An Experimental Study of Several Decision Issues for Feature Selection with Multi-Layer Perceptrons

E. Romero
Dept. de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
E-mail: eromero@lsi.upc.edu

J.M. Sopena
Lab. Neurocomputació
Universitat de Barcelona
E-mail: jsopena@ub.edu

***Abstract*—An experimental study of several decision issues for wrapper Feature Selection with Multi-Layer Perceptrons is presented, namely the stopping criterion, the data set where the saliency is measured and the network retraining before computing the saliency. Experimental results with the Sequential Backward Selection procedure 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. Regarding the stopping criterion and the data set where the saliency is measured, the procedure profits from measuring the saliency in a validation set, as reasonably expected. A somehow non-intuitive conclusion can be drawn by looking at the stopping criterion, where it is suggested that forcing overtraining may be as useful as early stopping.**

I. Introduction

Feature Selection (FS) plays an important role in many Supervised Machine Learning problems. In addition of reducing the storage requirements and the computational cost, FS may lead to the improvement of the generalization performance [7]. The problem of FS can be defined as follows: given a set of N_f features, select a subset that performs the best under a certain evaluation criterion. This definition leads to a search problem in a space of 2^{N_f} elements. Therefore, two components must be specified: the feature subset evaluation criterion and the search procedure through the space of feature subsets.

We will focus on FS with Multi-Layer Perceptrons (MLPs) within the *wrapper* approach [5]. Many FS algorithms for MLPs use, as feature subset evaluation criterion, different variations of the concept of *saliency* defined in some pruning methods [12]. A feature is considered more important whenever its saliency is larger, so that input units with small saliencies can be eliminated. Following the wrapper approach, the most commonly used saliency is the value of the loss function, usually the sum-of-squares error (SSE). Regarding the search procedure, most FS algorithms for MLPs use the Sequential Backward Selection (SBS) algorithm. 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 of 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 [7].

FS for MLPs with the SBS procedure and using the SSE as evaluation criterion involves taking a number of decisions, for which there are neither a commonly accepted criterion nor comparative studies. First, the stopping criterion of the training phase. Usually, networks are trained until a local minimum for the training set is found, although there are several exceptions, where an early stopping procedure is performed (see [17], [1], [19], for example). Second, the data set where the SSE should be measured. Many existing methods only use the training set to that end. Several exceptions use a validation or test set to compute the saliency (see [9], [4], [17], [19], [14], for example). Finally, whether or not the network should be retrained at every step with every feature temporarily removed before computing the saliency. To the best of our knowledge, the only models that retrain the network at every step with every feature temporarily removed/added before computing the saliency are those described in [16], [10], [14]. Among them, only the model presented in [14] is a pure SBS procedure.

An experimental study of the aforementioned decision issues when performing FS with MLPs and the SBS procedure is presented in this work. Experimental results indicate that the increase in the computational cost associated with retraining the network with every feature temporarily removed is rewarded with a significant performance improvement. This issue is shown to be critical, although, as previously mentioned, it has been hardly used in practice. Regarding the data set where the value of the SSE is measured, the SBS procedure for MLPs profits from measuring the SSE in a validation set, which is quite an intuitive idea. Instead, a somehow non-intuitive conclusion is drawn by looking at the stopping criterion, where forcing overtraining is shown to be potentially as useful as early stopping.

A significant improvement in the overall results with respect to learning with the whole set of variables is observed, which

[1] There is no commonly accepted definition of the relevance of a variable (see [3], [7], for example). Given a data set, we consider that a variable is irrelevant for a Supervised Machine Learning system when its optimal performance is not affected negatively by the absence of that variable ([7], page 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.

compares favorably with other existing FS wrappers in the literature.

The rest of the paper is organized as follows. A basic SBS scheme for MLPs and its decision issues are discussed in section II. The experimental work can be found in section III. Finally, section IV outlines some directions for further research.

II. Decision Issues in a Basic SBS Scheme for MLPs

A basic SBS scheme for MLPs using the SSE as the saliency of a feature is presented in figure 1. The outer loop follows the scheme of the classical SBS procedure, where after a training process a feature is permanently eliminated at every step. The inner loop selects the variable to eliminate: every feature is temporarily removed, the network is optionally retrained until a certain stopping criterion is met, and then the value of the SSE is computed (on a certain data set). The variable corresponding to the lowest value of the SSE is permanently eliminated. The algorithm in figure 1 involves three decision issues, as explained next.

