UDC 621.8

V. Lytvynenko
Kherson National Technical University
Dept. of Informatics & Computer Science

SYNTHESIS OF RBF-NETWORK FOR PREDICTION OF SECONDARY PROTEIN STRUCTURE

© *Lytvynenko V., 2014*

Запропоновано методологію синтезу радіально-базисних мереж для вирішення проблеми білкового середнього передбачення структури за допомогою алгоритму вибору клонів. Щоб вирішити цю проблему було використано метод "один проти всіх". Обчислювальні експерименти щодо випробуваного зразка показали, що точність прогнозування сягає до 72 %, що вказує на високу точність запропонованого способу.

Ключові слова: алгоритм клонального відбору, радіальна базисна функція, метод "один проти всіх", прогнозування, вторинна структура білка.

In this paper we propose the methodology of team radial-basis networks synthesis for solving the problem of protein secondary structure prediction using clonal selection algorithm. To solve such problem the method of "one against all" have been used. The carried out computational experiments on test sample have shown that the prediction accuracy allows to achieve up to 72 %, indicating a high accuracy of the proposed method.

Key words: clonal selection algorithm, radial basis function, "one against all" method, predicting, the secondary structure of protein.

1. Introduction

Proteins are large biological molecules with complex structures and constitute to the bulk of living organisms: enzymes, hormones and structural material [1]. The function of a protein molecule in a given environment is determined by its 3-dimensional (3-D) structure [1]. Protein 3-D structure prediction directly from amino acid sequences still remains as an open and important problem in life sciences. The bioinformatics approach first predicts the protein secondary structure (PSS) which represents an 1-D projection of the very complicated 3-D structure of a protein [3]. Secondary structures are regular structural elements which are formed by hydrogen bonds between relatively small segments of the protein sequence. Often the driving force for the formation of a secondary structure is the saturation of backbone hydrogen donors (NH) and acceptors (CO) with intramolecular hydrogen bonds. This saturation allows the protein to bury hydrophobic side chains in its interior (hydrophobic core) without conflicting with the polar backbone. There are three common secondary structures in proteins, namely a -helix, b -strand , and coil [12] . An a -helix is formed from a connected stretch of amino acids. The a -helix is characterized by hydrogen bonds along the chain, which are almost coaxial. The a -helix is the most abundant helical conformation found in globular proteins. The average length of an a -helix is around 10 residues. A b -strand is the principal component of a b -sheet. The b -sheet is characterized by hydrogen bonds crossing between chains. Each participating b -strand in a b -sheet is not continuous in terms of the primary sequence and does not even have to be close to another b -strand in the sequence. A b -strand has a sequence of 5-10 residues in a very extended conformation. Approximately one-third of all residues in globular proteins are contained in coils. The coils in a protein serve to reverse the direction of the polypeptide chain. Coils vary in length.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

The goal of secondary structure prediction is to classify a pattern of residues in amino acid sequences to a corresponding secondary structure element: an α-helix (H), β-strand (E) or coil (C, the remaining type). Many computational techniques have been proposed in the literature to solve the PSS prediction problem, which can be broadly fallen into three categories: (a) statistical methods, (b) neural network approaches, and (c) nearest neighbor methods. The statistical methods are mostly based on likelihood techniques [4, 5, 6]. Neural network approaches use residues in a local neighborhood or window to predict the secondary structure at a particular location of an amino acid sequence [7, 8]. The nearest neighbor method often uses the *k*-nearest neighbor techniques [9, 10]. SVMs have been earlier applied to PSS prediction [11]. One of the drawbacks in this approach is that the method does not capture the global information of the amino acid sequence due to the limited size of the local neighborhood. Additionally, the method only constructs a multi-class classifier by combining several binary classifiers.

Despite the existence of many approaches, the current success rates of existing approaches are insufficient; further improvement of the accuracy is necessary. Most existing secondary structure techniques are single-stage approaches, except the PHD [8] and PSIPRED [9] methods which combined two multi-layer perceptron (MLP) networks. Single-stage approaches are unable to find complex relations (correlations) among different elements in the sequence. This could be improved by incorporating the interactions or contextual information among the elements of the output sequence of secondary structures. We argue that it is feasible to enhance present single-stage approaches by augmenting with another prediction scheme at their outputs and propose to use SVMs as the second-stage.

This paper investigates the use of multi-class RBF neural networks which we synthesize using clonal selection algorithm for PSS prediction. We present new multi-class techniques based on binary classifiers to PSS prediction.

