ELSEVIER

Pattern Recognition Letters 23 (2002) 1323–1335

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Feature selection with neural networks

A. Verikas [a,b,*], M. Bacauskiene [b]

[a] Intelligent Systems Laboratory, Halmstad University, Box 823, S 301 18 Halmstad, Sweden
[b] Department of Applied Electronics, Kaunas University of Technology, LT-3031, Kaunas, Lithuania

Received 9 May 2001; received in revised form 5 November 2001

Abstract

We present a neural network based approach for identifying salient features for classification in feedforward neural networks. Our approach involves neural network training with an augmented cross-entropy error function. The augmented error function forces the neural network to keep low derivatives of the transfer functions of neurons when learning a classification task. Such an approach reduces output sensitivity to the input changes. Feature selection is based on the reaction of the cross-validation data set classification error due to the removal of the individual features. We demonstrate the usefulness of the proposed approach on one artificial and three real-world classification problems. We compared the approach with five other feature selection methods, each of which banks on a different concept. The algorithm developed outperformed the other methods by achieving higher classification accuracy on all the problems tested.

Keywords: Classification; Neural network; Feature selection; Regularization

1. Introduction

The pattern recognition problem is traditionally divided into the stages of feature extraction and classification. Feature extraction aims to find a mapping that reduces the dimensionality of the patterns being classified. The mapping found projects the N-dimensional data onto the M-dimensional space, where $M < N$. Feature selection is a special case of feature extraction. Employing feature extraction all N measurements are used for obtaining the M-dimensional data. Therefore, all N measurements need to be obtained. Feature selection, by contrast, enables us to discard $(N - M)$ irrelevant features. Hence, by collecting only relevant attributes, the cost of future data collecting may by reduced.

A large number of features can be usually measured in many pattern recognition applications. Not all of the features, however, are equally important for a specific task. Some of the variables may be redundant or even irrelevant. Usually better performance may be achieved by discarding such variables (Fukunaga, 1972; Mucciardi and

* Corresponding author. Tel.: +46-35-167-140; fax: +46-35-216-724.
E-mail addresses: antanas.verikas@ide.hh.se (A. Verikas), marija.bacauskiene@eaf.ktu.lt (M. Bacauskiene).

0167-8655/02/$ - see front matter
PII: S0167-8655(02)00081-8

Gose, 1971; Steppe et al., 1996). Moreover, as the number of features used grows, the number of training samples required grows exponentially (Duda and Hart, 1973). Therefore, in many practical applications we need to reduce the dimensionality of the data.

The principal component analysis (PCA) (Bishop, 1995; Fukunaga, 1972) and the linear discriminant analysis (Fukunaga, 1972) are two traditional techniques of feature extraction. These techniques attempt to reduce the dimensionality of the data by creating new features that are linear combinations of the original ones.

Feature selection in general is a difficult problem. In a general case, only an exhaustive search can guarantee an optimal solution. The branch and bound algorithm (Narendra and Fukunaga, 1977) can also guarantee an optimal solution, if the monotonicity constraint imposed on a criterion function is fulfilled. The branch and bound based optimization has been used for feature selection by several authors (Fortoutan and Sklansky, 1987; Ichino and Sklansky, 1984). A large variety of feature selection techniques that result in a suboptimal feature set have been proposed (Jain and Zongker, 1997; Kittler, 1986; Mucciardi and Gose, 1971), ranging from the sequential forward and backward selection (Mucciardi and Gose, 1971) to the sequential forward floating selection characterized by a dynamically changing number of features included or eliminated at each step (Pudil et al., 1994). Though not numerous, techniques for feature selection based on the fuzzy set theory have also been proposed (De et al., 1997; Pal, 1999; Pal et al., 2000).

Neural networks have proved themselves to be a powerful tool in a variety of pattern recognition applications. The use of neural networks for feature extraction or selection seems promising, since the ability to solve a task with a smaller number of features is evolved during training by integrating the processes of learning, feature extraction, feature selection, and classification. However, there are very few established procedures for extracting features with neural nets (Lotlikar and Kothari, 2000).

Feature selection with neural nets can be thought of as a special case of architecture pruning (Reed, 1993), where input features are pruned, rather than hidden neurons or weights. Pruning procedures extended to the removal of input features have been proposed in (Belue and Bauer, 1995; Cibas et al., 1996), where the feature selection process is usually based on some saliency measure aiming to remove less relevant features. However, since most of the procedures evaluate the saliency of features during the training process, they strictly depend on the learning algorithm employed.

Zurada et al. (1997) have recently proposed a saliency measure based feature selection method for regression. The authors assume that the trained network provides a continuous differentiable mapping. This assumption and the Jacobian matrix based saliency measure, which is derived from the approximate neural network mapping over the training set, allow application of the procedure directly to a trained network without multiple training runs.

An approach based on a formal hypothesis test for testing the statistical significance of a q-dimensional subset of weights has also been proposed for feature selection (Steppe et al., 1996). An inter- and intra-cluster scatter analysis based technique to select features for the radial basis function networks has recently been proposed (Basak and Mitra, 1999).

In this paper, we propose to add a term constraining the derivatives of the neural network output and hidden nodes transfer functions to the cross-entropy error cost function. The network is trained by minimizing such an extended cost function. Feature selection is based on the reaction of the cross-validation data set classification error due to the removal of the individual features. The rest of the paper is organized as follows. To clarify notations, Section 2 presents a description of the neural network used. A brief description of competing feature ranking techniques and the analysis of the shortcomings of the weights-based feature saliency measures and feature selection procedures are given in Section 3. Section 4 describes the feature selection procedure proposed. The results of the experimental investigations are presented in Section 5. Finally, Section 6 presents conclusions of the work.

