HOW THE SELECTION OF TRAINING PATTERNS CAN IMPROVE THE GENERALIZATION CAPABILITY IN RADIAL BASIS NEURAL NETWORKS

J. M. Vails, I. M. Galvan and P. Isasi
Carlos III University of Madrid, Computer Science Department.
Avd. Universidad, 30, 28911, Leganes, Madrid. ivaUs(aUnf.u.c3m.es

ABSTRACT

It has been shown that the selection of the most similar training patterns to generalize a new sample can improve the generalization capability of Radial Basis Neural Networks. In previous works, authors have proposed a learning method that automatically selects the most appropriate training patterns for the new sample to be predicted. However, the amount of selected patterns or the neighborhood choice around the new sample might influence in the generalization accuracy. In addition, that neighborhood must be established according to the dimensionality of the input patterns. This work handles these aspects and presents an extension of a previous work of the authors in order to take those subjects into account. A real time-series prediction problem has been chosen in order to validate the selective learning method for a n-dimensional problem.

KEY WORDS: Radial Basis Neural Networks, Selective Learning.

1 Introduction

The interest of the scientific community to improve the generalization capabilities of neural networks has increased. Some authors have paid attention to the nature and size of the training set. There is no guarantee that the generalization performance is improved by increasing the training set size [1]. It has been shown that with careful dynamic selection of training patterns, better generalization performance maybe obtained [2,3].

In [4] a selective learning mechanism was present to improve the generalization capabilities of Radial Basis Neural Networks (RBNN). In that work, the idea of selecting the patterns to train the network from the available data about the domain was proposed. The selection of patterns used in the training phase is based on novel samples, instead of based on other training patterns, as in other works [2,3]. Thus, the network will use its current knowledge of the new sample to have some deterministic control about what patterns should be used for training. The learning method involves finding relevant data to answer a particular novel pattern and defer the decision of how to generalize beyond the training data until each new sample is encountered. It was inspired on lazy strategies [5,6]. Thus, the decision about how to generalize is carried out when a test pattern needs to be answered constructing local approximations.

The method presented in [4] recognises from the whole training data set, the most similar patterns, in terms of the Euclidean distance, for each new pattern to be processed. This subset of useful patterns is used to train a RBNN and training is deferred until a test pattern is received. Taking advantage of the fast convergence of this kind of networks, a complete RBNN is trained for each test pattern, and the output is generated by propagating the pattern through its related RBNN.

Following the main ideas of the former work [4], some contributions are made in this paper. A threshold distance or cut, which determines the extension of the neighborhood of the novel pattern, (i.e. the training patterns selected to generalize the new sample), becomes the key issue to achieve a good generalization capability of RBNN. On the other hand, the dimensionality of data is taken into account in order to establish the cut, since the volume surrounding the novel pattern grows exponentially with the dimension.

The interest of this paper is, firstly, to introduce a relative threshold distance that is not dependent on the dimensionality of data. Secondly, to study the influence in the deferred learning method of the relative threshold distance. With these purposes, the learning method has been applied to two different problems, an artificial approximation problem and a real time-series prediction problem.

2 Automatic Selection of Training Data

The learning algorithm proposed in [4] consists of the selection, from the whole training data, of an appropriate subset of patterns to improve the answer of the RBNN for a novel sample. The general idea for the selection of patterns is to include only and several times those patterns close -in terms of the Euclidean distance- to the novel sample. Thus, the network is trained with the most useful information, discarding those patterns that not only provide no knowledge to the network, but, besides, can confuse the learning process.

In order to take into account the dimensionality of the data, the formulation of the method has been modified.

The general idea presented in this work for the selection of patterns is to establish a relative n-dimensional volume surrounding the test pattern, in order to include only and several times those patterns placed into this volume. This relative volume surrounding the test pattern is the fraction of the total volume into which the training patterns will be selected. This parameter allows that the dimensionality of data does not take part on the account of patterns selected.

