathbb{E}(\varepsilon_i | \mathbf{X}_i) = 0$.

The function g embodies the sistematic part of the stochastic relation between Y_i and $\mathbf{X}_i$. It can be approximated by using the output of a single hidden layer feedforward artificial neural network of the form

$$f(\mathbf{x}, \mathbf{w}) = w_{00} + \sum_{j=1}^{r} w_{0j} \psi\left(\tilde{\mathbf{x}}^T \mathbf{w}_{1j}\right) \tag{2}$$

where $\mathbf{w} \equiv \left(w_{00}, w_{01}, \ldots w_{0r}, \mathbf{w}_{11}^T, \ldots, \mathbf{w}_{1r}^T\right)^T$ is a $r(d+2)+1$ vector of network weights, $\mathbf{w} \in \mathbf{W}$ with $\mathbf{W}$ compact subset of $\mathbb{R}^{r(d+2)+1}$, and $\tilde{\mathbf{x}} \equiv \left(1, \mathbf{x}^T\right)^T$ is the input vector augmented by a bias component 1. The network (2) has d input neurons, r neurons in the hidden layer and identity function for the output layer. The (fixed) hidden unit activation function ψ is chosen in such a way that $f(\mathbf{x}, \cdot) : \mathbf{W} \to \mathbf{R}$ is continuous for each x in the support of the marginal distribution of $\mathbf{X}_i$ and $f(\cdot, \mathbf{w}) : \mathbf{R}^d \to \mathbf{R}$ is measurable for each $\mathbf{w}$ in $\mathbf{W}$.

The main difference with a parametric model is that instead of postulating a specific non-linear function, a neural network model is constructed by combining many "simple" non-linear functions via a multi-layer structure. In a feedforward network, the explanatory variables simultaneously activate r hidden units in an intermediate layer through some function ψ, and the resulting hidden-unit activations $\psi\left(\tilde{\mathbf{x}}^T \mathbf{w}_{1j}\right), j = 1, \ldots, r$, then activate output units to produce the network output.

Given a training set of N observations, $\mathcal{D} = \{(Y_i, \mathbf{X}_i), i = 1, 2, \ldots, N\}$ estimation of the network weights (learning) is obtained by solving the optimization problem

$$\min_{\mathbf{w} \in \mathbf{W}} \frac{1}{N} \sum_{i=1}^{N} q(Y_i, f(\mathbf{X}_i, \mathbf{w})) \tag{3}$$

where $q(\cdot)$ is a proper chosen loss function. Under general regularity conditions (White, [37]), denoting $\pi(\mathbf{z})$ the distribution of $\mathbf{Z}_i$, a weight vector $\hat{\mathbf{w}}_n$ solving equation (3) exists and converges almost surely to $\mathbf{w}^*$, which solves

$$\min_{\mathbf{w} \in \mathbf{W}} \int q(y, f(\mathbf{x}, \mathbf{w})) d\pi(\mathbf{z}) \tag{4}$$

provided that the integral exists and the optimization problem has a unique solution vector interior to $\mathbf{W}$. This is not necessarily true for neural networks without considering appropriate restrictions, since the parametrization of the network function is not unique. However, Ossen and Rügen [25] provide sufficient conditions to ensure uniqueness of $\mathbf{w}^*$ in a suitable parameter space $\mathbf{W}$ for specific network configurations.

Moreover, from an asymptotic point of view, the possible presence of multiple minima has no essential effect for solutions to equation (4) (see [37]), while, from a computational point of view, several global optimization strategies (simulation annealing, genetic algorithms, etc.) have been successfully employed to avoid to be trapped in local minima. Finally, when the focus is on prediction, as in this paper, it can be shown that the unidentifiability can be overcome and the problem disappears [17].

The estimated neural network model should capture the real functional relationship existing between the inputs and the output, so it should well approximate the observed data and, at the same time, it should perform reasonably well on new data (prediction). Clearly, a good neural network model is characterised by its ability to

approximate and to generalise in a proper way. A model structure that is chosen to be too complex in relation to the real functional relationship, captures the noise contained in the data (overfitting). Such a model will perform well in approximating the data used for the estimation of its parameters but very poorly on new data.

The trade-off between approximation accuracy and generalization ability is governed by the hidden layer size which, in neural network framework, plays the role of a smoothing parameter.

