Neural Networks For Molecular Sequence Classification

Cathy Wu[1,2], Michael Berry[3], Yuk-Shing Fung[2] and Jerry McLarty[1]

[1]Department of Epidemiology/Biomathematics
The University of Texas Health Center at Tyler, Tyler, Texas 75710
wu%jason.decnet@relay.the.net

[2]Department of Computer Science
The University of Texas at Tyler, Tyler, Texas 75799

[3]Department of Computer Science
University of Tennessee, Knoxville, Tennessee 37996-1301

Abstract

A neural network classification method has been developed as an alternative approach to the search/ organization problem of large molecular databases. Two artificial neural systems have been implemented on a Cray supercomputer for rapid protein/nucleic acid sequence classifications. The neural networks used are three-layered, feed-forward networks that employ back-propagation learning algorithm. The molecular sequences are encoded into neural input vectors by applying an n-gram hashing method or a SVD (singular value decomposition) method. Once trained with known sequences in the molecular databases, the neural system becomes an associative memory capable of classifying unknown sequences based on the class information embedded in its neural interconnections. The protein system, which classifies proteins into PIR (Protein Identification Resource) superfamilies, showed a 82% to a close to 100% sensitivity at a speed that is about an order of magnitude faster than other search methods. The pilot nucleic acid system, which classifies ribosomal RNA sequences according to phylogenetic groups, has achieved a 100% classification accuracy. The system could be used to reduce the database search time and help organize the molecular sequence databases. The tool is generally applicable to any databases that are organized according to family relationships.

Introduction

One major challenge in contemporary molecular biology is the analysis and management of the vast amount of sequence data. Currently, a database search for sequence similarities represents the most direct computational approach to decipher the codes connecting molecular sequences with protein structure and function. There exist good algorithms and mature software for database search and sequence analysis (see von Heijne 1991 and Gribskov & Devereux 1991 for recent reviews). However, the accelerating growth of sequence data has made the database search computationally intensive and ever more forbidding. It is, therefore, desirable to develop methods whose search time is not constrained by the database size. A classification/clustering method can be used as an alternative approach to the large database search/ organization problem with several advantages: (1) speed, because the search time grows linearly with the number of families, instead of the number of sequence entries; (2) sensitivity, because the search is based on information of a homologous family, instead of any sequence alone; and (3) automated family assignment. We have developed a new method for sequence classification using back-propagation neural networks (Wu et al. 1992; 1993). In addition, two other sequence classification methods have been devised. One uses a multivariant statistical technique (van Heel 1991), the other uses a binary similarity comparison followed by an unsupervised learning procedure (Harris, Hunter, & States 1992). All three classification methods are very fast, thus, applicable to the large sequence databases. The major difference between these approaches is that the classification neural network is based on "supervised" learning, whereas the other two are "unsupervised". The supervised learning can be performed using training set compiled from any existing second generation database and used to classify new sequences into the database according to the predefined organization scheme of the database. The unsupervised system, on the other hand, defines own family clusters and can be used to generate new second generation databases.

As an artificial intelligence and computational technique, neural network technology has been applied to many studies involving the sequence data analysis (see Hirst & Sternberg 1992 for a recent review). Back-propagation networks have been used to predict protein secondary structure (Qian & Sejnowski 1988; Holley & Karplus 1989; Kneller, Cohen, & Langridge 1990) and tertiary structure (Bohr et al. 1990; Chen 1993; Liebman 1993; Wilcox et al. 1993), to distinguish ribosomal binding sites from non-binding sites (Stormo et al. 1982) and encoding regions from non-coding sequences (Lapedes et al. 1990; Uberbacher & Mural 1991), and to predict bacterial promoter sequences (Demeler & Zhou 1991; O'Neill 1992; Horton & Kanehisa 1992). This paper updates the progress made in our neural classification systems, introduces a new sequence encoding schema, and discusses system applications.

System Design

The neural network system was designed to embed class information from molecular databases and used as an associative memory to classify unknown sequences (Figure 1). There are three major design issues: (1) the input/output mapping, (2) the neural network architecture, and (3) the sequence encoding schema.

