1
Model Selection for Neural Network Models: A Statistical Perspective

Michele La Rocca and Cira Perna

1.1
Introduction

It is generally accepted that liner analysis often gives poor performances in approximating real data. Therefore, although it is easy to handle and fast to compute, and many statistical results are available, it cannot be extensively used especially when complex relationships are recognized in the data. In these contexts, it is common the use of non linear analysis which can successfully be employed to reveal these patterns.

However, parametric analysis, both linear and nonlinear, requires an "a priori" specification of the links among the variables of interest, which is not always possible. Therefore, even if the results have the advantage of the interpretability (in the sense that the model parameters are often associated to quantities with a "physical" meaning), misspecification problem can arise and can affect seriously the results of the analysis. In this respect, nonparametric analysis seems to be a more effective statistical tool due to its ability to model non-linear phenomena with few (if any) "a priori" assumptions about the nature of the data generating process. Well-studied and frequently used tools in nonparametric analysis include nearest neighbours regression, kernel smoothers, projection pursuit, alternating conditional expectations, average derivative estimation, and classification and regression trees.

In this context, computational network analysis forms a field of research which has enjoyed rapid expansion and increasing popularity in both the academic and the research communities, providing an approach that can potentially lead to better non-parametric estimators and providing an interesting framework for unifying different non-parametric paradigms, such as nearest neighbours, kernel smoothers, and projection pursuit.

Computational network tools have the advantage, with respect to other non-parametric techniques, to be very flexible tools able to provide, under very general conditions, an arbitrarily accurate approximation to an unknown target the function of interest. Moreover, they are expected to perform better than other non-parametric methods since the approximation form is not so sensitive to the

Computational Network Theory: Theoretical Foundations and Applications, First Edition.
Edited by Matthias Dehmer, Frank Emmert-Streib, and Stefan Pickl.

increasing data space dimension (absence of "curse of dimensionality"), at least within particular classes of functions.

However, a major weakness of neural modeling is the lack of established procedures for performing tests for misspecified models and tests of statistical significance for the various parameters that have been estimated. This is a serious disadvantage in applications where there is a strong interest for testing not only the predictive power of a model or the sensitivity of the dependent variable to changes in the inputs but also the statistical significance of the result at a specified level of confidence. Significant correction for multiple hypothesis testing has been a central concern in many fields of research that deal with large sets of variables and small samples and where, as a consequence, the control of false positives becomes an important problem.

In such context data snooping, which occurs when a given set of data is used more than once for inference or model selection, it 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 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 computational network analysis, which is basically atheoretical.

The aim of this chapter is to develop model selection strategies useful for computational network analysis based on statistical inference tools. In particular, we propose hypothesis testing procedures both for variable selection and model adequacy. The approach takes into account the problem of data snooping and uses resampling techniques to overcome the analytical and probabilistic difficulties related to the estimation of the sampling distribution of the test statistics involved. The chapter is organized as follows. Section 1.2 describes the structure of the data generating process and the neural network model considered. In Section 1.3, we address the problem of input selection and in Section 1.4 the selection of the hidden layer size. In both cases, application to simulated and real data are considered. Some remarks conclude the papers.

1.2 Feedforward 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 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 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 play an important role in

interpreting what is learned by artificial neural network models. If $\mathbb{E}\left(Y_i\right) < \infty$, then $\mathbb{E}\left(Y_i \left| \mathbf{X}_i \right.\right) = g\left(\mathbf{X}_i\right)$ and we can write

$$Y_i = g\left(\mathbf{X}_i\right) + \varepsilon_i \tag{1.1}$$

where $\varepsilon_i \equiv Y_i - g\left(\mathbf{X}_i\right)$ and $g : \mathbb{R}^d \to \mathbb{R}$ is a measurable function.

The function g embodies the systematic part of the stochastic relation between Y_i and $\mathbf{X}_i$. On the data-generating process, we assume also that:

1) $\mathbf{Z}_i$ are independent and identically distributed (i.i.d.) random vectors; $\{\varepsilon_i\}$ are independent of $\{\mathbf{X}_i\}$, $\mathbb{E}(\varepsilon_i) = 0$ and $\mathbb{E}(\varepsilon_i^2) = \sigma_\varepsilon^2 < \infty$.
2) The random vectors $\mathbf{X}_i$ have a compact support, say $\chi_1 \in \mathbb{R}^d$.

These conditions guarantee that Y_i has finite variance.

The function g can be approximated by a single hidden layer feed-forward neural network $NN(d, r)$ defined as:

$$f(\mathbf{x}, \mathbf{w}) = w_{00} + \sum_{j=1}^{r} w_{0j} \psi\left(\tilde{\mathbf{x}}^T \mathbf{w}_{1j}\right) \tag{1.2}$$

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 (Eq. (1.2)) 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 μ and $f(\cdot, \mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ is measurable for each $\mathbf{w}$ in $\mathbf{W}$.

On the neural network model, we assume that

1) The activation function, $\psi(\cdot)$, is sigmoidal.
2) The function $\psi(\cdot)$ has all the derivatives.

This latter assumption guarantees (Hornik, Stinchcombe, and Auer, 1994 *inter alia*) that feedforward neural networks with sufficiently many hidden units and properly adjusted parameters can approximate any function arbitrarily well. Moreover, Barron (1993) gives convergence rates for hidden layer feedforward networks with sigmoidal activation functions, approximating a class of functions that satisfy certain smoothness conditions.

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

$$\min_{\mathbf{w} \in \mathbf{W}} \frac{1}{N} \sum_{i=1}^{N} q\left(Y_i, f\left(\mathbf{X}_i, \mathbf{w}\right)\right) \tag{1.3}$$

where $q(\cdot)$ is a proper chosen loss function. Under general regularity conditions White (1989), a weight vector $\hat{\mathbf{w}}_n$ solving Eq. (1.3) exists and converges almost surely to $\mathbf{w}_0$, which solves

$$\min_{\mathbf{w} \in \mathbf{W}} \int q\left(y, f(\mathbf{x}, \mathbf{w})\right) d\pi(\mathbf{z}) \tag{1.4}$$

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 the 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 $w_{01} \geqslant w_{02} \geqslant \cdots \geqslant w_{0r}$ (see Ossen and Rüugen, 1996). Secondly, the possible presence of multiple minima has no essential effect, at least asymptotically, for solutions to Eq. (1.4) (see White, 1989). Thirdly, several global optimization strategies (simulation annealing, genetic algorithms, etc.) are available to avoid being 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. In particular, let $l(\mathbf{z}, \mathbf{w}) \equiv q\left(y, f(\mathbf{x}, \mathbf{w})\right)$ and denote by ∇ and ∇^2 the gradient and the Hessian operators, respectively. Assume that $\mathbf{A}^* \equiv \mathbb{E}\left(\nabla^2 l\left(\mathbf{z}, \mathbf{w}_0\right)\right)$ and $\mathbf{B}^* \equiv \mathbb{E}\left(\nabla l\left(\mathbf{z}, \mathbf{w}_0\right) \nabla l\left(\mathbf{z}, \mathbf{w}_0\right)^T\right)$ are nonsingular matrices. If general regularity conditions hold, then

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

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

These results make it possible to test the hypotheses about the connection strengths, which can be of great help in defining pruning strategies with a strong inferential base. However, focusing on single weights might be misleading due to the black-box nature of the neural network model and better model selection strategies become necessary to select appropriate network architectures for the problem at hand.

1.3 Model Selection

Model selection in neural network models requires selecting both an appropriate number of the hidden units and a suitable set of explicative variables and, as a consequence, the connections thereof. The "atheoretical" nature of the tool, employed

for the lack of knowledge about the functional form of the data generating process, and the intrinsic misspecification of the model, makes this problem a hard task.

The problem is not a novel one and a number of different and effective solutions have been proposed. The most popular approaches are pruning, stopped training, and regularization. Although these techniques may lead to satisfactory results, they focus on single weights and this can be misleading due to the black-box nature of the neural network model. Indeed, they do not give any information on the most significant variables, which is useful in any model building strategy and, moreover, different topologies can achieve the same approximation accuracy. Therefore, a proper choice of the network topology cannot be just based on complexity reason and should also take into account model plausibility. All the techniques based on weight selection are much more on the side of computational standpoint than on the side of a statistical perspective. Instead, it would be of some interest to look at the choice of the network topology by including it in the classical statistical model selection approach.