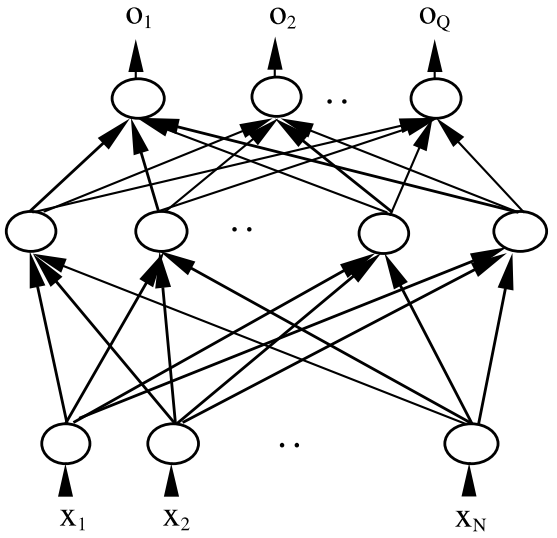

Fig. 1. A feedforward neural network.

2. The neural network used

Let us consider a fully connected feedforward neural network, as shown in Fig. 1. Let $o_j^{(q)}$ denote the output signal of the jth neuron in the qth layer and $w_{ij}^{(q)}$ the connection weight coming from the ith neuron in the $(q-1)$ layer to the jth neuron in the qth layer. Then $o_j^{(q)} = f(net_j^{(q)})$ and $net_j^{(q)} = \sum_{i=0}^{n_{q-1}} w_{ij}^{(q)} o_i^{(q-1)}$, where $net_j^{(q)}$ stands for the activation level of the neuron, n_{q-1} is the number of neurons in the $q-1$ layer and $f(net)$ is the sigmoid activation function given by $f(net) = 1/(1+\exp(-net))$.

When given an augmented input vector $\mathbf{x} = [1, x_1, x_2, \ldots, x_N]^t$ in the input (0th) layer, the output signal of the jth neuron in the output (Lth) layer is given by

$$o_j^{(L)} = f\left(\sum_m w_{mj}^{(L)} f\left(\ldots f\left(\sum_i w_{iq}^{(1)} x_i \right) \ldots \right) \right). \tag{1}$$

3. Competing feature selection techniques

We compare the proposed neural network based feature selection approach with five other methods, each of which banks on different concept, namely, the neural-network feature selector (NNFS) based on elimination of input layer weights (Setiono and Liu, 1997), the weights-based feature saliency measure (signal-to-noise ratio (SNR) based technique) (Bauer et al., 2000), the neural network output sensitivity based feature saliency measure (De et al., 1997), the fuzzy entropy (De et al., 1997), and the discriminant analysis (the criterion used is proposed in this paper). Next we briefly describe the methods used for the comparisons and discuss some shortcomings of the weights-based feature saliency measures and feature selection procedures.

3.1. Neural-network feature selector

To force the training process to result in weights manifesting larger differences between the values of weights connected to the relevant features and the useless ones, the NNFS is trained by minimizing the cross-entropy error function augmented with the additional term given by Eq. 2 (Setiono and Liu, 1997).

$$R_1(w) = \varepsilon_1 \left\{ \sum_{i=1}^{N} \sum_{j=1}^{n_h} \frac{\beta (w_{ij})^2}{1+\beta (w_{ij})^2} \right\} + \varepsilon_2 \left\{ \sum_{i=1}^{N} \sum_{j=1}^{n_h} (w_{ij})^2 \right\} \tag{2}$$

with w_{ij} being the weight between the ith input feature and the jth hidden node, n_h is the number of the hidden nodes, N is the number of features, and the constants ε_1, ε_2 and β have to be chosen experimentally.

Feature selection is based on the reaction of the cross-validation data set classification error due to the removal of the individual features. For our comparisons we use the results presented in (Setiono and Liu, 1997).

The second part of the term $R_1(w)$ is exactly the weight-decay term, except that only input to hidden weights are constrained. Weights connected to unimportant features should attain values near zero during the learning process. The first term of the function $R_1(w)$ can be considered as a measure

of the total number of nonzero input weights in the network.

However, concerning feature selection, weight-decay possesses the following drawback. A simple weight decay algorithm tries to get smaller weights. Smaller weights usually result in smaller inputs to neurons and larger sigmoid derivatives in general. Therefore, output sensitivity to the input increases. This drawback can be clearly observed from the tables presented in Section 5 (Tables 2, 4, and 6). Analyzing the classification results presented in the tables for the NNFS (Setiono and Liu, 1997) for the ⟨All Features⟩ case we observe the large difference between the classification accuracies achieved for the training and testing sets. The much lower accuracy obtained for the testing set points out that the output sensitivity to the input changes is high.

For the purpose of classification, by contrast, we need low sensitivity of output to the input. Hence, it seems reasonable to constrain the derivatives of the transfer functions of neurons instead of input layer weights during training. By constraining the derivatives we can force neurons to work in the saturation region. Therefore, the low sensitivity of output to the input can be obtained with relatively large values of weights.

3.2. Signal-to-noise ratio based technique

A significant number of feature saliency measures used for neural network based feature selection are weights-based (Bauer et al., 2000; Cibas et al., 1996; Steppe and Bauer, 1996), or neural network's output sensitivity based exemplified by Eq. 3 (Belue and Bauer, 1995; Priddy et al., 1993; Steppe and Bauer, 1996; Zurada et al., 1997):

$$\Lambda_{1i} = \sum_{j=1}^{n_L} \sum_{p=1}^{P} \sum_{k \neq j} \sum_{x_i \in D_i} \left| \frac{\partial o_{kp}^{(L)}}{\partial x_i} \right| \tag{3}$$

with n_L being the number of the output nodes, D_i is a set of sampled values of x_i, P is the number of training samples, and j and k are indices of the output nodes.

The weights-based feature saliency measures bank on the idea that weights connected to important features attain large absolute values while weights connected to unimportant features would probably attain values somewhere near zero.

