Artificial Neural Networks in Bioinformatics

Dr. Muditha M. Hapudeniya MBBS
Medical Officer and Postgraduate Trainee in Biomedical Informatics
Postgraduate Institute of Medicine
University of Colombo
Sri Lanka
E-mail: mmhapudeniya@gmail.com

Abstract

Bioinformatics is a new research area which integrates many core subjects such as biology, medicine, computer science, and mathematics. Since most of the problems in bioinformatics are inherently hard researches have used artificial intelligence techniques to solve such problems. Artificial neural networks are one such method used in many situations and have proved to be very effective. This paper will focus on issues related to construction of a neural network to solve bioinformatics problems and describes some of its current applications.

Keywords: Bioinformatics, Artificial neural networks

Sri Lanka Journal of Bio-Medical Informatics 2010;**1**(2):104-111
DOI: 10.4038/sljbmi.v1i2.1719

Introduction

Bioinformatics is a promising and novel research area in the 21st century. This field is data driven and aims at understanding of relationships and gaining knowledge in biology. In order to extract this knowledge encoded in biological data, advanced computational technologies, algorithms and tools need to be used. Basic problems in bioinformatics like protein structure prediction, multiple alignment of sequences, phylogenic inferences, etc are inherently non-deterministic polynomial-time hard in nature. To solve these kinds of problems artificial intelligence (AI) methods offer a powerful and efficient approach. Researchers have used AI techniques like Artificial Neural Networks (ANN), Fuzzy Logic, Genetic Algorithms, and Support Vector Machines to solve problems in bioinformatics. Artificial Neural Networks is one of the AI techniques commonly in use because of its ability to capture and represent complex input and output relationships among data. The purpose of this paper is to provide an overall understanding of ANN and its place in bioinformatics to a newcomer to the field.

Bioinformatics

Biomedical Information Science and Technology Initiative defines *Bioinformatics* as research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data. Computers are used to collect, store, analyze and integrate biological and genetic information which can then be applied to solve many problems inherited to biology. Different biological problems considered within the scope of bioinformatics(1) fall into main tasks which are given below.

- Alignment and comparison of DNA, RNA, and protein sequences.
- Gene finding and promoter identification from DNA sequences.
- Interpretation of gene expression and micro-array data.
- Gene regulatory network identification.
- Construction of phylogenetic trees for studying evolutionary relationship.

- Protein structure prediction and classification.
- Molecular design and molecular docking.

Therefore the aims of bioinformatics are:

- To organize data in a way that allows researchers to create and access information
- To develop tools that facilitate the analysis and management of data.
- To use biological data to analyse and interpret the results in a biologically meaningful manner.

In order to achieve the above mentioned goals, researchers have developed various algorithms. Some of the common algorithmic trends in bioinformatics are listed below.

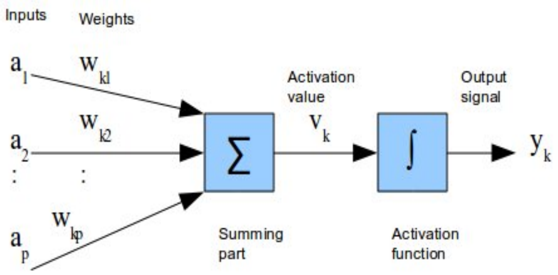
- Finding similarities among strings (such as DNA or proteins of different organisms).
- Detecting certain patterns within strings (such as genes, introns, and α-helices).
- Finding similarities among parts of spatial structures (such as motifs).
- Constructing phylogenetic trees.
- Classifying new data according to previously clustered sets of annotated data.
- Reasoning about micro-array data

Neural Networks (NN)

Neural networks are originally modeled as a computational model[(2)] to mimic the way the brain works. Brain is made from small functional units called neurons. A neuron has a cell body, several short dendrites and single long axon. Each neuron connects to several other neurons by dendrites and axons. Dendrites receive signals from other neurons and act as the inputs to the neuron. These inputs increase or decrease the electrical potentials of the cell body and if it reaches a threshold, an electrical pulse is sent down the axon. This output becomes the input to several other neurons.

Similarly, an artificial neural network builds from several computational units which are sometime called a neuron. These units are connected by links and each link has a weight associated with it. Weights are analogues to long term memory. Similar to the biological neuron, each unit receives inputs from input links. Each unit then computes the weighted sum of the input values and a transfer function transforms a final value that act as the unit's output value[(2)]. A simple model of a neural network is shown in Figure 1.

