A better selection of patterns in lazy learning radial basis neural networks

P. Isasi[1], J.M. Valls[2] and I.M. Galván[2]

Carlos III University of Madrid, Computer Science Department. Avd Universidad, 30, 28911, Leganés, Madrid
[1]isasi@ia.uc3m.es
[2]{jvalls, igalvan}@inf.uc3m.es

Abstract. Lazy learning methods have been proved useful when dealing with problems in which the learning examples have multiple local functions. These methods are related with the selection, for training purposes, of a subset of examples, and making some linear combination to generate the output. On the other hand, neural network are eager learning methods that have a high nonlinear behavior. In this work, a lazy method is proposed for Radial Basis Neural Networks in order to improve both, the generalization capability of those networks for some specific domains, and the performance of classical lazy learning methods. A comparison with some lazy methods, and RBNN trained as usual is made, and the new approach shows good results in two test domains, a real life problem and an artificial domain.

1 Introduction

Lazy learning methods [1,2,3] are conceptually straightforward approaches to approximating real-valued or discrete-valued target functions. These learning algorithms defer the decision of how generalize beyond the training data until a new sample or instance is encountered. When a new sample is received, a set of similar related patters is retrieved from the available training patters and used to approximate the new query sample. Similar patterns are chosen using a distance measured with nearby points having high relevance.

Lazy methods that appear in the literature generally work by selecting the k least distant input patters from the novel samples, often in terms of Euclidean distance. Afterwards a local approximation using the selected samples is carried out with the purpose of generalize the new sample. That local approximation can be constructed using different strategies. The most basic form is the k- nearest neighbor method [4]. In this case, the approximation of the new sample is just the most common output value among the k selected examples. A refinement of this method, called weighted k-nearest neighbor [4], can be also used, which consists of weighting the contribution of each of the k neighbors according to the distance to the new sample, giving greater weight to closer neighbors. Other strategy to determine the approximation of the new sample is the locally weighted linear regression [2] that constructs an explicit and

linear approximation of the target function over a region around the new sample. The coefficients regression is based on the k nearest input patterns to new sample.

When lazy learning techniques are used, the target function is represented by a combination of many local approximations constructed in the neighborhood of the new samples. On the other hand, eager learning methods construct global approximations and the generalization is carried out beyond the training data before observing the new sample. That global approximation over the training data representing the domain could lead to poor generalization properties, mainly if the target function is complex. In these cases, lazy methods could be appropriate because the complex target function could be described by a collection of less complex local approximations.

Artificial neural networks can be considered as eager learning methods because they construct a global approximation that covers the entire sample space and all future samples. Although Radial Basis Neural networks (RBNN) [5,6] use multiple local approximations, they are also eager leaning methods because the network must commit to the hypothesis before the query point is known. The local approximations they create are not specially targeted to the query point to the same degree as in a lazy learning methods. Instead, RBN networks are built eagerly from local approximations centered around the training samples o around clusters of training samples, but not around the unknown future query point. That could contribute to poor generalization properties of RBNN.

The goal of this study is to improve the generalization capabilities of RBNN using a lazy learning strategy. In this context, the most basic strategy should consist of constructing local non-linear regression based on RBNN, this is, on approximating the target function in the neighborhood surrounding the new sample using a RBNN. In this case, parameters of the network –centers, width and weights- should be determined using the k nearest training patterns. However, the idea of selecting the k nearest patterns might not be the most appropriate mainly because of one factor: the network will always be trained with the same number of training data for each new sample. That may be a disadvantage in the context of artificial neural networks because each new sample could require different training data.

In this paper a lazy learning strategy is proposed to improve the generalization capability of RBNN. The main idea is to recognize from the whole training data set, the most similar patterns to each new pattern to be processed. This subset of useful patterns is used to train a RBNN and the training is deferred until a test pattern is received. The patterns retrieved from the available training data set to train the RBNN and posterior approximation of the new sample are determined by using the inverse of the Euclidean distance and a threshold distance or cut, which determines the extension of the neighborhood of the novel pattern. The number of retrieved patters will depend on the localization of the new sample in the input space.

The proposed lazy learning strategy to train RBNNs is validated in different domains and compared with other lazy methods, as k-nearest neighbor, weighted k-nearest neighbor, locally weighted linear regression and locally non-linear regression based on RBNN. The proposed method is also compared against the traditional way of training RBNN that uses the complete training data and construct a global approximation of the target function.