However, a saliency measure alone does not indicate how many of the candidate features should be used. Therefore, some of feature selection procedures are based on making comparisons between the saliency of a candidate feature and the saliency of a noise feature (Bauer et al., 2000; Priddy et al., 1993; Steppe and Bauer, 1996).

The SNR based feature ranking technique proposed in (Bauer et al., 2000) exemplifies the use of a noise feature as the reference. Feature ranking is based on the feature saliency measure given by

$$\Lambda_{2i} = 10 \operatorname{Log}_{10} \left(\frac{\sum_{j=1}^{n_h} (w_{ij})^2}{\sum_{j=1}^{n_h} (w_{Ij})^2} \right) \tag{4}$$

with w_{Ij} being the weight from the injected noise feature I to the jth hidden node.

The number of features to be chosen is identified by the significant decrease of the classification accuracy of the test data set when eliminating a feature. The authors have demonstrated that the technique is competitive with the method proposed by Setiono and Liu (1997).

3.3. Neural network output sensitivity based feature ranking

After the multilayer perceptron learns a data set, a feature quality index (FQI_q) is computed for every feature q and then the features are ranked according to (FQI_q) (De et al., 1997). The computation of (FQI_q)s proceeds as follows. For each training data point $\mathbf{x}_i$ ($i = 1, 2, \ldots, P$), x_{iq} is set to zero. If $\mathbf{x}_i^{(q)}$ denotes this modified data point, then $x_{ij}^{(q)} = x_{ij}$ $\forall j \neq q$ and $x_{iq}^{(q)} = 0$. Let $\mathbf{o}_i$ and $\mathbf{o}_i^{(q)}$ denote the output vectors obtained from the MLP after the presentation of $\mathbf{x}_i$ and $\mathbf{x}_i^{(q)}$, respectively. Now the output vectors $\mathbf{o}_i$ and $\mathbf{o}_i^{(q)}$ are not expected to differ much, if feature q is not important. The feature quality index (FQI_q) is defined as

$$\mathrm{FQI}_q = \sum_{j=1}^{P} \|\mathbf{o}_j - \mathbf{o}_j^{(q)}\|^2. \tag{5}$$

The larger the index, the more important the feature is.

3.4. Fuzzy entropy based feature ranking

Let

$$\widetilde{A} = \{\mu_{\widetilde{A}}(x_i)/x_i \,|\, x_i \in X;\ i = 1, \ldots, P;\ \mu_{\widetilde{A}} \in [0,1]\} \tag{6}$$

be a fuzzy set defined on a universe of discourse $X = \{x_1, x_2, \ldots, x_P\}$, where $\mu_{\widetilde{A}}(x_i)$ is the membership of x_i to $\widetilde{A}$. Then entropy of Deluca–Termini of the fuzzy set $\widetilde{A}$ is defined as (Yager and Zadeh, 1994)

$$H(\widetilde{A}) = \frac{1}{P \ln 2} \sum_{i=1}^{P} \Big[-\mu_{\widetilde{A}}(x_i) \ln(\mu_{\widetilde{A}}(x_i)) - (1 - \mu_{\widetilde{A}}(x_i)) \ln(1 - \mu_{\widetilde{A}}(x_i)) \Big]. \tag{7}$$

The standard S-functions can be used for modelling μ: $\mu_{\widetilde{A}}(x_i) = S(x_i; a, b, c)$ (Pal and Rosenfeld, 1988).

A class C_j can be considered as a fuzzy set and then entropy H_{qj} of the class for the qth feature can be computed. The greater the tendency of the data points from the class C_j to cluster around mean value of the qth feature, the higher would be the value of H_{qj}. If we pool the classes C_j and C_k together, the value of H_{qjk} for the pooled cluster would decrease as the separation power of the qth feature increases, since for a good feature for most of the data points $\mu(x_q) \approx 0$ or 1. Based on these observations, the following overall feature evaluation index (OFEI) was proposed (De et al., 1997):

$$\mathrm{OFEI}_q = \frac{\sum_{j,k=1, j\neq k}^{Q} H_{qjk}}{\sum_{j=1}^{Q} H_{qj}}, \tag{8}$$

where Q is the number of classes. It is assumed that the lower the value of OFEI, the better the feature is.

3.5. Discriminant analysis based feature ranking

Let $\mathbf{m}_j$ denote the sample mean vector of the jth class

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{k=1}^{N_j} \mathbf{x}_{jk}, \tag{9}$$

where N_j is the number of samples in the jth class. Similarly, $\mathbf{m}$ denotes the mixture sample mean

$$\mathbf{m} = \sum_{j=1}^{Q} P_j \mathbf{m}_j \tag{10}$$

with P_j being a priori probability of the jth class. We can now define a within class–covariance and between class–covariance matrices $\mathbf{S}_w$ and $\mathbf{S}_b$, respectively.

$$\mathbf{S}_w = \sum_{j=1}^{Q} P_j \frac{1}{N_j} \sum (\mathbf{x}_{jk} - \mathbf{m}_j)(\mathbf{x}_{jk} - \mathbf{m}_j)^{\mathrm{t}}, \tag{11}$$

$$\mathbf{S}_b = \sum_{j=1}^{Q} P_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^{\mathrm{t}}. \tag{12}$$

Using $\mathbf{S}_w$ and $\mathbf{S}_b$, we form the following criterion function $J_i(\mathbf{x})$ for feature ranking:

$$J_i(\mathbf{x}) = \frac{\mathrm{tr}(\mathbf{S}_b)}{\mathrm{tr}(\mathbf{S}_w)} - \frac{\mathrm{tr}_{X\setminus i}(\mathbf{S}_b)}{\mathrm{tr}_{X\setminus i}(\mathbf{S}_w)}, \tag{13}$$

where $\mathbf{x}$ expresses the dependence of the criterion function on the data set and $\mathrm{tr}_{X\setminus i}(\mathbf{S}_b)$ stands for the trace of $\mathbf{S}_b$ with the ith diagonal element excluded. The larger the value of $J_i(\mathbf{x})$ the higher individual discrimination power the ith feature possesses.

