Performing Feature Selection With Multilayer Perceptrons

Enrique Romero and Josep María Sopena

***Abstract*—An experimental study on two decision issues for wrapper feature selection (FS) with multilayer perceptrons and the sequential backward selection (SBS) procedure is presented. The decision issues studied are the stopping criterion and the network retraining before computing the saliency. Experimental results indicate that the increase in the computational cost associated with retraining the network with every feature temporarily removed before computing the saliency is rewarded with a significant performance improvement. Despite being quite intuitive, this idea has been hardly used in practice. A somehow nonintuitive conclusion can be drawn by looking at the stopping criterion, suggesting that forcing overtraining may be as useful as early stopping. A significant improvement in the overall results with respect to learning with the whole set of variables is observed.**

***Index Terms*—Experimental work, feature selection (FS), multilayer perceptrons, wrapper approach.**

I. Introduction

FEATURE SELECTION (FS) procedures control the bias/variance decomposition by means of the input dimension, establishing a clear connection with the curse of dimensionality [24]. It has been known for a long time that, in addition to reducing the storage requirements and the computational cost, FS leads to improve generalization results [19], [38]. Today, FS is still an active line of research [14], [15]. We will focus on FS with multilayer perceptrons (MLPs).

In the situations where MLPs are sensitive to overfitting, the assumption that FS helps to improve the generalization performance by a reduction of the dimensionality and by eliminating irrelevant variables is, according to the bias/variance decomposition, theoretically sound. However, as far as we know, the results obtained by MLPs using different methods of FS do not show this improvement. We carefully reviewed the most common approaches in the literature of FS with MLPs (we found more than 60 papers, briefly described in Section II). Many of them do not report the differences between the models obtained with the whole set of features and those obtained after FS is carried out. When these differences are reported, the results are, in general, similar (if not worse) to the ones obtained without performing FS. Significant improvements are obtained in very few cases.

The motivation of this work is to study the methodology carried out for FS with MLPs. One of the reasons that can explain why after performing FS with MLPs the results usually do not improve is that many saliencies (the saliency is the measure of importance of a feature) are computed on the basis of a network trained with the whole set of features. This implicitly assumes that networks trained with the whole set of features are good models, which is not true when irrelevant variables are present. An MLP with irrelevant variables is more flexible than the same model without them, leading to better approximations in the training set. However, an MLP that uses irrelevant variables will have poor generalization. This problem can be alleviated if there are enough data to filter irrelevant variables, but this is not usually the case. In addition, the number of features subsets with irrelevant variables is $2^{N_i} - 1$ times larger than that without them, where N_i is the number of irrelevant variables. Since irrelevant variables may interact in a nonlinear and complex way with the rest of variables, we cannot expect to obtain useful information for the detection of irrelevant variables from networks trained with them. A simple way to adequately detect irrelevant variables is to retrain the network without the candidate subset of irrelevant variables. In addition, other issues in the whole process of FS with MLPs may also have a great impact on the final results.

The main contribution of this paper is to study the influence of some important decision issues in the methodology carried out for FS with MLPs within the *wrapper* approach [17]. Specifically, an experimental study on two decision issues when performing FS with MLPs and the sequential backward selection (SBS) procedure is presented. The decision issues studied are the convenience of retraining the network before computing the saliency and the stopping criteria for the training phase. To the best of our knowledge, this is the first time that this assessment is carried out. For practical purposes, this is a very important issue.

Experimental results indicate that, for complex problems, the increase in the computational cost associated with retraining the network with every feature temporarily removed before computing its saliency is rewarded with a significant performance improvement. The network retraining turns out to be a critical issue, although it has been hardly used in practice (most models found in the literature do not retrain the network before computing the saliency). A somehow nonintuitive conclusion is drawn by looking at the stopping criterion, where forcing overtraining is shown to be potentially as useful as early stopping.

Manuscript received July 26, 2006; revised March 20, 2007 and July 18, 2007; accepted July 18, 2007. This work was supported by the Consejo Interministerial de Ciencia y Tecnología (CICYT) under Project TIN2006-08114.

E. Romero is with the Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: eromero@lsi.upc.edu).

J. M. Sopena is with the Laboratori de Neurocomputacó, Universitat de Barcelona, 08035 Barcelona, Spain.

Digital Object Identifier 10.1109/TNN.2007.909535

After performing FS, we have obtained a significant improvement in the overall results with respect to learning with the whole set of variables in most of the data sets tested. To the best of our knowledge, these results would rank among the top published ones with MLPs for these data sets. However, we point out that these improvements are only obtained when adequate criteria for FS with MLPs are used. Although we have focused on FS with MLPs and the SBS procedure, we claim that the results can be extended to other neural models and other search strategies.

The rest of this paper is organized as follows. Section II describes several existing FS procedures for MLPs. A basic SBS scheme for FS with MLPs and its decision issues are discussed in Section III. The experimental work is described in Section IV and discussed in Section V. Finally, Section VI concludes and outlines some directions for further research.

II. Feature Selection With MLPs

According to the bias/variance decomposition, either high bias or variance can contribute to poor performance [12], [8]. Typically, very flexible models, such as MLPs, may lead to unbiased estimators but probably with high variance (due to overfitting). A rigid model, in contrast, may lead to small variance but high bias. There is a tradeoff between the bias and variance contributions to the error, where the optimal performance is achieved.

When too many variables are considered, there are many different solutions capable of fitting the same data set, but only a few of these solutions will lead to good generalization. If the system gives some importance to irrelevant variables it will use this information for new data, leading to poor generalization even if we try to control the overfitting. This is one of the most important motivations for FS from a machine learning point of view.

The problem of FS can be defined as follows: Given a set of N_f features, select a subset that performs the best under certain evaluation criterion. From a computational point of view, the definition of FS leads to a search problem in a space of 2^{N_f} elements. Therefore, two components must be specified: the search procedure through the space of feature subsets and the feature subset evaluation criterion.

A. The Most Common Approach for FS With MLPs: SBS as Search Procedure and Loss Function as Saliency

Regarding the search procedure, most FS algorithms for MLPs use the SBS algorithm, also known as sequential backward elimination. SBS is a top–down process. Starting from the complete set of available features, one feature is deleted at every step of the algorithm, chosen on the basis of which the available candidates gives rise, together with the remaining features, to the best value of the evaluation criterion. Ideally, the performance of the system is expected to improve until a subset of features remains for which the elimination of further variables results in performance degradation. In general, SBS helps to detect irrelevant[1] variables in the first steps, trying to maintain the interactions among the variables [9]. The SBS procedure with MLPs starts by training a network with the whole set of features. Then, the saliencies (see the following) are computed and the feature with the lowest saliency is removed. The weights of the network are modified and the loop starts again until a certain criterion is satisfied.