This issue is usually addressed by splitting the available data set into two subsets: (i) the training set and the test data set. The first is used for estimating the weights of a certain model structure with a specified number of hidden units; then the test data set is fed into the neural network model which runs in the prediction mode. The discrepancy between the computed and the observed data of the second subset, expressed as mean square error, is a measure of the generalisation property of the network. The relationship between training and test data is usually chosen to be 70% to 30%. To apply this method one usually starts with a small model structure which is stepwise increased by adding hidden units. For every structure the approximation and generalisation errors are computed as mean square error on the basis of the training and the test data respectively. Obviously, the approximation error is expected to decrease continuously with increasing complexity of the model. The generalisation will also improve with increasing number of hidden nodes,but beyond a certain complexity the model will have a poor generalisation performance, due to overfitting problems, even if the number of hidden nodes is still increasing. The number of hidden units corresponding to the minimum of the generalisation error determines the optimal model structure and represents the solution to the model selection problem.

In any case, all these model selection procedures are not entirely satisfactory. Since model selection criteria depend on sample information, their actual values are subject to statistical variations. As a consequence, a model with higher model selection criterion value may not *significantly* outperform its competitors. In recent years there is a growing literature addressing the problem of comparing different models and theories through the use of predictive performance and predictive accuracy tests ([9] and the references therein). In this literature, it is quite common to compare multiple models, which are possibly misspecified (they are all approximations of some unknown true model), in terms of their out of sample predictive ability, for a specified loss function. In such context data snooping, which occurs when a given set of data is used more than once for inference or model selection, can be a serious problem. When such data reuse occurs, there is always the possibility that any satisfactory results obtained may simply be due to chance rather than any merit inherent to the model yielding to the result. In other words, by looking long enough and hard enough at a given data set it will often reveal one or more forecasting models that look good but are in fact useless.

The data snooping can be particularly serious especially when there is no theory supporting the modeling strategy, as it happens when using neural network models which are basically atheoretical. Unfortunately, as far as we know, there are no results addressing the problem just described in a neural network framework.

3. **Testing superior predictive ability for neural network modeling design.**

Let $(Y_\tau, \mathbf{X}_\tau)$ denote a future observation that satisfies

$$Y_\tau = g(\mathbf{X}_\tau) + \varepsilon_\tau \tag{5}$$

At this stage, assume that the set of explicative variable has been selected by an appropriate variable selection technique (see [19, 21] inter alia).

Moreover, assume that $k+1$ alternative forecasting neural network models are available, namely $f_j(\mathbf{x}, \mathbf{w}), j = 0, 1, \ldots, k$, where j denotes the hidden layer size and k is fixed to maximum level of desired complexity. Obviously, $f_0(\mathbf{x}, \mathbf{w})$ is a neural network with skip layer and $r = 0$ neurons in the hidden layer (that is the linear model). In our framework it is assumed to be the benchmark model.

Let the generic forecast error be $u_{j,\tau} = Y_\tau - f_j(\mathbf{X}_\tau, \mathbf{w}^*), j = 0, 1, \ldots, k$ where $\mathbf{w}^*$ is defined as in the previous section. Let h be a loss function chosen to properly weight the forecasting error [12] and define

$$\theta_j = \mathbb{E}\left(h\left(u_{0,\tau}\right) - h\left(u_{j,\tau}\right)\right), j = 1, 2, \ldots, k. \tag{6}$$

Clearly, if model j beats the benchmark (i.e. it shows better expected predictive performances) we have $\theta_j > 0$, otherwise $\theta_j \leq 0$ and our goal is to identify as many models for which $\theta_j > 0$. In other words, for a given model j, consider

$$H_j : \theta_j \leq 0 \quad vs \quad H_j' : \theta_j > 0, \qquad j = 1, 2, \ldots, k \tag{7}$$

and, in a multiple testing framework, take a decision concerning each individual testing problem by either rejecting H_j or not. In this framework, to avoid declaring true null hypotheses to be false, the familywise error rate, defined as the probability of rejecting at least one of the true null hypotheses, should be taken under control.

This can be done by using the well known Bonferroni method or stepwise procedures such as Holm's approach, which are more powerful. Unfortunately, all these procedures are conservative since they do not take into account the dependence structure of the individual p-values [27]. To avoid these issues, it is possible to use the reality check as in White [39] and the modification for nested models as proposed in Clark and McCracken [7, 8].

