Effect of Neural Network Parameters on RNA Secondary Structure Classification

Ruchi Mann
Dept of computer science & Engineering
PEC University of Technology
Chandigarh

Shailendra Singh
Dept of computer science & Engineering
PEC University of Technology
Chandigarh

ABSTRACT

Helix, Hairpin, Bulge, external loop, internal loop, multi-branch loop are the elements of RNA secondary structure. We have designed a neural network to classify the RNA sequence in to three categories i.e Hairpin, helix, neither of two. This can be extended to classify into all secondary structure elements. If all the elements are predicted then we can determine the entire structure of a RNA family. The parameters of neural network affect the performance of the network. But there are no rules to define the value of these parameters of network. For a given problem the optimal value of parameters can be obtained by performing the experiments on their values. This paper shows the effect on the performance of classification by varying the number of hidden layers, number of neurons and activation functions.

Keywords:

Classification, RNA secondary structure, neural networks, activation function, number of hidden layers

1. INTRODUCTION

RNA plays important role in cellular process. One molecule of RNA consists of long chain of small sub units called ribonucleotides. Each ribonucleotide can have one of the bases Adenine, Guanine, Cytosine and Uracil. These bases are represented by alphabets A, G, C, and U respectively. The bases adenine (A) and guanine (G) belong to the group of purines and form a double ring, whereas cytosine (C) and uracil (U) are pyrimidine derivatives. Under normal physiological conditions, a ribonucleotide chain can bend back upon itself, and the bases can hydrogen-bond with one another, such that the molecule forms a coiled and looped structure. This leads to formation of various secondary structure elements like helix, bulge, hairpin etc. There are different methods for prediction of RNA secondary structure. Nussinov Algorithm, zuker algorithm, covariation analysis [4], stochastic context free grammars [5], Hopfield neural networks [2] are few common techniques used for prediction of secondary structure. The secondary structure of most functional RNA molecules is strongly conserved in evolution. Prediction of these conserved structures is therefore of particular interest when studying noncoding RNAs. Moreover, structure predictions on the basis of several sequences produce much more accurate results than energy directed folding of single sequences. The RNAalifold[3] program predicts the consensus structure for a set of aligned sequences taking into account both thermodynamic stability and sequence covariation. KNetFold[17] is software for predicting the consensus RNA secondary structure for a given alignment of nucleotide sequences.

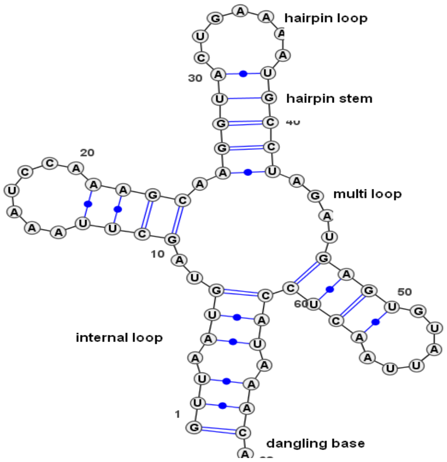

Figure 1

2. METHODOLOGY

First of all the sequences were aligned using the covariation model. The source of aligned sequences was Rfam 10.0 – a database of non coding RNA. The representative of the alignment was created by using the rule that the nucleotide which has the highest frequency in the column was chosen as representative of that column. In case gap has the highest frequency then it is chosen as representative only if its frequency is greater then 70 percent else next highest nucleotide is chosen as the representative. During the training phase the secondary structure of family is known. So using the representative and the secondary structure we extract the hairpin, non hairpin, helix and non helix portions. Since sequences are strings of alphabets A,U,C,G and vary in length so they can't be used as such for inputs of neural network. We extract the feature vector from sequences. Feature vector contains eight parameters. They are mutual stem length , stem mutual information, stem fraction of complementary base pairs(FCBP), stem energy , loop length , loop mutual information, loop fraction of complementary base pairs(FCBP), loop energy. Then the feature vectors for different training examples were used to train the network. During testing phase we extract the hairpin and helix region using heuristics. Then feature vector is formed from extracted sub sequences. This feature vector is used to get the output.

3. NEURAL NETWORK

The concept of an Artificial Neural Network (ANN) was inspired by the information processing capabilities of the biological neural networks. ANN has been developed as generalization of mathematical model of human cognition or neural biology. ANN consists of many processing units called neurons. Each neuron is connected with other neuron using the weights and has an activation function which is function of product of inputs and weights. The training algorithm decides how the weight changes with learning. There are different types of activation functions like linear activation function, logistic activation function, radial base functions etc.The type of activation function used in application depends on type of neural network used.