In order to determine the training patterns included in that relative volume, a relative threshold distance or cut r_r (relative radius of the sphere) must be calculated before the application of the learning algorithm. Let's V_r a relative n-dimensional volume. That volume can be written as:

where V_s is the volume of the selection-sphere centered at the test pattern and radius r_s - this sphere contains the patterns that will be selected- and V_{max} is the volume of the sphere centered at the test pattern and radius equals to the maximum distance, r^. Since V=kr" where r is the radius of the sphere, n is the dimension of the space and k is a constant, we can write:

$$v_r = i =_r ;$$

Hence:

The relative threshold distance, r_r, calculated as the n-th root of the relative volume_will be used to select patterns in the fraction volume surrounding the test pattern.

Given a test pattern q, described by a n-dimensional vector, q=(q,,..., q,), the steps to select the training set, named X_q, associated to the patterns, are the following:

Step 1. A real value, d_b is associated to each training pattern (x_b y*). That value is defined in terms of the standard Euclidean distance.

$$d_k = d(x_k,q) = J \wedge K \; ^{-}9i)^2$$

Step 2 A relative distance, cU is calculated for each training pattern. Let d_{niax} be the maximum distance to the novel pattern, this is d_{na}, = Max (d,, d_2, ...d_N). Then, the relative distance is given by:

$$d_rk = <Vd_{nla}($$

Step 3L A new real value, $f_k=l/d_{rk}$, where k=l,...,N is associated to each training pattern (x^y^. These values f_k are normalized in such a way that the sum of them equals the number of training patterns in X. The relative values, named as frit, are obtained by:

$$fn,. = \frac{/}{\quad} \quad \text{where} \quad S = \frac{1}{\quad} \; \sum_{k=1}^{N} f,$$

$$\text{Thus: } \sum_{k=\backslash}^{N} Y \wedge f^n k = N$$

Step 4. The relative distance, $d_{rk>}$ calculated in step 2 is used to decide if the training pattern (Xk, y^ is selected to train the network. If $d_{rk} < r_r$ -where r. is the relative threshold distance previously calculated- then the pattern (Xf, yO is included in the training subset.

The value fn_k calculated in step 4 is used to indicate how many times the training pattern (x^ y^ is repeated into the new training subset. Hence, they are transformed to natural numbers as: $n_k = \backslash nt(fn_k) + 1$

At this point, each training pattern in X that has been selected has an associated natural number, r^ which indicates how many times the pattern (Xfo y^ has been used to train the RBNN when the new instance q is reached.

Step 5. A new training pattern subset associated to the test pattern q, X_q, is built up. Once the training patterns are selected, the RBNN is trained with the new subset of patterns, X,. Training a RBNN involves to determine the centers, the dilations or widths, and the weights. The centers are calculated in an unsupervised way using the K-means algorithm presented in [4]. After that, the dilations coefficients are calculated as the square root of the product of the distances from the respective center to its two nearest neighbors. Finally, the weights of the RBNN are estimated in a supervised way to minimize the mean square error measured in the training subset X_q.

3 Experimental Results

The learning method presented in this work has been applied to two different problems: An artificial approximation problem, a piecewise-defined function whose dimension is 1, and a n-dimensional, with n>l, real problem, a time-series describing the behavior of the water level at Venice Lagoon. In order to validate the method, RJBNNs have also been trained as usual, this is, the network is trained using the whole training data set.

3.1 An artificial problem: A piecewise-defined function approximation

This function has been chosen because of the poor generalization performance that RBNN presents when approximating it. The function is given by the equation:

$$f(x) = \begin{cases} -2.186x - 12.864 & \text{if } -10 < x < -2 \\ 4.246x & \text{if } -2 < x < 0 \\ 10e^{(-0.05x)}\sin[(0.03x+0.7)x] & \text{if } 0 < x < 10 \end{cases}$$

The original training set is composed by 120 input-output points randomly generated by an uniform distribution in the interval [-10,10]. The test set is composed by 80 input-output points generated in the same way as the points in the training set. Both sets have been normalized in the interval [0,1].