This latter approach can be easily extended to our neural network framework. Let $\mathcal{S} = \{(Y_i, \mathbf{X}_i), i \in S\}$ and $\mathcal{P} = \{(Y_i, \mathbf{X}_i), i \in P\}$ denote, respectively, the estimation data set and the test data set, where $\mathcal{P}$ is the complement set of $\mathcal{S}$ with respect to $\mathcal{D}$, with $|\mathcal{P}| = N - |\mathcal{S}|$. Let the estimated forecast error be $\hat{u}_{j,\tau} = Y_\tau - f_j(\mathbf{X}_\tau, \hat{\mathbf{w}}), j = 0, 1, \ldots, k$ and let $\mathrm{MPE}_j = \sum_{\tau \in P} h\left(\hat{u}_{j,\tau}\right)$, where P is the cardinality of the set $\mathcal{P}$.

The test procedure can be based on the F-type statistic defined as

$$Fp_j = P\frac{\mathrm{MPE}_0 - \mathrm{MPE}_j}{\mathrm{MPE}_j}, \qquad j = 1, 2, \ldots, k. \tag{8}$$

It has a clear interpretation: large values of Fp_j indicate evidence against the null H_j.

The procedure for testing the system of hypotheses (7) keeping under control the family wise error rate, runs as follows. Relabel the hypothesis from H_{r_1} to H_{r_k} in redescending order with respect to the value of the test statistics Fp_j, that is $Fp_{r_1} \geq Fp_{r_2} \geq \ldots \geq Fp_{r_k}$. The procedure focuses on testing the joint null hypothesis that all hypotheses H_j are true, that is no competing model is able to beat the benchmark model. This hypothesis is rejected if Fp_{r1} is large, otherwise all hypotheses are accepted. In other words, the procedure constructs a rectangular joint confidence region for the vector $\left(Fp_{r_1}, \ldots, Fp_{r_k}\right)^T$, with nominal joint coverage probability $1-\alpha$. The confidence region is of the form $[Fp_{r_1} - c_{1-\alpha}, \infty) \times \cdots \times [Fp_{r_k} - c_{1-\alpha}, \infty)$ where the common value $c_{1-\alpha}$ is chosen to ensure the proper joint (asymptotic) coverage probability. If a particular individual confidence interval $\left[Fp_{r_j} - c_{1-\alpha}, \infty\right)$ does not contain zero, the corresponding null hypothesis H_{r_j} is rejected.

So, the testing procedure will select a set of models which delivers the greatest predictive ability, when compared to the benchmark model. All these models are somewhat equivalent and, for a parsimony principle, the one with smallest hidden layer size should be selected. If all the nulls are not rejected in the first step, there is no neural network model which is able to outperform the linear model (assumed as a benchmark) in terms of predictive ability. The quantile of order $c_{1-\alpha}$ is estimated by using the bootstrap [29].

The pseudo-code for the complete testing procedure is described in algorithm (1).

Algorithm 1 Testing algorithm for superior predictive ability.

1: Relabel the hypothesis from H_{r_1} to H_{r_k} in redescending order of the value of the test statistics Fp_j, that is $Fp_{r_1} \geq Fp_{r_2} \geq \ldots \geq Fp_{r_k}$.
2: Generate B bootstrap replicates $\mathbf{Z}^*_{N,1}, \mathbf{Z}^*_{N,2}, \ldots, \mathbf{Z}^*_{N,B}$ as iid samples from $\mathbf{Z}_N$
3: From each bootstrap data matrix $\mathbf{Z}^*_{N,b}$ with $b = 1, 2, \ldots, B$ compute the bootstrap counterparts of the individual test statistics $F^*p_{j,b}, j = 1, 2, \ldots, k$.
4: Let $\mathcal{K}$ be the set of indexes of models with better predictive performance
5: For $b = 1, 2, \ldots, B$ compute $\theta^{b,*}_N = \max_{1 \leq s \leq k} \left(Fp^*_{r_s,b} - Fp_{r_s} \right)$
6: Compute $\hat{c}_{1-\alpha}$ as the $1-\alpha$ quantile of the bootstrap values $\theta^{b,*}_N, b = 1, 2, \ldots, B$
7: **for** $s = 1$ to k **do**
8: **if** $0 \notin [Fp_{r_s} - \hat{c}_{1-\alpha}, \infty)$ **then**
9: reject H_{r_s} and include s in $\mathcal{K}$
10: **end if**
11: **end for**
12: Deliver the set $\mathcal{K}$ (if it is an empty set, no neural network model is able to beat the benchmark model)

4. **Some numerical results.** To illustrate the performance of the proposed model selection procedure, we use simulated data sets generated by models with known structure. The simulated data sets were generated by using different models often employed in the neural network literature as data generating processes.

