Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 2
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

PROTEIN SEQUENCES CLASSIFICATION USING RADIAL BASIS FUNCTION (RBF) NEURAL NETWORKS

[*]Dianhui Wang, [+]Nung Kion Lee, [*]Tharam S. Dillon, [#]N. J. Hoogenraad

[*]Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, Australia
[+]Faculty of Cognitive Science and Human Development
University Malaysia Sarawak
[#]Department of Biochemistry
La Trobe University, Melbourne, Australia

ABSTRACT

A protein super-family consists of proteins which share amino acid sequence homology and which may therefore be functionally and structurally related. Traditionally, two protein sequences are classified into the same class if they have high homology in terms of feature patterns extracted through sequence alignment algorithms. As the sizes of the protein sequence databases are very large, it is a very time consuming job to perform exhaustive comparison of existing protein sequence. Therefore, there is a need to build an improved classification system for effectively identifying protein sequences. This paper presents a modular neural classifier for protein sequences with improved classification criteria. The intelligent classification techniques described in this paper aims to enhance the performance of single neural classifiers based on a centralized information structure in terms of recognition rate, generalization and reliability. The architecture of the proposed model is a modular RBF neural network with a compensational combination at the transition output layer. The connection weights between the final output layer and the transition output layer are optimized by delta rule, which serve as an integrator of the local neural classifiers. To enhance the classification reliability, we present two heuristic rules to apply to decision-making. Two sets of protein sequences with ten classes of super-families downloaded from a public domain database, Protein Information Resources (PIR), are used in our simulation study. Experimental results with performance comparisons are carried out between single neural classifiers and the proposed modular neural classifier.

1. INTRODUCTION

The aim of classification is to predict target classes for given input patterns. There are many approaches available for classification tasks, such as statistical techniques, decision trees [9] and the neural networks [1]. Neural networks have been chosen as technical tools for the protein sequence classification task because: (i) the extracted features of the protein sequences are distributed in a high dimensional space with complex characteristics which is difficult to satisfactorily model using some parameterized approaches; and (ii) the rules produced by decision tree techniques are complex and difficult to understand because the features are extracted from long character strings.

A protein super-family consists of protein sequence members that are evolutionally related and therefore functionally and structurally relevant with each other [5]. One of the benefits from this category grouping is that some molecular analysis can be carried out within a particular super-family instead of individual protein sequence with the completion of the DNA sequencing of whole genomes it has also become apparent that the function of most genes is still unknown and classification into functionally related groups will provide valuable information on protein function. Traditionally, two protein sequences are classified into the same class if they have highly homology in terms of feature patterns extracted through sequence alignment algorithms. These algorithms, for instance, SAM[10], MEME[11], iPro-Class [8], compare an unseen protein sequence with all the identified protein sequences and provide a score based on similarity of sequences. As the sizes of the protein sequence databases are large, it is a very time consuming job to perform exhaustive comparison of existing protein sequences. Therefore, it is useful and helpful to build an intelligent classification system for effectively identifying protein sequences. Motivated by this, artificial neural networks have been applied in this area in the past and the results obtained demonstrate some merits of the methodology. Basically, there are two types of neural models applicable for protein sequences classification task, such as systems, Self Organizing Mapping (SOM) networks [2,3] and Multilayer Perceptrons (MLP) [12,13].

The SOM networks can be used to discover relationships within a set of protein sequences by clustering them into different groups. The main limitations of SOM for protein sequence classification are: (i) it is time consuming and difficult to interpret the results [11]; and (ii) the selection of the size of Kohonen layer is subjective and usually involves a trial and error procedure. The MLP based classification systems have several undesired features arising from system design, which include, for instance, (i) the determination of the neural network architecture, (ii) the lack of interpretability of the "black box"; and (iii) time needed for training as the number of inputs is over 100. Once offline training of the neural network is accomplished, the resulting neural classifier is ready to be used for future protein sequence classification and only few seconds are needed to classify a new protein sequence.

Radial Basis Function (RBF) networks have received much attention due to their merits in architecture interpretability and learning efficiency [6,14]. It has been successfully applied in many real world applications. Comparative studies showed that the RBF network outperforms the MLP model in both classification performance and learning efficiency. Another reason to employ the RBF neural network to permit design of intelligent classification systems is that employ crisp or fuzzy classification rules, which are the core of knowledge-based systems.

