PROTEIN STRUCTURE PREDICTION USING NEURAL NETWORKS & SUPPRT VECTOR MACHINES

Badal Bhushan[1], Manish Kumar Singh[2]

[1] *Badal Bhushan,* Assistant Professor, Department of Computer Science & Engineering,
IEC College of Engineering & Technology, Greater Noida (U.P.), India.
bhushan_badal@yahoo.com

[2] Manish Kumar Singh, Assistant Professor, Department of Computer Science & Engineering,
Galgotias College of Engineering & Technology, Greater Noida (U.P.), India.

Abstract

Predicting the structure of proteins is important in biochemistry because the 3D structure can be determined from the local folds that are found in secondary structures. Moreover, knowing the tertiary structure of proteins can assist in determining their functions. The objective is to compare the performance of Neural Networks (NN) and Support Vector Machines (SVM) in predicting the secondary structure of 62 globular proteins from their primary sequence. For each NN and SVM, we created six binary classifiers to distinguish between the classes helices (H), strand (E), and coil (C). For NN we use Resilient Back-propagation training with and without early stopping. We use NN with either no hidden layer or with one hidden layer with 1,2,...,40 hidden neurons. For SVM we use a Gaussian kernel with parameter fixed at = 0.1 and varying cost parameters C in the range [0.1,5]. 10-fold cross-validation is used to obtain overall estimates for the probability of making a correct prediction. Our experiments indicate for NN and SVM that the different binary classifiers have varying accuracies: from 69% correct predictions for coils vs. non-coil up to 80% correct predictions for stand vs. non-strand. It is further demonstrated that NN with no hidden layer or not more than 2 hidden neurons in the hidden layer are sufficient for better predictions. For SVM we show that the estimated accuracies do not depend on the value of the cost parameter. As a major result, we will demonstrate that the accuracy estimates of NN and SVM binary classifiers cannot distinguish. This contradicts a modern belief in bioinformatics that SVM outperforms other predictors. Keywords: Neural Networks, Support Vector Machines, Protein Secondary Structure Prediction

Keywords—*Protein secondary structure prediction, SVM, coding schem.*

--- *** ---

INTRODUCTION

Predicting the structure of proteins from their primary sequence is becoming increasingly important in 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. Secondary structure prediction is an important intermediate step in this process because 3D structure can be determined from the local folds that are found in secondary structures.To set the background for the research study, the research context is explored. The objectives of the research and a summary of research methodology used are then presented. Furthermore, the thesis organization is outlined. There are different databases that record available protein sequences and their tertiary structures. However, sequence-structure gap is rapidly increasing. There are about 5 million protein sequences available from http://www.ebi.ac.uk/trembl/ and about fifty thousand protein sequences that are available in http://www.rcsb.org/pdb/. Different techniques have been developed that can predict secondary structure of proteins from their amino acids sequences. They are based on different algorithms, such as Statistical Analysis (Chou and Fasman, 1974), Information theory, Bayesian Statistics and Evolutionary Information (Sen, Jernigan, Garnier, and Kloczkowski, 2005), Neural Networks (Holley and Kurplus, 1989; Qian and Sejnowski, 1988), Nearest Neighbour Methods (Salamov and Salovyev, 1995), a combination of multiple alignment and Neural Networks (Rost and Sander, 1993). For these approaches, the accuracy levels are in the range 65–80%. This is examines the prediction of secondary structure of proteins from their sequences.

The two methods of prediction that are used are Neural Networks and Support Vector Machines. Neural Networks have wide applications in pattern classification. They are useful for classification and function approximation. There are different types of networks used for different applications. The most commonly used is the Multilayer Feed Forward Networks. For these networks, there are 3 types of layers - input layer, hidden layers and output layer. The input layer consists of neurons that receive information (inputs) from the external environment. The hidden layer accepts information from the input layer and communicates it to other hidden layers. The information from the last hidden layer is sent to the output layer. Also for this type of networks, the layers are fully connected respectively and preceding layers are not allowed to communicate the information back to the other layers of a network. For many networks, one hidden layer is sufficient to approximate any given function with the required precision. Practical applications of Neural Networks most often employ Supervised Learning. For supervised learning, training data that includes both the input and the desired result (the target value) is provided. After successful training, input data is presented to the Neural Network and the Neural Network will compute an output value that approximates the desired result. However, if the outputs do not approximate the desired outputs well, more training is performed. This means that the parameters of the network are adjusted until the output is close to the target value. For training of Neural Networks, Resilient Back propagation is used. Once training has been performed, the network is evaluated to see whether the training process was successful. This evaluation process can be done using the test set. The purpose of test set is to see if the network is able to identify unfamiliar data, and classify them into different classes.

Neural Networks have been applied in secondary structure prediction. Some of the applications can be found on the work. Support Vector Machines are machine learning algorithms implemented for classification and regression. For classification, Support Vector Machines operate by finding a separating hyper plane in the space of possible inputs. This hyper plane attempts to split the positive examples from the negative examples. The split will be chosen to have the largest distance from the hyper plane to the nearest of the positive and negative examples. Data points that are at the margin are called Support Vectors. These data points are very important in the theory of Support Vector Machines because they can be used to summarise information contained in the dataset. The hyper plane with a maximum margin allows more accurate classification of new points. However, not all problems can be linearly separated by a hyper plane. For such problems, the resulting algorithm is formally similar, except that a non linear transformation of the data into a feature space is performed. This allows the algorithm to fit the maximum margin hyper plane in the transformed feature space. Kernels are used to perform the mapping. Support Vector Machines have previously been shown to predict the secondary structure of proteins.

