To appear: Pattern Analysis and Applications, special issue on Fusion of Multiple Classifiers.

A hybrid projection based and radial basis function architecture: Initial values and global optimization*

Shimon Cohen **Nathan Intrator**
School of Computer Science
Tel-Aviv University
Ramat Aviv 69978, Israel
www.math.tau.ac.il/∼nin

Revised: October, 2001

Abstract

We introduce a mechanism for constructing and training a hybrid architecture of projection based units and radial basis functions. In particular, we introduce an optimization scheme which includes several steps and assures a convergence to a useful solution. During network architecture construction and training, it is determined whether a unit should be removed or replaced. The resulting architecture has often smaller number of units compared with competing architectures. A specific overfitting resulting from shrinkage of the RBF radii is addressed by introducing a penalty on small radii.

Classification and regression results are demonstrated on various benchmark data sets and compared with several variants of RBF networks [1, 12]. A striking performance improvement is achieved on the vowel data set [8].

Keywords: Projection units, RBF Units, Hybrid Network Architecture, SMLP, Clustering, Regularization.

1 Introduction

The duality between projection-based approximation and radial kernel methods has been explored theoretically [9]. It was shown that a function can be decomposed into mutually exclusive parts: the radial part and the ridge (projection based) part. It is difficult however, to separate the radial portion of a function from its projection based portion before they are estimated. Thus, sequential methods which attempt to first find the radial part and

[*] This work was partially supported by the Israeli Ministry of Science, by the Hermann Minkowski – Minerva Center for Geometry at Tel Aviv University, and by the Israel Academy of Sciences and Humanities – Center of Excellence Program. Part of this work was done while N. I. was affiliated with the Institute for Brain and Neural Systems at Brown University and supported in part by ONR grants N00014-98-1-0663 and N00014-99-1-0009.

then proceed with a projection based approximation are likely to get stuck in sub-optimal solutions.

Earlier approaches to kernel based estimation were based on Volterra and Wiener kernels [30, 33] but they failed to produce a practical optimization algorithm that can compete with MLPs or RBFs. A relevant statistical framework is Generalized Additive Models (GAM) [17, 18]. In that framework, the hidden units (the components of the additive model) have some parametric form, usually polynomial, which is estimated from the data. While this model has nice statistical properties [31], the additional degrees of freedom, require strong regularization to avoid over-fitting. One of the more advanced RBF methods has been proposed by Orr [25]. He suggested to construct an RBF network using regression trees and presented a pruning process for model selection that is based on a Bayesian Information Criterion. Higher order networks increase the complexity of the RBF units. They include a quadratic term in addition to the linear term of the projections [21]. While they present a powerful extension of MLPs, they do so at the cost of squaring the number of input weights to the hidden nodes. Flake has suggested an architecture similar to GAM where each hidden unit has a parametric activation function which can change from a projection based to a radial function in a continuous way [12]. This architecture uses a squared activation function, thus called Squared MLP (SMLP) and only doubles the number of input weights. This architecture achieved overall best results among the RBF architectures that we have tested.

Motivated by Donoho and Johnstone result [9], this paper introduces a simple extension to both MLP and RBF networks by combining RBF and Perceptron units in the same hidden layer. Unlike the previously described methods, this does not increase the number of parameters in the model, but requires a determination of the number of RBF and Perceptron units in the network during training. The new hybrid architecture, which we call Perceptron Radial Basis Net (PRBFN), automatically finds the relevant functional parts from the data. By optimizing the units concurrently, it avoids local minima which often occur in sequential architecture optimization methods. The new architecture construction and training leads to superior results on data sets on which radial basis functions have so far produced best results, in particular, on the vowel classification data [8].

The paper extends an earlier version of hybrid architecture training [6] in a number of directions: (i) It provides a better initialization rule for choosing perceptrons. (ii) It presents an analytic computation for the initial weights in the first layer. (iii) It provides a mechanism for determining whether an RBF unit is essential. (iv) It introduces a constrained global optimization which reduces the chances of getting stuck in sub-optimal local minima solutions.