The first model is the same used in De Veaux et al. [11] and it is defined as

$$Y = 1.5 \cos\left(\frac{2\pi}{\sqrt{3}} \sqrt{(X_1 - 0.5)^2 + (X_2 - 0.5)^2 + (X_3 - 0.5)^2} \right) + \varepsilon$$

where ε is gaussian with zero mean and variance equal to 0.1 and $\mathbf{X} = (X_1, X_2, X_3)^T$ is drawn randomly from the unit hypercube. The function is radially symmetric in these three variables. Clearly, the number of the neurons in the hidden layer is unknown and the model we try to identify is, by construction, misspecified.

The second model has been used by Friedman [13] and it is defined as

$$Y = \left(10 \sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon\right)/25$$

where $\mathbf{X} = (X_1, X_2, X_3, X_4, X_5)^T$ is a vector of multivariate uniform random variables and ε is gaussian with zero mean and variance equal to 1. The model includes both linear and nonlinear relationships.

The third model is the same one used by Tibshirani [33] and it is defined as

$$Y = 3\psi(2X_1 + 4X_2 + 3X_3 + 3X_4) + 3\psi(2X_1 + 4X_2 - 3X_3 - 3X_4) + \varepsilon$$

where ψ is the logistic activation function, $\mathbf{X} = (X_1, X_2, X_3, X_4)^T$ is a vector of multivariate gaussian random variables with zero mean, unit variance and pairwise correlation equal to 0.5 and ε is gaussian with zero mean and variance equal to 0.25. Clearly a neural network with logistic activation function, four input neurons and two hidden neurons is a correctly specified model and no misspecification is present.

The last model is the same model used by Turlach [35] and it is defined as

$$Y = (X_1 - 0.5)^2 + X_2 + X_3 + X_4 + X_5 + \varepsilon$$

where $\mathbf{X} = (X_1, X_2, X_3, X_4, X_5)^T$ is a vector of multivariate uniform random variables and ε is gaussian with zero mean and variance equal to 0.05. The model includes both linear and nonlinear relationships.

For the numerical examples, we have considered a quadratic loss function h and $N = 600$, $P = 180$, $B = 1000$ and $k = 8$. All neural network models have been estimated by using nonlinear least squares, including a weight decay in the objective function to control overfitting. Moreover, to avoid to be trapped in local minima, the estimation procedure has been initialized 25 times with random starting values, keeping the estimated network with the lowest residual sum of squares.

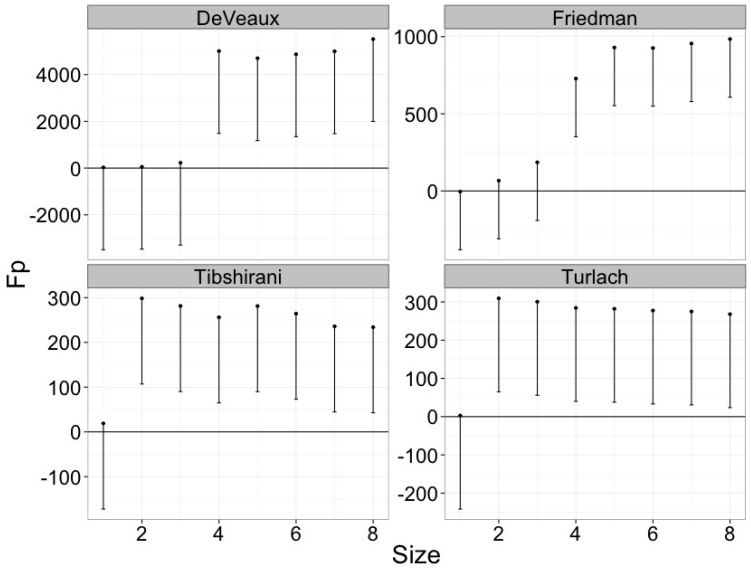

FIGURE 1. Joint confidence regions with nominal coverage probability $1 - \alpha = 0.95$

The results of the testing procedure for typical realizations are reported in figure (1). In the Tibshirani model case, the hidden layer size is known and equal to 2. The procedure correctly identifies the hidden layer size and indicates that it is not possible to improve accuracy by increasing the hidden layer size. All models with r ranging from 2 to 8 are basically equivalent with respect to the predictive accuracy. Similar remarks apply also to all other models. Note that for the DeVeaux and the Friedman data simply considering the statistical index would indicate $r = 8$ as the best choice, but this does not give any significant improvement with respect to $r = 4$.