4. The technique proposed

Using the error backpropagation rule from Eq. (1) we can get

$$\frac{\partial o_j^{(L)}}{\partial x_i} = \delta_j^{(L)} \sum_m w_{mj}^{(L)} \delta_m^{(L-1)} \cdots \sum_l w_{lt}^{(3)} \delta_l^{(2)} \times \sum_q w_{ql}^{(2)} \delta_q^{(1)} w_{iq}^{(1)}, \tag{14}$$

where δ is the derivative of the neuron's transfer function. For the sigmoid function $\delta_j^{(L)} = o_j^{(L)}(1 - o_j^{(L)})$.

From Eq. (14) it can be seen that output sensitivity to the input depends on both the weight values and derivatives of the transfer functions of the hidden and output layer nodes. To obtain the low sensitivity desired we have chosen to constrain the derivatives. We train a neural network by minimizing the cross-entropy error function augmented with two additional terms:

$$E = \frac{E_0}{n_L} + \alpha_1 \frac{1}{Pn_h} \sum_{p=1}^{P} \sum_{k=1}^{n_h} f'(net_{kp}^h) + \alpha_2 \frac{1}{Pn_L} \sum_{p=1}^{P} \sum_{j=1}^{n_L} f'(net_{jp}^{(L)}), \quad (15)$$

where α_1 and α_2 are parameters to be chosen experimentally, P is the number of training samples, n_L is the number of the output layer nodes, $f'(net_{kp}^h)$ and $f'(net_{jp}^{(L)})$ are derivatives of the transfer functions of the kth hidden and jth output nodes, respectively, and

$$E_0 = -\frac{1}{2P}\left[\sum_{p=1}^{P} \sum_{j=1}^{n_L=Q} \left(d_{jp} \log o_{jp}^{(L)} + (1-d_{jp}) \log\left(1 - o_{jp}^{(L)}\right)\right)\right], \quad (16)$$

where d_{jp} is the desired output for the pth data point at the jth output node and Q is the number of classes.

The second and third terms of the cost function constrain the derivatives and force the neurons of the hidden and output layers to work in the saturation region. In (Jeong and Lee, 1996), it was demonstrated that neural networks regularized by constraining derivatives of the transfer functions of the hidden layer nodes possess good generalization properties. We can expect that different sensitivity of the hidden and output nodes could be required for solving a task with the lowest generalization error. Therefore, two hyper-parameters α_1 and α_2 are used in the error function.

The feature selection procedure is summarized in the following steps.

4.1. Feature selection procedure

1. Randomly initialize the weights for each member of a set of $j = 1, \ldots, L$ neural networks. For each neural network do Steps 2–8.
2. Randomly divide the data set available into `Training`, `Cross-Validation`, and `Test` data sets.
3. Train the neural network by minimizing the error function given by Eq. 15 and validate the network at each epoch on the `Cross-Validation` data set. Equip the network with the weights yielding the minimum `Cross-Validation` error.
4. Compute the classification accuracy A_{Tj} for the `Test` data set.
5. Identify the feature yielding the smallest drop of the classification accuracy for the `Test` data set when eliminating the feature. Elimination is implemented by setting the value of the feature to zero.
6. Eliminate the feature.
7. If the actual number of features $M > 1$ goto Step 3.
8. Record the feature ranking obtained and the test set classification accuracy A_{Tj} achieved using the whole feature set.
9. Compute the expected feature ranking and the expected accuracy $\widehat{A}_T$ by averaging the results obtained from the L runs.
10. Eliminate the least salient feature according to the expected ranking and execute Step 3.
11. Compute the `Test` data set classification accuracy and the drop in the accuracy ΔA when compared to $\widehat{A}_T$.
12. If $\Delta A < \Delta A_0$, where ΔA_0 is the acceptable drop in the classification accuracy go to Step 10.
13. Retain all the remaining and the last removed feature.
14. Retrain the network with the parsimonious set of features.

Classification accuracy obtained from a trained neural network depends upon the randomly selected training data set and the initial weight values. We can, therefore, expect that the neural network based feature ranking will also depend upon these factors. Aiming to reduce the dependence, we use the expected feature ranking obtained from the L networks trained on the different training data sets.

5. Experimental investigations

In all the tests, we run an experiment 30 times with different initial values of weights and different partitioning of the data set into ⟨`Training`⟩,

⟨Cross-Validation⟩, and ⟨Test⟩ sets. The mean values and standard deviations of the correct classification rate presented in this paper were calculated from these 30 trials.

5.1. Training parameters

There are four parameters to be chosen, namely the regularization constants α_1 and α_2, the number of networks L, and the parameter of the acceptable drop in classification accuracy ΔA_0 when eliminating a feature. The parameter affects the number of features included in the feature subset sought. The values of the parameters α_1 and α_2 have been found by cross-validation. The values of the parameters ranged: $\alpha_1 \in [0.001, 0.02]$ and $\alpha_2 \in [0.001, 0.2]$. The value of the parameter ΔA_0 has been set to 3%. We used $L = 10$ networks to obtain the expected feature ranking.

We use one hidden layer network with the sigmoid nonlinearities. Any number of hidden layers could be used in a general case. The cost function given by Eq. 15 is the function being minimized. To train the network, we used the backpropagation training algorithm with momentum implemented in the ⟨Matlab⟩ software package. In the implementation, the learning rate step size and the momentum rate are found automatically. For example, if the new error exceeds the old error by more than a predefined ratio (typically 1.04), the new weights are discarded and the learning rate is decreased (typically by multiplying by 0.7). If the new error is less than the old error, the learning rate is increased (typically by multiplying by 1.05). The influence of the momentum term is controlled in a similar manner. To make our results comparable with those presented in (Setiono and Liu, 1997), we also used 12 nodes in the hidden layer when learning the problems.