Input/output mapping

The neural system is designed to classify new (unknown) sequences into predefined (known) classes. In other words, it would map molecular sequences (input) into sequence classes (output). Two neural systems have been developed using "second generation" molecular databases as the training sets. These databases are termed second generation because they are organized according to biological principles, or more specifically, within these databases, the sequence entries are grouped into classes based on sequence similarities or other properties. The first neural system, the Protein Classification Artificial Neural System (ProCANS), is trained with the annotated PIR database (Barker et al. 1992) and classifies protein sequences into PIR superfamilies. The second system, a pilot Nucleic Acid Classification Artificial Neural System (NACANS), is trained with the Ribosomal RNA Database Project (RDP) database (Olsen et al. 1992), and maps ribosomal RNA sequences into phylogenetic classes (Woese 1987).

Neural network architecture

The neural networks used are three-layered, feed-forward networks (Figure 1) that employ back-propagation learning algorithm (Rumelhart & McClelland 1986) (see

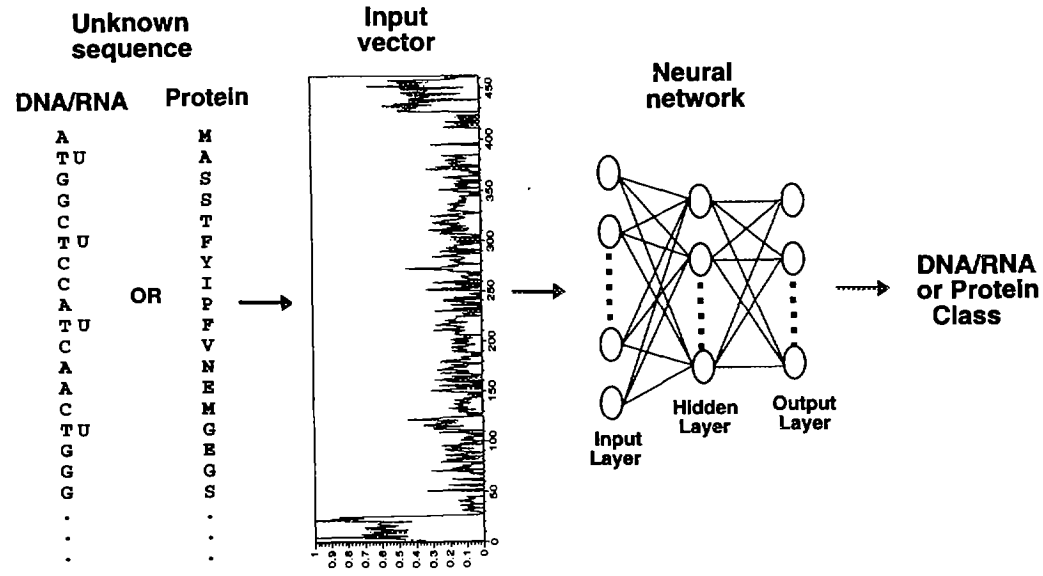

Figure 1. A neural network system for molecular sequence classification. The molecular sequences are first converted by a sequence encoding schema into neural net input vectors. The neural networks then classifies them into predefined classes according to sequence information embedded in the neural interconnections after network training.

Wu et al. 1992 for a detail description of the neural network model). In the three-layered architecture, the input layer is used to represent sequence data, the hidden layer to capture information in non-linear parameters, and the output layer to represent sequence classes. The size of the input layer is dictated by the sequence encoding schema chosen, the output layer size is determined by the number of classes represented in the network, whereas the hidden size is determined heuristically, usually a number between input and output sizes. The networks are trained using weight matrices initialized with random weights ranging from -0.3 to 0.3. Other network parameters included the learning factor of 0.3, momentum term of 0.2, a constant bias term of -1.0, and training epochs (iterations) of 400.