2 The hybrid RBF/FF architecture and training

For simplicity, we shall consider a single hidden layer architecture, since the extension to a multi-layer net is simple. In the hybrid architecture, some hidden units are of radial functions and the others are of projection type. All the hidden units are connected via a set of weights to the output layer which can be linear, for regression problems, or non-linear for classification problems.

There are several steps in estimating parameters to the hybrid architecture; First, the

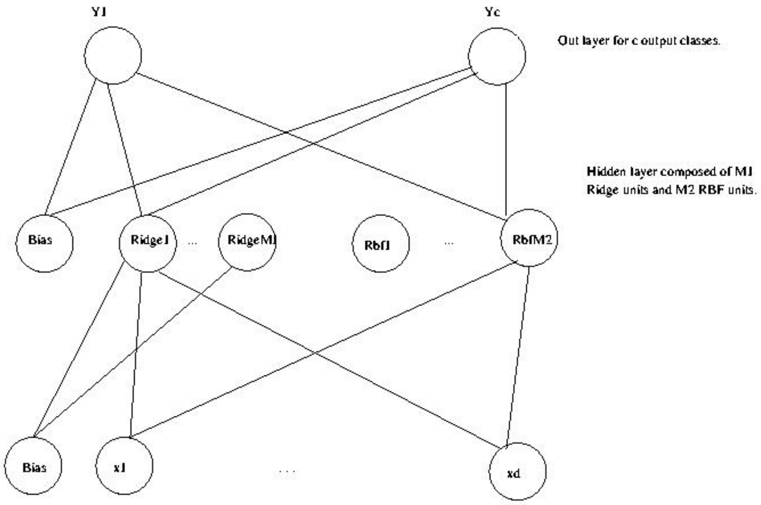
Y1
Yc
Out layer for c output classes.
Hidden layer composed of M1
Ridge units and M2 RBF units.
Bias
Ridge1
...
RidgeM1
Rbf1
...
RbfM2
Bias
x1
...
xd

Figure 1: PRBF hybrid neural network with M1 Perceptrons and M2 RBFs.

number of cluster centers is determined from the data and the number of RBF hidden units is chosen accordingly. Each RBF units is assigned to one of the cluster centers. The clustering can be done by a k-means procedure [10]. A discussion about the benefits of more recent approaches to clustering is beyond the scope of this paper. Unlike Orr [24], we assume that the clusters are symmetric, although each cluster may have a different radius. This reduces the number of free parameters. It is likely that in data-sets where Orr's method outperforms other RBF methods (e.g. on Friedman's data below), the assumption is not valid.

We are left with setting the initial weights for the projection based units. These weights are set using a linear discriminating criterion. The second layer of weights can then be found using a pseudo-inverse of the activity matrix or via a least mean square procedure. The last step in the parameter estimation is to refine the weights of the hybrid network via some form of gradient descent minimization on the full architecture.

The basic philosophy of our algorithms is as follows. We start with a large architecture that includes sufficient projection units and sufficient RBF units for the given problem. Thus we have to devise an algorithm to eliminate units which are not functional or have small contribution to the overall approximation. Projection units are reduced or tampered by the familiar weight decay constraint, using a strong weight decay imposes high penalty on weights that are not zero and practically drive un-necessary projection units to zero weights. With regards to RBF units, we need to test whether they are centered around a Gaussian bump in the data, or whether that area of the data could be better approximated by a projection unit.

Following the clustering algorithm, we describe in the next section a crude form of density estimation to test whether RBF units should be used with the cluster centers that were found. If the criteria is not met, we replace the RBF units with projection units. An example can be seen in Figure 2 where data that is composed of a sigmoidal surface and clusters is shown. If the three adjacent clusters belong to the same class than it is better to use a perceptron unit to separate them from the rest of the data. If, on the other hand, they are from different classes it would be better to use RBFs. The sigmoidal surface part of the data should always be approximated with a projection unit.