A moderate Monte Carlo experiment has also been performed considering the same data generating processes as before. We have considered 240 Monte Carlo runs with three different sample sizes $N = 300, 400, 600$ using the last 30% observations for prediction. The results are reported in figure (2). In the Tibshirani case, the hidden layer size (which is known and equal to 2) the proportion of correct identification is very high for all the sample sizes, reaching 100% for $N = 600$. For the other data sets, the simulations confirm the results shown by the numerical examples and highlight the steep improvement as the sample size increases.

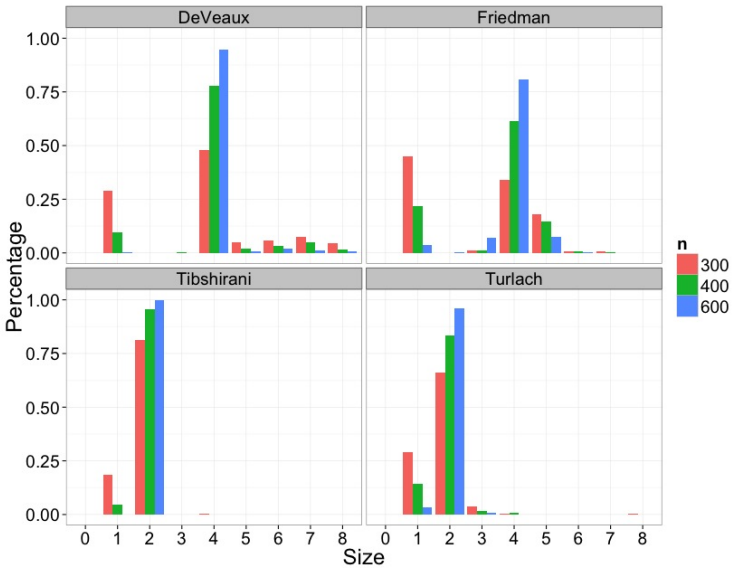

FIGURE 2. Proportion of hidden layer size identification by using the testing procedure for superior predictive ability

5. **Real data applications.** To validate the performance of the proposed procedure two applications to real data are discussed in this section. The aim is to verify if the proposed procedure is able to correctly identify an appropriate model structure for the data at hand.

As a first example, we use the Prostate Cancer data set which comes from a study by Stamey et al [30] and also used by Hastie et al [14]. The dependent variable is the level of prostate-specific antigen which depends on 8 clinical measures in men who were about to receive prostatectomy. The data set, already used in many biostatistical studies, has a well known regression structure and so it is suitable for testing new procedures. The data set has 97 observations and it is splitted in two subsets: 67 observations have been used for the modeling step while 30 observations have been used for the validation step. By using a linear model and a best subset variable selection rule, just two explanatory variables (out of eight) are identified as relevant: lweight (log prostate weight) and lcavol (log cancer volume). For sake of comparison, as identification tools for the number of hidden neurons, we also use the k-fold Cross-Validation (CV) selection rule (see [14] inter alia) and the Bayesian

Information Criterion (BIC) [28], proved to be consistent (almost surely) in the case of multi-layer perceptrons with one hidden layer in [38].

Clearly, the BIC identifies a neural network with skip layer and zero hidden neurons (i.e a linear model), for all the weight decay values considered (see figure 3, left panel). In the following, all the computations are based on a weight decay equal to zero, since it delivers the lowest BIC value. The CV also confirms the model selected by using the BIC and the same conclusions can be drawn by using the proposed test of superior predictive ability (see figure 4, left panel). To validate these results, a linear model and neural networks with hidden neurons ranging from 1 to 8 have been estimated and used to predict the observations in the validation set. The distributions of the absolute prediction errors are reported in figure 4, right panel. The plot shows that the neural networks considered are not able to provide better predictions with respect to the linear model (as predicted by the CV, the BIC and the novel test). Even a neural network with 6 hidden neurons (which shows the lowest median absolute prediction error) does not appear to provide prediction errors statistically different from those provided by the linear model. These results are confirmed by a formal statistical comparison between the two distributions: the Brunner Munzel test and the Wilcoxon rank sum test give p-values equal to 0.497 and 0.495, respectively.