Sequence encoding schema

The sequence encoding schema is used to convert molecular sequences (character strings) into input vectors (numbers) of the neural network classifier (Figure 1). An ideal encoding scheme should satisfy the basic coding assumption so that encodings of similar sequences are represented by 'close' vectors. There are two different approaches for the sequence encoding. Once can either use the sequence data directly, as in most neural network applications of molecular sequence analysis, or use the sequence data indirectly, as in Uberbacher and Mural (1991). Where sequence data is encoded directly, most studies (e.g., Qian& Sejnowski 1988; Lapedes et al. 1990) use an indicator vector to represent each molecular residue in the sequence string. That is, a vector of 20 input units (among which 19 have a value of zero, and one has a value of one) to represent an amino acid, and a vector of four units (three are zeroes and one is one) for a nucleotide. This representation, however, is not suitable for sequence classifications where long and varied-length sequences are to be compared.

N-gram method. We have been using a n-gram hashing function (Cherkassky & Vassilas 1989) that extracts and counts the occurrences of patterns of n consecutive residues (i.e., a sliding window of size n) from a sequence string. The counts of the n-gram patterns are then converted into real-valued input vectors for the neural network. The size of the input vector for each n-gram extraction is m^n, where m is the size of the alphabet. (See Wu et al. 1992 or Wu 1993 for a detail description and schematic representation). The n-gram method has several advantages: (1) it maps sequences of different lengths into input vectors of the same length; (2) it provides certain representation invariance with respect to residue insertion and deletion; and (3) it is independent from the a priori recognition of certain specific patterns.

The original sequence string is represented by different alphabet sets in the encoding. The alphabet sets used for protein sequences include the 20-letter amino acids and the six-letter exchange groups derived from the PAM matrix. The alphabets for nucleic acid sequences include the four-letter AT(U)GC, the two-letter RY for purine and pyrimidine, and the two-letter SW for strong and weak hydrogen binding.

Twenty five n-gram encodings were tested for ProCANS, among which ae12 encoding was the best (Wu et al. 1992). The input vector for ae12 encoding is concatenated from vectors representing four separate n-grams, namely, a1 (monograms of amino acids), e1 (monograms of exchange groups), a2 (bigrams of amino acids), and e2 (bigrams of exchange groups). The vector has 462 units, which is the sum of the size of the four vectors (i.e., 20 + 6 + 400 + 36). An example of one such input vector is shown in Figure 1.

The major drawback of the n-gram method is that the size of the input vector tends to be large. This indicates that the size of the weight matrix (i.e., the number of neural interconnections) would also be large because the weight matrix size equals to n, where n = input size x hidden size + hidden size x output size. This prohibits the use of even larger n-gram sizes, e.g., the trigrams of amino acids would require 20^3 or 8000 input units. Furthermore, accepted statistical techniques and current trends in neural networks favor minimal architecture (with fewer neurons and interconnections) for its better generalization capability (Le Cun et al. 1989). To address this problem, we have attempted different approaches to reduce the size of n-gram vectors.

SVD (Singular-Value Decomposition) method. Recently, we have developed an alternative sequence encoding method by adopting the Latent Semantic Indexing (LSI) analysis (Deerwester et al. 1990) used in the field of information retrieval and information filtering. The LSI approach is to take advantage of implicit high-order structure in the association of terms with documents in order to improve the detection of relevant documents on the bases of terms used. The particular technique used is SVD, in which a large "term-by-document" matrix is decomposed into a set of factors from which the original

matrix can be approximated by linear combination (Figure 2).

In the present study, the term-by-document matrix is replaced by the "term-by-sequence" matrix, where "terms" are the n-grams. For example, a 8000 by 894 matrix can be used to represent the term vectors of 894 protein sequences, with each term vector containing the 8000 trigrams of amino acids. The large sparse term-by-sequence matrix would be decomposed into singular triplets, i.e., the singular (s) values, and the left and right singular vectors (Figure 2). The right s-vectors corresponding to the k-largest s-values are then used as the input vectors for neural networks. In this example, if the right s-vectors corresponding to the 100-largest s-values are used, then the size of the input vector would be reduced from 8000 to 100.

System Implementation

The system software has three components: a preprocessor to create from input sequences the training and prediction patterns, a neural network program to train and classify patterns, and a postprocessor to summarize classification results. All programs have been implemented on the Cray Y-MP8/864 supercomputer of the Center for High Performance Computing of the University of Texas System.

