Secondary Structure Prediction using Machine Learning

Kayahan Saritas[1, *]

[1]*Department of Materials Science and Technology, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139*

(Dated: May 13, 2016)

We study protein secondary structure prediction method that uses feed-forward neural network. A random training phase was used to teach the network to recognize the relation between secondary structure and aminoacid sequences on a sample set of 75 proteins taken from Rost and Sander data set[1]. The set is divided into training, validation and test sets where its accuracy and predictive abilities are tested. The method achieves overall 63.4 % maximum predictive accuracy for three states: helix, sheet and coil.

PACS numbers:

I. INTRODUCTION

Accurate prediction of protein secondary structure is a step toward the goal of understanding protein folding. Several efforts have been made to identify the secondary structure of proteins using chemistry of aminoacids, pattern matching and statistical analyses of known proteins. In this work, we describe a secondary structure prediction method using neural networks. Neural networks attempt to simulate the information processing that occurs in the brain and are widely used in a variety of applications, including automated pattern recognition[2], content-addressable memory[3] and certain optimization problems[4]. A large number of simple, highly interncon-nected units operate in parallel. Each unit integrates its input, which might be excitatory or inhibitory, according to some threshold and generates an output, which propagates to other units.

We use feed-forward networks for secondary structure prediction. We implement the approach used by Karplus et. al. [5], on a different and larger set of protein structures. Our neural network is divided into three main layers, which are called input layer, hidden layer and output layer. Protein structures are used in the input layer of all the structures considered. A critical choice must be made about the network topology, where number of layers, size of each layer and the connection patterns must be supplied. In the training phase, random connection weights and biases for each connection is updated to yield the optimized network where difference between desired and observed output is minimized. Meanwhile, in order to prevent overfitting, the model obtained using the training set is also evaluated in the validation set and the solution is truncated when both sets are optimized. This is effectively similar to performing a pattern recognition algorithm where pattern selection rules are defined using the training set and it is applied to the new problems, such as protein structures in the test set.

II. METHODS

II.A Data used in the study

In our work, we used the a sub set of the data set of proteins where secondary structures are classified by Rost and Sander [1]. Our set consists of proteins whose structures span a relatively wide range of domain types, composition and length. Residues that are not classified either sheet or helix are classified as coil. The training set has a composition of 32 % helix, 21 % sheet and 47 % coil.

II.B Network formation and calculation

Network used in tha calculations include an input layer, a hidden layer and an output layer. Input layer is obtained in a way that existing protein structures are sliced through a moving window of a fixed interval, effectively forming a Hankel matrix. In a Hankel matrix, ith column represents the subsequence starting from the ith element of the original sequence. Size of the slices are chosen based on the prior literature works, where it was shown that a window size of 17 yields the largest statistical correlation with respect to identification secondary structure using aminoacid sequence[5]. Each aminoacid is encoded in binary form using a binary array of size 20, meaning one element for each type of aminoacid. For each group of 20 inputs, element corresponding to specific aminoacid is denoted as one. As one window contains 17 aminoacids, binarized form of the same sequence contains 17x20 elements in the array, which are used as input layer. Input data is randomly split into three sections where 60 % is used in training set, 20% is used in the validation set and 20% is used in the test set. Throughout this paper, we use MATLAB neural network toolbox for implementing our work.

Hidden layer and output layer consist of 10 units. In general larger networks with more hidden units perform better in the training set, but their performance declines

in the validation and test set since they lose their ability for generalization. In order to compensate for this, large training sets must be supplied to the neural networks with large number of hidden layers. Therefore, when there is a limited number of variables in the training set, a compromise must be made between training and prediction accuracy when networks are constructed. The output layer is also encoded using a binary form. For each window considered in the input layer, secondary structure of only the central aminoacid is evaluated. Remaining 8 aminoacids on the left and the right of the central aminoacids would be considered as the neighboring aminoacids which remain within correlation length. In the binarized form of the output layer, [1,0,0] stands for coil, [0,1,0] is for sheet and [0,0,1] is for helix. Besides, as for the size of the hidden later in the neural network, when an average size of a protein is considered to be composed of 300 aminoacids, hidden layer with 20 units and output layer of 3 units create nearly 6000 free variables. In our dataset, excluding the first and last 8 aminoacids in each protein, we have nearly 15471 residues which can be used in neural network training. Although it is fundamentally possible to perform training with larger number of hidden layers, increasing the number will yield solutions favoring more the training set. Therefore, we limit ourselves to a maximum of 20 hidden layers in investigating the neural network performance.