The remainder of the paper is organized as follows: Section 2 gives some detailed information on the design of modular RBF network classifier. Section 3 presents a heuristic approach for enhancing classification reliability. The data preprocessing including features for the extraction of protein sequences, the experimental results are reported in Section 4 and conclusions are given in the last section.

2. DESIGN OF RBF NEURAL CLASSIFIERS

Figure 1 depicts the architecture for a fully connected RBF network. The network consists of N input features x, M hidden units with center C_i and L output units y_i. The activation functions ϕ in the hidden units are the Gauassian functions defined as:

$$\phi_j(x) = \exp(-\| x - C_j \|^2 / 2\rho_j^2) \quad (1)$$

The output is calculated by

$$y_k(x) = f(\sum_{j=1}^{M} w_{kj}\phi_j(x) + w_{k0}), k = 1, 2, ..., L \quad (2)$$

where $f(x) = (1 + \exp(-x))^{-1}$ is a sigmoid function. A supervised algorithm, APC-III [14], is adopted to determine the hidden units centers and widths.

In this paper, we investigate the effect of learning criteria on classification performance. Two different objective functions for training neural classifiers are employed, i.e., the means squared error (MSE) cost function and the cross entropy (CE) [1] cost function:

$$MSE = \frac{1}{2}\sum_p \sum_{j=1}^{L} (t_i^p - y_i^p)^2 = \frac{1}{2}\sum_p E(p) \quad (3)$$

$$CE = -\sum_p \sum_{k=1}^{L} t_k^p \ln y_k^p = -\sum_p L(p) \quad (4)$$

where t_i^p and y_i^p are the i-th target and the network output for the p-th input pattern, respectively.

The activation function at the output layer of the RBF network using CE learning criteria is the *softmax* operation, i.e.,

$$f(z_i) = \exp(z_i) / (\sum_{j=1}^{L} \exp(z_j))^{-1} \quad (5)$$

where z_i is the network input to output unit i.

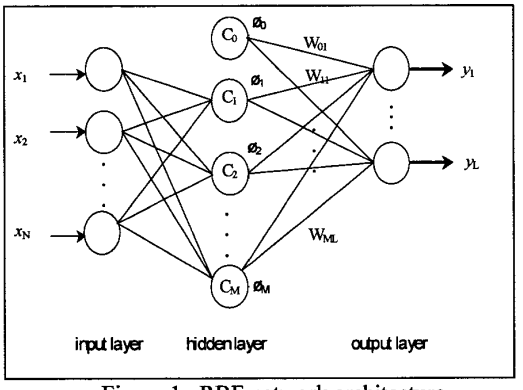

Figure 1. RBF network architecture

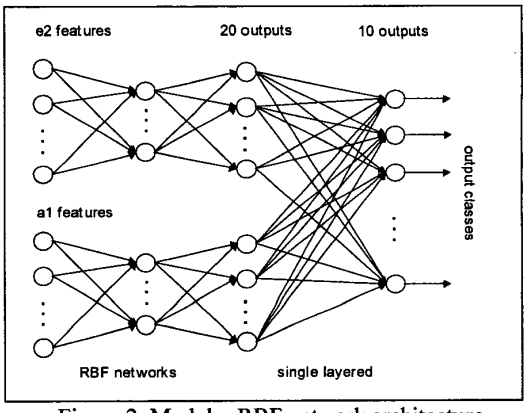

Figure 2. Modular RBF network architecture

There have been many attempts in the past to combine the outputs of modular neural classifiers. The main purpose is to reduce the model variance of the classification systems. Two well-known methods are the bootstrap and ensembles of committee. For protein

classification, a variety of features can be extracted from a protein sequence. The features combination can produce various feature subsets with different strength and characteristic. Besides the global features, the local features or motifs are also applicable to classifier design. To take advantage of the different set of features, it is desirable to improve the classification system performance by using different features as the input to the network and effectively combining the local classification results. Figure 2 shows our proposed neural classifier architecture, which consists of two RBF networks that use different input features e2 and a1, respectively.

