ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

Trimming the Inputs of RBF Networks

Colin ANDREW[1], Miroslav KUBAT[2], and Gert Pfurtscheller[1]

[1] Department of Medical Informatics, Institute of Biomedical Engineering, Graz University of Technology, Brockmanngasse 41, A-8010 Graz, Austria.

[2] Institute for Systems Science, Johannes Kepler University in Linz, Altenbergerstr. 69, A-4040 Linz, Austria.

Abstract

The performance of radial basis function (RFF) networks is compared to alternative learning algorithms in the domain of automatic EEG-data classification. In spite of good recognition performance, RBF networks can suffer from the presence of irrelevant features in the input vector. Here, a search-based approach to the problem of automatic selection of inputs to hidden-layer neurons is proposed.

1 Introduction

Radial-basis-function (RBF) networks have been widely accepted as a powerful means for function approximation and for automatic classification. They build on the theorem proved in Cover (1965), stating that a complex pattern-classification problem has a higher chance of becoming linearly separable if it is non-linearly mapped to a space of higher dimensionality. This idea is implemented by the network in Figure 1. The hidden-layer neurons carry out the nonlinear mapping, and the weights of the output-layer neurons (with linear transfer functions) define the separating hyperplane.

The theoretical and practical considerations of Powell (1988) show that the type of the non-linearity in the hidden neurons is not crucial to the overall performance. However, the following gaussian function is prevalent:

$$\varphi_j(\mathbf{x}) = \frac{1}{2\pi\sigma_\mathbf{j}^2} \exp(-\frac{\|\mathbf{x} - \mu_\mathbf{j}\|^2}{\sigma_\mathbf{j}^2}) \qquad (1)$$

where $\mathbf{x}$ is the input vector, $\mu_\mathbf{j}$ is the center of the j-th gaussian and $\sigma_\mathbf{j}$ is its standard deviation. The activation (output value) of the i-th output neuron is given by the function $y_i = F(\mathbf{x}) = \Sigma_{j=0}^{m} w_{ij}\varphi_j(\mathbf{x})$, where w_{ij} is the weight of the link from the j-th hidden neuron to the i-th output neuron. The weights w_{0j} are connected to a bias with the fixed value of 1. The network is defined by the output-layer weights, by the number and locations of the gaussian centers $\mu_\mathbf{j}$, by the respective standard deviations $\sigma_\mathbf{j}$, and by the dimensionality of input vectors.

Park and Sandberg (1991) proved that a RBF network with enough hidden-layer neurons can approximate any continuous function with arbitrary accuracy.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

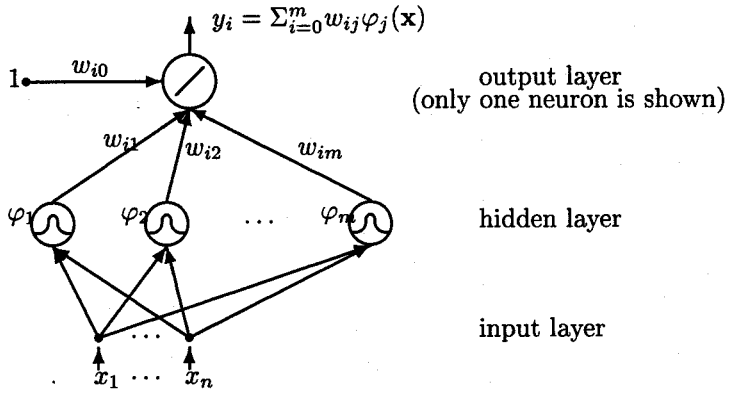

Figure 1: Radial-basis function network

In classification tasks, each output neuron represents one class. If i is the index of the output neuron with the highest activation, then the input vector $\mathbf{x}$ is assigned the i-th class.

In realistic implementations, the centers μ_j are often positioned at the coordinates of properly selected input vectors, and this implies two important questions: which of the input vectors are sufficiently representative to be chosen as centers, and which of the features are relevant for the classifications? While the former issue received considerable attention in the literature, the latter has largely been neglected. For this reason, we supplemented an existing RBF learning algorithm with a simple technique capable of trimming those features that are considered as having only insignificant impact. The motivation is to reduce the computational time needed for classification tasks. The price is increased computational costs of learning and slightly degraded accuracy of the classifier.

We compare the performance of the program to a few other learners on a difficult recognition problem from the domain of automatic classification of EEG signals. The referential systems include the nearest-neighbour classifier, LVQ (Kohonen, 1990), and the machine-learning programs C4.5 (Quinlan, 1993), and CN2 (Clark and Niblett, 1989). Experiments indicate that RBF networks compare very favourably to the other systems, the reasons for which are detailed in the last section.