Method:

Different approaches have been proposed to achieve prediction of secondary structures from the amino acid sequences. The following methodology is employed:

1. The main aim is to train the Neural Network and a Support Vector Machine to respond to the sequence of proteins when the predictions of the secondary structures are known. We also want to achieve recognition of amino acid patterns associated with the secondary structures. To perform the required classification of primary sequences into their secondary sequences, Matlab programs or codes will be set up. For Neural Networks, the toolbox Version 5.0.2 (R2007a) will be used while for SVM, Matlab codes will be used.

2. The data to be used consists of 62 proteins from Rost and Sander (1983) database available from http://www.anteprot-pbil.ibcp.fr/. It contains a protein name, its primary and secondary sequences.

3. Preparation for the data for processing is done in steps. The first step performed is pre-processing. The data is presented in letters and the purpose of pre-processing is to convert those letters into real numbers. To achieve this, orthogonal coding, a similar coding scheme adopted by Holley and Karplus (1989) is used. The second step is secondary structure assignment. Secondary structures are classified into 8 categories H,G,E,B,I,T,S and the last category is for unclassified structures. This are reduced to 3 categories of H, E and C by using a secondary structure assignment called DSSP.

4. The Matlab codes implemented are for two class problems. And the following binary classifiers are created: the One-Against-All classifiers (helix vs. no helix, sheet vs. no sheet and coil vs. no coil) and the One-against-One classifiers (helix vs. sheet, sheet vs. coil and coil vs. helix). 5. 10-fold cross validation will be used to

estimate the performance of the selected model. It also assists in comparing the two machine learning algorithms since it ensures equal and consistent partitioning of the data. 6. For the comparison of both Neural Network and Support Vector Machines, the overall

5. 10-fold cross validation will be used to estimate the performance of the selected model. It also assists in comparing the two machine learning algorithms since it ensures equal and consistent partitioning of the data.

6. For the comparison of both Neural Network and Support Vector Machines, the overall accuracy is used by

$$P(\mathrm{correct}) = \frac{1}{n}\sum_{j=1}^{c} N_{jj}$$

Here N is the number of all samples and Njj is the number of correctly classified samples in class j. This accuracy measure gives the proportion of correctly classified samples. Their names are abbreviated in a three letter code or a one letter code.

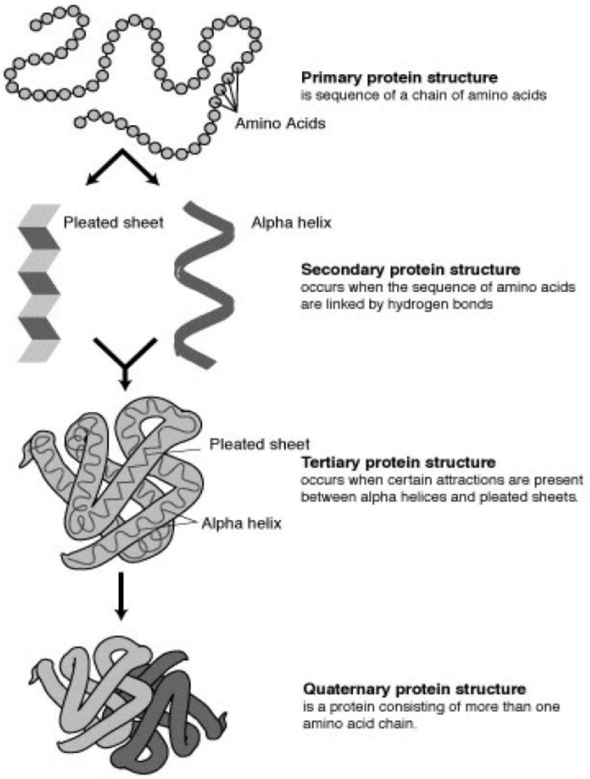
Proteins: Proteins are made of simple building blocks called amino acids. According amino acid is “a compound consisting of a carbon atom to which are attached a primary amino group, a carboxylic acid group, a side chain (R group), and an H atom. Also called an amino acid.” There are 20 different amino acids that can occur in proteins. The amino acids and their letter codes are given in the Table

Table-1: List of Amino acids and Symbols

Sl. No.	Amino acids	Ambiguous characters	Symbols
1	Glycine	Gly	G
2	Alanine	Ala	A
3	Serine	Ser	S
4	Threonine	Thr	T
5	Cysteine	Cys	C
6	Valine	Val	V
7	Isoleucine	Iie	I
8	Leucine	Leu	L
9	Proline	Pro	P
10	Phenylalanine	Phe	F
11	Tyrosine	Try	Y
12	Methionine	Mer	M
13	Tryptophan	Trp	W
14	Asparagine	Asn	N
15	Glutamine	Glu	Q
16	Histidine	His	D
17	Aspartic Acid	Asp	D
18	Glutamic Acid	Glu	E
19	Lysine	Lys	K
20	Arginine	Arg	R

Functions of proteins: Proteins are important for living organisms. For example, our body fabric is made of protein molecules. Some of the proteins are found in the food we eat. Proteins also help to build their own protein molecules. They serve as hormones, receptors, storage, defence, enzymes and as transporters of particles in our bodies.

