

Resampling techniques and neural networks: some recent developments for model selection

Tecniche di ricampionamento e reti neurali: alcuni recenti sviluppi per la selezione del modello

Michele La Rocca and Cira Perna

Dipartimento di Scienze Economiche e Statistiche, Università degli Studi di Salerno
e-mail: perna@unisa.it

Riassunto: In questo lavoro viene proposta una strategia di selezione della topologia di una rete neurale di tipo *feedforward*. La procedura, basata su una sequenza di test, permette di selezionare il modello in base alla sua accuratezza previsiva ed allo stesso tempo è costruita in modo da evitare il problema del *data snooping*. Essa è resa operativa mediante l'implementazione di tecniche di ricampionamento per la stima consistente della distribuzione campionaria delle statistiche test coinvolte.

Keywords: artificial neural networks, model selection, resampling, data snooping.

1. Introduction

Artificial neural networks are widely accepted as a potentially useful way of modeling non linear structures. They are mainly helpful when the underlying data generating process is not fully understood or when the nature of the relationships being modeled may display complex nonparametric structure.

A neural network can be considered as a parallel distributed model made up of simple data processing units. This parallel structure gives reason to its well known approximation capability: given a sufficiently large number of non linear terms and a suitable choice of the parameters, they are able to approximate arbitrary functions of variables arbitrarily well. Because of their flexibility, and because of demonstrated success in a variety of empirical applications, artificial neural networks have become the focus of considerable attention as a possible tool for forecasting economic and financial variables. In this context, they are usually competitive with respect to alternative nonparametric techniques.

Unfortunately, some problems in model building still have not been solved in a systematic way. Most of the techniques proposed in the literature use trial and error procedures, adequately combined with pruning of the network graph, based on weight selection or classical information criteria. These techniques generally lead to over-parametrized models with heavy consequence on overfitting and, consequently, poor ex-post forecast accuracy.

In a regression framework, the neural network model building strategy could be faced in a statistical perspective, relating it to the classical model selection approach. The selection of a proper topology should be based on statistical test procedures and on the out of sample predictive ability of the model, for a given loss function, in order to avoid over-parametrized structures. This strategy implies comparison of several different models and data snooping arises as a serious problem, since a given set of data is used more than once for inference or model selection. In the context of neural network models, the problem of

data snooping can be even more dangerous, due to the lack of theory supporting the model selection strategy, a general problem when dealing with non parametric techniques.

The neural network model structure and the complexity of the test procedures make the asymptotic distribution of the test statistics involved not one of the familiar tabulated distributions. Therefore, standard critical values cannot be used to construct inference on the parameters of interest. The problem can be overcome by using resampling techniques, such as the bootstrap or the subsampling, which can be used as a simulation tool to approximate the unknown sampling distributions. The use of resampling techniques in neural network models is increasing (Refenes and Zapranis, 1999; White and Racine, 2001; Giordano *et al.* 2002, 2004; La Rocca and Perna, 2005a,b *inter alia*) since they provide good alternative to asymptotic results giving better approximation to the actual sampling distribution.

In this paper we propose a strategy for neural network model selection which allows to select both the input variables and the hidden layer size with a strong statistical perspective. It is based on a sequence of tests and, to avoid the data snooping problem, familywise error rate is controlled by using the reality check of White (2000), which allows for the joint comparison of multiple misspecified models against a given benchmark. The procedure requires the implementation of resampling techniques in order to obtain valid asymptotic critical values for the test.

The paper is organized as follows. In the next section, we describe the structure of the data generating process and the neural network model employed. In section 3 we discuss a strategy of model selection for neural networks. Numerical examples on simulated data are reported in section 4. Some concluding remarks close the paper.

2. The neural network model

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

The random variables Y_i represent targets (in the neural neural 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 play an important role in interpreting what is that is learned by 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

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

where $\varepsilon_i \equiv Y_i - g(\mathbf{X}_i)$ and $g : \mathbb{R}^d \rightarrow \mathbb{R}$ 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 systematic 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

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

where $\mathbf{w} \equiv (w_{00}, w_{01}, \dots, w_{0r}, \mathbf{w}_{11}^T, \dots, \mathbf{w}_{1r}^T)^T$ 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 $\tilde{\mathbf{x}} \equiv (1, \mathbf{x}^T)^T$ 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 $f(\mathbf{x}, \cdot) : \mathbf{W} \rightarrow \mathbf{R}$ is continuous for each x in the support of μ and $f(\cdot, \mathbf{w}) : \mathbf{R}^d \rightarrow \mathbf{R}$ is measurable for each $\mathbf{w}$ in $\mathbf{W}$.

