Radial Basis Functions Artificial Neural Networks and their Application to Sequence Recognition

Joël Toyigbé Hounsou[1], Michele Ceccarelli[2]

[1]*Institut de Mathématiques et de Sciences Physiques,*
Université Nationale du Bénin,
Porto–Novo, BENIN
B.P. 613 Porto–Novo, BENIN
[1]*International Institute for Advanced Studies, IIASS,*
Via G. Pellegrino 19,
84019 Vietri sul Mare (SA), ITALY
[2]*Instituto per la Ricerca sui Sistemi Informatici Paralleli, IRSIP-CNR,*
Via P. Castellino 111,
80131 Napoli, ITALY

June 2, 1996

Résumé

Une machine automatique apprenante et efficace, voir une sorte de machine comportant les aptitudes du cerveau animal constitue de nos jours un grand défi pour l'humanité. Certes, des Scientifiques s'y attelent sérieusement.

A travers ce qui suit, nous apportons notre contribution à ce type de machine, considérant plusieurs méthodes d'apprentissasge pour les réseaux à Fonction Radiale de Base (FRB) utilisés pour la reconnaissance de la parole, des vocabulaires isolés en particulier. Du fait, observant la dynamicité de la parole, des réseaux à connexions retardées et aux unités intégratives sont les plus indiqués. Enfin, nous comparons l'aptitude à la généralisation et les coûts machines de nos modèles à ceux des perceptrons multicouches. Nos études montrent que l'apprentissage supervisé des centroïdes des fonctions de base donne des résultats appréciables et à moindre coût.

Abstract

Efficient and automatic adaptative (learning) Machine, a kind of brain–like machine, is today a challenge to human being. Surely, Scientists are working hardly in the field.

In this paper we contribute to that effort of achieving this type of machine, considering several learning procedure for Radial Basis Function (RBF) Networks applied to a problem of speech recognition, namely isolated word recognition We do consider, owing the dymanic nature of speech, delayed connections and integration units to the network. The results obtained are compared on both generalization ability and computational costs with the ones of multilayer perceptrons. Our study shows that supervised learning of the centroids of the basis functions gives appreciable results at a significant lower cost.

1 Introduction

Since the mists (dawn) of time, speech is an efficient and reliable communication aid. It is shown that speech, as an acoustic wave, is propagated with high velocity, got round obstacles and altered less. The speech complexity makes it difficult to process as well for recognition or/and understanding. Cognitive tasks, such as vision, speech, language processing and motor control are characterized by a high degree of uncertainty and variability and it has proved difficult to achieved good performance for these tasks using standard serial programming methods. A suitable and attractive systems for these tasks might make use of multiple interacting constraints. In general, such contraints are too complex to be easily programmed and require the use of automatic learning strategies.
Therefore, massively parallel distributed processors models, such as "brain-like" models called artificial neural networks (ANN) are interesting in the light of their numerous interrelated constraints to tacle these problems of cognitive tasks, namely speech recognition task.
Till today, despite the availability of high speed parallel or non supercomputers the training procedures costs of (ANN) limit their applications to natural problems. Our slight contribution to the topic is to compare in terms of computational costs and generalization results on landmark applications learning algorithms. In this paper, we consider time delayed Radial Basis Functions (RBF) neural networks [Broo 88, Mood 89] for sequence recognition. It is our aim to compare the learning procedures of RBF networks in term of generalization abilities and learning time on a specific speech recognition task.These experiments show that the simplicity of the learning rules available for RBF neural networks makes them preferable, extending the results already achieved in [Cecc 93, Cecc 94].