RBNNs with different number of neurons have been trained, using the whole training data until the convergence of the network has been reached, that is, either when 200 cycles are performed or when the derivative of the train error equals zero. In figure 1, the mean errors obtained for different architectures are shown. The best results have been achieved using a RBNN with 20 neurons, although no significant differences have been found for networks between 10 and 60 neurons. The selective learning method has also been used to train RBNNs with different architectures and different relative volumes (Vr) during 200 learning cycles, and their generalization capability has been tested. Mean errors on the test set achieved by these networks are shown in figure 2. In this case, 11 neurons are necessary to obtain the best results using a selection relative volume of 0.06. As dimension n= 1, then $r_r = V_r$

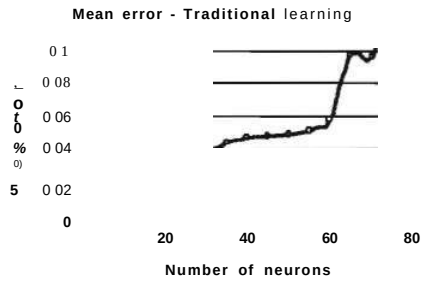

Fig. 1. Mean error on the test set for the Piecewise function achieved by different architectures trained with the whole training data set

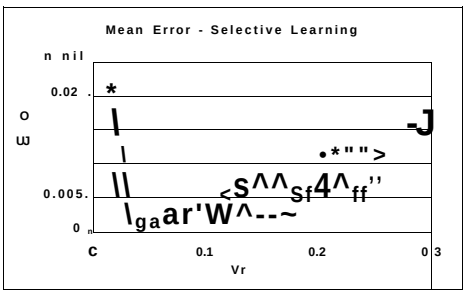

Fig. 2. Mean error on the test set for the Piecewise function achieved by different architectures trained with a selection of patterns using different selection volumes (Vr)

As it can be seen, the performance of the network is influenced by the value of the relative volume and by the network architecture.
In figure 2 it is observed the mean error does not depends significatively on Vr, provided Vr is bigger than a certain value.

As it is possible to observe in table l, the mean error over the test set is significantly **reduced when** an appropriate selection of patterns is made.

Table 1. Performance of different training methods for the piecewise function

	Mean error	Number of Neurons
Training with the whole data set	0 0396	20
Training with a selection of data and $V_r = r_r = 0\ 06$	0 002085	11

The computational cost is higher when the deferred training method is used, although, on the other hand, the number of neurons is smaller, and the RBNN is trained in a shorter **time. In that** case, the RBNN has been trained until **it** reaches **the** convergence. Thus, the generalization capability of the network using the whole training data can not be improved if it is trained for more learning cycles.

It has been observed how the network trained with the whole training data has some difficulties to approximate the points. This can be described as a deficiency in the generalization capabilities of the network. However, the generalization is improved when an appropriate selection of patterns is made. Figure 3 shows the errors committed by the different learning strategies for each test pattern. Most of test patterns are better approximated when the specific learning method is used to train RBNN.

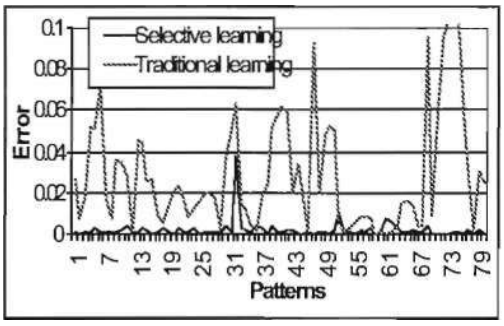

Fig. 3. Piecewise-defined function: Errors for each set patterns.