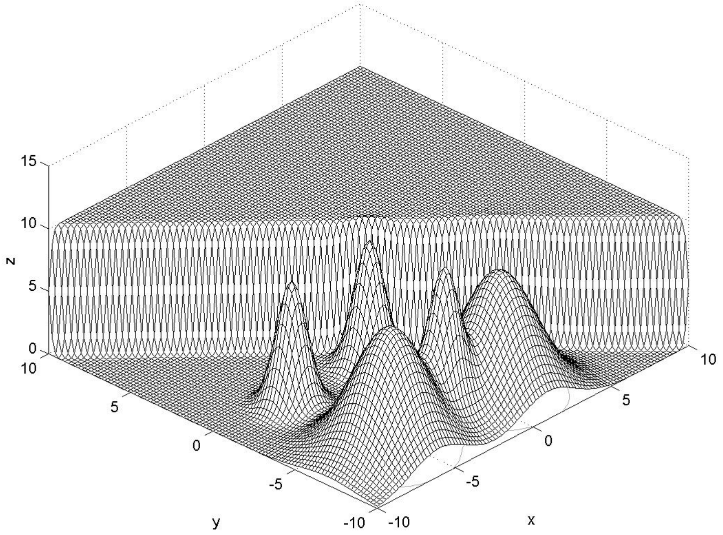

Figure 2: Data that is composed of five clusters and a sigmoidal surface.

2.1 Gaussian Density hypothesis testing

When the data dimensionality is low, the conventional χ^2 test [23] can be used to determine whether the density of the cluster is Gaussian. As the dimensionality grows, this test becomes impractical. Fukunaga has analyzed the one dimensional random variable (r.v.) ξ which measures the Mahalanobis distance [16] of the high dimensional r.v. X from the mean.

$$\xi = \frac{1}{N-1}(X-m)^T\Sigma^{-1}(X-m),$$

where m is the sample mean: $m = \frac{1}{N}\sum_{i=1}^{N} X_i$, and Σ is the sample covariance matrix: $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(X_i - m)(X_i - m)^T$. When the data is Gaussian, ξ has a Gamma distribution, This can be expanded and the sample mean and the sample covariance matrix can be used for the Mahalanobis distance:

$$p(\xi) = \frac{\Gamma(\frac{N-1}{2})}{\Gamma(\frac{n}{2})\Gamma(\frac{N-n-1}{2})}\xi^{\frac{n}{2}-1}(1-\xi)^{\frac{N-n-1}{2}-1}. \tag{1}$$

Fukunaga suggested to test Gaussianity by analyzing the variance of ξ; When the mean and variance of the distribution are not known but estimated from the data, then ξ has a

β distribution [16] with variance given by:

$$Var(\xi) = \frac{2d}{(N-1)^2} \frac{1-(d+1)/N}{1+1/N}. \tag{2}$$

Thus, we compute the sample mean m and sample covariance Σ in each cluster, then for each pattern in the cluster we compute:

$$\xi^k = \frac{1}{N-1}(X^k - m)^T \Sigma^{-1} (X^k - m). \tag{3}$$

Then we compute the variance vs of ξ^k and compute the variance vb as given by Equation 2. Using the fact ξ has a β distribution, we can decide on the confidence p value on which we want to reject the null hypothesis (that the distribution is Gaussian) and set the threshold of $\mid (vs - vb) \mid / vb$ accordingly.

The above test for Gaussianity can be replaced by other tests depending on the nature of the problem and the desired solution. Tests like Kurtosis [32] and other forms of Exploratory Projection Pursuit [15, 13, 20] are possible. We used the Gamma distribution, as it does not require any optimization, although the calculation of the covariance and its inverse are involved.