In this perspective, information criteria such as the Akaike information criterion (AIC) and the Schwarz information Criterion (SIC) could be used. 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 conservative 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 that these measures should be used with care in choosing the best model in a neural network context. Indeed, Swanson and White (1997) and Qi and Zhang (2001) show that these procedures might lead to over-parameterized models with heavy consequence on overfitting and 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 depends 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. Moreover, they lack a strong inferential statistical perspective and, usually, they contain a strong judgemental component not giving explicitly any information on the most "significant" variables.

A better model selection strategy should be faced in a statistical framework, relating it to the classical model selection approach, emphasizing the different role in the model of the explanatory variables and of the hidden layer neurons. In a regression framework, input neurons are related to the explanatory variables (useful for identification and interpretation of the model), while the hidden layer size has no clear interpretation, and it should be considered basically as a smoothing parameter taking into account the trade-off between estimation bias and variability.

However, while in principle, the hidden layer size could be chosen according to one of the many results available in the statistical literature, ranging from the information criteria based on the fitting, to the indexes based on prediction accuracy, the input selection should be addressed focusing on procedures for variable selection in regression models.

In this perspective, the model selection strategy discussed in the following, identifies both the input variables and the hidden layer size by using formal test procedures. Particularly, the input variables are selected by using relevance measures, while the hidden layer size is selected by looking at the predictive performance of the neural network model. Both procedures use extensively resampling techniques that are able to deliver consistent results under general assumptions, a very important requirement in a neural network framework.

1.3.1 Feature Selection by Relevance Measures

To select a proper set of input variables, we focus on a selection rule based on relevance measures (White and Racine, 2001; La Rocca and Perna, 2005a,b) following the usual strategy generally employed when selecting a model in the classical regression framework.

There are, of course, a lot of representative criteria that have traditionally been used to quantify the relevance of input variables in neural models. These relevance criteria, often referred to as sensitivity measures, are traditionally obtained by the computation of partial derivatives.

As in White and Racine (2001), the hypotheses that the independent variable X_j has no effect on Y, in model (Eq. 1.1) can be formulated as:

$$\frac{\partial g(\mathbf{x})}{\partial x_j} = 0, \forall x. \tag{1.5}$$

Of course, the function g is unknown but we equivalently investigate the hypotheses

$$f_j\left(\mathbf{x}; \mathbf{w}_0\right) = \frac{\partial f\left(\mathbf{x}; \mathbf{w}_0\right)}{\partial x_j} = 0, \forall x. \tag{1.6}$$

since f is known and $\mathbf{w}_0$ can be closely approximated.

The general form for relevance measures is,

$$\theta_j = \mathbb{E}\left[h|f_j(\mathbf{x}, \mathbf{w}_0\right] \tag{1.7}$$

where $h(.)$ is a proper chosen function and $\mathbb{E}[.]$ is the expected value w.r.t. the probability measure of the vector of the explanatory variables.

As relevance measures, several alternative functions can be used; for example, the average derivative ($h(x) = x$); the absolute average derivative ($h(x) = |x|$); the square average derivative ($h(x) = x^2$); the maximum and the minimum derivative ($h(x) = max(x)$ and ($h(x) = min(x)$)). Each of these measures reflects different aspects of the model and, as a consequence, it can provide different ranks of the

variables, according to its magnitudes. The most natural sensitive measure is, of course, the average derivative but, because of cancellations between negative and positive values, the absolute average derivative and the square average derivative are the most used. In many financial applications, such as the construction of the risk neutral portfolios of assets, and in all applicative context where the interest is on inflection points, the maximum and the minimum derivative are also quite common. However, the most general and natural choice in the statistical literature is the square function leading to the following relevance measure for the d input neurons:

$$\theta_j = \mathbb{E}\left[f_j^2\left(\mathbf{x}, \mathbf{w}_0\right)\right], \qquad j = 1, 2, \ldots, d.$$

Therefore, the hypothesis that a given set of variables has no effect on Y can be formulated in a multiple testing framework as

$$H_j : \theta_j = 0 \quad vs \quad H_j' : \theta_j > 0, \quad j = 1, 2, \ldots, d. \tag{1.8}$$

Each null H_j can be tested by using the statistic,

$$\widehat{T}_{n,j} = n\hat{\theta}_j = \sum_{i=1}^{n} f_j^2\left(\mathbf{X}_i, \widehat{\mathbf{w}}_n\right) \tag{1.9}$$

where $\hat{\theta}_j = \frac{1}{n}\sum_{i=1}^{n} f_j^2\left(\mathbf{X}_i, \widehat{\mathbf{w}}_n\right)$ and the vector $\widehat{\mathbf{w}}_n$ is a consistent estimator of the unknown parameter vector $\mathbf{w}_0$. Clearly, large values of the test statistics indicate evidence against H_j.

So, the problem here is how to decide which hypotheses to reject, accounting for the multitude of tests. Significant correction for multiple hypothesis testing has been a central concern in many fields of research that deal with large sets of variables and small samples and where, as a consequence, the control of false positives becomes an important problem.

The most relevant methods for significant adjustment are based on standard measures like the familywise error rate (FWE), defined as the probability of finding at least one false positive, that is, rejecting at least one of the true null hypotheses.

The FWE can be controlled by using the well-known Bonferroni method or stepwise procedures proposed by Holm (1979) 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.

Here, we use the StepM procedure proposed by Romano and Wolf (2005), suitable for joint comparison of multiple misspecified models.

The step-down procedure begins by constructing a rectangular joint confidence region (with nominal coverage probability $1 - \alpha$), which is used to test the joint null hypothesis that all the nulls $H_j, j = 1, 2, \ldots, d$ are true. If all hypotheses are not rejected, the procedure stops. Otherwise, the rejected hypotheses are removed and a new rectangular joint confidence region with nominal level $1 - \alpha$ is constructed. The process is repeated until no further hypotheses are rejected.

The procedure can be described by Algorithm 1.1.

Algorithm 1.1: Multiple testing algorithm.

Relabel the hypothesis from H_{r_1} to H_{r_d} in redescending order of the value of the test statistics $\widehat{T}_{n,j}$, that is $\widehat{T}_{n,r_1} \geq \widehat{T}_{n,r_2} \geq \ldots \geq \widehat{T}_{n,r_d}$.
Set $L = 1$ and $R_0 = 0$.
for $j = R_{L-1} + 1$ to d **do**
  Fix $c_L(1-\alpha)$ such that the joint asymptotic coverage probability is $1-\alpha$
  if $0 \notin \left[\widehat{T}_{n,r_j} - c_L(1-\alpha), \infty\right)$ **then**
    reject H_{r_j}
  end if
end for
if no (further) null hypothesis are rejected **then**
  Stop
else
  R_L = number of rejected hypothesis
  $L = L + 1$
  Go to step 3
end if

Of course, in order to apply the StepM procedure, it is necessary to know the distribution of the test statistic $\widehat{T}_{n,j}$. Under general conditions, (Giordano, La Rocca, and Perna 2014) it is straightforward to show that:

$$\widehat{T}_{n,j} \xrightarrow{d} F(0, \mathbf{C}^*, \mathbf{M}^*) \quad \forall j \in \{1, \ldots, d\}$$

where F denotes the mixture of independent χ^2 random variables and

$$\mathbf{M}^* = \mathbb{E}\left(\left[\bigtriangledown f_j(\mathbf{X}_i, \mathbf{w}_0) \bigtriangledown f_j(\mathbf{X}_i, \mathbf{w}_0)^T + \bigtriangledown^2 f_j(\mathbf{X}_i, \mathbf{w}_0) f_j(\mathbf{X}_i, \mathbf{w}_0)\right]\right).$$

Even if this result is relevant from a theoretical point of view, it does not allow easy the estimation of the quantiles $c_L(1-\alpha)$ in StepM algorithm. Therefore, the sampling distribution can be better approximated by some types of resampling techniques. Here, we propose the use of subsampling. This choice can be justified by its property of being robust against misspecifications, a key property when dealing with artificial neural network models. Moreover, the procedure delivers consistent results under very weak assumptions.