The outputs of these two networks are applied as inputs to the third single layered network to produce the final class prediction. The two RBF networks have the exact same output classes. The single layered network can be viewed as a way to integrate the results contributed by the two local RBF neural classifiers. The advantages of this architecture are: (i) it reduces the model variance of a single classifier by assigning different weights to the networks output; (ii) it speeds up the learning process because of the use of less features for each RBF neural network; and (iii) it enhances the scalability and robustness of the intelligent classification system with respect to the weights decay. The neural network is trained by two stages, i.e., individual trainings of the RBF network classifiers followed by a training of the single layered network using delta rule. The details of the learning algorithm are omitted here.

3. CLASSIFICATION CRITERIA

The network outputs can be interpreted as the target class posterior probabilities [4]. A test pattern x is assigned to class i if $p(C_i|x) > p(C_j|x)$ for all $i \neq j$. This classification criterion suffers from several drawbacks although it is widely used in the literature. Firstly, it does not define the confidence of the output value. Suppose that two of the largest network outputs are 0.45 and 0.44, respectively. By using this criterion, the pattern will be assigned to the class with output 0.45. However, we have less confidence on the classification quality in such a situation caused by the small bias. Secondly, if the maximum output value is less than some certain threshold, for example, 0.2, it still quite difficult to assure the result has sufficient reliability. This implies that the commonly used maximum posterior probabilities classification criterion or maximum component principle will not provide a secure prediction. To overcome this shortcoming in decision-making, we suggest a heuristic approach for enhancing the classification confidence and quality. The heuristic measure can be used in decision-making with improved faith and tradeoff between the classification rate, misclassification rate and un-classification rate. Our suggested heuristic classification criterion are as follows:

Rule 1: IF ($pred(x) \geq beta$ AND $diff(x) \geq alpha$), THEN x is classified/misclassified

Rule 2: IF ($pred(x) < beta$ OR $diff(x) < alpha$), THEN x is unclassified

To apply these rules, the network prediction output values $pred(x)$ for the pattern x are sorted in decreasing order. The $diff(x)$ represents the difference between the largest output value and second largest output value of the neural classifier. The classification performance is controlled by the two parameters *alpha* and *beta*. The *beta* value essentially characterizes the confidence of the predicted result. Whilst the *alpha* value controls the quality of the classification by only allowing the $diff(x)$ value to be large enough to confidently identify the protein classes. To simplify the rules above, in this paper we introduce a mathematical expression to characterize the relationship between the two parameters in the rule, that is,

$$alpha = \frac{beta}{1+beta} \quad (6)$$

Obviously, the value of alpha is proportional to the value of beta. Figure 3 show the trade-off between classification rate, miss-classification rate and un-classification rate for difference values of *beta*. It shows that the classification rate (cls) and misclassification rate (miscls) are decreased as the value of *beta* increases. Whereas the un-classification rate (uncls) is increased as the value of *beta* increases. By setting an appropriate value of *beta*, a reliable classification performance with higher quality and confidence is attainable.

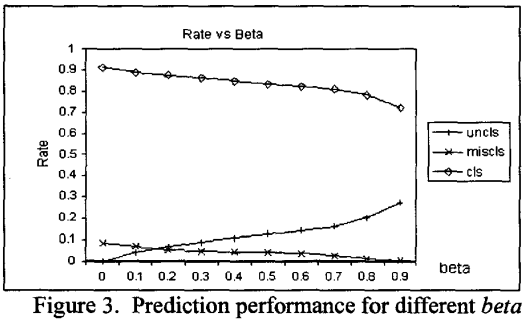

Figure 3. Prediction performance for different *beta* values

4. SIMULATION RESULTS

The protein sequences are transformed from DNA sequences using the predefined genome code. Protein sequences are more reliable than DNA sequence because of the redundancy of the genetic code [5]. Two protein sequences are believed to be functional and structurally related if they show similar sequence identity or

homology. These conserved patterns are of interest for the protein classification task.

