International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume- 3, Issue- 6, November 2015

An Approach for Protein Secondary Structure Prediction Using Neural Network

Md. Nazrul Islam Mondal, Md. Al Mamun, Md. Zahidur Rahman

Abstract— **Prediction of tertiary structure of protein and the function of a given protein as efficiently, it is necessary to know the secondary structure of protein, however it is a critical need in biological science. The final destination of this research is to measure the performance of predicting secondary structure of protein by using Neural Network (NN) that inputs are primary sequences of protein. There are two phase in Neural Network, one is training phase that are used to learn the network, to recognize the relation between primary structure and their corresponding secondary structure on a sample set of 75 proteins (16671 residues) of that are known as secondary structure. In testing phase where used on 5 proteins (381 residues) primary sequences that try to predict corresponding secondary structure. In our approach using Neural Network approach, we use one hidden layer with no hidden unit or 25,50,75,100,125 hidden units and we also use different window size (1,2,3,5,7,9…21) to find maximum output. Maximum predicative accuracy of Neural Network is 71.89% at window size 17 and 75 hidden unit of hidden layer for three states helix (H), strand (E) and coil(C). We say that presented approach in this paper is simple with better time complexity in comparison to Jaewon's work[4].**

Keywords— ***Protein, Secondary Structure of Protein, Sliding Window, Back propagation.***

I. Introduction

It is important to predicting the structure of proteins from their primary sequence in the field of biochemistry. The key to the wide variety of functions shown by individual proteins is in their three dimensional structure adopted by this sequence. In order to understand protein function at the molecular level, it is important to study the structure adopted by a particular sequence. This is one of the greatest challenges in Bioinformatics.

There are 4 types of structures; Primary structure, Secondary structure, Tertiary structure and Quaternary structure [1]. Prediction of Secondary structure is an important intermediate step in this process because 3D structure can be determined from the local folds that are found in secondary structures. There are different databases that record available protein sequences and their tertiary structures. However, sequence structure gap is rapidly increasing. There are about 50 million protein sequences are discovered already but no anybody predict how to their function properly. This paper examines the prediction of secondary structure of proteins efficiently from their sequences by neural network.

Secondary structure of a protein is the folding or coiling of its polypeptide chains. Dictionary of Secondary Structure Prediction (DSSP) has defined 8 diferente categories, H (α-helix), G (3_{10}-helix), I(π-helix), E(β-strand), B(isolated-β bridge), T (turn), S (bend), and – (other). The reduction scheme that converts this eight state assignment to three states by assigning the helix state (H), strand state (E), and the rest (I, T, S and -) to a coil state (C). This is the simplest format used in this research works.[4]

II. PROPOSED METHOD

In the proposed method, predicting secondary structure of a protein is identified from the input sequence of amino acid using the method based on neural network. The method is modified by varying the different hidden unit in one hidden layer and different window size.

The proposed approach is briefly described step by step below:

A. ***Sliding Window:*** At first the secondary structure that have to be detected from the amino acids. In that case firstly the sequence of amino acid can be divided into as the law of sliding window. In this research, we used different number of sliding window like as 1, 3,5,7,9…21. Sliding window used here to incorporate the influence of the neighbors into the prediction.[3]
Consider, a secondary structure (X,E) and the window of length 5 with the special position in the middle (bold letters).

First position of the window is:

Primary Sequence, X= A R **N** **S** T V V S T A

Secondary Sequence, E= H H **H** **H** C C C E E E

B. ***Encoding Normalization:*** The data is presented in letters and the purpose of preprocessing is to convert those letters into real numbers. To achieve this, orthogonal coding, a similar coding scheme

Manuscript received October 08,2015

Md. Nazrul Islam Mondal, Dept. of CSE, RUET, Rajshahi, Bangladesh, Mobile No. +8801912744327 (e-mail: mondal@ruet.ac.bd).
Md. Al Mamun, Dept. of CSE, RUET, Rajshahi, Bangladesh, Mobile No. +8801962143416 (e-mail: cse_mamun@yahoo.com).
Md. Zahidur Rahman, Dept. of CSE, RUET, Rajshahi, Bangladesh, Phone No. +880721750838, (e-mail: zahidcse09@gmail.com).

adopted by Holley and Karplus (1989) is used. There are 20 amino acids so that we must be assign 20 different random values between in range 0-1.Its used here because of all the 20 amino acids were translated into numeric digits.

C. ***Define output:*** It's a supervised learning method so that we must be defining the output for Helix (H), Sheet (E) and Coil(C) [2] as illustrated in Table I.

Table I: Structure of output pattern.