The resampling scheme runs as follows. Fix b such that $b < n$ and let $\mathbf{Y}_1, \ldots, \mathbf{Y}_S$ be equal to $S = \binom{n}{b}$ subsets of $\{\mathbf{Z}_1, \ldots, \mathbf{Z}_n\}$. Let $\widehat{T}^s_{b,j}$ be the test statistic evaluated at $\mathbf{Y}_s, s = 1, \ldots, S$. Then, for $\mathbf{x} \in \mathbb{R}^d$, the true joint cdf of the test statistics evaluated at $\mathbf{x}$ is given by

$$G_n(\mathbf{x}) = \Pr\left\{\widehat{T}_{n,1} \leq x_1, \widehat{T}_{n,2} \leq x_2 \ldots, \widehat{T}_{n,d} \leq x_d\right\} \tag{1.10}$$

and it can be estimated by the subsampling approximation

$$\widehat{G}_n(\mathbf{x}) = \binom{n}{b}^{-1} \sum_{s=1}^{S} \mathbb{I}\left\{ \widehat{T}^s_{b,1} \leq x_1, \widehat{T}^s_{b,2} \leq x_2, \ldots, \widehat{T}^s_{b,d} \leq x_d \right\} \tag{1.11}$$

where as usual $\mathbb{I}(\cdot)$ denotes the indicator function.

As a consequence, for $D \subset \{1, \ldots, d\}$, the distribution of the maximum of the test statistics, let's say $H_{n,D}(x)$, can be estimated by the empirical distribution function $\widehat{H}_{n,D}(x)$ of the values $\max\left\{ \widehat{T}^s_{b,j}, j \in D \right\}$, that is

$$\widehat{H}_{n,D}(x) = \binom{n}{b}^{-1} \sum_{s=1}^{S} \mathbb{I}\left\{ \max\left\{ \widehat{T}^s_{b,j}, j \in D \right\} \leq x \right\} \tag{1.12}$$

and the quantile of order $1-\alpha$ can be estimated as

$$\widehat{c}_L(1-\alpha) = \inf\left\{ x : \widehat{H}_{n,D}(x) \geq 1-\alpha \right\}. \tag{1.13}$$

The consistency of the subsampling procedure has been proved in (Giordano, La Rocca, and Perna 2014) as a straightforward extension of a result in Romano and Wolf (2005). In particular, under general assumptions, if $b/n \to 0$ when $b \to \infty$ and $n \to \infty$, then

$$\rho\left(\widehat{G}_{n,D}, G_{n,D}\right) \xrightarrow{P} 0$$

for any metric ρ metrizing weak convergence on $\mathbb{R}^{|D|}$ with $|D|$ the cardinality of D. Moreover, the subsampling critical values satisfy

$$\widehat{c}_L(1-\alpha) \xrightarrow{P} c_L(1-\alpha)$$

and

$$\limsup_n FWE \leq \alpha$$

using Algorithm (1.1) with the subsample estimator, $\widehat{c}_L(1-\alpha)$.

The choice of the subsampling as resampling technique can be justified as follows. First, the method does not require any knowledge of the specific structure of the data and so it is robust against misspecifications, a key property when dealing with artificial neural network models. Moreover, the procedure delivers consistent results under very weak assumptions. In our case, by assuming: (i) $b \to \infty$ in such a way that $\frac{b}{n} \to 0$, as $n \to \infty$, (ii) conditions that guarantee asymptotic normality of $\widehat{\mathbf{w}}_n$ are fulfilled (White, 1989), (iii) smoothness conditions on the test statistics $\widehat{T}_{n,j}$ (White and Racine, 2001), the subsampling approximation is a consistent estimate of the unknown (multivariate) sampling distribution of the test statistics (Romano and Wolf, 2005). Observe that, the number of subsets of length b which can be formed out of a sample of size n grows very fast with n. Therefore, usually, just B random selected subsets are considered for computing the subsampling approximation.

Clearly, the main issue when applying the subsampling procedure lies in choosing the length of the block, a problem which is common to all blockwise resamplig

techniques. Nevertheless, Politis, Romano, and Wolf (1999) proposed a number of strategies to select b and theorems that ensure that the asymptotic results are still valid for a broad range of choices for the subsample size. More recently, Giacomini, Politis, and White (2013) proposed an approach to reduce the computational effort when conducting Monte Carlo experiments involving resampling techniques. It could be used in the neural network framework to make feasible the block selection calibration algorithm.

1.3.2 Some Numerical Examples

To illustrate the performance of the proposed input selection procedure, we use simulated data sets generated by models with known structure. The aim is to evaluate the ability of the test procedure to select a proper set of explanatory variables for the given data generating process. For the experimental setup, we assume $n = 300$, $b = 100$, $r = 2$, $B = 1000$, $\alpha = 0.05$. The hidden layer size of the neural networks has been determined by using the cross-validation (CV) and all the neural network models have been estimated by using a square loss function in Eq. (1.3), repeating the estimation process with different randomly chosen starting points to avoid being trapped in local minima. The software procedures have been implemented in R.

The simulated data sets have been generated by the following models.

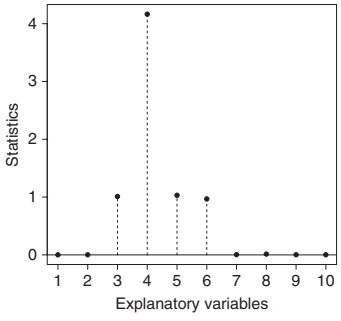
The first model (Model M1) is the same model used in Tibshirani (1996). We assume that Y depends on 10 esplicative variables $\{X_1, X_2, \ldots, X_{10}\}$ but just variables $\{X_3, X_4, X_5, X_6\}$ are relevant to the model, that is,

$$Y = 3\psi\left(2X_3 + 4X_4 + 3X_5 + 3X_6\right) + 3\psi\left(2X_3 + 4X_4 - 3X_5 - 3X_6\right) + \varepsilon$$

where ψ is the logistic activation function, $\mathbf{X} = (X_3, X_4, X_5, X_6)^T$ is a vector of multivariate Gaussian random variables with zero mean, unit variance and pair-wise correlation equal to 0.5 and ε Gaussian with zero mean and variance equal to 0.7. This gave 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.

The results of the multiple-testing procedure for variable selection are reported in Figure 1.1. After the first step, the procedure rejects the hypothesis that variable 4 is not relevant and accepts all others hypotheses. At the second step, variables 5, 3, and 6 are recognized as relevant, as well. At the third step, the remaining variables are recognized as not relevant and the procedure stops. The procedure gives results that are consistent with the data-generating process and the plot reported in Figure 1.1.

The second model (Model M2) is the same model used in De Veaux *et al.* (1998). Again, we assume that Y depends on 10 esplicative variables $\{X_1, X_2, \ldots, X_{10}\}$ but



j	$\hat{T}_{n,r_j}$	r_n	$\hat{T}_{n,r_j} - \hat{c}_1$	$\hat{T}_{n,r_j} - \hat{c}_2$	$\hat{T}_{n,r_j} - \hat{c}_3$
1	4.1649	4	**2.8040**	–	–
2	1.0315	5	–0.3295	**0.5303**	–
3	1.0105	3	–0.3505	**0.5092**	–
4	0.9680	6	–0.3930	**0.4667**	–
5	0.0142	8	–1.3468	–0.4871	–0.1836
6	0.0038	7	–1.3571	–0.4975	–0.1940
7	0.0025	9	–1.3585	–0.4988	–0.1952
8	0.0019	10	–1.3590	–0.4993	–0.1958
9	0.0016	2	–1.3594	–0.4997	–0.1962
10	0.0010	1	–1.3599	–0.5002	–0.1967

Figure 1.1 Model M1. Results of the multiple testing procedure ($n = 300$, $b = 100$, $r = 2$, $B = 1000$, $\alpha = 0.05$). Figures in bold refer to the rejection of the corresponding hypotheses H_{r_j}.

just variables $\{X_4, X_5, X_6\}$ are relevant to the model, that is,

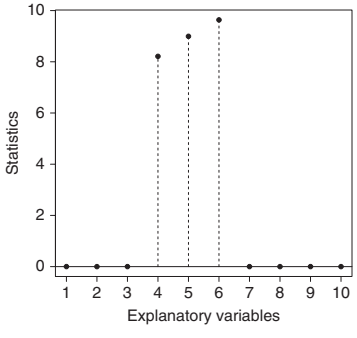
$$Y = 1.5 \cos\left(\frac{2\pi}{\sqrt{3}}\sqrt{\left(X_4 - 0.5\right)^2 + \left(X_5 - 0.5\right)^2 + \left(X_6 - 0.5\right)^2}\right) + \varepsilon$$

where $\mathbf{X} = (X_4, X_5, X_6)^T$ 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. In this latter case, the procedure is able to select the correct set of relevant variables in two steps, as clearly shown in Figure 1.2.

