REALIZATION OF GENERALIZED RBF NETWORK

Nung Kion Lee[a] and Dianhui Wang[b]

[a]Faculty of Cognitive Science and Human Development, University Malaysia Sarawak, Malaysia,
email: nklee@fcs.unimas.my
[b]Department of Computer Science and Computer Engineering,
La Trobe University, Melbourne, Australia
email: csdhwang@ieee.org

Abstract - **This paper aims at developing techniques for design and implementation of neural classifiers. Based on our previous study on generalized RBF neural network architecture and learning criterion function for parameter optimization, this work addresses two realization issues, i.e., supervised input features selection and genetic computation techniques for tuning classifiers. A comparative study on classification performance is carried on by a set of protein sequence data.**

Keywords: **Radial Basis Function Network, Neural Classifiers, Genetic Algorithm, Proteins Classification**

1. INTRODUCTION

Classification involves the assignment of an unlabelled pattern to a known class or group in a particular problem domain. Type of classifiers includes decision tree classifiers (Quilan, 1994), fuzzy classifiers (Nauck & Kruse, 1995), k-Nearest Neighborhood (Fukunaga, 1990), and neural classifiers (Rumelhart *et al.*, 1986). Each classifier has its own strengths and weaknesses and its performance is problem dependent. For example, the decision tree is not suitable for noisy data whereas neural networks are more tolerant to noise. There are a number of criteria for evaluating a classifier's performance, such as the overall coverage rate and the overall misclassification rate.

Neural networks have demonstrated great potentials in resolving complex engineering problems including intelligent control and estimation, signal separation and filtering, time-series prediction, pattern recognition and classification. Multiplayer perceptron (MLP) or BP neural net (Rumelhart *et al.*,

1986) is the most popular neural network architecture due to its universal approximation power and well developed error back propagation learning algorithm (Hornik *et al.*, 1989). Despite being widely used, the MLP network has its own limitations. The number of hidden units and the number of layers for MLP network are determined in a subjective manner which requires a great number of tests. It is also considerably difficult to include prior knowledge into the network architecture even though attempts have been explored. The standard MLP network has numerous parameters (i.e. connection weights) to be updated during training, which results in a large training time and also leads to a local minima.

An alternative to MLP network is the Radial Basis Function (RBF) network. The RBF network is becoming more popular in recent years. The increasing popularity of this network architecture is mainly due to its explanatory architecture, universal approximation property, hybrid learning strategy and functional equivalence to a fuzzy inference system (Jang *et al.*, 1993). The main concern in utilizing the RBF networks is to determine the right number of hidden units and their associated parameters. Methods proposed for tackling this issue, includes using a subset of training data (Kubat & Cooperson, 1999), clustering (Jain *et al.*, 1999), mixture models and orthogonal least squares. MLP and RBF neural networks can achieve comparable results although RBF networks have better generalization capability in its general sense.

Training neural classifiers requires a criterion function or objective function to update the connection weights. The well-known Mean Squared Error (MSE) criterion function is the most commonly used. This criterion function calculates the errors between the desired outputs and actual model outputs and uses them for updating the network weights. Although a probabilistic interpretation exists for this learning criterion, it cannot ensure consistencies between learning criterion and classification criterion. Therefore, the MSE is more suitable for modeling tasks rather than classification problems. This is because neural classification system requires the desired outputs to be represented artificially, e.g. by using the 1-of-p coding. There is also no mathematical evidence to prove that this coding is optimal.

The motivation of the Generalized RBF (GRBF) neural network architecture is to improve the interpretability of the architecture and to overcome some of the limitations of the standard RBF network. This paper extends the earlier works in (Wang *et al.*, 2002(a); Wang & Dillon, 2001) with some improvements and modifications. A modified algorithm on feature subset selection is used to reduce the number of input dimensions in the hidden units. The previously proposed learning criterion for training the GRBF network is fixed, and a genetic algorithm is applied to implement the optimization of the new objective function. The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 describes the GRBF model, Section 4 gives the GRBF objective function, Section 5 details the implementation issues, Section 6 reports our experimental results and the last section concludes this paper.