The first decision issue is the stopping criterion in the training phase. Two different stopping criteria were tested. The first one is to stop where a minimum of the SSE for a validation set is achieved. The second one is to train until a minimum for the training 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. But it could also be argued that overtraining the network until a local minimum of the SSE for the training set forces the system to use all the available variables as much as possible. In this situation, irrelevant variables could be more outstanding when the system is not allowed to use them [14].

The second decision issue is the data set where the SSE is measured. The measurement of the SSE in a validation set is, probably, the most reasonable choice, since it can be considered as an estimator of the generalization error. But selecting the minimum number of features that allows to fit the training set as well as the whole set of variables does could also be thought as a quite reasonable way to obtain a good feature subset. In this case, the SSE should be measured in the training set. Both schemes were tested.

The third decision issue involves whether the network is retrained or not after the feature is temporarily removed and before computing the saliency. With this idea, 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 involves training $N_f(N_f+1)/2$ networks (in this case, the training prior to the inner loop can be omitted).

```
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, and keep its generalization
      performance (decision issue, see text for details)
    for each $v \in V_N$ do
      Set $V = V_N - \{v\}$
      Optionally, train the network with the features in $V$
        (decision issue, see text for details)
      Obtain the saliency of $v$ by computing the value of
        the sum-of-squares error function $E_v$ on a certain
        data set (decision issue, see text for details)
    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 at any step
    of the previous loop
end Algorithm
```

Fig. 1. A basic SBS procedure for MLPs and the SSE as the saliency.

Note that these two ways of computing the saliency may yield very different results for the same feature, since the corresponding output functions of the trained networks may be very different as well. Both possibilities were tested.

In summary, there are three combinations of stopping criterion/SSE measurement data set: Training/Training, Training/Validation and Validation/Validation (the Validation/Training combination makes no sense):

1) Training/Training: The network is trained until a minimum of the SSE for the training set, where the saliency is computed. Therefore, variables that are not necessary to fit the training set will be removed.
2) Training/Validation: The network is trained until a minimum of the SSE for the training set is achieved (probably overtrained). The system is forced to use all the available variables as much as possible. In this situation, a validation set is used to remove the variables.
3) Validation/Validation: The network is trained until a minimum of the SSE for a validation set is obtained. The saliency is also computed in the validation set.

Combined with the two possibilities regarding the network retraining, there is a total of six configurations to be tested and compared.

III. Experiments

Some experiments on both artificial and benchmark classification data sets were performed. For every data set, the six aforementioned configurations were tested with the SBS procedure for MLPs described in figure 1.

TABLE I

TEST SET RESULTS AND SELECTED VARIABLES FOR THE *Augmented Two Spirals* DATA SET FOR DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION/SSE MEASUREMENT DATA SET IN THE SBS PROCEDURE.

Retrain	Stopping Criterion/SSE Measurement	Test	MSE	SVar
Yes	Training/Validation	99.89%	0.01	x_1 x_6 x_7
Yes	Validation/Validation	99.66%	0.02	x_6 x_1 x_2
Yes	Training/Training	93.76%	0.19	x_4 x_5 x_3 x_2
No	Training/Validation	92.30%	0.25	x_2 x_6 x_7 x_4 x_8 x_3
No	Training/Training	92.10%	0.24	x_6 x_2 x_7 x_8 x_9 x_4 x_3
No	Validation/Validation	87.10%	0.39	x_6 x_9 x_7 x_4 x_8 x_3