2. Data and methods

2.1. Problem definition

One main sub problem of the domain is '*Protein Secondary Structure Prediction'*. The primary sequence of a protein can be represented as [13] $\{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}^n$, where the letters are the one letter codes of the amino acid residues (total 20 possible amino acids) and n is the length of the protein to be predicted. The secondary structure of the sequence having length n is $\{a, b, c\}^n$, where a, b, c are different secondary structure classes. So, the problem of secondary structure prediction can be represented as a mapping problem as follows [13]:

$$\{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}^n \rightarrow \{a, b, c\}^n$$

2.2. Orthogonal encoding of amino acid

Orthogonal encoding of amino acid types has been used in many bioinformatic neural network models: 20 input units are assigned to describe one protein residue. In the 20-dimensional space, the vector [1, 0, 0, 0…0, 0, 0] represents alanine, and [0, 0, 0 …0, 0, 0, 1] stands for valine. With this encoding, a typical input window of 13 residues requires 260 (13×20) input units. It can easily lead to large input layers, many connecting weights, and hence complex models. Without sufficient data to support training, over-complex models are prone to overfitting. Unfortunately, in many bioinformatic problems, huge data sets can be simply unavailable. Even when they are available, analysing them is often very computationally demanding. Simplified encoding schemes use less input units to describe a given amino acid sequence; thus, we can use smaller models to describe the same phenomena. By introducing these simplified models, we can reduce the reliance on huge data sets and improve performance. To increase the level of neural network generalization, in work [14] defined a 10-unit input scheme for representation of amino acid type. Each amino acid was described using ten numbers. In work [15] their representation was based on the amino acid features described by: each unit corresponds to one biochemical feature; amino acids sharing many features have similar codes. In work [16] suggested two differing properties, ''sequence-derived hydrophobicity'' and ''sequence-derived polarity'', based on correlations in protein sequences. In work

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

[17] applied an adaptive encoding neural network to find automatically a classifier with a low-dimensional encoding matrix. Their encoding scheme was tested on the prediction of cleavage sites in human signal peptides of secretory proteins.

2.3. Data Set

The set 126 nonhomologous globular protein chains used in the experiment of Rost and Sander [18], referred to as the RS126 set, was used to evaluate the accuracy of the classifiers. The dataset contained 23349 residues with 32% a -helix, 23% b -strand, and 45% coil. Many current generation secondary structure prediction methods have been developed and tested on this dataset.

The RS126 set is available at http://www.compbio.dundee.ac.uk/~www-jpred/data/. The single stage approaches and second-stage approaches were implemented, with multiple sequence alignments, and tested on the dataset, using a sevenfold cross validation technique to estimate the prediction accuracy. With sevenfold cross validation approximately one-seventh of the database was left out while training and, after training, the left one-seventh of the dataset was used for testing. In order to avoid the selection of extremely biased partitions, the RS126 set was divided into seven subsets with each subset having similar size and content of each type of secondary structure.

2.4. Synthesis of radial-basis network

As the classifier, in general, is called a function that for object attributes vector makes the decision: to which of classes it belongs:

$$F : \Re^n \to Y . \tag{1}$$

The function *F* reflects the vector characteristics space in the space of class labels *Y*. In the case of two classes $Y = \{0,1\}$, '1' corresponds to case of event you are looking for; '0' - an event not found. We consider the option of training with a teacher (***supervised learning***), when for the classifier training available a set of vectors $\{x\}$ for which is known their true identity to one of the classes.

In binary classification the class identifiers can be interpreted as states of the system (active or passive, normal or abnormal), which are presented by number of properties.

The vector of properties determines the system state. Each state of $x^i = \left(x_1^i, \mathbf{K}, x_n^i\right) \in [0,1]^n$ is represented by set $U \subseteq [0,1]^n$. The properties vector elements can be scaled or normalized in the interval $[0,1]$.

The properties vector set of $Positiv \subseteq U$ represents the normal state of system. Its complement is called *Negativ* and determined as *Negativ = U - Positiv*. In some cases, we will define a set *Positiv* (or *Negativ*), using its characteristic function [26] $c_{Positiv} : [0,1]^n \to \{0,1\}$:

$$c_{Positiv}(\mathbf{r}_x) = \begin{cases} 1 & if \ \mathbf{r}_x \in Positiv \\ 0 & if \ \mathbf{r}_x \in Negativ \end{cases} . \tag{2}$$

For a given set of positive examples $Positiv' \subseteq Positiv$ we have to evaluate the characteristic function ($c_{Positiv}$) of normal space, which should have the ability to solve whether the observed state of positive or negative.