2. RELATED WORK

Data driven initialization is commonly used to determine the number of hidden units, its cluster center and shapes of RBF network. The covariance matrix of a cluster determines the cluster shape (could be a sphere, or ellipsoid[+]) and orientation. Clusters centers could be determined by using clustering techniques such as k-means (Jain *et al.*, 1999), fuzzy-clustering (Gustafson & Kessel, 1979), expectation-maximization (Mak & Kung, 2000), or agglomerate (Aggarwal *et.al*, 1999). Thus, the work reported in (Hwang & Bang, 1997; Gustafson & Kessel, 1979; Mak & Kung, 2000) is directly applicable to our work. These papers propose a generalized fuzzy classifier initialized by the data. Similar to Looney (Looney, 2002), each output node is connected only to hidden units resulting from the data in the same class. Unlike (Looney, 2002), our work uses the RBF network instead of the Probabilistic neural network.

The EM clustering techniques (Jain *et al.*, 1999; Mak & Kung, 2000) produce a Gaussian mixture model that can be used to initialize the GRBF network. A supervised version of the EM clustering algorithm can be used on the training data from the same class. In this way, each class will be partitioned into particular subclasses with each subclass responsible for generating the part of the examples in the same class (Duda & Hart, 1973). The works reported in (Setnes & Roubus, 2000) using genetic algorithm is related to our work in the ways the GRBF network parameters are encoded.

3. GRBF DESIGN

The GRBF architecture and learning strategy was initially proposed in (Wang *et al.*, 2002(a)), where the fuzzy plus operator was employed as an activation function in the output layer and various inputs is adopted for each hidden units. The purpose of this was to reduce the number of connection weights for enhancing the model's generalization capability. For completeness, we first present GRBF model based on our previous works in the following subsection 3.1 and 3.2. In subsection 3.3, we present a modified feature selection technique, from an implementation viewpoint.

[+] Other cluster shapes are not considered in this paper.

3.1 GRBF Model

Let p be a set of the target class. Let each class i in p be associated with a set of cluster C_{ij} ($d \times 1$) in d-dimensional space, where C_{ij} is the j-th cluster of class i. This set of clusters can be obtained through the supervised clustering techniques. Let M be $r \times r$ dimensional matrix, M^t be the transpose of the matrix M, and M^{-1} be the inverse of the matrix M. The *Mahalanobis* distance D between an input vector x and the cluster centre C is defined by,

$$D(x,C) = (x-C)\Sigma^{-1}(x-C)' \qquad (1)$$

where Σ is the positive definitive covariance matrix with diagonal elements. This matrix defines the data points with cluster shapes information. Let $\tilde{C}_{ij}$ be the sub-space of C_{ij} subjected to constraint *dim* $\tilde{C}_{ij} \geq 2$, where *dim*(v) represents the dimension of the vector v. The confidence factor (CF) of an input pattern x to a cluster C_{ij} measures the probability that this input lies within this cluster, that is, the conditional probability of $p(C_{ij}|x)$. The CF can be measured in terms of the distance between $\tilde{x}$ and the cluster centre of $\tilde{C}_{ij}$. The CF value φ_{ij} is defined by,

$$\varphi_{ij} = \exp\left(-D(\tilde{x}, \tilde{C}_{ij})\right) \qquad (2)$$

Equation 2 defines a kernel activation function of the hidden units in the GRBF network. This function has the property $f(x) \to 0$ as $|x| \to +\infty$. The covariance matrix only retains the elements corresponding to the feature selected in a cluster. Because the existence of disjunctive clusters, it is necessary to define an aggregation operator that aggregates the memberships of a data point to a class. Let $CI_i(\tilde{x}_n)$ be the confidence indicator for the input pattern x with respect to the i-th class. The aggregation operator is defined by the fuzzy algebraic sum (Wang et.al, 2002(a)), that is,

$$CI_i(\tilde{x}) = a \oplus b = a + b - a \cdot b \qquad (3)$$

where a and b are the confidence factors. The plus operator $\oplus$ achieves a partial membership or overlapping concept by favoring a stronger membership to a class. For a set of confidence factors, the confidence indicator can be expressed as in (Wang *et al.*, 2002(a)),

$$CI_i(\tilde{x}) = \sum_{k=1}^{N_i} (-1)^{k+1} \sum_{j_1 < \cdots j_k} \varphi_{ij_1} \cdots \varphi_{ij_k} \qquad (4)$$

where N_i is the number of clusters in class i. Equation (4) can be calculated by the recursive scheme proposed in (Wedding & Cios, 1998).