With respect to the feature subset evaluation criterion, FS algorithms for MLPs with the SBS procedure use different variations of the concept of *saliency*. The concept of saliency is also present in some pruning methods [33]. Features with large saliencies are considered more important than features with small ones, so that features with small saliencies can be eliminated. Following the wrapper approach, the most commonly used saliency of a feature is the difference in the values of the loss function, usually the sum-of-squares error (SSE), in presence and absence of that feature. This is surely the optimal criterion for FS from a machine learning point of view [24].

Several variations of FS methods for MLPs and the SBS procedure for which the saliency is computed as the value of the loss function can be found in the literature, as explained in the following.

Some works compute an approximation to the difference in the SSE when the feature is removed from the network [28], [25] or its weights are set to zero [6], [7], [40]. The latter models are based on the optimal brain damage [22] and optimal brain surgeon [16] procedures that approximate the difference in the SSE when a weight is removed from a trained network. These values are obtained under several assumptions, such as approximating the difference by the derivative or discarding the nondiagonal terms in the Hessian matrix.

In other studies [37], [35], [44], [46], the saliency of a feature is directly computed as the value of the loss function when the feature is removed from the network (or, equivalently, its weights are set to zero). In [37] and [46], the cross entropy with a penalty function is used as a loss function. In [37], the penalty function encourages small weights to converge to zero and prevents the weights from taking values too large. In [46], the penalty function constrains the derivatives of the output and hidden units, looking for good generalization properties. In [44], a new network is analytically constructed, starting from the original one, to approximate the network that would be obtained if the feature was eliminated. The analytical construction is based on finding a new neural network with similar activations in the hidden layer to those on the original network.

An alternative to setting to zero the values of a feature is substituting them by their average value [1], [27], assuming that the influence of the feature on the network output will be removed. In [9] and [10], the input value is replaced by a value estimated from its least mean square regression parameters, so that it is possible to compute a lower bound of the accuracy of the network.

[1]There is no commonly accepted definition of the relevance of a variable [4], [24]. Given a data set, we consider that a variable is irrelevant for a machine learning system when its optimal performance is not affected negatively by the absence of that variable, following [24, p. 29]. Note that this is a dynamic definition, since the relevance of a variable may be affected by the presence or absence of other ones.

B. Other Saliencies Different From the Loss Function

The derivative of the outputs with respect to the input units is the motivating idea of the saliency defined in [31] and [36]. With the same underlying idea, three measures of sensitivity are defined in [48] (the squares average, the absolute value average, and the maximum of the derivative). The saliency defined in [11] is the variance of the mean derivative.

In [41], the definition of saliency in [36] is used to compare an artificially introduced noisy variable with the original features. Assuming normality on the saliency of the noisy variable, features whose saliency falls inside a certain confidence interval are removed. In [13], the input values are perturbed with the injection of Gaussian noise. The variance of the injected noise is allowed to be different for every feature and it is modified during the training procedure. The saliency of a feature is the inverse of its relative variance at the end of the training process.

C. Other Search Schemes Different From SBS

Starting from a full network, hidden and input units are sequentially removed in [42] whenever the performance of the reduced network improves. The procedure stops when the elimination of any feature does not lead to a better performance.

The classical sequential forward selection procedure is performed in [29] with the SSE as saliency.

In [20], the Taguchi method is used to construct an orthogonal array where every row is associated with a subset of features (indicating presence/absence). These subsets of features are fixed *a priori*, previous to the training of the networks. The Taguchi method is a procedure to set up experiments that only require a fraction of the whole (factorial) combinations [43]. With this method, only $O(N)$ MLPs are trained.

Classical genetic algorithms guide the search process in [47]. The fitness of every chromosome (the subset of features) is computed taking into account the accuracy of the model obtained with that chromosome. In [21], multiobjective genetic algorithms are performed to simultaneously minimize the error rate and the number of features.

The sequential forward floating selection search algorithm [32] is used in [23] with piecewise linear networks and the orthogonal least squares procedure [5]. First, the number of clusters for the piecewise linear network model is determined. Then, the sequential forward floating selection algorithm is used to guide the search. At every step, the saliency is the error reduction ratio after adding or removing a feature, which is computed with the orthogonal least squares procedure. Since each cluster is modeled by a linear network, the algorithm is computationally efficient.

III. Decision Issues in a Basic SBS Scheme for FS With MLPs

The assumption that FS helps to improve the generalization performance by a reduction of the dimensionality and by eliminating irrelevant variables is, according to the bias/variance decomposition, theoretically sound. However, as far as we know, the results obtained by MLPs using different methods of FS do not show this improvement. We carefully reviewed the most common approaches in the literature of FS with MLPs (we found more than 60 papers, briefly described in Section II). Many of them do not report the differences between the models obtained with the whole set of features and those obtained after FS are carried out. When these differences are reported, the results are, in general, similar (if not worse) to the ones obtained without performing FS. Significant improvements are obtained in very few cases.

Many saliencies aim to remove the features that contribute more to the error of a network trained with the whole set of features. This implicitly assumes that networks trained with the whole set of features are good models, which is not true when irrelevant variables are present. An MLP with irrelevant variables is more flexible than the same model without them, leading to better approximations in the training set. However, an MLP that uses irrelevant variables will have poor generalization. This problem can be alleviated if there are enough data to filter irrelevant variables, but this is not usually the case. In addition, the number of features subsets with irrelevant variables is $2^{N_i} - 1$ times larger than that without them, where N_i is the number of irrelevant variables. Other saliencies are based on the hypothesis that irrelevant features produce smaller variations in the output values than relevant ones. Since irrelevant variables may interact in a nonlinear and complex way with the rest of variables, we cannot expect to obtain useful information for the detection of irrelevant variables from networks trained with them.

As stated in Section II, the most common approach for FS with MLPs is based on the SBS search procedure and it uses the SSE as evaluation criterion. This approach involves taking a number of decisions that have not been always addressed in the literature. In general, there are neither a commonly accepted criterion nor comparative studies about the following issues.