The entire set of neural networks Δ be divided into subset *A* caused by the chosen topology [27] of RBF-neural network (number of RBF-neuron network is Γ). Within each class $RBFN_i \subset RBFN$ the neural networks are characterized by an additional set of parameters: the number of inputs *n*; the set of synaptic weights of the output layer $W = \left\{w^i, i = 1, \mathbf{K}, p\right\}$; the number of RBF-neurons of network $\Gamma = \left\{g^i, i = 1, \mathbf{K}, p\right\}$; the centers of RBF $C = \left\{c^i, i = 1, \mathbf{K}, p\right\}$; the parameters of RBF $\Sigma = \left\{s^i, i = 1, \mathbf{K}, p\right\}$; the parameter of the output neuron activation function of *i*-th network $\mathrm{A} = \left\{a^i, i = 1, \mathbf{K}, p\right\}$; in the case of RBF-neural network scale $S = \left\{s^i, i = 1, \mathbf{K}, p\right\}$ of RBF shifts to time axis $T = \left\{t^i, i = 1, \mathbf{K}, p\right\}$. Thus, parameters vector RBF-neural networks $q = \{\Gamma, W, C, \Sigma, \mathrm{A}\}$ is formed. The natural criterion for selecting RBF-neuron network will function defined by the standard RMS for any input.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Thus, the task of RBF-neural network synthesis can be reduced to optimization problems such as:

$$F^* = F(q^*) = \min F(q), \qquad (3)$$
$$a_1 \le x_1 \le b_1, ..., a_n \le x_n \le b_n,$$

where the function F is not imposed any restrictions, such as differentiation, Lipschitz condition, continuity, etc.

The problem solution of multiparametric function optimization of the form (3) could be used the appropriate operators of clonal selection.

RBF-network consists of input, single hidden (radial-basis) and linear (output) layer. Input layer consists of sensors that connect the network with the external environment. The hidden layer neurons act by the principle of centering on the elements of training sample. As the center stands the weight matrix (W^r). Box (*dist*) the Euclidean distance between input vector (X) and the corresponding center is calculating. Around each center there is a region, called the radius. Radius (sensitivity of the network) is adjusted by means of the smoothing coefficients vector $(s_1, ..., s_m)$.

The conversion function (usually Gauss - $f(x) = e^{-\frac{(x-c)^2}{2s^2}}$), which varies from 0 to 1, determines the output of hidden layer. Output layer contains the usual linear or sygmoidal neurons and by adjustment of their weights (W^l) determines the output network.

The behavior of RBF-network depends largely on the number and position of radial basis function of hidden layer. Indeed, for any real n-dimensional input vector $x = (x_1, x_2, ..., x_n)$, where $x \in X \subset \Re_n$, the network output will be determined as follows:

$$y_i = \sum_{k=1}^{m} w_{ik}^l f_k\left(dist(x, w_k^r), s_k\right), \qquad (4)$$

where $w_{ik}^l \in W^l$ - is the weight of linear layer; $w_k^r \in W^r$ - centers of radial-basis functions. If as basic functions used Gauss-function one, then

$$f_k(x) = -\frac{dist(x, w_k^r)^2}{2s_k^2}, \; k = \overline{1, m}. \qquad (5)$$

Architecture of RBF-neural network used by us for solving the classification tasks presented in Fig. 1. The hidden layer neurons are the RBF-neurons. As the RBF parameters used its scale (s) and shift (t) in terms of the time axis.

Output layer contains the usual linear or sigmoidal neurons and by adjustment of their weights (W^l) determines the output network.

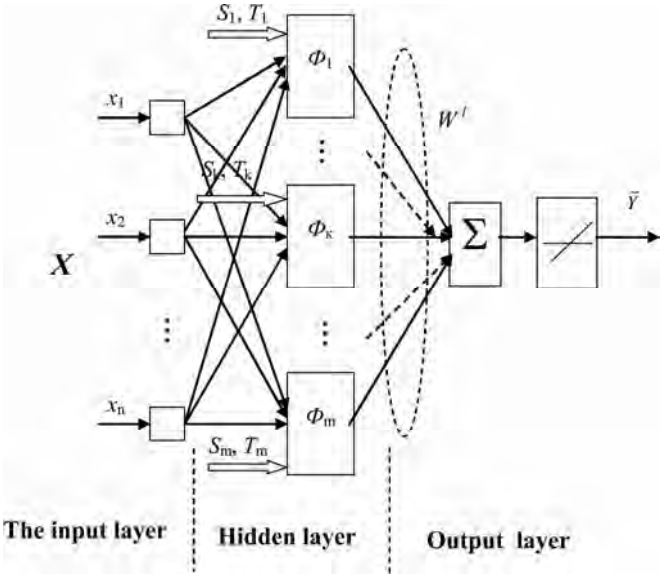

Fig. 1. Generalized RBF-neural network architecture for solving classification problems [19]