Feedforward neural networks with sufficiently many hidden units and properly adjusted parameters can approximate any function arbitrarily well (Cybenko, 1989 *inter alia*). Convergence rates for hidden layer feedforward networks with sigmoidal activation functions, approximating a class of functions satisfying certain smoothness conditions, are given in Barron (1993).

Given a training set of N observations, estimation of the network weights (learning) is obtained by solving the optimization problem

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

where $q(\cdot)$ is a proper chosen loss function. Under general regularity conditions (White 1989), a weight vector $\hat{\mathbf{w}}_n$ solving equation (3) exists and converges almost surely to $\mathbf{w}_0$, which solves

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

provided that the integral exists and the optimization problem has a unique solution vector interior to $\mathbf{W}$. Observe that this is not necessarily true for neural network models in absence of appropriate restrictions since the parametrization of the network function is not unique and certain simple symmetry operations applied to the weight vector do not change the value of the output. For a sigmoid activation function ψ centered around 0, these symmetry operations correspond to an exchange of hidden units and multiplying all weights of connections going into and out of a particular hidden unit by -1. The permutability of hidden units generally results in a non-unique $\mathbf{w}_0$ as there are numerous distinct weight vectors yielding identical network outputs. In any case this may not be a main concern for different reasons. Firstly, several authors provide sufficient conditions to ensure uniqueness of $\mathbf{w}_0$ in a suitable parameter space $\mathbf{W}$ for specific network configurations. Particularly, for the case of sigmoidal activation functions with $\psi(-a) = -\psi(a)$, it is possible to restrict attention only to weight vectors with $w_{01} \geq w_{02} \geq \dots \geq w_{0r}$ (see Ossen and Rügen, 1996). Secondly, the possible presence of multiple minima has no essential effect, at least asymptotically, for solutions to equation (4) (see White, 1989). Thirdly, several global optimization strategies (simulation annealing, genetic algorithms, etc.) are available to avoid to be trapped in local minima and they have been successfully employed in neural network modeling. Finally, when the focus is on prediction, it can be shown that the unidentifiability can be overcome and the problem disappears (Hwang and Ding, 1997).

Asymptotic normality of the weight vector estimator can also be established. Let $l(\mathbf{z}, \mathbf{w}) \equiv q(y, f(\mathbf{x}, \mathbf{w}))$ and denote by ∇ and ∇^2 the gradient and the Hessian operators, respectively. Assume that $\mathbf{A}^* \equiv \mathbb{E}(\nabla^2 l(\mathbf{z}, \mathbf{w}_0))$ and $\mathbf{B}^* \equiv \mathbb{E}(\nabla l(\mathbf{z}, \mathbf{w}_0) \nabla l(\mathbf{z}, \mathbf{w}_0)^T)$ are nonsingular matrices. If general regularity conditions hold then

$$\sqrt{n}(\hat{\mathbf{w}}_n - \mathbf{w}_0) \xrightarrow{d} N(\mathbf{0}, \mathbf{C}^*)$$

where $C^* = A^{*-1} B^* A^*$ (see White 1989, theorem 2, p. 457).

These results make it possible to test hypotheses about the connection strengths which can be of great help in defining pruning strategies with a strong inferential base.

Better approximations to the actual distribution than standard asymptotics can be obtained by using the bootstrap. It is a good alternative to analytical results and it often provides higher accuracy. The bootstrap may be used for calculating confidence intervals for predictions and critical values for statistical tests, and it is also helpful for model selection and automatic choice of the amount of smoothing in nonparametric frameworks.

Several authors have used the bootstrap in a neural network framework for estimating sampling variability (Baxt and White, 1995; Refenes and Zapranis, 1999 *inter alia*) and for calculating critical values for statistical tests (White and Racine, 2001). LeBaron (1997) propose a technique which combines evolutionary optimisation algorithms, for searching the large number of potential network architecture, along with bootstrap based statistical tests, for estimating objectives out of sample. Weigend and LeBaron (1994) and Tibshirani (1996) discuss the use of the bootstrap to determine the quality and reliability of a neural network predictor.