1) Whether or not the network should be retrained at every step with every feature temporarily removed before computing its saliency. To the best of our knowledge, the only methods that retrain the network at every step with every feature temporarily removed/added before computing its saliency are those described in [20], [21], [23], [29], [35], [42], and [47]. Among them, only the model presented in [35] is a pure SBS procedure. See Section II for a brief description of these methods.
2) The stopping criterion for the training phase. Usually, networks are trained until a local minimum of the loss function for the training set is found, although this point is not always addressed in the literature. We assume that the training process under expressions like "train a network for a number of epochs ..." or "after the network was trained ..." tries to find a local minimum of the loss function for the training set. There are several exceptions, where an early stopping procedure is performed, usually with a validation set (see, for example, [1], [11], [44], and [46]).

When we analyzed the literature in depth, we observed that the improvement of the results may be correlated with these decision issues. The main motivation of this paper is to study their influence when performing FS with MLPs.

A basic SBS scheme for FS with MLPs using the SSE as the saliency of a feature is presented in Fig. 1. This scheme will be the basis for the experiments presented in Section IV. The outer loop follows the scheme of the classical SBS procedure, where a feature is permanently eliminated at every step. The inner loop

Algorithm

Let V_1 the whole set of N_f features

for $N = 1$ **up to** $N_f - 1$ **do**

Train the network with V_N until a certain stopping criterion is satisfied (*decision issue*, see text for details), and keep its generalization performance

for each $v \in V_N$ **do**

Set $V = V_N - \{v\}$

Optionally, retrain the network with V (*decision issue*, see text for details)

Obtain the saliency of v by computing the value of the sum-of-squares error E_v on a validation data set

end for

Set $V_{N+1} = V_N - \{v^*\}$, where v^* corresponds to the lowest value of E_v in the previous loop

end for

Return V_{N^*}, where N^* corresponds to the best generalization performance of the network in the previous loop

end Algorithm

Fig. 1. Basic SBS procedure for FS with MLPs using the SSE as evaluation criterion (saliency).

selects the variable to eliminate the following: 1) the network is trained with the whole set of available features until a certain stopping criterion is satisfied, 2) every feature is temporarily removed, 3) the network is optionally retrained (with the same stopping criterion), and 4) the value of the SSE is computed (on a validation data set). The variable corresponding to the lowest value of the SSE is permanently eliminated.

The two aforementioned decision issues are highlighted in the algorithm in Fig. 1. Several configurations were tested in our experiments, as explained in the following.

The first decision issue involves whether the network is retrained or not after the feature is temporarily removed before computing its saliency. Therefore, the saliency of a feature can be computed following two approaches.

1) First, the network is trained with the whole set of available features. Then, every feature is temporarily removed and the SSE is computed. The saliency of every feature is computed in the same trained network. This procedure involves training $N_f - 1$ networks.
2) For every feature, the network is retrained with that feature temporarily removed. For every trained network, the SSE is computed. This procedure is computationally more expensive than the previous one, since it involves training $N_f(N_f+1)/2$ networks (in this case, the training prior to the inner loop can be omitted).

Both possibilities were tested. Note that these two ways of computing the saliency will yield very different results for the same feature, since the corresponding output functions of the trained networks will be very different as well. Suppose that a trained network uses a certain feature to fit the data and a new network is trained without it. The new network will use other features to fit the data. There is no reason to think that the relative saliencies of every feature remain unchanged with respect to those obtained in the original network without retraining it. The same happens if the feature values are substituted by its average value. Intuitively, a more reliable estimation of the saliency should be obtained by retraining the network with every feature temporarily removed. The good results obtained (in general) by models that retrain the network before computing the saliency seem to validate this intuition.

The second decision issue is the stopping criterion in the training phase. Two different stopping criteria were tested. The first one is to train until a minimum of the SSE for the training set is achieved. The second one is to stop where a minimum for a validation set is obtained. Suppose that the properties of the data set allow the negative effect of overfitting to appear. It seems that performing early stopping with a validation set could be the most promising idea. However, it can also be argued that overtraining the network until a local minimum of the SSE for the training set is achieved forces the system to use all the available variables as much as possible. In this situation, irrelevant variables should be more outstanding when the system is not allowed to use them, as pointed out in [35].

In summary, four configurations with different combinations of network retraining/stopping criterion will be tested and compared.

IV. Experiments

Some experiments on both artificial (Section IV-B) and standard benchmark data sets (Section IV-C) for classification problems were performed. For every data set, the four aforementioned configurations were tested with the SBS procedure for MLPs described in Fig. 1, showing important differences.

A. Experimental Setting

All the experiments were performed with stratified cross validation (CV). Prior to every CV, the examples in the data set were randomly shuffled. Five runs of a double five-fourfold CV [34] were performed as follows. A fivefold CV (the outer CV) was performed to obtain five folds (four folds to "learn" and one fold to test). Then, a fourfold CV (the inner CV) was performed with the four folds of the "learning set" of the outer CV (three folds to train and one fold to validate). Therefore, the number of trained networks in every double five-fourfold CV was 20,

TABLE I
ARCHITECTURES, LEARNING PARAMETERS, AND EXECUTION TIMES OF A DOUBLE FIVE-FOURFOLD CV FOR EVERY DATA SET

Data Set	# Hidden Units	# Epochs	Weights Range	Learning Rates	Time
Augmented XOR	20	500	I-H:3.0 H-O:0.001	I-H:0.2 H-O:0.002	56"
Augmented Two Spirals	40	1500	I-H:5.0 H-O:0.001	I-H:0.002 H-O:0.0002	5' 32"
Australian Credit	3	1500	I-H:0.5 H-O:0.001	I-H:0.02 H-O:0.0001	2' 17"
Cleveland Heart	20	1000	I-H:2.0 H-O:0.02	I-H:0.2 H-O:0.002	1' 48"
Hepatitis	20	300	I-H:3.0 H-O:0.001	I-H:0.2 H-O:0.005	38"
Ionosphere	20	300	I-H:1.0 H-O:0.1	I-H:0.05 H-O:0.005	1' 10"
Sonar	35	150	I-H:2.0 H-O:0.1	I-H:0.05 H-O:0.005	56"
Vehicle	25	2000	I-H:3.0 H-O:0.001	I-H:0.01 H-O:0.00005	3' 38"
Voting	10	500	I-H:2.0 H-O:0.001	I-H:0.1 H-O:0.005	1' 7"
Lung Cancer	10	500	I-H:2.0 H-O:0.02	I-H:0.02 H-O:0.01	43"

giving a total of 100 runs for each training step. The saliency of every feature was computed as the average SSE over the 100 validation folds.

