NEURAL NETWORK IMPLEMENTATION OF A DECISION BLOCK FOR THE EVALUATION OF GENOMIC SEQUENCES RECOGNITION SCORES

Lucian-Iulian Fira, Horia-Nicolai Teodorescu

Institute for Theoretical Computer Science, Romanian Academy, Iasi Branch

Abstract: We present a neural network that implements a decision block for the evaluation of the scores of genomic sequence recognition. The recognition scores are computed based on the mean square prediction error of one-step-ahead predictors of the genomic sequence. Four predictors are used for each series obtained as a distance between bases representation. The recognition scores are feed to a classification and decision system, which represents the highest level in the hierarchical recognition system. All neural networks in the system and being used for prediction and classification are MLP type NNs. The method is tested on sequences of the HIV-1 virus.

Keywords: genomic sequence, one-step-ahead predictor, decision, classification, recognition score.

1. INTRODUCTION

Bioinformatics has been one of the fastest developing segments of informatics during the last decade (Mount, 2002). One of the main applications of bioinformatics is the analysis by data mining of the data gathered in the genome projects. Recently, huge databases with genomic sequences were published, deriving from the complete sequencing of the human, virus, microbe and parasite genomes. Gaining a competitive advantage in this post-genome age will depend on the capacity to perform rapid and precise annotation of collected sequences (Wu and Mclarty, 2000).

Because the classification of genomic sequence is an important issue in molecular biology, various methods have been proposed aiming to increase the capabilities of the classification. These methods might be classified into three main categories (Wang et al., 1999), namely i) based on consensus search, ii) based on inductive learning / neural networks, and iii) based on sequence alignments.

In an ample program (Teodorescu, 2003), a method has been proposed, which does not fall into the above categories. The method could be seen as an indirect method and could be named "recognition by prediction ability". The approach consists in two main parts: i) a novel representation of the genomic sequence, and ii) a new method to determine if a given sequence is similar to a known one. The principle of the method is based on the hypothesis that the prediction ability as acquired by training on a specified sequence is preserved only for "similar" sequences. Consequently, similar sequences will be determined as sequences well predicted, while sequences dissimilar to the one(s) predicted will be determined by poor prediction results. Thus, the recognition method consists in the following steps: i) learn to predict a sequence or a set of sequences; ii) test the prediction ability on a given sequence; iii) determine the prediction error on that sequence; iv) decide that the sequence is known or unknown, based on the prediction ability.

A hierarchical hybrid system, able to learn genomic sequences and to detect specific components or known patterns has been developed to implement

steps ii)-iv). The proposed structure consists in a set of four one-step-ahead predictors, which perform the analysis of the genomic sequences separately on each nucleotide type (A, C, G, T). The predictors might be linear systems, neural systems (MLP or RBF), or hybrid systems like neuro-fuzzy predictors. On the superior hierarchical level, a neural decision-making system, receives information from the preceding systems (Teodorescu, 2003). The decision block might consist in a neural network. Technical details about time series predictors and about neural networks for classification can be found in (Liao et al., 2002) and in (Principe et al., 2000).

The coding of the sequence consists in a set of four sequences, each constituted by the distances between successive occurrences of the basis (Teodorescu and Fira, 2003a). The time series obtained as the distances between successive occurrences of the same basis are then separated in two components and independently predicted (Teodorescu and Fira, 2003a, b, c).

By training of the predictors to minimize the MSE (Mean Square Error) for one-step-ahead prediction, the predictor learns the sequences. With the trained predictors, others sequences are tested. A good predictor would work well only on the desired type of sequences, and flags by poor prediction results any unknown type of sequence (Teodorescu and Fira, 2004).

In this paper, the one-step-ahead predictors and the decision block are MLP neural networks. The genomic sequences are obtained from (LANL, 2005), consisting in segments of HIV-1.

The paper is structured as follow: the next section is devoted to the description of the methodology. The third section contains several simulation results. In the fourth section, conclusions are outlined.

2. METHODOLOGY

2.1. Decision block architecture

For the decision block, a neural network with two groups of inputs was considered. The overall architecture is sketched in the Figure 1.

The first group of four inputs. These inputs $(\varepsilon_A, \varepsilon_C, \varepsilon_G, \varepsilon_T)$ represent the prediction error values obtained after the training of the four predictors, for each nucleotide type. The predictors are trained on the series of distances between basis, as was extracted from a specific sequence.