2 General Considerations

2.1 Database Description and Preprocessing

The database we used is composed of italian vocabularies. For this task of word recognition issue, the database is the ten first italian digits uttered by 64 native female and male speakers (28 speakers for training and 36 rests

for test). Training speakers spoken three times the ten digits, while the test speakers produce ten repetitions of all the digits; this set is about 4440 isolated words.
An high fidelity noise reduction microphone was used in recording room consisting in an ordinary office. The silent portions at the beginning and the end of each utterance were removed by using an energy-based tresholding algorithm. The preprocessing and feature extraction phase is performed on a Personal Computer equipped with the DIVA board (ALCATEL Face) wich extracts in real-time the mel-cepstral parameters described above. The neural network learning algorithms, programmed in C language, run on a standard UNIX workstation (SUN Microsystems Sparcstation 2)

2.2 Models architecture

Basically, Radial Basis Functions (RBF) networks are considered throughout these task. The main idea behind RBF networks is the construction of complex decision regions by superposition of simple kernel functions. During the learning phase the centers and the widths of the kernel functions are the main parameters to be estimated. The typical choice of kernel function is a Gaussian which gives the highest output when the input is near its center and monotically decreases as the distance from the center increases.
Indeed, RBF networks were originally introduced as a means of function interpolation [Broo 88]. Confident of the results obtained by Cybenko [Cybe 89], for sigmoid hidden units i.e., two-layers perceptrons can uniformly approximate any continuous function, Park and Sandberg [Park 91] and Hartman *et al* [Hart 90] independentely extended these results to RBF networks where each output computes the following set of M approximating functions constructed from a set of L basis functions $\phi(x)$:

$$f_i(x) = \sum_{j=1}^{L} w_{ij}\phi_j(x) \quad 1 \leq i \leq M \tag{1}$$

where M is the number of output units and the ϕ_j's are L kernel functions computed by the hidden units. Their form is generally a Gaussian:

$$\phi_j(x) = kexp\left(-\frac{1}{2}(x-m_j)^T\Sigma_j^{-1}(x-m_j)\right)$$

where k is a normalising constant and Σ_j is a covariance matrix which describes the contour of a multidimensional ellipsoid, it is frequently assumed to be diagonal $\Sigma_j = \sigma I$, where σ is a constant smoothing factor equal for all hidden units which measures the radius of a multidimentional sphere, and I is the identity matrix. The vector m_j represents the input stimulus to which the unit j is maximally receptive, i.e., it is the centroid of the kernel function ϕ_j.

Another important peculiarity of RBF neural networks is their use in maximum likelihood classification. Indeed, as the Parzen Window analogy suggests, the role of the function f_i is the approximation of the class conditional probability density function by means of the potential functions method. With this classification scheme an unknown input sample is classified according to the index of the potential function which have the maximum value.

In the order to achieve the above goals several learning procedures can be used. The main points are the number and the location of the centroids. In [Spect 90] a kernel function is placed on each training pattern and the w_{ij} weights are fixed by solving a linear system. While an exact fitting of the training set is obtained with this method, overfitting [Bish 91, Gema 92] and memory/processing requirements can be a problem.

In concret terms, the number of the kernel centers must be reduced by clustering the training data and the selection of the kernel centers as the cluster code vectors [Broo 88, Mood 89, Reyn 92, Beng 92, Chou 92, Musa 92]. Apart from the algorithm adopted, the most important question is whether the clustering must be super- or unsuper-vised. In the first case the clusters will contain data points belonging to the same class and the kernel centers will be 'representative samples' of a given class. Whereas, for the second case the clustering method produces a self-organising quantization of the input space. This process tries to automatically discover the features of the input data, and the class-membership information is used only in teh upper layer.

The last step of the learning procedure for RBF networks involves the computation of the hidden-to-output weights. With LMS training one has:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \left[d_i - f_i(x(t))\right] \phi_j(x(t)) \qquad (2)$$

where d_i is the desired activation of the $i-th$ output unit for the pattern $x(t)$. Other method, based on pseudo-inverse, can also be used.

