Soft computing on small data sets

Bojan Novak
University of Maribor, Faculty of Electrical Engineering, Computer Science and Informatics,
Smetanova 17, 2000 Maribor, novakb@uni-mb.si

Keywords: soft computing, machine learning, neural networks

Received: October 24, 2000

The fusion of artificial neural networks (ANN) with soft computing enables to construct learning machines that are superior compared to classical ANN because knowledge can be extracted and explained in the form of simple rules. If the data sets are small it is hard to find the optimal structure of ANN because classical statistical laws do not apply. One possible remedy is the structural risk minimization method applied together with a VC dimension estimation technique. The construction of the optimal ANN structure is done in the higher dimensional space. The distortion of an image in this transformation can happen and the widely used expression for VC estimations based on minimal input data enclosing hypersphere and margin is not precise. An improvement of VC dimension estimation is presented. It enables better actual error estimation and is particularly suitable for the small data sets. Tests on some real life data sets have confirmed the theoretical expectations.

1 Introduction

One of the important steps in improving the usefulness of artificial neural networks (ANN) was fusion with soft computing [Zadeh 1994]. The construction of learning machines that are able to extract knowledge and explain it in the form of simple rules is now possible. The problem of an optimal number of neurons and weights of connections and their values is in a fact global optimization problem because of the nonlinear relations between the input and output data.

When data sets are small classical statistical laws do not apply and the construction of an optimal ANN is even harder. This problem is handled with the application of the structural risk estimation method together with the VC dimension estimation technique. Mapping of nonlinear relations between the input and the output is done by a transformation into the higher dimensional Hilbert kernel space where the problem is linearized. A distortion of image in this transformation can happen and widely used VC estimations based on minimal enclosing hypersphere and margin are not precise anymore.

In this paper a different approach that enables better VC estimation is presented. It is integrated into the structural risk minimization technique. An efficient strategy for constructing fuzzy artificial neural network (FANN) with the minimal actual error has been developed that can be easily implemented as a small addition to the existing FANN learning algorithm.
The performances of the proposed method were tested on some small data sets from the UC Irvine machine learning repository. The obtained results have confirmed theoretical expectations.

2 Support vector fuzzy modeling

In this chapter some basic definitions and modeling procedures are set. For given k observations, each consisting of a pair: x_i, y_i, where $x_i \in R^n$, $i=1,\ldots,k$ is the input vector and y_i is the associated output having values −1 or 1. Learning a machine is actually building up a mapping ability $x \rightarrow f(x,\alpha)$ where the functions $f(x,\alpha)$ themselves are labeled by adjustable parameters α. For the ANN α represents weights and biases. The expectation of test error for the trained machine is

$$R(\alpha) = \int \frac{1}{2} \left| y - f(\mathbf{x},\alpha) dP(\mathbf{x},\alpha) \right| \tag{2.1}$$

where R(α) is the expected risk. The measured mean error rate on the finite number of observations is "empirical risk"

$$R_{emp}(\alpha) = \frac{1}{2k} \sum_{i=1}^{k} \left| y - f(\mathbf{x}_i,\alpha) \right| \quad . \tag{2.2}$$

$R_{emp}(\alpha)$ is fixed for a particular choice of α and for a particular training set $\{x_i, \alpha\}$. The probability is not included in the equation. The quantity $\frac{1}{2} \mid y_i - f(x_i, \alpha) \mid$ is loss function. Empirical risk minimization does not imply a small error on a test set if the number of training data is limited. The structural risk minimization is one of new techniques for handling efficiently a limited amount of data. For a chosen η: $0 \leq \eta \leq 1$ the bound holds

$$R(\alpha) \leq R_{emp}(\alpha) + \Phi(\frac{h}{k}, \frac{\log(\eta)}{k}) \qquad (2.3)$$

where Φ is defined as :

$$\Phi(\frac{h}{k}, \frac{\log(\eta)}{k}) = \sqrt{\frac{h(\log\frac{2k}{h}+1) - \log(\frac{\eta}{4})}{k}} \qquad (2.4)$$

The parameter h is the Vapnik Chervonenkis (VC) dimension [Vapnik 1998]. It describes the capacity of a set of functions implemented on the learning machine.

