

# Transmembrane Alpha Helix Prediction Using Neural Networks

Saumitra Singh

Defence Accounts Department, Prayagraj, Uttar Pradesh, India

## ABSTRACT

This paper presents a simple artificial neural network which classifies proteins into two classes from their sequences alone: the alpha helix transmembrane protein class and the non-alpha helix transmembrane protein class. The network described here has a simple feed-forward topology and a limited number of neurons which makes it very fast.

**Keywords :** Alfa Helix Transmembrane, Artificial Neural Network, Feed-forward Topology, Neural Network.

## I. INTRODUCTION

The artificial neural network which I present is not concerned with the psychological implication of the network, and I will at most occasionally refer to biological neural models. I consider neural networks as an alternative computational scheme rather than anything else. The artificial neural network was inspired by the biological finding relating to the behavior of the brain as a network of units called neurons. The human brain is estimated to have around 10 billion neurons each connected on average to 10,000 other neurons. Each neuron receives signals through synapses that control the effects of the signal on the neuron. These synaptic connections are believed to play a key role in the behavior of the brain. The artificial neural network work on the same thing. The artificial neural network has three different layers known as input layer, hidden layer and the output layer.

The number of neurons in each layer is defined according to the application. In this paper I use 41 neurons in input layer, 2 in hidden layer, and one neuron in output

layer. Each neuron of input layer is connected to the every neuron of hidden layer, and each neuron of hidden layer is connected to the every neuron of output layer. This is known as full connected neural network.

## II. BACK PROPAGATION ALGORITHM

In this paper the back-propagation algorithm is used in the prediction of transmembrane alpha helix. Backpropagation algorithm is a supervised learning technique used for training artificial neural network. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires that the transfer function used by the artificial neurons (or "nodes") be differentiable. The general algorithm of the backpropagation is that in which a number of sequences and their output (which shows the parts where the transmembrane alpha helix exists and also shows where it is not exists) for the training of network. In training of the neural network is simply work to adjust the weights such as the difference between the exact output and the output provided by

neural network is minimized. When the weights are adjusted then the network is ready for the testing. Then we give another input sequence for the prediction. The technique which is used in this paper is

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.

The actual Algorithm which is used is as follows:

1. Initialize the weights in the network (often randomly)
2. repeat foreach example e in the training set do
  - (i)  $O = \text{neural-net-output}(\text{network}, e)$ ; forward pass
  - (ii)  $T = \text{teacher output for } e$
  - (iii) Calculate error  $(T - O)$  at the output units
  - (iv) Compute  $\Delta w_i$  for all weights from hidden layer to output layer ; backward pass
  - (v) Compute  $\Delta w_i$  for all weights from input layer to hidden layer ; backward pass continued
  - (vi) Update the weights in the network
- end
3. until all examples classified correctly or stopping criterion satisfied
4. return(network)

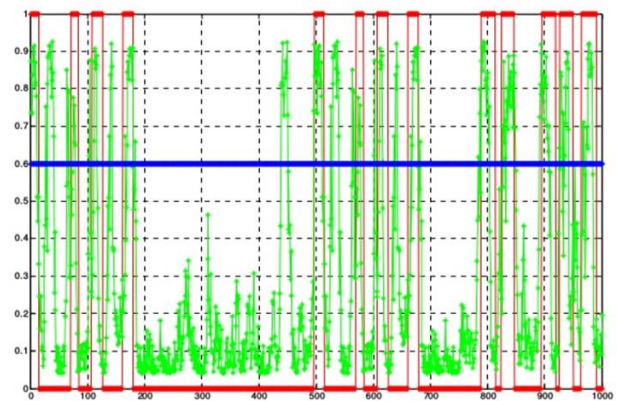
### III. RESULTS

I have used the "Matlab Software" for writing the source code. It provides many different functions which was very useful to accomplish the results. This

software also facilitates to plot the different functions at run time by which I was able to understand the working of these functions. I used the transmembrane alpha helix protein sequences from the protein data bank. PDBTM database is maintained in the Institute of Enzymology by the Protein Structure Research Group. The PDBTM database was created by scanning all PDB entries with TMDET algorithm. In this paper the 41 neurons in input layer i.e. a window of 41 neurons, 2 neurons in the hidden layer, and only one neuron (that predict either a transmembrane alpha helix or a non-transmembrane alpha helix) in output layer have been used.