Protein Structure: There are four different structure types of proteins, namely Primary, Secondary, Tertiary and Quaternary structures. Primary structure refers to the amino acid sequence of a protein. It provides the foundation of all the other types of structures. Secondary structure refers to the arrangement of connections within the amino acid groups to form local structures. α helix strand are some examples of structures that form the local structure. Tertiary structure is the three dimensional folding of secondary structures of a polypeptide chain. Quaternary structure is formed from interactions of several independent polypeptide chains. The four structures of proteins are shown in figure.


Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.
Alpha helix
Quaternary protein structure
is a protein consisting of more than one amino acid chain.

Ref: http://en.wikibooks.org/wiki/Proteomics/Introduction_to_Proteomics

Fig-1: Protein Structure

Prediction: The primary structure of proteins can be used to predict its tertiary structure. It is through the tertiary structure of the protein that we can derive its properties as well as how they function in an organism. Currently, there exist databases with sequence structures of proteins whose tertiary structures are known e.g. Research Collaborator for Structural Bioinformatics (RCSB) available from http://www.rcsb.org/pdb/. As of 29 January 2008, RCSB reported 48,638 available structures in the protein data bank. The results are achieved based on the following experimental approaches: X-ray, Nuclear Magnetic Resonance (NMR), Electron Microscopy (EM), and (other) which are the methods not mentioned. Table 2.2 shows a breakdown of structures available in RCSB from different experimental methods. Table 1: Protein Data Bank current holdings breakdown as at Tuesday 29 January 2008

Prediction:

The primary structure of proteins can be used to predict its tertiary structure. It is through the tertiary structure of the protein that we can derive its properties as well as how they function in an organism. Currently, there exist databases with sequence structures of proteins whose tertiary structures are known e.g. Research Collaborator for Structural Bioinformatics (RCSB) available from http://www.rcsb.org/pdb/. As on 29 January 2008, RCSB reported 48,638 available structures, as on Tuesday 26 May 2009 RCSB reported 57,558 available structures and as on Tuesday 26 May 2009 RCSB reported 81,369 available structures in the protein data bank. The results are achieved based on the following experimental approaches: X-ray, Nuclear Magnetic Resonance (NMR), Electron Microscopy (EM), and (other) which are the methods not mentioned.

Table-2: Protein Data Bank holdings breakdown as on Tuesday 29 January 2008

Experimental methods	Protein Structures	Nucleic Acids	Protein/ NA Complexes	Other	Total
X-ray	38541	1016	1770	24	41351
NMR	6080	802	137	7	7076
Electron Microscopy	112	11	41	0	164
Other	87	4	4	2	97
Total	**44820**	**1833**	**1952**	**33**	**48688**
Reference from : http://www.rcsb.org/pdb/.					

Table-3: Protein Data Bank holdings breakdown as on 26 January 2009

Experimental methods	Protein Structures	Nucleic Acids	Protein/ NA Complexes	Other	Total
X-ray	46064	1142	2118	17	49341
NMR	6840	849	143	6	7838
Electron Microscopy	163	16	59	0	238
Hybrid	13	1	1	1	16
Other	108	4	4	9	125
Total	**53188**	**2012**	**2325**	**33**	**57558**

Reference from :
http://www.rcsb.org/pdb/statistics/holdings.do

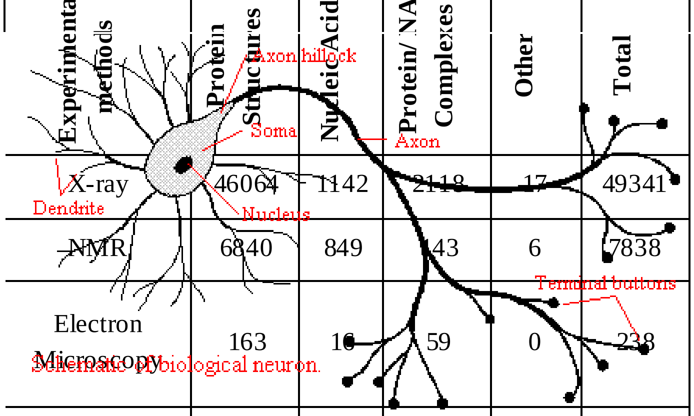


Fig-2 The Biological Neuron
Ref : http://vv.carleton.ca/~neil/neural/neuron-a.html

Table-4: Protein Data Bank holdings breakdown as of Tuesday May 08, 2012 at 5 PM PDT there are 81369 Structures

Experimental methods	Protein Structures	Nucleic Acids	Protein/ NA Complexes	Other	Total
X-ray	66672	1358	3315	2	71347
NMR	8211	979	186	7	9383
Electron Microscopy	285	22	120	0	427
Hybrid	44	3	2	1	50
Other	140	4	5	13	162
Total	**75352**	**2366**	**3628**	**23**	**81369**

Reference from :
http://www.rcsb.org/pdb/statistics/holdings.do

Table-5: Protein Data Bank current holdings breakdown As of Tuesday Feb 04, 2014 at 4 PM PST there are 97591 Structures

Experimental methods	Protein Structures	Nucleic Acids	Protein/ NA Complexes	Other	Total
X-ray	80621	1502	4194	4	86321
NMR	9024	1072	197	7	10300
Electron Microscopy	508	51	170	0	729
Hybrid	57	3	2	1	63
Other	155	4	6	13	178
Total	90365	2632	4569	25	97591

Reference from :
http://www.rcsb.org/pdb/statistics/holdings.do

Neural Networks:

Basic Theory of Neural Networks , Biological inspiration Artificial Neural Networks receive their motivation from biological systems because they are based on the idea of imitating the human brain. The neuron is the basic unit of the brain that processes and transmits information. A neuron has the following structural components: synapses, dendrites, cell body, and axon. Neurons have their connections through the synapses. Synapses communicate and process information through the dendrites into the cell body. Each dendrite may have many synapses attached to it. The signals are accumulated in the cell body and when their quantity reaches a certain level (usually referred to as the threshold), then the new signal is released and carried to other neurons through the axons.