$y_{d,1}(p)$	$y_{d,1}(p)$	$y_{d,1}(p)$	*Structure*
1	0	0	Helix(H)
0	1	0	Sheet(E)
0	0	1	Coil(C)

D. ***Back Propagation algorithm:*** We used here Back Propagation algorithm of neural network. Finally apply the Back propagation algorithm for finding the desired output and the input neuron depends on the size of the sliding window. This is three layer architecture as illustrated in Fig.1; input layer, hidden layer and output layer. The input layer consists of neurons that receive information to the real world, hidden layer accepts the information from input layer and communicate with other hidden layer, the output layer receive the output from last hidden layer for final processing the data and full process back propagated until error rate is minimum.

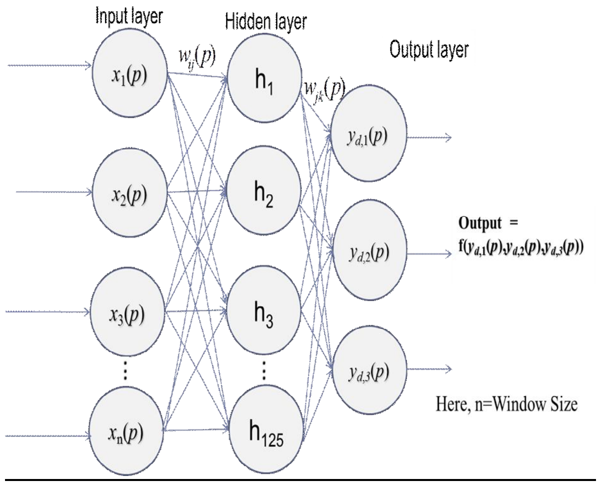

Fig.1. Back propagation algorithm

The algorithm is described using some steps:

Step 1: Initialisation

Assign weights and threshold levels of the network to random numbers. Assume that, Error rate is .001.

Step 2: Activation

Activate the back-propagation neural network by applying inputs $x_1(p)$, $x_2(p)$,…, $x_n(p)$ and 3 desired outputs $y_{d,1}(p)$, $y_{d,2}(p)$,$y_{d,3}(p)$.

(a). Calculate the actual outputs of the neurons in the hidden layer:

$$y_j(p) = sigmoid\left[\sum_{j=1}^{m} x_i(p)\cdot w_{ij}(p)\right]$$

Here, n = Size of the Window size.
j = Neuron in the hidden layer.
Sigmoid = Sigmoid activation function.

(*b*) Calculate the actual outputs of the neurons in the output layer:

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p)\cdot w_{jk}(p)\right]$$

Where, *m* is the number of inputs of neuron *k* in the output layer.

Step 3: Weight training output layer

Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

(a) Calculate the error :

$$e_k(p) = y_{d,k}(p) - y_k(p)$$

And then the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p)\cdot[1 - y_k(p)]\cdot e_k(p)$$

Then the weight corrections:

$$\Delta w_{jk}(p) = \alpha \cdot y_j(p)\cdot \delta_k(p)$$

Here, α = Learning rate.
Then the new weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

Step 4: Weight training hidden layer

(b) Calculate the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p)\cdot[1 - y_j(p)]\cdot\sum_{k=1}^{l}\delta_k(p)\, w_{jk}(p)$$

(c) Calculate the weight corrections:

$$\Delta w_{ij}(p) = \alpha \cdot x_i(p)\cdot \delta_j(p)$$

Update the weights at the hidden neurons:

Step 5: Iteration

Increase iteration *p* by one, go back to *Step 2* and repeat the process until the selected error criterion is satisfied.[2]

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

III. DATA AND TOOLS

Rost and Sander (1983) database is used here and we worked here as totally 80 protein sequences that are divided into training stage and testing stage.

In training stage, 75 protein sequences (both primary & secondary, 16671 residues) are used on training stage and 5 primary protein sequences (381 residues) are used on testing stage. And when we finally get output of secondary structure of 5 given primary sequence, then compare it with actual output & find percentage of accurate output.

The software used for the experiments is Matlab Version 7.4.0.287 (R2007a). The computer that was used to perform the experiments for model selection is an Intel(R) Core(TM) 2CPU6300@1.86GHz

IV. RESULT AND PERFORMANCE ANALYSIS

Final objective of the research is to find the performance of neural network in protein secondary structure prediction. We worked for totally 80 protein sequences that are divided into training stage and testing stage.

In training stage, 75 protein sequences (both primary & secondary) are used on learning stage and 5 primary protein sequences (381 amino acid samples) are used on testing stage. And when we finally get output secondary structure of 5 given primary sequence, then compare it with actual output & find percentage of accurate output.