Figure 1. A simple neural network model



From this model the internal activity of the neuron can be shown by:

$\theta = bias\ term$

$$v_k = \sum_{j=1}^{p} w_{kj} x_j - \theta$$

When □ is the activation function, output y_k can be given by $y_k = \varphi v_k$

There are several activation functions used in the designing of neural networks and few common functions are the step function, the sigmoid or logistic function and the Gaussian function.

Similar to the brain, NN can learn from examples and apply the knowledge to new situation. This process is called the training of the NN.

Architecture of Neural Networks

There are several common network architectures used in field of bioinformatics which have unique application.

Perceptron

This is the simplest form of neural network which has 2 layers; the input layer and the output layer. Perceptrons are very restricted in their use since it can be used only to classify patterns in to one of two classes.

Multi layer perceptron (MLP)

Multi layer perceptrons are perceptrons which have more than 2 layers of neurons. MLP has an input layer, one or more hidden layers and an output layer. Usually, the transfer function of the hidden and output layer is logistic or sigmoid in function. Normally, neurons of one layer are connected to all the neurons of the next layer and are called a fully connected network but exceptions can occur. MLP is capable of classifying the data set by the use of hyper-planes that divides the data in to discrete areas.

Radial Basis function network

The Architecture of the radial basis function is similar to the MLP but the principle of action and training is different from MLP. Radial basis function can cluster the data in to finite number of ellipsoid areas. Transfer function is usually a Gaussian function, a splice function or various quadratic functions. Each hidden unit of the network acts as the center of the region. Inputs to these units are not a weighted sum but a distance measure. Most common is the Euclidean function. The hidden unit then computes the output as a function of the input vector and its center.

Kohonen self organizing maps

These kinds of networks are very different from previous networks. Kohonen self organizing maps have the input layer but no hidden or output layer(2). Input layer units connect to a grid of discrete units. They are fully connected and links are associated with weights. The input vectors are mapped to one of the grid points by computing the distance matrix (Euclidean distance) for each grid point to decide on the closest point which matches the input.

Training of Neural Network

Before using any NN model, it must be trained with representative data. There are basically two types of training; supervised and unsupervised. The basic idea behind training is to pick up set of weights (often randomly), apply the inputs to the NN and check the output with the assigned weights. If it is not the required output, modify the weights with some algorithm and repeats the procedure. This process continues until some stopping criterion is reached. When using supervised training it is important to address the following practical issues.

Over training – This is a serious problem where NN reduces error to an extent that it simply memorizes the data set used in training. Then it will not be possible to categorise the new data set and the generalisation is not possible, which is not the required behavior of the NN.

Validation set – To prevent over-training, validation set of data can be used. As the training proceeds the training error will decreases and the result of applying the validation set improves.

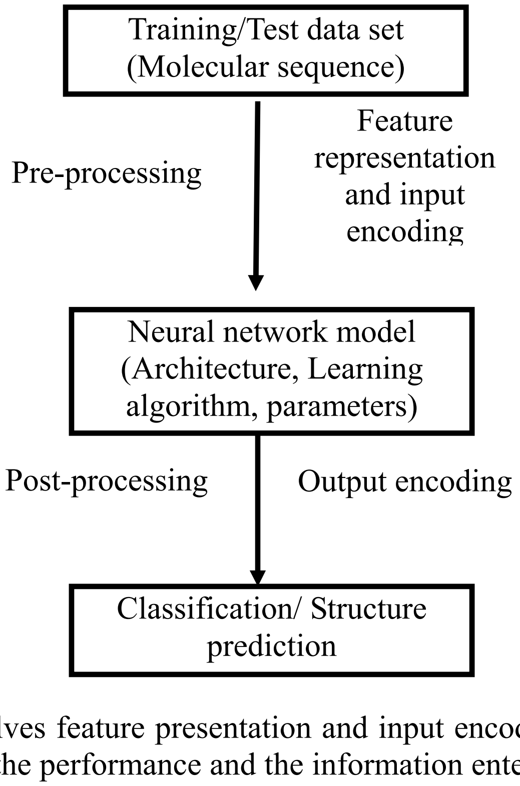
Test data – Test data is a separate dataset which is used to test the trained NN to determine whether the NN has generalised the training data set accurately.

Data preparation – Sometimes it is useful to scale data before training. This will improve the training process.

Designing Neural Networks for Bioinformatics

When designing NN for bioinformatics applications, there are common designing issues that needs to be addressed. Figure 2 summarises these issues(2).