3.2 A Real Problem: Prediction of water level at Venice Lagoon

Unusually high tides, result from a combination of chaotic climatic elements in conjunction with the more normal, periodic, tidal systems associated with a particular area. The prediction of such events have always been subjects of high interest. The water level of Venice Lagoon is a clear example of these events. That phenomenon is known as "high water".

Different approaches have been developed for the purpose of predicting the behavior of sea level at Venice Lagoon [7,8,9,10]. In this work RBNNs have been used for predicting the sea level.

There is a great amount of data representing the behavior of the Venice Lagoon time series. However, the part of data associated to the stable behavior of the water is very abundant as opposed to the part associated to **high** water phenomena. This situation leads to **the** following: RBNN trained with a complete data set is **not** very accurate in predictions of high water phenomena. It seems natural that if the network is trained with selected patterns, the predictions will improve.

In this work, a training data set of 3000 points, corresponding to the level of water measured each hour has been extracted from available data in such a way that both stable situations and high water situations appear represented in the set. The test set has also been extracted from the available data and it is formed by 50 samples including the high water phenomenon. Both sets of data have been normalized in the interval [0,1].

Since the goal in this work is to predict only the next sampling time, a nonlinear **model using** the six previous sampling times seems appropriate.

RBNNs with different number of neurons have been trained, using **the** whole training data until the convergence of **the** network has been reached, that is, either when 300 cycles are performed or when the derivative of the train error equals zero. In figure 4, mean errors obtained for different architectures are shown. The best results have been achieved using a RBNN with 30 neurons, although no significant differences have been found for networks between 15 and 150 neurons. It is observed that the test mean error can not be improved even if more learning cycles are performed using the whole training data set.

The selective learning method described in section 2 has also been used to train RBNNs with different architectures and different relative volumes (Vr) during 300 learning cycles, and their generalization capability has been measured. Mean errors on the test set achieved by these networks are shown in figure 5. As in the previous example, the performance of the network is influenced by the value of the relative volume and the architecture of the network. It is possible to observe that, as in the previous case, as the relative volume increases, the mean error does not change significatively. In this case, a network with 15 neurons obtains the best results using a selection volume of 0.000003.

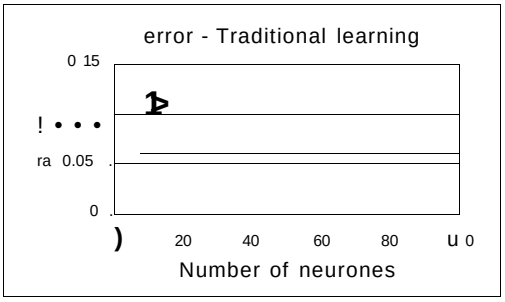

Fig. 4. Mean error on the test set for the Venice Lagoon time series achieved by different architectures trained with the whole training data set

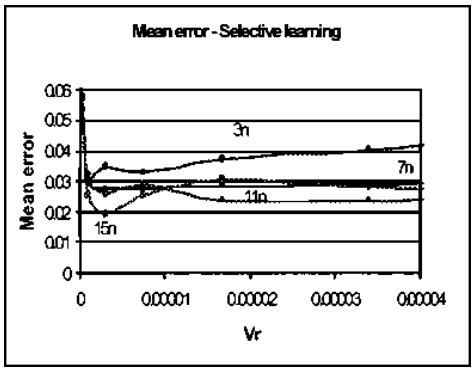

Fig. 5. Mean error on the test set for the Venice Lagoon time series achieved by different architectures trained with a selection of patterns using different selection volumes (Vr).

As in the previous example, in order to compare the selective learning method with the traditional one, mean errors over the test set obtained by both methods are shown in table 2 for the best architectures. As it can be observed in table 2, the mean error over the test set is reduced when the network is trained with a selection of patterns.