2 Selecting Representative Pieces of Data

We briefly revise the previous work and then describe our own contribution.

As for the positioning of the hidden neurons, we follow the recommendation of Poggio and Girosi (1990), placing the gaussian centers at the coordinates of the individual input vectors and understanding the task as an interpolation problem. Interpolation, in its *strict* sense, is stated as follows: 'Given a set of input and output vectors, find a function $F : R^n \rightarrow R^p$ such that for each input vector $\mathbf{x_i} = \{x_{i1}, \ldots, x_{in}\}$ and the desired output vector $\mathbf{d_i} = \{d_{i1}, \ldots, d_{ip}\}$ we

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

have $F(\mathbf{x_i}) = \mathbf{d_i}$.' By proper parameter setting, the network should reasonably generalize to vectors that have not been used for the network definition.

Two aspects of the available training data can have detrimental effect on the quality of the resulting network: improperly selected vectors to serve as gaussian centers, and improperly chosen features.

If the input vectors are noisy, the strict interpolation will overfit them. Hence, a few techniques for the selection of a proper subset of input vectors defining the network have been suggested. Lowe (1989) uses a random subset but this works only in simple tasks; Cheng and Lin (1994) apply a search technique with operators 'add a neuron' and 'delete a neuron'; and Chen, Cowan, and Grant (1991) apply only the single operator 'add a neuron' and use the orthogonal least square criterion to select the next candidate[1].

The problem of the feature selection has been attacked by Sanger (1991), where a tree-like structure of RBF-neurons (in multiple hidden layers) has been suggested. However, this approach seems to produce too large trees as the number of features grows. As an alternative, we implemented a system that applies hill-climbing search to reduce the number of inputs. As a basis of the algorithm, we used the orthogonal least square learning of Chen, Cowan, and Grant (1991) to generate, one by one, the hidden neurons. However, we supplemented it with a module to cut off those features that are considered as irrelevant.

This is accomplished by a hill-climbing mechanism that, at each step, searches for the features whose deletion maximally reduces the mean square error (MSE) on the training data. The process terminates when the next single trimming would increase MSE.

Algorithm.

1. Select the input vector by the orthogonal least square method and use it to create a new hidden neuron;
2. Apply the hill-climbing algorithm to trim input features off this neuron, one by one, as long as the mean square error does not increase;
3. Use the pseudomatrix approach to determine the output-layer weights minimizing the mean square error;
4. If the termination criterion is satisfied, stop; otherwise goto 1.

The methodology of pseudoinverse matrices as described in Duda and Hart (1973) was used to set the output-layer weights; proper values of σ_j as well as the number of hidden neurons were determined empirically.

3 Case Study

We applied the above algorithm for the classification of EEG signals. Input vectors involve 44 features, representing estimated signal powers at 11 electrodes at 4 time slices. The signals are pre-classified into four different classes. The

[1] An alternative method for adding one neuron at a time with subsequent *tuning* of the centers was presented by Fritzke (1994).

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

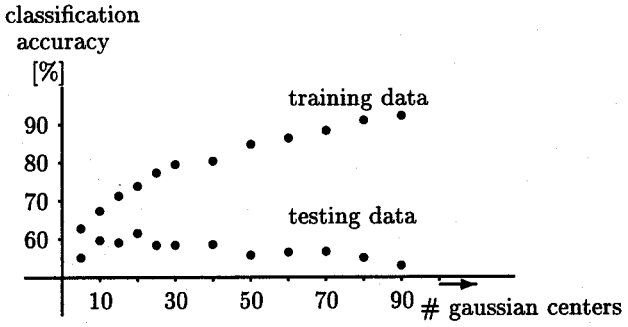

Figure 2: The problem of overfitting in the EEG domain

learner's task is to use a subset of the vectors (*training* vectors) and develop from them internal knowledge that will facilitate the classification of the remaining (*testing*) vectors. The neurophysiological interpretation of the results is, for the needs of this paper, irrelevant. Three data sets were used, referred to as A4M4, B6M2, and B8M2. The sizes of the sets are 318, 226, and 177 vectors. All classes were roughly equally represented.

For several reasons, the task just outlined represents a very difficult learning problem. First, the input vectors contain many features whose relevance for the classification is insignificant but no apiori information about feature relevance exists; moreover, the available features do not suffice for reliable classifications. Second, the decision surface has a very complicated shape because of the many hidden interrelations of the individual features. Third, the input vectors are noisy. Fourth, the number of training vectors is small considering the dimensionality of the problem. Fifth, it is a well-known phenomenon of EEG signals that their quality crucially depends on external factors and, hence, the measurements are, to a certain degree, inconsistent.

