Combining Neural Networks for Protein Secondary Structure Prediction

Søren Kamaric Riis
Electronics Institute, Building 349
Technical University of Denmark, 2800 Lyngby, Denmark
Email: riis@ei.dtu.dk

ABSTRACT

In the statistics and neural networks communities there has recently been an increasing interest in combining multiple experts for difficult classification problems. In this paper structured neural networks are applied to the problem of predicting the *secondary structure* of proteins. A hierarchical approach is used where specialized neural networks are designed for each structural class and then combined using another neural network. The submodels are designed by using *a priori knowledge* of the mapping between protein building blocks and the secondary structure and by using *weight sharing*. Since none of the individual networks have more than 600 adjustable weights over-fitting is avoided. When ensembles of specialized experts are combined the performance is better than most secondary structure prediction methods based on single sequences even though this model contains much fewer parameters.

1. Introduction

It is a common assumption in the statistics and neural networks communities that the use of multiple submodels can improve performance in difficult classification tasks. By using the 'divide and conquer' principle a set of specialized submodels can be combined in a hierarchical way to form the final model. The 'divide' step can be done either by purely probabilistic methods [10] or simply by designing separate models for each of the different classes to be recognized. Often some of the classes to be recognized are of very different nature, and designing specialized models for each of the classes instead of using only one general model can lead to better performance. The 'conquer' step can be based either on the 'Winner–Take–All' principle or the optimal class can be some function of the outputs of the individual experts. In this work the 'divide' step is performed by using different neural networks for each class and the 'conquer' step is carried out by combining the individual experts using another neural network. The method is illustrated on the problem of predicting the secondary structure of proteins.

The subunits of a protein are the so called polypeptide chains which fold in space to form the 3D-structure of the protein. Polypeptide chains are build from *amino acids* of which there are 20 naturally occuring. Normally the amino acids are specified by a unique one-letter code. The sequence of amino acids in a given protein is called the *primary structure* and it is believed that the 3D–structure of most proteins are defined by their primary structures. Prediction of the protein structure from the primary sequence of amino acids is a very challenging task, and the problem has been approached from several angles. A step on the way to a prediction of the full 3D structure is to predict the *local* conformation of the polypeptide chain, which is called the secondary structure. Most often the various secondary structures are grouped into the three main categories α-helix, β-strand and coil. A lot of interesting work has been done on predicting secondary structures, and over the last 10 to 20 years the methods have gradually improved in accuracy. This improvement is partly due to the increased number of reliable structures from which rules can be extracted and partly due to improvement of methods. Around 1988 the first attempts were made to use neural networks to predict protein secondary structure by Qian and Sejnowski [7]. In this work fully connected feed-forward networks with more than 10,000 adjustable weights were trained by the Backpropagation algorithm to predict the three secondary structures from the amino acid sequence. The accuracy of the predictions made by Qian and Sejnowski seemed better than those obtained by previous methods, although tests based on different protein sets are hard to compare. This work started a wave of applications of neural networks to the secondary structure prediction problem [2, 9], sometimes in combination with other methods [11, 5].

Our goal has been to get as good predictions as possible from single sequences, *ie*, only the amino acid sequence of the considered protein is used as input. This work had three stages. Firstly, individual networks were designed for prediction of the three structures. Due to the use of weight sharing these networks contain much fewer weights than fully connected networks and over-fitting is thereby avoided. Secondly, instead of using only one network for each type of structure, an ensemble of 5 networks were used for each structure. Thirdly, these ensembles of single structure networks were combined by another neural network to obtain a three state prediction. In the combining network the outputs were constrained to sum to one by using *Softmax* [1]. If only weight sharing is used to reduce the number of parameters a network with only 311 adjustable weights is found to give results comparable to Qian and Sejnowski's network containing more than 10,000 weights. However, using the hierarchical approach 66–67% of the amino acids are correctly classified which is 3–4% better than a fully connected network on the same dataset. A result of 71–72% correctly classified amino acids is obtained when multiple alignments of *homologous* proteins are used as input, see [8].

When using neural networks for secondary structure prediction the choice of protein database is complicated by potential homology (structural similarity) between proteins in the training and testing set. For evaluation of the methods presented below *seven-fold cross-validation* on the set of 126 non-homologous globular proteins from [9] is therefore used. The seven subsets are denoted set A–G and contain a total of 24,395 amino acids distributed on 32% α-helix, 28% β-strand and 47% coil. As measures of prediction accuracy the percentage of correctly classified amino acids is used: Q_3 is the three–state percentage and $Q_{2,i}$ is the two–state (single–structure) percentage for secondary structure $i = \alpha, \beta, c$. A complementary measure is the Matthews' correlation coefficients [6] for each of the three secondary structures; C_α, C_β and C_c. The correlation coefficients are 1.0 if the predictions are all correct, -1.0 if all the predictions are false and close to zero for random or trivial predictions.