What are Neural Networks?

"A massively parallel processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects: knowledge is acquired by the network through a learning process.

Interneuron connection strengths known as synaptic weights are used to store the knowledge." It is difficult to distinguish between Neural Networks and other modern statistical methods. However, it is imperative to note that for the test set of synthetic datasets, the two populations are normally distributed. This implies that the results of Neural Networks were also obtained under the same assumptions as other methods. This might have a possible influence on the results and perhaps if there were no statistical model assumptions, Neural Networks would have performed better. The motivation could be that Neural Networks do not need statistical model assumptions to operate.

Why use Neural Networks rather than statistical approaches for prediction?

Neural Networks have applications in speech and pattern recognition, time series and in recent years, predicting the secondary structure of proteins. Literature shows their performance and success in predicting the secondary structure of proteins from their amino acid sequences (primary sequence) with prediction accuracy of 64.3% from earlier approaches to 70% and better based on multiple sequence alignment rather than single sequences. Neural Networks have properties which distinguish them from other machine-learning and statistical techniques. They are adaptive systems that build a model that finds relationships between inputs and outputs. The input-output mapping is established through changing the parameters of the network to obtain the desired output. They can approximate any function to any prescribed precision. Unlike most statistical approaches, Neural Networks are not based on the assumption of a particular probability model.

Why use Neural Networks rather than statistical approaches for prediction?

Neural Networks have applications in speech and pattern recognition, time series and in recent years, predicting the secondary structure of proteins. Literature shows their performance and success in predicting the secondary structure of proteins from their amino acid sequences (primary sequence) with prediction accuracy of 64.3% from earlier approaches to 70% and better based on multiple sequence alignment rather than single sequences. Neural Networks have properties which distinguish them from other machine-learning and statistical techniques. They are adaptive systems that build a model that finds relationships between inputs and outputs. The input-output mapping is established through changing the parameters of the network to obtain the desired output. They can approximate any function to any prescribed precision. Unlike most statistical approaches, Neural Networks are not based on the assumption of a particular probability model.

A model of a Neuron:

Suppose we have an input vector $x = (x_1, x_2, x_3, \dots \dots \dots \dots, x_n)2R^n$ of real numbers. The inputs are sent through the connections to the accumulator and the connections may carry some weight vector $w = (w_1, w_2, w_3, \dots \dots \dots \dots, w_n)2R^n$ .These are applied to the inputs x_i by multiplying the input with its corresponding weight w_i. The products are then added and the information sent to the accumulator is of the form

$$\langle w, x\rangle = \sum_{i=1}^{n} w_i * x_i$$

The accumulated sum of the inputs is compared to a value called the threshold such that if greater than the threshold, then the output is 1 and if less, it is 0. The threshold adds a charge known as the bias to the sum of the signals. The bias is denoted as b. The output produced is given as

$$y = f[\langle w|, |x\rangle + b] \qquad (1)$$

where f is an activation function and y is the output signal. An example of an activation function is the Heaviside function and it is given by

$$f(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases} \qquad (2)$$

The bias can be included in the neuron in which case, the input x0 has a value of +1 attached and multiplied to give a threshold value w0 = b and added to the sum to produce equivalent output.

$$y = f[\langle w, x\rangle] = f(\textstyle\sum_{i=0}^{n} w_i * x_i) \qquad (3)$$

The mapping from the inputs $(x_1, x_2, x_3, \dots \dots \dots \dots, x_n)$ to the output y is shown in Figure 3 .

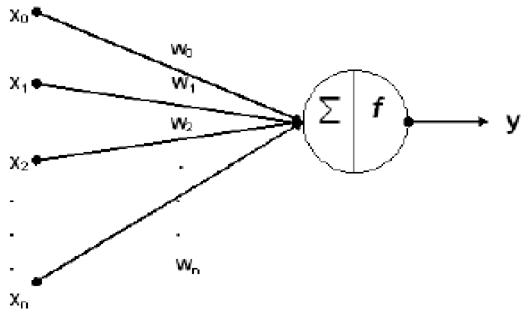


Fig 3 : Basic neuron model

As per following figure, a step function f is applied to P which represents the sum obtained from the products of weights and biases for the neuron. This model represents a Perception. A perception is a neural network with binary outputs, i.e. as in a Heaviside function, where the output has values of 0 or 1. For a set of points, a perception is used in classification. Consider a situation in which we have inputs $x = (x_1, x_2)$ belonging to classes A and B. This gives rise to the form of classification illustrated in Figure. A point that lies above the line belongs to class 1 and a point below the line belongs to class 2.

The decision boundary W is generated by the perception. If the boundary does not give a correct classification the weights of the perception are changed until the correct output is achieved. The success of this method depends on whether classification gets the learning task (inputs and matching targets) right or not.