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Based on the architecture, and at $p = 1$ (for time series), the output of the neural network will be determined as follows:

$$y_i = \sum_{k=1}^{m} w_k^l \Phi_k \left(x, S_k, T_k \right), \tag{6}$$

where $x = \left(x_1, x_2, ..., x_n \right)$, $x \in X \subset \Re_n$ - an arbitrary *n*-dimensional vector of input variables; $w_k^l \in W^l$ are the weights of linear layer; S_k - the scale parameters for the hidden layer neurons; T_k is the displacement parameters for the hidden layer neurons.

Based on the architecture of neural network (Fig. 2), as parameters are customizable, one can distinguish the following [19]:

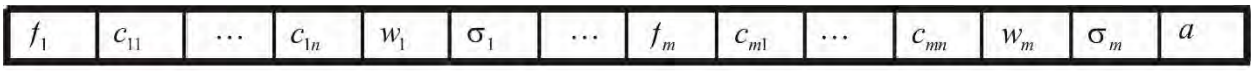

Fig. 2. Encoding configurable parameters RBF neural network in the form of antibodies (chromosomes) where: n – number of inputs; m – number of neurons; f - ON / OFF neuron (0 or 1); c – RBF centers; w – synaptic weights of output layer; s – RBF parameter; a – parameter of function activate the output neuron.

In general, the procedure of the synthesis of each neural network for each class, are performed in accordance with the procedure presented in Fig. 3.

In this chapter the synthesis RBF-neural networks, that aimed at solving classification tasks. The classifier, in general, is a function that on attributes vector base of the object makes a decision, which is a class it belongs, respectively (1). In the binary classification the class identifiers can be interpreted as states of the system (active or passive, normal or abnormal), which presented by the number of properties. Types of basis functions, their number, type and parameters of activation function in the linear layer, are setting up as the parameters of AIS.

Study and synthesis of neural networks groups is carried out by the scheme shown in Fig. 3

As an objective function and the function affinity selected the RMS error of network on the training data. The training procedure has the following characteristics [19, 20]:

– selection is implemented on the tournament selection, which allows to control convergence and to maintain the diversity of population at the required level;

– because of the binary encoding specific, scheme of mutation has been proposed, whereby the probability of single bit line changing depends not only on antibody affinity in general, but also on the significance of this bit.

In the case of RBF-neural networks, the network behavior is largely dependent on the number and position of radial basis function of hidden layer. Indeed, for any real *n*-dimensional input vector $x = \left(x_1, x_2, ..., x_n \right)$, where $x \in X \subset \Re_n$, the network output will be determined as follows:

$$y_i = \sum_{k=1}^{m} w_{ik}^l f_k \left(dist\left(x, w_k^r \right), s_k \right), \tag{7}$$

where $w_{ik}^l \in W^l$, $i = \overline{1, p}$ are the linear layer weights; $w_k^r \in W^r$ are the center of radial-basis functions.

If as basic function is used the Gauss function, then

$$f_k(x) = -\frac{dist(x, w_k^r)^2}{2 s_k^2}, \; k = \overline{1, m}. \tag{8}$$

In the context of the classification problem debugging the network is to find functions $y : \Re_n \to \Re$ that satisfy the equation (7.5) at $p = 1$. Let we have the sample is composed of S training data points $X_1, ..., X_S$, $X_i \in \Re_n$. If you know the output value for each of these points $d_1, ..., d_S$, $d_i \in \Re$,

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

then each basis functions can be centered on one of the points X_i. Thus, in the limiting case the number of centers and, accordingly, the hidden layer neurons is equal to the number of data points in training sample $m = S$.