2.2.1 Unsupervised learning of center locations

Classical unsupervised clustering algorithms can be used for the computation of center locations. The *K-means* [Spat 80] algorithm and the *Self-Organizing Feature Map (SOFM)* [Koho 90] are commonly used for this task. The SOFM algorithm has the propertiy of creating a topologically ordered codebook, i.e., the spatial proximity between units implies proximity between the corresponding code vectors. The SOFM algorithm can be outlined as follows: given a set of L neural cells, organized as a bidimensional grid, with weight vectors $m_1(t), m_2(t), \ldots, m_L(t)$, for each input sample $x(t)$ select the nearest weight vector to it, $m_i(t)$, and apply the following rule:

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \quad if\ i \in N_l(t) \tag{3}$$

$$m_i(t+1) = m_i(t) \quad i \notin N_l(t) \tag{4}$$

where $0 < \alpha(t) < 1$ is the learning rate and $N_i(t)$ is the neighbourhood of the neuron l of radius $\gamma(t)$. The learning rate $\alpha(t)$ and the neighbourhood radius $\gamma(t)$ are slowly decreasing functions. The SOFM model is inspired by the cortical maps of animal brains and has been widely studied both from analytical and applicability point of view [Koho 90]. The parameters $\alpha(t)$ and $\gamma(t)$ play an important role in the convergence of the algorithm and the topological ordering. Typical choice are $\alpha(t) = \alpha_0 e^{-t/\alpha_1}$ and $\gamma(t) = \gamma_0 e^{-t/\gamma_1}$ with $\alpha_0, \alpha_1, \gamma_0, \gamma_1$ suitable constants. For example, in order to guarantee the topological ordering, the value of γ_0 must be more than half of the size of the grid and γ_1 must ensure a slow decrease in the initial phase and a faster decrease once the map is ordered (e.g., $500 \leq \gamma_1 \leq 2000$ is a good range of values). Similar considerations apply to the parameters α_0 and α_1. If α_0 value is too small, it could remarkably slow down the learning process, whereas a too high value could lead to the divergence of the algorithm; the range of values $0.05 \leq \alpha_0 \leq 0.5$ is appropriate for many real applications.

2.2.2 Supervised learning of center locations

The supervised version of SOFM is known as *Learning Vector Quantization (LVQ)*. It does not take the neighbourhood interaction into account, i.e., the updating is carried out just for the winner neuron, except that the class

membership of the input pattern is used to establish the sign of the updating term in (3b).

Specifically, each hidden unit is labelled with a class index, and the updating rule for the training pattern $x(t)$ is the following. Let $l = arg\min_l\{||x(t) - m_i(t)||\}$ then

$$m_l(t+1) = \begin{cases} m_l(t) + \alpha(t)[x(t) - m_l(t)] & \text{if Class}(m_l)\text{=Class}(x(t)) \\ m_l(t) - \alpha(t)[x(t) - m_l(t)] & \text{if Class}(m_l)\neq\text{Class}(x(t)) \end{cases} \quad (5)$$

$$m_l(t+1) = m_i(t) \quad \text{for } i \neq l \quad (6)$$

The learning rate $0 < \alpha(t) < 1$ may be constant or a slowly decreasing function as before. The above algorithm and its variants have been successfully applied to many classification tasks.

The asymptotic values of vectors $m_i(t)$ resulting from this procedure will be the center locations of the RBF hidden nodes; they can be considered as representative of the classes to be discriminated.

2.2.3 Kernel width

Once the centroid of the kernel functions have been computed the structure of the covariance matrices σ_j allows to change the overlapping between kernels belonging to different classes. An efficient method for kernel widths estimation can be found in [Musa 92]. It uses the Gram-Schmidt orthogonalization to find the eigenvectors and the eigenvalues of the covariance matrix, representing respectively the principal axes of a multidimensional ellipsoid and their lenghts. However, when unsupervised kernel center computation is performed Σ_j is a diagonal matrix of the form $\Sigma_j = \sigma_j I$. If T is the number of training examples and f_j is the winning frequency of neuron j at the end of the clustering procedure, the smoothing factor σ_j could be estimated with the following rule [Chou 92]:

$$\sigma_j = \sigma_0(1 - f_j/T)f \quad 1, \ldots, L$$

where σ_0 is a constant. Chou and Chen [Chou 92] propose to choose $\sigma_0 = \sqrt{L}$.