5.2. Data used

To test the approach proposed we used one artificial and three real-world problems. The data used in the experiments are available at: www.ics.uci.edu/mlearn/MLRepository.html.

We randomly assign available data exemplars into learning D_l, validation D_v, and testing D_t data sets. The learning set data are used in the learning algorithm for estimating the neural network weights. The validation data set is used for setting learning parameters. The testing set is used to test the developed procedures. Each data set is normalized in the following way. The average $\bar{x}$ and the variance s^2 are computed for the set $D_l \cup D_v$. Then the normalized $x_n = (x - \bar{x})/s$ are computed for the sets D_l, D_v, and D_t.

5.2.1. The Wave-form recognition problem

The ability of the technique to detect pure noise features has been tested on the 21-dimensional "*Wave-form*" data (Breiman et al., 1993) augmented with four additional independent noise components. There are given three waveforms $h_1(t)$, $h_2(t)$ and $h_3(t)$:

$$h_1(t) = \begin{cases} t & \text{if } 0 \leqslant t \leqslant 6, \\ 12 - t & \text{if } 7 \leqslant t \leqslant 12, \\ 0 & \text{if } 12 \leqslant t \leqslant 20, \end{cases} \tag{17}$$

$$h_2(t) = \begin{cases} 0 & \text{if } 0 \leqslant t \leqslant 8, \\ t - 8 & \text{if } 8 \leqslant t \leqslant 14, \\ 20 - t & \text{if } 14 \leqslant t \leqslant 20, \end{cases} \tag{18}$$

$$h_3(t) = \begin{cases} 0 & \text{if } 0 \leqslant t \leqslant 4, \\ t - 4 & \text{if } 4 \leqslant t \leqslant 10, \\ 16 - t & \text{if } 10 \leqslant t \leqslant 16, \\ 0 & \text{if } 16 \leqslant t \leqslant 20. \end{cases} \tag{19}$$

Patterns of the three decision classes ($\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in R^{21}$) are formed as a random convex combination of two of these waves (waveforms $(1,2)$, $(1,3)$, and $(2,3)$, respectively, for classes 1, 2, and 3) with noise added. We extended the dimensionality of the vectors to 25 by using four additional independent noise components. More specifically,

$$x_t^{(q)} = \begin{cases} uh_k(t) + (1-u)h_l(t) + \varepsilon_t^{(q)}, & 0 \leqslant t \leqslant 20, \\ \varepsilon_t^{(q)}, & 21 \leqslant t \leqslant 24, \end{cases} \tag{20}$$

where u is a uniform random number in $[0,1]$, and $\varepsilon_t^{(q)}$ are independent normally distributed random numbers with mean 0 and variance 1. The data sets D_l, D_v, and D_t contain 300, 1000, and 4000 samples, respectively.

5.2.2. US Congressional Voting Records problem

The United States Congressional Voting Records data set consists of the voting records of 435 congressman on 16 major issues in the 98th Congress. The votes are categorized into one of the three types of votes: (1) *Yea*, (2) *Nay*, and (3) *Unknown*. The task is to predict the correct political party affiliation of each congressman. The 98th Congress consisted of 267 Democrats and 168 Republicans.

We used the same learning and testing conditions as in (Bauer et al., 2000; Setiono and Liu, 1997), namely 197 samples were randomly selected for training, 21 samples were selected for cross-validation, and 217 for testing.

5.2.3. The diabetes diagnosis problem

The Pima Indians Diabetes data set contains 768 samples taken from patients who may show signs of diabetes. Each sample is described by eight features: (1) number of times pregnant, (2) plasma glucose concentration, (3) diastolic blood pressure, (4) triceps skin fold thickness, (5) two-hour serum insulin, (6) body mass index, (7) diabetes pedigree function, and (8) age. There are 500 samples from patients who do not have diabetes and 268 samples from patients who are known to have diabetes.

From the data set, we have randomly selected 345 samples for training, 39 samples for cross-validation, and 384 samples for testing.

5.2.4. The breast cancer diagnosis problem

The University of Wisconsin Breast Cancer data set consists of 699 patterns. Amongst them there are 458 benign samples and 241 malignant ones. Each of these patterns consists of nine measurements taken from fine needle aspirates from a patient's breast. The measurements used are: (1) clump thickness, (2) uniformity of cell size, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial cell size, (6) bare nuclei, (7) bland chromatin, (8) normal nucleoli, and (9) mitoses. All nine measurements were graded on an integer scale from 1 to 10, with one being the closest to benign and 10 being the most malignant. In the data, 16 samples of feature number (6) were missing. To estimate values of the missing variables we employed the same technique as in (Bauer et al., 2000), namely we performed a linear regression, using feature (6) as the independent variable and the other features as the dependent variables.

To test the approaches we randomly selected 315 samples for training, 35 samples for cross-validation, and 349 for testing.

5.3. Results of the tests

For the artificial Wave-form data set, we tested the ability of the different techniques to detect the noise features amongst other ones that were also corrupted by noise. Table 1 presents the first 10 features eliminated by the different techniques. The feature rankings presented are averaged over the 30 runs. As can be seen from the table, the *Proposed*, discriminant analysis (DA) based, and SNR methods have been able to detect all the noise features $\langle 1, 21, 22, 23, 24, 25\rangle$. Note that the features $\langle 1, 21\rangle$ are also equivalent to the noise features added. However, the OFEI and FQI techniques have failed to include all the noise features into the set of the first ten eliminated features. We have observed quite large variation between the different rankings obtained from the FQI technique in the different runs.