The preprocessor has two programs, one for the n-gram extraction, the other for the SVD computation. The n-gram program converts sequence strings into real-valued n-gram vectors that are scaled between 0 and 1. In the n-gram encoding, these vectors are directly used as neural network input vectors. In the SVD encoding, the n-gram vectors are further reduced into right singular vectors using a SVD program. The program, which is developed by Michael Berry, one of the co-authors, employs a single-vector Lanczos method (Berry, 1992). The right s-vectors are then processed before input into the neural network such that the component values are scaled between 0 and 1.

ProCANS Performance

N-gram encoding studies

In ProCANS, a modular neural network architecture that involves multiple independent network modules is used to embed the large PIR database (please see Wu, 1993 for a discussion of the modular network architecture). The current system has four modules developed with seven protein functional groups, consisting of 690 superfamilies and 2724 entries of the annotated PIR database. These include the electron transfer proteins and the six enzyme groups (oxidoreductases, transferases, hydrolases, lyases,

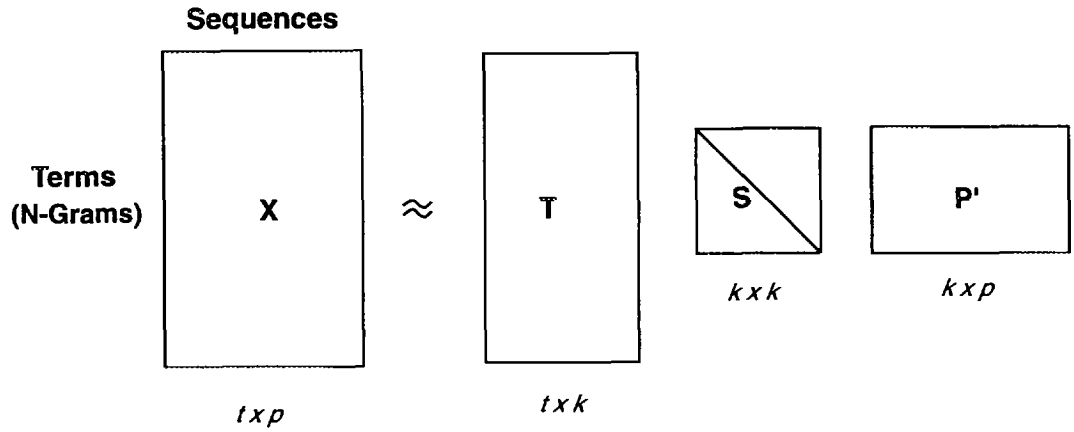

Figure 2. The singular-value decomposition (SVD) of a "term-by-sequence" matrix. The original term-by-sequence matrix (X) is approximated using the k-largest singular values and their corresponding singular vectors. S is the diagonal matrix of singular values. T and P, both have orthogonal, unit-length columns, are the matrices of the left and right singular vectors, respectively. t and p are the numbers of rows and columns of X, respectively. m is the rank of X ($m \leq \min(t,d)$), whereas k is the chosen number of dimensions in the reduced model ($k \leq m$).

Table 1. Comparisons of n-gram and SVD sequence encoding methods for ProCANS.

Encoding Method	Network Configuration	Number of Connections	Trained Patterns(%)	Predictive Accuracy(%)
N-gram (ae12)*	462 x 200 x 164	125,200	96.05	94.89
SVD (ae123)	50 x 100 x 164	21,400	97.57	94.47
SVD (a23)	100 x 100 x 164	26,400	98.63	94.89
N-gram + SVD	512 x 100 x 164	67,600	97.88	97.02

*The code used in parenthesis represents the type of n-gram vectors (please see text).

isomerases and ligases). The configuration of the four networks is 462 input units (derived from the ae12 n-gram encoding), 200 hidden units and 164, 180, 192, and 154 output units, respectively. During the training phase, each network module is trained separately using the sequences of known superfamilies (i.e., training patterns). During the prediction phase, the unknown sequences (prediction patterns) are classified on all modules with classification results combined. The classification score ranges from 1.0 for perfect match to 0.0 for no match. A protein entry is considered to be accurately classified if one of the five best-fits (the superfamilies with five highest scores) matches the target value (the known superfamily number of the entry) with a threshold (the cut-off classification score) of 0.01.