2.2 Computation of initial weights of projection units

It is important to find a useful initial weight values when it is determined that an RBF unit should be replaced by a projection unit. Otherwise, training is required in order to determine whether the unit is needed at all and the process of model selection becomes very slow. We take advantage of the prior knowledge about the rough membership of the patterns in the cluster (which we have determined to be approximated with a projection unit). We require the new projection unit to have maximal response for the patterns in its cluster which are denoted by C_j. We further require that the projection unit will have a minimal response to the rest of the data. Let N be the number of patterns in the data set. Let $N_1 = |C_j|$ be the number of patterns in the cluster, and $N_2 = N - N_1$. We wish to maximize:

$$J_1 = \frac{1}{N_1} \sum_{x^i \in C_j} w^T x^i, \tag{4}$$

and for data points that do not belong to this cluster minimize the term:

$$J_2 = \frac{1}{N_2} \sum_{x^i \notin C_j} w^T x^i. \tag{5}$$

Let y_i be 1 for $x_i \in C_j$ and -1 otherwise. The following criterion is maximized:

$$J = \frac{1}{N_1} \sum_{x^i \in C_j} w^T x^i y^i + \frac{1}{N_2} \sum_{x^i \notin C_j} w^T x^i y^i, \tag{6}$$

subject to the constraint

$$\sum_{k=1}^{d} w_k^2 = 1.$$

Using Lagrange multipliers, we arrive at the following criterion:

$$J = \frac{1}{N_1} \sum_{x^i \in C_j} w^T x^i y^i + \frac{1}{N_2} \sum_{x^i \notin C_j} w^T x^i y^i + \alpha(\sum_{k=1}^{d} w_k^2 - 1). \tag{7}$$

The partial derivative with respect to the cluster center weight vector is:

$$\frac{\partial J}{\partial w_j} = \frac{1}{N_1} \sum_{x^i \in C_j} y^i x^i + \frac{1}{N_2} \sum_{x^i \notin C_j} y^i x^i + 2\alpha w, \tag{8}$$

and the partial derivative with respect to α is

$$\frac{\partial J}{\partial \alpha} = \sum_{i=1}^{d} w_i^2 - 1. \tag{9}$$

For convenience, let

$$Z = \frac{1}{N_1} \sum_{x^i \in C_j} y^i x^i + \frac{1}{N_2} \sum_{x^i \notin C_j} y^i x^i.$$

Setting Equation 8 to zero gives:

$$Z = -2\alpha w.$$

Squaring both sides and using (9) gives:

$$\parallel Z \parallel^2 = 4\alpha^2.$$

Thus, we obtain:

$$2\alpha = \pm \parallel Z \parallel,$$

or,

$$w = \pm \frac{Z}{\parallel Z \parallel}. \tag{10}$$

The Hessian, which is derived from Equation 8, provides the correct sign of w and ensures the maximization procedure:

$$\frac{\partial^2 J}{\partial w^2} = 2\alpha I. \tag{11}$$

Thus, the Hessian is a diagonal matrix, and it is negative when α is negative, leading to setting w as follows:

$$w = \frac{Z}{\parallel Z \parallel}. \tag{12}$$

2.3 Gradient based parameter optimization

Problems with estimating sub-optimal cluster centers, can be alleviated by performing post parameter estimation of the full model after the estimation of the forward weights and the replacement of some radial basis units by projection units. An initial step in this direction is to perform a gradient descent on the cluster centers, the projection units and the forward weights. Full optimization of an RBF architecture was described in [5]. We describe here the extension of that algorithm to the hybrid architecture.

The last step in the parameter estimation is to refine the weights of the hybrid network via some form of gradient descent minimization on the full architecture. We start by deriving the gradient of the full architecture. The output of a radial basis unit is given by:

$$\phi(x, w_i) = \exp^{\frac{-(x-w_i)^2}{2r_i^2}} .$$

The output of a projection based unit is given by:

$$a_j = g(\sum_i (w_{ji} \cdot z_i)),$$

Where z is the output of the previous layer or the input to the hidden layer and w is the weight vector associated with this unit. The transfer function g is monotone and smooth such as sigmoidal. It is linear in the case of regression. The total error is given by the sum of the errors for each pattern:

$$E = \sum_{n=1}^{N} E^n. \tag{13}$$

We estimate the architecture parameters by performing a gradient descent search after the initial parameter estimation. The search should include the cluster centers, the clusters' radii and the weights of the input and output layers. This concurrent search on all the parameters is non-trivial, as it appears that the force that is driving the cluster radii to zero is stronger than the other optimization forces. Wang and Zhu [26] addressed this problem by assuming a fixed size of the radii and thus, performing the optimization on the remaining parameters, namely the cluster centers and the forward weights. We address the shrinking radius problem by applying a constrained optimization, which penalizes small radii. The optimization objective function is:

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{M} (y_k^n - t_k^n)^2 + \alpha \sum_{k=1}^{M} \frac{1}{r_k},$$

where r_k is the radius of cluster k. Note that we assume a radially symmetric cluster, this assumption can be relaxed, by performing a local Mahalanobis transformation around each cluster. α is a small regularizing parameter, which should be estimated by cross validation on the training data.

The error on K outputs for the n'th pattern is given by:

$$E^n = \frac{1}{2} \sum_{k=1}^{K} (y_k^n - t_k^n)^2, \tag{14}$$

where t_k^n and y_k^n are the target value and output value for the n'th pattern of the k'th output respectively. The partial derivatives of the error function with respect to the output weights is given by:

$$\frac{\partial E^n}{\partial w_{kj}} = g'(a_k)(y_k^n - t_k^n)z_i. \tag{15}$$

where z_i is the output of the previous layer and $g'(a_k)$ is the derivative of the transfer function at the linear value a_k.

The error term δ for the output units is given by:

$$\delta_k^n = (y_k^n - t_k^n)g'(a_k),$$

The error term for the hidden units by:

$$\delta_j^n = g'(a_j)\sum_{k=1}^{K} \delta_k^n w_{kj}. \tag{16}$$

Using this notation, the partial derivatives of the error function with respect to first layer of weights (from the patterns to Ridge units) is given by:

$$\frac{\partial E^n}{\partial w_{ji}} = \delta_j^n x_i^n. \tag{17}$$

The partial derivatives of the error function with respect to the centers of the RBFs is given by:

$$\frac{\partial E^n}{\partial m_j} = \sum_{k=1}^{K} \delta_k^n w_{kj} \frac{(x^n - m_j)}{r_j^2} \phi(x^n, w_j). \tag{18}$$

The partial derivatives of the error function with respect to the radii is given by:

$$\frac{\partial E^n}{\partial r_j} = \sum_{k=1}^{K} \delta_k^n w_{kj} \frac{\| x^n - m_j \|^2}{r_j^3} \phi(x^n, w_j). \tag{19}$$

A momentum term can be added to the gradient, however it was not found to be useful with the hybrid gradient. A Levenberg Marquardt updating rule was found to be very useful for updating the weights, the centers and the radii. It is given by

$$w_{new} = w_{old} - (Z^T Z + \lambda I)^{-1} Z^T w_{old},$$

where the matrix Z is given by:

$$(Z)_{ni} = \frac{\partial y^n}{\partial w_i}. \tag{20}$$

3 Experimental results

This section describes regression and classification results of several variants of RBF and the proposed PRBFN architecture on several data sets. Orr's RBF [11] method ($RBF-Reg-Tree$) is based on regression tree for clusterization. This methods builds a large tree and then prunes it using model selection criteria to achieve a smaller tree. Matlab's RBF package ($RBF-OLS$) implements an incremental algorithm [34], a new unit is added with a center that correspond to the pattern with the largest contribution to the current objective function. Bishop's algorithm [2] is based on the Expectation Maximization algorithm [7] for clustering ($RBF-EM$).

The results which are only given for the test data are an average over 100 runs and include the standard deviation.

3.1 Regression

We start with a comparison on four simulated regression data sets that were used by Orr to asses the performance of RBF. The results are summarized in Table 1.

The first data set [22, 25] is based on a one dimensional Hermite polynomial

$$y = (1 + (x + 2x^2))e^{-x^2}.$$

100 input values are sampled randomly between $-4 < x < 4$, and Gaussian noise of standard deviation $\sigma = .1$ was added to the output.