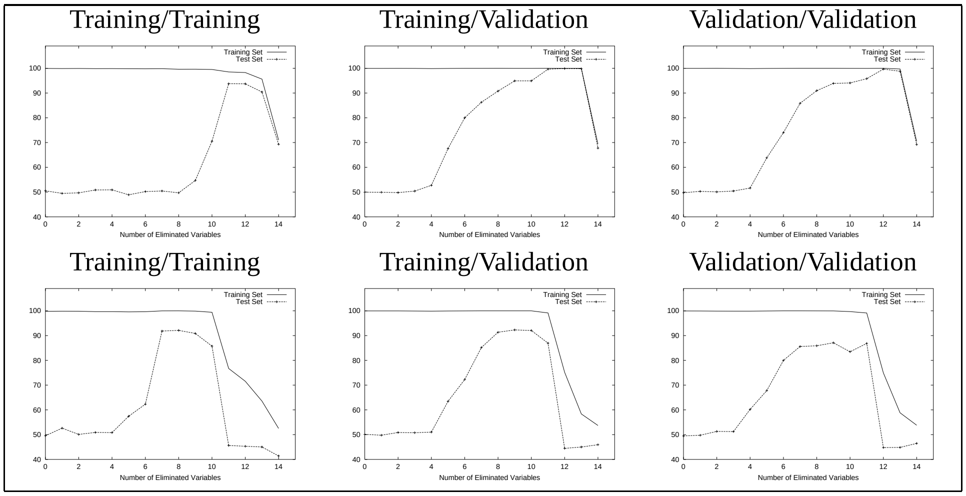

Fig. 2. Percentage of correct examples for the *Augmented Two Spirals* data set in the training and test sets with respect to the number of eliminated variables in the SBS procedure for MLPs with retraining (top) and without it (bottom).

A. Experimental Setting

All the experiments were performed with stratified Cross-Validation (CV). Previously to every CV, the examples in the data set were randomly shuffled. For the Training/Training configuration, 5 runs of a 5-fold CV were conducted (the folds that are not in the training set were used as test data), giving a total of 25 runs for each training step. For the Training/Validation and Validation/Validation configurations, 5 runs of a double 5-4-fold CV [13] were performed as follows. A 5-fold CV (the outer CV) is performed to obtain 5 folds (4 folds to "learn" and 1 fold to test). Then, the 4 folds of the "learning set" of the outer CV were used as follows: 3 folds to train and 1 fold to validate, as in a 4-fold CV (the inner CV). Therefore, the number of trained models in every double 5-4-fold CV was 20, giving a total of 100 runs for each training step.

A computational cost as small as possible for the whole process was required. As pointed out in [15], 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 Back-Propagation (BP). Therefore, we used MLPs with one hidden layer of sinusoidal units, the hyperbolic tangent in the output layer, and trained with standard BP in pattern mode.

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

B. The Augmented Two Spirals Data Set

An augmented version of the well-known *Two Spirals* data set was constructed, where 13 irrelevant features were artificially added to the two original variables. Some of them were noisy. The whole set of variables was defined as

$$\begin{aligned}\{&x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2, x_1 + x_2, x_1 - x_2,\\ &x_1^2 + \mathcal{N}(0,1), x_2^2 + \mathcal{N}(0,1), x_1 \cdot x_2 + \mathcal{N}(0,1),\\ &x_1 + x_2 + \mathcal{N}(0,1), x_1 - x_2 + \mathcal{N}(0,1),\\ &\mathcal{U}(0,1), \mathcal{N}(0,1), \mathcal{N}(0,5)\}\end{aligned}$$

where (x_1, x_2) are the original features in the data set, and $\mathcal{N}$ and $\mathcal{U}$ are the normal and the uniform distributions, respectively. Each original training, validation and test sets comprise 194 two-dimensional points with balanced classes. In order to perform the experiments with CV, the three data sets were joined into a single data set.

The results are shown in table I (column 'Test') as the average percentage of correctly classified patterns on the respective test sets in the following trained networks:

1) For the Training/Training configuration, the networks with minimum test set error among the networks with minimum training set error after every variable is permanently eliminated.

TABLE II

DESCRIPTION OF THE BENCHMARK DATA SETS. THE COLUMN 'NVAR' SHOWS THE NUMBER OF VARIABLES AND THE 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', 'SAMPLING', 'TEST' AND 'SOURCE'). COLUMN 'ML ALG.' INDICATES THE MACHINE LEARNING ALGORITHM USED.

Data Set	**NVar**	**NExa**	**ML Alg**	**Sampling**	**Test**	**Source**
Hepatitis	19	155	MLP+BP	10-fold CV	82.1%	[8]
			DistAl	10-fold CV	84.7%	[20]
Ionosphere	33	351	MLP+BP	10-fold CV	90.3%	[11]
			DistAl	10-fold CV	94.3%	[20]
Sonar	60	208	MLP+BP	10-fold CV	83.4%	[11]
			DistAl	10-fold CV	83.0%	[20]

TABLE III

RESULTS OBTAINED IN [20] FOR THE BENCHMARK DATA SETS USED IN THIS WORK AFTER THE APPLICATION OF A WRAPPER FS PROCEDURE. THE FINAL NUMBER OF SELECTED VARIABLES CAN BE SEEN IN COLUMN 'NVAR'.