3.2 Architecture

Figure 1 depicts the architecture described by the model in the previous section. The "S" symbols at the output nodes represents the operator operation. Unlike the standard RBF network, the input layer and the hidden layer are not fully connected. The hidden units from class *i* are only connected to the output unit in the same class with a connection weight 1 (i.e. $w_{ij} = 1$). The connection weight values may take values in the range (0, 1), however, unit weights or equal weights will simplify the neural classifier design. The hidden unit activations represent the confidence factor. The activations belonging to the same class are applied with the operator operation at the output nodes to find the confidence indicator value.

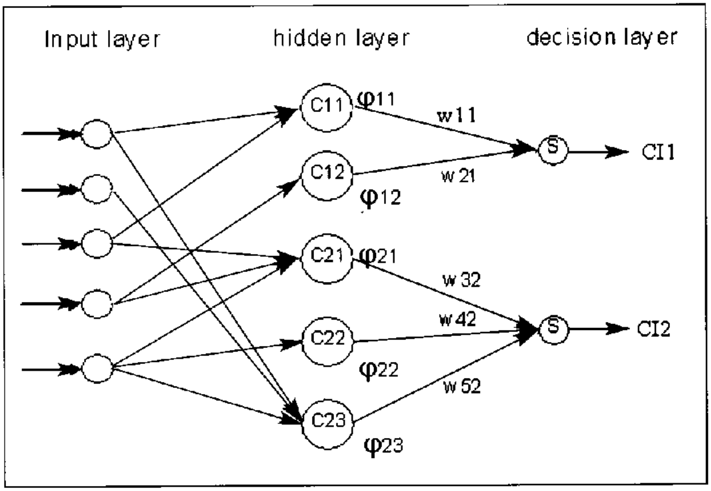

Figure 1: GRBF neural network architecture

To realize the GRBF network architecture, the number of hidden units is first determined. This is followed by reducing the number of connections between the input and the hidden layers. The Expectation Maximization (EM) clustering algorithm (Jain *et al.*, 1999) is used to establish the number of hidden units for each class. EM clustering algorithm estimates a class probability density by using a mixture of components, each with a covariance matrix that contains the cluster shape information (LeCun *et al.*, 1995). Each class can be modeled with one mixture model separately (i.e. supervised) (Mak & Kung, 2000; Duda & Hart, 1973). Reducing the full covariance obtained from the EM algorithm to the diagonal one in Equation 1 will greatly simplify the computational process, but this may result in the loss of cluster information.

3.3 Feature Subset Selection

After setting up the right number of hidden units, the number of features in each hidden unit is reduced using a features subset selection algorithm. Feature selection method picks a subset of features that are strongly relevant to the target concept. The aim of feature selection is to choose a subset of features for improving the generalization capability and decreasing the computational complexity *without significantly* reducing prediction accuracy of the classifier. Because the hidden units for each class were determined separately, their boundaries may overlap. Feature selection reduces these overlaps to maximize the membership value for each data point to its class.

