Bidirectional Recurrent Neural Networks for Biological Sequences Prediction

Isis Bonet[1], Abdel Rodriguez[2], and Isel Grau[2]

[1] Escuela de Ingeniería de Antioquia, Envigado, Antioquia, Colombia
ibonetc@gmail.com
[2] Cetro de Estudios de Informática. Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba
igrau@uclv.edu.cu

Abstract. The aim of this paper is to analyze the potentialities of Bidirectional Recurrent Neural Networks in classification problems. Different functions are proposed to merge the network outputs into one single classification decision. In order to analyze when these networks could be useful; artificial datasets were constructed to compare their performance against well-known classification methods in different situations, such as complex and simple decision boundaries, and related and independent features. The advantage of this neural network in classification problems with complicated decision boundaries and feature relations was proved statistically. Finally, better results using this network topology in the prediction of HIV drug resistance were also obtained.

Keywords: Bidirectional recurrent neural network, classification, feature relation, output combination, HIV drug resistance, bioinformatics.

1 Introduction

The classification task is based on assigning a new pattern to one class of a set of N discrete classes. The pattern is represented by a vector $X = (x_1, x_2, \ldots, x_N)$ of N characteristics or features. Classification problems are just as common in bioinformatics as they are in other areas. In this work we focused on the classification of biological sequences, such as nucleotide and protein sequences.

Just like in any classification problem, the search for appropriate features is the first step in building a knowledge database. The representation of biological sequences is particularly difficult; analyzing most biological sequences is easier if we have the three-dimensional structure, but unfortunately it is very difficult and expensive to obtain. This is one of the motives to use primary or secondary structures as an alternative to represent the sequences. These representations are linear and very different to the three-dimensional structure. Complex relations between the amino acids or between some parts of the sequence are hypothetically presumed in order to relate these structures. To represent the sequences, some authors use biological properties such as: hydrophobicity, polarity, etc., in order to end the problem of the variable size of the sequences. However, sometimes it is common to keep the natural

F. Castro, A. Gelbukh, and M. González (Eds.): MICAI 2013, Part II, LNAI 8266, pp. 139–149, 2013.

representation or replace the amino acids or nucleotides with any quantitative measure. This representation takes into account the complex relations that may exist between the elements; this representation can therefore be seen as a time sequence that induces the incentive to use dynamic structures to solve this problem.

Although we will use the primary structure to represent the biological sequences, the variable size is not the objective to use dynamic structures in this paper. In this paper we have supposed that the sequences are all the same sizes or an alignment method was applied. This assumption was done to compare the network against the classic classification methods. The motivation of this paper is to see the one specific structure's ability to deal with problems of complex relation between the features, that we suppose biological sequences have.

To solve classification problems there are some different models of machine learning. Recurrent Neural Network (RNN) has become an increasingly popular method in bioinformatics problems over recent years. Given its temporal connections, the RNN has the particularity of making possible a temporal memory, regardless of whether they are future or past times. Temporal problems are not the only ones that can be solved with this network. Just like a Multilayer Perceptron (MLP) or how a Support Vector Machine (SVM) does with nonlinear kernels, RNN makes an internal feature extraction, By separating the features in subsets associated with times, more complex feature extraction combinations can be achieved.

In particular Bidirectional Recurrent Neural Networks (BRNN) have been used for protein secondary structure prediction [1]. Currently this architecture is considered to be one of the best models for addressing this problem [2]. Some authors have used methods based on BRNN [3] or a combination with other methods [4].

Bidirectional Recurrent Neural Network is a type of Recurrent Neural Networks [5]. This structure has the advantage of not using fixed windows like MLP and can use information from both sides of the sequence, right and left.

The objective of this paper is to compare the behavior of BRNN and classical classification methods when dealing with problems of different dependencies of the features. The topology of BRNN used is the one proposed by Baldi [1], specifically the topology already described in [6]. The main difference between these topologies is the way they combine the outputs. In this paper some output combination functions are used to take into account that the network has one output for each time and in classification problems there is only one output.