Two data sets are used to evaluate the system performance. The first data set divides the 2724 PIR1 (containing annotated and classified entries) entries into disjoined training and prediction sets. The prediction patterns are every third entries from superfamilies with more than two entries. The second data set uses all 2724 PIR1 entries for training, and 482 PIR2 (containing unclassified entries) entries for prediction. The predictive accuracy is 91.7% and 81.6%, respectively, for the two data sets. A detail analysis of the misclassified sequence patterns reveals three important factors affecting classification accuracy: the size of the superfamily, the sequence length, and the degree of similarity (see Wu 1993 for detail discussion). It is observed that the superfamily size is inversely correlated with the misclassification rate. And, generally, a sequence can be correctly classified if its length is at least 20% of the original length, although some sequences as short as 10% are classified. The main reason that the second prediction set has a lower accuracy is due to the large number of sequences belonging to single-membered or double-membered superfamilies, and sequences of small fragments. In other studies that involve only large superfamilies, the predictive accuracy has approached 100% (Wu et al., 1993). Therefore, the classification accuracy of ProCANS is expected to increase with the continuing accumulation of sequence entries available for training.

SVD encoding studies

Preliminary studies have been conducted for the SVD encoding method and compared with the ae12 n-gram encoding (Table 1). The data set used has 894 PIR1 proteins classified into 164 superfamilies. Among the sequences, 659 are used for training, and the remaining 235 for prediction. The SVD results of two n-gram vectors are shown, one for a23 (concatenates a2 and a3, the bigrams and trigrams of amino acids), one for ae123 (concatenates a1, a2, a3, the monograms, bigrams and trigrams of amino acids, and e1, e2, e3, the monograms, bigrams and trigrams of exchange groups). The a23 n-gram extraction of the 894 protein sequences generates a term-by-sequence matrix with a dimension of 8400 x 894. The ae123 n-gram extraction generates a 8678 x 894 term-by-sequence matrix. The SVD computation of the ae123 matrix yields 884 s-values, only ten less than the total number (Figure 3). The plot shows a sharp drop of the values at ca. the first 30 s-value positions. Similar plots are obtained from all term-by-sequence matrices studied. This seems to suggest that a set of less than 50 or 100 orthogonal factors are sufficient to approximate the original matrix. Indeed, it is found that the right s-vectors corresponding to the 50 to 100-largest s-values usually give the best results (i.e., better than if a larger s-vectors is used). In the present study, a reduced model of 50 to 100 dimensions is used to reduce the size of input vectors from 8400 or 8678 to 50-100. Figure 4 plots the right s-vectors corresponding to the 20-largest s-values computed from the a3 term-by-sequence matrix. While the s-vectors of sequences within the same superfamily are similar, the

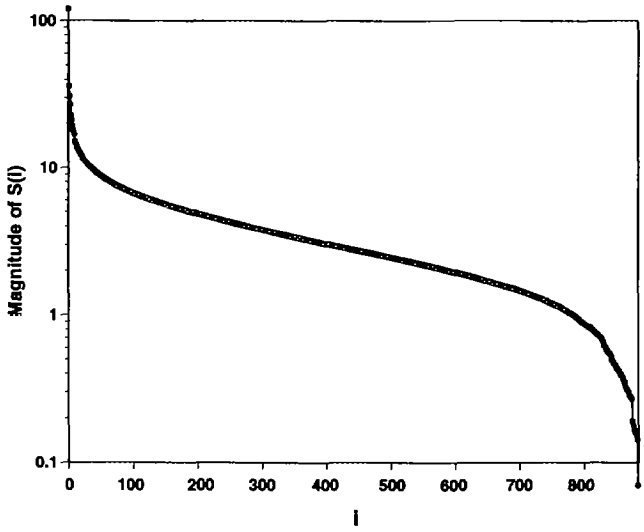

Figure 3. The 884 singular values computed from a 8678 x 894 "term-by-sequence" matrix.