Fig. 2 presents the criterion $J_i(\mathbf{x})$ (Eq. 13) values calculated for all the individual features $\langle 1 \ldots 25\rangle$ of the $\langle$`Wave-form`$\rangle$ data set. Observe that features eliminated by the *Proposed* technique are those of the lowest individual discrimination power.

The US Congressional Voting Records problem is an easy task from the feature selection point of view, since there is only one feature $\langle 4\rangle$ exhibiting almost the same discrimination power as the whole

Table 1
Ten least salient features as deemed by the different techniques for the Wave-form data set

Method	Features
Proposed	23 21 1 2 24 22 25 20 3 19
DA	24 23 2 1 25 22 21 20 3 19
SNR	25 22 23 1 2 24 21 3 20 14
OFEI	18 8 16 22 24 19 17 23 20 9
FQI	23 2 1 25 21 3 14 24 20 16

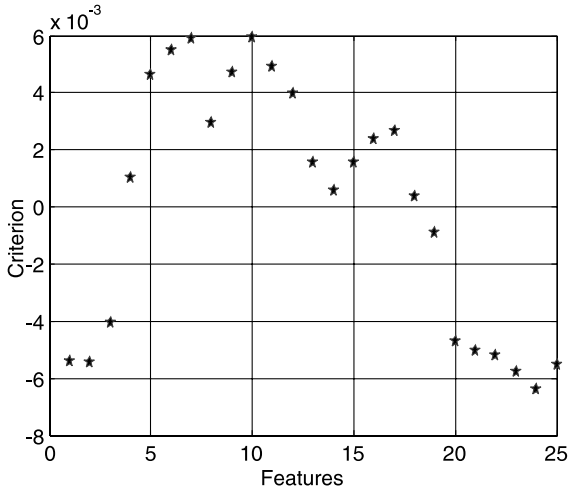

Fig. 2. Criterion $J_i(\mathbf{x})$ values for the individual features of the *Wave-form* data set.

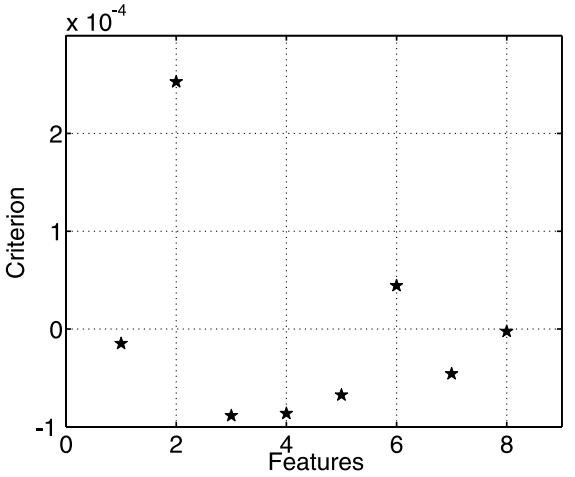

Fig. 3. Criterion $J_i(\mathbf{x})$ values for the individual features of the *Pima Indians Diabetes* data set.

feature set. All the techniques tested deemed the feature ⟨4⟩ as the most salient feature. Table 2 presents the test data set correct classification rate obtained from the method *Proposed*. In the table, we also provide the results taken from the references (Bauer et al., 2000; Setiono and Liu, 1997) describing the SNR and the NNFS method, respectively. In the parentheses, the standard deviations of the correct classification rate are provided. As the SNR method, the technique proposed selected only one feature for solving the task. Both techniques selected the same feature ⟨4⟩. The method proposed achieved the highest classification accuracy on the test data set. Note that the accuracy achieved is higher than that obtained in (Setiono and Liu, 1997) using two selected features. Setiono and Liu do not notify, which two features were most often selected by their technique. We do not provide test results for the DA, OFEI, and FQI approaches, since the methods and our technique selected the same feature ⟨4⟩.

The Pima Indians Diabetes problem is more difficult, since there are several salient features of approximately the same discrimination power. Fig. 3 presents the criterion $J_i(\mathbf{x})$ values calculated for all the individual features of the *Pima Indians Diabetes* data set. The feature ranking results obtained from the FQI technique were very dependent upon the network initialization and the data set partitioning into the ⟨`Training`⟩, ⟨`Cross-Validation`⟩, and ⟨`Test`⟩ sets. Table 3 exemplifies the feature rankings obtained from the techniques tested. As can be seen from the table,

Table 2
Correct classification rate for the Congressional Voting Records data set

Case	Proposed	SNR	NNFS
All features			
Training set	99.32(0.13)	98.92(0.22)	100.0(0.00)
Testing set	96.04(0.14)	95.42(0.18)	92.0(0.18)
Sel. features			
# of features	1(0.00)	1(0.00)	2.03(0.18)
Training set	95.71(0.25)	96.62(0.30)	95.63(0.08)
Testing set	95.66(0.18)	94.69(0.20)	94.79(0.29)

Table 3
Ranking of features starting from the most salient ones for the Pima Indians Diabetes data set

Method	Features
Proposed	2 6 7 8 3 1 5 4
DA	2 6 8 1 7 5 4 3
SNR	2 6 1 7 5 4 3 8
OFEI	2 3 6 7 5 8 1 4
FQI	8 2 1 4 7 6 5 3

Table 4
Correct classification rate for the Pima Indians Diabetes data set

Case	Proposed	SNR	NNFS	OFEI	FQI
All features					
Training set	80.64(0.53)	80.35(0.67)	95.39(0.51)		
Testing set	77.83(0.30)	75.91(0.34)	71.03(0.32)		
Sel. features					
# of features	2(0.00)	1	2.03(0.18)	2(0.00)	2(0.00)
Training set	76.83(0.52)	75.53(1.40)	74.02(1.10)	75.74(0.62)	75.68(2.17)
Testing set	76.81(0.45)	73.53(1.16)	74.29(0.59)	75.85(0.71)	75.28(2.49)

four techniques deemed the feature $\langle 2 \rangle$ to be the most salient one. Using the method proposed two features $\langle 2, 6 \rangle$ have been selected for solving the task. Note that two of the most salient features selected by the DA and SNR techniques are also $\langle 2, 6 \rangle$.

