Prediction of Protein Folding
Using the Shift-Learn Method with a Large Scale Neural Network

Marius O. Poliac
George L. Wilcox
Yiyi Xin
Tidhar Carmeli
and
Michael Liebman

Minnesota Supercomputer Insititue
1200 Washington Ave. S
Minneapolis MN 55415, USA

Abstract

We have demonstrated in previous studies the utility of large neural network simulations for encoding the association between protein sequence and three dimensional structure for a small heterologous training set of small proteins. In the present study, we report the application of this approach to a selected homologous training set of 8 proteins using the Cray 2 supercompter at the Minnesota Supercomputer Center, Minneapolis USA. The large memory of this machine allowed us to configure a network with more than .3 million connections and 30,000 neural units; a network of this size was necessary to accommodate a new training/testing set with 8 proteins of up to 140 amino acid residues. This training set was constructed to investigate the performance of the neural network approach in prediction of structures within the protease class of proteins; proteases are enzymes which cleave the peptide bonds which join individual amino acid residues of other proteins. The network learned the sequence-structure association for 4 of the proteins within 100 iterations selected in a random order and shifted by a random offset to the left or to the right. When presented with novel sequences from related proteins, the network was able to predict three dimensional structures of the four proteins in the testing set. The results of this study suggest that a neural network trained to recognize the entire sequence of a protein using the shift-learn method can retain some of the rules of protein folding in a form which allows prediction of three dimensional structures. Our findings indicate that large scalar or vector supercomputer architectures are ideal for implementation of useful backpropagation neural networks.

Introduction

The three dimensional structure of a protein is intimately related to its biological function, and the amino acid sequence of a protein apparently determines this structure completely. The mechanism by which a protein's sequence determines its structure is not yet understood, and numerous previous attempts to predict how a protein folds have failed. Early mechanical or physical-chemical approaches (Levitt and Warshel, 1975; Chou and Fasman, 1978; Tanaka and Scherraga, 1978) could predict secondary structure with only 60% accuracy, and tertiary structure was inaccessible. Later neural network approaches were similarly inaccurate (Qian and Sejnowski, 1988; Holley and Karplus, 1989). One recent study attempted to apply a fairly large backpropagation neural network to learn sequence to secondary structure mapping for the serine protease family of proteins (Bohr et al., 1990). They encoded secondary structure as the distances from each amino acid on the polypeptide backbone to nearby amino acids, the near-diagonal elements of their distance matrices (see methods below). Limitations in computer memory and time limited their attention to these near-diagonal elements of distance matrices relating mostly to secondary structure, the local elements of protein folding. Our approach, by contrast, involves

CH 3065-0/91/0000-1323 $1.00 © IEEE

applying a very large neural network to learn entire distance matrices which encode tertiary as well as secondary structure elements (Wilcox et al., 1991).

We have applied a large neural network simulation called BigNet implemented on a four processor Cray 2 in an attempt to learn associations between sequences and structures of a group of proteins with known structure. The structures are derived from the Brookhaven Protein Data Bank (PDB) protein structure database. We have found that after several hundred learning iterations through a training set of 15 to 20 small proteins, training runs routinely converge to solutions where inputs of sequence coded by amino acid hydrophobicity yield output structure descriptions with less than 0.1% RMS deviation from actual structures.

Most recently, we have seen evidence of generalization after presentation of four testing proteins with known structure but with sequences which are new to the network; the trained network can correctly classify all of the novel sequences into one of four families and can accurately predict secondary and tertiary structures for the novel sequences (Wilcox, Xin, Carmeli and Liebman, unpublished). This performance was obtained with a heterologous training set of 20 small proteins. The present study is investigating the shift-learn approach to a homologous training set of only 8 proteins, most of which belong to a family called the serine proteases.

Method

Our simulation environment consists of: (1) a neural network simulator called BigNet, which is capable of handling backpropagation networks of 100 million connections at speeds of 0.5 million 64-bit connections per second using a single Cray 2 processor, and (2) a Network Description Language (NDL) which allows for the flexible design of backpropagation neural networks. NDL allows easy definition of a generalized back propagation network, provides a convenient means for conducting complex simulation experiments and permits the user to construct various input and output data presentation formats. The major NDL commands include the following functions: *layer* defines a layer of nodes; *connection* defines a connection (alias edge); *file* attaches a file to a layer or connection; *load* loads a layer or a connection; *save* saves a layer or a connection; *loadConnections* loads all connections (alias *loadEdges*); *saveConnections* saves all connections (alias *saveEdges*); *learn* initiates a simple learning session (*shiftLearn* learns from inputs or outputs which shift within the respective windows); *learnFrom* uses a list of files to build a training set which will be scanned sequentially or randomly through multiple iterations;*propagate* propagates the input to the output through the existing weights (*shiftPropagate* generates input arrays which have been shifted in the input window); array constructs a multidimensional array for input, output or hidden layers.

Learning takes place during each learning iteration as each weight in the network is changed in such a way as to minimize the error at the output layer. The change imposed on a weight at each learning iteration is determined on the basis of measured output errors (difference between desired and actual outputs), the values of nodes propagating through the connection and a parameter called e. We have found that setting e to 0.1 minimizes inter-iteration jumps and gives best results for the complexity of our training set. Each learning iteration consists of two distinct phases characteristic to the backpropagation algorithm. In the first phase, the state of the input layer given by its attached file is propagated to the output layers using the current values of the weights of each connection. During the second phase, the state of the output layers is compared to the desired output stored in the file attached to this layer. Then the weights are adjusted after each item in the training set is presented in such a way that the mean squared output error is minimized; if this process were conducted over the entire training set at once, the result would be very similar to multiple linear regression with several million coefficients. In effect each learning iteration on each item in the training set changes the weights in such a way that the newly created network will better match the input patterns stored in the files attached to the input layer with the output patterns stored in the files attached to the output layer. Learning is the most potentially time-consuming aspect of backpropagation learning.

This paper focused on a training set of 4 proteins which were presented to the network in random order and a testing set of 4 related proteins. The sequence and structure data were derived from the Protein Data Bank (PDB) files for each protein. For brevity, each protein is identified here using its four letter PDB code. Four proteases (2lz2, 5cyt, 1pzp, 1azu) were submitted as a training set with inputs shifted by a variable number of positions and keeping the outputs fixed. Four related proteases (4lyz, 1ccr, 3bp2, 2aza) were included in the test set.

The input or bottom layer of the network consisted of 140 units whose value was supplied by an input array describing the sequence of the protein in question; the sequences, ranging from 120-130 residues in length, were always registered in the 140 unit input layer. We have used one "alphabet" to describe the amino acids: hydrophobicity values adapted from Liebman et al. (1986) that ranged from -3.4 for the hydrophobic amino acid tyrosine to +3.3 for the hydrophilic amino acid lysine and were normalized into the range of ±1 (described in Wilcox et al., 1991). Since several amino acids have almost identical hydrophobicities, this alphabet is degenerate. We have chosen this alphabet for initial use because hydrophobicity may represent an important physico-chemical interaction driving protein folding (Eisenberg et al., 1984) and because the neural network could be configured most economically if continuous values were used instead of name-like categories.

The output or top layer of the network used in this study constitutes a window for distance matrices and is composed of 29,600 units (a 140 by 140 unit two-dimensional array) whose values are determined by forward propagation (i.e., they depend on the sums of all the weighted inputs) from lower layers. The three dimensional structure of each protein from the PDB was transformed into this two dimensional rotation-independent representation to facilitate presentation to the network. Essentially, the distance from each amino acid in each protein to every other amino acid is calculated from the alpha carbon coordinates in the PDB and placed in each element of the distance matrix; the distance matrix represents a "finger print" of each protein that preserves much of the three dimensional structure of the protein in an easily recognized, easily memorized form. An accurately predicted distance matrix for a protein would describe much of the detail of how the protein folds, locally in secondary structures such as alpha helices, and globally in supersecondary and tertiary structural elements. These distances were normalized into the range of 0-1 by dividing by the maximum distance in the training set, 90 Å. The unit-by-unit differences between this output layer and the superimposed target pseudo-layer (i.e., the actual distance matrix of the protein whose sequence is encoded in the input layer) are calculated for each learning iteration; these differences are used to determine the error propagated back to alter weights connecting the lower layers to the upper layers (Rummelhart et al., 1986).

Results

We report here the results of 12 learning and testing sessions which required 2 hours of CPU time on the Cray 2. The network of 29,600 output nodes and 60 to 90 hidden units contained more than .3 million connections. Because the value of each connection was held in one 64 bit floating point word, the problem required almost 3 million bytes of central memory. One hundred learning iterations through the training set of 8 proteases required 1.2 billion cumulative updates of inter-unit connection weights. Using the central processor in scalar, double precision mode with BigNet compiled using the standard C language compiler delivered a speed of 437,000 connection updates per second; this learning speed is the most meaningful benchmark for backpropagation networks because the programs spend most time in the learning mode. The speed of forward propagation (the act of producing an output from an input through the existing weights) is perhaps ten times this learning mode speed, but the program spends relatively little time in this mode.

We performed 4 experiments allowing an input shift of ±1, with a hidden layer ranging from 60 hidden units to 90 hidden units. The best performance in learning and generalization was obtained using the network with 80 hidden units. The results are plotted on a graph representing the performance of the network on the learning set in the forefront of each graph and the test set in the background. We performed two more sets of 4 experiments allowing for an input shift of ±2, and ±3 respectively. The best overall performance was obtained with an input shift of ±2 and 80 hidden layers. The graphs are presented at the end of this paper.

BigNet performed well in acquisition of this new, homologous training set of larger proteins using roughly the same number of hidden units required for our previous heterologous training set of smaller proteins (Wilcox et al, 1991). Associations between protein sequence and structure in the training set were learned to better than 99% precision within 100 iterations. Our research indicates that for 8 families of small (<140 amino acids) proteins for which 4 members were included in the training set, testing with the sequence from the remaining test set of 4 proteins recalled the correct structure to 93% accuracy.

Acknowledgements

The authors gratefully acknowledge the expert technical assistance of Ken Chin-Purcel and Bill King of the Minnesota Supercomputer Center, Inc. and of Joseph Habermann of the Minnesota Supercomputer Institute (MSI) for their help in development of graphic display programs.

References

Bohr, H., Bohr, J., Brunak, S., Cotterill, R.M., Fredholm, H., Lautrup, B., Petersen, S.B. "A novel approach to prediction of 3-dimensional structures of protein backbones by neural networks". *FEBS Lett*, 1990 261:43-46.

Chou, P. and Fasman, G.D. "Prediction of the secondary structure of proteins from their amino acid seqence", *Adv. Enz.*, 1978, 47: 45-147.

Eisenberg, D.; Schwarz, E.; Komaromy, M.; Wall, R. "Analysis of membrane and surface protein sequences with the hydrophobicity moment plot", *J. Mol. Biol.*, 1984, 179:125-142.

Holley, L.H.; Karplus, M. "Protein structure prediction with a neural network". *Proc. Nat. Acad. Sci. USA*, 1989, 86: 152-156.

Levitt, M.; Warshel, A. "A computer simulation of protein folding". *Nature* (London), 1975, 253: 694-698.

Liebman, M.N. "Molecular modeling of protein structure and function: a bioinformatic approach". *J. Comput. Aided Mol. Des.*, 1987, 1: 323-341.

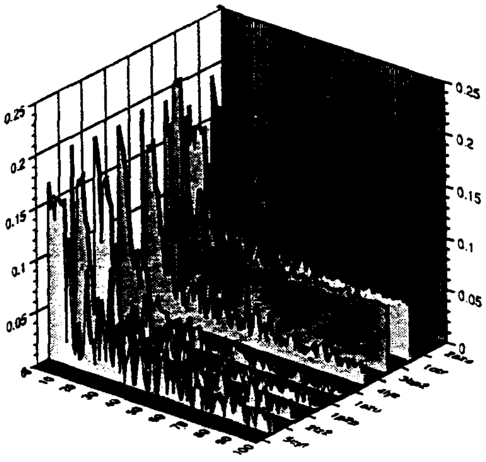
Liebman, M.N.; Venanzi, C.A.; Weinstein H. "Structural analysis of carboxypeptidase A and its complexes with inhibitors as a basis for modeling enzyme recognition and specificity". *Biopolymers*, 1985, 24:1721-58.

Qian, N.; Sejnowski, T.J. "Predicting the secondary structure of globular proteins using neural network models". *J. Mol. Biol.*, 1988, 202: 865-884.

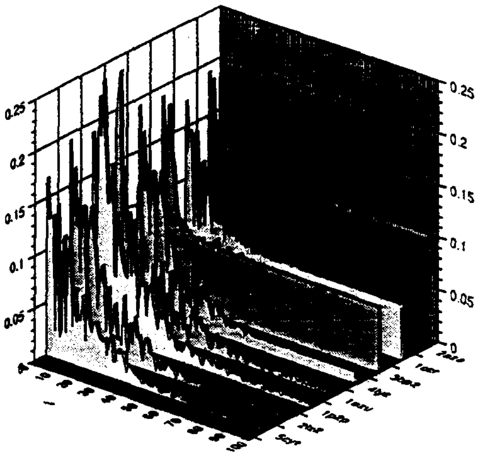
Rummelhart, D.E.; Hinton, G.E.; Williams R.J. "Learning representations by error propagation". In *Parallel Distributed Processing*, vol. 1, 1986, pp. 318-362, MIT Press, Cambridge, MA.

Tanaka, S.; Scheraga, H.A. (1976) "Medium- and long-range interaction parameters between amino acids for predicting 3D structures of proteins". *Macromolecules*, 1976, 9: 945-950.

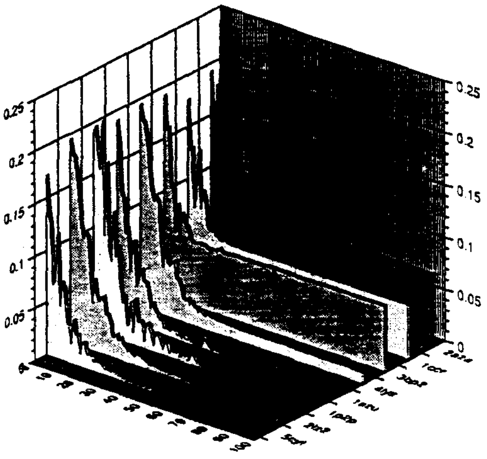
Wilcox, GL, Poliac, MO and Liebman, MN (1991 in press) "Neural Network Analysis of Protein Tertiary Structure", Tetrahedron Computer Methodology.



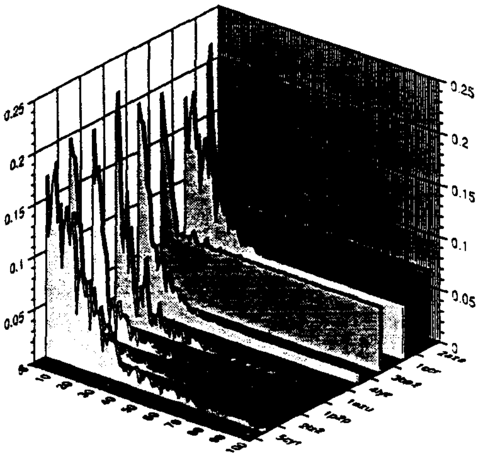
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 1, 60 hidden units



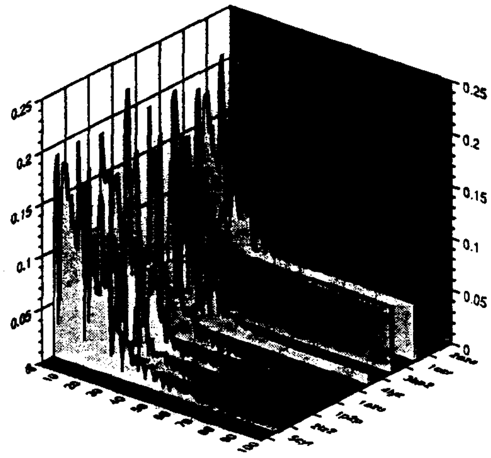
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 1, 70 hidden units



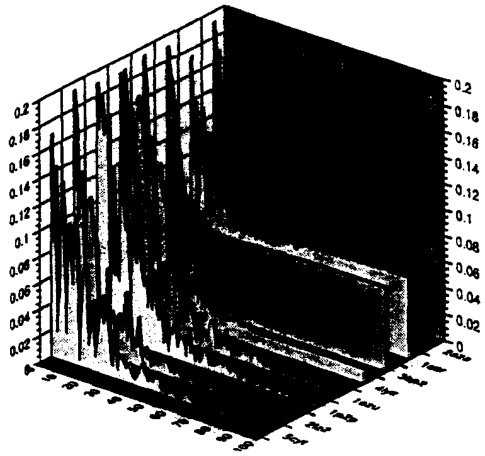
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 1, 80 hidden units



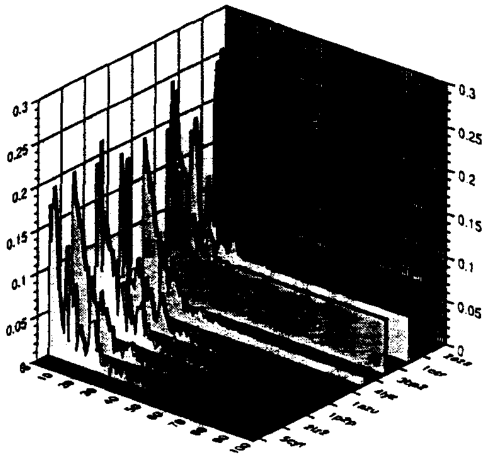
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 1, 90 hidden units



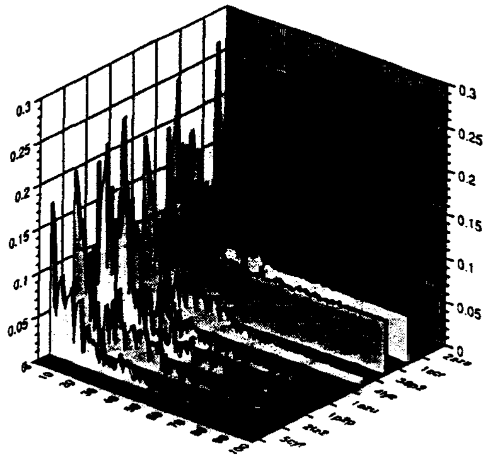
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 2, 60 hidden units



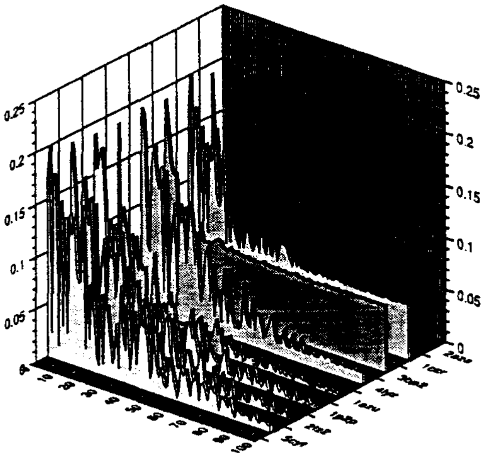
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 2, 70 hidden units



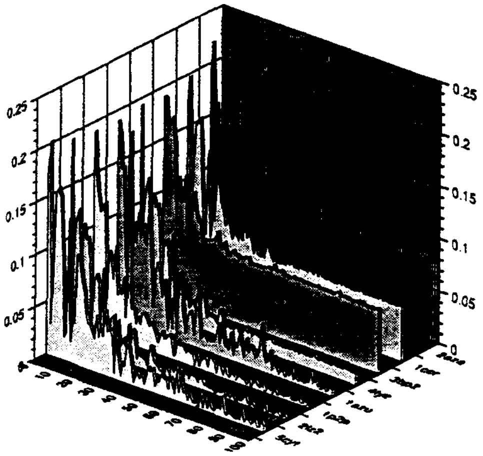
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 2, 80 hidden units



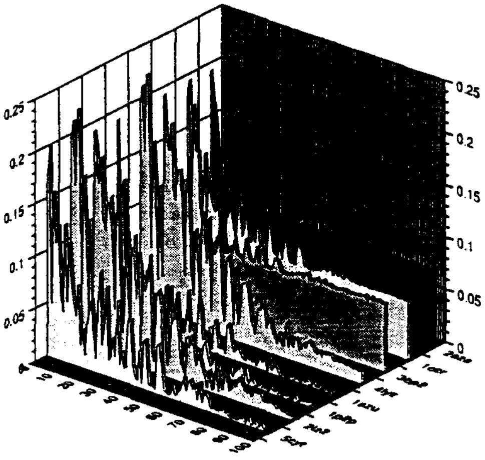
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 2, 90 hidden units



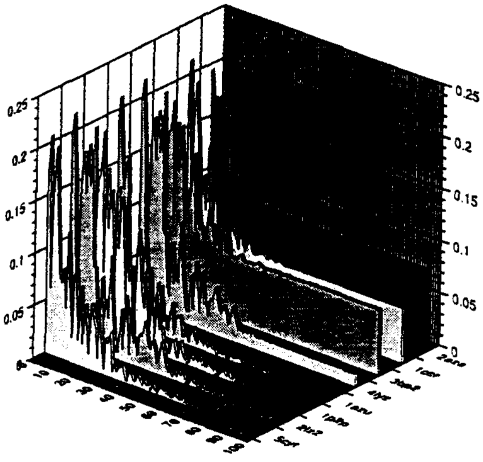
STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 3, 60 hidden units



STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 3, 70 hidden units



STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 3, 80 hidden units



STANDARD DEVIATION OF LEARNING PROGRESSION
shifted by 3, 90 hidden units