The second group of four inputs. These inputs $(\varepsilon'_A, \varepsilon'_C, \varepsilon'_G, \varepsilon'_T)$ represent the prediction error values obtained by testing of the predictors on other sequences. The parameters of the predictors used for test, are the same parameters obtained before, by training on the specific sequence.

The network outputs represent the code that identifies the sequence on which we have tested the recognition capacity of the predictors. Those codes might be binary numbers or a 1-of-n coding (in this case only single output is one, the others are zero).

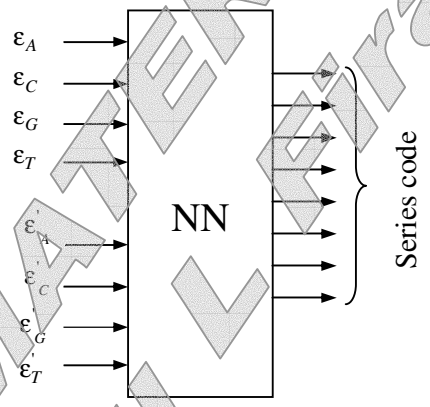

Fig. 1. The decision block architecture

The neural network is supervised trained, the input-output tuples are made from the prediction errors on a specific sequence, the prediction errors obtained by testing on another sequence, and the identification code of the second sequence.

2.2. Construction of the training data set

For this paper, was used a data set consisting in HIV-1 sub-sequences (ENV and GAG), obtained from (LANL, 2005). The constructed database includes all ENV and GAG sequences (725 and 458) available at (LANL, 2005).

Each sequence was translated into four distances time series, for each nitric bases from DNA structure, according to methodology elaborated by the second author (Teodorescu, 2003).

Each distance series was normalized to the [-1,1] interval. From each normalized series, the slow and fast varying components were derived using a causal MA filter. In fact, an average on the current and the last two samples was used for the low-pass filtering; in such way the slow (or trend) component was obtained. The fast component results by subtraction of the trend component from the original series.

From the ENV sequence labeled B.FR.83.HXB2_K03455, for each distances series, a double-block parallel predictor was trained: one block for the trend component and the second block for the fast varying component. The prediction results of the individual blocks are added to achieve a

better prediction quality than in the case of direct distance prediction approach. The prediction was made with one-step-ahead and the Mean Square Prediction Error (MSPE) was output as a quality prediction score.

The predictors trained as above have been tested using all the left sequences, according to the nucleotide type, and the MSPE score has been computed.

The MSPE scores obtaining at the predictors training and the MSPE scores from the predictors testing constitute a record for the database for the training and the testing of the decision system. A database with 1183 records was obtained. The database has the format shown in Table 1.

Table 1 An example with few records from decision systems database.

Name label	Gene	EA	EC	EG	ET	eA	eC	eG	eT	Class
1	2	3	4	5	6	7	8	9	10	
B.FR.83.HXB2_K03455	ENV	0.03	0.17	0.14	0.10	0.03	0.17	0.14	0.10	0
B.FR.83.HXB2_K03455	GAG	0.03	0.17	0.14	0.10	0.14	1.12	0.38	0.12	1
A.CD.97.KCC2_AJ401034	ENV	0.03	0.17	0.14	0.10	0.07	0.54	0.07	0.07	0
A.CD.97.KMST91_AJ401040	ENV	0.03	0.17	0.14	0.10	0.07	0.51	0.16	0.11	0
...	.	...	...	...	..	...	...	...	...	.

The first column contains the sequence name from the Internet database at (LANL, 2005). The second column contains the name of the considered genomic region from HIV-1. There is a correspondence between the second and the last column: 0 for ENV and 1 for GAG.

The columns from 3 to 6 represent the prediction scores for the "witness" sequence, for which the predictors were trained: ENV from HIV-1, labeled B.FR.83.HXB2_K03455. EA, EC, EG, and ET are the MPSE for distance between the bases A, C, G, and T. These values are constant for the entire database. The columns from 7 to 10, labeled eA, eC, eG, and eT, contain the scores obtained by the testing of the predictors on the 1183 sequences. The columns from 2 to 10 represent the inputs of the neural network for classification. The column 11 gives the desired (target) output.