If the model is correctly specified, that is the mapping g can be represented exactly by a network of the form considered, the neural network can be considered the particular case of parametric nonlinear regression models. In this case the validity and second-order efficiency of the bootstrap has been described and investigated in practice (Huet *et al.*, 1990 *inter alia*) and those results extent immediately to neural network models.

In the more general case of misspecified models, formal proof of the consistency of the bootstrap can be found in Franke and Neumann (2000).

Alternatively, under basically the same assumptions of asymptotic normality, the subsampling (Politis *et al.* 1999) gives consistent results when constructing test procedure and confidence intervals for the weights.

3. Model selection for neural networks

When using neural networks, one of the most critical problems is how to select appropriate network architectures for the problem at hand. This requires selecting both an appropriate number of hidden units and a suitable set of the explicative variables and, as a consequence, the connections thereof.

Practitioners usually refer to information criteria such as the Akaike information criterion (AIC) and the Schwarz information criterion (SIC). These criteria add a complexity penalty to the usual sample log-likelihood, and the model that optimizes this penalized log likelihood is preferred. Generally, the SIC, imposing a more severe penalty than the AIC, delivers the most conservatives models (i.e. least complex) and has been found to perform well in selecting forecasting models in other contexts. Therefore, in the neural network framework, SIC is usually preferred (Franses and Draisma, 1997 *inter alia*). However, many statistical studies agree on the failure of these measures in choosing the best forecasting model. For example, Swanson and White (1997) and, more recently, Qi and Zhang (2001) show that they are not able to provide a reliable guide to out of sample performance since all these procedures generally lead to over-parameterized models with heavy consequence on overfitting and, consequently, poor ex-post forecast accuracy. Kuan and Liu (1995) instead propose the predictive stochastic

complexity criterion, which is based on forward validation, a better choice for forecasting purposes.

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 outperform *significantly* its competitors. In recent years there is a growing literature addressing the problem of comparing different models and theories via use of predictive performance and predictive accuracy test (Corradi and Swanson, 2004 and 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 give 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 in the model yielding to the result. In other words looking long enough and hard enough at a given data set will often reveal one or more forecasting models that look good but are in fact useless (see White, 2000; Romano and Wolf, 2005 *inter alia*).

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

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

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

Assume then that $k + 1$ alternative forecasting neural network models are available, namely $f(\mathbf{x}^j, \mathbf{w}^j)$, $j = 0, 1, \dots, k$. The models can differ in hidden layer size and/or in number and type of explanatory variables. Model $f(\mathbf{x}^0, \mathbf{w}^0)$ is the benchmark model. In our framework a sensible choice is the linear model, that is a neural network with skip layer and $r = 0$ neurons in the hidden layer.

Let the generic forecast error be $u_{j,\tau} = Y_\tau - f(\mathbf{X}_\tau^j, \mathbf{w}_0^j)$, $j = 0, 1, \dots, k$ where $\mathbf{w}_0$ is defined as in the previous sections. Let h be a proper chosen loss function (Elliot and Timmerman, 2004) and define

$$\theta_j = \mathbb{E}(h(u_{0,\tau}) - h(u_{j,\tau})), j = 1, 2, \dots, k. \quad (6)$$

Clearly, if model j beats the benchmark (i.e. 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

$$H_j : \theta_j \leq 0 \quad vs \quad H'_j : \theta_j > 0 \quad (7)$$

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 (FWE), defined as the probability of rejecting at least one of the true null hypotheses, should be taken under control.

The FWE can be controlled by using the well known Bonferroni method or stepwise procedures which are more powerful. Unfortunately both procedures are conservative

since they do not take into account the dependence structure of the individual p -values. A possible solution can be obtained by using the reality check proposed by White (2000) which can be easily extended to our framework.

For a given loss function, the reality check tests the null hypothesis that a benchmark model (i.e. model 0) performs equal or better than all competitor models (i.e. models $1, \dots, k$). The alternative is that at least one competitor performs better than the benchmark. Formally we have

$$H_0 : \max_{j=1, \dots, k} \theta_j \leq 0 \quad vs \quad H_1 : \max_{j=1, \dots, k} \theta_j > 0. \quad (8)$$