2 Lazy learning for Radial Basis Neural Networks

The lazy learning method studied in this work to train RBNN consists of selecting, from the whole training data, 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 only include, and one or more times, those near patterns -in terms of Euclidean distance- to the novel sample. This way, 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.

The amount of training data to be selected to approximate the new sample encountered is given by a relative n-dimensional volume, named V_r, surrounding the test pattern, where n is the dimension of the input space. This relative volume V_r is a fraction of the total volume from where training patterns are selected. In order to determine the training patterns included in that relative volume, a threshold distance or cut r_r (radius of the sphere) must be calculated before the learning algorithm is applied. The relative n -dimensional volume can be written as:

$$V_r = \frac{V_s}{V_{max}}$$

where V_s is the volume of the sphere centered in 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 in the test pattern and radius equals to the maximum distance, r_{max}. Since $V=kr^n$ where r is the radius of the sphere, n is the dimension of the space and k is a constant, we can write:

$$V_r = \frac{kr_s^n}{kr_{max}^n} = r_r^n \qquad \text{Hence:} \qquad r_r = \sqrt[n]{V_r}$$

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 around the test pattern.

Given a test pattern $\boldsymbol{q}$, described by a n-dimensional vector, $\boldsymbol{q}=(q_1,..., q_n)$, the steps to select the training set, named X_q, associated to the patterns, are the following:

Step 1. A real value, d_k, is associated to each training pattern (x_k, y_k). That value is defined in terms of the standard Euclidean distance as:

$$d_k = d(x_k, q) = \sqrt{\sum_{i=1}^{n} (x_{ki} - q_i)^2}$$

Step 2. A relative distance, d_{rk}, is calculated for each training pattern. Let d_{max} be the maximum distance to the novel pattern $\boldsymbol{q}$, this is $d_{max} = Max\ (d_1, d_2, ...d_N)$. Then, the relative distance is given by:

$$d_{rk} = d_k / d_{max}$$

Step 3. A new real value, $f_k=1/d_{rk}$, where $k=1,...,N$ is associated to each training pattern (x_k,y_k). These values f_k are normalized in such a way that its sum is equal to the number of training patterns in X. The relative values, named as fn_k, are obtained by:

$$fn_k = \frac{f_k}{S} \quad \text{where} \quad S = \frac{1}{N}\sum_{k=1}^{N} f_k \qquad \text{Thus:} \sum_{k=1}^{N} fn_k = N$$

Step 4. The relative distance, d_{rk}, calculated in step 2 and the relative threshold distance, r_r, previously calculated as the n-th root of the relative volume V_r, are used to decide if the training pattern (x_k, y_k) is selected:

If $d_{rk} < r_r$ then the pattern (x_k, y_k) is included in the training subset.

Value fn_k calculated in step 4 is used to indicate how many times the training pattern (x_k, y_k) is going to be repeated in the new training subset. Hence, they are transformed into natural numbers as:

$$n_k = \text{int}(fn_k) + 1$$

At this point, each training pattern in X that has been selected has an associated natural number, n_k, which indicates how many times the pattern (x_k, y_k) has been used to train the RBNN when the new instance $\boldsymbol{q}$ is reached.

Step 5. Once the training pattern subset associated to the test pattern q, X_q, is built up, the RBNN is trained with this new subset. As usual, 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 [7]. After that, the dilations coefficients are calculated as the square root of the product of the distances to their two 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 Validation

The lazy learning method described in section 2 has been applied to RBNN and the generalization capability of the network has been measured in terms of the mean error over the test data set. Different domains have been used with that purpose. The results obtained with that lazy strategy have been compared, firstly, with the results provided by the network when a global approximation over the whole training data set is constructed, this is when the network is trained as usual. Secondly, the results are also compared with other lazy methods, as k-nearest neighbour, weighted k-nearest neighbour, the weighted local linear regression and a weighted local nonlinear regression methods.

In this section, the features of different domains and the conditions of the experiments carried out are described. Finally, the results obtained with the different lazy learning methods are presented and compared.

3.1 Experimental definition

Different domains have been used to compare the different lazy strategies: one-dimensional theoretical approximation problem, a piecewise-defined function, and a n-dimensional (n>1) real life problem, defined by means of a time-series describing the behaviour of the water level at Venice Lagoon. In the next, the characteristics of both of them are presented.