The second data set is a 1D sine wave [24].

$$y = \sin(12x),$$

with $x \in [0,\ 1]$. A Gaussian noise was added to the outputs with a standard deviation of $\sigma = 0.1$. 100 sets of 50 train and 50 test patterns randomly sampled from the data with the additive noise were used.

	MacKay	1D Sine	2D Sine	Friedman
RBF-Reg-Tree	0.44 ±0.14	0.44 ±0.14	0.91 ±0.19	0.12±0.03
RBF-OLS	0.69±0.41	0.57±0.27	0.74±0.4	0.2 ±0.03
RBF-EM	6.82±0.82	0.33 ±0.16	0.53 ±0.19	0.17 ±0.02
PRBFN	0.39±0.11	0.33±0.16	0.51 ±0.19	0.15±0.02

Table 1: Comparison of Mean squared error results on four data sets (see [11] for details). Results on the test set are given for several variants of RBF networks which were used also by Orr to asses RBFs. MSE Results of an average over 100 runs including standard deviation are presented.

The third data-set is a 2D sine wave,

$$y = 0.\sin(x_1/4)\sin(x_2/2),$$

with 200 training patterns sampled at random from an input range $x_1 \in [0,\ 10]$ and $x_2 \in [-5,\ 5]$. The clean data was corrupted by additive Gaussian noise with $\sigma = 0.1$. The test set contains 400 noiseless samples arranged as a 20 by 20 grid pattern, covering the same input ranges. Orr measured the error as the total squared error over the 400 samples. We follow Orr and report the total squared error on this test set.

The fourth data-set is a simulated alternating current circuit with four input dimensions (resistance R, frequency ω, inductance L and capacitance C) and one output impedance $Z = \sqrt{R^2 + (\omega L - 1/\omega C)^2}$. Each training set contained 200 points sampled at random from a certain region [14, 11, for further details]. Additive noise was added to the outputs. The experimental design is the same as the one used by Friedman in the evaluation of MARS [14]. Friedman's results include a division by the variance of the test set targets. We follow Friedman and divide the MSE by the variance of the test targets on this set. Orr's regression trees method [11] outperforms the other methods on this data set. We believe that this is due to the high inhomogeneity (nonlinearity) in the data which is better captured by the tree split of the data.

3.2 Classification

We have used several data sets to compare the classification performance of the proposed method to other RBF networks. The sonar data set attempts to distinguish between a mine and a rock. It was used by Gorman and Sejnowski [27] in their study of the classification of sonar signals using neural networks. The data has 60 continuous inputs and one binary output for the two classes. It is divided into 104 training patterns and 104 test patterns. The task is to train a network to discriminate between sonar signals that are reflected from a metal cylinder and those that are reflected from a similar shaped rock. There are no results for Bishop's algorithm as we were not able to get it to reduce the output error. Gorman and Sejnowski report on results with feed-forward architectures [29] using 12 hidden units. They achieved 90.4% correct classification on the test data with the angle dependent task. This result outperforms the results obtained by the different RBF methods, and is only surpassed by the proposed hybrid RBF/FF network.

The Deterding vowel recognition data [8, 12] is a widely studied benchmark. This problem may be more indicative of the type of problems that a real neural network could be faced with. The data consists of auditory features of steady state vowels spoken by British English speakers. There are 528 training patterns and 462 test patterns. Each pattern consists of 10 features and it belongs to one of 11 classes that correspond to the spoken vowel. The speakers are of both genders. The best score so far was reported by Flake using his SMLP units. His average best score was 60.6% [12] and was achieved with 44 hidden units. Our algorithm achieved 67% correct classification with only 27 hidden units. As far as we know, it is the best result that was achieved on this data set.