The objective of subspace feature selection is to select the feature subset in a cluster whereby the data points assigned are more similar to each other, in order to minimize the overlapping with other clusters. The minimization must be viewed from the global perspective. The PROCLUS (**Pro**jected **Clus**tering) (Aggarwal *et al.*, 1999) subspace feature selection approach was adopted with modification. The modified algorithm is called FSUBS (i.e. Feature **Sub**set Selection). In general, the FSUBS algorithm constructs a partition of the points into clusters with the data points within each cluster close to one another. The FSUBS algorithm is now described in Figure 2:

```
Procedure FSUBS(k, $C_i$, $D_i$)
{k is total number of clusters}
{$C_i$: cluster centers for $1 \leq i \leq k$}
{$D_i$: set of features for $C_i$}
{$L_i$: set of points assigned to cluster i}
          CurrentBestObjective = -∞
          do
                    BestObjective = CurrentBestObjective
                    $L_i$=AssignPoints(k, $C_i$, $D_i$, T)  {T is the training data}
                    CurrentBestObjective = FindDimensions($C_i$,$D_i$,$L_i$,k,T)
          While (CurrentBestObjective ≤ BestObjective)
          save $D_i$ for $1 \leq i \leq k$
end
Procedure FindDimensions($C_i$, $D_i$, $L_i$,k,T)
{ {$S_i$ is the total number of features for cluster $C_i$}
  {$X_{ij}$ is the average distance from the points in $L_i$ to the cluster center $C_i$
along feature j}
   Objectivevalue = EvaluateObjective($C_i$, $D_i$, T)
   for each cluster $C_i$ do
            $Y_i = \frac{\sum_{j=1}^{S_i} X_{ij}}{S_i}$
```

```
        $\gamma_i = \sqrt{\dfrac{\sum_{j=1}^{S_i}(X_{ij} - Y_i)^2}{S_i - 1}}$
        for each feature j do $Z_{i,j} = \left(X_{ij} - Y_i\right)/\gamma_i$
        Sort $Z_{i,j}$ in ascending order
        Remove feature $Z'_{i,s_t}$ from $D_i$
        ReducedObjective = **EvaluateObjective**(C_i,D_i,T)
        if (ReducedObjective > Objectivevalue)
           Restore feature $Z'_{i,s_t}$ into D_i {*don't remove this attribute*}
      else
           Remove feature $Z'_{i,s_t}$ permanently
           Objectivevalue = ReducedObjective    {*best objective so far*}
      end if
   end for
   return Objectivevalue and (D_1...D_k)
}

Procedure **AssignPoints**(k, C_i, D_i, T)
{  for each $x \in$ T
      for each C_i
         d_i= calculate distance($x$, C_i, D_i)
      end of
      assign $x$ to C_j, where j = $\min\{d_i\}$ for all i
   end for
}
Procedure **EvaluateObjective**(C_i, D_i,T)
{
      *correctclassify* = 0
      for each $x \in$ T
         calculate confidence indicator CI_j for 1≤j≤p
         c= $\arg\max_j\{CI_j\}$
         if c = class($x$) then
```

Figure 2: FSUBS algorithm

The *FSUBS* main module in Figure 2 iteratively improves the performance of the whole training data set and stops when the performance decreases. The main loop in the *FSUBS* module calls the *FindDimensions* module to reduce the clusters' features. Before the *FindDimensions* module is called, the *AssignPoints* module assigns each training data to the nearest cluster.

The *FindDimensions* module finds a feature in each hidden units or cluster that is furthest from its center. This feature is removed and the new performance is indicated by using the *EvaluateObjective* module. This indicator is compared with the best performance so far to derive the best performance measure. The performance function (i.e. EvaluateObjective) determines the classification rate (CR) of the training data. The CR of the training data is calculated after each removal of a feature from a cluster. If the training data is too large, a subset of the data is used for this purpose. Using the CR as the objective function is suitable because this criterion is global.

The FSUBS algorithm differs from the PROCLUS feature selection in two aspects. In PROCLUS, the evaluation on the objective function is performed after feature subsets are selected for all the clusters. In FSUBS however, the evaluation is done immediately after a feature is removed from a cluster. The sole aim of finding subspace features using PROCLUS is to increase the similarity distance between points assigned to a cluster. Besides achieving this FSUBS further reduces the overlap between clusters from different classes.

4. MODEL OPTIMIZATIONS

The initial model is not optimal even after the feature subset selection has been performed. The GRBF model constructed so far is based on the data belonging to a single class, but the GRBF classifier needs to be optimized to handle data from all classes. Further optimization is necessary to reduce overlap between clusters and to optimize its boundaries. Due to the aggregation operator applied at the output nodes, the CI value at one output node is increased when all the hidden nodes connected to it have a high activation level. Thus, the optimization process is responsible to increase the activation level of the hidden units when the examples from the same class are presented.

The objectives of a classifier are to minimize misclassification (MR) and increase classification rate (CR). The CR and MR are defined (Wang *et al.*, 2002(a)) as follows:

$$CI^{+}(x) = \sum_{l=1}^{p} CI_{l}(x) \qquad (5)$$

$$CR_{p}(x) = \frac{CI_{p}(x)}{CI^{+}(x)}; x \in c_{p} \qquad (6)$$

$$MR_{p}(x) = 1 - CR_{p}(x); x \in c_{p} \qquad (7)$$

The overall objective for class p is expressed by,

$$Objective_p = \sum_{x \in c_p} \left[MR_p(x) + \frac{1}{CR_p(x)} \right] \qquad (8)$$