Although in the neural networks practice it is known that the constant inputs bring no useful information, in our case the inputs given by the 2-6 columns represent the reference vis-a-vis from which the decision system must classify the input patterns using data from columns 7-10. An alternate solution to the elimination of the constant inputs and consequently to decrease the number of the neural network inputs (with the advantage of dimensionality reduction for the input space) is to compute only the difference between EA and eA, EC and eC, EG and eG, ET and eT. However, we have not used that method because of future development of the training-testing database by including of records obtained by training of the predictors on another sequences. Moreover, we aim to the inclusion of more classes.

3. RESULTS

3.1. Neural network for decision block architecture

A MLP with a single hidden layer was used. The input layer has a dimension of 8 and the output layer contains a single sigmoid neuron, with output values in the [0,1] interval. Configurations with 2 up to 50 neurons, with one unit step were used for the hidden layer.

In order to determine the classifier accuracy (number of cases correctly classified versus the total number of cases), the neuronal network output values have been rounded, as in equation 1.

$$y = \begin{cases} 0, & if\ x \le 0.4 \\ 1, & if\ x \ge 0.6 \\ *, & else \end{cases} \quad (1)$$

The values from the case "*" belong neither to class 0, nor to class 1. The interval (0.4, 0.6) is a delimitation band between the two classes.

For the training, the available database was surrogated and then it was split in three sets, as follow: 80% data for training, 10% data for validation, and 10% data for testing. Another variant was also implemented and tested: 90% data for training and 10% data for testing, without validation. By surrogating, the order of records in the database was randomly permuted.

3.2. Neural network training with validation

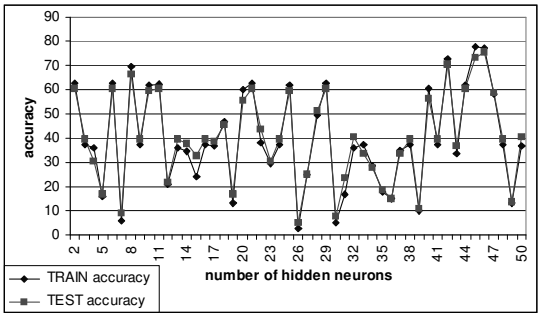

Fig. 2. Neural networks trained with cross-validation

In the Figure 2 we present the results of training and testing of several neural networks with 2 up to 50 neurons into hidden layer. Better performances are obtained for the configuration with 46 neurons in the hidden layer: an accuracy of 77.21% for the training data set and an accuracy of 75.63% for the testing data set.

3.3. Neural network training without validation

By modifying the database partition to 90% data for the training set and to 10% data for the testing set (giving up cross-validation), an increase of accuracy for the training period was obtained, without any modification for the testing period. In the Figures 2 and 3, the graphics for the test accuracy are identical.

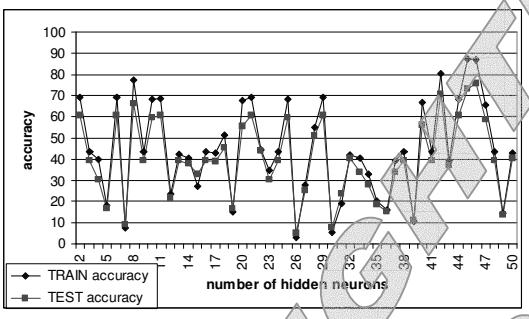

Fig. 3. Neural networks trained without cross-validation

3.4. Statistical validation

The available database was surrogated for nine times and ten supposed different configurations of records in data set were obtained. For all versions of database, were trained neural networks according to methodology described above: MLPs with 3 layers, the hidden layer containing 2 up to 50 neurons. Because better results are again obtained for the configuration with 46 neurons into the hidden layer, in the Table 2 are shown the values of the main statistical parameters for that neural network.

Table 2. Statistical parameters computed for the ten database configurations for MLP with 46 neurons into the hidden layer.