3.1. BackPropagation Neural Network

In feed forward neural network the neurons are organized in layers. There is one input layer, one output layer and one or more hidden layers. The signals or output of one layer travels to other in forward direction. The backpropagation training algorithm is supervised training method. It uses a gradient descent method to minimize the total squared error of the output produced by the neural network. The training involves three steps: 1) computation of the output 2) computation of the error and backpropagation 3) adjusting the weights. There are several variants of backpropagation like backpropagation with momentum, resilient backpropagation etc which increases the rate of learning. Weight changes are made according to following equation.

$$w(t+1) = w(t) - € \frac{\partial E}{\partial w(t)}$$

Where $\frac{\partial E}{\partial w(t)}$ is change in error.

$$\frac{\partial E}{\partial w(t)} = \frac{\partial E}{\partial s} * \frac{\partial s}{\partial net} * \frac{\partial net}{\partial w}$$

where W is the weight from neuron j to neuron i,s is the output, and net is the weighted sum of the inputs of neuron i.

Experiment on Parameters of Neural Network

In this work we have used backpropagation neural network. Feature vector having eight inputs is given as input and there are two neurons in the output layer first for helix and second for hairpin. Figure 2 shows the design of network used in this work. The learning algorithm we have used is iRPROP which is improved resilent back propagation learning algorithm. The number of hidden layers, number of neurons in hidden layers and activation functions are varied to get the values for which the network performance is best.