Equation 6 and 7 can be weakly interpreted as model predictions in terms of posterior probabilities. The normalized CR_P in Equation 6 ensures the MR and the CR values in the cost function are summed to 1. Equation 7 on the other hand, corresponds to the "other class" in a two-class classification problem. The overall results are to maximize the probability of examples assigned to the correct class (i.e. Equation 6) and minimize the probability of it being assigned to the wrong class (i.e. Equation 7). Equation 8 is a weaker notion if compared to the Bayesian is posterior probabilities as no formal proof is defined. However, the outputs of the GRBF network are equivalent to the neuro-fuzzy classifiers, which are widely used to predict the extent of membership of a test data to a class (Wang *et al.*, 2002(a); Wang *et al.*, 2002(b)). We propose to indirectly express the generalization power of the GRBF classifier using the trace of the covariance matrix, that is,

$$GP_p = \sum_{l=1}^{n_p} Trace(\Sigma_{pl}) \qquad (9)$$

where n_P is the number of clusters for class p and n is the total number of training data. The overall objective function with equal weights for optimizing the GRBF classifier is given by

$$Objective_p = \sum_{x \in c_p} \left[MR_p(x) + \frac{1}{\left(CR_p(x) + GP_p\right)} \right]; x \in c_p \qquad (10)$$

$$Objective = \sum_{l=1}^{p} Objective_l \qquad (11)$$

To make the objective function a stronger indicator, the misclassification rate for training dataset MR^{tra} is embedded into the objective function. This term is defined by,

$$MR^{tra}(x) = \frac{1}{n} \sum_{k=1}^{n} c_k(x) \neq \hat{c}_k(x) \qquad (12)$$

where $\hat{C}$ is the predicted class. The modified objective function is defined as follows:

$$Objective = \left[\sum_{j=1}^{p} Objective_j \right] \times MR^{tra} \qquad (13)$$

The optimization minimizes the objective function in Equation 13 accordingly.

The generic algorithm (GA) is a powerful computationed technique that emulates biological evolutionary theories to solve derivative-free optimization problems. A GA comprises of a set of individual elements (the population) and a set of biologically inspired operators defined over the population itself. According to the evolution theories, as stronger elements are evolved to the next generation weaker ones will disappear. The genomes encode possible solutions of the problem to be optimized. These genomes can be real coded or bit-string coded. The former is more efficient and computationally less expensive because transformation from a real number to a bit string and vice versa is a rather expensive operation. Furthermore, a large memory space is required for the bit string genome. A GA computation system for dealing with a specific task needs to contain the following five components (Michalewicz, 1996):

a) a genetic representation for the potential solutions to the problem;

b) a way to create an initial population of the potential solutions;

c) an evaluation function that plays the role of the environment, rating solutions in terms of their fitness;

d) genetic operators that alter the composition of children, and

e) values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators etc.).

This paper adopted the GA representation techniques used in (Setnes & Roubus, 2000). There are only two terms in the GRBF neural classifier adjusted in order to improve the classification performance. These two terms are the cluster centres Ci and the clusters shape information Âi. These terms are encoded into the genomes and represented as follows:
g_i=[R_1, R_2, …, R_r]
where R_i is the i-th hidden unit parameters and r is the total number of hidden units from all classes. The total length of a genome is $2\sum_{i=1}^{r}\dim(C_i)$. In real coded genomes, each single gene is stored as the real number data type in a programming language. The initial solution after the feature subset selection is encoded into a single genome. The initial population is established with a random

variation from this initial genome. Each Ri contains the following information,

$$R_i = [C_i^1, C_i^2, \ldots, C_i^{Ni}, (\sigma_i^1)^2, (\sigma_i^2)^2, \ldots, (\sigma_i^{Ni})^2]$$

where C_i^j is the j-th element of C_i, and σ^2 is the variance.

5. PERFORMANCE EVALUATION

In this section, we evaluate the GRBF network classifier's performance using a set of protein sequence data. A protein sequence will be identified to a predefined superfamily (i.e. class) by using the neural classifier. The superfamilies are usually formed by sequence alignment algorithm such as BLAST or FASTA (Mount, 2001). In this study, ten protein superfamilies were obtained from the PIR (Wu *et al.*, 2002) protein databases. These superfamilies are Cytochrome c, Cytochrome c6, Cytochrome b, Cytochrome b5, Triose-phosphate isomerase, Plastocyanin, Photosystem II D2 protein, Ferredoxin, Globin, and Cytochrome b6-f complex 4.2K. A total of 949 protein sequences data were used for training/validation and 533 for testing. The global features (n-grams) (Wang *et al.*, 2002(b))of the protein sequences were extracted. They are a total of 56 continuous features for each protein sequence comprising of e2 and a1. A comparative study is carried out and promising results were obtained.