Artificial databases with different dependences of the features were built in order to illustrate the potentiality of this type of network in datasets with complex relation between the features. In this paper, we selected a Multilayer Perceptron, as the classic neural network to classification problems as well as the Support Vector Machine and Bayes Network. With this comparison we don't pretend to generalize when the use of BRNN is appropriate. Our purpose is to justify that this method improves the prediction in some problems of biological sequences analysis in comparison to the classical classification methods.

To conclude this paper shows the results using the BRNN to solve the problem of prediction of HIV resistance, using the information of one protein: protease. Also, the results obtained by the BRNN are compared with the other methods.

2 Methods

2.1 Data Preparation

Artificial datasets were generated for this experiment. To build the datasets three factors were kept in mind: feature relation, direction of the relation and decision region of classes.

For each feature a further subset of features was randomly selected. A mathematical dependency was built between the feature and the selected subset. Dependencies were generated by linear, polynomial and piecewise polynomial functions $f(X)$ as can be seen in equations 1, 2 and 3 respectively. $X = (x_1, x_2, \ldots, x_N)$ represents the feature vector with dimension N.

$$f(X) = \frac{\sum_{i=1}^{N} a_i x_i + c}{\sum_{i=1}^{N} a_i + c} \tag{1}$$

In this linear function, a_i is the coefficient for each i, and c is an independent term. This function is finally normalized by the maximum possible value of the numerator, keeping in mind the features generated in the interval [0,1].

As was explain before each feature has a subset of features of which they are dependent upon, named as SDF_k. $SDF_k = (d_{k1}, d_{k2}, \ldots, d_{km})$, where d_{kj} represents the index of features and m is also generated randomly. The coefficient a_i is generated randomly if i is a member of SDF_k, or else the value will be 0.

In equation 2, the coefficients $b_i \in [1,10]$ are added, so the equation behaves polynomially.

$$f(X) = \frac{\sum_{i=1}^{N} a_i x_i^{b_i} + c}{\sum_{i=1}^{N} a_i + c} \tag{2}$$

On the other hand, piecewise generated by polynomial function $h(X)$ defines different behaviors for the last kind of functions: piecewise polynomial functions (equation 3).

$$f(X) = g_i(X), \qquad u_{i-1} \le h(X) < u_i \tag{3}$$

Function g_i defines this behavior for each subdomain. A set of thresholds was generated at random: $U = (u_0, u_1, \ldots, u_R)$, where R represents the number of intervals (generated at random too in the [5,15] interval). A final consideration: $u_0=0$, $u_R=1$ and $\forall i \in [1,R]$: $u_{i-1} < u_i$.

At first, a subset without any feature relation was generated; and used as our reference for the comparison.

Dependencies between the features were analyzed in three ways: forward, backward and in both directions. To build dependencies forward, the selection of the subset of dependent features for a particular feature in the position i, a subset of indexes j were generated in the interval [0, i-1], to backward [i+1, N] and to both directions [1,N], $j \neq i$.

Additionally, the classes were generated with the same features described below.

In total, 285 datasets were generated, 95 datasets with the class generated from each function. Each dataset is described by 9 features and a dichotomy class.

2.2 Models Used

All models used in this paper were implemented in Weka (version 3.6; Waikato Environment for Knowledge Analysis), a software developed at the Waikato University, New Zealand, and available at: http://www.cs.waikato.ac.nz/ml/weka/index.html.

To compare the results, a Multilayer Perceptron, Support Vector Machine, Bayes Network (BayesNet) and C45 decision tree (named J48 in Weka) were selected.

Also BRNN mentioned before was implemented in Java using the Weka package and added to it.

2.3 Bidirectional Recurrent Network Topology

The use of these networks in dissimilar fields has increased in the last few years. These networks have the particularity of making a temporal memory given their temporal connections possible, no matter whether they are future or past times. There are many real problems with these characteristics.