For the third model (Model M3) introduced by Friedman (1991), again, we assume that Y depends on 10 esplicative variables $\{X_1, X_2, \ldots, X_{10}\}$ but just variables $\{X_3, X_4, X_5, X_6, X_7\}$ are relevant, that is

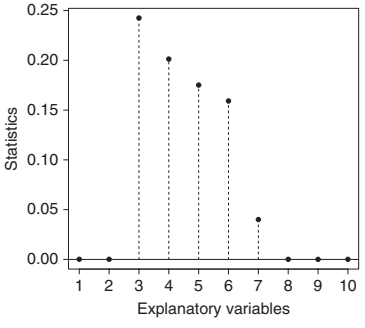
$$Y = \left(10 \sin\left(\pi X_3 X_4\right) + 20\left(X_5 - 0.5\right)^2 + 10X_6 + 5X_7 + \varepsilon\right)/25$$

where $\mathbf{X} = (X_3, X_4, X_5, X_6, X_7)^T$ is drawn randomly from the unit hypercube.



j	$\hat{T}_{n,r_j}$	r_j	$\hat{T}_{n,r_j} - \hat{c}_1$	$\hat{T}_{n,r_j} - \hat{c}_2$
1	10.0015	6	**8.2969**	–
2	9.0820	5	**7.3773**	–
3	8.2477	4	**6.5431**	–
4	0.0020	2	–1.7027	–0.0159
5	0.0010	1	–1.7037	–0.0170
6	0.0007	7	–1.7040	–0.0173
7	0.0006	8	–1.7041	–0.0173
8	0.0003	3	–1.7043	–0.0176
9	0.0003	9	–1.7044	–0.0176
10	0.0002	10	–1.7045	–0.0177

Figure 1.2 Model M2. Results of the multiple-testing procedure ($n = 300$, $b = 100$, $r = 2$, $B = 1000$, $\alpha = 0.05$). Figures in bold refer to the rejection of the corresponding hypotheses H_{r_j}.



j	$\hat{T}_{n,r_j}$	r_j	$\hat{T}_{n,r_j} - \hat{c}_1$	$\hat{T}_{n,r_j} - \hat{c}_2$	$\hat{T}_{n,r_j} - \hat{c}_3$
1	0.2422	3	**0.1951**	–	–
2	0.2019	4	**0.1548**	–	–
3	0.1750	5	**0.1280**	–	–
4	0.1591	6	**0.1120**	–	–
5	0.0400	7	–0.0070	**0.0354**	–
6	0.0002	1	–0.0470	–0.0045	–0.0020
7	0.0001	2	–0.0470	–0.0045	–0.0020
8	0.0001	8	–0.0470	–0.0045	–0.0020
9	0.00009	10	–0.0470	–0.0045	–0.0020
10	0.00006	9	–0.0470	–0.0045	–0.0020

Figure 1.3 Model M3. Results of the multiple-testing procedure ($n = 300$, $b = 100$, $r = 2$, $B = 1000$, $\alpha = 0.05$). Figures in bold refer to the rejection of the corresponding hypotheses H_{r_j}.

Again, the procedure is able to correctly identify the set of relevant variables in three steps, as clearly shown in Figure 1.3.

The results of a more detailed simulation experiment are reported in Giordano, La Rocca, and Perna (2014) in which we analyse the sensitivity of the CV procedure to omitted or redundant variables and the sensitivity of the StepM testing scheme to hidden layer size error identification. The results show that redundant variables do not appear to be harmful in selecting the hidden layer size; in almost all cases, the true hidden layer size is correctly identified. On the contrary, omitting relevant variables might have negative effects on the hidden layer size. However, this appears to be connected to the number and type of omitted variables. In any case, increasing the sample size may improve the performance. Moreover, the experimental results also suggest the possibility to use the block length as a calibration tool to bring the empirical FWE closer to the nominal FWE. This calibration technique could even be effective if the hidden layer size is not correctly identified but it has been fixed in a neighbourhood of the true value. When the hidden layer size is correctly identified, the procedure correctly identifies the true relevant variables in all cases, for all sample sizes. When the hidden layer size is incorrectly identified (under/over estimation), the results depend on the sample size and on the subsample size. For the case of $n = 400$ and $n = 600$, the proportion is very close to 1. There are some identification problems for the case $n = 200$ for some variables. However, again, the block length of the subsampling can be used to mitigate the problem and to increase the proportion of true relevant variables correctly identified.

1.3.3 Application to Real Data

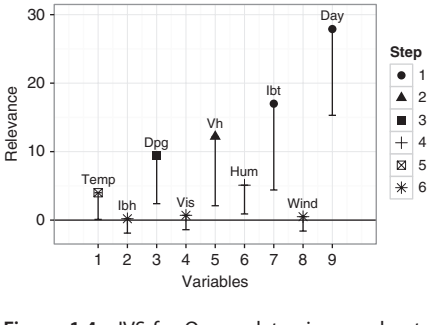
As an application to real data, we considered a very popular data set, often used to check the performance of non-parametric regression techniques with respect to variable selection. The data are daily measurement of ozone concentration

(maximum one hour average) and eight meteorological quantities for 330 observations starting from the beginning of 1976. The data were used by Breiman and Friedman (1985) when introducing the ACE algorithm. The variables considered are: Ozone (Upland ozone concentration, ppm), Temp (Sandburg Air Force Base temperature, °F), Ibh (inversion base height, feet), Dpg (Dagget pressure gradient, mmHg), Vis (visibility, in miles), Vh (Vandenburg 500 millibar height, m), Hum (humidity, percent), Wind (wind speed, mph), Ibt (Inversion base temperature, degrees F), and Day (day of year).

The hidden layer size has been selected by CV, while the input relevant variables have been selected by using the proposed procedure. The StepM procedure has been calibrated with the subsampling, where the subsample size has been fixed by using the minimum volatility method (Politis, Romano, and Wolf , 1999).

The procedure clearly selects the variables Day, Ibt, Vh, Dpg, Hum, and Temp as relevant, while the variables Vis, Wind, and Ibh are classified as not relevant. Note that the variables are selected in six steps (just the variables Day and Ibt are selected in the first step) and so, a multistep procedure appears to be necessary to avoid masking effects (Figure 1.4).

For the sake of comparison with other neural network variable selection schemes, we considered the Bayesian approach proposed in Lee (2004), where it is also reported a comparison with other non-parametric variable selection techniques: The stepwise ACE, the stepwise GAM, adaptive regression splines (TURBO), and adaptive backfitting (BRUTO). All the results are summarized in Table 1.1. The proposed procedure largely agrees with the best network selected by using the Bayesian approach proposed by Lee. Interestingly enough, however, the variable Temp is considered relevant by the multiple testing scheme, while it is never selected by the Bayesian approach when applied to neural networks. Note that this variable is always selected by the other non-parametric techniques, suggesting that the multiple testing scheme is able to uncover possibly masked relationships. Even if the alternative methods disagree about which variable subset is optimal, it does seem clear that some variable selection procedure is necessary.