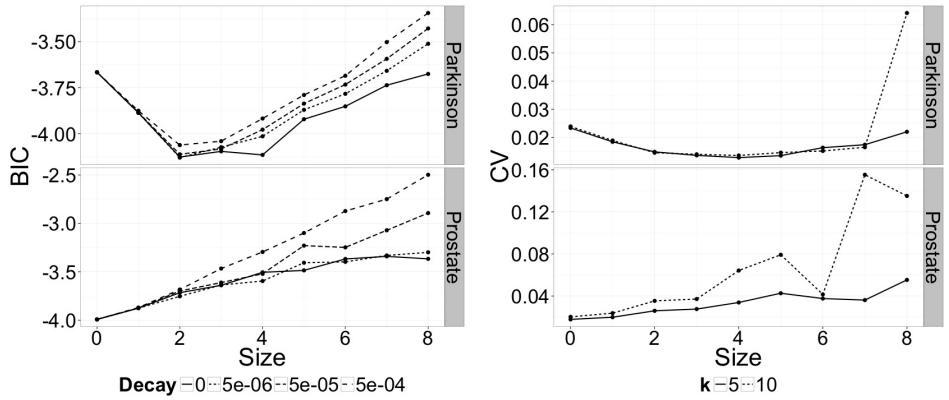

FIGURE 3. Bayesian Information Criterion values for different hidden layer sizes and different weight decay values (left panel). k-fold cross validation values for $k = 5$ and $k = 10$ using a weight decay equal to zero (right panel).

The second data set used as an example has been downloaded from the UCI Machine Learning Repository and is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring device for remote symptom progression monitoring (for details see [34]). The data set has 5,875 observations on age, gender and on 16 biomedical voice measures. The statistical model is used to predict the total UPDRS score. For computational reasons, just the subset of the first 887 observations (corresponding to the first 5 patients) has been considered. Again, the data set is splitted in two subsets: 731 observations (the first 4 individuals) have been used for the modeling step while 156 observations (corresponding to the 5th patient)

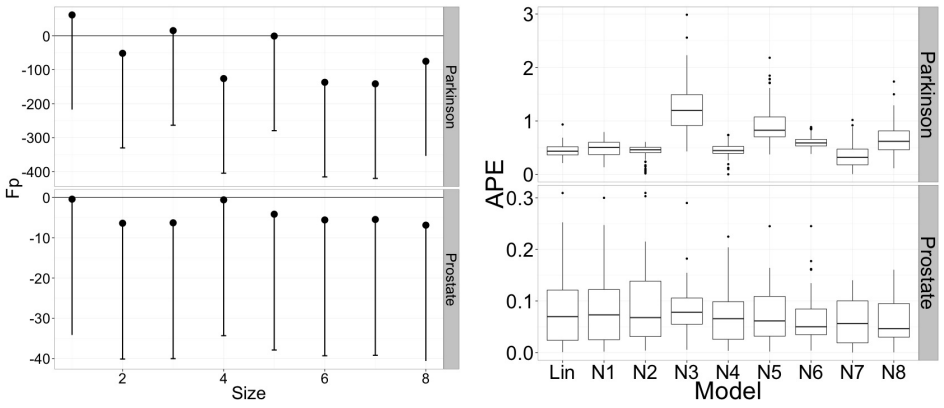

FIGURE 4. Joint confidence regions with nominal coverage probability $1 - \alpha = 0.95$ (left panel). Absolute prediction error distributions computed on the test set for linear models and neural networks with hidden layer size ranging from 1 to 8 (right panel).

are used for validation purposes. In this case, the CV model selection rule would suggests a neural network model with 4 hidden neurons while, following the BIC, also a neural network with 2 neurons would be appropriate. On the contrary, by using the proposed test, there is no superior predictive ability of neural network models with respect to linear models (see figure 4, left panel). This is confirmed by the distribution of the absolute predictive errors reported in figure 4, right panel: the linear model and the neural networks with 2 or 4 neurons in the hidden layer perform similarly. This latter result is also supported by the Brunner Munzel test and the Wilcoxon rank sum test whose p-values are, respectively, equal to 0.219 and 0.218. In this example, the regression model uses 17 explanatory variables so networks with 2 or 4 hidden neurons include 39 or 77 parameters, respectively. These networks appear to be heavily overparametrized, with no clear advantages in terms of predictive ability.