In order to deal with biological sequences problems, a bidirectional model to establish recurrences in two directions was used. On the left in Figure 1 the proposed topology, with three-hidden-layers is shown. This makes the correlations independent of each time with the others. On the right, one can see the unfolded network to the time t. A size of window for the sequence is defined as a parameter of this model. The sequence is divided in sub-sequences according to the size of window (n) defined, where each one represents a time for the network. According to the size of windows it is also possible to define the number of neurons in the input layer. As is shown in figure 1, the topology has a recurrent to one step backward (time t-1) and one forward (time t+1), that is the unfolding is of T times, where T = Sequence Length / n. For example for a sequence divided into three parts (times), the unfolding of the network will be replicated exactly three times.

The network is trained with the Bakpropagation Through Times algorithm [7].

Once the basic algorithm steps for processing a problem have been defined, a procedure for combining the results was introduced.

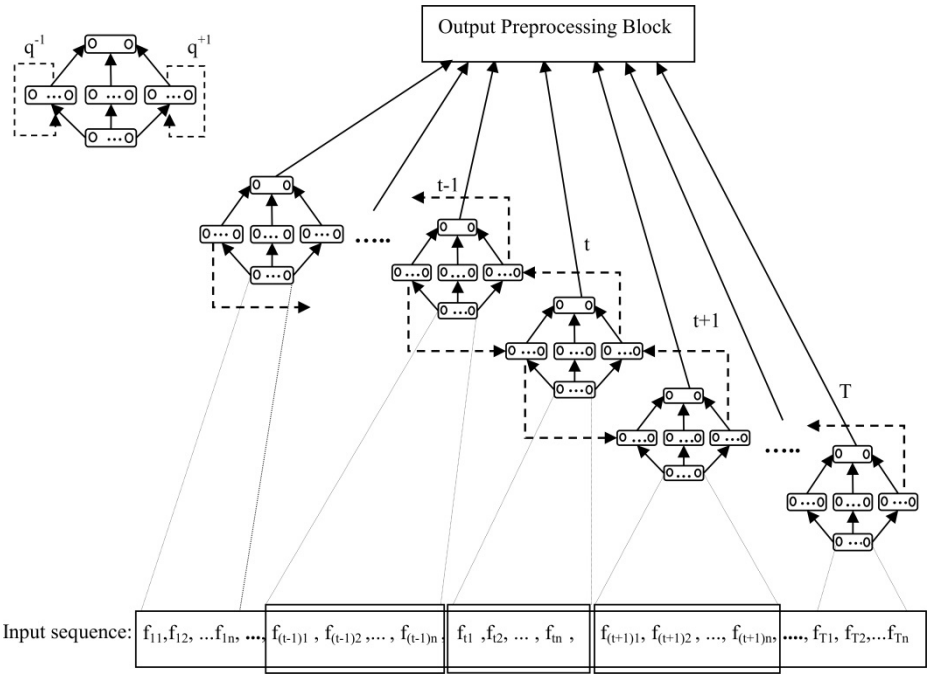

Fig. 1. Bidirectional Recurrent Neural Network and its unfolding in *t*, *t*-1 and *t*+1 times

Output Aggregation Functions

The outputs can be either labels or continuous values. Label outputs refer to the discrete value assigned to each class label. On the other hand, when continuous outputs are used, a *c*-dimensional vector $[d_1, d_2, ..., d_c]$ is provided, where d_j represents the support for the hypothesis that output vector comes from the j^{th} class, and *c* is the total amount of classes.

The model can be compared with a multi-classifier, where each time is a classifier with its own output. Taking into account this idea, the model output can be represented as *T* vectors, one for each time.

In literature, several approaches for aggregating these values into a single output have been proposed and discussed. In this paper, we use the three following variants. Each function returns one vector of membership probabilities for each class, where the final result is the class associated to the index of value in the vector:

- Average function: Calculates the average of the probabilistic values associated with the class membership of the network outputs.
- Max Probability function: Calculates the highest value of class membership probability and returns the class with more probability.
- Mode function: Calculates the class with more probability for each network output and return the class that appears most often as result.