A protein sequence is made from combinations of variable length of 20 amino acids $\sum$ = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y} (using the ... letter code). The n-grams or k-tuples [12] features will be extracted as an input vector of the neural network classifier. The n-gram features are a pair of values (v_i, c_i), where v_i is the feature i and c_i is the counts of this feature in a protein sequence for i = 1... 20^n. In general, a feature is the number of occurrences of an animal in a protein sequence. These features are all the possible combinations of n letters from the set $\sum$. For example, the 2-gram (400 in total) features are (AA, AC, ..., AY, CA, CC, ..., CY,..., YA, ..., YY) . Consider a protein sequence VAAGTVAGT, the extracted 2-gram features are {(VA, 2), (AA, 1), (AG, 2), (GT, 2), (TV, 1)}. The 6-letter exchange group is another commonly used piece of information. The 6-letter group actually contains 6 combinations of the letters from the set $\sum$. These combinations are A={H,R,K}, B={D,E,N,Q}, C={C}, D={S,T,P,A,G}, E={M,I,L,V} and F={F,Y,W}. For example, the protein sequence VAAGTVAGT mentioned above will be transformed using 6-letter exchange group as EDDDDEDDD and their 2-gram features are {(DE, 1), (ED, 2), (DD, 5)}. We will use e_n and a_n to represent *n*-gram features from a 6-letter group and 20 letters set. Each sets of n-grams features, i.e., e_n and a_n, from a protein sequence will be scaled separately to avoid skew in the counts value using equation (7) below:

$$\bar{x} = \frac{x}{L-n+1} \tag{7}$$

where x represents the count of generic gram feature, $\bar{x}$ is the normalized x, which will be the inputs of the neural networks; L is the length of the protein sequence and n is the size of n-gram features.

In this simulation study, the protein sequences covering ten super-families (classes) were obtained from the PIR databases comprised by PIR1 and PIR2 [8]. The 949 protein sequences selected from PIR1 were used as the training data and the 533 protein sequences selected from PIR2 as the test data. The ten super-familes to be trained/classified in this study are: Cytochrome c (113/17), Cytochrome c6 (45/14), Cytochrome b (73/100), Cytochrome b5 (11/14), Triose-phosphate isomerase (14/44), Plastocyanin (42/56), Photosystem II D2 protein (30/45), Ferredoxin (65/33), Globin (548/204), and Cytochrome b6-f complex 4.2K(8/6). The 56 features were extracted and comprised by e_2 and a_1.

To compare the performance with MLP network classifiers, an MLP architecture 56-15-10 was selected as it has the same number of weights in the model. The NevProp [7] simulation software was applied for MLP network training, where an automatic stopping criterion to define the optimal target mean squared error and learning rate were offered.

Parameters	RBF-MSE	RBF-CE	BP-MSE/CE
Learning rate (LR)	0.015	0.02	Optimized
Stopping Criterion	Cross-validation	Cross-validation	MSE
No.Hidden units	71	71	15
Centers LR	0.005	0.01	-
Widths LR	0.005	0.01	-
Momentum	0.9	0.9	Default

Table 1. Parameter settings in neural classifiers training

To avoid the over-fitting of training for the RBF network classifiers, we created a set of perturbed sequences made by 20-30% training data by adding small noises, and used them to check the classification performance on-line. The learning process was stopped once the recognition rate (with *beta*=0) inspected started decreasing for the noise sequences. Table 2 and Table 3 show the results for training data set and test data set with *beta*=0.4. All of the neural classifiers perform well on the training data set. However, the performance for the test data set varied extremely, which demonstrates the difference in generalization capability. It has been observed that the modular RBF network classifier with MSE learning cost function performs better on the training data set, but performs worse on the test data set. The CE learning criteria results better results for both MLP and RBF networks.

NN Classifier	CR	MR	UR
BP-CE	98.63 (936)	0.32 (3)	1.05 (10)
BP-MSE	99.37 (943)	0.11 (1)	0.52 (5)
RBF-CE	95.47 (906)	1.58 (15)	2.95 (28)
RBF-MSE	99.26 (942)	0.00 (0)	0.74 (7)
MRBF-CE	98.42 (934)	0.42 (4)	1.16 (11)
MRBF-MSE	99.78 (947)	0.11 (1)	0.11 (1)

Table 2. Performance comparison for training data set (*beta*=0.4)

In Table 3, we also give the performance results obtained using average and product combination strategies [15] for the test data set. The proposed modular RBF network classifier outperforms both the average combination method and the product combination method. This suggests that the single layered network makes better decision with information fusion of the local classifiers. Figure 4 shows the classification rates from different

neural classifiers versus the varying values of the *beta* for the test data set. The overall evaluation based on the experimental results demonstrate that the RBF neural networks with CE learning cost function can produce better classification performance in terms of generalization. The modular RBF network classifiers with CE learning cost function and linear fusion of different global features RBF network classifiers can further improve classification performance.