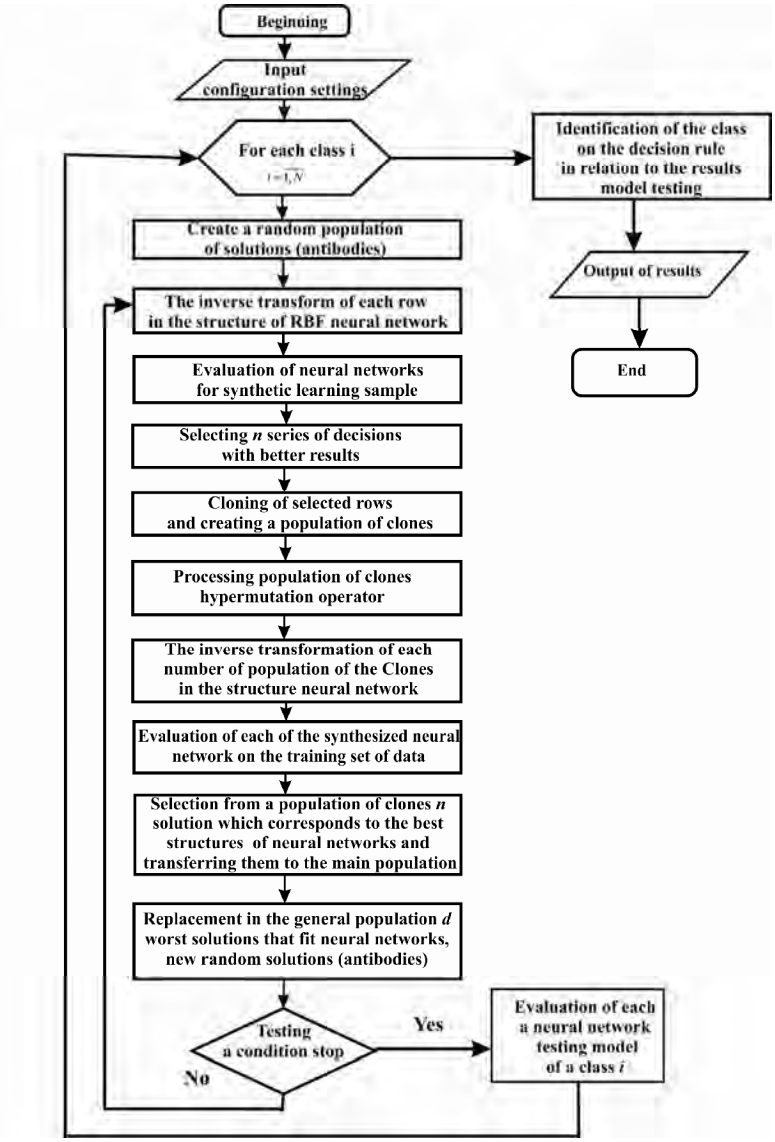

Fig. 3. The procedure of the RBF-neural network synthesis using the clonal selection

Synthesis of collective neural networks, where each neural network recognizes only a single class, is similar to the procedure described for RBF-neural networks according to the adjustment parameters presented in Fig. 4.

Many discriminative methods, including Support vector machine, neural network and classifiers based on the artificial immune systems, are often most accurate and efficient when dealing with two classes only (they can deal with more classes, but usually at reduced accuracy and efficiency) . For large number of classes, higher-level multi-class methods are developed that utilize these two-class classification methods as the basic building blocks.

To solve the problem we used the strategy of one-versus-all based on multiparameter optimization function of the form (3) use the corresponding operators clonal selection algorithm [21]. Types of basis functions and the activation function of the linear layer are defined as parameters to clonal selection algorithm. Learning and synthesis collectives of neural networks is performed according to the scheme shown in Fig. 4.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