Rank	Variable	$\hat{T}_{n,j}$	C_j	Limit	Step
1	Day	27.9	12.6	15.3	1
2	Ibt	17.0	12.6	4.4	1
3	Vh	12.2	10.1	2.1	2
4	Dpg	9.4	7.0	2.4	3
5	Hum	5.1	4.2	0.9	4
6	Temp	4.0	3.9	0.1	5
7	Vis	0.7	2.1	−1.4	6
8	Wind	0.5	2.1	−1.6	6
9	Ibh	0.2	2.1	−1.9	6

Figure 1.4 IVS for Ozone data via neural networks. The relevance measure is the statistic $\hat{T}_{n,j}$. The hidden layer size has been selected by k-fold CV ($r = 3$). Subsample size selected by using minimum volatility method. The nominal size is $\alpha = 0.05$.

Table 1.1 Comparison of variable selection procedures on the Ozone data.

Method	Vh	Wind	Hum	Temp	Ibh	Dpg	Ibt	Vis	Day
Multiple test (NN(3))	×		×	×		×	×		×
(Ranks)	(3)		(5)	(6)		(4)	(2)		(1)
Lee's Bayes (Best NN(3))	×		×			×	×		×
Lee's Bayes (Second best NN(6))			×			×	×	×	×
Lee's Bayes (Third best NN(3))			×				×		×
ACE (Stepwise)				×	×	×		×	
GAM (Stepwise)	×	×	×	×	×	×		×	×
TURBO	×			×	×	×		×	×
BRUTO				×	×	×		×	×

1.4 The Selection of the Hidden Layer Size

The hidden layer size should be selected by looking at the predictive performance of the neural network model, as in the case of CV. However, 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, 2006, 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.

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

$$Y_\tau = g\left(\mathbf{X}_\tau\right) + \varepsilon_\tau \tag{1.14}$$

Assume then that $k+1$ alternative forecasting neural network models are available, namely $f\left(\mathbf{x}^j, \mathbf{w}^j\right), j = 0, 1, \ldots, k$. The models can differ in hidden layer size and/or in number and type of explanatory variables. Model $f\left(\mathbf{x}^0, \mathbf{w}^0\right)$ 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 $u_{j,\tau} = Y_\tau - f\left(\mathbf{X}_\tau^j, \mathbf{w}_0^j\right), j = 0, 1, \ldots, k$ where $\mathbf{w}_0$ is defined as in Section 1.2. Let h be a proper chosen loss function (Elliot and Timmermann, 2004) and define

$$\theta_j = \mathbb{E}\left(h\left(u_{0,\tau}\right) - h\left(u_{j,\tau}\right)\right), j = 1, 2, \ldots, k. \tag{1.15}$$

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

$$H_j : \theta_j \leqslant 0 \quad vs \quad H_j' : \theta_j > 0 \tag{1.16}$$

and, in a multiple testing framework, make a decision concerning each individual testing problem by either rejecting H_j or not. Also in this case, the data snooping problem can arise and the FWE should be taken under control. In this framework, possible alternative solutions can be obtained by using the reality check and by using test of superior predictive ability, which can be easily extended to our neural network framework.

1.4.1 A Reality Check Approach

To avoid data snooping problems, it is possible to use the reality check as in White (2000) and the modification for nested models as proposed in Clark and McCracken (2012a,b).

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

$$H_0 : \max_{j=1,\dots,k} \theta_j \leqslant 0 \quad vs \quad H_1 : \max_{j=1,\dots,k} \theta_j > 0. \tag{1.17}$$

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

$$S_P = \max_{j=1,\dots,k} S_P\left(0, j\right) \tag{1.18}$$

where

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

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

$$\max_{j=1,\dots,k} \left\{S_P\left(0, j\right) - \sqrt{P}\theta_j\right\} \xrightarrow{d} \max_{j=1,\dots,k} S\left(0, j\right). \tag{1.19}$$

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

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

where the generic element of vector $\mathbf{v}_i$ is defined as $v_{i,j} = h\left(u_{0,i}\right) - h\left(u_{j,i}\right)$. 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

$$S_P^* = \max_{j=1,\ldots,k} S_P^*(0,j) \tag{1.20}$$

where

$$S_P^*(0,j) = \frac{1}{\sqrt{P}} \sum_{i=R+1}^{N} \left\{ \left(h\left(\hat{u}_{0,i}^*\right) - h\left(\hat{u}_{0,i}\right) \right) - \left(h\left(\hat{u}_{j,i}^*\right) - h\left(\hat{u}_{j,i}\right) \right) \right\} \tag{1.21}$$

with $\hat{u}_{j,i}^* = Y_i^* - f\left(\mathbf{X}_i^{*j}, \hat{\mathbf{w}}_R^j\right)$ and $\left(Y_i^*, \mathbf{X}_i^{*j}\right)$ denote the resampled data. Note that the bootstrap statistics contain only estimators based on the original sample and this is particularly convenient when dealing with neural network models. If an estimation is needed for each bootstrap sample, the procedure will soon become 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 $P, R \to \infty$

$$\Pr\left(\sup_{\nu \in \mathbb{R}} \left| \Pr_*\left(S_P^* \leqslant \nu\right) - \Pr\left(S_P^\kappa \leqslant \nu\right) \right| > \varepsilon \right) \to 0 \tag{1.22}$$

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

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

As usual, the bootstrap procedure can be implemented by Monte Carlo. For any bootstrap replication, compute the bootstrap statistics, 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 1.2.

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

1.4.2 Numerical Examples by Using the Reality Check

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:

$$Y = \mathbf{X1} + \varepsilon$$

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 the ones of appropriate length. This

Algorithm 1.2: Bootstrap resampling algorithm.

Fix P, R such that $P + R = N$.
Fix B, the number of bootstrap replicates.
Generate B sets of random observation indexes of length P, namely $\left\{\theta_b(i), i = R+1, \dots, N; b = 1, \dots, B\right\}$.
$M_0 \leftarrow -\Delta$, with Δ finite big constant.
$M_0^{(b)} \leftarrow -\Delta$, with Δ finite big constant, $b = 1, \dots, B$.
$\hat{\mathbf{w}}_R^0 \leftarrow \arg\min_{\mathbf{w}\in\mathbf{W}} \frac{1}{R} \sum_{i=1}^{R} q\left(Y_i, f\left(\mathbf{X}_i^0, \mathbf{w}\right)\right)$.
$\hat{u}_{0,i} \leftarrow Y_i - f\left(\mathbf{X}_i^0, \hat{\mathbf{w}}_R^0\right), i = R+1, \dots, N$.
$\hat{h}_{0,i} \leftarrow h\left(\hat{u}_{0,i}\right), i = R+1, \dots, N$.
for $j = 1$ to k **do**
  $\hat{\mathbf{w}}_R^j \leftarrow \arg\min_{\mathbf{w}\in\mathbf{W}} \frac{1}{R} \sum_{i=1}^{R} q\left(Y_i, f\left(\mathbf{X}_i^j, \mathbf{w}\right)\right)$.
  $\hat{u}_{j,i} \leftarrow Y_i - f\left(\mathbf{X}_i^j, \hat{\mathbf{w}}_R^j\right), i = R+1, \dots, N$.
  $\hat{h}_{j,i} \leftarrow h\left(\hat{u}_{j,i}\right), i = R+1, \dots, N$
  $S_P\left(0, j\right) \leftarrow \frac{1}{\sqrt{P}} \sum_{i=R+1}^{N} \left\{\hat{h}_{0,i} - \hat{h}_{j,i}\right\}$.
  $M_j \leftarrow \max\left\{S_p\left(0, j\right), M_{j-1}\right\}$.
  for $b = 1$ to B **do**
    $\hat{u}_{0,i}^{(b)} = Y_{\theta(b)} - f\left(\mathbf{X}_{\theta(b)}^0, \hat{\mathbf{w}}_R^0\right)$
    $\hat{u}_{i,j}^{(b)} = Y_{\theta(b)} - f\left(\mathbf{X}_{\theta(b)}^j, \hat{\mathbf{w}}_R^j\right)$.
    $S_P^{(b)}\left(0, j\right) \leftarrow \frac{1}{\sqrt{P}} \sum_{i=R+1}^{N} \left\{\left(h\left(\hat{u}_{0,i}^{(b)}\right) - \hat{h}_{0,i}\right) - \left(h\left(\hat{u}_{j,i}^{(b)}\right) - \hat{h}_{j,i}\right)\right\}$
    $M_j^{(b)} \leftarrow \max\left\{S_P^{(b)}\left(0, j\right), M_{j-1}^{(b)}\right\}$.
  end for
end for
return p-value $\leftarrow \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}\left(M_k^{(b)} > M_k\right)$.