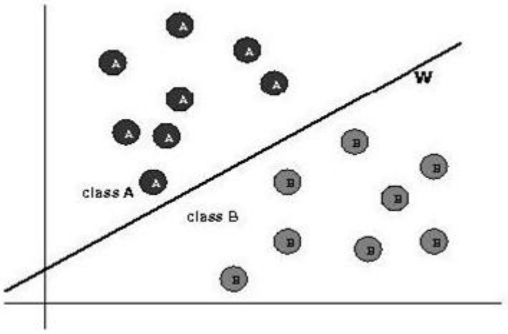


Fig 4: Classification to illustrate the perception

If the perception gets the classification right, no adjustment is required. However, if it gets the classification wrong, then the weights are changed to get it right.

A layer of Neurons:

One layer of neurons generally, there will be more than one neuron in a network. For such networks, inputs are connected to the neurons and then the output is obtained. Consider a layer with two neurons. Figure 5 shows such a network.

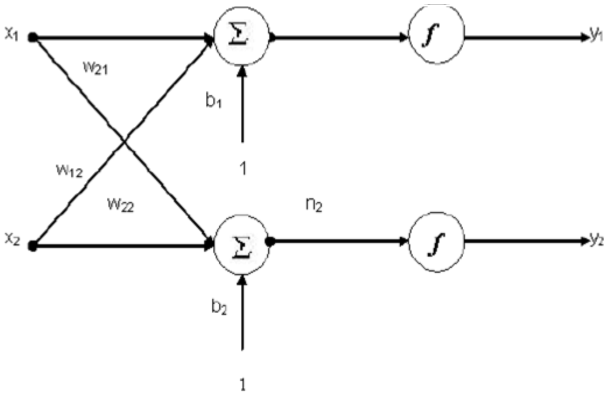


Fig 5 : A network with a layer of neurons

This network has two neurons in a layer. A typical network will have more than two neurons for each layer. The weights of connections of neurons 1 and 2 are respectively denoted as follows;

$$(w_{11}, w_{12}), (w_{21}, w_{22})$$

By convention, the indices of the weights are:

1. First index = relates to a neuron
2. Second index = relates to the input

For the net outputs n_1 and n_2

$$n_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$n_2 = w_{21}x_1 + w_{22}x_2 + b_2 \qquad (4)$$

and using the matrix multiplication, we can write this in matrix-vector form as:

$$\begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

The matrix

$$W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

is called the weight matrix. To the net output, a transfer function f is applied component wise to obtain the following output:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = f\begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = f\left(\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right) \qquad (5)$$

To generalize to the case of a neural network with n inputs x_1 , x_2 ……….., x_n and m neurons and m outputs (y_1 , y_2 ……….., y_n) the weight matrix is

$$\begin{bmatrix} w_{11}\, w_{12} \,.\;.\;.\;. & w_{ln} \\ w_{21}\, w_{22} \,.\;.\;.\;. & w_{2n} \\ . & . \\ . & . \\ . & . \\ . & . \\ w_{m1}\, w_{m2} \,.\;.\;.\;. & w_{mn} \end{bmatrix}$$

and the bias vector (b_1 , b_2 ……….., b_n) The net output becomes

$$\begin{pmatrix} n_1 \\ n_2 \\ : \\ . \\ . \\ n_m \end{pmatrix} = \begin{bmatrix} w_{11}\, w_{12} \,.\;.\;.\;. & w_{ln} \\ w_{21}\, w_{22} \,.\;.\;.\;. & w_{2n} \\ . & . \\ . & . \\ . & . \\ . & . \\ w_{m1}\, w_{m2} \,.\;.\;.\;. & w_{mn} \end{bmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ : \\ . \\ . \\ x_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ : \\ . \\ . \\ b_m \end{pmatrix} \qquad (6)$$

Component wise evaluation of the transfer function yields

$$\begin{pmatrix} y_1 \\ y_2 \\ : \\ . \\ . \\ y_m \end{pmatrix} = f(\begin{bmatrix} w_{11}\, w_{12} \,.\;.\;.\;. & w_{ln} \\ w_{21}\, w_{22} \,.\;.\;.\;. & w_{2n} \\ . & . \\ . & . \\ . & . \\ . & . \\ w_{m1}\, w_{m2} \,.\;.\;.\;. & w_{mn} \end{bmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ : \\ . \\ . \\ x_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ : \\ . \\ . \\ b_m \end{pmatrix}) \qquad (7)$$

Therefore, from a given list of inputs $(x_1 , x_2$ ……….., $x_n)' \in R^n$ and the output $(y_1 , y_2$ ……….., $y_m)' \in R^m$ is computed. With the notation x0 = 1 and w0k = bk this equation becomes:

$$\begin{pmatrix} y_1 \\ y_2 \\ : \\ . \\ . \\ y_m \end{pmatrix} = f(\begin{bmatrix} w_{01}\, w_{02} \,.\;.\;.\;. & w_{0n} \\ w_{11}\, w_{12} \,.\;.\;.\;. & w_{ln} \\ . & . \\ . & . \\ . & . \\ . & . \\ w_{m1}\, w_{m2} \,.\;.\;.\;. & w_{mn} \end{bmatrix} . \begin{pmatrix} 1 \\ x_1 \\ : \\ . \\ . \\ x_n \end{pmatrix}) \qquad (8)$$

$F = R^n \rightarrow R^m$ given by (7) or (8) is called the input-output mapping of the neural network. It depends on the weights and biases. More than one layer of Neurons