According to eq. (2.3), the risk could be controlled by two quantities: R_{emp} *(α)* and *h(f(x,α): α ∈* k_{sub} *)*, where k_{sub} is some subset of index set *k*. The empirical risk R_{emp} depends on the choice of the optimal function (*α*) applied in the learning machine. The VC dimension *h* depends on the set of functions *{f(x,α) : α ∈* k_{sub}*}*. The parameter *h* is controlled by introducing the structure of nested subsets $S_n := \{f(x,\alpha) : \alpha \in k_n\}$

$$S_1 \subset S_2 \subset S_3 \subset .... \subset S_n \subset .... \qquad (2.5)$$

with adequate VC dimensions satisfying

$$h_1 \leq h_2 \leq ..... \leq h_n \leq ......$$

The structural minimization principle chooses the function *f(x,α*)* in the subset *{f(x,α): α∈* k_{sub}*}* with the minimal right hand side of (2.3). The guaranteed risk bound is minimal. For a given set of data

$$(\mathbf{x}_1, y_1),.....,(\mathbf{x}_k, y_k),\ \ \mathbf{x} \in R^n, y \in \{-1,1\} \qquad (2.6)$$

a separation of two classes can be performed with an optimal hyperplane

$$(\mathbf{w}_0 \cdot \mathbf{x}) + b_0 = 0 \ . \qquad (2.7)$$

The margin

$$\rho = \frac{1}{\|\mathbf{w}\|}$$

can be maximized by the following quadratic programming model

$$\min \Phi(\mathbf{x}, \xi) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C\sum_{i=1}^{k} \xi_i \qquad (2.8)$$

subject to constraints

$$y_i(\mathbf{w}_0 \mathbf{x}_i + b_0) \geq 1 - \xi_i$$

and

$$\xi_i \geq 0$$

where ζ are slack variables introduced in the case when the problem is not separable, and C is the pre-specified value.

For the nonlinear cases a non-linear support vector approach is applied. A non-linear mapping is applied to map the data in a higher dimension feature space where a linear classification is applied. This is possible with the kernel functions. These functions originate from the theory of Reproducing Kernel Hilbert Spaces [Aronszajn 1950]. An inner product in the feature space has an equivalent kernel input space

$$K(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}) \cdot k(\mathbf{y}) \ . \qquad (2.9)$$

If K is a positive definite function, satisfying Mercer's conditions

$$K(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^{\infty} \alpha_m \psi(\mathbf{x})\psi(\mathbf{y}), \quad \alpha_m \geq 0 \qquad (2.10)$$

$$\iint K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} > 0, \quad \int g^2(\mathbf{x}) d\mathbf{x} < \infty \qquad (2.11)$$

then the kernel is a legitime product in the feature space.

There are different functions satisfying Mercer's condition: polynomial, splines, B-splines, radial basis functions, etc. In the present work the Gaussian radial basis function is used:

$$K(\mathbf{x}, \mathbf{y}) = \exp[-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}] \ . \qquad (2.12)$$

The support vector technique places one local Gaussian function in each support vector. This means that there is no need for a clustering method. The basis width σ is selected using structural minimization principle (2.3) and (2.5). The non-linear classification support vector solution using kernels can be solved by

$$\min f(\mathbf{x}, \alpha) = sign(\sum_{i=1}^{k} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \qquad (2.13)$$

subject to constraints

$$\alpha_i \geq 0 \ . \qquad (2.14)$$

The coefficients α_i can be found by the following quadratic optimization problem

$$\max W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{k}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{y}_j)+\sum_{i=1}^{k}\alpha_i \tag{2.15}$$

subject to constraints

$$\sum_{i=1}^{k}\alpha_i y_i = 0$$
$$0 \le \alpha_i \le C, \quad i = 1,....,k \tag{2.16}$$

In the solution of (2.15) only some coefficients α_i differ from zero. The corresponding vectors are support vectors.

The model described by (2.15) has only one optimum (that is also global) which is a great advance against the backpropagation based learning algorithm in ANN.

The following support vector FANN architecture can be defined as presented in the fig. 2.1

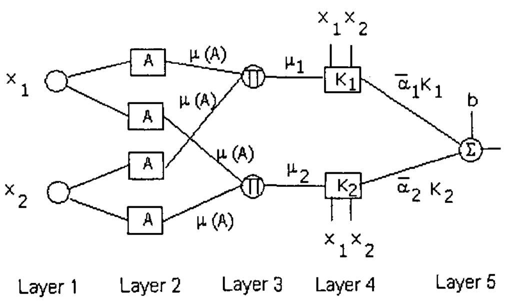

Figure 2.1 Support vector FANN architecture

Layer 1 calculates membership values. Layer 2 performs T norm operator (multiplication). Layer 3 derives the product of each rule's output. Layer 4 performs the kernel Gaussian radial basis operation (2.12). Layer 5 sums its inputs as the overall output where $\acute{\alpha}_i = \alpha_i$. The non-linear classification function is

$$f(x) = sign(\sum_{i=1}^{SV}\alpha_i K(x_i,x)+b) \tag{2.17}$$

where SV is the number of support vectors.