A computational cost as small as possible for the whole process was required. We used MLPs with one hidden layer of sinusoidal units and hyperbolic tangents in the output layer, trained with standard backpropagation (BP) in online mode. MLPs using sine activation functions (and an appropriate choice of initial parameters) usually need less hidden units and learn faster than MLPs with sigmoid functions when both types are trained with BP [39].

The number of hidden units and learning rates was chosen, for every data set, so as to achieve a small and smoothly decreasing training error in a reasonable number of epochs. Table I shows, for every data set, the architectures and learning parameters chosen. Weights were initialized randomly within a certain interval $[-W/2, +W/2]$, different for every layer. This is shown in Table I as I–H:W for the "input-to-hidden" layer and H–O:W for the "hidden-to-output" layer. Learning rates were also different for every layer. Momentum was set to 0. The execution times of a double five-fourfold CV on a Pentium 4 CPU at 1.8 GHz are also shown (our program was implemented in C).

In order to introduce the least external variability in the experiments, all the configurations were tested with the same (although different for every data set) network architecture and parameters.

For every data set, we applied a paired t-test (confidence level $\alpha = 0.05$) to determine if the mean accuracy of the four configurations tested was significantly different. The percentages of correctly classified patterns on the CV test sets were used for the paired t-test. The group of the best results that are not significantly different from each other but significantly better than the rest are shown in boldface in Tables III and IV.

B. Experiments on Artificial Data Sets

The artificial data consisted of augmented versions of the two well-known data sets: XOR and *two spirals*. These data sets were chosen as two prototypes of an easy and a difficult problems for FS, respectively. Irrelevant variables were added to the original ones, prior to test the aforementioned configurations. Therefore, the optimal subsets of variables (there were equivalent ones) were known *a priori*.

TABLE II
ADDED VARIABLES FOR THE *AUGMENTED XOR* (LEFT) AND *AUGMENTED TWO SPIRALS* (RIGHT) DATA SETS. $\mathcal{N}$ AND $\mathcal{U}$ ARE THE NORMAL AND UNIFORM DISTRIBUTIONS, RESPECTIVELY. THE REASON WHY x_5 WAS NOT ADDED TO THE *AUGMENTED XOR* DATA SET IS THAT THIS VARIABLE ALONE ALLOWS TO CORRECTLY CLASSIFY THE WHOLE DATA SET. ITS NOISY VERSION x_{10} WAS NOT INCLUDED

Variable	*Augmented XOR*	*Augmented Two Spirals*
x_3	x_1^2	x_1^2
x_4	x_2^2	x_2^2
x_5	-	$x_1 \cdot x_2$
x_6	$x_1 + x_2$	$x_1 + x_2$
x_7	$x_1 - x_2$	$x_1 - x_2$
x_8	$x_1^2 + \mathcal{N}(0,1)$	$x_1^2 + \mathcal{N}(0,1)$
x_9	$x_2^2 + \mathcal{N}(0,1)$	$x_2^2 + \mathcal{N}(0,1)$
x_{10}	-	$x_1 \cdot x_2 + \mathcal{N}(0,1)$
x_{11}	$x_1 + x_2 + \mathcal{N}(0,1)$	$x_1 + x_2 + \mathcal{N}(0,1)$
x_{12}	$x_1 - x_2 + \mathcal{N}(0,1)$	$x_1 - x_2 + \mathcal{N}(0,1)$
x_{13}	$\mathcal{U}(0,1)$	$\mathcal{U}(0,1)$
x_{14}	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$
x_{15}	$\mathcal{N}(0,5)$	$\mathcal{N}(0,5)$

1) Augmented XOR Data Set: An augmented version of the XOR data set was created as follows. First, a symmetric data set was constructed as an extension of the 2-D XOR data set to $[-1,+1] \times [-1,+1]$, with 150 points for each class. The input values of the two original variables (x_1, x_2) were taken from

$$\{(z_i + \delta(z_i, c), z_j + \delta(z_j, c)) \mid i, j = 0, \ldots, 9,\ c = 0, 1, 2\}$$

where $z_k = -(9/10) + ((2.k/10))$ and $\delta(z, c) = -(c.\mathrm{sign}(z)/30)$. The target value is the sign of $x_1 \cdot x_2$. Second, 11 new variables were artificially added to the two original ones. The whole set of added variables can be seen in Table II. These new variables were defined to be redundant or independent of the original ones (and, therefore, irrelevant). Some of them were noisy. When needed, input variables were linearly scaled in $[-1, +1]$.

TABLE III
RESULTS OF DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION TESTED FOR ARTIFICIAL DATA SETS. FIGURES IN BOLDFACE INDICATE SIGNIFICANT DIFFERENCES OF THE BEST RESULTS

Augmented XOR				
Retrain	**Stopping Criterion**	**Test**	**MSE**	**NVar**
No	Training	99.28%	0.06	3
Yes	Training	99.22%	0.05	3
No	Validation	99.15%	0.07	2
Yes	Validation	99.08%	0.06	2

Augmented Two Spirals				
Retrain	**Stopping Criterion**	**Test**	**MSE**	**NVar**
Yes	Training	**99.89%**	0.01	3
Yes	Validation	**99.66%**	0.02	3
No	Training	92.30%	0.25	6
No	Validation	87.10%	0.39	6

TABLE IV
DESCRIPTION OF THE BENCHMARK DATA SETS. COLUMN "NVAR" SHOWS THE NUMBER OF VARIABLES, COLUMN "NCLA" THE NUMBER OF CLASSES, AND COLUMN "NEXA" THE NUMBER OF EXAMPLES. SEVERAL RESULTS FOUND IN THE LITERATURE FOR THESE DATA SETS WITH THE WHOLE SET OF FEATURES ARE ALSO SHOWN (COLUMNS "ML ALG," "TEST," "SAMPLING," AND "REF."). COLUMN "ML ALG" INDICATES THE MACHINE LEARNING ALGORITHM USED. MLP + BP MEANS "MLPS TRAINED WITH BP"

Data Set	**NVar**	**NCla**	**NExa**	**ML Alg**	**Sampling**	**Test**	**Ref.**
Australian Credit	15	2	690	MLP+BP	10-fold CV	85.2%	[30]
Cleveland Heart	13	2	303	MLP+BP	10-fold CV	81.4%	[30]
Hepatitis	19	2	155	MLP+BP	10-fold CV	79.9%	[30]
Ionosphere	33	2	351	MLP+BP	10-fold CV	90.3%	[30]
Sonar	60	2	208	MLP+BP	10-fold CV	83.4%	[30]
Vehicle	18	4	846	MLP+BP	10-fold CV	75.1%	[30]
Voting	16	2	435	MLP+BP	10-fold CV	95.1%	[30]
Lung Cancer	56	3	32	MLP+BP	5-4-fold CV	40.0%	This work

The data can be correctly classified only by looking at the sign of the first two variables. This can be easily done by combining sines and cosines, the activation functions used in the experiments. This is the reason why we consider this data set an easy one for FS.