NN Classifier	CR	MR	UR
BP-CE	84.24	3.75	12.01
BP-MSE	85.00	4.00	11.00
RBF-CE	86.00	6.00	8.00
RBF-MSE	83.5	3.60	12.90
MRBF-CE	87.00	8.00	5.00
MRBF-MSE	83.15	5.80	11.05
Average	79.21	4.87	15.92
Product	78.09	2.43	19.45

Table 3. Performance comparison for test data set (*beta*=0.4)

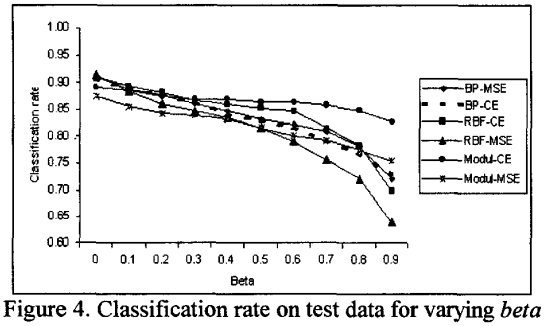

Figure 4. Classification rate on test data for varying *beta*

5. CONCLUSIONS

This paper studies the protein super-family classification problem using a modular RBF neural network with transition output fusion. The main investigations in this study containing: (i) the design for a modular RBF neural networks using different global features for predicting the protein patterns; (ii) a proposal for a heuristic approach for enhancing the quality and reliability in decision-making; and (iii) a comparative study using 10 classes of protein sequence for the classification problem. The experimental results demonstrate the potential of our proposed techniques, which confirm that the modular RBF network classifier with CE learning cost function and linear fusion at the transition output layer presented here is the best candidate for this domain application. Further studies on scalability and robustness of this modular RBF neural protein classification system are in progress.

AKNOWLEDGEMENT

This project is financially supported by VPAC grants and FSTE small grants (# 104110) at La Trobe University.

REFERENCES

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

[2] H. C Wang, Dapazo J., L. G. De La Fraga, Y. P. Zhu, Carazo J. M., Self-organizing tree-growing network for the classification of protein sequences, *Protein Science*, (1998) 2613-2622

[3] Ferran EA, Pflugfedder B, Ferrarap, Self Organized neural maps of human protein sequences, *Protein Science*, 3(1994) 507-521

[4] S. Lawrence, I. Burns, Andrew Back, Ah Chung Tsoi, and C. Lee Giles, Neural Network Classification and Prior Class Probabilities, *Neural Networks Tricks of the Trade*, Lecture Notes in Computer Sciences, Springer, 1524 (1998)

[5] I. Jonassen, *Methods for finding motifs in sets of related biosequences*, PhD Thesis, Department of Informatics, University of Bergen, (1996)

[6] J. Moody and C. Darken, Faster learning in networks of locally-tuned processing units, *Neural Comput.*, 1(1989) 281-294

[7] NevProp-v3 *http://www.scs.unr.edu/nevprop/*

[8] PIR, *http://pir.georgetown.edu*

[9] Quilan, J. R. 1994, C4.5: programs for machine learning, San Mateo, CA Morgan Kaufmann

[10] SAM: Sequence Alignment and Modeling Software System, Baskin Center for Computer Engineering and Science, *http://www.cse.ucsc.edu/researchcompbio /sam.html*

[11] MEME: Multiple EM for Motif Elicitation UCSD Computer Science and Engineering *http://meme.sdsc.edu*

[12] Wu, C. H., Berry, M., Shivakumar, S. & McLarty, J., Neural networks for full-scale protein sequence classification: sequence encoding with singular value decomposition, *Machine Learning*. 21(1995) 177-193

[13] Wu C. H., Artificial neural networks for molecular sequence analysis. *Computers Chemistry, 21(1997) 237-256*

[14] Y. S. Hwang, and Y. S. Bang. An Efficient Method to Construct a Radial Basis function Neural Network Classifier, *Neural Networks, 10(1997) 1495-1503*

[15] David M.J. Tax, Martijin van Breukelen, Rober P.W. Duin, and Josef Kittler,Combining multiple classifiers by averaging or by multiplying?, *Pattern Recognition 33* (2000) 1475-1485.