Data Set	**Search**	**ML Alg**	**Sampling**	**Test**	**NVar**	**Source**
Hepatitis	Genetic	DistAl	10-fold CV	88.7%	10	[20]
Ionosphere	Genetic	DistAl	10-fold CV	96.0%	13	[20]
Sonar	Genetic	DistAl	10-fold CV	85.5%	28	[20]

2) For the Training/Validation and Validation/Validation configurations, the networks with minimum test set error among the networks with minimum validation set error after every variable is permanently eliminated.

The mean SSE on the test set (column 'MSE') and the variables that allowed to obtain these results (column 'SVar') are also shown. Values in table I are computed as the mean over the different folds in the respective CV. Figure 2 shows, for every configuration, the evolution of the percentage of correct examples in the training and test sets with respect to the number of eliminated variables.

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 (see figure 2). 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 may be the relatively small number of examples in the data set, that did not allow to filter this kind of features. As far as variables are eliminated, performance improves. However, many variables must be eliminated to obtain good performance.

Retraining the network with every feature temporarily removed before computing the saliency has a positive effect on the SBS procedure for MLPs, both for the number of selected features and the overall performance. When retraining is present, it seems that the SBS procedure uses the validation set to construct a better subset of variables than if only the training set is used, although there is no significant difference regarding the stopping criterion (note that the Training/Validation configuration with retraining obtains similar results to the Validation/Validation one). The observed results can be explained by looking at the behavior of every configuration during the SBS procedure, as explained next.

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. This happens because the obtained solution after the training process uses all the variables in a significant way. In addition, noise-free variables (from x_1 to x_7) do not seem to be used more than the rest in order to learn the data set. Therefore, the elimination of a feature is decided in quite a random way.

When retraining is present,

1) In the Training/Training configuration, the training set can be almost perfectly learned independently of the temporarily eliminated variable. Therefore, no significant difference can be stated among the variables. This behavior was observed for the first 9 or 10 steps in all the experiments performed. Therefore, and similar to the case of absence of retraining, the permanent elimination of a variable is decided in quite a random way during (too) many steps.
2) For the Training/Validation and Validation/Validation configurations, in contrast, the variables are much more clearly differentiated from the beginning of the SBS procedure. The criterion to eliminate a variable permanently does not seem random.

Surprisingly, the evolution of the training set error when retraining is present is also better with a validation set than without it (see figure 2). A common aspect of the configurations that do not obtain satisfactory results is the fact that they consider variables x_3 and x_4 (the squared of the original variables) as important. These variables do not seem the most promising ones for this problem. Although they allow, together with other ones, to fit the training set, those feature subsets are not good for generalization purposes.

C. Experiments on Benchmark Data Sets

In this section, the experiments on several benchmark data sets with the SBS procedure for MLPs described in figure

TABLE IV

TEST SET RESULTS AND NUMBER OF SELECTED VARIABLES FOR THE *Hepatitis* DATA SET FOR DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION/SSE MEASUREMENT DATA SET IN THE SBS PROCEDURE.

Retrain	**Stopping Criterion/SSE Measurement**	**Test**	**Mse**	**NVar**
Yes	Training/Validation	93.90%	0.24	3
Yes	Validation/Validation	93.77%	0.25	3
Yes	Training/Training	92.26%	0.25	3
No	Training/Training	92.13%	0.26	3
No	Validation/Validation	88.97%	0.40	1
No	Training/Validation	88.26%	0.36	3

TABLE V

TEST SET RESULTS AND NUMBER OF SELECTED VARIABLES FOR THE *Ionosphere* DATA SET FOR DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION/SSE MEASUREMENT DATA SET IN THE SBS PROCEDURE.

Retrain	**Stopping Criterion/SSE Measurement**	**Test**	**Mse**	**NVar**
Yes	Training/Validation	93.61%	0.22	5
No	Validation/Validation	92.77%	0.24	5
Yes	Validation/Validation	92.73%	0.24	5
No	Training/Validation	92.57%	0.24	5
No	Training/Training	92.40%	0.25	5
Yes	Training/Training	90.51%	0.33	3

TABLE VI

TEST SET RESULTS AND NUMBER OF SELECTED VARIABLES FOR THE *Sonar* DATA SET FOR DIFFERENT CONFIGURATIONS OF RETRAINING/STOPPING CRITERION/SSE MEASUREMENT DATA SET IN THE SBS PROCEDURE.