The seismic1 and seismic2 data sets are two different representations of seismic data. The data sets include waveforms from two types of explosions and the task is to distinguish between the two types. This data was used in a "Learning" course in the last two years for performance evaluation of many different classifiers[1]. The dimensionality of seismic1 is 352 representing 32 time frames of 11 frequency bands, and the dimensionality of seismic2

[1] For details see `http://www.math.tau.ac.il∼nin/learn98,9`

Algorithm	Sonar	Vowel	Seismic1	Seismic2	waveform
RBF-Reg-Tree	71.7±0.5	–	63±0	79±0	–
RBF-OLS	82.3±2.4	51.6±2.9	73±4	81±3	83.8±0.2
RBF-EM	–	48.4±2.4	60±4	77±5	83.5±0.2
PRBFN	91±2	67±2	89±0	85±3	85.8±0.3

Table 2: Percent classification results of different classifiers variants on four data sets.

patterns is 242 representing 22 time frames of 11 frequency bands. Principal Component Analysis (PCA) was used to reduce the data representation into 12 dimensions. Both datasets have 65 training patterns and 19 test patterns which were chosen to be the most difficult for the desired discrimination.

The waveform data set is a three class problem which was constructed by Brieman to demonstrate the performance of the Classification and Regression Trees method [3]. Each class consists of a random convex combination of two out of three waveforms sampled discretely with added Gaussian noise. The data set contains 5000 instance, and 300 are used for training. Recent reports on this data-set can be found in [19, 4]. Each used a different size training set. We used the smaller training set size as in [19] who report best result of 19.1% error. The Optimal Bayes classification rate is 86% accuracy, the CART decision tree algorithm achieved 72% accuracy, and Nearest Neighbor Algorithm achieved 38% accuracy. PRBFN has achieved 85.8% accuracy on this data set. There is not much room for improvement over the PRBFN classifier, in this example.

Table 2 summarizes the percent correct classification results on the different data sets for the different RBF classifiers and the proposed hybrid architecture. As in the regression case, the STD is also given however, on the seismic data, due to the use of a single test set (as we wanted to see the performance on this particular data set only) the STD is often zero as only a single classification of the data was obtained in all 100 runs.

4 Discussion

The general idea of our hybrid architecture falls under the theory of generalized additive models [18]. The theory does not address however, what type of architectures to combine and how to train them so that each part of the function approximation is used to fit the most appropriate portion of the data. Motivated by the theoretical work of Donoho and Johnstone [9], we have chosen to combine perceptron units with RBF units and not use more complex architectures which can have a larger variety of transfer functions. The issue of appropriate exploitation of each function approximation family was addressed during the construction of the architecture, by the various regularization constraints and by the global parameter optimization that is used at the end of architecture selection. This is made practical by the simplicity of training, and the tight control over the number of model parameters via regularization. A particular type of overfitting due to shrinkage of the RBF radii was avoided by additional penalty on small radii. This can be seen as another smoothness criterion.

In the extensively studied vowel data set, the proposed hybrid architecture achieved average results which are superior to the best known results [28] while using a smaller number of hidden units. On the waveform classification problem [3], our results are close to the Bayes limit and are better than the current known results. The hybrid network also outperformed feed-forward network results and RBF results on the sonar data [27].

The proposed hybrid network is thus, a viable alternative to either projection based or radial basis functions. It shares the good convergence properties of both, and with a careful parameter estimation procedure, it is expected to outperform either on difficult tasks.

5 Acknowledgement

We wish to thank the anonymous referees for their helpful comments.

References

[1] C. M. Bishop. Improving the generalization properties of radial basis function neural networks. *Neural Computation*, 3(4):579–588, 1991.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* The Wadsworth Statistics/Probability Series, Belmont, CA, 1984.

[4] J. Buckheit and D. L. Donoho. Improved linear discrimination using time-frequency dictionaries. Technical Report, Stanford University, 1995.

[5] S. Cohen and N. Intrator. Global optimization of RBF networks, 2000. Submitted to IEEE TNN.

[6] S. Cohen and N. Intrator. A hybrid projection based and radial basis function architecture. In J. Kittler and F. Roli, editors, *Proc. Int. Workshop on Multiple Classifier Systems (LNCS1857)*, pages 147–156, Sardingia, June 2000. Springer.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society*, B-39:1–38, 1977.

[8] D.H. Deterding. *Speaker Normalisation for Automatic Speech Recognition.* PhD thesis, University of Cambridge, 1989.