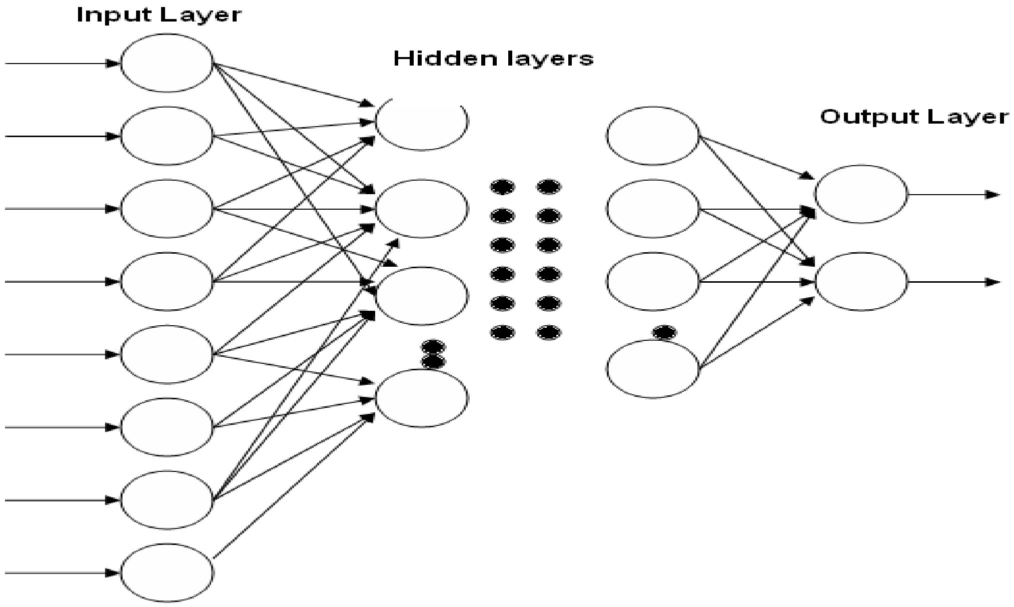

Figure 2

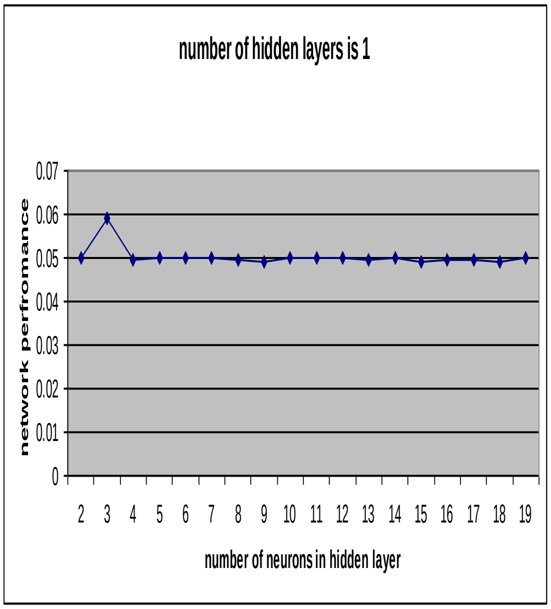

Figure 3

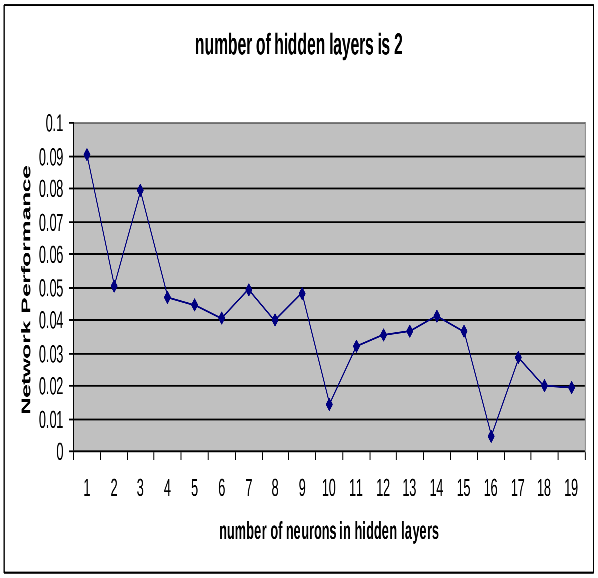

Figure 4

Figure 3 and 4 shows the variation of network performance with number of neurons in hidden layers. Network performance is measured in terms of the error. Error is calculated by subtracting the target output from the output obtained using neural network. To see the effect of number of neurons in hidden layers we fixed the number of maximum epoch to be 5000. In order to prevent the early stopping of the training of

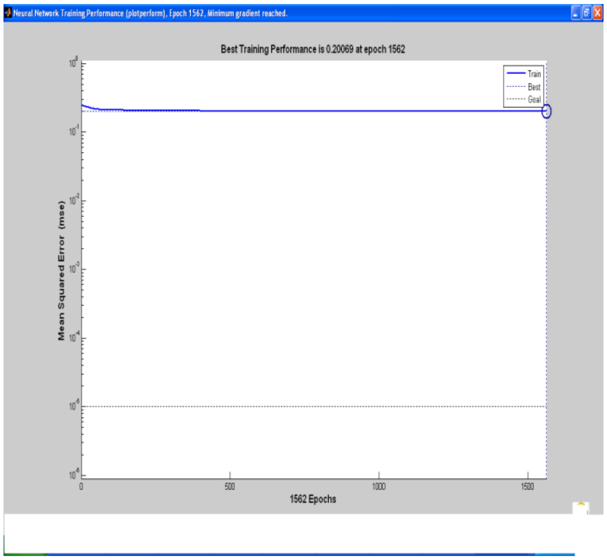
Neural Network Training Performance (plotperform), Epoch 1562, Minimum gradient reached.
Best Training Performance is 0.20069 at epoch 1562
Train
Best
Goal
Mean Squared Error (mse)
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
500
1000
1500
1562 Epochs

Figure5

neural network we gave lower values of target network performance and gradient. Goal to be achieved by network was given a value of 1.0e-05 and gradient also had a value of 1.0e-05. Learning rate was set to 0.05.Experiments were done for one and two hidden layers in network. The numbers of neurons in hidden layers were varied from 2 to 19. Activation function was varied between hyperbolic tangent sigmoid and log sigmoid. Figure 5 and Figure 6 shows the network performance using hyperbolic tangent sigmoid and log sigmoid respectively. Matlab is used to create the neural network and perform all the experiments.

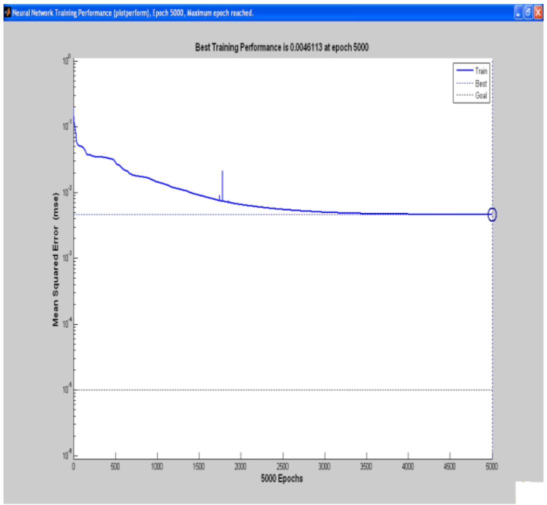

Figure 6

4. RESULTS and DISCUSSION

In network with one hidden layer best performance is achieved when number of neurons is hidden layer is either 9 or 15. The best network performance is 0.0491. When number of hidden layers in network is increased to two the best performance is achieved with 16 neurons in both the layers. The best network performance is 0.00461. This is much better then the best performance of network with one hidden layer. Figure 4 shows the performance of network having two hidden layers and 16 neurons in each layer with log sigmoid as activation function. The best performance in this case is 0.20069 at epoch 1562. Training was stopped earlier because gradient reached its maximum value. Figure 5 shows the performance of network having two hidden layers and 16 neurons in each layer with hyperbolic tangent function as activation function at each layer. Here the best performance achieved is 0.00461 which is much better than performance achieved using log sigmoid as activation function. More experiments were performed by changing the activation function of output layer to purely linear (Purelin in matlab). The performance achieved using linear activation function in output layer was better than that obtained using log sigmoid activation function in all layers. But the performance was poor than that obtained using Hyperbolic Tangent function. In most of the above experiments training took place until 5000 epochs but in few cases early stopping was also there due to the fact that gradient reached its maximum value