3 Optimization of parameters

Described are the conditions for optimizing the parameters of FANN, considering that some asymptotical laws from statistics are not valid for small data sets.

The empirical risk $R_{emp}(\alpha)$ in (2.2) is fixed for a particular choice of α and for a particular training set $\{x_i, \alpha\}$, and the probability is not included in the equation. The risk functional (2.1) depends on the conditional distribution function (c.d.f.) $P(x, \alpha)$, which is not known in advance. The only available information is from the finite independent and identically distributed (i.i.d.) training data sets. There are two possible approaches. The first one is to estimate the unknown c.d.f. $P(x, \alpha)$, or also unknown probability distribution function (p.d.f.) in the form of $p(x)$, and then compute the optimal estimate $f(x, \alpha_0)$. The second approach is to find a minimum of the empirical risk, which is calculated with the empirical risk minimization procedure (ERM). The second approach is preferable on the small data sets. It works if the asymptotic consistency is fulfilled

$$R(\alpha^*|\ k) \rightarrow R(\alpha 0) \quad when\ k \rightarrow \infty \tag{3.1}$$

$$R_{emp}(\alpha^*|\ k) \rightarrow R(\alpha 0) \quad when\ k \rightarrow \infty, \tag{3.2}$$

where $R_{emp}(\alpha^*|\ k)$ is the optimal value that minimizes the empirical risk in the loss function (2.2). $R(\alpha^*|\ k)$ is the unknown value of the true risk for the same loss function. The ERM is consistent if the true risk $R(\alpha^*|k)$ and the empirical risk $R_{emp}(\alpha^*|\ k)$ converge to the same limit $R(\alpha_0) = min_\alpha\ R(\alpha)$ as the number of samples k grows toward the infinite value (fig. 3.1).

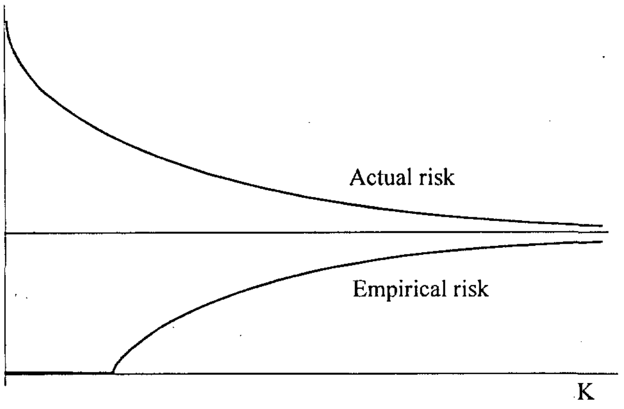

Fig. 3.1 Convergence of the empirical risk toward the actual risk

The condition that a small actual risk will be guaranteed at a small empirical error is given by the following limit [Vapnik et al. 1989]

$$\lim_{k\to\infty} P[\sup_{\alpha} |\ R(\alpha) - R_{emp}(\alpha)\ | > \varepsilon] = 0, \quad \forall\varepsilon > 0 \tag{3.3}$$

The consistency is determined for the worst case function.

The structural risk minimization principle is applied to find the optimal value for the true risk estimation

R(α|k)*. The penalization method is used to find the optimal function *f(x, α₀)* from the set of functions *f(x, α)*. Then the true risk *R(α)* is related to the empirical risk

$$R(\alpha) = R_{emp} r(p) \tag{3.4}$$

where *r(p)* is the penalization factor. In the support vector approach the penalization factor is of the form (2.4) and is proportional to the ratio of the VC dimension divided by the number of samples *k*.

The structural risk minimization principle (2.5) can be realized by estimating the VC dimension as a product of the radius of the minimal sphere that encloses data in the feature space divided by the margin (the distance between the hyperplane and the closest training vector in the feature space)

$$h = \frac{D^2}{\rho^2} \tag{3.5}$$

The actual risk prediction can be estimated very efficiently by the leave-one-out procedure. It consists of removing sequentially one sample from the training data and constructing *k* learning machines and testing all *k* elements for an error *L(x_i,y_i)*. The leave-one-out estimator is almost unbiased, that is

$$E\frac{L(\mathbf{x}_1 y_1,\dots,\mathbf{x}_k y_k)}{k} = ER(\alpha_{k-1}) \quad . \tag{3.6}$$

For optimal hyperspheres passing through the origin the equivalent of the expression (3.6) is

$$E\left[p^{k-1}_{error}\right] \le \frac{E\left[D^2 w^2\right]}{k} \tag{3.7}$$

where p^{k-1}_{error} is the probability of error on the test set. The expectation on the left is over all training sets of size *k-1*. The expectation on the right is over all training sets of size *k*.