2.4 Performance Evaluation

As was mentioned before, the databases were built artificially with binary classes. The most commonly used parameter to assess the predictively of the classification models is the percent of well-classified cases (eq. 4).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} 100 \tag{4}$$

Where TP is the true positive rate (positives correctly classified/total positives), TN is the true negative rate (negatives correctly classified/total negatives), FP is the false positive rate (negatives incorrectly classified/total negatives) and FN is the false negative rate (positives incorrectly classified/total positives).

Here we used the accuracy and 10-fold cross-validation to show and compare the results.

3 Results and Discussion

3.1 Results from Artificial Databases

The training of the BRNN is based on the topology presented before. To simplify the experiment, three *times* and the same amounts of neurons for each hidden layer were selected: 4, 6, 8 and 10 neurons. Three output combination functions were tested: mode, max and average. Backpropagation Through Time algorithm was used with learning rate 0.01 and momentum 0.9.

We trained a J48, BayesNet, 10 SVMs with polynomial kernels (from 1 degree to 10 degrees), 10 MLPs with 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 neurons in the hidden layer. Then a 10-fold cross-validation was performed for each base, taking the accuracy as performance measures. Also statistical tests were applied.

The analysis of the results is focused on the three factors used to build the databases: feature relation, direction of this relation and the decision boundary, beginning with the last one.

When the class is obtained by a linear function the results of BRNN are not as good as the other methods. In this case, SVMs and MLPs provide better results than BRNNs. This could be due to their capability to find hyperplanes to separate the classes. They are also cheaper computationally speaking, so it is not advisable to use BRNNs in problems with linear separation.

On the other hand, when the class is obtained by polynomial or piecewise polynomial functions, BRNNs are superior to other classifiers depending on the output combination method used.

Figure 2 shows the results of accuracies in datasets with the features generated by a polynomial function. Vertical axes show results obtained by BRNNs with the proposed output combination functions (average, max and mode), and horizontal axes represent the highest accuracy values of the other classifiers: J48, BayesNet, SVM and MLP. The BRNN superiority can be seen, at first sight, in this unfair comparison.

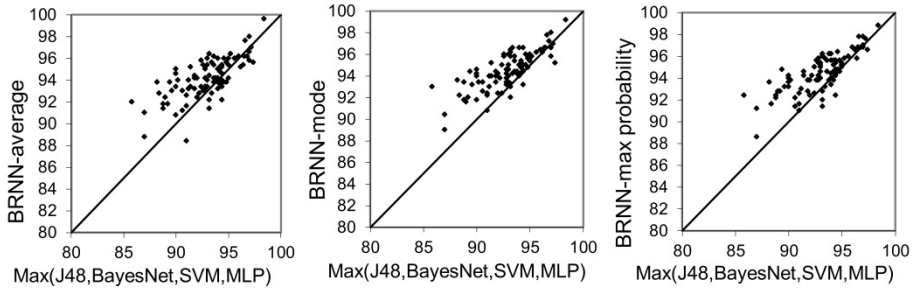

Fig. 2. BRNN accuracy using average, max and mode as combination functions against the J48, SVM, Bayes Net and MLP highest accuracy in datasets with class relation by polynomial function

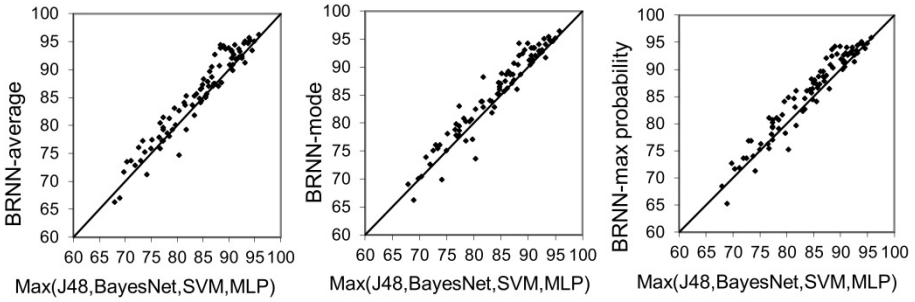

Fig. 3. BRNN accuracy using average, max and mode as combination functions against the J48, SVM, Bayes Net and MLP highest accuracy in datasets with class relation by piecewise polynomial function

The best results are obtained with the mode as output combination function.