Following a common practice often used to select the best predictive model, the sample of size N is splitted into $N = R + P$ observations where R observations are used for estimation and P observations are used for predictive evaluation. Let $\hat{u}_i = Y_i - f(\mathbf{X}_i^j, \hat{\mathbf{w}}_R^j)$, $i = R + 1, \dots, N$, where $f(\mathbf{X}_i^j, \hat{\mathbf{w}}_R^j)$ is the model estimated on the data set $\{(Y_i, \mathbf{X}_i^j), i = 1, \dots, R\}$. Following White (2000) define the statistic

$$S_P = \max_{j=1, \dots, k} S_P(0, j) \quad (9)$$

where

$$S_P(0, j) = \frac{1}{\sqrt{P}} \sum_{i=R+1}^N \{h(\hat{u}_{0,i}) - h(\hat{u}_{j,i})\}, \quad j = 1, \dots, k.$$

It can be shown that, if general regularity conditions hold, under H_0 , as $P, R \rightarrow \infty$,

$$\max_{j=1, \dots, k} \{S_P(0, j) - \sqrt{P}\theta_j\} \xrightarrow{d} \max_{j=1, \dots, k} S(0, j). \quad (10)$$

The $k \times 1$ vector $S = (S(0, 1), S(0, 2), \dots, S(0, k))$ has gaussian distribution with zero mean and covariance matrix defined as

$$V = \lim_{N \rightarrow \infty} \text{var} \left(\frac{1}{\sqrt{P}} \sum_{i=R+1}^N \mathbf{v}_i \right)$$

where the generic element of vector $\mathbf{v}_i$ is defined as $v_{i,j} = h(u_{0,i}) - h(u_{j,i})$. The matrix V is supposed to be positive semi-definite.

Since it is well known that the maximum of a Gaussian process is not Gaussian in general, standard critical values cannot be used to conduct inference on S_P . Alternatively resampling techniques such as the subsampling or the bootstrap can be used.

The bootstrap analogue of the statistic S_P can be computed as

$$S_P^* = \max_{j=1, \dots, k} S_P^*(0, j) \quad (11)$$

where

$$S_P^*(0, j) = \frac{1}{\sqrt{P}} \sum_{i=R+1}^N \{(h(\hat{u}_{0,i}^*) - h(\hat{u}_{0,i})) - (h(\hat{u}_{j,i}^*) - h(\hat{u}_{j,i}))\} \quad (12)$$

with $\hat{u}_{j,i}^* = Y_i^* - f(\mathbf{X}_i^{*j}, \hat{\mathbf{w}}_R^j)$ and $(Y_i^*, \mathbf{X}_i^{*j})$ denote the resampled data. Note that the bootstrap statistics contain only estimators based on the original sample and this is particular convenient when dealing with neural network models. If estimation is needed for each bootstrap sample, the procedure will become soon not feasible in our framework.

The bootstrap procedure is consistent in the neural network framework. Under general regularity conditions, it can be shown that, if $q = h$, for $P, R \rightarrow \infty$

$$\Pr \left(\sup_{v \in \mathbb{R}} |\Pr_*(S_P^* \leq v) - \Pr(S_P^\kappa \leq v)| > \varepsilon \right) \rightarrow 0 \quad (13)$$

where $\Pr_*$ denotes probability induced by the bootstrap resampling scheme and

$$S_P^\kappa = \max_{j=1, \dots, k} \left\{ S_P(0, j) - \sqrt{P} \theta_j \right\}$$

As usual the bootstrap procedure can be implemented by Monte Carlo. For any bootstrap replication, compute the bootstrap statistic, S_P^* . Perform B bootstrap replications (B large) and compute the quantiles of the empirical distribution of the B bootstrap statistics. Reject the null hypothesis H_0 if S_P is greater than the $(1 - \alpha)$ th-percentile. Otherwise, do not reject.

The bootstrap procedure can be implemented as described in algorithm (1).

Note that, to estimate a percentile, B should be quite large (usually $B > 1000$) and that indexes are generated just once at the beginning of the procedure. Moreover we assume that $h = q$.

4. Numerical examples

In order to evaluate the ability of the procedure to select a proper model for a given data generating process, we use simulated data sets with known structure.

The first is a linear model (M1) with two regressors defined as:

$$Y = \mathbf{X}\mathbf{1} + \varepsilon$$

where $\mathbf{X} = (X_1, X_2)^T$ are drawn from the uniform distribution, ε is a standard Gaussian and $\mathbf{1}$ denotes a column vector of ones of appropriate length. This model can be correctly modeled by using a network, with skip layer, two input units and zero hidden units.