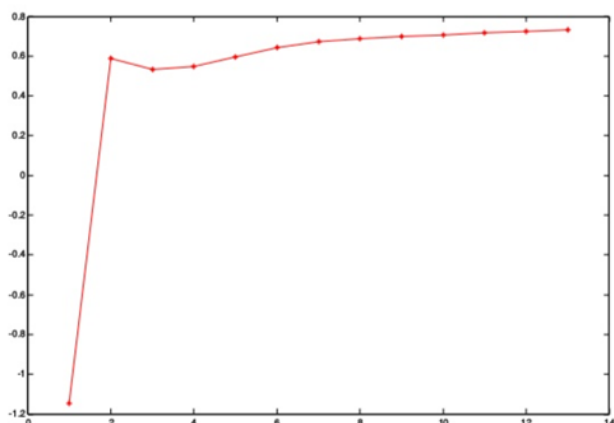
The given figure shows the value of real output, predicted output and threshold.

According to this I say which part is transmembrane alpha helix and which is not.



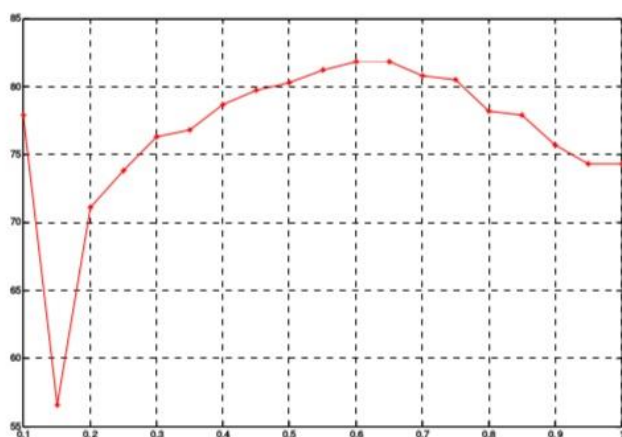
**Figure 1:** Results of the actual outputs, predicted outputs with threshold

The above figures the points which shown above to threshold (which is 0.6) are alpha helix transmembrane and which points shown to the below of threshold are the non alpha helix transmembrane. And the given figure shows the convergence of weight in the neural network. We shows only the weight {1,1}.



**Figure 2:** Results of convergence of the weight

Now we show how the prediction percentage increase or decrease according to the increasing the threshold of predicted output value. The given figure shows this.



**Figure 3:** Results of the percent of prediction at different thresholds

#### IV. CONCLUSION

The prediction of transmembrane segments within proteins is a central problem of computational biology. A number of methods have been developed over the past 30 years. Some of them accomplish high accuracy and are available via the internet.

However, most of these methods are focused on the localized of transmembrane segments in known integral membrane proteins and produce a number of false segment detections when applied to globular water-soluble proteins. The system we show here that

a simple and very fast neural network which can be successfully applied to the prediction of transmembrane alpha helix proteins. The novelty in our neural network topology is the small number of neurons and connection required. It is known that the successful generalization of a prediction by a neural network requires a much larger number of cases that the number of weights adjusted during the training phase. With our architecture, the total number of connection associated with a weight is only 84 (82 to connect 41 input neurons to the hidden layer and 2 from this layer to the unique output). In addition, the simple feed-forward topology of the network and their limited numbers of connections allow proteins to be processed very quickly and could open the way for a new implementation able to handle longer segments of amino acids and, perhaps, complete sequences. In this paper when I increase the number of neurons in the hidden layer, the neural network increase the percent of prediction. In future by using the preprocessing and introducing the moment term we increase the rate of performance.

#### V. REFERENCES

- [1] Yan Liu, "Conditional Graphical Model for Protein Structure Prediction".
- [2] Jean-Luc Popot and Donald M. Engelman, "Helical Membrane Protein Folding, Stability, and Evolution".
- [3] Aramak Afzal, "Application of Neural Networks in Protein Structure Prediction"
- [4] Ben Krose and Patrick Vander Smagt, "An Introduction to Neural Networks", Eighth Edition November 1996.

**Cite this article as :** Saumitra Singh, "Transmembrane Alpha Helix Prediction Using Neural Networks", *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 7 Issue 7, pp. 102-104, January-February 2020. Available at doi : <https://doi.org/10.32628/IJSRSET20715>  
Journal URL : <http://ijsrset.com/IJSRSET20715>