Table 4 provides the test data set correct classification rate achieved using feature subsets selected by the different techniques. Again the method proposed achieved the highest classification accuracy on the test data set. To obtain classification results for the OFEI and FQI techniques we also used two features as suggested by our method. The features employed were those selected by the techniques, namely $\langle 2, 3 \rangle$ and $\langle 8, 2 \rangle$. To train the networks with the selected features we minimized the same error function (Eq. (15)) as in our approach.

The University of Wisconsin Breast Cancer problem. In all the 30 runs performed, our technique suggested that two features should be selected for solving the task. Fig. 4 illustrates the criterion $J_i(\mathbf{x})$ values for the individual features of the data set. As can be seen from the figure, features $\langle 2, 3, 6 \rangle$ are of approximately equal individual discrimination power. Table 5 exemplifies the feature rankings obtained from the different techniques. Three techniques, namely FQI, OFEI, and SNR selected the same subset of two features: $\langle 6, 1 \rangle$. The *Proposed* technique and the DA approach made the same choice $\langle 6, 3 \rangle$ when a subset of two features was considered. However, examining the subsets consisting of three features, we see that none of the three trainable techniques (*Proposed*, SNR, and FQI) selected the best three

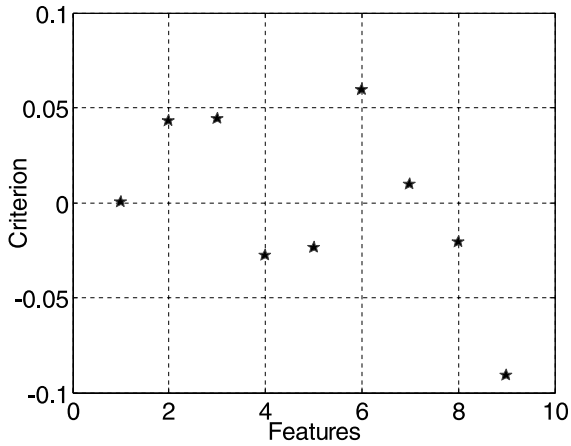

Fig. 4. Criterion $J_i(\mathbf{x})$ values for the individual features of *The University of Wisconsin Breast Cancer* data set.

Table 5
Ranking of features starting from the most salient ones for the University of Wisconsin Breast Cancer data set

Method	Features
Proposed	6 3 1 2 7 8 4 9 5
DA	6 3 2 7 1 8 5 4 9
SNR	6 1 3 2 7 8 4 5 9
OFEI	6 1 3 2 7 9 5 8 4
FQI	6 1 8 3 4 7 5 2 9

features as deemed by the DA approach. The feature selection result obtained from the FQI technique depended heavily upon the randomly chosen training set and the network initialization.

Table 6 presents the test data set correct classification rate obtained using feature subsets se-

Table 6
Correct classification rate for the University of Wisconsin Breast Cancer data set

Case	Proposed	SNR	NNFS	OFEI(FQI)
All features				
Training set	97.93(0.54)	97.66(0.18)	100.0(0.00)	
Testing set	96.44(0.31)	96.49(0.15)	93.94(0.17)	
Sel. features				
# of features	2(0.00)	1	2.7(1.02)	2(0.00)
Training set	95.69(0.44)	94.03(0.97)	98.05(0.24)	95.64(0.68)
Testing set	95.77(0.41)	92.53(0.77)	94.15(0.18)	95.46(0.62)

lected by the different techniques. Note that the results presented in the table for the SNR technique are taken from the literature (Bauer et al., 2000). Observe also that the results provided for the OFEI(FQI) approaches are obtained by minimizing the proposed error function given by Eq. 15. We do not provide classification results for the DA approach, since the subset of two features selected by the approach was the same as that obtained from the technique proposed. The results obtained indicate that the feature subsets $\langle 6,3\rangle$ and $\langle 6,1\rangle$ are of approximately the same discrimination power. As can be seen from the table, the proposed training and feature selection approach again yielded the highest classification accuracy on the test data set.

5.4. Tests for the k-NN classifier

In the next experiment, we used the feature subsets selected by the different approaches to classify the data sets by the *k-NN* classifier. Note that several approaches selected the same feature subsets. For example, for the *Voting* data set, the same feature $\langle 4\rangle$ has been selected by all the approaches tested. As in the previous tests, we run the experiment 30 times with different random partitioning of the data sets into the $\langle$`Training`$\rangle$ and $\langle$`Test`$\rangle$ parts. The nearest neighbours are selected from the $\langle$`Training`$\rangle$ part of the data, while the correct classification rate is evaluated on the $\langle$`Test`$\rangle$ data part. The size of the parts is the same as in the previous tests. Table 7 summarizes the results of the tests, where k stands for the number of nearest neighbours and CCR means correct classification rate. The values of $k \in \{1,3,5,7\}$ have been used in the tests. The value of k presented in Table 7 is the value yielding the highest correct classification rate. In all the tests, we used the *Euclidian* distance measure.