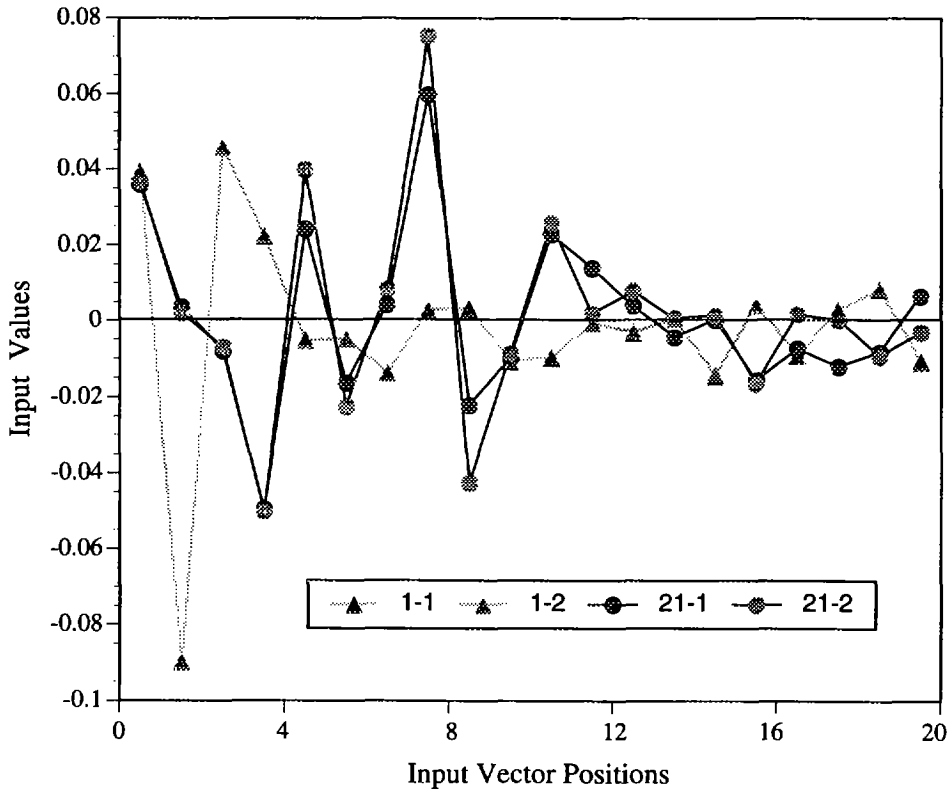

Figure 4. The input vectors derived from the SVD encoding method for a 8000 x 894 "term-by-sequence" matrix. The right singular vectors corresponding to the 20-largest singular values are plotted. 1-1, 1-2, 21-1, and 21-2 represents the first and second sequence entries of superfamily 1, and the first and second sequence entries of superfamily 21, respectively.

Table 2. Comparisons of n-gram and SVD sequence encoding methods for NACANS.

Encoding Method	Network Configuration	Number of Connections	Trained Patterns(%)	Predictive Accuracy(%)
N-gram (drs5)*	1088 x 80 x 28	89,280	100.00	96.18
SVD (d45)	80 x 50 x 28	5,400	100.00	99.32
SVD (d7)	80 x 50 x 28	5,400	100.00	100.00

*The code used in parenthesis represents the type of n-gram vectors (please see text).

s-vectors of different superfamilies (i.e., superfamilies 1 vs. 21) are very different. Therefore, as with the n-gram sequence encoding method, the SVD method also satisfies the basic coding assumption.

The comparison between the ae12 n-gram encoding and the two SVD encodings shows that the sizes of the input vectors and the weight matrices can be reduced by SVD without reducing the predictive accuracy. When the input vectors from the ae12 n-gram method and the a23 SVD method are combined, the classification accuracy is improved to 97.02% (p.s., the 94.89% has been a performance ceiling for this data set using n-gram encodings) (Table 1). Conceivably, the improvement results from additional sequence information embedded in the a3 n-grams. It would be difficult to input the a3 n-gram vector directly to the neural network without a reduction: it would be too large (with 8000 units), and the vector would be too sparse (too many zeros) for the neural network to be trained effectively.