2. Single Structure Prediction

2.1. Adaptive encoding of amino acids

As in most of the existing methods, the secondary structure of the j'th residue R_j is predicted from a window of amino acids, $R_{j-n}, ..., R_j, ..., R_{j+n}$ where $W = 2n+1$ is the window size. Usually the amino acids are encoded by 21 binary numbers, such that each number corresponds to one amino acid. The last number corresponds to a space, and is used to indicate the ends of a protein. This encoding, which we will call the *orthogonal encoding,* has the advantage of not introducing any artificial correlations between the amino acids, but it is highly redundant, since 21 symbols can be encoded in 5 bits. This redundancy is one of the reasons why networks for secondary structure prediction tend to have a very large number of weights. However, by using *weight sharing* [4] it is possible to let the network itself choose the best encoding of the amino acids. The starting point is the orthogonal encoding, but we omit the spacer input unit used by Qian and Sejnowski, and instead all inputs are set to zero for that part of the window where no residues are present. For each window position the 20 inputs are connected to M hidden units by $20 \times M$ weights. This set of weights (and the M thresholds) corresponding to one window position is identical to those used for all the other window positions, see Figure 1. More precisely, if the weight from input j to hidden unit i is called w_{ij}^k for the k'th window position, then $w_{ij}^k = w_{ij}^l$ for all k and l. These sets of weights are forced to stay identical during training; they always share the same values. In this way the encoding of the amino acids is the same for all positions in the window. The weights are learned by a straightforward generalization of back-propagation in which weight updates are summed for weights sharing the same value [4]. The use of weight sharing implies that the first layer only contains $21 \times M$ adjustable parameters including thresholds no matter the size of the window. In this work $M = 3$ is used and each of the 20 amino acids are thus represented by only three real numbers. This leads to a dramatic reduction of the almost 11,000 weights used in the first layer of Qian and Sejnowski's fully connected network, even if an extra hidden layer is added to the network. The adaptive encoding scheme of the amino acids we call *local encoding.* Since the encoding is learned along with the other weights in the network it will be the 'optimal' encoding, in the sense that it yields the minimum error on the training set for that specific network and that specific task.

2.2. Structured networks

Many existing prediction methods use the same model for predicting the three types of secondary structure (helix, strand, and coil). Since the three secondary structures are very different it is possible that performance could be enhanced if separate networks are specifically designed for each of the three structures.

The majority of the helices in the database used are so called α-helices. A residue in an α-helix is hydrogen bonded to the fourth residue above and the fourth residue below in the primary sequence,

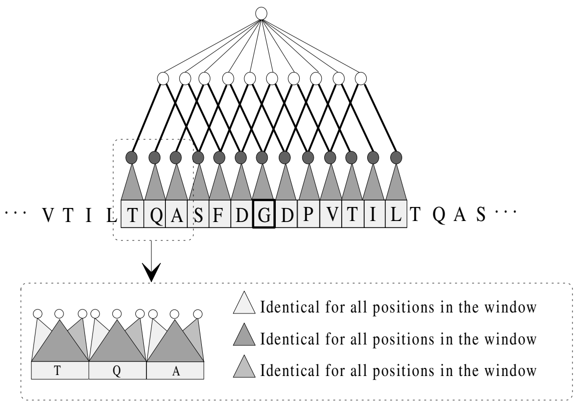

Fig. 1: Network for predicting helices. Grey circles symbolize three hidden units, emphasized lines three weights, shaded triangles 20 shared weights and shaded rectangles 20 input units.

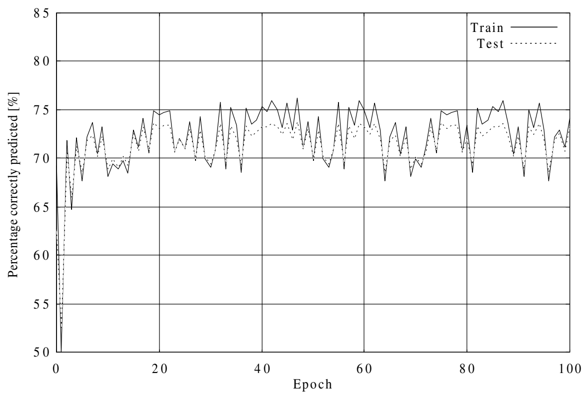

Fig. 2: Percentage ($Q_{2,\alpha}$) of residues predicted correctly by the α-network as a function of the number of training epochs (full sweeps through the training set). Training set B–G; testing set A.