The results of the four aforementioned configurations tested are shown in Table III (column "test") as the average percentage of correctly classified patterns on the respective test sets in the networks with minimum validation set error after every variable is permanently eliminated. Column "MSE" indicates the mean SSE on the test set and column "NVar" the number of variables where these results were obtained (see Fig. 1).

2) Augmented Two Spirals Data Set: In a similar way to the one used to create the *augmented XOR* data set, an augmented version of the well-known *two spirals* data set[2] was constructed, where 13 irrelevant variables were artificially added to (x_1, x_2), the two original ones. Some of them were noisy. The whole set of added variables can be seen in Table II. The values of the input variables were linearly scaled in $[-6.5, +6.5]$ (the range of the original variables) when they were not in this range. The target value was that as (x_1, x_2) in the original data set. Each of the original training, validation, and test sets comprises 194 2-D points with balanced classes. In order to perform the experiments with CV, the three data sets were joined into a single data set. Results are shown in Table III.

[2]C source code available in the Carnegie Mellon University Artificial Intelligence Repository [18].

C. Experiments on Benchmark Data Sets

Several widely used data sets for classification problems were selected from the University of California at Irvine (UCI) [3] and Statlog [26] repositories. A brief description of these data sets can be found in Table IV, together with several results found in the literature for MLPs with these data sets and the whole set of features.

A summary of the results obtained for every configuration tested with the SBS procedure for MLPs described in Fig. 1 is shown in Table V. Detailed results are shown in Table VI.

V. DISCUSSION

A different behavior was observed for the two artificial data sets (see Section IV-B). Whereas for the *augmented XOR* data set no difference was observed among the configurations under study, only two configurations showed good results in the *augmented two spirals* data set. The experiments on benchmark data sets (see Section IV-C) confirmed that the behavior observed in the *augmented XOR* data set is not the general rule. However, it shows that in some cases any of these configurations can be successfully used. A detailed analysis comparing the behavior along the FS process is given in this section, providing an explanation of the observed differences.

A. Artificial Data Sets

Fig. 2 shows, for every configuration, the evolution of the percentage of correct examples in the training and test sets with

TABLE V
SUMMARY (FROM TABLE VI) OF THE BEST RESULTS OF DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION TESTED FOR BENCHMARK DATA SETS

Data Set	Retrain	Stopping Criterion	Test	Mse	NVar
Australian Credit	Yes	Training	87.41%	0.39	5
Cleveland Heart	No	Validation	82.85%	0.53	12
Hepatitis	Yes	Training	**93.90%**	0.24	3
Ionosphere	Yes	Training	**93.61%**	0.22	5
Sonar	Yes	Validation	**89.73%**	0.33	14
Vehicle	Yes	Validation	79.98%	0.29	9
Voting	Yes	Validation	**96.69%**	0.11	6
Lung Cancer	Yes	Validation	**86.88%**	0.31	9

TABLE VI
RESULTS OF DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION TESTED FOR BENCHMARK DATA SETS. FIGURES IN BOLDFACE INDICATE SIGNIFICANT DIFFERENCES OF THE BEST RESULTS

Australian Credit

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Training	87.41%	0.39	5
Yes	Validation	87.33%	0.39	6
No	Validation	86.79%	0.40	7
No	Training	86.51%	0.41	8

Cleveland Heart

Retrain	Stopping Criterion	Test	Mse	NVar
No	Validation	82.85%	0.53	12
No	Training	82.65%	0.53	11
Yes	Training	82.63%	0.53	12
Yes	Validation	82.58%	0.53	4

Hepatitis

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Training	**93.90%**	0.24	3
Yes	Validation	**93.77%**	0.25	3
No	Validation	88.97%	0.40	1
No	Training	88.26%	0.36	3

Ionosphere

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Training	**93.61%**	0.22	5
No	Validation	92.77%	0.24	5
Yes	Validation	92.73%	0.24	5
No	Training	92.57%	0.24	5

Sonar

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Validation	**89.73%**	0.33	14
Yes	Training	87.95%	0.36	11
No	Training	85.02%	0.46	46
No	Validation	84.49%	0.47	50

Vehicle

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Validation	79.98%	0.29	9
Yes	Training	79.51%	0.30	8
No	Validation	78.86%	0.31	14
No	Training	78.79%	0.31	14

Voting

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Validation	**96.69%**	0.11	6
Yes	Training	**96.43%**	0.12	6
No	Training	96.18%	0.12	8
No	Validation	96.15%	0.12	12

Lung Cancer

Retrain	Stopping Criterion	Test	Mse	NVar
Yes	Validation	**86.88%**	0.31	9
Yes	Training	**85.30%**	0.33	10
No	Validation	71.41%	0.57	5
No	Training	71.91%	0.54	7

respect to the number of eliminated variables in the artificial data sets.

As expected, the addition of irrelevant features affects very negatively the performance of sinusoidal MLPs in this problem, even if overfitting is tried to be controlled. The information needed to learn the problem is present, but the system is not able to use it in a proper way. The reason for this fact probably is the relatively small number of examples in the data set that does not allow to filter irrelevant features. As far as variables are eliminated, performance improves. However, many variables must be eliminated to obtain good performance.

1) Augmented XOR Data Set: Results in Table III seem to indicate that there is no difference among the different configurations under study, but this behavior is not the general rule (see the experiments with the *augmented two spirals* data set and benchmark data sets in Section IV-C). As previously mentioned, this data set was constructed as a prototype of an easy FS problem.