6. **Concluding remarks.** In this work, the main point was to introduce a strategy for the selection of the hidden layer size in feedforward neural network models. The numerical examples and the Monte Carlo experiment show that looking at the predictive ability of the model, simply measured by statistical indexes of predictive accuracy, might be misleading. In this case, the selected model might be overparametrized with heavy consequences on the generalization ability of the network. A better approach should be based on testing procedures of superior predictive ability. However, this strategy generates a sequence of tests and as a consequence the data snooping problem arises. This multiple testing structure, which is inherent to most model selection strategies, can be effectively addressed by reality check type tests. The proposed testing procedure, which takes under control the familywise error rate, is able to select parsimonious neural network models with the highest predictive accuracy. The real data analysis also supports this latter conclusion showing also that the CV and the BIC might lead to neural network models much more complex than necessary.

Acknowledgments. The authors gratefully acknowledge the helpful comments of two anonymous referees which greatly improved the final version of the paper. The authors also acknowledge the support from the University of Salerno grant program "Sistema di calcolo ad alte prestazioni per l'analisi economica, finanziaria e statistica (High Performance Computing - HPC) - prot. ASSA098434, 2009.

REFERENCES

[1] D. K. Agrafiotis, W. Cedeño and V. S. Lobanov, On the use of neural network ensembles in QSAR and QSPR, *J. Chem. Inf. Comput. Sci.*, **42** (2002), 903–911.

[2] A. R. Barron, Universal approximation bounds for superposition of a sigmoidal function, *IEEE Trans. Inform. Theory*, **39** (1993), 930–945.

[3] J. K. Basu, D. Bhattacharya and T. Kim, Use of artificial neural network in pattern recognition, *International Journal of Software Engineering and its Applications*, **4** (2010), 23–33.

[4] H. M. Cartwright, Artificial neural networks in biology and chemistry- the evolution of a new analytical tool, in *Artificial Neural Networks: Methods and Applications* (ed. D. J. Livingstone), Methods in Molecular Biology, Vol. 458, Humana Press, Totowa N.J., 2009, 1–13.

[5] X. Chen and H. White, Improved rates and asymptotic normality for nonparametric neural network estimators, *IEEE Trans. Inform. Theory*, **45** (1999), 682–691.

[6] W. Choe, O. K. Ersoy and M. Bina, Neural network schemes for detecting rare events in human genomic DNA, *Bioinformatics*, **16** (2010), 1062–1072.

[7] T. E. Clark and M. W. McCracken, Reality checks and comparison of nested predictive models, *J. Bus. Econom. Statist.*, **30** (2012), 53–66.

[8] T. E. Clark and M. W. McCracken, In-sample tests of predictive ability: A new approach, *J. Econometrics*, **170** (2012), 1–14.

[9] V. Corradi and N. R. Swanson, Predictive density evaluation, in *Handbook of Economic Forecasting*, Vol. 1 (eds. G. Elliott, C. W. J. Granger and A. Timmermann), North-Holland, 2006, 197–284.

[10] J. Devillers, *Neural Networks in QSAR and Drug Design*, Academic Press, London, 1996.

[11] R. De Veaux, J. Schumi, J. Schweinsberg and L. H. Ungar, Prediction intervals for neural networks via nonlinear regression, *Technometrics*, **40** (1998), 273–282.

[12] G. Elliot and A. Timmermann, Optimal forecast combinations under general loss functions and forecast error distribution, *Journal Econometrics*, **122** (2004), 47–79.

[13] J. H. Friedman, Multivariate adaptive regression splines, *Ann. Statist.*, **19** (1991), 1–141.

[14] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, 2nd edition, Springer, 2008.

[15] M. Jalali-Heravi, Neural network in analytical chemistry, in *Artificial Neural Networks: Methods and Applications* (ed. D. J. Livingstone), Methods in Molecular Biology Series, Vol. 458, Humana Press, Totowa, NJ, 2009, 78–118.

[16] K. Hornik, M. Stinchcombe and P. Auer, Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives, *Neural Computation*, **6** (1994), 1262–1275.

[17] J. T. G. Hwang and A. A. Ding, Prediction intervals for artificial neural networks, *J. Amer. Statist. Assoc.*, **92** (1997), 748–757.

[18] L. J. Lancashire, D. G. Powe, J. S. Reis-Filho, E. Rakha, C. Lemetre, B. Weigelt, T. M. Abdel-Fatah, A. R. Green, R. Mukta and R. Blamey, et al., A validated gene expression profile for detecting clinical outcome in breast cancer using artificial neural networks, *Breast Cancer Research and Treatment*, **120** (2010), 83–93.