	Training Accuracy (%)	Testing Accuracy (%)
MIN	75.9494	75.6303
MAX	79.1139	83.1933
AVERAGE	77.0464	78.7395
STANDARD DEVIATION	0.9239	3.1455
MEDIAN	76.9515	78.1513
MODE	77.2152	75.6303
SKEWNESS	1.2012	0.3092

The maximum performance, on the testing sets, was 83.1933% patterns correctly classified, with an average of 78.7395%. The best value from Figure 2, that is 75.6303%, is under the average; we can conclude that this value is not an opportune or "lucky" case. On the other hand, the maximum value of accuracy is a consequence of a convenient set for the testing data set, because 83.1933 is greater than 79.1139, the maxim value obtained for the train period. Also, the average of performances for the testing period is superior to the average of performances for the training period. The previous observation is also valid for the median values.

4. CONCLUSIONS AND FURTHER WORKS

In this paper, neural networks that implement a decision block for the evaluation of some genomic sequences recognition scores have been proposed and tested. These recognition scores were computed as the mean square prediction errors generated by one-step-ahead predictors that have been trained before on a specific sequence. From these recognition scores, a database was constructed by training of the predictors on ENV gene from HIV-1 labeled B.FR.83.HXB2_K03455 and testing on other 1183 sequences that include ENV and GAG.

The decision system was trained using two variants of training and testing data set construction: the first includes a validation set and the last no validation set included. No improvements are obtained for testing period in case of use a 90% of data for training. A statistical validation was made using 10 configurations of the database that are obtained by means of surrogating. The average value of accuracy for the decision system was 78.739%.

We can conclude that using the described NNs, the overall hierarchical system as proposed in (Teodorescu, 2003) has been completed and successfully demonstrated.

We intend to further develop the training-testing database by including records obtained by training the predictors on other sequences, and, consequently,

with records with the testing of such predictors on other sequences. Another goal is the inclusion of more classes in the decision systems, by training and testing the predictors on other HIV-1 regions, like POL, LTR, NEF, VIF, or other biological entities.

ACKNOWLEDGEMENTS

The CNCSIS Grant 149/2005 "System for the analysis and prediction of genomic sequences based on neuro-fuzzy data-mining methods" has supported part of the research for this paper, namely the experimenting by the first author. This research is party performed for the Romanian Academy prioritary grant "Cognitive systems and applications" ("Sisteme cognitive şi aplicaţii"); however, there was no financial support from this Grant for this research. The second author has received no grant or other financial support for the research reported here; consequently he reserves all the rights on this research.

REFERENCES

LANL (2005). Los Alamos National Laboratory. http://hiv-web.lanl.gov/cgi-bin/ALIGN_CURRENT/ALIGN-INDEX.cgi. Accessed: 01/MAR/2005

Liao, Y., Moody, J., and Wu, L., (2002). *Applications of Artificial Neural Networks to Time Series Prediction*, in *Handbook of Neural Network Signal Processing*, CRC Press, Boston, USA.

Mount, D.W., (2002). *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press.

Principe, J. C., Euliano, N. R., and Lefebvre, W. C., (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. John Wiley & Sons, New York, USA.

Teodorescu, H.N. (2003). *Genetics, Gene Prediction, and Neuro-Fuzzy Systems–The Context and a Program Proposal*. Fuzzy Systems & A.I. - Reports and Letters, Vol. **9**, Nos. 1–3, 15–22.

Teodorescu, H.N., Fira, L.I. (2003a). *Predicting the Genome Bases Sequences by Means of Distance Sequences and a Neuro-Fuzzy Predictor*. Fuzzy Systems & A.I. - Reports and Letters, Vol. **9**, Nos. 1–3, 23-33.

Teodorescu, H.N., Fira, L.I. (2003b). *A Hybrid Data-Mining Approach in Genomics and Text Structures*, Proc. The Third IEEE International Conference on Data Mining ICDM '03, Melbourne, Florida, USA, November 19 - 22, pp. 649-652.

Teodorescu, H.N., Fira, L.I. (2004). *DNA Sequence Pattern Identification using A Combination of Neuro-Fuzzy Predictors*, 11th International Conference on Neural Information Processing, ICONIP2004, November 22-25, Science City, Calcutta, India.

Wang, J.T., Rozen, S., Shapiro, B.A., Shasha, D., Wang, Z., and Yin, M., (1999). *New techniques for DNA sequence classification.* J. Comput. Biol. 1999 Summer; 6(2):209-18.

Wu, C. H., Mclarty, J. W. (2000). *Neural Networks for Genome Informatics*, In Vol. *Neural Networks and Genome Informatics*, pp. 3-16, Elsevier, USA.