Table 2. Performance of different training methods for the piecewise function

	Mean error	Number of Neurons
Training with the whole data set	0 055	30
Training with a selection of data and V_r= 0 000003	0 019	15

As it is shown in figure 6, where the errors committed by the different learning strategies for each test pattern are shown, most of the test patterns are better approximated when the selective strategy is used. In this figure, errors corresponding to pattern 17 which represents the high water phenomenon in the test set, have been marked. The error when the network is trained in the traditional way is significantly higher than the corresponding to the selective learning method, when an appropriate selection of patterns is made.

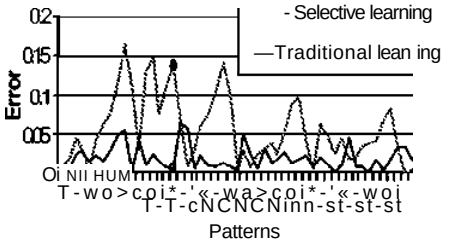

Fig. 6. The Venice Lagoon time series: Errors for each set patterns.

4 Conclusions

The results presented in the previous section show that if RJBNNs are trained with a selection of training patterns, the generalization performance of the network is improved.

The specific learning method proposed in this work involve storing the training data in memory, and finding relevant data to answer a particular test pattern. Thus, the decision about how to generalize is carried out when a test pattern needs to be answered constructing local approximations. That implies a large computational cost because the network has to be trained when a new sample test is presented. However, that is not a disadvantage of the method because in many cases that computational effort can be broached and to achieve lower approximation errors is an important advantage. Moreover, the number of neurons of the network trained with the selective method is much lower, thus, the computational effort is not as high as it appears to be.

The selection of the most relevant training patterns, the neighbours of the novel pattern, helps to obtain RJBNN's able to better approximate complex functions. On the order hand, since the volume grows exponentially with the dimension, a threshold distance or cut must be calculated using the relative volume.

The experimental results show that the performance of the RBNN does not depend significatively on the relative volume obtaining similar mean errors by different architectures and different relative volumes. However, this behaviour does not occur when the relative volume is smaller than a certain value because a miaimumamount of data is necessary to train the network appropriately.

References

[1] Abu-Mostafa Y. S.: The Vapnik-Chervonenkis dimension: information versus complexity in learning. *Neural Computation* 1, (1989), 312-317.

[2] Cohn D., L. Atlas and R. Ladner: Improving Generalization with Active Learning, *Machine Learning,Vo* 15,(1994), 201-221.

[3] Vijayakumar S. And H. Ogawa: Improving Generalization Abolity through Active Learning. IEICE *Transactions on Information and Syterns.* Vol E82-D,2, (1999), 480487.

[4] J. M. Vails, P. Isasi and I. M. Galvan: Deferring the learning for better generalization in Radial Basis Neural Networks. *Lecture Notes in Computer Science 2130. Artificial Neural Networks* - ICANN 2001, 189-195.

[5] Atkeson C. G., A. W. Moore and S. Schaal. *Locally Weighted Learning. Artificial Intelligence Review* 11, (1997), 11-73.

[6] Wettschereck D., D.W. Aha and T. Mohri: A review and Empirical Evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review* 11, (1997), 273-314.

[7] A. Tomasin 1973, A computer simulation of the Adriatic Sea for the study of its dynamics and for the forecasting of floods in the town of Venice. *Comp. Phys. Comm.* 5, 51.

[8] Vittori G.: On the Chaotic features of tide elevation in the lagoon Venice. In Proc. of the ICCE-92, *23rd International Conference on Coastal Engineering,* 4-9 (Venice 1992), 361-362.

[9] Bergamasco L. M. Serio A.R. Osborne and L. Cavaleri: Finite correlation dimension and positive Lyapuniv exponents for surface wave data in the Adriact sea near Venice. Fractals 3, (1995) 55-78.

[10] Zaldivar J.M., E. Gutierrez, I.M. Galvan, F. Strozzi and A. Tomasin: Forecasting high waters at Venice Lagoon using chaotic time series analysis and nonlinear neural networks. *Journal of Hydroinformatics,* 02.1, (2000), 61-84.