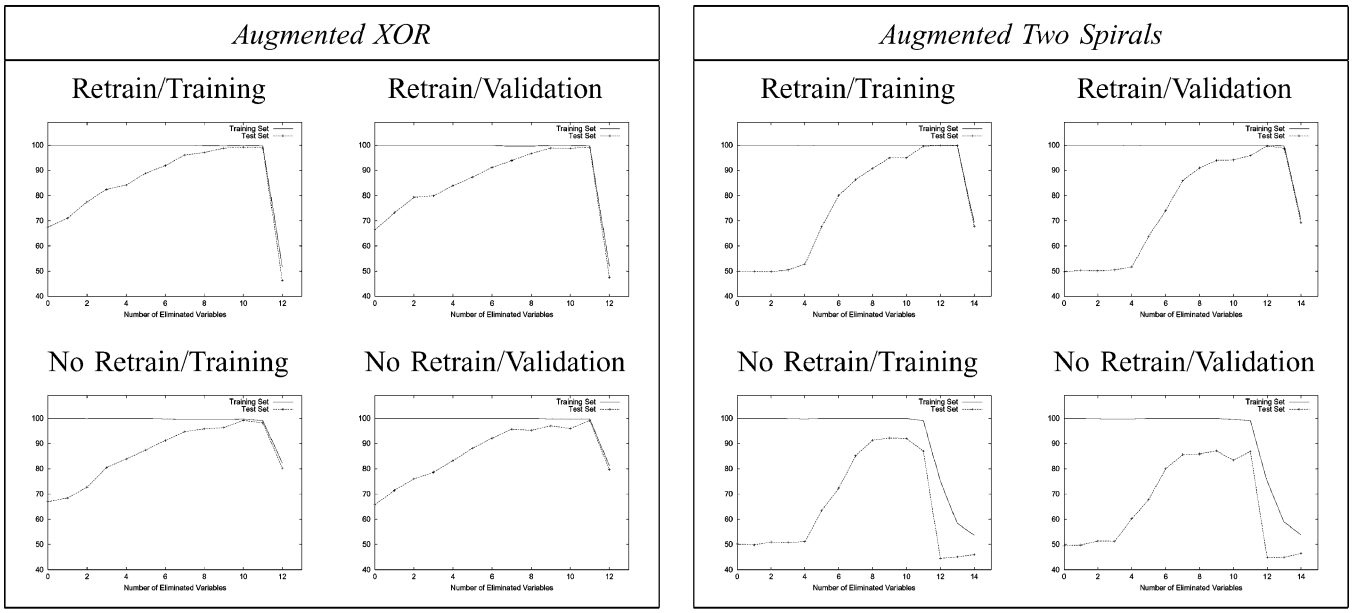


Fig. 2. Percentage of correct examples for the artificial data sets in the training and test sets with respect to the number of eliminated variables in the SBS procedure for MLPs with retraining (top) and without it (bottom).

TABLE VII
ORDER OF VARIABLE ELIMINATION (FROM LEFT, FIRST, TO RIGHT, LAST) FOR THE *AUGMENTED TWO SPIRALS* DATA SET. FOR EVERY CONFIGURATION, VARIABLES ON THE RIGHT-HAND SIDE ARE THE MOST IMPORTANT CONSIDERED ONES. THE SYMBOL ▶ INDICATES THE POINT FROM WHICH THE VARIABLES HAVE BEEN SELECTED

Retrain	Stopping Criterion	Order of Variable Elimination
Yes	Training	$x_4\ x_{11}\ x_{13}\ x_{14}\ x_{15}\ x_{12}\ x_{10}\ x_9\ x_8\ x_3\ x_5\ x_2 \blacktriangleright x_1\ x_6\ x_7$
Yes	Validation	$x_9\ x_7\ x_{15}\ x_{13}\ x_{14}\ x_{12}\ x_{11}\ x_{10}\ x_8\ x_4\ x_3\ x_5 \blacktriangleright x_6\ x_1\ x_2$
No	Training	$x_9\ x_1\ x_{13}\ x_{14}\ x_{15}\ x_{11}\ x_{12}\ x_{10}\ x_5 \blacktriangleright x_2\ x_6\ x_7\ x_4\ x_8\ x_3$
No	Validation	$x_{14}\ x_{15}\ x_5\ x_{13}\ x_{11}\ x_{12}\ x_{10}\ x_2\ x_1 \blacktriangleright x_6\ x_9\ x_7\ x_4\ x_8\ x_3$

2) Augmented Two Spirals Data Set: Different from the *augmented XOR* data set, the best results were obtained retraining the network with every feature temporarily removed before computing its saliency. This positive effect can be seen both in the number of selected features and the overall performance. There was no significant difference with regard to the stopping criterion. The evolution of the training set error was also better when retraining is present than without it (see Fig. 2).

3) Comparative Analysis: The order of variable elimination for the *augmented two spirals* data set can be seen in Table VII. Ideally, variables x_{13}, x_{14}, and x_{15} should never be considered as important, and variables from x_1 to x_7 should be more important than variables from x_8 to x_{12}. The claim about x_{13}, x_{14}, and x_{15} is satisfied by all the configurations. However, when retraining is not performed, several noisy variables (x_8 or x_9, for example) were considered to be important. This did not happen when retraining was present and it was not observed for the *augmented XOR* data set. These results can be explained by looking at the behavior of every configuration during the SBS procedure, as explained in the following.

Without retraining, the temporary elimination of any variable leads to large errors in the training set when compared to the training error with the whole set of features. The third column in Table VIII shows this fact for the first step of the SBS procedure in the *augmented two spirals* data set. This happened because the obtained solution after the training process used all the variables in a significant way. Therefore, the elimination of a feature substantially modified the obtained function. In addition, noise-free variables (from x_1 to x_7) did not seem to be used more than the rest in order to learn the data set. The same behavior was observed in several subsequent steps. Therefore, when retraining is not performed the permanent elimination of a variable is decided in quite a random way. For the *augmented XOR* problem, in contrast, a line can be drawn which clearly separates variables x_1 to x_7 from the rest (see the second column in Table VIII). The reason for this different behavior is not clear, but it is probably related to the respective percentages of correct examples in the test set during the first steps of the SBS procedure (the *augmented two spirals* suffers a much stronger performance degradation than the *augmented XOR* problem; see Fig. 2), indicating that irrelevant variables are not equally important in the respective obtained solutions.

B. Benchmark Data Sets

Similar to the *augmented two spirals* data set, the best results for benchmark data sets were obtained by retraining the network with every feature temporarily removed before computing its saliency (recall that figures in boldface indicate significant differences). The retraining of the network turns out to be critical, although, as previously mentioned, it is hardly used

TABLE VIII
PERCENTAGE OF CORRECT EXAMPLES IN THE TRAINING SET AFTER THE TEMPORARY ELIMINATION OF EVERY VARIABLE IN THE FIRST STEP OF THE SBS PROCEDURE WITHOUT RETRAINING. THE FIRST COLUMN INDICATES THE TEMPORARILY ELIMINATED VARIABLE. VARIABLES WITH LARGE PERCENTAGES CAN BE INTERPRETED AS VARIABLES THAT ARE NOT VERY IMPORTANT IN THE OBTAINED SOLUTION. NOTE THAT THE WHOLE SET OF FEATURES ALLOWS THE TRAINING SET TO FIT PERFECTLY

Variable	*Augmented XOR*	*Augmented Two Spirals*
x_1	84.55%	60.56%
x_2	83.38%	59.58%
x_3	84.40%	51.34%
x_4	82.70%	52.39%
x_5	-	66.27%
x_6	90.60%	59.08%
x_7	88.97%	57.59%
x_8	98.30%	54.17%
x_9	97.90%	52.95%
x_{10}	-	65.12%
x_{11}	96.25%	61.40%
x_{12}	94.73%	59.86%
x_{13}	94.53%	54.94%
x_{14}	97.92%	61.87%
x_{15}	96.32%	61.23%
None	100.0%	100.0%

in practice. Therefore, the increase in the computational cost associated with this scheme is rewarded with a significant performance improvement.