Similar results are shown in figure 3, but in this case, the piecewise polynomial function to generate the features is being used. BRNNs are superior again. This suggests that when the decision boundary is complex the BRNN is an alternative method to solve the problem.

Taking into account the factor of feature relation, one could predict that, the results obtained by BRNN in datasets without relation between the features are not really better than others methods. BRNN is computationally expensive and complex. For this reason we suggest not using these networks when the problem has independent features. On the other hand, there are significant differences in datasets with feature relations, no matter the complexity of these relations. Fig 4 shows the comparison between the classical classification methods and the BRNN. Most of the time, BRNN achieve the higher result.

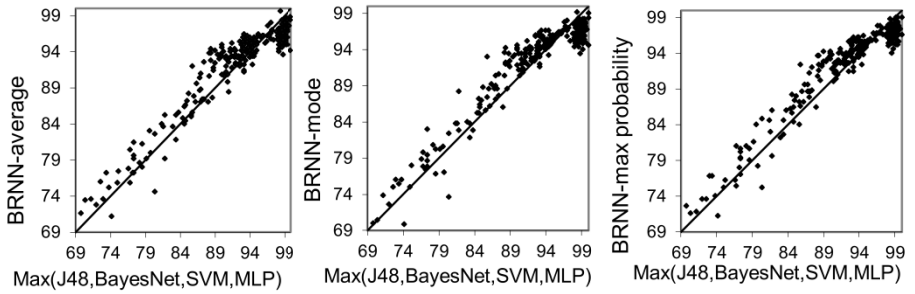

Fig. 4. BRNN accuracy using average, max and mode as combination functions against the J48, SVM, Bayes Net and MLP highest accuracy in datasets with feature relation by linear, polynomial and piecewise polynomial functions

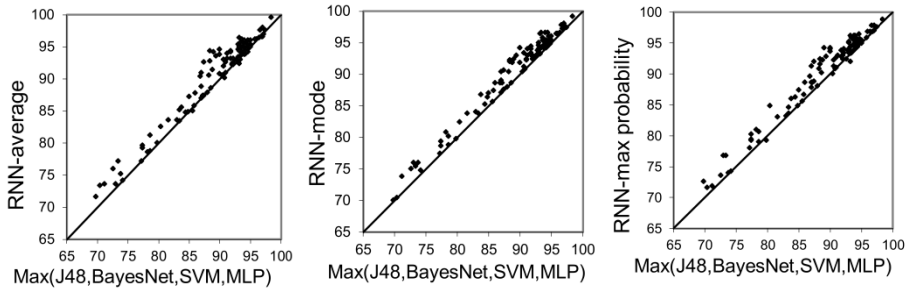

Fig. 5. BRNN accuracy using average, max and mode as combination functions against the J48, SVM, Bayes Net and MLP highest accuracy in datasets with feature and class relation by polynomial and piecewise polynomial function

In figure 5 only the database with relation between the features and a non-linear decision boundary is displayed. As can be seen, the results are now better for the BRNN, aiding towards a conclusion that the combination of a complex decision boundary and a dependency between features suggests that it is more suitable to use methods like BRNN for classification problems.

Additionally, the results obtained taking into account the directions of dependencies between features were analyzed. It is important to observe that BRNN achieves the best results again when the features have dependencies, in the three directions: forward, backward and in both directions. Although the BRNN achieves the best results for the three cases, the best are obtained when the data has dependence in both directions: forward and backward. The output combinations with best results are max probability and mode. As figure 6 illustrates, when features have both dependencies (forward + backward) and the output combinations are max probability or mode the BRNN is always superior or at least similar to the other methods.

To corroborate these conclusions statistic tests were used specifically nonparametric tests and more specifically the two-way Anova Friedman test. It was necessary to carry out a 2-related sample test to contrast the groups.