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 Model M3 is the same model used in De Veaux *et al.* (1998). Both models have already been used in previous sections.

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

In Table 1.2, 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 >0.50, and so the benchmark (i.e., the linear model) shows better expected predictive performance with

Table 1.2 Values of the test statistics for different input neuron sets and different hidden layer size.

Model	Inputs/size	1	2	3	4	5	6
M1	1	*−0.116*	*−0.116*	*−0.116*	*−0.116*	*−0.116*	*−0.116*
	2	*−0.290*	*−0.290*	*−0.290*	*−0.290*	*−0.290*	*−0.290*
	3	*−0.721*	*−0.721*	*−0.721*	*−0.721*	*−0.721*	*−0.721*
	4	*−0.986*	*−0.986*	*−0.986*	*−0.986*	*−0.986*	*−0.986*
	5	*−0.844*	*−0.844*	*−0.844*	*-0.844*	*−0.844*	*−0.844*
	6	*−0.873*	*−0.873*	*−0.873*	*−0.873*	*−0.873*	*−0.873*
M2	1	*−0.477*	*−0.477*	*−0.477*	*−0.477*	*−0.477*	*−0.477*
	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**

The benchmark model is a neural network with skip layer and zero hidden neurons. Values in italics correspond to p-values >0.50. Values in bold correspond to p-values <0.005.

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*.

In order to get a deeper insight, in Figures 1.5–1.7, 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.

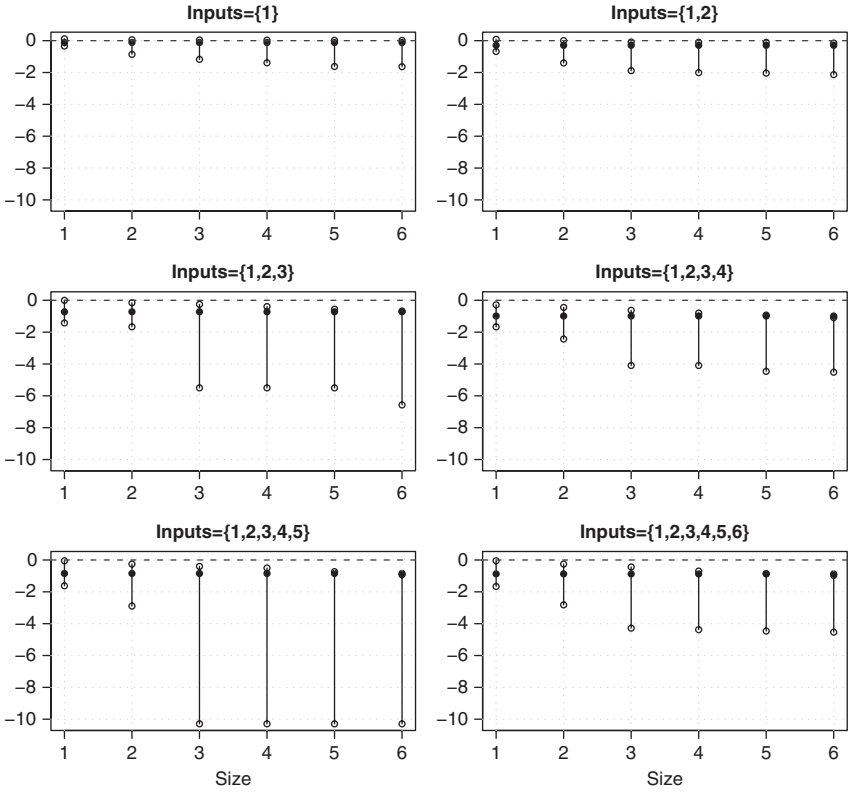


Figure 1.5 Model M1. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark.

1.4.3 Testing Superior Predictive Ability for Neural Network Modeling

Testing the superior predictive ability approach can be easily extended to our neural network framework (La Rocca and Perna, 2014). Let $\mathcal{S} = \left\{\left(Y_i, \mathbf{X}_i\right), i \in S\right\}$ and $\mathcal{P} = \left\{\left(Y_i, \mathbf{X}_i\right), i \in P\right\}$ 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 $\hat{u}_{j,\tau} = Y_\tau - f_j\left(\mathbf{X}_\tau, \hat{\mathbf{w}}\right), j = 0, 1, \ldots, k$ and let $\text{MPE}_j = \sum_{\tau \in P} h\left(\hat{u}_{j,\tau}\right)$, where P is the cardinality of the set $\mathcal{P}$.

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

$$Fp_j = P\frac{\text{MPE}_0 - \text{MPE}_j}{\text{MPE}_j}, \qquad j = 1, 2, \ldots, k. \tag{1.23}$$

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 (Eq. 1.16) keeping under control the family wise error rate, runs as follows. Relabel the hypothesis from

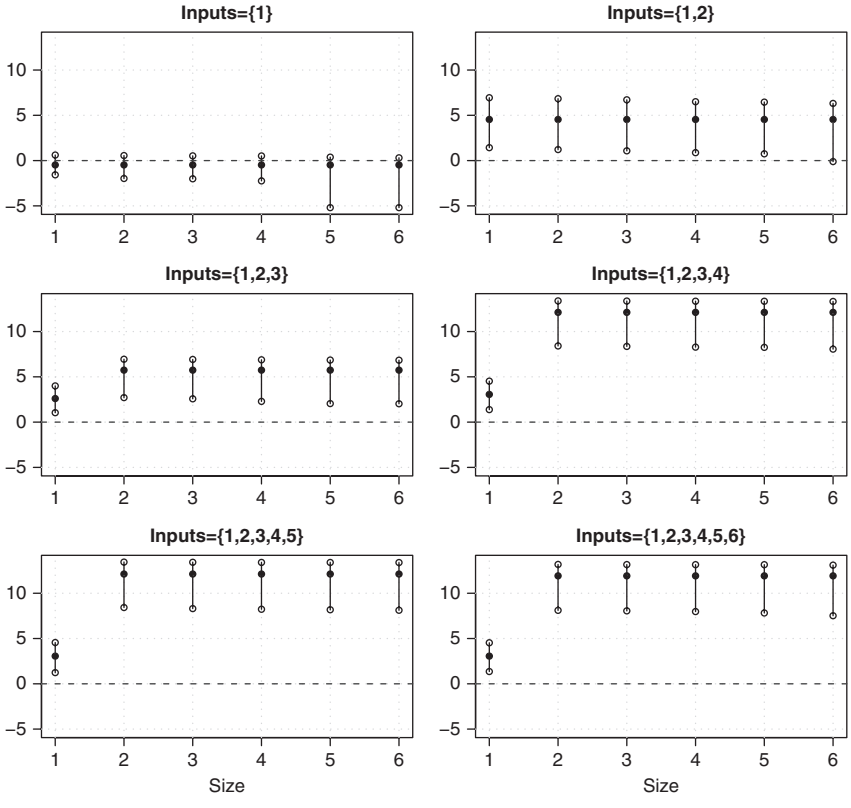


Figure 1.6 Model M2. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark.