NACANS Performance

The pilot NACANS is developed with 473 entries in 28 phylogenetic classes of the RDP database. The neural network is trained with 316 of the 473 16S ribosomal RNA sequences, and tested with the remaining 157 sequences. The best n-gram encoding method is drs5, which concatenates d5, r5 and s5, pentagram patterns of AUGC, RY (purine, pyrimidine) and SW (strong, weak hydrogen bonding) alphabets. Other network parameters are the same as the ProCANS. A predictive accuracy of 96.2% is achieved counting only first-fit at a cut-off classification score of 0.01 (Table 2).

The same data set is also processed by the SVD method as a comparison. Two n-gram vectors are used, d45 (concatenates d4 and d5, the tetragram and pentagram patterns of AUGC alphabet) and d7 (heptagrams of AUGC alphabet). These generates two term-by-sequence matrices with dimensions of 1280 (d45 n-grams) x 473 (RNA sequences) and 16384 (d7 n-grams) x 473. Both matrices are reduced to a 80 x 473 matrix of right s-vectors by SVD. The results show that with the SVD encoding, classification accuracy can be improved (up to 100%) even with a much smaller network architecture (Table 2). Significantly, the information embedded in d7 n-grams alone is sufficient for 100% classification. This information would be very difficult to capture without the SVD reduction, however, due to its size (16,384 units).

The result of the two systems indicates that both sequence encoding methods can apply equally well to the protein or nucleic acid sequences, although the former has a 20-letter alphabet and the latter has a four-letter alphabet.

System Applications

The major applications of the classification neural networks are rapid sequence annotation and automated family assignment. ProCANS is an alternative database search method. It can be used as a filter program for other database search methods to minimize the time required to find relatively close relationships. The saving in search time will become increasingly significant due to the accelerating growth of the sequence databases. A second version of ProCANS, ProCANS II, is being developed using the Blocks database (Henikoff & Henikoff, 1991), which lists all sequences of the motif region(s) of a protein family. Since the Blocks database is compiled based on the protein groups of the ProSite database (Bairoch, 1992), the system would map protein sequences into ProSite groups. This second system is aiming at sensitive protein classification by applying motif information. The goal is to build an integrated system that permits both rapid identification of close relationships and sensitive detection of distant relationships. As an automated classification tool, the neural systems are hoped to help organize databases according to family relationships and to handle the influx of new data in a timely manner.

The neural networks used in this research are larger than most used in other sequence analysis studies. While the sensitivity of the system is expected to increase with the continuing accumulation of sequence entries available for training, minimal architecture will be adopted to improve network generalization. When the n-gram encoding method is used, the number of weights trained in the networks exceeds by two orders of magnitude the number of training samples. The SVD computation has reduced the input vector and weight matrix sizes significantly.

The neural system has two products: a "neural database" which consists of a set of weight matrices that embed family information in the neural interconnections after iterative training, and a system software that utilizes the neural database for rapid sequence classification. The neural database and the system software has been ported to other computer platforms, including an intel iPSC hypercube and a microcomputer, for speedy on-line protein classification. The system will be distributed to the research community via the use of an anonymous ftp and an electronic mail server.

Acknowledgements

This work is supported by the University Research and Development Grant Program of the Cray Research, Inc. The author also wishes to acknowledge the computer system support of the Center for High Performance Computing of the University of Texas System.

References

Bairoch, A. 1992. PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Research* 20 (Suppl.): 2013-2018.

Barker, W. C., George, D. G., Mewes, H. -W. and Tsugita, A. 1992. The PIR-international protein sequence database. *Nucleic Acids Research* 20 (Suppl.): 2023-2026.

Berry, M. W. 1992. Large-scale sparse singular value computations. *International Journal of Supercomputer Applications* 6: 13-49.

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

Chen, S. 1993. Characterization and learning of protein conformation. In Proceedings of the Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis. River Edge, NJ: World Scientific Publishing Co. Forthcoming.

Cherkassky, V. and Vassilas, N. 1989. Performance of back propagation networks for associative database retrieval. In Proceedings of the International Joint Conference on neural Networks, Volume I, 77-83.