Regarding the stopping criterion, it is unclear which is the best choice. For the *sonar* data set, the "validation" strategy worked best. For the *ionosphere* data set, in contrast, the "training" strategy selected a better subset of variables.[3] For the rest of the problems, both configurations can be considered as equivalent. The goodness of the "validation" configuration can be intuitively explained, since it tries to obtain the best possible generalization results at every step. The "training" configuration, in contrast, improves performance by forcing overtraining (and measuring the SSE in a validation set). This is a nonintuitive result. The explanation pointed out in [35] is that forcing the system to use all the available features as much as possible makes irrelevant variables more outstanding when the system is not allowed to use them, since it leads to larger saliency differences.

Although in a different scale, a similar behavior to that of the *augmented two spirals* data set was observed. First, the training set could be fitted with a much smaller subset of features than the original one. Regarding generalization results, performance improved until a subset of features remained for which the elimination of further variables resulted in performance degradation. This behavior seems to reveal the existence of irrelevant variables that the SBS procedure has detected and eliminated. However, the differences among the different configurations suggest that, as in the *augmented two spirals* data set, there are several variables that allow to fit the training set, but do not provide good generalization. The number of available examples is not large enough to filter these variables.

Finally, an important improvement in the overall results can be appreciated with respect to learning with the whole set of variables (see Tables IV and V) An important reduction in the final number of selected variables is also observed. The good results obtained when retraining is present are mainly due, in our opinion, to a proper detection of irrelevant variables.

[3]Equal number of variables does not imply the same variables.

VI. CONCLUSION AND FUTURE WORK

An experimental study comparing different criteria to perform FS with MLPs and the SBS procedure within the *wrapper* approach has been carried out. Two decision issues have been highlighted and studied, namely, the stopping criterion for the network training and the network retraining before computing the saliency.

Experimental results on artificial and benchmark data sets indicate that the increase in the computational cost associated with retraining the network with every feature temporarily removed before computing its saliency is rewarded with a significant performance improvement. Despite being quite intuitive, this strategy has been hardly used in practice, probably because of its high computational cost. This issue turns out to be critical. A somehow nonintuitive conclusion can be drawn by looking at the stopping criterion, where it is suggested that forcing overtraining may be as useful as early stopping.

A significant improvement in the overall results with respect to learning with the whole set of variables has been obtained, with an important reduction in the final number of selected variables. To our knowledge, the obtained results would rank among the top results with MLPs for the data sets tested. Parenthetically, this confirms that FS plays a very important role for MLPs, specially when the number of available examples is small. In our opinion, the good results obtained are mainly due to a proper detection of irrelevant variables.

In summary, for FS with MLPs using the SBS procedure and the SSE as evaluation criterion, we recommend to retrain the network with every feature temporarily removed before computing its saliency, specially, for complex problems. A literature review indicates that this is not the most common approach.

We claim that these recommendations can be extended to other neural models and other search strategies that could be adjusted to the required specifications. For example, it could be performed with support vector machines [45] using some function of the margin as saliency and different hardness of the margin as the stopping criterion.

The main drawback of the SBS procedure for MLPs presented in this work is its computational cost, particularly when retraining is performed. Training algorithms faster than BP may obviously be used, but BP was not the main source of the computational cost in our experiments. The algorithm is quadratic with respect to the number of variables, and the first steps of the algorithm, when many irrelevant variables still remain, take most of the computational time. Several heuristics can be designed to eliminate the most clearly irrelevant variables with a low computational cost. Then, when a reasonable number of features remains, the whole procedure will start. An alternative

solution can be to parallelize the process of elimination of one variable (that is, the retraining of the network with every feature temporarily removed).

The network architecture can also be considered as part of the process, since it might happen that different set of variables need different network parameters. Constructive or adaptive learning algorithms can be used to this end.

The analysis of the experiments on artificial data sets suggests that the number of examples (with respect to the input dimension) can explain the different behavior of the different configurations tested. This issue deserves further research. Several experiments can be designed in order to study whether the differences observed among the configurations are maintained or not when the number of examples vary.

A theoretical analysis, that could shed light on how and why a particular combination is better than others or whether there is any relationship between the selection of a combination of network retraining/stopping criterion and the bias/variance tradeoff, would also be interesting.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] B. Baesens, S. Viaene, J. Vanthienen, and G. Dedene, "Wrapped feature selection by means of guided neural network optimisation," in *Proc. Int. Conf. Pattern Recognit.*, 2000, vol. 2, pp. 113–116.

[2] C. M. Bishop*, Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.

[3] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998 [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html

[4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1–2, pp. 245–271, 1997.

[5] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their applications to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[6] T. Cibas, F. F. Soulié, P. Gallinari, and Š. Raudys, "Variable selection with optimal cell damage," in *Proc. Int. Conf. Artif. Neural Netw.*, 1994, vol. 1, pp. 727–730.

[7] T. Cibas, F. F. Soulié, P. Gallinari, and Š. Raudys, "Variable selection with neural networks," *Neurocomputing*, vol. 12, no. 2–3, pp. 223–248, 1996.

[8] P. Domingos, "A unified bias-variance decomposition and its applications," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 231–238.

[9] M. Egmont-Petersen, W. R. M. Dassen, and J. H. C. Reiber, "Sequential selection of discrete features for neural networks—A Bayesian approach to building a cascade," *Pattern Recognit. Lett.*, vol. 20, no. 11–13, pp. 1439–1448, 1999.

[10] M. Egmont-Petersen, J. L. Talmon, A. Hasman, and A. W. Ambergen, "Assessing the importance of features for multi-layer perceptrons," *Neural Netw.*, vol. 11, no. 4, pp. 623–635, 1998.