After extracting the protein features, the number of hidden units for each class is determined using the EM clustering algorithm. Figure 3 shows the number of clusters for each class. The number of features used in each hidden unit is reduced by the FSUBS algorithm described in subsection 3.3.

Class	**Number of clusters**
1	7
2	2
3	6
4	4
5	4
6	2
7	2
8	4
9	16
10	1
Total	48

Figure 3: Number of clusters for each class

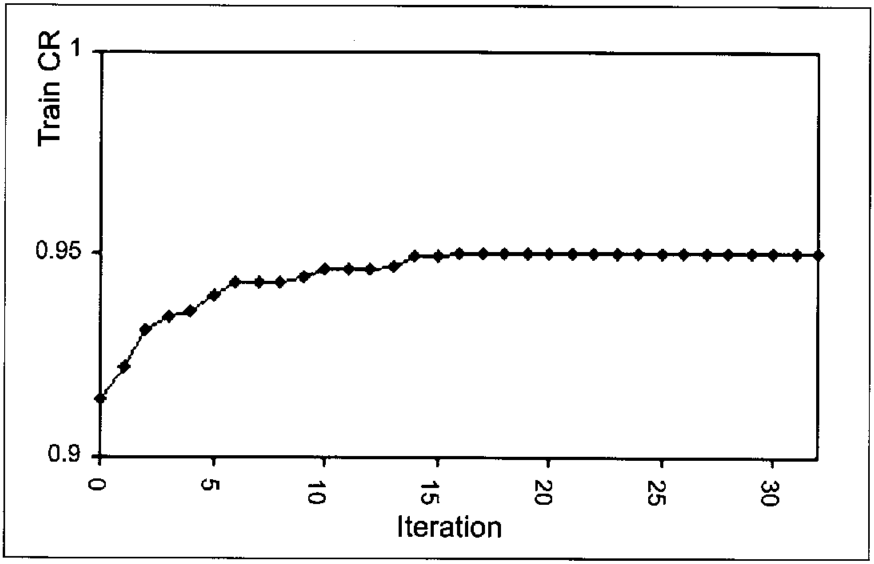

Figure 4: FSUBS training progress

Figure 4 shows the FSUBS algorithm feature reduction on the classifier performance in terms of the classification rate. After applying the feature subset selection, an average of 43 features were selected. This is a reduction of about 23.21% on the number of features in each hidden units. We observed that there are less features selected for classes with less training data.

The genetic algorithm is then applied to further optimize the weights of the GRBF network. In this experiment, a population size of 30 is setup with maximum generation of 1000. The training parameters are crossover probability 0.90, mutation probability 0.01, and replacement percentages 0.85. The steady state (Wall, 1996) GA algorithm was adopted for the training where a new generation will replace the old generation population set by the replacement percentages. The selection for replacement is based on the fitness measure of the individuals in the population. The GALib libraries (Wall, 1996) were used for the GA implementation. Figure 5 illustrates the optimization program using the GA algorithm, where the graph on the top denotes the test data and the other one is of the training data. It can be seen that the objective function optimized using the GA algorithm decreased sharply during the first 30 generation and became nearly stable for the remaining generation. This indicates that the FSUBS feature selection algorithm produces a sub-optimal neural architecture. Further optimization using the GA algorithm can further searched for the best parameters of the hidden units.