	Reference	Structured
$Q_{2,\alpha}$	72.54	74.98
$Q_{2,\beta}$	73.84	76.48
$Q_{2,c}$	70.78	71.33
C_α	0.37	0.39
C_β	0.36	0.37
C_c	0.41	0.42

Table 1: Two-state predictions of α-helix, β-strand and coil found by seven-fold cross-validation.

and it takes 3.6 amino acids to make a turn in an α-helix. It is likely that this periodic structure is essential for the characterization of an α-helix. These characteristics are all of local nature and can therefore easily be built into a network that predicts helices from windows of the amino acid sequence. In Figure 1 a network with local encoding (in the first hidden layer), a built-in period of 3 residues and a window size of 13 residues is shown. The second hidden layer in the network contains 10 units that are fully connected to the output unit giving a total of 144 adjustable parameters. For comparison a standard network with no hidden units at all, orthogonal encoding, and a window length of 13 residues has 261 adjustable parameters.

In contrast to helices, β-strands and coil do not have such a locally described periodic structure. Therefore, the strand and coil networks only use the local encoding scheme, and a second hidden layer with 5-10 units fully connected to the first hidden layer as well as to the output layer. Early studies indicated that a window size of 15 residues was optimal for all three types of single-structure networks. Thus, a typical structured helix network contains 160 weights, while typical strand and coil networks contain 300-530 weights. As shown in Figure 1 the single-structure networks only have one output. If the output is larger than some decision threshold the prediction is α-helix, β-strand or coil depending on the type of structure under consideration. For an input/output interval of [0;1] a decision threshold of 0.5 was found to be optimal.

The performance of the constrained single-structure networks are compared with the predictions obtained from perceptrons with no hidden units having window lengths of 13 amino acids. The single-structure networks are all trained balanced, *ie*, for each positive example (helix) a negative example (non-helix) is chosen at random from the training set. In this way the same number of positive and negative examples are used in the training.

The result of training the structured α-network on set B–G and using set A as testing set is shown in Figure 2. This figure shows two interesting features: 1) Over-fitting is gone; 2) the training and testing percentages oscillate in phase. The first observation means that this network gives reliable estimates of prediction accuracy on new proteins not in the database used for developing the method. The observed fluctuations are mostly due to the use of balanced training where a different set of negative examples (non-helix) are used in each training epoch. Since the in-phase oscillations are observed for all of our networks, the final network weights are chosen as those corresponding to the minimal training error observed during 100 epochs of training (*ie* full sweeps through the training set).

In Table 1 the results obtained with the single-structure networks are summarized. From the table it is seen that the structured networks predict the three secondary structures better than the reference models, both in terms of $Q_{2,i}$ and the correlation coefficients. This shows that the learned representation of the amino acids is considerably better than the orthogonal representation.

3. Combining single–structure predictions

To combine the single–structure predictions a neural network is used. The network takes a window of 15 single–structure predictions of helix, strand and coil as input and is fully connected to the three outputs via 10 hidden units. In addition to combining the submodels this network acts like a filter, *ie* it has a tendency to eliminate unrealistic predictions and it results in more realistic average lengths of the predicted secondary structure segments, see [9]. However, this type of network does not necessarily choose one of the three structures. For instance it can (and sometimes do) classify one input pattern as all three types of structure, *ie*, it gives large outputs on all three output units. In practice of course, the input is classified as the structure giving the largest output, but conceptually this type of classification is more suited for independent classes. It may be beneficial to build in the constraint that a given input belongs to only one of the three structures. This can be done by *Softmax* [1], which ensures that the three outputs always sum to one. Hence, the outputs can be interpreted as the conditional probabilities that a given input belongs to each of the three classes.

In Table 2 is shown the results achieved when combining the individual submodels with the above described network. For comparison, the performance of a network identical to Qian and Sejnowski's with 40 hidden units is also shown. The performance of this network is evaluated on the same set of non-homologous proteins by seven-fold cross-validation, and it is seen that the fully connected network only obtains $Q_3 = 63.2\%$ compared to $Q_3 = 65.4\%$ obtained by combining the single-structure predictions. Note that the results obtained with the Qian and Sejnowski model is found by using the best performance on each of the seven testing sets [7], which over-estimates the performance. For all other networks the stop criterion based on in–phase oscillations of the training and testing error is used.