Model M2 is the same model used in Tibshirani (1996) 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 gaussian with zero mean and variance equal to 0.7. This gives a signal-to-noise ratio roughly equal to 1.2. 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.

Model M3 is the same model used in De Veaux et al. (1998) and it is defined as

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

Algorithm 1 Bootstrap resampling algorithm.

- 1: Fix P, R such that $P + R = N$.
 - 2: Fix B , the number of bootstrap replicates.
 - 3: Generate B sets of random observation indexes of length P , namely $\{\theta_b(i), i = R + 1, \dots, N; b = 1, \dots, B\}$.
 - 4: $M_0 \leftarrow -\Delta$, with Δ finite big constant.
 - 5: $M_0^{(b)} \leftarrow -\Delta$, with Δ finite big constant, $b = 1, \dots, B$.
 - 6: $\hat{\mathbf{w}}_R^0 \leftarrow \arg \min_{\mathbf{w} \in \mathbf{W}} \frac{1}{R} \sum_{i=1}^R q(Y_i, f(\mathbf{X}_i^0, \mathbf{w}))$.
 - 7: $\hat{u}_{0,i} \leftarrow Y_i - f(\mathbf{X}_i^0, \hat{\mathbf{w}}_R^0), i = R + 1, \dots, N$.
 - 8: $\hat{h}_{0,i} \leftarrow h(\hat{u}_{0,i}), i = R + 1, \dots, N$.
 - 9: **for** $j = 1$ to k **do**
 - 10: $\hat{\mathbf{w}}_R^j \leftarrow \arg \min_{\mathbf{w} \in \mathbf{W}} \frac{1}{R} \sum_{i=1}^R q(Y_i, f(\mathbf{X}_i^j, \mathbf{w}))$.
 - 11: $\hat{u}_{j,i} \leftarrow Y_i - f(\mathbf{X}_i^j, \hat{\mathbf{w}}_R^j), i = R + 1, \dots, N$.
 - 12: $\hat{h}_{j,i} \leftarrow h(\hat{u}_{j,i}), i = R + 1, \dots, N$
 - 13: $S_P(0, j) \leftarrow \frac{1}{\sqrt{P}} \sum_{i=R+1}^N \{\hat{h}_{0,i} - \hat{h}_{j,i}\}$.
 - 14: $M_j \leftarrow \max\{S_P(0, j), M_{j-1}\}$.
 - 15: **for** $b = 1$ to B **do**
 - 16: $\hat{u}_{0,i}^{(b)} = Y_{\theta(b)} - f(\mathbf{X}_{\theta(b)}^0, \hat{\mathbf{w}}_R^0)$
 - 17: $\hat{u}_{i,j}^{(b)} = Y_{\theta(b)} - f(\mathbf{X}_{\theta(b)}^j, \hat{\mathbf{w}}_R^j)$.
 - 18: $S_P^{(b)}(0, j) \leftarrow \frac{1}{\sqrt{P}} \sum_{i=R+1}^N \left\{ \left(h(\hat{u}_{0,i}^{(b)}) - \hat{h}_{0,i} \right) - \left(h(\hat{u}_{j,i}^{(b)}) - \hat{h}_{j,i} \right) \right\}$
 - 19: $M_j^{(b)} \leftarrow \max\{S_P^{(b)}(0, j), M_{j-1}^{(b)}\}$.
 - 20: **end for**
 - 21: **end for**
 - 22: **return** $p\text{-value} \leftarrow \frac{1}{B} \sum_{b=1}^B \mathbb{I}(M_k^{(b)} > M_k)$.
-

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 variable. Clearly, the number of the neurons in the hidden layer is unknown and the model we try to identify is, by construction, misspecified.

We have considered $N = 600, R = 400, P = 200$ and $B = 4999$.

In table 1 we consider values of the test statistics for different input neurons, from X_1 to X_6 , and different hidden layer size, from 1 to 6. It is clear that for model M1 and M2 the proposed procedure is able to identify the correct data generating process. In the first case, the p -values of the tests are all greater than 0.50 and so the benchmark (i.e. the linear model) shows better expected predictive performance with respect to neural networks of all orders and sizes. In the case of model M2, the values of the test statistics do not change significantly starting from a neural network model with 4 inputs and 2 hidden layer neurons. In the case of model M3, clearly test statistics stabilize starting from a model with 3 inputs (as expected) and 4 hidden layer neurons. The small increases in

some test statistics possibly are not significant.