Before the optimization of the network, all the weights are randomized in the range between -0.1 and 0.1. In each cycle all of the training proteins that are presented to the neural network are used and optimization is performed using one window at a time, therefore does not consider correlations between different windows. At the end of the cycle, weights are updated. Scaled conjugate gradient algorithm is used to update consequent cycles. Optimization is terminated when any of the following criteria are met: the size of the gradient is below ¡TODO¿ $10^{-}6$, number of iterations is more than 1000, and more than 6 validation checks are performed. Validation checks are performed when any further optimization of the training data does not improve the accuracy in the validation set. Therefore when more than 6 subsequent steps are obtained, where accuracy of the validation set does not improve, then the optimization is terminated and (N-6)th iteration is considered as the optimized neural network.

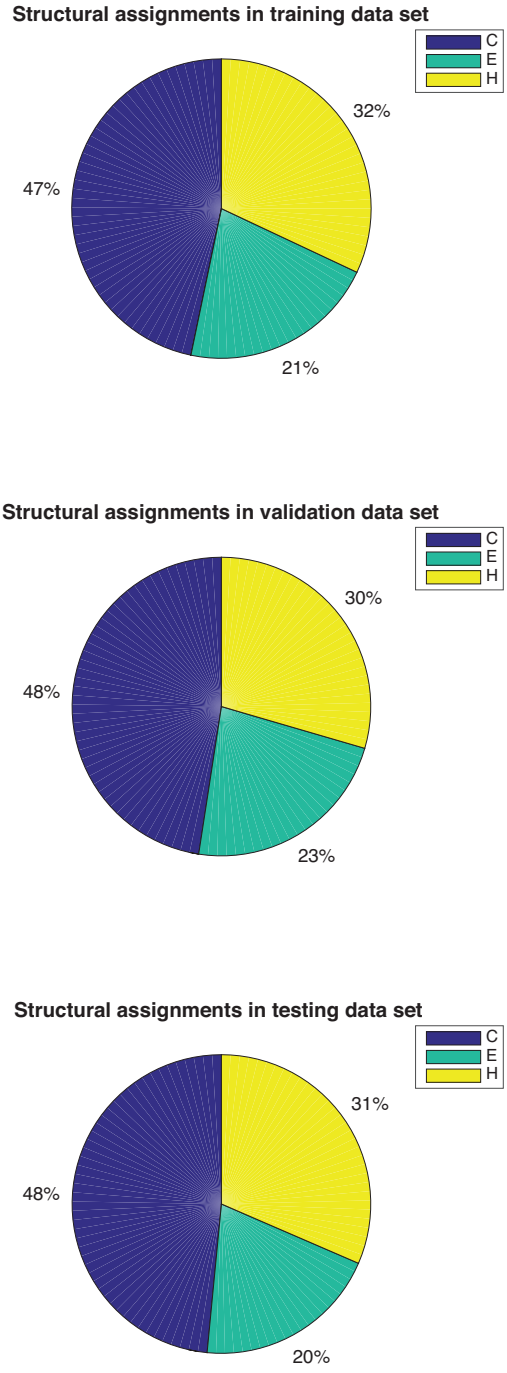

FIG. 1: Composition of training, validation and test sets with respect to secondary structure of each aminoacid. C stands for coil, E stands for sheets and H stands for helix.

III. RESULTS

We initially start with analyzing how data splitting occurs between training, validation and test sets. It is crucial that these three sets should contain similar percentages of coil, sheet and helix structures. Therefore in Figure 1 we show that composition of each set with respect to secondary structure of the aminoacids is almost equal.

Additionally, we perform a few tests in order to determine prediction accuracy with respect to neural network parameters. In Table I, we show how the prediction accuracy changes with respect to slicing windows that we choose to train subset of protein sequences. As pointed out in ref. [5], we reproduce that the window size of 17 is optimal choice which provides the most accurate results

TABLE I: Effect of input window size, W, on prediction accuracy. Percentages represent the likelihood of match between predicted and observed secondary structures for all residues.

W	Total % correct	Coil % correct	Sheet % correct	Helix % correct
3	56.3	62.3	47.1	51
5	59.1	64.6	49.3	54
7	60	64.9	62	55
9	60.3	63.9	52.8	56.1
11	57.8	63.7	55.6	49
13	62.4	66.7	55.6	58.6
15	63	66.6	56.3	59.8
17	63.4	66.3	55.8	62
19	62.9	67.2	55.6	59.6
21	62.5	65.3	56.4	60.5

with respect to the training set. In general, increasing the window size increases the prediction accuracy for each secondary structure, but with the increased window size, number of subset of sequences decreases which actually decreases the size of the training set.