[19] M. La Rocca and C. Perna, Variable selection in neural network regression models with dependent data: A subsampling approach, *Comput. Statist. Data Anal.*, **48** (2005), 415–429.

[20] M. La Rocca and C. Perna, Neural network modeling by subsampling, in *Computational Intelligence and Bioinspired Systems* (eds. J. Cabestany, A. Prieto and F. Sandoval), Lecture Notes in Computer Science, Vol. 3512, Springer, Berlin-Heidelberg, 2005, 200–207.

[21] M. La Rocca and C. Perna, Neural network modeling with applications to euro exchange rates, in *Computational Methods in Financial Engineering: Essays in Honour of Manfred Gili, Part II* (eds. E. Kontoghiorghes, B. Rustem and P. Winker), Springer, Berlin-Heidelberg, 2008, 163–189.

[22] C.-M. Kuan and T. Liu, Forecasting excange rates using feedforward networks and recurrent neural networks, *Journal of Applied Econometrics*, **10** (1995), 347–364.
[23] Y. Makovoz, Random approximates and neural networks, *J. Approx. Theory*, **85** (1996), 98–109.
[24] H. Merdun and O. Cinar, Artificial neural network and regression techniques in modelling surface water quality, *Environment Protection Engineering*, **36** (2010), 95–109.
[25] A. Ossen and S. M. Rügen, An analysis of the metric structure of the weight space of feedforward networks and its application to time series modelling and prediction, in *Proceedings of the 4th European Symposium on Artificial Neural Networks (ESANN96)*, Bruges, Belgium, April 24–26, 1996, 315–322.
[26] M. Qi and G. P. Zhang, An investigation of model selection criteria for neural network time series forecasting, *European Journal of Operational Research*, **132** (2001), 666–680.
[27] J. P. Romano and M. Wolf, Stepwise multiple testing as formalized data snooping, *Econometrica*, **73** (2005), 1237–1282.
[28] G. E. Schwarz, Estimating the dimension of a model, *Ann. Statist.*, **6** (1978), 461–464.
[29] J. Shao and D. Tu, *The Jackknife and the Bootstrap*, Springer Series in Statistics, Springer-Verlag, New York, 1995.
[30] T. Stamey, J. Kabalin, J. McNeal, I. Johnstone, F. Freiha, E. Redwine and N. Yang, Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate II radical prostactomy treated patients, *Journal of Urology*, **16** (1989), 1076–1083.
[31] N. R. Swanson and H. White, A model selection approach to real time macroeconomic forecasting using linear models and artificial neural networks, *The Review of Economics and Statistics*, **79** (1997), 540–550.
[32] I. V. Tetko, A. E. P. Villa and D. J. Livingstone, Neural network studies. 2. Variable selection, *J. Chem. Comput. Sci.*, **36** (1996), 794–803.
[33] R. Tibshirani, A comparison of some error estimates for neural network models, *Neural Computation*, **8** (1996), 152–163.
[34] A. Tsanas, M. A. Little, P. E. McSharry and L. O. Ramig, Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests, *IEEE Transactions on Biomedical Engineering*, **57** (2010), 884–893.
[35] B. Turlach, Discussion of Least angle regression by Efron, Hastie, Jon- stone and Tibshirani, *Ann. Statist.*, **32** (2004), 494–499.
[36] D. Urda, J. Subirats, L. Franco and J. M. Jerez, Constructive neural networks to predict breast cancer outcome by using gene expression profiles, in *Trends in Applied Intelligent Systems: 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part I*, Lecture Notes in Computer Science, 6096, Springer-Verlag, Berlin-Heidelberg, 2010, 317–326.
[37] H. White, Learning in artificial neural networks: A statistical perspective, *Neural Computation*, **1** (1989), 425–464.
[38] H. White, Connectionist nonparametric regression: Multi-layer feedforward networks can learn arbitrary mappings, *Neural Networks*, **3** (1990), 535–549.
[39] H. White, A reality check for data snooping, *Econometrica*, **68** (2000), 1097–1126.
[40] A. Yasri and D. Hartsough, Toward an optimal procedure for variable selection and QSAR model building, *J. Chem. Inf. Comput. Sci.*, **41** (2001), 1218–1227.

Received October 21, 2012; Accepted April 04, 2013.

E-mail address: larocca@unisa.it
E-mail address: perna@unisa.it