Table 1: Values of the test statistics for different input neurons sets and different hidden layer size. The benchmark model is a neural network with skip layer and zero hidden neurons. Values in *italics* correspond to p -values greater than 0.50. Values in **bold** correspond to p -values less than 0.005.

Model	Inputs / Size	1	2	3	4	5	6
M1	1	<i>-0.116</i>	<i>-0.116</i>	<i>-0.116</i>	<i>-0.116</i>	<i>-0.116</i>	<i>-0.116</i>
	2	<i>-0.290</i>	<i>-0.290</i>	<i>-0.290</i>	<i>-0.290</i>	<i>-0.290</i>	<i>-0.290</i>
	3	<i>-0.721</i>	<i>-0.721</i>	<i>-0.721</i>	<i>-0.721</i>	<i>-0.721</i>	<i>-0.721</i>
	4	<i>-0.986</i>	<i>-0.986</i>	<i>-0.986</i>	<i>-0.986</i>	<i>-0.986</i>	<i>-0.986</i>
	5	<i>-0.844</i>	<i>-0.844</i>	<i>-0.844</i>	<i>-0.844</i>	<i>-0.844</i>	<i>-0.844</i>
	6	<i>-0.873</i>	<i>-0.873</i>	<i>-0.873</i>	<i>-0.873</i>	<i>-0.873</i>	<i>-0.873</i>
M2	1	<i>-0.477</i>	<i>-0.477</i>	<i>-0.477</i>	<i>-0.477</i>	<i>-0.477</i>	<i>-0.477</i>
	2	4.541	4.541	4.541	4.541	4.541	4.541
	3	2.603	5.741	5.741	5.741	5.741	5.741
	4	3.060	12.122	12.122	12.122	12.122	12.122
	5	3.058	12.121	12.121	12.121	12.121	12.121
	6	3.060	11.921	11.921	11.921	11.921	11.921
M3	1	0.748	2.159	2.159	2.159	2.159	2.159
	2	0.752	2.143	4.857	4.857	4.857	4.857
	3	0.807	2.722	5.391	7.215	7.222	7.249
	4	0.824	2.737	5.402	7.226	7.232	7.246
	5	0.886	2.811	5.531	7.264	7.269	7.277
	6	0.816	2.826	5.520	7.262	7.267	7.295

In order to get a deeper insight, in figures 1, 2 and 3 we report the prediction performance, with respect to the benchmark model, of neural network models with increasing hidden layer size. The segments refer to bootstrap confidence intervals for the parameters $\max_{j=1,\dots,r} \theta_j$ with hidden layer size $r = 1, \dots, 6$ and confidence level equal to 0.95. Each panel refers to different choice of the input variables. Again, for models M1 and M2 the proposed procedure is able to identify the correct data generating process while for model M3 the identified model is a neural network with 3 input variables and hidden layer size equal to 4. Moreover, an aspect which arises from all the figures is that the predictive performance is improved when relevant variables are included into the model while it remains unchanged when adding irrelevant ones.

5. Concluding remarks

In this paper we have proposed a strategy for neural network model selection. It is based on a sequence of tests and it uses extensively the bootstrap in order to obtain valid asymptotic critical values for the test. The results on simulated data are encouraging. In any case, several different aspects should be further investigated. In particular, extended Monte carlo simulations and applications to real data are needed to study the overall performance of the test procedure and the impact on the splitting of the data. Moreover, it

Figure 1: Model M1. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark.