We used one hidden layer where no hidden unit or 25, 50,75,100,125 hidden units. The networks had three output node and we classify that if the output had (1, 0, 0) that the output is helix (H), if the output has (0, 1, 0) that the output strand (E), if the output has (0, 0, 1) that the output is coil(C).

A. Sliding Unit Size Vs accuracy

Performance depends on the sliding window size, we worked here as different window size and find accuracy that is shown in Table II.

Table II: Observing performance across the different window size.

Window size	*Correct Predicted residues among 381 residues*	*Accuracy(Q_3)*
1	210	55.1562%
3	216	56.7263%
5	222	58.1621%
7	229	60.0954%
9	242	63.5241%
11	248	65.0912%
13	260	68.1211%
15	270	70.9128%
17	272	71.3411%
19	267	70.1271%
21	260	68.1733%

From Table II, the overall prediction accuracy of our recognition rate depends on window size. While window size is 17 then we get maximum overall accuracy of about 71.3411%, now compare the above result with Jaewon Yang Work [4] in Fig.2.

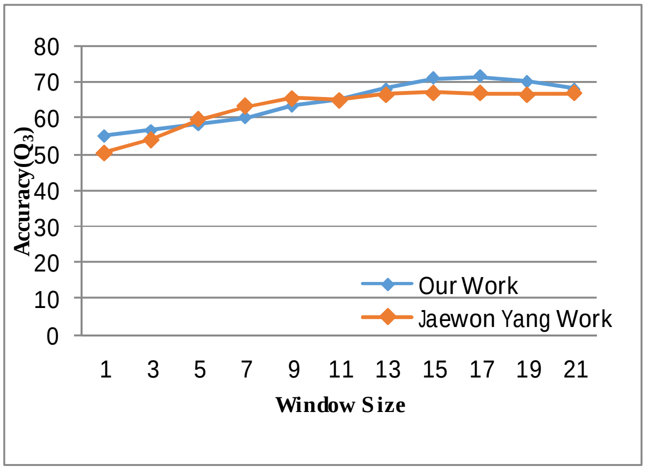

Fig.2. Graphical representation of Window Size vs. Overall accuracy (Q_3).

From above Fig.2, we have seen that we found the best result when window size is 17.It means that when we consider large number of neighbors then accuracy is high. For larger window sizes, the performance deteriorated, probably because of the effects of extra weights that could not contain any information about the secondary structure of the centre. Thus, irrelevant weights can interfere with the performance of the network. Time complicity of our work is better, however this is not included for page limitation.

B. Hidden Unit (W = 17) Vs Accuracy

Performance depends on the size of the hidden units; we worked here as different hidden unit in one hidden layer and get overall accuracy. Here, we consider window size as 17 in this step.

Table III Observing performance across the different hidden unit.

Hidden unit(W=17)	*Correct Predicted residues among 381 residues*	*Accuracy(Q_3)*
0	246	64.6453%
25	257	67.5363%
50	266	69.8742%
75	274	71.8931%
100	267	70.0911%
125	259	68.0563%

From Table III, we have concluded that our recognition rate depends on different hidden units. It is also shown that high hidden unit is not always good because it can cause high variance problem. Let us compare our work with result of Jaewon Yang Work [4] in Fig.3. In this figure, it is shown that the accuracy of our work is slightly lower than Jaewon Yang Work [4]. However, the presented approach is simple with better time complexity which is omitted for the page limitation.

.

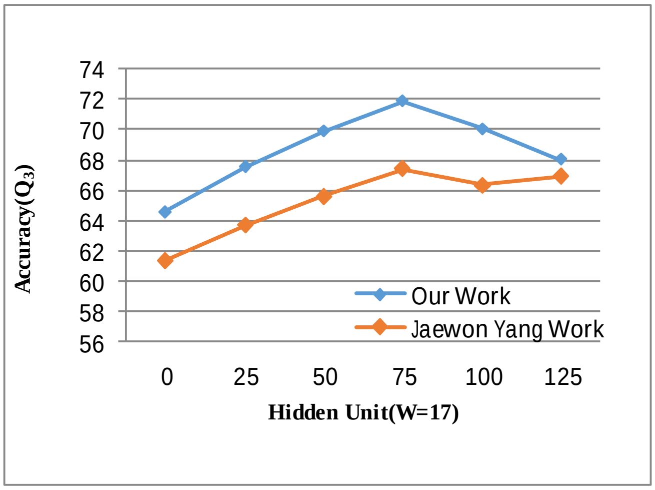

Fig.3. Graphical representation for Hidden unit of hidden layer vs. Accuracy.