A neural network can have a single output, e.g. a perception or more than one output. Networks may be designed to have more than one layer, in which case we will have the input layer, hidden layers and the output layer. The hidden layers are the layers between the input and output layer. The case of one hidden layer is shown in Figure 6

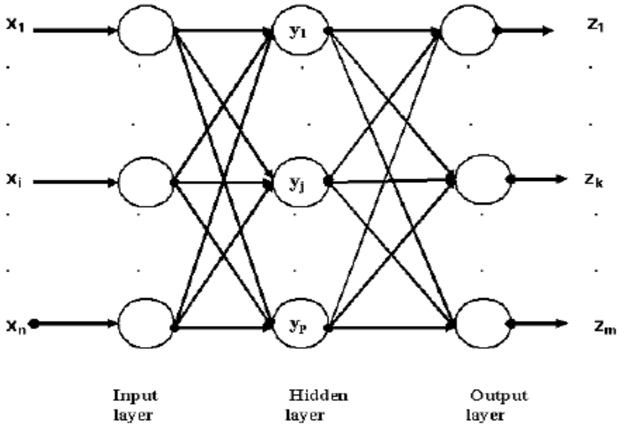


Fig: 6 : More than one layer of Neurons

The input-mapping $F: R^n \rightarrow R^m$ of a neural network with a layer of neurons can be extended to the case in which there is a layer of neurons between the inputs and the outputs. Let us change the notation a little. The inputs are $(x_1 , x_2$ ……….., $x_n)' \in R^n$ as before, but the notation for the outputs $(y_1 , y_2$ ……….., $y_m) \in R^m$ will change to $(y_1 , y_2$ ……….., $y_p) \in R^p$ Therefore, (3.8) is rewritten as:

$$\begin{pmatrix} y_1 \\ y_2 \\ : \\ . \\ . \\ y_p \end{pmatrix} = f(\begin{bmatrix} w_{01}\, w_{02} \,.\;.\;.\;. & w_{0n} \\ w_{11}\, w_{12} \,.\;.\;.\;. & w_{ln} \\ . & . \\ . & . \\ . & . \\ . & . \\ w_{p1}\, w_{p2} \,.\;.\;.\;. & w_{pn} \end{bmatrix} . \begin{pmatrix} 1 \\ x_1 \\ : \\ . \\ . \\ x_n \end{pmatrix}) \qquad (9)$$

If we denote the weight matrix in (9) as W, then (9) can also be written as y = f(W · x). To compute the output (z1, z2, ..., zm)0 2 Rm for a network with 1 hidden layer, the output(y1, y2, ..., yp) 2 Rp from the previous layer is used as an input into the next layer. Therefore, it follows that

$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ . \\ . \\ z_m \end{pmatrix} = f\left(\begin{bmatrix} v_{01} v_{02} . & . & . & . & v_{0p} \\ v_{11} v_{12} . & . & . & . & v_{1p} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ w_{m1}\, w_{m2} . & . & . & . & w_{mp} \end{bmatrix} . \begin{pmatrix} 1 \\ y_1 \\ \vdots \\ . \\ . \\ y_p \end{pmatrix} \right) \qquad (10)$$

and if V denotes the weight matrix in (10), then $z = f\langle V.y\rangle$ .We can therefore summarise that $F\ (x) = f\langle V.y\rangle = f\langle V.f\langle W.x\rangle\rangle$. Hence, the input-output mapping of a network with 1 hidden layer is a function $F: R^n \rightarrow R^m$

Early stopping approach :

A problem of learning with NN is overtraining. The reason is that NN adapts too strongly to the training set and predicts unseen samples poorly. One approach that can be used to achieve better performance on unseen data is early stopping. In this technique, the available data is divided into three subsets: training set, validation set and the test set. The training set is used to compute and update the parameters of the network. During training, the training error decreases as the number of iterations increases. The error on the validation set is also monitored. It decreases as well up to a certain number of iterations but then will increase with further training. Training will stop after a certain specified number of iterations i.e. 500 epochs. And the weights and biases that occur at the minimum of validation error are the parameters of the network. The error measures on training set and validation set are shown in Figure 7 to 8 the error on the training set and validation set decreased until at epoch 19. After 19 epochs, the validation error started to increase and training stopped after 42 epochs. Up to that point, the validation error was increasing. The parameters of the network in this case will be those that occur at 19 epochs where the validation error is at minimum. To obtain a good estimate of the accuracy of the classifier, we use a separate data, the test set. This should be independent from data used for training and validation.

Training and validation set

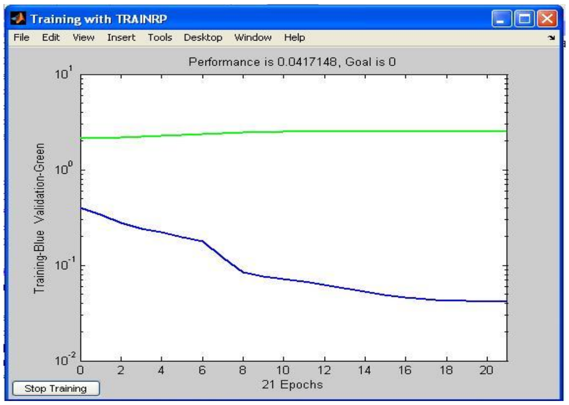

Training with TRAINRP
File Edit View Insert Tools Desktop Window Help
Performance is 0.0417148, Goal is 0
10^1
10^0
10^{-1}
10^{-2}
Training-Blue Validation-Green
0 2 4 6 8 10 12 14 16 18 20
21 Epochs
Stop Training