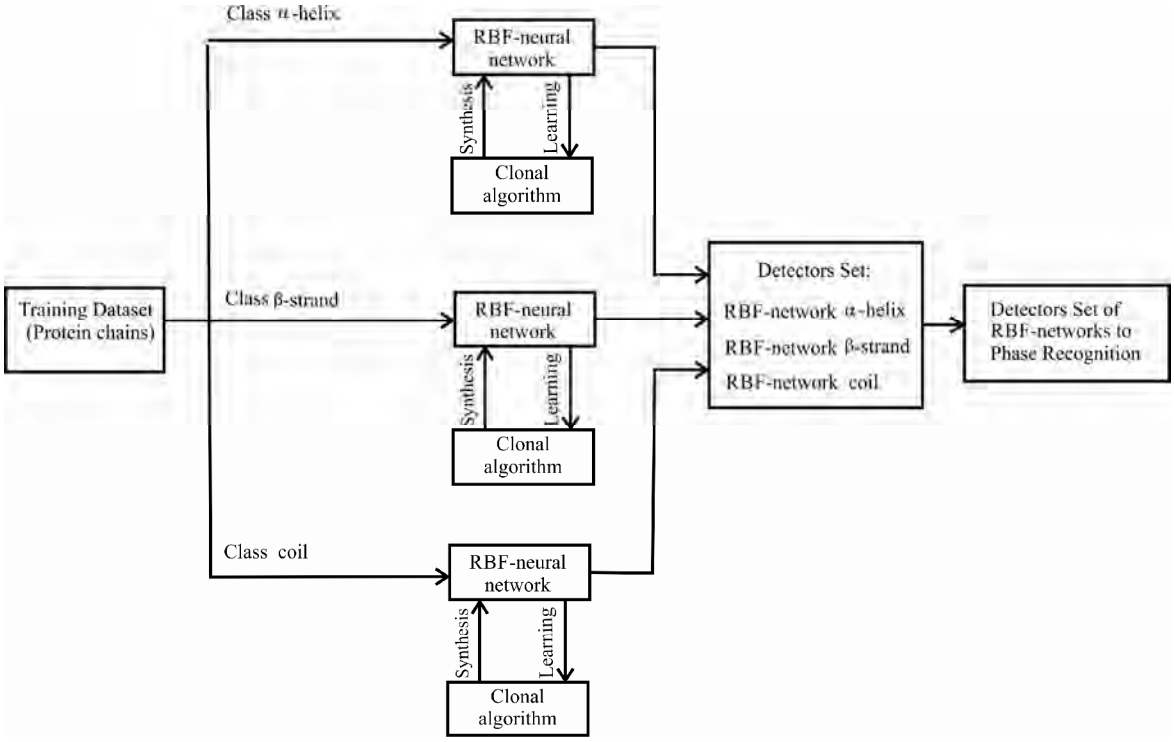

Fig. 4. Synthesis of binary classifiers collective RBF-neural networks

The simplest approach is to reduce the problem of classifying among K classes into K binary problems, where each problem discriminates a given class from the other K −1 classes [22]. For this approach, we require N = K binary classifiers, where the k th classifier is trained with positive examples belonging to class k and negative examples belonging to the other K − 1 classes. When testing an unknown example, the classifier producing the maximum ouput is considered the winner, and this class label is assigned to that example. In work [22] state that this approach, although simple, provides performance that is comparable to other more complicated approaches when the binary classifier is tuned well [23].