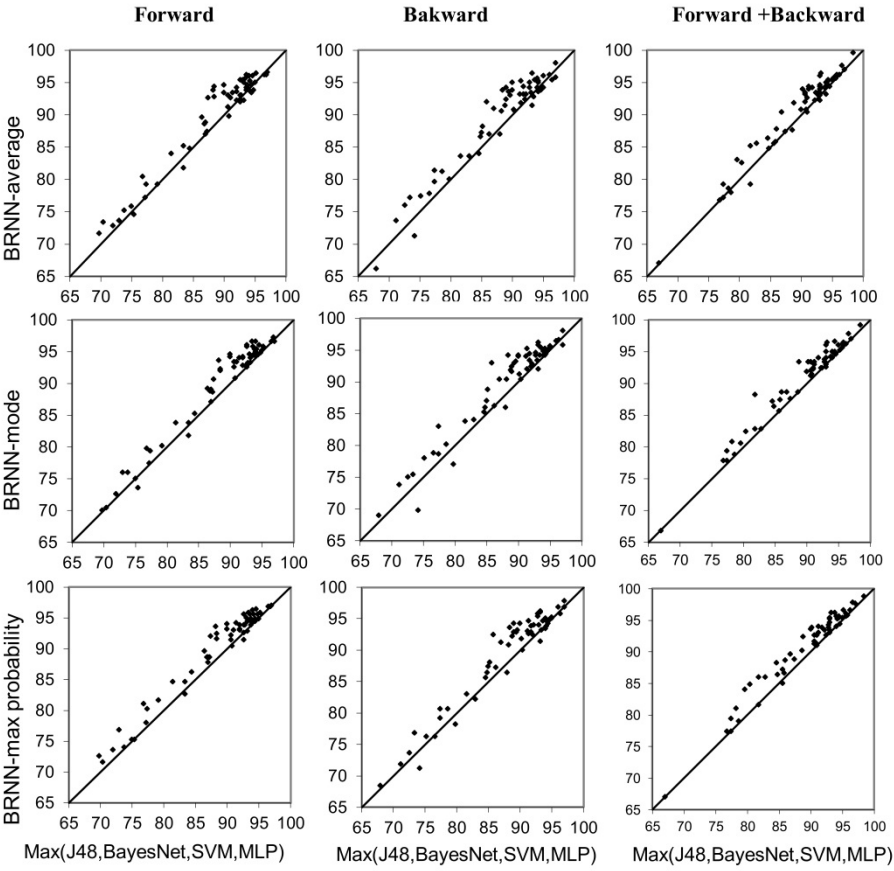

Fig. 6. BRNN accuracy using average, max and mode as combination functions against the J48, SVM, Bayes Net and MLP highest accuracy in datasets with different directions of dependencies between features

The Wilcoxon test shows highly significant differences between results obtained by BRNN and the other classifiers in those datasets where the class is obtained by polynomial and piecewise polynomial functions.

Furthermore, the comparison between the results related with features relation and with the dependencies between them reconfirms the superiority of the BRNN when the data has dependencies between the features in any direction.

Finally, output combinations were compared. Mode and max probability were best for accuracy, instead of the expected average function (the continuous central tendency measure).

3.2 Results Using the HIV Drug Resistance Database

The information to build the databases is from the "Stanford Database" [8]. There are 7 databases corresponding to drug resistance in the following protease inhibitors: Amprenavir (APV), Atazanavir (ATV), Indinavir (IDV), Lopinavir (LPV), Nelfinavir (NFV), Ritonavir (RTV) y Saquinavir (SQV).

In [6] the use of this BRNN topology is shown with mode as the output combination function, to solve this problem, but with another version of the database. Here more cases from the database were used. BRNN is compared with previous results with others methods. In [9] the results obtained by a lot of classification methods to predict the HIV drug resistance is shown.

Here the BRNN is trained with the three output combination functions used before. Also other classification models were trained: J48, SVM with different kernels: linear, polynomial and Gaussian; BayesNet, MLP.

In this work the amino acids are represented with their contact energies and the database is the last version of the Stanford Database. For these reason the obtained results are a slightly different to those obtained in [9] and [6].