TABLE II: Effect of hidden layer size on prediction accuracy

# of hidden layers	Total % correct	Training % correct	Validation % correct
6	55.5	56.1	55.3
8	63.2	64.2	58.8
10	63.4	64.3	59.9
12	62.4	63.1	59.9
14	62.7	63.6	59.9
16	61	62.2	58
18	62.1	63.1	60.1
20	59.8	60.4	57.1

In Table II, we show how the size of hidden layer affects the prediction ability and fitting with respect to training. Although we would expect to have increasing prediction ability in training sets and decreasing prediction ability, we did not observe such a strong pattern in our neural network. Compared to the work of Karplus et. al. [5], where fixed number of iterations, 500, were used to train the neural network, we think that using number of validation checks as a convergence criteria prevents such overfitting of the training set. For most of the times, when we train our neural networks, optimization is terminated using less than 100 iterations, but in all cases effective optimization stopping criteria was the number of validation checks.

In Figure 2, we plot confusion matrix in order to analyze the performance of predicted and observed secondary structures. The bold numbers in diagonal cells show the number of residue positions that are correctly identified with the neural network for each structural class. Bold numbers in off-diagonal, red colored cells show the number of misidentified structures for each target class in the horizontal axis. Three vertical gray cells show the accuracy for each output class, whereas three horizontal cells represent the probability that target class and the output class is the same. Finally in the blue cell, total percentage accuracy of predicted residues is represented. In gray and blue cells, percentages in green color represent the cases when neural network is accurately able to predict the correct structure, whereas red colored percentage represents the case when it is represented inaccurately.

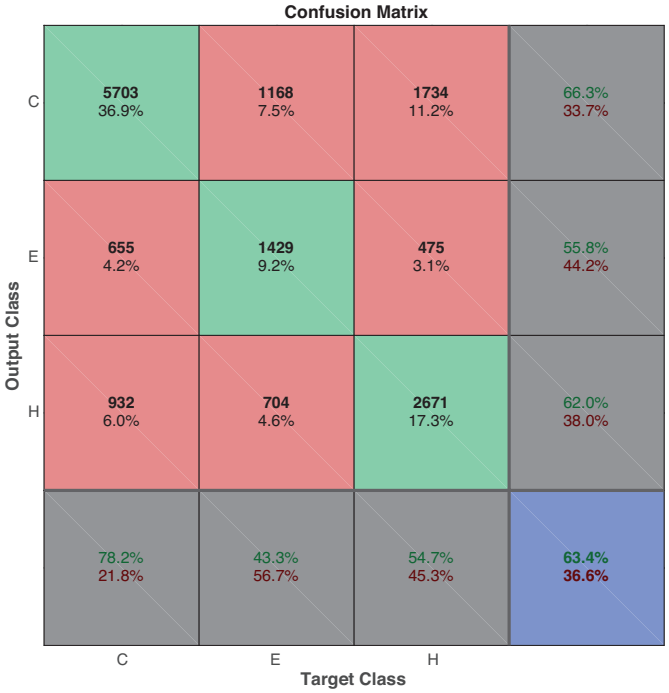

FIG. 2: Confusion matrix for all the sets sontained in this study: training, validation and training sets. Output and target classes of C, E and H again stands for coil, sheet and helix respectively. Diagonal squares show the number of residues where a match between observed and predicted secondary structure occurs. Non diagonal squares in red represent when inaccurate match occurs between the predicted and observed secondary structure. Blue square shows the success rate of secondary structure assignments for all residues.

IV. DISCUSSION

The method presented here only takes into account the sequence of the aminoacids in the structure, however it does not take into account more complicated interactions, such as length of each structure, long range interactions which lead to tertiary structure formation. Although 65% accuracy is impressive for such a simple model, realistic applications may require much larger accuracies in order to determine tertiary structure of a protein. Information related to long range interactions

can be incorporated in to such a model that can increase its predictive abilities. In its present form, this network trains all the residues independently from each other. However, we can say that a sheet would consist of a minimum of 3 residues, where as three or four distinct points are required to define a helix[6]. Therefore, in practice, we can train a subsequent neural network which will train the results of the neural network studied here, to make use of such geometrical constraints. Also, given additional structural information, such as angles between each residue, it can be possible to obtain larger accuracy.

* Electronic address: kayahan@mit.edu

[1] B. Rost and C. Sander, Journal of Molecular Biology **232**, 584 (1993).
[2] G. E. H. David E. Rumelhart and R. J. Williams, Nature **323**, 533 (1986).
[3] J. J. Hopfield, PNAS **79**, 2554 (1982).
[4] J. Hopfield and D. Tank, Science **233**, 625 (1986).
[5] L. H. Holley and M. Karplus, PNAS **286**, 152 (1989).
[6] A. H. Alain Goriely, Sebastien Neukirch, Note Mat. **32**, 87 (2012).