This bound is tight when data fill almost the whole area of the sphere (fig 2.2 left) enclosing the training data. The consequence of data transformation into the feature space is that the sphere is often transformed into a flat ellipsoid (fig. 2.2 right). The bound (3.7) is not tight anymore.

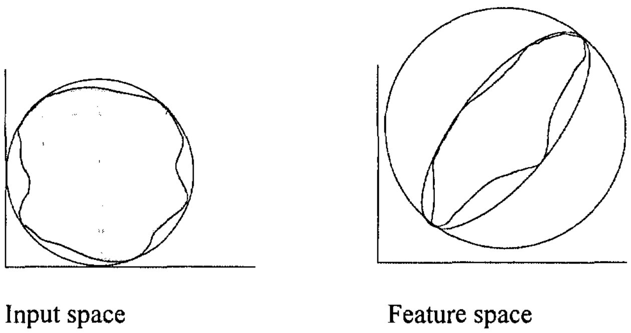

Fig. 3.2 the shape of the feature space

It is possible to achieve a better upper bound on the estimate of the expected error rate. The upper bound is constructed from the leave-one-out bound [Luntz 1969], Opper-Winther bound [Opper et al.], Khun Tucker optimality conditions [Karush 1939, Kuhn et al. 1951] and properties of the essential support vectors [Vapnik 1998].

The optimal supporting plane is unique, but can be expressed with different expansions of support vectors. Essential support vectors are those support vectors that appear in all possible expansion of an optimal hyperplane. Their number is presented by kesv and they have the following properties

$$k_{esv} \le n \tag{3.8}$$

where *n* is the dimensionality of the transformed input data in the feature space.

Let *ER(α)* is the expectation of the probability of an error for optimal hyperplanes constructed on the basis of training samples of size *k*, then the following inequality holds

$$ER(\alpha_{k-1}) \le \frac{EK_{esv(k)}}{k} . \tag{3.9}$$

In the case of a learning machine without threshold under assumption that the set of support vector does not change after removing example p (essential support vector), Opper-Winther equality applies

$$y_p(f^0(\mathbf{x}_p) - f^p(\mathbf{x}_p)) = \frac{\alpha_p^0}{(K_{SV}^{-1})_{pp}} , \tag{3.10}$$

K_{SV} is the kernel matrix (2.12). The f_0 is decision function (2.13) trained on the whole training set and f_p is decision function after one point x_p has been removed. It follows from (3.9) that the number of errors in the leave-out procedure is proportional to the number of essential support vectors. Instead of computing them we can use (3.10) and count the number of cases L_E when

$$y_p f^0(\mathbf{x}_p) \le \frac{\alpha_p^0}{(K_{SV}^{-1})_{pp}} \qquad (3.11)$$

is true. Then expectation is

$$ER(\alpha_{k-1}) = \frac{L_E}{k} \qquad (3.12)$$

4 Experimental results

The program developed on the described principle for constructing optimal fuzzy learning machine on small data set was tested on some real data sets from practice. The data are form the UC Irvine machine learning repository_(www.ics.uci.edu/~mlearn/MLRrepository). The first data set is a sonar data set (originally from www.boltz.cs.cmu.edu/ benchmarks/ sonar.html). The input consists of 104 samples of dimension 60, and the test data set is of the same size.

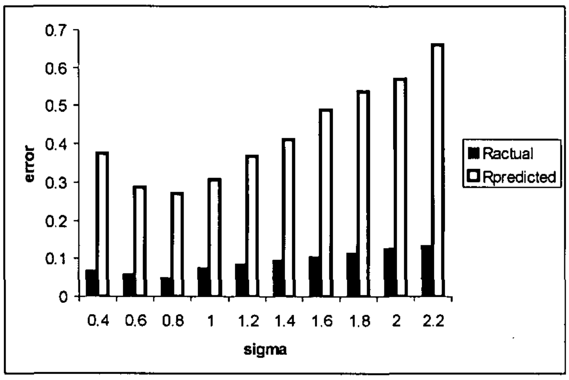

Fig. 4.1 The actual and the predicted risk for the sonar data

The actual error $R(\alpha)$ and its prediction (3.12) for different σ (sigma) values is presented in the fig 4.1. In our case a different set of functions $f(x, \alpha)$ is generated by varying σ in (2.12) for the radial basis function (α is actually σ in our case). The results are average values of a 100 fold repetition for each sigma value. Errors are given in the relative value (as in (2.2)).