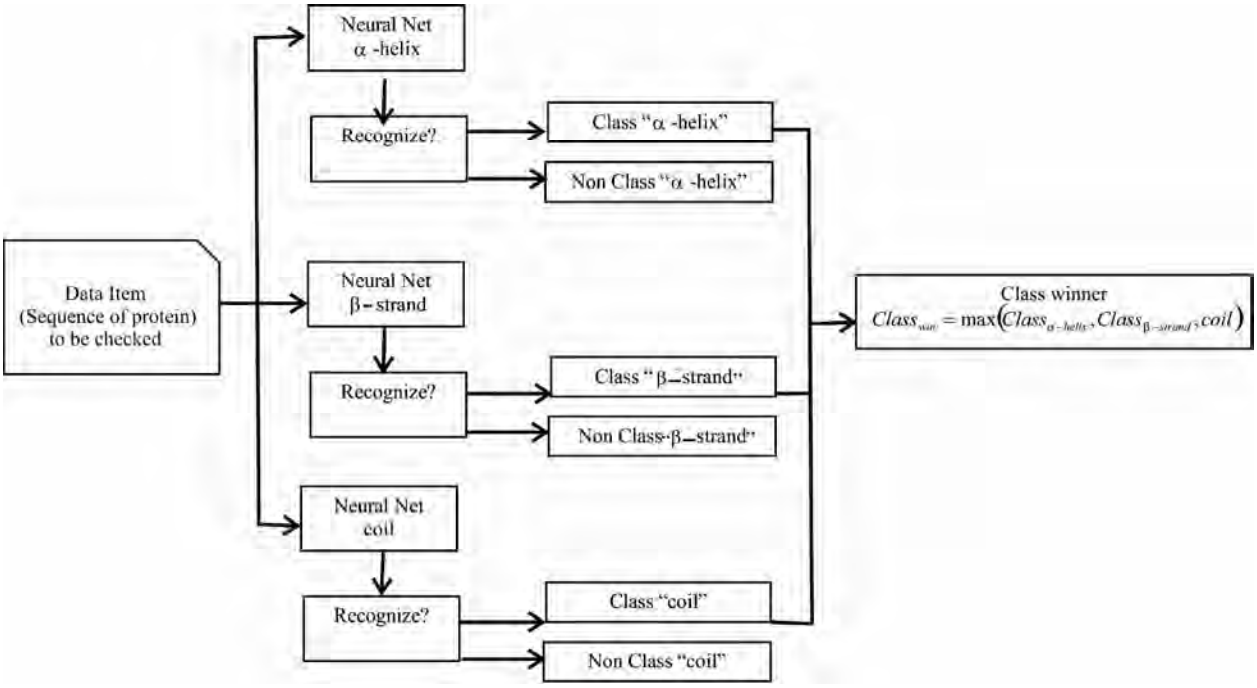

Fig. 5. Testing of synthesized neural networks

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

The identification of the system state performed by the expression (9):

$$Class_{win} = \max\left(Class_{a=helix}, Class_{b-s\tan d}, Class_{coil}\right) \tag{9}$$

2.5. Measurements of accuracy

The most commonly reported measure of secondary structure prediction accuracy is the success rate, or Q_3. This is the overall percentage of correctly predicted residues of all **3** types [24] , i.e.,

$$Q_3(\%) = \frac{R_{helix} + R_{sheet} + R_{coil}}{N} \times 100 \tag{10}$$

Here, R_i, is the number of correctly predicted residues of type i, and N is the total number of residues.

Although the Q_3 score provides a quick measure of the accuracy of the algorithm, it does not account for differing success rates on different types of secondary structure. We therefore also calculated the correlation coefficients [25] for prediction of helix C_H, sheet C_E, and coil C_C.

$$C_a = \frac{(p_a \cdot n_a) - (u_a \cdot o_a)}{\sqrt{(n_a + u_a)(n_a + o_a)(p_a + u_a)(p_a + o_a)}} \tag{11}$$

where p_a is the number of positive cases that were correctly predicted, n_a is the number of negative cases that were correctly rejected, o_a is the number of overpredicted cases (false positives), and u_a is the number of underpredicted cases (misses). Similar expressions hold for C_b , and C_{coil} . The Q_3 measure will be used to assay the overall success rate of network models during learning, although it is not as good an indicator as the individual correlation coefficients.

3. Results and discussion

We studied the dependence of testing success rate on the size of the input window using a standard network with 15 hidden units (RBF-functions). The results shown in Table 1 indicate that when the size of the window was small the performance on the testing set was reduced, probably because information outside the window is not available for the prediction of the secondary structure. When the size of the window was increased, the performance reached a maximum at around 20 groups (6 on either size of the center residue).

Table 1

Dependence of testing success rate on window size

Window size	$Q_3(\%)$	C_H	C_E	C_C
2	51	0.49	0.50	0.48
4	55	0.50	0.52	0.54
6	54	0.51	0.53	0.52
8	56	0.52	0.52	0.54
10	56	0.54	0.53	0.53
12	58	0.54	0.55	0.57
14	60	0.57	0.58	0.57
16	64	0.61	0.66	0.59
18	65	0.61	0.65	0.65
20	73	0.69	0.68	0.68
22	69	0.67	0.65	0.68
44	50	0.51	0.46	0.49
54	48	0.41	0.55	0.45
64	45	0.49	0.38	0.48
74	39	0.44	0.35	0.40

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

We used 12 RBF-neurons in hidden layer, window size 20, and different number of epochs. Results are in Table 2

Table 2.

Design of neural networks for different number of epoch

	Number of training epochs	Test	
		Q_3% (training set)	Q_3% (testing set)
1	100	45.5	43.4
2	200	50.1	48.2
3	300	53.2	49.9
4	400	59.6	55.8
5	500	60.1	57.2
6	600	63.0	60.8
7	700	64.9	61.6
8	800	65.6	61.2
9	900	67.1	63.1
10	1000	68.2	64.1
11	1100	68.4	64.6
12	1200	72.1	70.0
13	1300	74.5	72.2
14	1400	75.8	73.2
15	1500	75.7	72.1

4. Conclusion

The paper shows the results of research carried out by the authors of the combined classification algorithm based on group RBF-networks for solving the problem of classification of mass spectra. Analysis of the problem solutions demonstrates the effectiveness of this algorithm that uses parallel-distributed organization of calculations. Feasibility of using it explains their high flexibility, the ability to search for parallel, resistant to noise, associative memory, self-organizing, structural flexibility and high adaptive capacity.