Deerwester, S., Dumais, S. T., Furnas, Landaur, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of American Society for Information Science* 41: 391-407.

Demeler, B. and Zhou, G. 1991. Neural network optimization for E. coli promoter prediction. *Nucleic Acids Research* 19: 1593-1599.

Gribskov, M and Devereux, J. eds. 1991. *Sequence Analysis Primer*. New York, NY: Stockton Press.

Harris, N., Hunter, L. and States, D. 1992. Megaclassification: discovering motifs in massive data streams. In Proceedings of Tenth National Conference on Artificial Intelligence. Menlo Park, Calif.: AAAI Press.

van Heel, M. 1991. A new family of powerful multivariant statistical sequence analysis techniques. *Journal of Molecular Biology* 220: 877-887.

Henikoff, S. and Henikoff, J. G. 1991. Automated assembly of protein blocks for database searching. *Nucleic Acid Research* 19: 6565-6572.

Hirst, J. D. and Sternberg, M. J. E. 1992. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry* 31: 7211-7218.

Holley, L. H. and Karplus, M. 1989. Protein secondary structure prediction with a neural network. *Proceedings of National Academy of Science* USA 86: 152-156.

Horton, P. B. and Kanehisa, M. 1992. An assessment of neural network and statistical approaches for prediction of E. coli promoter sites. *Nucleic Acids Research* 20: 4331-4338.

Kneller, D. G., Cohen, F. E. and Langridge, R. 1990. Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology* 214: 171-182.

Lapedes, A., Barnes, C., Burks, C., Farber, R. and Sirotkin, K. 1990. Application of neural networks and other machine learning algorithms to DNA sequence analysis. In: *Computers and DNA, SFI Studies in the Sciences of Complexity,* Volume VII, 157-182. eds. Bell, G and Marr, T., Reading, Mass.: Addison-Wesley.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. 1989. Back-propagation applied to handwritten zipcode recognition. *Neural Computation* 1: 541-551.

Liebman, M. N. 1993. Application of neural networks to the analysis of structure and function in biologically active macromolecules. In Proceedings of the Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis. River Edge, NJ: World Scientific Publishing Co. Forthcoming.

Olsen, G. J., Overbeek, R., Larsen, N., Marsh, T. L., McCaughey, M. J., Maciukenas, M. A., Kuan, W.- M., Macke, T. J., Xing, Y. and Woese, C. R. 1992. The ribosomal RNA database project. *Nucleic Acids Research* 20 (Suppl.): 2199-2200.

O'Neill, M. C. 1992. Escherichia coli promoters: neural networks develop distinct descriptions in learning to search for promoters of different spacing classes. *Nucleic Acids Research* 20: 3471-3477.

Qian, N. and Sejnowski, T. J. 1988. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology* 202: 865-884.

Rumelhart, D. E. and McClelland, J. L. eds. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations.* Cambridge, Mass.:MIT Press.

Stormo, G. D., Schneider, T. D., Gold, L. and Ehrenfeucht, A. 1982. Use of the 'Perceptron' algorithm to distinguish translation initiation sites in E. coli. *Nucleic Acids Research* 10: 2997-3011.

Uberbacher, E.C. and Mural, R. J. 1991. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proceedings of National Academy of Science* USA 88: 11261-11265.

von Heijne, G. 1991. Computer analysis of DNA and protein sequences. *European Journal of Biochemistry* 199: 253-256.

Wilcox, G. L., Temple, L., Xin, Y. and Liu, X. 1993. Prediction of protein folds from sequence using a neural network trained with sequence-structure association. In Proceedings of the Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis. River Edge, NJ: World Scientific Publishing Co. Forthcoming.

Woese, C. R. 1987. Bacterial evolution. *Microbiological Reviews* 51: 221-271.

Wu, C. H., G. Whitson, J. McLarty, A. Ermongkonchai and T. Chang. 1992. Protein classification artificial neural system. *Protein Science* 1: 667-677.

Wu, C. H. 1993. Classification neural networks for rapid sequence annotation and automated database organization. *Computers & Chemistry.* Forthcoming.