Fig: 7 (21 Epochs)

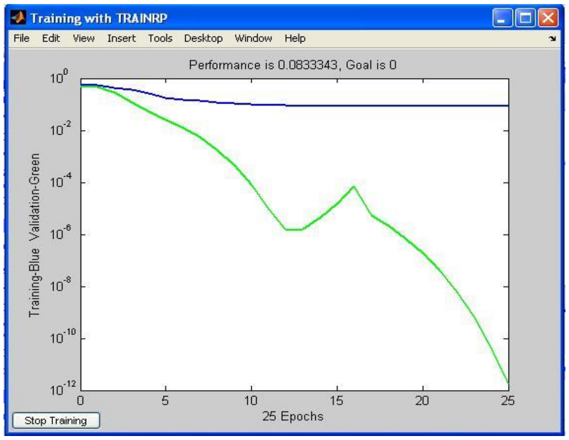


Fig: 8 (25 Epochs)

SUPPORT VECTOR MACHINES

Support Vector Machines are machine-learning algorithms that were developed by Vapnik and co-workers. They constructed a separating hyperplane with the objective of separating the data into different classes. The separating hyper plane should have maximum distance to the nearest data point in each class.

Table no 6 : Comparison of Neural Networks and Support Vector Machines

Binary Classifier	NN % accuracy	SVM % accuracy
H/~H	74.63	75.35
E/~E	79.68	80.15
C/~C	68.82	68.81

H/E	72.37	74.65
E/C	76.16	75.94
C/H	73.27	73.63

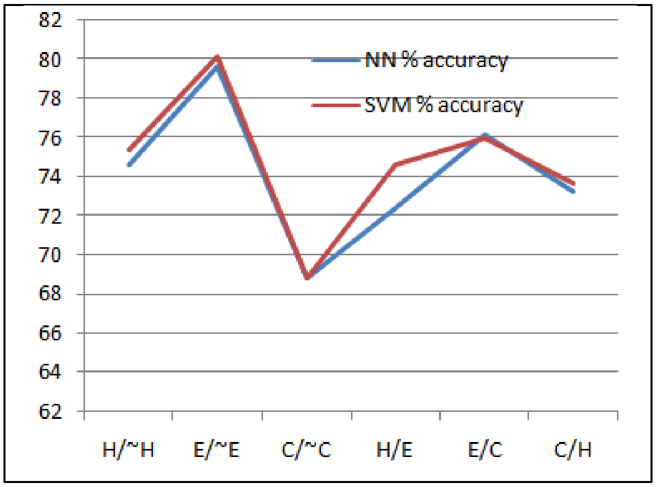


Fig no 8: Comparison of Neural Networks and Support Vector Machines

The results in the above Table show the performance of both NN and SVM in predicting the secondary structure of proteins. Generally, there are no big differences in the overall accuracies of the classifiers for NN and SVM. For the One-against-All approaches, SVM achieved the highest prediction accuracy of about 80%. But in One-Against-One approaches, NN achieved the highest accuracy of about 76%. The highest prediction accuracies were observed for E/~E in the two methods while the lowest overall accuracy was observed for C/~ C in both methods. Figure is created to reveal if there are any differences in the performance of NN and SVM. The results in Tabl are used to achieve this. From this graph, it is apparent that accuracies for SVM are higher than in NN on all binary classifiers. However, the difference in the values is very small. For H/E, the difference is about 3% between the SVM and NN. Numerically,76.16% and 75.94% are different. However, we must ask whether it is possible to distinguish between the performances of NN and SVM approaches from the practical point of view based on these accuracies.

As in Neural Networks, differences in the time required to train some classifiers at C = 5 can also be attributed to the different number of samples presented to each classifier. It has been established that One-Against-All classifiers take longer time during training than One-Against-One binary classifiers. Let us recall that One-Against-All classifiers use 10766 samples for each learning task while for One-Against-One, samples differ: for H/E there are 5535 samples, 7719 samples for E/C and H/C has 8478 samples (The samples are stored). The total number of samples for each classifier (for each learning task) is related to training time and it can be expected that H/E will take the shortest time to train while, H/C will take the longest time to train for One-Against-One classifiers. At the same time, it can be expected that when the total number of samples used to create a learning task is about the same, time required during training should be about the same.

Protein structure prediction is an important step towards predicting the tertiary structure of proteins (and Quartenary structure). The reason is that knowing the tertiary structure of proteins can help to determine their functions. The main aim of this thesis was to compare performance of Support Vector Machines and Neural Networks in predicting the secondary structure of proteins from their amino acid sequences. The following conclusions were derived:
The contributions of the research are indicated and identified the future research.

Conclusions and discussion:

- For Neural Networks, there is a dependence of prediction accuracy on the number of hidden neurons. Furthermore, either a simple architecture with no hidden layer, or one hidden layer with 1 hidden neuron is more sufficient than architecture with e.g. 40 hidden neurons. Also, early stopping guarantees a good classifier that is independent of the number of hidden neurons.

CONCLUSIONS:

As per above study the presented the results obtained from comparing the performance of Neural Networks and Support Vector Machines in predicting secondary structure of proteins from their secondary structures. The above figure gives a conclusion and discussion of the overall research investigation. Although in Support Vector Machines there seems to occur over-adaptation to the training set (the training samples are learnt completely), it is interesting that the overall accuracy (P(correct)) on the test data is unaffected.

- Neural Networks with early stopping or with simple architecture are far superior to Support Vector Machines with respect to training time. Support Vector Machines are also demanding in terms of memory, whereas Neural Networks can be applied on less powerful computers.

- In respect of the overall accuracy estimates for the test data, Support Vector Machines are slightly superior in almost all classification tasks. However, from a statistical point of view, they are not different. Except for the alpha vs. beta (H/E) classifier, confidence intervals do overlap for most of the binary classifiers, suggesting that the true accuracies obtained from both methods can be identical. However, from a practical point of view, there seems to be no real difference between the two methods.

- We conclude from this that reported differences in the literature for Support Vector Machines and Neural Networks are not due to the methods but are influenced by different designs of the classifiers. Therefore, some of the variables that may influence differences in the results include different input coding systems, window length, cross validation methods, the parameters used to create a learning task and even the structural class assignments.

Samples of proteins are take :

Acprotease :

GVGTVPMTDYGNDVEYYGQVTIGTPGKSFNLNFDTGS
SNLWVGSVQCQASGCKGGRDKFNPSDTFKATGYDASI
GYGDGSASGVLGYDTVQVGGIDVTGGPQIQLAQRLGG
GGFPGDNDGLLGLGFDTLSITPQSSTNAFQDVSAQGKVI
QPVFVVYLAASNISDGDFTMPGWIDNKYGGTLLNTNID
AGEGYWALNVTGATADSTYLGAIFQAILDTGTSLLILPD
EAAVGNLVGFAGAQDAALGGFVIACTSAGFKSIPWSIY
SAIFEIITALGNAEDDSGCTSGIGASSLGEAILGDQFLKQ
QYVVFDRDNGIRLAPVACTTC

CCCCBCTTSSCBBCCEECSSSCCEECCEEESSCCCBEEC
BSCCCSSSSCSSCCCBCTTTCSSCCCCCCBCCCCCSSCC
CEEBCCCCCCBSSSCBCCSCTTCEEEEECSSSSSSSSCSS
EEECSCSSSCSSSSSCCCHHHHHHHSSSCSSSCCCCEEEC
TTTCEEECCSSCCCTTSBCSCCCCBCCCCSSSSCCEECC
CCCSSSSCCCCSCEECCCTTSSSEECCSSTTTGGGTTSCC
CCCSSSSCCCBCSCTTSCCCCCCBSSSCBCCCSSSSCCCC
SSSCCCCCCCCCCSSSSBEECHHHHTTBCCEEETTTEEEC
CBBC

Aproteinase:

AASGVATNTPTANDEEYITPVTIGGTTLNLNFDTGSADL
WVFSTELPASQQSGHSVYNPSATGKELSGYTWSISYGD
GSSASGNVFTDSVTVGGVTAHGQAVQAAQQISAQFQQ
DTNNDGLLGLAFSSINTVQPQSQTTFFDTVKSSLAQPLF
AVALKHQQPGVYDFGFIDSSKYTGSLTYTGVDNSQGF
WSFNVDSYTAGSQSGDGFSGIADTGTTLLLLDDSVVSQ
YYSQVSGAQQDSNAGGYVFDCSSSVPDFSVSISGYTAT
VPGSLINYGPSGNGSTCLGGIQSNSGIGFLIFGDIFLKSQY
VVFDSDGPQLGFAPQA

CBCEEEEEEEECTTSCCEEEEEEEETTEEEEEEEEESSCCCEE
ECBSSSCHHHHTTSCCBCHHHHCEEEEEEEEEEEECTTSC
EEEEEEEEEEEEEETTEEEEEEEEEEECSEECHHHHHCTTCS
EEEECSCGGGCCBSSSCCCCHHHHHTTTBSSSEEEEECC
SSSCEEEEESCCCGGGBSSCCEEEECCGGGTSCBCCEEE
EEETTEEECCCCEEECTTCSSEEECHHHHHHHTTCSSE
EEETTTTEEEECTTSCCCCEEEEETTEEEEECGGGGEEEE
ETTTTEEEESEEECCSCSSEEECHHHHTTBCEEEETTTTE
EEECBBC
etc.

References

[1] Jaewon Yang , Departments of Electrical Engineering, Stanford University , CS229 Final Project, Dec 2008

[1] Sushmita Mitra,Senior Member , IEEE , and Yoichi Hayashi, Senor Member , IEEE, vol -36 No. 5, September 2006

[2] E. E. Lattman and G. D. Rose, "Protein folding-what's the question?" Proc. Natl. Acad. Sci. U.S.A. 90, 439-441, 1993.

[3] Hin Hark Gan, Rebecca A. Perlow, Sharmili Roy,Joy Ko, Min Wu, Jing Huang, Shixiang Yan,Angelo Nicoletta, Jonathan Vafai, Ding Sun, Lihua Wang, Joyce E. Noah, Samuela Pasquali, and Tamar Schlick, "Analysis of Protein Sequence/Structure Similarity Relationships" Biophysical Journal ,Volume 83,2781–2791, 2002.

[4] Pierre Baldi, Soren Brunik, Paolo Frasconi, Giovanni Soda and Gianluca Pollastri , "Exploiting the past and the future of protein secondary structure prediction"Bioinformatics,Vol.15,937-946, 1999

Author 1 : Badal Bhushan

Author 2 : Manish Kumar Singh