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 $\left[Fp_{r_1} - c_{1-\alpha}, \infty\right) \times \cdots \times \left[Fp_{r_k} - c_{1-\alpha}, \infty\right)$ 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 the smallest hidden layer size should be selected. If all the nulls are not rejected in the first step,

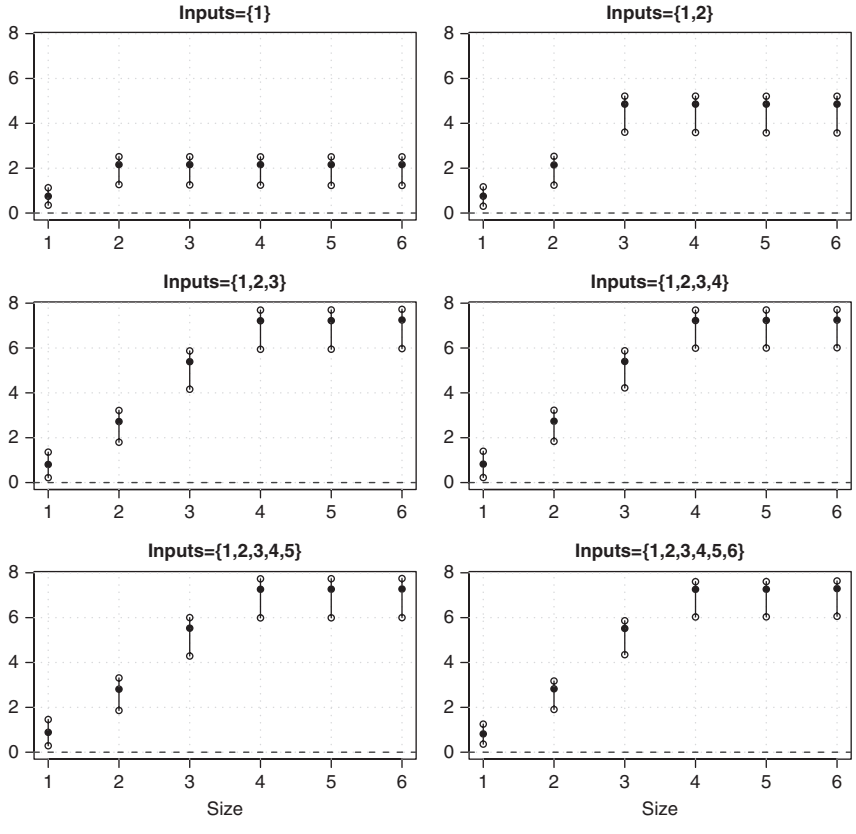


Figure 1.7 Model M3. Bootstrap confidence intervals for the maximum expected predictive performance of the neural networks with respect to the benchmark.

there is no neural network model, which is able to outperform the linear model (assumed as a benchmark) in terms of predictive ability. Again, the quantile of order $c_{1-\alpha}$ can be estimated by using resampling techniques.

The pseudo-code for the complete testing procedure is described in Algorithm 1.3.

1.4.4 Some Numerical Results Using Test of Superior Predictive Ability

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.

Again, to generate sintetic data sets, we have used the same models used in De Veaux *et al*. (1998), Friedman (1991), and Tibshirani (1996). We also added as an

Algorithm 1.3: 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 $Fp_{r_1} \geq Fp_{r_2} \geq \cdots \geq Fp_{r_k}$.
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** $0 \notin \left[Fp_{r_s} - \hat{c}_{1-\alpha}, \infty \right)$ **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)

additional model the one used by Turlach (2004) and defined as

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

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, as in Friedmans's case, includes both linear and nonlinear relationships and it is known to be problematic for other variable selection schemes.

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 non linear 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.

The results of the testing procedure for typical realizations are reported in Figure 1.8. 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

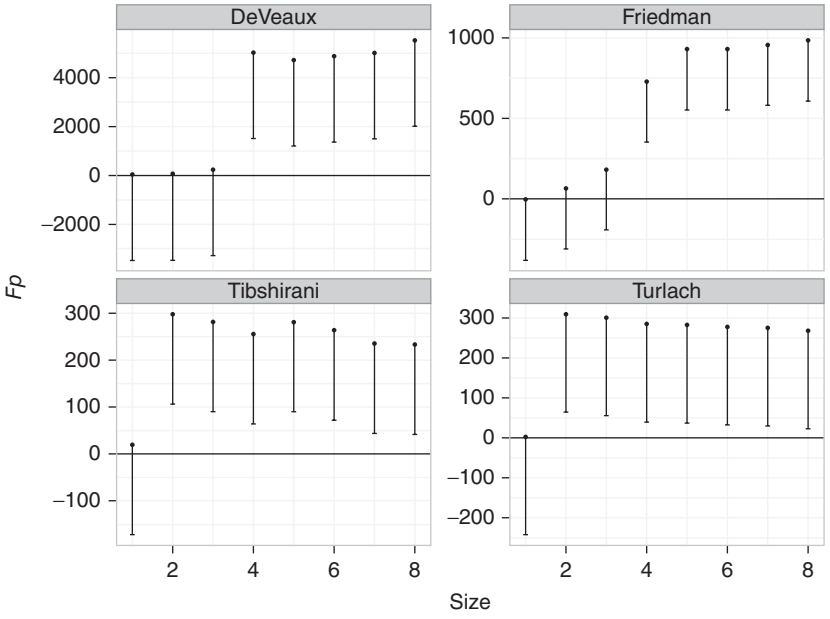


Figure 1.8 Joint confidence regions with nominal coverage probability $1 - \alpha = 0.95$.

runs with three different sample sizes $N = 300, 400, 600$ using the last 30% observations for prediction. The results are reported in Figure 1.9. 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.

1.4.5 An Application to Real Data

To validate the performance of the proposed procedure, two applications to real data are discussed in La Rocca and Perna (2014).

As a first example, we use the Prostate Cancer data set which comes from a study by Stamey *et al.* (1989). 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 has 97 observations and it is split 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 the sake of comparison, as identification tools for the number of hidden neurons, we also use the k-fold CV selection rule and the Bayesian information criterion, proved

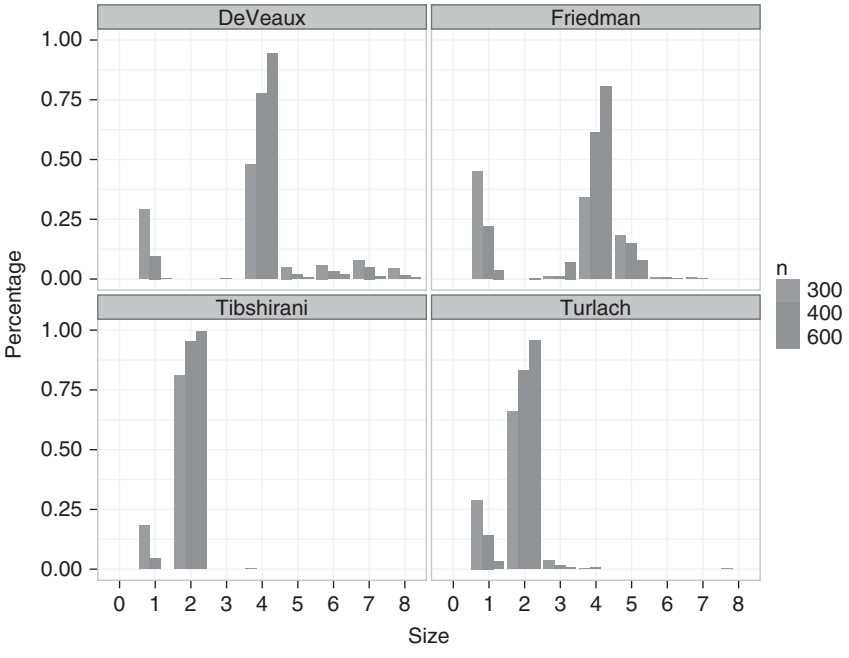


Figure 1.9 Proportion of hidden layer size identification by using the testing procedure for superior predictive ability. (See inset for color representation of this figure.)

to be consistent (almost surely) in the case of multi-layer perceptrons with one hidden layer in White (1990).

Clearly, the BIC identifies a linear model, which is equivalent to a neural network with skip layer and zero hidden neurons, for all the weight decay values considered (Figure 1.10a,b). The latter result is confirmed by looking at the CV and our test of superior predictive ability (Figure 1.11a). To validate these results,

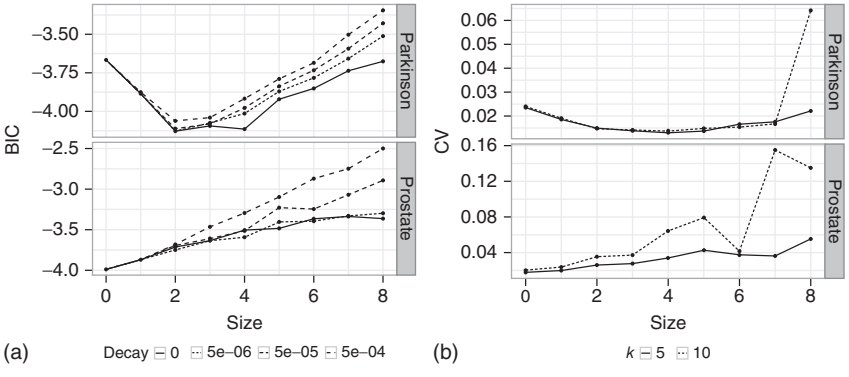


Figure 1.10 Bayesian information criterion values for different hidden layer sizes and different weight decay values (a). k-fold CV values for $k = 5$ and $k = 10$ using a weight decay equal to zero (b).