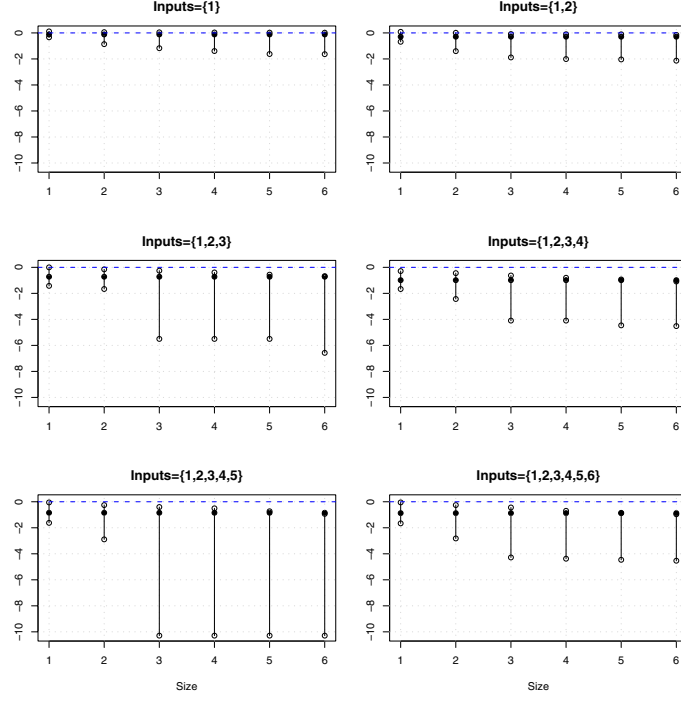


Figure 2: Model M2. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark. .

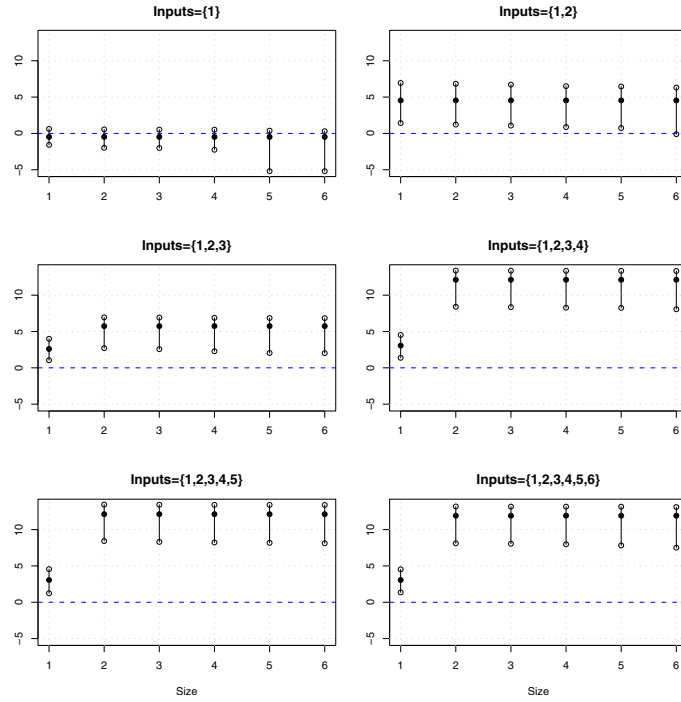
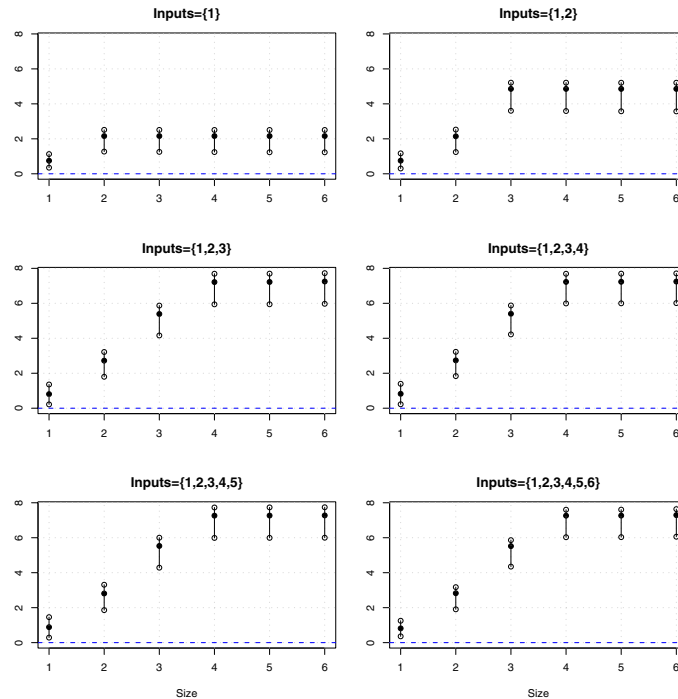


Figure 3: Model M3. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark.



would be of interest to study the advantage in using studentized version of the reality check statistic or multiple step test procedures such as in Romano and Wolf (2005). Alternatively, the test procedure can be calibrated via subsampling (Politis *et al.*, 1999) and a comparison with the bootstrap approach is under investigation.