[9] D. L. Donoho and I. M. Johnstone. Projection-based approximation and a duality with kernel methods. *Annals of Statistics*, 17:58–106, 1989.

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, New York, 1973.

[11] M.J.L Orr et al. Combining regression trees and radial basis functions. *Int. J. of Neural Systems*, 10(6):453–466, 2000.

[12] G.W. Flake. Square unit augmented, radially extended, multilayer percpetrons. In G. B. Orr and K. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 145–163. Springer, 1998.

[13] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266, 1987.

[14] J. H. Friedman. Mutltivariate adaptive regression splines. *The Annals of Statistics*, 19:1–141, 1991.

[15] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C(23):881–889, 1974.

[16] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, London, 2^{nd} edition, 1990.

[17] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–318, 1986.

[18] T. Hastie and R. Tibshirani. *Generalized Additive Models.* Chapman and Hall, London, 1990.

[19] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89:1255–1270, 1994.

[20] N. Intrator and L. N Cooper. Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, 5:3–17, 1992.

[21] Y.C. Lee, G. Doolen, H.H. Chen, G.Z.Sun, T. Maxwell, H.Y. Lee, and C.L. Giles. Machine learning using higher order correlation networks. *Physica D*, pages 22–D:276–306, 1986.

[22] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[23] G. E. Noether. *Elements of Non-Paramteric Statistics.* Wiley, New York, 1967.

[24] M. J. Orr. Introduction to Radial Basis Function networks. Technical report, Institute for Adaptive and Neural Computation, Edinburgh University, 1996. http://www.anc.ed.ac.uk/∼mjo/rbf.html.

[25] M. J. Orr, J. Hallam, A. Murray, and T. Leonard. Assessing RBF networks using DELVE. *IJNS*, 2000.

[26] Zheng ou Wang and Tao Zhu. An efficient learning algorithm for improving generalization performance of radial basis function neural networks. *Neural Neworks*, 13(4,5), 2000.

[27] Gorman R. P. and Sejnowski T. J. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Network*, 1:75–89, 1988.

[28] A.J. Robinson. *Dynamic Error Propogation Networks.* PhD thesis, University of Cambridge, 1989.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986.

[30] M. Schetzen. *The Volterra and Wiener Theories Of Nonlinear Systems.* John Wiley and Sons, New York, 1980.

[31] C. J. Stone. The dimensionality reduction principle for generalized additive models. *The Annals of Statistics*, 14:590–606, 1986.

[32] A. Stuart, M. Kendall, and J. K. Ord. *Advanced Theory of Statistics 1: Distribution Theory.* Oxford University Press, 1987.

[33] V. Volterra. *Theory of Functional and of Integro-differential Equations.* Dover, 1959.

[34] P.D. Wasserman. *Advanced Methods in Neural Computing.* Van Nostrand Reinhold, New York, 1993.

Biographies

Shimon Cohen

Shimon Cohen received a B.Sc. degree in Mathematics and Computer Science from Tel Aviv University and a M.Sc. degree from the Weizmann Institute of Science. Currently he is with the research and development group of Orbotech Israel working on pattern recognition and working on his Ph.D in Computer Science at Tel Aviv University.

Nathan Intrator

Nathan Intrator received his M.Sc. and Ph.D. in Applied Mathematics from Brown University. Prof. Intrator's research focuses on statistical methods and hybrid neural networks in pattern recognition and model evaluation. His earlier work included object representation and recognition and dimensionality reduction via exploratory projection pursuit methods. He developed various methods for estimating predictor confidence, which can be easily used for sensor fusion as well as classification and detection based on multi-scale representations. Recently, Prof. Intrator has contributed to methods that search for best basis and optimal mother wavelet (in a wavelet packet representation and in analytic wavelet transform). This was used for detection of mine-like object in sonar images as well as discrimination between malignant and benign tumors from mammogram images. He has lately been working on the analysis of acoustic signals for purpose of echolocation, in bats, dolphins and various sonars.