2.2.4 Generalised radial basis function

Another learning scheme for RBF networks is the so called Generalised Radial Basis Functions (GRBF) [Pogg 90], i.e., the use of gradient descent method for supervised learning of center locations. Given an input vector $x = (x_1, \ldots, x_N)$ and the corresponding desired output vector $d = (d_1, \ldots, d_N)$ the updating formulae for this algorithm look like:

$$\Delta w_{ij} = \eta(d_i - f_i(x))exp\frac{||x - m_j||}{\sigma_j} \quad (7)$$

$$\Delta m_{jk} = \eta\frac{2(x_k - m_{jk})exp\frac{||x-m_j||}{\sigma_j}}{\sigma_j}\sum_{m=1}^{M}(d_m - f_m(x))w_{mj} \quad (8)$$

This method has been applied to speech recognition by Wu and Chan [Chan 91] obtaining no appreciable improvement of performance with respect to classical multilayer perceptrons with sigmoid activation units. Indeed, the long learning times do not allow large networks to be used, and the advantages of easy learning procedures carried out layer by layer are completely lost.

3 Results and comparisons

In many speech recognition systems, a large discrepancy is found between the training procedure and the testing procedure. The training criterion, generally Maximun Likelihood Estimation, is very far from the word accuracy the system is expected to maximize.
The considered networks have a first RBF hidden layer, a second layer for input data categorization built with sigmoid-activation nodes, and last an output layer integrating over time the partial classification performed.

3.1 Unsupervised clustering of data

In the first experiment the kernel centroids, m_i, were computed using the SOFM algorithm as clustering tool, and the weights w_{ij} were adjusted with the LMS algorithm. The parameters σ_i were considered constant, because

Table 1: The recognition performance of an RBF network trained with unsupervised clustering of data for the computation of the center locations. The network architecture is depicted in figure 5.1.

CLASS	%
0	94.4
1	96.6
2	97.8
3	86.6
4	94.2
5	96.4
6	94.2
7	97.5
8	98.6
9	92.1
Total	94.8

Table 2: The recognition performance of an RBF network trained with supervised clustering (LVQ) of data for the computation of the center locations. The network architecture is the same as Table.

CLASS	%
0	97.2
1	96.1
2	97.8
3	85.2
4	94.7
5	98.6
6	95.3
7	90.0
8	98.6
9	95.5
Total	95.5

Table 3: The recognition performance of an RBF network trained with gradient descent algorithm. The net architecture has 90 input units and 10 output nodes. Three experiments were considered: NET1, NET2 and NET3 corresponding to different numbers of hidden units. These numbers are respectively 16, 32 and 64.

CLASS	NET1	NET2	NET3
0	92.7	95.0	94.1
1	91.3	96.4	96.4
2	93.6	98.6	97.8
3	69.1	82.7	86.4
4	88.9	89.4	92.2
5	98.3	98.1	98.3
6	86.6	91.1	91.9
7	95.0	95.5	98.1
8	88.9	98.9	98.9
9	92.7	97.7	97.7
Total	89.7	94.3	95.1

Table 4: The recognition performance of a multilayer perceptron on the same task. The network in this case has 9 input nodes which are fully connected to the hidden nodes with tree delays (0, 1 and 2), successive windows are shifted one frame at a time. The second hidden layer is fully connected with the first hidden layer with five delays (0, 1, 2, 3 and 4).

CLASS	%
0	94.7
1	94.4
2	86.4
3	80.2
4	92.5
5	99.4
6	76.0
7	86.9
8	95.8
9	92.1
Total	80.8

Table 5: The computing resources needed by the described algorithms in trem of floatting point operations. Here N is the number of training samples, of the applied algorithms, n and o are respectively the number of hidden and output units whereas d is the dimension of each input sample, $\tilde{\gamma}$ is the neighbourhood radius. Note that for time delay networks the number of units as each layer must be multiplied by the number of delays with the upper layer.