Table 7
Correct classification rate obtained from the *k-NN* classifier for the different data sets

Data set	Features used	k	CCR
Diabetes	All features	1	82.98(1.50)
Diabetes	$\langle 2,6\rangle$	1	83.14(1.90)
Diabetes	$\langle 2,3\rangle$	1	79.10(1.99)
Diabetes	$\langle 8,2\rangle$	1	80.87(1.41)
Cancer	All features	7	96.95(0.63)
Cancer	$\langle 6,3\rangle$	7	95.52(0.75)
Cancer	$\langle 6,1\rangle$	7	95.07(0.56)
Voting	All features	3	92.60(1.34)
Voting	$\langle 4\rangle$	5	95.42(0.92)

For the *Diabetes* data set the values of k larger than unity provided a significantly lower (4–6% lower) classification accuracy than that obtained for $k = 1$. For the other two data sets this was not the case. Examining Tables 3, 5, and 7 we find that the feature subsets selected by the method proposed yielded the best performance even using the *k-NN* classifier. Note that the DA based approach has also selected the same subsets of 1 and 2 features as the approach proposed.

6. Conclusions

The reduced risk of data-overfitting and the reduced cost of future data acquisition are the

main advantages of using small feature sets of only relevant features when solving classification problems. Therefore, robust feature selection procedures are of great value.

In this paper, we presented a neural network based feature selection technique. A network is trained with an augmented cross-entropy error function. The augmented error function forces the neural network to keep low derivatives of the transfer functions of neurons when learning a classification task. Such an approach reduces output sensitivity to the input changes. The feature selection is based on the reaction of the cross-validation data set classification error due to the removal of the individual features.

We have tested the technique proposed on one artificial and three real-world problems and demonstrated the ability of the technique to detect noisy features. The algorithm developed removed a large number of features from the original sets without reducing the classification accuracy of the networks noticeably. We compared the proposed approach with five other methods, each of which banks on a different concept, namely, the fuzzy entropy, the discriminant analysis, the neural network output sensitivity based feature saliency measure, the weights-based feature saliency measure, and the NNFS based on elimination of input layer weights. The technique developed outperformed the other methods by achieving the higher test data set classification accuracy on all the problems tested.

Acknowledgements

We gratefully acknowledge the support we have received from The Foundation for Knowledge and Competence Development.

References

Basak, J., Mitra, S., 1999. Feature selection using radial basis function networks. Neural Comput. Appl. 8, 297–302.

Bauer, K.W., Alsing, S.G., Greene, K.A., 2000. Feature screening using signal-to-noise ratios. Neurocomputing 31, 29–44.

Belue, L.M., Bauer Jr., K.W., 1995. Determining input features for multilayer perceptrons. Neurocomputing 7, 111–121.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1993. Classification and regression trees. Chapman & Hall, London.

Cibas, T., Soulie, F., Gallinari, P., 1996. Variable selection with neural networks. Neurocomputing 12, 223–248.

De, R.K., Pal, N.R., Pal, S.K., 1997. Feature analysis: neural network and fuzzy set theoretic approaches. Pattern Recognition 30 (10), 1579–1590.

Duda, R.O., Hart, P.E., 1973. Classification and Scene Analysis. Wiley, New York.

Fortoutan, I., Sklansky, J., 1987. Feature selection for automatic classification of non-Gaussian data. IEEE Trans. Systems Man Cybernet. 17 (2), 187–198.

Fukunaga, K., 1972. Introduction to Statistical Pattern Recognition. Academic Press, New York.

Jain, A., Zongker, D., 1997. Feature selection: evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Machine Intell. 19 (2), 153–158.

Ichino, M., Sklansky, J., 1984. Optimum feature selection by zero-one integerprogramming. IEEE Trans. Systems Man Cybernet. 14 (5), 737–746.

Jeong, D.G., Lee, S.Y., 1996. Merging back-propagation and Hebian learning rules for robust classifications. Neural Networks 9 (7), 1213–1222.

Kittler, J., 1986. Feature selection and extraction. In: Young, T.Y., Fu, K.S. (Eds.), Handbook of Pattern Recognition and Image Processing. Academic Press, New York, pp. 60–81.

Lotlikar, R., Kothari, R., 2000. Bayes-optimality motivated-linear and multilayered perceptron-based dimansionality reduction. IEEE Trans. Neural Networks 11 (2), 452–463.

Mucciardi, A., Gose, E.E., 1971. A comparison of seven techniques for choosing subsets of pattern recognition properties. IEEE Trans. Comput. 20 (9), 1023–1031.

Narendra, P.M., Fukunaga, K., 1977. A branch and bound algorithm for feature selection. IEEE Trans. Comput. 26 (9), 917–922.

Pal, N.R., 1999. Soft computing for feature analysis. Fuzzy Sets and Systems 103, 201–221.

Pal, S.K., De, R.K., Basak, J., 2000. Unsupervised feature evaluation: a neuro-fuzzy approach. IEEE Trans. Neural Networks 11 (2), 366–376.

Pal, S.K., Rosenfeld, A., 1988. Image enhancement and thresholding by optimization of fuzzy compactness. Pattern Recognition Lett. 7, 77–86.

Priddy, K.L., Rogers, S.K., Ruck, D.W., Tarr, G.L., Kabrisky, M., 1993. Bayesian selection of important features for feedforward neural networks. Neurocomputing 5, 91–103.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Lett. 15, 1119–1125.

Reed, R., 1993. Pruning algorithms – a survey. IEEE Trans. Neural Networks 5, 740–747.

Setiono, R., Liu, H., 1997. Neural-network feature selector. IEEE Trans. Neural Networks 8 (3), 654–662.

Steppe, J.M., Bauer, K.W., 1996. Improved feature screening in feedforward neural networks. Neurocomputing 13, 47–58.

Steppe, J.M., Bauer, K.W., Rogers, S.K., 1996. Integrated feature and architecture selection. IEEE Trans. Neural Networks 7 (4), 1007–1014.

Yager, R.R., Zadeh, L.A., 1994. Fuzzy sets, neural networks, and softcomputing. Van Nostrand-Reinhold, New York.

Zurada, J.M., Malinowski, A., Usui, S., 1997. Pertubation method for deleting redundant inputs of perceptron networks. Neurocomputing 14, 177–193.