Retrain	**Stopping Criterion/SSE Measurement**	**Test**	**Mse**	**NVar**
Yes	Validation/Validation	89.73%	0.33	14
Yes	Training/Training	88.59%	0.37	7
Yes	Training/Validation	87.95%	0.36	11
No	Training/Training	87.41%	0.40	57
No	Training/Validation	85.02%	0.46	46
No	Validation/Validation	84.49%	0.47	50

1 are shown. Three data sets from the UCI repository [2] were selected, namely *Hepatitis*, *Ionosphere* and *Sonar*. A brief description of these data sets can be found in table II, together with several results found in the literature from settings in which the whole set of features was used. In table II, MLP+BP means “Multi-Layer Perceptrons trained with Back-Propagation” and DistAl is a constructive learning algorithm for Neural Networks specific for classification problems [20]. The key idea behind DistAl is to add hidden units with a hyperspherical Radial Basis Function (RBF) based on a greedy strategy which ensures that the new hidden unit correctly classifies a maximal subset of training patterns belonging to the same class. Additionally, table III shows several results found in the literature after the application of an FS procedure on these data sets. We did not find more references (for wrapper approaches and with a similar experimental setting to that performed in this work) than those showed in the table. However, given the good results of DistAl with the whole set of features (see table II) and the good behavior of genetic algorithms for FS [6], the results in table III can be considered as a useful reference for comparison purposes. The search procedure used in [20] was a genetic algorithm where the fitness function for a given feature subset is computed as the mean percentage of correctly classified patterns on the test sets of a 10-fold CV trained with DistAl.

Our results are shown in tables IV to VI. Similar to the *Augmented Two Spirals* data set, the best results are always obtained retraining the network with every feature temporarily removed before computing the saliency. Therefore, the increase in the computational cost associated with this scheme is rewarded with a significant performance improvement. This issue is shown to be critical, although, as previously mentioned, it has been hardly used in practice.

Regarding the stopping criterion/SSE measurement data set, the SBS procedure seems to profit from measuring the SSE in a validation set, although it is unclear which is the best stopping criterion. For the *Sonar* problem the Validation/Validation strategy appears to work best. For the *Ionosphere* problem, in contrast, the Training/Validation strategy selects a better subset of variables. For the *Hepatitis* problem both configurations can be considered as equivalent.[2] The goodness of the Val-

[2] For the *Ionosphere* problem, the variables selected by the Training/Validation configuration were $\{x_2, x_4, x_5, x_7, x_{20}\}$. The rest of configurations with 5 features selected $\{x_2, x_4, x_7, x_{20}, x_{26}\}$. For the *Hepatitis* problem, the variables selected by the Training/Validation and Validation/Validation configuration were $\{x_2, x_5, x_{18}\}$. The rest of configurations with 3 features selected $\{x_{13}, x_{15}, x_{18}\}$, $\{x_8, x_{13}, x_{18}\}$ or $\{x_{13}, x_{17}, x_{18}\}$.

idation/Validation configuration can be intuitively explained, since it tries to obtain the best possible generalization results at every step. The Training/Validation configuration, in contrast, improves performance by forcing overtraining (and measuring the SSE in a validation set). This is a non-intuitive result. The explanation pointed out in [14] is that forcing the system to use all the available features as much as possible helps to detect irrelevant variables.

Although in a different scale, a similar behavior to that of the the *Augmented Two Spirals* data set was observed. First, the training set can be fitted with a much smaller subset of features than the original one. Regarding the test set, performance improves until a subset of features remains for which the elimination of further variables results 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 they do not provide good generalization. The number of examples may not be large enough to filter these variables in some cases.

Finally, we can appreciate an important improvement in the overall results with respect to learning with the whole set of variables (see Table II) and compared with existing FS wrappers in the literature (see Table III).[3] An important reduction in the final number of selected variables is also observed. The good results obtained with the Validation/Validation and Training/Validation with retraining configurations are mainly due, in our opinion, to a proper detection of irrelevant variables.

IV. Future Work

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 first steps of the algorithm, when probably many irrelevant variables still remain, take most of the computational time. Several heuristics could 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 would start.