Table 1 illustrates the results obtained by different models. BRNN achieves accurate similar or superior results in all cases. The best output combinations for this problem are mode and max probability.

Table 1. Results of accuracy for database of protease inhibitors

	J48	SVM linear	SVM Polynomial	SVM Gaussian	BayesNet	MLP	BRNN Average	BRNN Mode	BRNN Max Probability
APV	0.82	0.82	0.82	0.69	0.81	0.79	0.82	**0.83**	**0.83**
ATV	0.65	0.75	0.73	0.61	0.68	0.76	0.74	0.75	**0.77**
IDV	0.89	0.89	0.88	0.82	0.89	0.88	0.87	**0.90**	**0.90**
LPV	0.89	0.87	0.88	0.85	0.86	0.89	0.89	**0.91**	0.89
NFV	0.90	0.88	0.86	0.71	0.90	0.88	0.91	0.91	**0.92**
RTV	**0.93**	0.90	0.90	0.80	0.91	0.90	0.91	**0.93**	**0.93**
SQV	**0.76**	0.74	0.74	0.74	0.72	0.73	0.72	0.74	**0.76**
Average	0.83	0.84	0.83	0.74	0.83	0.83	0.84	**0.85**	**0.85**

In this biological sequence problem the BRNN also achieves the best or at least similar results in most of the databases, as is shown in table 1. Although the mode achieves better results with respect to the rest of methods, the max probability is now the aggregation function with best results.

4 Conclusions

BRNN is not the best classifier in linear decision boundary problems. In these problems, other simpler methods are in fact better, such as, SVM and MLP. However, in problems with complex decision boundaries, as soon as relations start emerging between features, BRNN becomes the best classifier. The best results of this model are

when the features have dependencies in the sequences on both backward and forward. It is recommended to use the mode or the max probability as output combination.

In regards to the problem of HIV drug resistance the results of the topology of BRNN proposed has superior results or at least similar to the results obtained by the other techniques. These results and conclusions do not mean that the model described here is better than other methods for any type of biological problem, but it is a promising method to bear in mind.

Acknowledgements. The authors would like to thank Luke James for his help and proof reading.

References

1. Baldi, P., Brunak, S., Frasconi, P., Soda, G., Pollastri, G.: Exploiting the past and the future in protein secondary structure prediction. Bioinformatics 15, 937–946 (1999)
2. Agathocleous, M., Christodoulou, G., Promponas, V., Christodoulou, C., Vassiliades, V., Antoniou, A.: Protein Secondary Structure Prediction with Bidirectional Recurrent Neural Nets: Can Weight Updating for Each Residue Enhance Performance? In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds.) AIAI 2010. IFIP AICT, vol. 339, pp. 128–137. Springer, Heidelberg (2010)
3. Walsh, I., Martin, A.J.M., Di Domenico, T., Tosatto, S.C.E.: ESpritz: accurate and fast prediction of protein disorder. Bioinformatics 28, 503–509 (2012)
4. Ceroni, A., Frasconi, P., Passerini, A., Vullo, A.: A Combination of Support Vector Machines and Bidirectional Recurrent Neural Networks for Protein Secondary Structure Prediction. In: Cappelli, A., Turini, F. (eds.) AI*IA 2003. LNCS, vol. 2829, pp. 142–153. Springer, Heidelberg (2003)
5. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681 (1997)
6. Bonet, I., García, M.M., Saeys, Y., Van de Peer, Y., Grau, R.: Predicting Human Immunodeficiency Virus (HIV) Drug Resistance Using Recurrent Neural Networks. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4527, pp. 234–243. Springer, Heidelberg (2007)
7. Werbos, P.J.: Backpropagation Through Time: What it does and How to do it, pp. 1550–1560 (1990)
8. HIV Drug Resistance Database, `http://hivdb.stanford.edu//cgi-bin/GenoPhenoDS.cgi`
9. Rhee, S.-Y., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D.L., Shafer, R.W.: Genotypic predictors of human immunodeficiency virus type 1 drug resistance. Proceedings of the National Academy of Sciences 103, 17355–17360 (2006)