- **Theoretical problem: A piecewise-defined function approximation**
  The function is given by the equation:

$$f(x) = \begin{cases} -2.186x - 12.864 & \text{if} \quad -10 \le x < -2 \\ 4.246x & \text{if} \quad -2 \le x < 0 \\ 10e^{(-0.05x-0.5)} \sin[(0.03x + 0.7)x] & \text{if} \quad 0 \le x \le 10 \end{cases}$$

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]

- **Real life 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 [8].

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. A nonlinear model using the six previous sampling times seems appropriate because the goal is to predict only the next sampling time.

3.2 Experimental conditions

As it has been previously mentioned, different lazy learning strategies have been used to deal with different problems. Now, the conditions of the experiment run are described. The k-nearest neighbour, the weighted k-nearest neighbour and the local weighted linear regression methods [2] have been run for different values of k parameter (number of patterns selected). For the piecewise- defined function, k is varied from 1 to 23 and for the prediction of the water level at Lagoon Venice k is varied form 1 to 75, because more data are available.

A local nonlinear regression method based on RBNN is also tested. In this case, when the test sample is encountered the k nearest input training patterns are retrieved and a RBNN is trained with those patterns. The initial centers of the RBNN are fixed around the geometric medium of the k training patters selected, and the training of the network is carried out as usual. The value of k is also varied in the same range that for the other lazy methods, but in this case the k value is incremented by 5 units. RBNNs with different number of hidden neurons have been proven. After several experiments it has been verified that the number of hidden neurons depends on the value of k, for instance int(k/r)+1, where r=2,3,4,5.... In this work, the best results are obtained using int(k/2)+1 hidden neurons, although significant differences do not exist.

The lazy learning method described in section 2 is used to train RBNNs. In this case different relative volumes (*Vr*) have been used to run the experimental simulations. As in the previous case, the results do not depend significantly on the number of

hidden neurones. The results shown in the next section are obtained using 19 hidden neurones for the piecewise-defined function and 15 hidden neurones for the time series prediction domain. Finally, by comparative reasons, the RBNN have also been trained as usual, this is, the network is trained using the whole training data set. When the lazy strategy involves the use of a RBNN, the training is carried out until the convergence of the network is reached, that is, when the derivative of the training error is near zero.

3.3 Experimental results

Figure 1 shows the behaviour of the different lazy strategies in both studied domains. The mean error over the test data sets for each value of k is represented. In figure 2, the behaviour of the lazy strategy proposed in section 2 to train RBNN is shown. In this case, the mean error over the test data is evaluated for every value of relative volume.

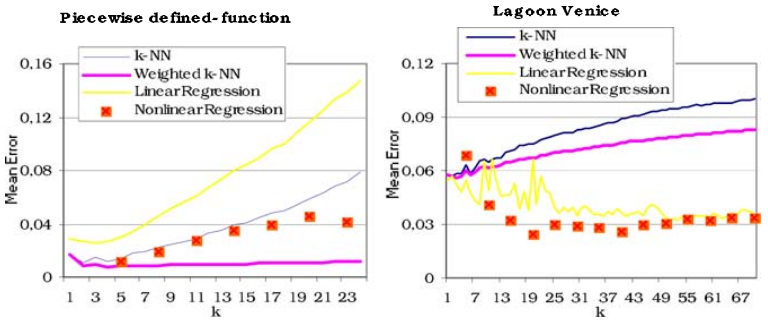

Figure 1. Evolution of the test mean errors for k-NN Weighted k-NN, linear and nonlinear local regression methods

In figure 1 it is observed that for the piecewise-defined function, the performance of classical lazy strategies is in general influenced by the value of k, increasing the error as the k value increases, although when a nonlinear local approximation is made, that influence is smaller. However when a weighted method is used, weighted k-NN, the influenced of the parameter k almost disappears, as expected. Moreover the error decreases reaching the best values of all traditional methods.

For the prediction problem, the behaviour of the traditional methods is very different. On the one hand, there is a stabilization of the error after a certain value of k, but only when using regression methods. On the other hand, however for the k-NN algorithms the error has a worse behaviour. It increases when k increases, and even the best results, in k, are very bad. This is due to the local influence of data in the Venice lagoon domain. Any way, it seems clear that the influence of the domains in the results when using traditional lazy approaches is very high.