So in the final design of network we used two hidden layers with 16 neurons in each layer and having hyperbolic tangent function as activation function in every neuron of all layers. With these experiments we can conclude that number of neurons in hidden layer and number of hidden layers affects the performance of the RNA secondary structure classification.

5. REFERENCES

[1] Monther Aldwairi, Rehab Duwairi, Wafa'a Alqarqaz "A Classification System for Predicting RNA Hairpin Loops", 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing

[2] Steeg E., Neural networks adaptive optimization and RNA secondary structure prediction, American Association for Artificial Intelligence, pp 121-160, 1993.

[3] The Vienna RNA Servers: RNAalifold server, at http://rna.tbi.univie.ac.at/cgibin/RNAalifold_beta.cgi.

[4] Siebert S, Backofen R, and MARNA: Multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons, BMC Bioinformatics, 21(16): 3352-3359, 2005.

[5] Knudsen B. Hein H., Pfold: RNA secondary structure prediction using stochastic context-free grammars, Nucleic Acids Research, 13(13): 3423-3428, 2003.

[6] Gardner P., Giegerich R., A Comprehensive Comparison Of Comparative RNA Structure Prediction approaches, BMC Bioinformatics, 5(140), 2004.

[7] D.H. Turner, Nearest neighbor parameters for RNA based on melting studies of synthetically constructed oligoribonucleotides, at http://www.bioinfo.rpi.edu/zukerm/rna/energy/node2.html.

[8] Christian Igel and Michael H¨usken "Improving the Rprop Learning Algorithm" , Proceedings of the Second International Symposium on Neural Computation, NC'2000, pp. 115–121, ICSC Academic Press, 2000

[9] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks, pages 586–591. IEEE Press, 1993.

[10] R. A. Jacobs. Increased rates of convergence through learning rate adaptation. Neural Networks, 1(4):295–307, 1988

[11] Griffiths-Jones S, Bateman A, Marshall M, Khanna A and Eddy SR" Rfam: an RNA family database." Nucleic acids research 2003;31;1;439-41

[12] S. Lindgreen, P. P. Gardner, and A. Krogh, "Measuring covariation in RNA alignments: physical realism improves information measures," Bioinformatics, vol. 22, pp. 2988-95, Dec 15 2006.

[13] D. H. Turner and N. Sugimoto, "RNA structure prediction," Annu Rev Biophys Biophys Chem, vol. 17, pp. 167-92, 1988.

[14] B. Knudsen and J. Hein, "Pfold: RNA secondary structure prediction using stochastic context-free grammars," Nucleic Acids Res, vol. 31, pp. 3423-8, Jul 1 2003.

[15] I. L. Hofacker, M. Fekete, and P. F. Stadler, "Secondary structure prediction for aligned RNA sequences," J Mol Biol, vol. 319, pp. 1059- 66, Jun 21 2002

[16] Nebel M., Identifying good predictions of RNA secondary structure, pacific symposium on biocomputing, 9: 423-434, 2004

[17 Bindewald E., Shapiro B., RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor
classifiers, RNA, 12(3): 342 – 352, 2006.]

[18] Tahi F., Gouy M , Regnier M , Automatic RNA secondary structure prediction with a comparative approach , Computers and Chemistry , 26:521-530, 2002.

[19] Gardner P., Giegerich R., A Comprehensive Comparison Of Comparative RNA Structure Prediction approaches, BMC
Bioinformatics, 5(140), 2004.

[20] Hofacker I., Fekete M., Stadler PF., Secondary structure prediction for aligned RNA sequences, Journal of Molecular Biology,
319(5):1059-1066, 2002.

[21] Larranaga P., Calvo B., Santana R., Bielza C., Galdiano J., Inza I., Lozano J., Arman R., Santafe G., Perez A. , Robles V., Machine Learning In Bioinformatics, Oxford Journals, 7(1) : 86-112 , 2005.

[22] Ruan J., Stormo GD., Zhang W., An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots, 20(1):58-66, 2004.

[23] Rivas E, Eddy SR (2000) Secondary structure alone is generally not statistically significant for the detection of noncoding rnas. Bioinformatics16: 583–605.

[24] Werbos, P.J. (1993) "Backropagation through time: What it does and how to do it" *Proc. ICNN*,San Francisco, CA.