First of all, we had to determine the number of gaussians μ_j and the standard deviations σ_j. To this end, we carried out a single run on one of the data sets. A subset of 60% vectors has been used for training, the rest being used for testing. The results are depicted in Figure 2. Note that the accuracy measured on the training vectors increases with the growing number of hidden neurons. However, the accuracy on the testing data will decrease after reaching the maximum by about 15–25 hidden neurons. Following this experience, we set the number of hidden neurons in all subsequent experiments to 20. Standard deviations of the gaussians have been determied similarly. No feature trimming was applied during these preliminary experiments.

Table 1 summarises the performances of various systems on the three data sets. The results represent averages over 20 runs on different splittings of the data into training and testing sets. In each run, 60% of the vectors were used for learning and the remaining 40% of the vectors were used for testing. Column 1-NN shows the performance of the nearest neighbour classifier. CN2 is a public-domain machine-learning program distributed by P. Clark and T. Niblett. The decision-tree growing program C4.5 has been implemented by J.R. Quinlan and is distributed by Morgan Kaufmann. The results of LVQ were obtained using the package LVQPack. RBF is represented by two columns: RBF, which is

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

Table 1: Performance of various learners on three data sets

	1-NN	LVQ	CN2	C4.5	RBF	RBFt
A4M4	50.2	49.1	44.1	55.4	61.1	57.8
B6M2	42.1	41.8	37.2	50.1	54.9	53.0
B8M2	41.5	36.9	38.6	45.6	47.1	45.6

a reimplementation of the algorithm of Chen, Cowan, and Grant (1991), and RBFt, which is the same algorithm supplemented with component trimming.

The experiments have shown that the RBFt algorithm was capable of reducing the number of inputs into each hidden neurons by circa 60% on average, with only a slight reduction in performance.

4 Discussion

Explanation of the extent to which RBF outperforms well-established learning systems on the EEG data can be found in the pecularities of the domain as explained in the previous section. Thus, for instance, C4.5 creates decision surfaces that are piecewise linear and 'piecewise axis-parallel'. Further on, the system builds on the assumption that the features are pairwise independent, which is not the case of our data. The performance of C4.5 is reduced when the output class depends not only on the absolute values of the components but, rather, on their mutual relations. Similar considerations hold also for CN2.

LVQ, in turn, is able to discover linear interdependecies because it generates piecewise-linear decision surfaces. Even though this algorithm is known as a powerful classifier, its performance degrades if the training vectors are sparse and thus do not permit reasonable positioning of the code vectors. This was the case of our application, where the number of learning examples was small considering the dimensionality of the problem space and the number of classes.

Nearest-neigbour classifiers are known to suffer from the presence of noise. Also, sufficient number of vectors are necessary for good recognition.

Even though trimming irrelevant features somewhat worsened the results of RBF, the system stills outperform all other learners.

To summarize, we were interested in the performance of various systems from machine learning, pattern recognition, and neural networks on the difficult task of EEG classification. It turns out that RBF networks are powerful classifiers but are not able to prune out irrelevant features unlike, say, decision-tree growing algorithms. For this reason, we suggested a simple algorithm capable of trimming unnecesary features from the RBF inputs.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 291-296

Acknowledgements

The first author has been supported by a fellowship from the Foundation for Research and Development of South Africa. The experimental data have been measured in the framework of the Brain-Computer-Interface project sponsored by the agency 'Fonds zur Förderung der wissenschaftlichen Forschung.'

References

Chen, S., Cowan, C.F.N., and Grant, P.M. (1991). Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 2:302–309

Cheng, Y.-H. and Lin, C.-S. (1994). A Learning Algorithm for Radial Basis Function Networks: with the Capability of Adding and Pruning Neurons. *Proceedings of the IEEE*, 797–801

Clark, P. and T. Niblett (1989). The CN2 Induction Learning. *Machine Learning*, 3:261–283

Cover, T.M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*. EC-14:326–334

Duda R.O. and P.E. Hart (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York

Fritzke, B. (1994). Fast Learning with Incremental RBF Networks. *Neural Processing Letters*, 1:2–5

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78:1464–1480

Lowe, D. (1989). Adaptive Radial Basis Function Nonlinearities and the Problem of Generalization. *1st International Conference on Artificial Neural Networks*, pp.171–175, London, UK

Park, J. and I.W. Sandberg (1991). Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation* 3:246–257

Poggio, T. and F. Girosi (1990). Regularization Algorithms for Learning that are Equivalent to Multilayer Networks. *Science* 247:987–982

Powell, M.D.J. (1988). Radial Basis Function Approximations to Polynomials. *Numerical Analysis 1987 Proceedings*, pp.223–241, Dendee, UK

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo

Sanger, T.D. (1991). A Tree-Structured Adaptive Network for Function Approximation in High-Dimensional Spaces. *IEEE Transactions on Neural Networks*, 2:285–293