The performance of the lazy method proposed in this work (see figure 2) does not depends significantly on the value of relative volume for both domains. In this case, the mean error is more or less the same for every relative volume, once a certain

amount of data are selected. It can also be seen that similar results can be found in both domains, the proposed method seems to be less domain dependent. The new method takes advantage of the property of selecting a different set, in number and elements, of learning patterns. It is able to discover the most appropriate set of examples for each new test pattern.

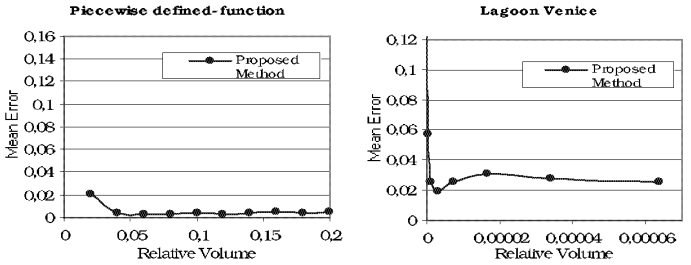

Figure 2. Evolution of the test mean errors for the proposed lazy method to RBNN

The best mean errors in test obtained by the different methods and the different domains are shown in tables 1 and 2. Table 2 shows also the mean error obtained when a global approximation of the target function using RBNN is made. As it is observed in both tables, the mean error over the test set in all domains is significantly reduced when the lazy strategy proposed in this work is used. When k patterns are selected and a nonlinear local regression is made, the results do not improve with respect to linear regression. However, when the training patterns are selected based on the new test sample received, better generalization capabilities are obtained. In addition, the RBNN has poor generalization capabilities if the network is trained with the whole training patterns, that is when a global approximation is built up.

Table 1. Best mean error for k-NN, Weighted k-NN, linear and nonlinear local methods

Mean Error (k value)	k-nearest neighbor	Weighted k-nearest neighbor	Local linear regression	Local nonlinear regression based on RBNN
Piecewise-defined function	0.01113 (k=2)	0.00793 (k=4)	0.02513 (k=3)	0.01120 (k=5)
Lagoon Venice	0.05671 (k=2)	0.05610 (k=3)	0.0323 k∈[50,70]	0.02334 (k=19)

Table 2. Best mean error for the proposed lazy method to RBNN

Mean Error (relative volume)	Lazy method	Traditional method
Piecewise-defined function	0.002085	0.0396
Lagoon Venice	0.019	0.055

4 Conclusions

The idea of representing the target function by a combination of many local approximations constructed in the neighbour of the new sample could provide better performance of those learning methods than construct global approximation, mainly if the target function is complex. However, the performance of lazy methods is influenced by the criterion of selecting the patterns that determine each local approximation. When local approximation are built up using the lazy strategies based on the selection of k patterns -as the k-nearest neighbour, weighted k-nearest neighbour and locally weighted regression- for every test sample, the same amount of patterns are selected. The results presented in the previous section show that those lazy strategies have poor generalization capabilities when approximation and prediction problems are formulated, even if nonlinear local approximations are made.

The lazy method presented in this work makes an automatic selection of training patterns for each test samples allowing that number of training patterns is variable depending on the position in the input space. In addition, the performance of the lazy method presented in this work is higher than those lazy strategies selecting the k nearest patterns.

On the other hand, the lazy strategy presented in that work to train RBNN improve the generalization capabilities of those type of artificial neural network. In this paper is also shown that the selection of the most relevant training patterns, the neighbours of the novel pattern, helps to obtain RBNN's able to better approximate complex functions.

5 References

[1] Aha D., D. Kibler and M. Albert. Instanced-based learning algorithms. Machine Learning, 6, (1991), 37-66.

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

[3] 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.

[4] Dasarathy B.V. (Ed.). Nearest neighbor(NN) norms: NN pattern classification techniques. Los Alamitos, CA: IEEE Computer Society Press. (1991)

[5] Moody J.E. and Darken C.J.: Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computation 1, 281-294, 1989.

[6] Poggio T. and Girosi F.: Networks for approximation and learning. Proceedings of the IEEE,78, 1481-1497, 1990.

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

[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.