Algorithm	FLOP
SOFM + LMS	$N(2nd + 3\tilde{\gamma}^2 d) + 4Nno$
LVQ + LMS	$N(2nd + 2d) + 4Nno$
GRBF	$N(5o + 4no + n + 5nd)$
MLP	$N(5o + 4no + n + 5nd)$

it had been demonstrated repectively that, the code vectors tend to be uniformly distributed due to the learning rules adopted [Koho 90] and also the kernel widhts learning does not improve the generalization performences of the resulting network [Wett 92].
Obviously, we notice that the performance improves as the number of hidden units increases. On the other hand, the centroid computation and LMS estimation of the hidden to output weights can lead to unacceptable computational requirements when very large networks are used. Table 1 shows the recognition performance of 256 hidden units network. The convergence and consequently the learning times strictly depend on the adopted parameters $\alpha(t)$, $\gamma(t)$, η which are for our experiments $\alpha(t) = 0.2e^{-t/2000}$ and $\gamma(t) = 5e^{-t/1000}$. The learning procedures required respectively for each iteration about $N(2nd + 3\tilde{\gamma}d$ floatting point operations for the first learning step (n the number of hidden neurons, d the input dimensions, $N$the training vectors number 6494 for our case) and $4Nno$ floating point operations for the second step (o the number of output units).

3.2 Supervised clustering of data

Using the LVQ algorithm we tried to set the center locations with supervised learning. We notified that the first hidden layer performs a raw classification of the input speech snapshots refined later at the second stage. The hidden units were uniformly divided among the ten classes. When the number of such units was not not evenly divided by 10 the remaining units were assigned

to class 'three'. Table 2 and 5 summarises respectively the recognition percentages and the floatting point operations needed for each iteration. It is important to emphasise that only 50 epochs of the algorithm were sufficient for convergence while about 4200 epochs were needed for the first experiment.

3.3 Generalized RBF

As mentionned in [Wett 92] this method can perform better than the previous ones. Therefore, three experiments were considered: NET1, NET2 and NET3 corresponding to different numbers of hidden units (16, 32 and 64). The overall performance and the complexity of the algorithm are summarised in Table 3 and 5 respectively. The overall computing time is about 10 times greater than that of layer by layer learning scheme of previous experiment.

3.4 Multilayer perceptrons

The last experiment makes use of the common Time Delay Neural Network (TDNN) [Waib 91]. The direct comparison with the previous results is not immediate. Indeed, in this case the hidden units have a completely different behaviour and therefore it is possible that the sizing adopted before does not fit for this case. For example, the input layer dimension of 90 corresponding to a segment of 200 mseconds of speech, is too wide and several trials have shown that it is difficult to reach good local minima in the error surface. Due to this reason, we implemented a TDNN network whose the hidden nodes look at a window of three consecutive frames (27 input nodes), and successive windows are shifted one frame at time; furthermore, the second hidden layer is fully connected to the first hidden layer with five delays (0, 1, 2, 3 and 4). Table 4 and 5 summarises the recognition rate obtained and the complexity of the algorithm respectively.

4 Conclusion and future works

In sight, to take advantage of the significant low training cost of RBF networks, while obtaining better recognition performance; modified forms of RBF networks have been investigated for sequence recognition task, isolated word recognition in particular. Therefore, we shown in this paper that the

models considered perform quite well, although the absolute results (about 95.5% of correct classification) do not seem impressive, due to the database limits.

Acknowledgements

The authors are indebted with ALCATEL FACE S.p.a. for the provision of the speech data. This work was supported by contratto quinquennale IIASS-CNR and MURST 40% unità INFM Università di Salerno 50% IMSP-ICTP Universit Nationale du Bénin.

References

[Beng 92] Bengio Y. (1992). Radial basis function for speech recognition. in: P. Laface and R. de Mori, eds., *Speech Recogniyion and Understanding: Recent Advances, trends and Applications* (NATO ASI Series vol. F 75).

[Bish 91] Bishop C. (1991). Improving the generalization properties of radial basis function neural networks, *Neural Computation*, **3**, pp. 579-588.