[11] A. P. Engelbrecht, "A new pruning heuristic based on variance analysis of sensitivity information," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1386–1399, Nov. 2001.

[12] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, 1992.

[13] Y. Grandvalet, "Anisotropic noise injection for input variables relevance determination," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1201–1212, Nov. 2000.

[14] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[15] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005, vol. 17, pp. 545–552.

[16] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1993, vol. 5, pp. 164–171.

[17] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 121–129.

[18] M. Kantrowitz, "CMU artificial intelligence repository," Schl. Comput. Sci., Univ. Carnegie Mellon, Pittsburgh, PA, 1993 [Online]. Available: http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/areas/n% eural/bench/cmu

[19] J. Kittler, "Feature set search algorithms," in *Pattern Recognition and Signal Processing*, C. H. Chen, Ed. Alphen ann den Rijn, The Netherlands: Sijthoff & Noordhoff, 1978, pp. 41–60.

[20] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 143–159, Jan. 2002.

[21] H. C. Lac and D. A. Stacey, "Feature subset selection via multi-objective genetic algorithm," in *Proc. Int. Joint Conf. Neural Netw.*, 2005, vol. 3, pp. 1349–1354.

[22] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 598–605.

[23] J. Li, M. T. Manry, P. L. Narasimha, and C. Yu, "Feature selection using a piecewise linear network," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1101–1115, Sep. 2006.

[24] H. Liu and H. Motoda*, Feature Selection for Knowledge Discovery and Data Mining*. Norwell, MA: Kluwer, 1998.

[25] J. Mao, K. Mohiuddin, and A. K. Jain, "Parsimonious network design and feature selection through node pruning," in *Proc. Int. Conf. Pattern Recognit.*, 1994, vol. 2, pp. 622–624.

[26] D. Michie, D. J. Spiegelhalter, and C. C. Taylor*, Machine Learning, Neural and Statistical Classification*. Chichester, U.K.: Ellin Horwood, 1994 [Online]. Available: http://www.amsta.leeds.ac.uk/~charles/statlog

[27] J. Moody and J. Utans, "Principled architecture selection for neural networks: Application to corporate bond rating prediction," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 683–690.

[28] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assessment," in *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, 1989, vol. 1, pp. 107–115.

[29] V. Onnia, M. Tico, and J. Saarinen, "Feature selection method using neural network," in *Proc. Int. Conf. Image Process.*, 2001, vol. 1, pp. 513–516.

[30] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, 1999.

[31] K. L. Priddy, S. E. Rogers, D. W. Ruck, and G. L. Tarr, "Bayesian selection of important features for feedforward neural networks," *Neurocomputing*, vol. 5, no. 2–3, pp. 91–103, 1993.

[32] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[33] R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Networks*, vol. 4, no. 5, pp. 740–747, Sep. 1993.

[34] B. D. Ripley, "Statistical ideas for selecting network architectures," in *Neural Networks: Artificial Intelligence and Industrial Applications*, B. Kappen and S. Gielen, Eds. London, U.K.: Springer-Verlag, 1995, pp. 183–190.

[35] E. Romero, J. M. Sopena, G. Navarrete, and R. Alquézar, "Feature selection forcing overtraining may help to improve performance," in *Proc. Int. Joint Conf. Neural Netw.*, 2003, vol. 3, pp. 2181–2186.

[36] D. W. Ruck, S. K. Rogers, and M. Kabrisky, "Feature selection using a multilayer perceptron," *J. Neural Netw. Comput.*, vol. 2, no. 2, pp. 40–48, 1990.

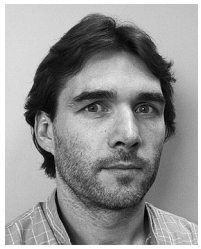
[37] R. Setiono and H. Liu, "Neural-network feature selector," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 654–662, May 1997.

[38] W. Siedlecki and J. Sklansky, "On automatic feature selection," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 2, no. 2, pp. 197–220, 1988.

[39] J. M. Sopena, E. Romero, and R. Alquézar, "Neural networks with periodic and monotonic activation functions: A comparative study in classification problems," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, 1999, vol. 1, pp. 323–328.

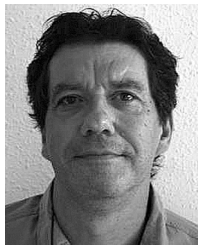
[40] A. Stahlberger and M. Riedmiller, "Fast network pruning and feature extraction using the Unit-OBS algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9, pp. 655–661.

[41] J. M. Steppe and K. W. Bauer, "Improved feature screening in feedforward neural networks," *Neurocomputing*, vol. 13, no. 1, pp. 47–58, 1996.
[42] J. M. Steppe, K. W. Bauer, and S. K. Rogers, "Integrated feature and architecture selection," *IEEE Trans. Neural Netw.*, vol. 7, no. 4, pp. 1007–1013, Jul. 1996.
[43] G. Taguchi, *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. New York: ASME, 1993.
[44] P. Van de Laar, T. Heskes, and S. Gielen, "Partial retraining: A new approach to input relevance determination," *Int. J. Neural Syst.*, vol. 9, no. 1, pp. 75–85, 1999.
[45] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[46] A. Verikas and M. Bacauskiene, "Feature selection with neural networks," *Pattern Recognit. Lett.*, vol. 23, no. 11, pp. 1323–1335, 2002.
[47] P. Zhang, B. Verma, and K. Kumar, "A neural-genetic algorithm for feature selection and breast abnormality classification in digital mammography," in *Proc. Int. Joint Conf. Neural Netw.*, 2004, vol. 3, pp. 2303–2308.
[48] J. M. Zurada, A. Malinowski, and S. Usui, "Perturbation method for deleting redundant inputs of perceptron networks," *Neurocomputing*, vol. 14, no. 2, pp. 177–193, 1997.



Enrique Romero received the B.Sc. degree in mathematics from the Universitat Autònoma de Barcelona, Barcelona, Spain, in 1989, and the B.Sc. and Ph.D. degrees in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1994 and 2004, respectively.

In 1996, he joined the Department of Llenguatges i Sistemes Informátics, UPC, as an Assistant Professor. His research interests include pattern recognition, neural networks, support vector machines, and feature selection.



Josep María Sopena received the B.S. and Ph.D. degrees in psychology from the Universitat de Barcelona, Barcelona, Spain, in 1980 and 1985, respectively.

Since 1986, he has been an Associate Professor at the Universitat de Barcelona. His research interests include machine learning, pattern recognition, neural networks, and computational linguistics.