1. Nguyen M.N., Rajapakse J.C. Multi-Class Support Vector Machines for Protein Secondary Structure Prediction, Genome Informatics 14, 218-227, 2003. 2. Clote, P. and Backofen, R., Computational Molecular Biology, Wiley and Sons, Ltd., Chichester, 2000. 3. Mount, D.W., Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, 2001. 4. Garnier, J., Osguthorpe, D.J., and Robson, B., Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins, Journal of Molecular Biology, 120:97-120, 1978. 5. Garnier, J., Gibrat, J.F., and Robson, B., GOR method for predicting protein secondary structure from amino acid sequence, Methods Enzymol, 266:541-553, 1996. 6. Gibrat, J.F., Garnier, J., and Robson, B., Further developments of protein secondary structureprediction using information theory, Journal of Molecular Biology, 198:425-443, 1987. 7. Jones, D.T., Protein secondary structure prediction based on position-specific scoring matrices, Journal of Molecular Biology, 292:195 202, 1999. 8. Rost, B. and Sander, C., Prediction of protein secondary structure at better than 70% accuracy, Journal of Molecular Biology, 232:584-599, 1993. 9. Salamov, A.A. and Solovyev, V.V., Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments, Journal of Molecular Biology, 247:11-15, 1995. 10. Salamov, A.A. and Solovyev, V.V., Protein secondary structure prediction using local alignments, Journal of Molecular Biology, 268:31-36, 1997. 11. Hua, S. and Sun, Z., A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach, Journal of Molecular Biology, 308:397-407, 2001. 12. Lipo Wang and Xiuju Fu Data Mining With Computational Intelligence Berlin: Springer-Verlag, 2005, pp. 276, (ISBN 3-540-24 522-7). 13. Sudipta Saha Protein Secondary Structure Prediction by Fuzzy Min-Max Neural Network with

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Compensatory Neuron / Thesis submitted in partial fulfillment of the requirements for the degree of Master of Technology In Computer Science & Engineering/ Department of Computer Science & Engineering Indian Institute of Technology, Kharagpur-721302, West Bengal, India May, 2008, 126 p. 14. Skolnick, J., A. Kolinski, and A. R. Ortiz. 1997. MONSSTER: a method for folding globular proteins with a small number of distance restraints. J. Mol. Biol. 265:217–241. 15. Taylor, W. R. & Thornton. J. M. (1984). Recognition of super-secondary structure in proteins. J . ..Yol. Bioi. 173. 487-514. 16. O. Weiss and H. Herzel. Measuring Correlations in Protein Sequences. Z. Phys. Chem., 204, 183-197 (1998). 17. Jagla, B. and Schuchhardt, J. 2000. Adaptive encoding neural networks for the recognition of human signal peptide cleavage sites. Bioinformatics 16: 245–250. 18. Rost, B. and Sander, C., Prediction of protein secondary structure at better than 70% accuracy, Journal of Molecular Biology, 232:584-599, 1993. 9. Литвиненко В.І., Фефелов А.О., Дідик О.О. Методологія синтезу колективу радіально–базисних мереж для розв'язування задач класифікації за допомогою алгоритму клонального відбору // Наукові праці ЧДУ ім. Петра Могили. Науково–методичний журнал. Серія "Комп'ютерні науки", – ЧДУ ім. Петра Могили. – 2009. – Вип. 93. – Том.106. – С.111–123. 20. Литвиненко В.И. Искусственные иммунные системы как средство индуктивного построения оптимальных моделей сложных объектов// Проблемы управления и інформатики. – 2008. – №3. – С. 30–42. 21. Chris H.Q.Ding and Inna Dubchak Multi-class protein fold recognition using support vector machines and neural networks / Bioinformatics, Vol.17 no. 4, 2004 p. 349-358. 22. Ryan Rifkin and Aldebaro Klautau. Parallel networks that learn to pronounce english text. Journal of Machine Learning Research, pages 101–141, 2004. 23. Mohamed Aly. Survey on Multi-Class Classification Methods. Technical Report, Caltech, USA, 2005. 24.N. Qian T. J. Sejnowski Predicting the Secondary Structure of Globular Proteins Using Neural Network Models/ J. Mol. Biol. (1988) 202, p. 865-884. 25.Matthews, B.W., Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta 1975, 405, 442–451. 26. Fabio Gonzalez. A Study of Artificial Immune Systems Applied to Anomaly Detection [Ph.D. thesis]. USA The University of Memphis; 2003. -184 p. 27. Тютерев В.В. Метод эволюционного накопления признаков для автоматического построения нейронных сетей // Вычислительные методы и программирование.- 2001.- Т.2.- С.88-108.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua