SAR and QSAR in Environmental Research, Vol. 1, pp. 153–159
Reprints available directly from the publisher
Photocopying permitted by license only

PROTEIN SECONDARY STRUCTURE PREDICTION WITH PARTIALLY RECURRENT NEURAL NETWORKS

M. RECZKO

Department of Molecular Biophysics, German Cancer Research Center, Im Neuenheimer Feld 280, W-6900 Heidelberg (Germany).

(Received June 10, 1992; in final form December 20, 1992)

Partially recurrent neural networks with different topologies are applied for secondary structure prediction of proteins. The state of some activations in the network is available after a pattern presentation via feedback connections as additional input during the processing of the next pattern in a sequence. A reference data set containing 91 proteins in the training set and 15 non-homologous proteins in the test set is used for training and testing a network with a modified, hierarchical Elman architecture. The network predicts the secondary structures α-helix, β-sheet, and "coil" for each amino acid. The percentage of correctly classified amino acids is 67.83% on the training set and 63.98% on the test set. The best performance of a three-layer feedforward network is 62.7% on the same test set. A cascaded network, where the outputs of the recurrent network are processed by a second net with 13×3 inputs, four hidden and three output units has a predictive performance of 64.49%. The best corresponding feedforward net has a performance of 64.3%.

KEY WORDS: protein secondary structure prediction; recurrent neural networks; hierarchical architectures; incremental training.

1 INTRODUCTION

The determination of three-dimensional protein structures using crystallographic or NMR methods is very time-consuming compared to the determination of protein sequences. Since the structure of a protein depends mostly on its sequence, great efforts are taken to develop methods for predicting structural information using only the protein sequence.[1] The secondary structure of a protein identifies unique local structures comprised of small groups of residues. The most important motifs are α-helix and β-sheet. All other structures are denoted as coil. If the three-dimensional structure of a protein is known, the local secondary structures are defined using the classification of Kabsch and Sander.[2] There are different approaches for secondary structure prediction[3] using statistical,[4] information theory[5] or neural network[6–8] methods. The best performance of these methods is roughly 65% for a three-state classification. Most of the neural networks studied for this task have a hierarchical feedforward topology. The connectivity graph of these networks contains no loop so that the network output for a pattern is independent of the order in which previous patterns have been presented.

Presented at the First International Workshop on Neural Networks Applied to Chemistry and Environmental Sciences, June 8–10, 1992, Lyon, France.

This paper shows how *partially recurrent networks* can be applied for protein secondary structure prediction. Most of the connections in these networks are feedforward and only a few, fixed feedback connections are added. The recurrency provides some additional information in the form of activity patterns about previous pattern presentations for the processing of the currently presented pattern. All units in the network are updated synchronously with each new input pattern. Since only the feedforward connections are modifiable, these networks are trained using the conventional backpropagation algorithm[9] as an effective approximately correct method, neglecting the error propagation along the recurrent connections. The first architecture of this kind, in which the previous output activations after each pattern are fed back to serve as additional input units was defined by Jordan[10] and has been studied by Qian and Sejnowski[6] for protein secondary structure prediction. This form of recurrency does not improve the predictive performance since there is only a very small amount of information about the previous pattern in the three-output activities.

A computationally more powerful architecture was described by Elman,[11] where an additional set of *context units* held a copy of the activations from the previous time step of the units in the first hidden layer. It was shown by Cleeremans *et al.*[12] that this *Elman*-architecture can learn pattern sequences recognized by finite state automatons. A network having several hidden layers with associated context units[13] leads to a hierarchical processing system for temporal task decomposition similar to the systems described by Albus.[14] The context units may have an additional self-connection which gives some individual memory to each context unit to accumulate a trace of the previously hidden activations. A partially recurrent, hierarchical architecture is used in this application.

2 METHODS

2.1 Database

The secondary structure assignments based on the coordinates of 106 proteins from the Brookhaven Protein Data Bank using the method of Kabsch and Sander[2] were prepared by Qian and Sejnowski.[6] This data is publicly available for benchmarking neural networks and is used here. The secondary structures are grouped into the classes α-helix, β-sheet and coil. The data set is split into a training set containing 18 105 residues from 91 proteins and a non-homologous test set containing 3620 residues from the remaining 15 proteins.

2.2 Performance Measures

The most common measure of predictive performance is the percentage of correct predictions for all residues:

$$Q_3 = \frac{P_\alpha + P_\beta + P_{\text{coil}}}{N} \tag{1}$$

where N is the total number of residues and P_i the number of correctly predicted residues of class i.

2.3 *Network Architecture*

The topology of the recurrent network is shown in Figure 1. The network consists of four layers of processing units. The input layer contains $13 \times 20 = 260$ units, since at each time step t, a subsequence $P(t)$ of 13 amino acids from the protein sequence is represented in this layer using a local encoding for each amino acid. In this representation, each amino acid is translated into an activity pattern where exactly one input unit corresponding to that amino acid has an output activation of 1.0 and all other input units have an output activation of 0. In this way the subsequence $P(t)$ of 13 amino acids is mapped onto a sequence of activity patterns $I_P(t)$ in the input window. After each time step the amino acid sequence is shifted one position to the left in the input window. The network predicts the secondary structure to which the amino acid in the center of the input window belongs. To predict the first and last six residues of a sequence, each chain is padded with end markers at the head and tail. These end markers translate into an activity pattern containing only zeros at the corresponding positions in the input window.

The secondary structures are also encoded with a local code where each struc-

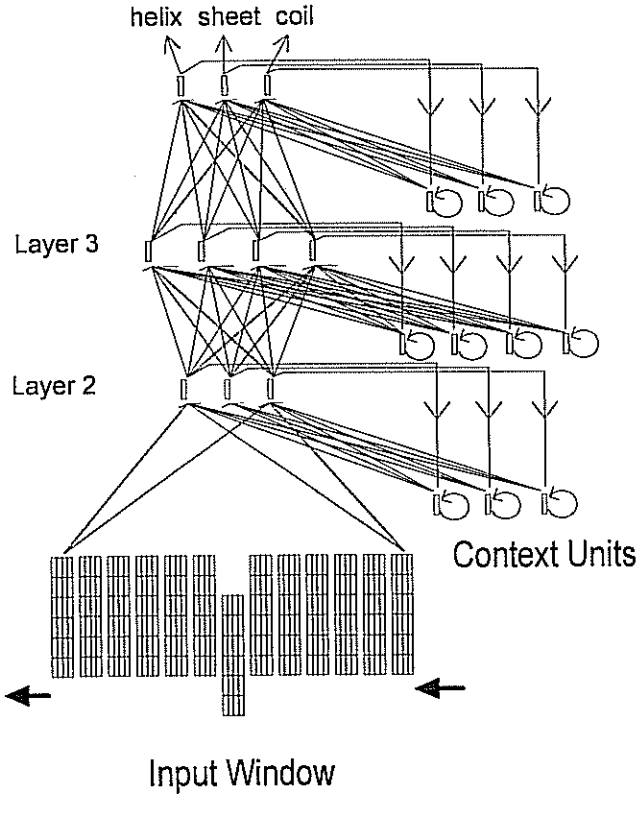

Figure 1 The hierarchical, partially recurrent network. The primary sequence is shifted from right to left through the input window. The output activities indicate the secondary structure for the amino acid in the middle of the input window. Each non-input unit has a feedback connection to its associated context unit.

ture class j has a corresponding output unit j that should have an activation of 1.0 only when the class j occurs. For a prediction the class j that has an output unit with the highest activity $o_j(t)$ compared to the other output units after presentation of an input pattern is assigned to the residue in the center of the input window. Each unit j in the network is fully connected to the previous layer where each connection from unit i to unit j has an associated weight w_{ij}. For each unit a network input net_j is calculated using

$$net_j(t) = \sum_i w_{ij} o_i(t) \tag{2}$$

The network net_j input is nonlinearly transformed into output activities $o_j(t)$ using

$$o_j(t) = \frac{1}{1 + \exp\left[net_j(t) - \theta_j\right]} \tag{3}$$

where θ_j is an activation threshold for each unit j. The context units calculate their output values $C_j(t)$ differently. Each context unit j has one single feedback connection coming from an associated unit i in the layer directly above the context unit. This connection has a fixed weight $w_{ij} = 1$ and is only used to transfer the activation $o_i(t-1)$ from the previous pattern presentation. The output values of the context units depend linearly on the associated hidden units:

$$C_j(t) = o_i(t-1) + w_j C_j(t-1) \tag{4}$$

The w_j in Eq. (4) acts as a self-connection from each context unit to itself and is restricted by $0<w_j<1$. With this self-feedback the context unit j accumulates a weighted moving average of activities of its corresponding unit i

$$C_j(t) = o_i(t-1) + w_j o_i(t-2) + w_j^2 o_i(t-3) + \ldots \tag{5}$$

In this way the context units trace the activities of their corresponding units back into the past.[15]

An update cycle for a new input window configuration consists of a copy phase, where all context units transfer the activities on the feedback connections using Eq. (4) and a normal feedforward phase, where all other units calculate their activities using Eq. (3) in an update order from layer 2 to the output layer.

The weights, thresholds and self-connections in the network are adapted using the backpropagation algorithm.[9] Each pattern in the training set is propagated through the network and an error information is generated by comparison between the output layer activities and the desired target values for these activities. This error information is propagated only along the feedforward connections.

3 RESULTS

The network is built and trained in an incremental manner. This is motivated by the ability of larger networks to overgeneralize on the training set. A network with

a large number of weights is able to memorize features from the training set which are too specialized for a good generalization to unseen patterns in a test set. Therefore the first network works like a "bottleneck" for irrelevant sequence features. A two-layer network with three output units and three context units is trained until convergence on the training set. The architecture is essentially like a Jordan network without hidden units.[10] This network classifies 62.16% of the residues in the test set correctly. The output of this network is then processed by an additional layer of three units. In this way an Elman network with three hidden and three context units is initialized with the weights of the two-layer Jordan net. This net has a predictive performance of 62.66%. The output layer of this network is then connected to three associated context units. The self-connections of these context units are initialized to 0.9 and an additional layer of three units processes the output activations. The self-connections are set to 0.9 to extend the weighted moving average of the former output units back on the sequence so that these context units may capture larger structural details. After these modifications, the network has

- three units in the first hidden layer with three associated context units,
- three units in the second hidden layer with three associated context units and
- three units in the output layer.

The performance increases slightly to 62.72%. For the final architecture, an extra

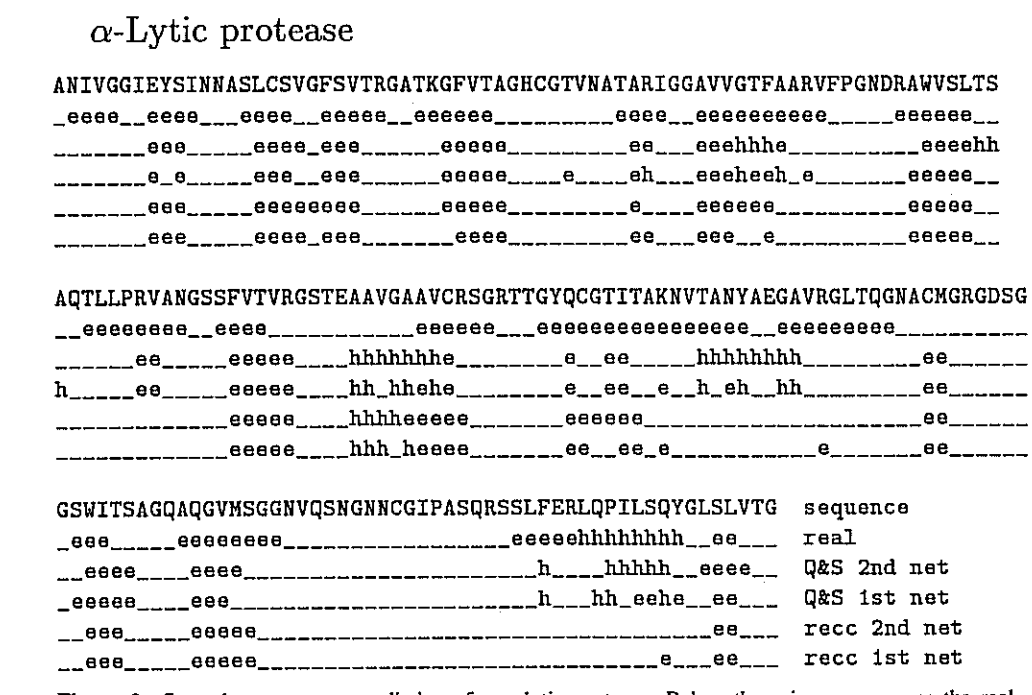

Figure 2 Secondary structure predictions for α-lytic protease. Below the primary sequence the real structure is listed, where h denotes helix, e denotes sheet (extended, and _ denotes coil. The next two lines show the predictions of the cascaded feedforward network and the three-layer feedforward network of Qian and Sejnowski,[6] respectively. The two lines at the bottom of each row show the predictions of the cascaded recurrent network and the recurrent network in Figure 1.

hidden unit and a corresponding context unit is added to the second hidden layer and the output layer is also connected to three new context units. This is the architecture in Figure 1 with a performance of 63.98% on the test set. This is slightly better than the performance of a three-layer feedforward net with 40 hidden units predicting 62.7% residues correctly.[6] In comparison, the recurrent network contains significantly fewer weights than the feedforward net.

A method used in similar applications[6,16,17] to improve the performance of the network is the addition of a second feedforward network that processes the sequence of output activations the same way as the first network processes the primary sequence. The input to the second network is a window with 13 consecutive predictions of the first network. The second network improves the prediction in the center of this window using the context around the prediction. A second network with 13×3 input units, four hidden and three output units resulted in a performance of 64.49% on the test set, which is an improvement over the corresponding cascaded feedforward architecture with 64.3% correct predictions.

For comparison with the predictions of a feedforward net, the predictions of the recurrent network and the cascaded recurrent network for two test proteins not contained in the training set are shown in Figures 2 and 3 together with the predictions of the feedforward and cascaded feedforward networks taken from Qian and Sejnowski.[6]

Both cascaded architectures are able to fill in gaps in larger secondary structures

Hemoglobin V (cyano, met)

```
PIVDTGSVAPLSAAEKTKIRSAWAPVYSTYETSGVDILVKFFTSTPAAQEFFPKFKGLTTADELKKSADVRW
___________hhhhhhhhhhhhhhhh_hhhhhhhhhhhhhhhh_____________hhhhh__hhhhh
___________hhhhhhhh_____ee_______eeeeee_______hh_______hhhhhhhhhhhhh
___________hhhhhh_______ee_______eeeee_________hh_________hhhhh____hhh
_____________hhhh_____________eeeeee_____________hhhhhhhhhhhh
_____________hhhhh____________eeeee_____________hhhh__hhhhh

HAERIINAVDDAVASMDDTEKMSMKLRNLSGKHAKSFQVDPEYFKVLAAVIADTVAAGDAGFEKL
hhhhhhhhhhhhhh____hhhhhhhhhhhhhhhh_______hhhhhhhhhhhh___hhhhhh
hhhhhhhh__hhhh_____hhhhhhhhhhhh_________hhhhhhhhhhhhh____hhhhhh
hhh_eee___hhh_____hhhhhhhhh__h__h___e___hhhhhhheh_hhh_____hhhhh
hhhhhhhh_hhhhh___hhhhhhhhhh___________hhhhhhhhhhhhhh____hhhhh
hhhhhhh__hhhhh___hhhhhhhhhh___________hhhhhhhhhhhhhh_____hhhh

MSMICILLRSAY  sequence
hhhhhhhhh___  real
hhhhhhhhh___  Q&S 2nd net
hhhheeehh___  Q&S 1st net
hhhhhhhh____  recc 2nd net
hhhhhhhh____  recc 1st net
```

Figure 3 Comparison of the predictions for hemoglobin V.

and to eliminate isolated false predictions. The recurrent net without cascade has apparently a lower tendency to make predictions of very short length than the simple feedforward net. This can be explained by the inertia of the activations of the context units associated to the second hidden and output layer.

The application of recurrent networks for sequential tasks like the process of protein folding seems very appropriate. The network studied here contains significantly fewer weights and has a slightly better performance than the best feedforward networks reported in literature for secondary structure prediction. This is an indication that recurrent networks can build more suitable models of some complicated processes in nature like the extremely complex folding of a protein.

References

1. Robson, B. and Garnier, J. (1986). *Introduction to Proteins and Protein Engineering*. Elsevier, Amsterdam, p. 230.
2. Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22,** 2577–2637.
3. Garnier, J. (1990). Protein structure prediction. *Biochimie* **72,** 513–524.
4. Chou, P. and Fasman, G. (1974). Prediction of protein conformation. *Biochemistry* **13,** 222–245.
5. Gibrat, J., Garnier, J., and Robson, B. (1987). Further development of protein secondary structure prediction using information theory. New parameters and consideration of residue pairs. *J. Mol. Biol.* **198,** 425–443.
6. Qian, N. and Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.* **202,** 865–884.
7. Holley, L. and Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proc. Natl. Acad. Sci. USA* **86,** 152–156.
8. Bohr, H., Bohr, J., Brunak, S., Cotterill, R., Lautrup, B., Norskov, L., Olsen, O., and Petersen, S. (1988). Protein secondary structure and homology by neural networks. *FEBS Letters* **241,** 223–228.
9. Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations* (D.E. Rumelhart and J.L. McClelland, Eds.). MIT Press, Cambridge, Massachusetts, p. 318.
10. Jordan, M. (1989). Serial order: A parallel, distributed processing approach. In, *Advances in Connectionist Theory: Speech* (J. Elman and D. Rumelhart, Eds.). Erlbaum, Hillsdale.
11. Elman, J. (1990). Finding structure in time. *Cognitive Science* **14,** 179–211.
12. Cleeremans, A., Servan-Schreiber, D., and McClelland, J. (1989). Finite state automata and simple recurrent networks. *Neural Computation* **1,** 372–381.
13. Bengio, Y., Cardin, R., and De Mori, R. (1990). Speaker independent speech recognition with neural networks and speech knowledge. In, *Advances in Neural Information Processing Systems II* (D. Touretzky, Ed.). Morgan Kaufmann, San Mateo, pp. 218–225.
14. Albus, J. (1991). Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics* **21,** 473–509.
15. Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, p. 180.
16. Kneller, D., Cohen, F., and Landridge, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network. *J. Mol. Biol.* **214,** 171–182.
17. Stolorz, P., Lapedes, A., and Xia, Y. (1992). Predicting protein secondary structure using neural net and statistical methods. *J. Mol. Biol.* **225,** 363–377.