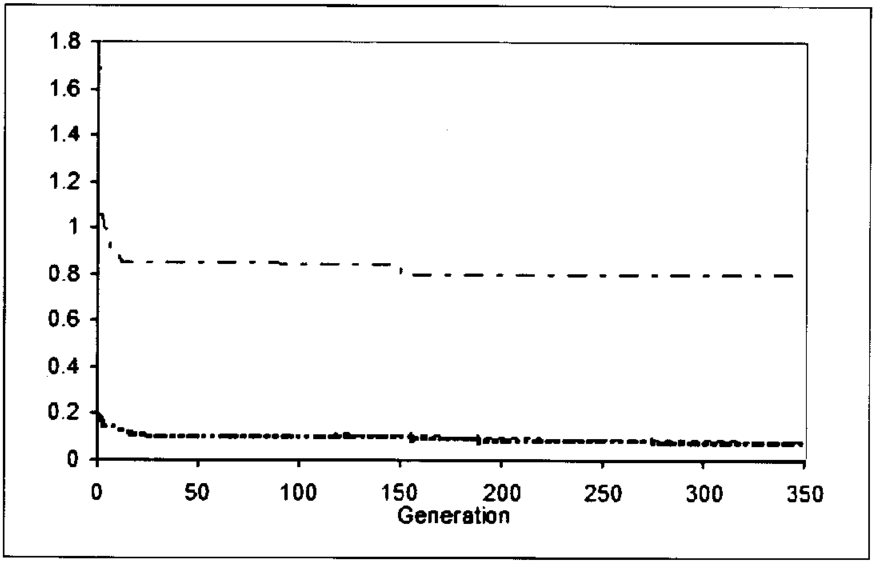

Figure 5: GA optimization

Table 1. Classifiers performance on protein sequences data

Classifier	**Training data**		**Testing data**	
	CR	MR	CR	MR
MLP-MSE	99.37%	0.63%	91.37%	8.63 %
MLP-CE	99.16%	0.84%	90.81%	9.19%
RBF-MSE	99.26%	0.74%	91.56%	8.44%
RBF-CE	97.15%	28.45%	90.99%	9.01%
C4.5	98.40%	1.60%	79.74%	20.26%
GRBF	97.37%	2.63%	92.68%	7.32%

Table 1 gives the comparison results between different classifiers. Both MSE and the Cross Entropy (CE) cost functions were used for the MLP and the standard RBF networks. The MLP network with the CE cost function obtained the highest classification rate of 99.37% on the training data. This is followed by RBF-MSE 99.26%, MLP-CE 99.16%, C4.5 98.40%, GRBF 97.37%, and RBF-

CE 97.15%. These results shows that the neural classifier with MSE cost function could classify the training data very well. For the test data, on the other hand, the GRBF neural classifier achieves the best result of 92.68%. This clearly shows that the GRBF can even generalize more on the test data as compared to the MLP and RBF networks with the MSE learning criterion function on the training data. The decision tree classifier C4.5 is unable to achieve a better performance compared to the neural classifiers. The main shortcoming of the GRBF neural classifier is that it requires a longer training time due to the optimization process using the GA algorithm.

6. CONCLUSION

This paper discusses implementation issues of a generalized RBF network classifier. A modified feature subset selection algorithm is proposed. A standard GA algorithm is applied to optimize the GRBF network. The GRBF network architecture has good interpretability property, and it has demonstrated a better generalization capability to the protein sequence data set. The comparison results showed that the performance of the GRBF network classifier outperformed other classifiers in terms of generalization capability. There are however several limitations with the GRBF network classifier. E.g. the ordering problem of the FSUB algorithm whereby the feature subset selected for each hidden unit depends on the order of the data presented.

7. REFERENCES

Aggarwal, C., Procopiuc, C., C., Wolf, J. L., Yu, P. S., and Park, J. S. (1999), Fast algorithms for projected clustering, *SIGMOD'99*, Philadephia, PA., June, pp. 61-82.

Blumenstein, M. and Verma, M. (1998), A neural network for real-world postal address recognition, In P. K. Chawdhry, R. Roy, and R. K. Pant (Eds.). *Soft computing in engineering design and manufacturing*, London: Springer Verlag.

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

Fukunaga, K. (1990), *Introduction to Statistical Pattern Recognition*, 2nd ed., CA: Academic Press Inc.

Gustafson, D. E., and Kessel, W. C. (1979), Fuzzy clustering with a fuzzy covariance matrix, *Proceeding of IEEE Conf. Decision Contr.* San Diego, CA., pp 761-766.

Hornik, K.M., Stinchcombe, M., and White, H. (1989), Multilayer feedforward networks are universal approximators, *Neural Networks* 2, pp. 359-366.