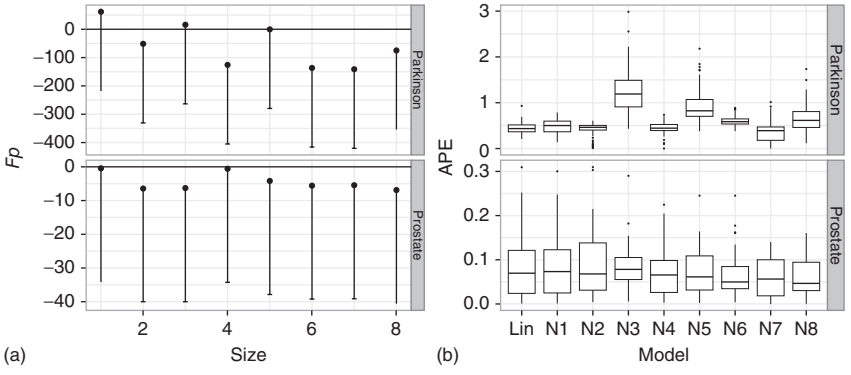


Figure 1.11 Joint confidence regions with nominal coverage probability $1 - \alpha = 0.95$ (a). Absolute prediction error distributions computed on the test set for linear models and neural networks with hidden layer size ranging from 1 to 8 (b).

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 1.11b. 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). These results are confirmed by a formal statistical comparison between the two distributions made by using the Brunner Munzel test and the Wilcoxon rank sum test which give p-values equal to 0.497 and 0.495, respectively.

The second data set is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a 6-month trial of a telemonitoring device for remote symptom progression monitoring (for details see Tsanas *et al*. (2010)). The data set has been downloaded from the UCI Machine Learning Repository and consists of 5875 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 five patients) has been considered. Again, the data set is split into two subsets: 731 observations (the first four individuals) have been used for the modeling step, while 156 observations (corresponding to the fifth patient) are used for validation purposes. In this case, the CV model selection rule identifies a neural network model with four hidden neurons, while the BIC identifies a neural network with two neurons. However, the proposed test shows that there is no superior predictive ability of neural network models with respect to linear models (Figure 1.11a). This is confirmed by the distribution of the absolute predictive errors reported in Figure 1.11b: the linear model and the neural networks with two or four 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 equal to 0.219 and 0.218, respectively. Clearly, neural networks with two or four hidden neurons and 17 explanatory variables appear to be heavily overparametrized, since they have 39 or 77 parameters, respectively. These

networks show no clear advantages in terms of predictive ability with respect to the linear model.

1.5 Concluding Remarks

In this chapter, a novel model selection procedure in neural network modeling has been presented and discussed. The basic idea of the proposed approach is that input neurons and hidden neurons play a different role in neural network modeling so that they should be selected by using different criteria. Specifically, the proposed approach identifies the number and the type of input variables by using a formal statistical test focusing the problem on a procedure for variable selection in regression models. On the contrary, the number of the hidden neurons is considered smoothing parameter and it is selected by looking at the predictive performance of the network model.

The proposed strategy addresses the problem of data snooping, which arises when a data set is used more than once for inference and model selection. Moreover, to overcome the analytical and probabilistic difficulties related to the estimation of the sampling distribution of the test statistics involved, the approach uses extensively resampling techniques.

The proposed test procedures have been tested on simulated and real data sets, which confirm their ability to detect correctly the set of input variables and to discriminate among alternative models. Clearly, joint usage of neural network models and resampling techniques are usually quite demanding from a computational point of view. In any case, it is worthwhile to underline that they are suitable to be implemented on parallel and cluster computers almost without any modification of the computing algorithms.

References

Barron, A.R. (1993) Universal approximation bounds for superposition of a sigmoidal function. *IEEE Transactions on Information Theory*, **39**, 930–945.

Breiman, L. and Friedman, J.H. (1985) Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, **80**, 580–619.

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

Clark, T.E. and McCracken, M.W. (2012b) Reality checks and comparison of nested predictive models. *Journal of Business and Economic Statistics*, **30**, 53–66.

Corradi, V. and Swanson, N.R. (2006) Predictive density evaluation, *Handbook of Economic Forecasting*, Vol. **1**, Elsevier.

De Veaux, R., Schumi, J., Schweinsberg, J., Shellington, D., and Ungar, L.H. (1998) Prediction intervals for neural networks via nonlinear regression. *Technometrics*, **40**, 277–282.

Elliot, G. and Timmermann, A. (2004) Optimal forecast combinations under general loss functions and forecast error distribution. *Journal of Econometrics*, **122**, 47–49.

Franses, P.H. and Draisma, G. (1997) Recognizing changing seasonal patterns using artificial neural networks. *Journal of Econometrics*, **81**, 273–280.

Friedman, J.H. (1991) Multivariate adaptive regression splines. *Annals of Statistics*, **19**, 1–67.

Giacomini, F.R., Politis, D.N., and White, H. (2013) A warp-speed method for conducting Monte Carlo experiments involving bootstrap estimators. *Econometric Theory*, **29**, 567–589.

Giordano, F., La Rocca, M., and Perna, C. (2014) Input variable selection in neural network models. *Communications in Statistics - Theory and Methods*, **43**, 735–750.

Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, **6**, 65–70.

Hornik, K., Stinchcombe, M., and Auer, P. (1994) Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Computation*, **6**, 1261–1275.

Hwang, J.T.G. and Ding, A.A. (1997) Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, **92**, 748–757.

Kuan, C.-M. and Liu, T. (1995) Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, **10**, 347–364.

La Rocca, M. and Perna, C. (2005a) Neural network modeling by subsampling, in *Computational Intelligence and Bioinspired Systems* (eds J. Cabestany, A. Prieto, and F. Sandoval), Springer.

La Rocca, M. and Perna, C. (2005b) Variable selection in neural network regression models with dependent data: a subsampling approach. *Computational Statistics and Data Analysis*, **48**, 415–429.

La Rocca, M. and Perna, C. (2014) Designing neural networks for modeling biological data: a statistical perspective. *Mathematical Biosciences and Engineering*, **11**, 331–342.

Lee, H.K.H. (2004) *Bayesian Nonparametrics via Neural Networks*, SIAM.

Ossen, A. and 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), pp. 315–322.

Politis, D.N., Romano, J.P., and Wolf, M. (1999). *Subsampling*, Springer.

Qi, M. and Zhang, G.P. (2001) An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research*, **132**, 666–680.

Romano, J.P. and Wolf, M. (2005) Stepwise multiple testing as formalized data snooping. *Econometrica*, **73**, 1237–1282.

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E., and Yang, N. (1989) Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate ii radical prostactomy treated patients. *Journal of Urology*, **16**, 1076–1083.

Swanson, N.R. and White, H. (1997) A model selection approach to real time macroeconomic forecasting using linear models and artificial neural networks. *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.

Tsanas, A., Little, M.A., McSharry, P.E., and Ramig, L.O. (2010) Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests. *IEEE Transactions on Biomedical Engineering*, **57**, 884–893.

Turlach, B. (2004) Discussion of least angle regression by Efron, Hastie, Jonstone and Tibshirani. *Annals of Statistics*, **32**, 494–499.

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

White, H. (1990) Connectionist nonparametric regression: multi-layer feedforward networks can learn arbitrary mappings. *Neural Networks*, **3**, 535–549.

White, H. (2000) A reality check for data snooping. *Econometrica*, **68**, 1097–1126.

White, H. and Racine, J. (2001) Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. *IEEE Transactions on Neural Networks*, **12**, 657–653.