References

- Barron A. (1993) Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Transaction on Information Theory*, 39, 930–945.
- Baxt, W. G., White, H. (1995) Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction, *Neural Computation*, 7, 624–638.
- Corradi V., Swanson, N. R. (2004) Predictive density accuracy tests, *Working paper, Rutgers University and Queen Mary University of London*.
- Cybenko G. (1989) Approximation by superposition of a sigmoidal function, *Mathematics of Control, Signal and Systems*, 2, 303–314.
- De Veaux R., Schumi J., Schweinsberg J., Shellington D., Ungar L. H. (1998) Prediction intervals for neural networks via nonlinear regression, *Technometrics*, 40, 277–282.
- Elliot G., Timmermann A. (2004) Optimal forecast combinations under general loss functions and forecast error distribution, *Journal of Econometrics*, 122, 47–79.
- Franke J., Newmann M. H. (2000) Bootstrapping neural networks, *Neural computation*, 12, 1929–1949.

- Franses P.H.; Draisma G. (1997) Recognizing changing seasonal patterns using artificial neural networks, *Journal of Econometrics*, 81, 273–280.
- Giordano F., La Rocca M., Perna C. (2002) Standard error estimation in neural network regression models: the sieve bootstrap approach, in *Neural Nets*, R. Tagliaferri and M. Marinaro (eds.), Springer-Verlag.
- Giordano F., La Rocca M., Perna C. (2004) Bootstrap variable selection in neural network regression models, in *Advances in Multivariate Data Analysis*, H.-H. Bock, M. Chiodi, A. Mineo (eds.), Springer-Verlag, Heidelberg-Berlin.
- Huet, S., Jolivet, E., Messeean, A. (1990) Some simulation results about confidence intervals and bootstrap methods in nonlinear regression, *Statistics*, 21, 369–432.
- Hwang J. T. G., Ding A. A. (1997) Prediction intervals for artificial neural networks, *Journal of the American Statistical Association*, 92, 748–757.
- Kuan C.M., Liu T. (1995) Forecasting exchange rates using feedforward networks and recurrent neural networks, *Journal of Applied Econometrics*, 10, 347–364.
- La Rocca M., Perna C. (2005a) Variable selection in neural network regression models with dependent data: a subsampling approach, *Computational Statistics and Data Analysis*, 48, 415–429.
- La Rocca M., Perna C. (2005b) Neural Network Modeling by Subsampling, in *Computational Intelligence and Bioinspired Systems*, J. Cabestany, A. Prieto and F. Sandoval (Eds.), Lecture Notes in Computer Science 3512, 2005, Springer.
- LeBaron B. (1997) An evolutionary bootstrap approach to neural network pruning and generalization, *Working Paper, Department of Economics, University of Wisconsin - Madison*.
- Ossen A., Rügen S. M. (1996) An analysis of the metric structure of the weight space of feedforward networks and its application to time series modelling and prediction, *Proceedings of the 4th European Symposium on Artificial Neural Networks (ESANN96)*, Bruges, Belgium, April 24 – 26, 1996, 315–322.
- Politis D. N., Romano J. P., Wolf, M. (1999). *Subsampling*, Springer-Verlag, NY.
- Qi M., Zhang G.P. (2001) An investigation of model selection criteria for neural network time series forecasting, *TEuropean Journal of Operational Research*, 132, 666–680.
- Refenes, A.-P.N., Zapraniš, A. D (1999) Neural model identification, variable selection and model adequacy, *Journal of Forecasting*, 18 299–332.
- Romano J. P., Wolf M. (2005) Stepwise multiple testing as formalized data snooping, *Econometrica*, 73, 1237–1282.
- Swanson N. R., White H. (1997) A model selection approach to real time macroeconomic forecasting using linear models and artificial neural networks, *The Review of Economics and Statistics*, 79, 540–550.
- Tibshirani R. (1996) A comparison of some error estimates for neural network models, *Neural Computation*, 8, 152–163.
- Weigend A. S., LeBaron B. (1994) Evaluating Neural Network Predictors by Bootstrapping in *Proceedings of the International Conference on Neural Information Processing (ICONIP'94)*, Seoul, Korea, 1207–1212.
- White H. (1989) Learning in artificial neural networks: a statistical perspective, *Neural Computation*, 1, 425–464.
- White H. (2000) A reality check for data snooping, *Econometrica*, 68, 1097–1126.
- White, H. Racine, J. (2001) Statistical inference, the bootstrap, and neural network modeling with application to foreign exchange rates, *IEEE Transactions on Neural Networks*, 12, 657–673.