Next example is the ionosphere data set with the input dimension n=33 and 200 learning samples and 151 test examples randomly generated from the complete set of 351 samples. The actual error $R(\alpha)$ and its prediction (3.12) for different σ (sigma) values is presented in the fig 4.2.

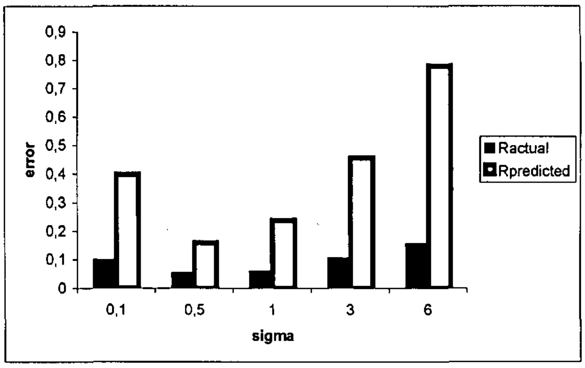

Fig. 4.2 The actual and the predicted risk for the ionosphere data

The last example is the Pima Indians diabetes data set with the input dimension n = 8 and 200 learning samples and 200 test examples randomly generated from the complete set of 768 samples.

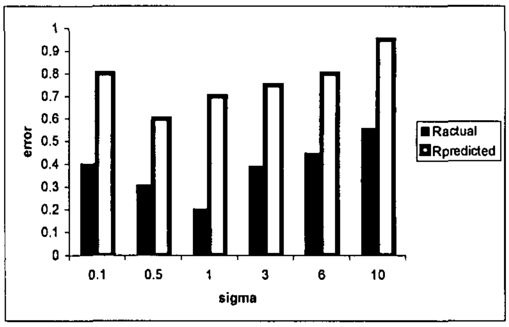

Fig. 4.3 The actual and the predicted risk for the Pima data

The actual error $R(\alpha)$ and its prediction (3.12) for different σ (sigma) values is presented in the fig 4.3.

5 Conclusion

In practice it often happens that the amount of data is limited. Such cases appear in engineering where data are collected through expensive experiments or in medicine where records on certain diseases are rare. When a data set is small a significant discrepancy between the empirical error achieved on the learning data set and the actual error on the testing data set appears. The actual error can be minimized with the structural risk minimization principle. It is calculated with the application of the VC dimension, which has to be estimated precisely to achieve good results.

The estimation based purely on the margin and the minimal diameter of sphere including input data can be inadequate due to possible flattening of the sphere caused by mapping data into the feature space.

In this paper a different approach that enables better VC estimation is presented. It is integrated into the structural risk minimization technique. An efficient strategy for constructing FANN with the minimal actual error has been developed that can be easily implemented as a small addition to the existing FANN learning algorithm.

The performances of the proposed method were tested on some small data sets from the UC Irvine machine learning repository. The obtained results have confirmed theoretical expectations.

6 References

[1] N. Aronszajn, Theory of Reproducing Kernels, Trans. Amer. Math. Soc. 68, 337-404 , (1950).

[2] W. Karush, Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dep. Of Mathematics, Univ. of Chicago, 481-492, (1939).

[3] W. Kuhn. & A.W. Tucker, Nonlinear Programming in (J. NEYMANN ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, CA, 481-492, (1951),

[4] A. Luntz & V. Brailovsky, On estimation of characters obtained in the statistical procedure of recognition (in Russian), Techniceskaya Kibernetica, 3, (1969).

[5] M. Opper & O. Winther, Gaussian processes and SVM: Mean field and leave one out, in (A. J. SMOLA, P. L. BARTLETT, B. SCHÖLKOPF AND D. SCHUURMANS, editors), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 311-326, (2000).

[6] V. N. Vapnik, Statistical Learning Theory, John Wiley and Sons, (1998).

[7] V. Vapnik & J. Chervonenkis: The necessary and sufficient conditions for the consistency of the method of empirical risk minimization (in Russian), Yearbook of the Academy of Science of the USSR on Recognition, Classification, and Forecasting, 2, Moscow, 217-249, (1989). (English translation, The necessary and sufficient conditions for the consistency of the method of empirical risk minimization, Pattern Recog. Image Anal. 1, 284-305, (1991).

[8] L. A. Zadeh, Fuzzy logic, neural networks, and soft computing, Commun. ACM, 37/3, 77-84, (1994).