The results provided in this work were obtained with standard BP and sinusoidal hidden units, but the basic scheme presented in this work can be tested within any other framework which can be adjusted to the required specifications. In particular, the SBS procedure in figure 1 could be performed with Support Vector Machines [18] using some function of the margin as saliency and different hardness of the margin as the stopping criterion.

[3] The results obtained for the *Ionosphere* data set could be seen as unsatisfactory when compared with those obtained in [20]. In contrast, they can be considered as very satisfactory when compared with those obtained by MLP models. It is worth noting that RBF networks allow to obtain better solutions than MLP ones for this problem (see Table II). Therefore, FS models based on RBF units, such as DistAl, are expected to obtain excellent results.

Acknowledgements

This work was supported by Consejo Interministerial de Ciencia y Tecnología (CICYT), under projects DPI2002-03225 and CGL2004-04702-C02-02.

References

[1] B. Baesens, S. Viaene, J. Vanthienen, and G. Dedene, “Wrapped Feature Selection by means of Guided Neural Network Optimisation,” in *International Conference on Pattern Recognition*, vol. 2, 2000, pp. 113–116.

[2] C. L. Blake and C. J. Merz, “UCI Repository of Machine Learning Databases,” 1998, university of California, Irvine, Department of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[3] A. L. Blum and P. Langley, “Selection of Relevant Features and Examples in Machine Learning,” *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997, special Issue on Relevance.

[4] T. Cibas, F. F. Soulié, P. Gallinari, and Š. Raudys, “Variable Selection with Optimal Cell Damage,” in *International Conference on Artificial Neural Networks*, vol. 1, 1994, pp. 727–730.

[5] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” in *11th International Conference on Machine Learning*, 1994, pp. 121–129.

[6] M. Kudo and J. Sklansky, “Comparison of Algorithms that Select Features for Pattern Classifiers,” *Pattern Recognition*, vol. 33, no. 1, pp. 25–41, 2000.

[7] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

[8] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, “*Machine Learning, Neural and Statistical Classification*,” 1994, ellin Horwood. Results available at http://www.phys.uni.torun.pl/kmk/projects/datasets-stat.html.

[9] J. Moody and J. Utans, “Principled Architecture Selection for Neural Networks: Application to Corporate Bond Rating Prediction,” in *Advances in Neural Information Processing Systems*, vol. 4. Morgan Kaufmann, 1992, pp. 683–690.

[10] V. Onnia, M. Tico, and J. Saarinen, “Feature Selection Method using Neural Network,” in *International Conference on Image Processing*, vol. 1, 2001, pp. 513–516.

[11] D. Opitz and R. Maclin, “Popular Ensemble Methods: An Empirical Study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[12] R. Reed, “Pruning Algorithms - A Survey,” *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, 1993.

[13] B. D. Ripley, “Statistical Ideas for Selecting Network Architectures,” in *Neural Networks: Artificial Intelligence and Industrial Applications*, B. Kappen and S. Gielen, Eds. Springer-Verlag, London, 1995, pp. 183–190.

[14] E. Romero, J. M. Sopena, G. Navarrete, and R. Alquézar, “Feature Selection Forcing Overtraining May Help to Improve Performance,” in *International Joint Conference on Neural Networks*, vol. 3, 2003, pp. 2181–2186.

[15] J. M. Sopena, E. Romero, and R. Alquézar, “Neural Networks with Periodic and Monotonic Activation Functions: A Comparative Study in Classification Problems,” in *9th International Conference on Artificial Neural Networks*, vol. 1, 1999, pp. 323–328.

[16] J. M. Steppe, K. W. Bauer, and S. K. Rogers, “Integrated Feature and Architecture Selection,” *IEEE Transactions on Neural Networks*, vol. 7, no. 4, pp. 1007–1013, 1996.

[17] P. Van de Laar, T. Heskes, and S. Gielen, “Partial Retraining: A New Approach to Input Relevance Determination,” *International Journal of Neural Systems*, vol. 9, no. 1, pp. 75–85, 1999.

[18] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, NY, 1998.

[19] A. Verikas and M. Bacauskiene, “Feature Selection with Neural Networks,” *Pattern Recognition Letters*, vol. 23, no. 11, pp. 1323–1335, 2002.

[20] J. Yang and V. Honavar, “Feature Subset Selection using a Genetic Algorithm,” in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Kluwer Academic Publishers, 1998, pp. 117–136.