The effect of the local encoding scheme is illustrated by a three-state network, which uses the adaptive encoding of amino acids in the first layer and 5 hidden units in the second layer. This network has a window size of 15 residues leading to a total of only 311 adjustable weights compared to approximately 11,000 weights in Qian and Sejnowski's network. Despite this difference the local encoding network gives about the same Q_3 and better correlation coefficients, indicating that the amino acids are well described by only three real parameters, and that the fully connected networks are highly over-parameterized. It should be noted again that the performance of the fully connected network is overestimated since it corresponds to a minimum in the *testing error* whereas the performance for the network with local encoding corresponds to a minimum in the *training error*. In Table 2 is also shown the effect of applying the 'combining' network as a filter to the prediction from the three-state local encoding network.

	Q_3 (%)	C_α	C_β	C_c
Combined SSN	65.39	0.46	0.41	0.43
Ensemble SSN	66.27	0.48	0.41	0.44
Q & S network	63.16	0.40	0.35	0.41
LEN	63.10	0.42	0.36	0.41
LEN filtered	64.20	0.44	0.37	0.41

Table 2: Cross-validated three-state predictions obtained by various methods (SNN: single–structure networks, Q & S: Qian and Sejnowski and LEN: local encoding network).

3.1. Ensembles of single–structure networks

For complex classification tasks the use of ensembles can be thought of as a way of averaging out statistical fluctuations. Furthermore, the combination of two or more different solutions can in some cases contribute valuable information. This is especially true if the ensemble members disagree as discussed in [3]. One obvious way to make the ensemble members disagree is to use different networks and/or training methods. In this work ensembles of 5 different single-structure networks (for each type of secondary structure) are used. The networks all use the local encoding scheme and the differences are introduced by using various periods in the α-network and by using different numbers of hidden units. When combining ensembles of single–structure networks an increase of 0.9% in the percentage of correctly classified amino acids is obtained as shown in Table 2.

4. Conclusion

The use of specialized submodels for protein secondary structure prediction has been investigated. By using ensembles of specialized neural networks for predicting each of the three secondary structures over-fitting was avoided and a consistent stop criterion based on in-phase fluctuation of the training and testing error was developed. The hierarchical approach gave a cross validated accuracy of 66.3% which is as good as or even better than results obtained by most other prediction methods based on single-sequences as input. By applying the method to multiple alignments of homologous proteins the performance is increased to 71–72% accuracy [8] which is comparable to 'state–of–the–art' methods [9] using about 5–10 times as many parameters.

One of the features of the single-structure networks were an adaptive encoding of the amino acids, in which each of the 20 amino acid were represented by three real numbers. This alone decreases the number of network weights tremendously as compared to fully connected networks. The effect of this method was illustrated by a network for three state prediction containing only 311 adjustable weights, which outperforms a standard fully connected network with more than 10,000 weights. The low number of weights in the single-sequence networks indicates that the implemented mapping from a window of the amino acid sequence to the secondary structure is relatively simple.

Acknowledgments

Numerous discussions with Anders Krogh are greatfully acknowledged. I would also like to thank Burchard Rost for supplying me with details of his own work, as well as helpful comments.

References

[1] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Neural Information Processing Systems 2*, (D. Touretzky, ed.), (San Mateo, CA), pp. 211–217, Morgan Kaufmann, 1990.

[2] D. Kneller, F. Cohen, and R. Langridge, "Improvements in protein secondary structure prediction by an enhanced neural network.," *Journal of Molecular Biology*, vol. 214, pp. 171–82, Jul 5 1990.

[3] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation and active learning," in *Advances in Neural Information Processing Systems 7*, (D. Touretzky, G. Tesauro and T. Leen, eds.), (Cambridge, MA), MIT Press, 1995. To appear.

[4] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[5] R. Maclin and J. Shavlik, "Using knowledge-based neural networks to improve algorithms: Refining the Chou-Fasman algorithm for protein folding," *Machine Learning*, vol. 11, pp. 195–215, 1993.

[6] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta*, vol. 405, pp. 442–451, 1975.

[7] N. Qian and T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models.," *Journal of Molecular Biology*, vol. 202, pp. 865–84, Aug 20 1988.

[8] S. Riis and A. Krogh, "Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments," 1995. NORDITA preprint 95/34 S.

[9] B. Rost and C. Sander, "Combining evolutionary information and neural networks to predict protein secondary structure.," *Proteins*, vol. 19, pp. 55–72, 1994.

[10] S. Waterhouse and A. Robinson, "Classification using heirarchical mixtures of experts," in *Proc. 1994 IEEE Workshop on Neural Networks for Signal Processing IV*, (J. Hwang, J. Vlontzos and E. Wilson, eds.), (Piscataway, New Jersey), pp. 177–186, 1994.

[11] X. Zhang, J. Mesirov, and D. Waltz, "Hybrid system for protein secondary structure prediction.," *Journal of Molecular Biology*, vol. 225, pp. 1049–63, Jun 20 1992.