Hwang, Y.-S. and Bang, S.-Y. (1997), Recognition of unconstrained handwritten numerals by a radial basis function neural network classifier, *Pattern recognition letters*. 18, pp. 657-664.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999), Data clustering: a review, *ACM Computing Surveys* (CSUR), 31(3), pp. 264-323.

Kovaĉeviĉ, D., and Lonĉariĉ, S. (1997), "Radial basis function based segmentation using a receptive field," in *10th IEEE Symposium on computer based medical systems* (CBMS'97), pp. 126-131.

Kubat, M. and Cooperson, M., Jr. (1999), "Initializing RBF-networks with small subsets of training examples," in *Proc. of the 16th National Conference on Artificial Conference*, AAAI-99, Orlando, FL, July 26-28, pp. 188-193.

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., and Vapnik, V. (1995), Comparison of learning algorithm for handwritten digit recognition, In F. Fogelman Souli and P. Gallinari (Eds.), *Proceedings of ICANN*, Nanterre, France.

Looney, C. G. (2002), A fuzzy classifier network with ellipsoidal Epanechnikovs, *NeuroComputing*, pp. 489-509.

Mak, M.-W. and Kung, S.-Y. (2000), Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification, *IEEE Transactions on Neural Networks*, 11(4), pp. 961-969

Michalewicz, Z. (1996), *Genetic Algorithms + Data Structures= Evolution Programs*, 3rd rev. and extended ed., n.p.:Springer Verlag.

Mount, D. W. (2001), *Bioinformatics: sequence and genome analysis*, New York: Cold Spring Harbor Laboratory Press.

Nauck, D., and Kruse, R. (1995), NEFCLASS - A Neuro-Fuzzy approach for the classification of data, presented at the *Symposium on Applied Computing*, Nashville, USA.

Quilan, J. R. (1994), *C4.5: programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Roger Jang, J.-S. and Sun, C.-T. (1993), Functional equivalence between Radial Basis Function Networks and Fuzzy Inference Systems, *IEEE Transactions, Neural Networks*, pp. 156-159.

Rumelhart, D., Hinton, G. E., and Williams, R. (1986), Learning internal representations by error propagation, In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. 1: Foundations*, Cambridge, MA: MIT Press, pp. 318-362.

Setnes, M., and Roubos, H. (2000), GA-Fuzzy Modeling and classification: complexity and performance, *IEEE Transactions on Fuzzy Systems*, 8(5), pp. 509-522.

Setnes, M., Babu_ka, R., Kaymak, U., and van Nauta Lemke, H. R. (1998), Similarity Measures in Fuzzy Rule Base Simplication, *IEEE Tranactions on Systems, Man, and Cybernetics-Part B: Cybernetics.* 28(3).

Wall, M. (1996), GALib: A C++ *Library of Genetic Algorithm Components*, Mechanical Engineering Department, Massachusetts Institute of Technology.

Wang, D. H. and Dillon, T. S. (2001), Extraction and optimization of classification rules for continuous or mixed mode data using neural net, *Proceedings of SPIE, International Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, 16-20, April, Orlando, FL. USA, pp. 38-45.

Wang. D. H., Dillon, T. S., and Chang, E. (2002), Data mining for constructing ellipsoidal fuzzy classifier with various input features using GRBF neural networks, in *Proceedings of the IEEE international conference on artificial intelligence systems*, Sept. 5-10, Divnomorskoe, Russia, 2002 (a).

Wang, D. H., Lee N. K., Dillon, T. S. and Hoogenraad, N. J. (2002), Protein sequences classification using modular neural networks, in *AI 2002: Advances in Artificial Intelligence, LNAI 2557, M. Mckay, J. Slaney, Eds.*, 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia, December 2002 (b).

Wedding, D. K. and Cios, K. J. (1998), Certainty factors versus parzen windows as reliability measures in RBF networks, *Neurocomputing 19 Special issue on RBF neural networks*, pp. 151-165.

Wu, C. H., Huang, H.Z., Arminski, L., Castro-Alvear, J., Chen, Y.X., Hu, Z. Z., Ledley, R. S., Lewis, K. C., Mewes, H.W., Orcutt, B. C., Suzek, B. E., Tsugita, A., Vinayaka, C. R., Yeh, L-S L., Zhang, J., and Barker, W. C. (2002), The Protein Information Resource: An Integrated Public Resource of Functional Annotation of Proteins, *Nucleic Acids Research*, 30(1), pp. 35-37.