From above Fig.3, we have also seen that we found the best result at about 71.89%, when number of hidden unit in hidden layer is 75. In our work, we faced "over fitting" problem in Neural Networks algorithm. The problem occurred due to huge number of weight value needed to be deducted or updated. Thus, we could not get maximum output when the increasing of hidden unit of hidden layer. For the reader's benefit, we say that the total output of helix state (H), strand state (E) and coil state (C) for the research works are given as follows:

Table IV Predictive output for training & testing set.

Total training residues =16671		
Total testing residues = 381		
	Train Set (%)	**Test Set (%)**
Helix state (H)	36.09%	39.34%
Strand state (E)	19.61%	18.67%
Coil (C)	44.3%	41.99%

V. CONCLUSION AND FURTHER WORK

The main goal of this research is to propose a simple approach to measure accuracy for protein secondary structure prediction based on neural network and compare with Jaewon Yang Work [4]. Our result says that our accuracy is slightly lower than that of Jaewon Yang Work [4], however proposed approach is simple with better time complexity. In future we have a plan to include details about time complexity. Also we have a plan to do this work using Support Vector Machine.

REFERENCES

[1] Whitford, D. A book of "Proteins Structure and Function". John Wiley & Sons Ltd, England, 2005.

[2] Hornik, K., Stinchcombe, M. &White, H. Multilayer Feedforward Networks Are Universal Approximators. Neural Networks, 2: pp. 359-366, 1989.

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

[4] Jaewon Yang, "Protein Secondary Structure Prediction based on Neural Network Models and Vector Machines" Stanford University, Vol-31.

[5] Riedmiller,M. Prop Description and Implementation Details. Technical Report, University of Karlsruhe, Germany, 1994.

[6] Rost, B. & Sander, C. Improved Prediction of Protein Secondary Structure by Use of Sequence Profiles and Neural Networks. Proc. Natl. Acad. Sci. 90: pp. 7558-7562, 1993.

[7] Qian, N. &Sejnowski, T.J. Predicting the Secondary Structure of Globular Proteins Using Neural Network Models.J. Mol. Biol. 202: pp. 865-884, 1988.

M. N. I Mondal received BE degree from the Department of Electrical & Electronic Engineering, RUET, Bangladesh in 2000, ME degree from the Department of Information & Communication Technologies, Asian Institute of Technology, Thailand in 2008 and Ph.D. degree from the Department of Information Engineering, Hiroshima University, Japan in 2012. He joined as a lecturer to the Department of Computer Science & Engineering, RUET in 2001. In 2004 and 2013, he became Assistant Professor and Associate Professor in the same Department respectively. He also joined as a Professor in 2015 at the same Department. From March, 2012 to September, 2012 and from May, 2013 to April, 2014, respectively he was Visiting Research Scholar and Specially Appointed Assistant Professor in the Department of Information Engineering, Hiroshima University, Japan. He has been serving as a CISCO instructor since 2006. He has published his contributions extensively in journals, conference proceedings. He served as an Organizing Chair, a PC member, reviewer and sub-reviewer for many Journals and Conferences such as Journal of Foundation of Computer Science, Journal of Communication and Computer, International Journal of Networking and Computing, IEICE, APDCM, PDP, ICNC, CANDAR, IJPEDS, ICPP and so on. He is a Fellow of Institution of Engineers, Bangladesh and IEEE Member. His research interest includes FPGA-based Reconfigurable Computing, Parallel Computing, Algorithms and Architectures, Image Processing, DSP and Computer Networks and Data Communications.

Dr. Md. Al Mamun have 8 long years of teaching experience in the various fields of computer science and engineering. Graduated from Rajshahi University of Engineering & Technology, Bangladesh Mr. Mamun got his first teaching opportunity in the same university to take courses like Computer Programming, Database Management System, Computer Architecture, Computer Graphics, Object Oriented Programming, Digital Image Processing and many more. In 2009, he got teaching assistantship in the University of New South Wales, Australia. This was the opportunity, which he got when he was doing his PHD in the same university. He was responsible for lecturing various computer science courses like Object oriented programming (Java), Computer Graphics (Game Simulation-Alice) etc in UNSW@ADFA, Australia . Now he is serving as associate professor

in RUET. He is a Fellow of Institution of Engineers, Bangladesh and IEEE Member. His research interest includes Satellite Image Processing, Data Mining, Computer Vision.

Md. Zahidur Rahman received BE degree from the Department of Computer Science & Engineering, RUET, Bangladesh in 2013. He is now working in famous IT industry in Bangladesh. He was an excellent organizer of many Int'l Conference in Bangladesh. He is a member of Institution of Engineers, Bangladesh. His research interest includes FPGA-based Reconfigurable Computing, Parallel Computing, Algorithms and Architectures, Image Processing, DSP and Computer Networks and Data Communications.