[Broo 88] Broomhead D.S. and Lowe D. (1988). Multivariate functional interpolation and adaptive networks, *Complex System*, **2**, pp. 321-355.

[Cecc 93] Ceccarelli M. and Hounsou J.T. (1993). RBF networks vs. multilayer perceptrons for sequence recognition. *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, II, pp. 651-656.

[Cecc 94] Ceccarelli M. and Hounsou J.T. (1994). A dynamic Mixture of Gaussians Neural Network for Sequence Classification. *Proceedings of ICANN-94*, M. Marinaro and P.Morasso eds., Sorrento, Italy, Springer-Verlag, pp. 475-478.

[Cecc 95] Ceccarelli M. and Hounsou J.T. (1995). Sequence recognition with radial basis function neyworks: experiments with spoken digits. *Neuro-computing* ???.

[Chou 92] Chou W.S. and Chen Y.C. (1992). A new fast algorithm for effective training of neural classifier, *Pattern Recognition* **25** no. 4, pp. 423-429.

[Cybe 89] Cybenko G. (1989). Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signal, and Systems*, 2:303-314.

[Gema 92] Geman S., Bienenstock E. and Dousart R. (1992). Neural networks and the bias/vartiance dilemma, *Neural Computation*, vol.4, no. 1, pp. 1-58.

[Hart 90] Hartman E.J., Keeler J.D. and Kowalsky J.M. (1990). Layered neural networks with gaussian hidden units as universal approximators, *Neural Computations*, **2**, pp. 210-215.

[Koho 90] Kohonen T. (1990). The self-organizing map. *IEEE Proc.*, **78**, pp. 1464-1480.

[Micc 86] Micchelli C.A. (1986). Interpolation of scattered data: distance metrices and conditionally positive definite functions, *Constr. Approximation*, **2**, pp. 11-22.

[Mood 89] Moody J. and Darken C.J. (1989). Fast learning in networks of locally tuned processing units, *Neural Computation*, **1**, pp. 281-294.

[Moor 88] Moore W.R. (1988). Conventional fault-tolerance and neural computers. NÁTO ASI Series vol.F41, *Neural Computers*, Eckmiller & v.d. Marlsburg eds. pp.29-37.

[Musa 92] Musavi M.T., Ahmed W. Chan K.H., Faris K.B. and Hummels D.M. (1992). On the training of radial basis function classifiers, *Neural Networks* **5**, pp. 595-603.

[Park 91] Park J. and Sandberg I.W. (1991). Universal approximation using radial basis function, *Neural Computation*, vol. 3, no. 2, pp. 246-257.

[Pogg 90] Poggio T. and Girosi F. (1990). Networks for approximation and learning. *Proc. IEEE*, vol. 78, no. 9, pp. 1481-1497.

[Pome 89]

[Reyn 92] Reynolds J. and Tarassenko L. (1992). Spoken letter recognition with neural networks, *Int. Neural Syst.*, vol 3, no. 3, pp. 219-235.

[Spat 80] Spath H. (1989). Cluster Analysis Algorithms for Data Reduction and Application of Objects Elis Horwood, NY.

[Spect 90] Spect D.F. (1990). Probalistic neural networks, *Neural Networks*, **3**, pp. 109-118.

[Waib 91] Waibel A., Hanazawa T., Hinton G., Shikano K. and Lang K.J. (1991). Phoneme recognition using Time-Delay Neural Networks. *IEEE Trans. ASSP*, vol. 24, no. 11, pp. 1085-1091.

[Wett 92] Wettschereck D. and Dietterich T. (1992). Improving the performance of radial basis function networks by learning center locations. *Neural Information Processing System 4.* In Moody J.E., Hanson S.J. and Lippman R.P. eds., Morgan Kauffman, pp. 1133-1140.

[Chan 91] Wu J.X. and Chan C. (1991). Recognition of phonetic labels of the timit speech corpus by means of an artificial neural network. *Pattern Recognition*, vol. 24, no. 11, pp. 1085-1091.