Figure 2. Design Neural Networks for bioinformatics


Training/Test data set (Molecular sequence)
Pre-processing
Feature representation and input encoding
Neural network model (Architecture, Learning algorithm, parameters)
Post-processing
Output encoding
Classification/ Structure prediction

Preprocessing of data involves feature presentation and input encoding. This is an important element which determines the performance and the information entered to the NN. In order to get the full benefit of the NN, the designer has to represent prior knowledge about sequence structure and functions(2). This allows the extraction of salient features in the given sequence. When considering proteins, they are made up of a combination of 20 amino acids and have various lengths. A sequence can be written using a defined alphabet set. But in biology these letters carry more information such as residue structure and function. It is well known that the amino acid side chain exhibits a large number of chemical and physical properties. The interaction of these side chains with one another and with the backbone of the protein determines its structure and function. Several important properties of amino acids like hydrophobisity, surface area, chemical properties, polarity, volume, secondary structure propensity, etc. have been encoded as prior knowledge when using as the input for NN [2].

Some amino acids play more than one structural or functional role; therefore their properties need to be assessed considering the entire protein. These are known as protein context features which can be local or global. In order to represent these features; hydrophobic moment, hydrophobicity profile or amino acid frequency can be calculated and encoded to use in NN[2].

Another type of feature is the protein evolutionary feature which is normally represented in substitution metrices like PAM or BLOSUM. This information can be encoded and used in the NN.

After identifying the features needed to be represented in the NN application, there are several ways to represent these data[2] to maximize information extraction. They are as follows:

- Real number measurements in a continuous scale e.g. Mass
- Vectors of distance or frequencies e.g. PAM matrix
- Categorised into classes
- Using alternative alphabet to represent AA with similar features
- Hierarchical classes

After identifying the features that need to be represented in the NN application, data need to be encoded. Encoding can be local (involving single or neighboring residues in short sequence segments) or global (involving long range relationship in entire sequence)(2). The sequence encoding method can be direct or indirect. Direct encoding converts each residue in to a vector and it preserves the positional information. Indirect encoding on the other hand provides overall information measures of the entire sequence(2).

Output encoding is not as complex as input encoding. It depends on the number of classification required in the application. However networks like self organising maps automatically configure the number of output units. The value of the output units can be used qualitatively or quantitative measure of confidence level or activity level.

Applications in Bioinformatics

Although NNs are described as a separate entity, they are rarely used as a standalone application. They are often a section or part of a larger application.

Coding region recognition and gene identification

Pattern recognition methods are mainly used in describing the location and significance of genes in a genome. In prokaryotes, the coding region is a continuous single reading frame. In eukaryotes, it consists of introns and exons. Therefore, the main task is to differentiate introns, exons and splice site detection. The GRAIL(3) system is an example of an application developed using NN for cording region recognition. GRAIL is a multiple sensor-neural network based system. It can locate genes in anonymous DNA sequence by recognising features related to protein coding regions and the boundaries of coding regions. These recognised features are combined using a neural network system. Developers of the GRAIL claim that it consistently achieved about 90% of coding portions of test genes with a false positive rate of about 10%. Eric E. Snyder and Gary D. Stormo(4) have used a simple feed forward NN to identify coding regions in DNA sequences and were able to demonstrate correlation coefficient for exon prediction of 0.85. There are several web services constructed using NNs; NetGene2 server is one of them which is producing neural network based predictions of splice sites in human, *C. elegans* and *A. thaliana* DNA and is available at URL http://genome.cbs.dtu.dk/services/NetGene2/

Recognition of transcription and translational signals

This task involves the prediction of promoters and sites that function in initiation and termination of transcription and translation. Kalate, Tambe and Kulkarni(5) in their study of the prediction of mycobacterial promoter sequences have used a multi layered feed-forward NN architecture trained using the error-back-propagation algorithm. They were able to achieve high prediction capability (97%) with their approach. Reese and Eeckman(6) have used a neural network prediction systems for human promoters and splice sites, and were able to recognize 50% of the human gene promoters with a false positive classification of 0.8% (correlation coefficient of 0.61). Tikolea and Sankararamakrishnan(7) have experimented with several NN models in prediction of translation initiation sites in human mRNA sequences and found that the neural network with two hidden layers have a sensitivity of 83% and specificity of 73% indicating a vastly improved performance.

Sequence feature analysis and classification

Blekas, Fotiadis, and Likas(8) have presented a system for multi-class protein classification based on neural networks. They have found that the experimental results on real datasets indicate that their proposed method is highly efficient and is superior to other well known methods for protein classification.

Protein structure prediction

Protein structure prediction involves the predicting of the secondary and tertiary structure of proteins. Identification of three classes of secondary structures; α- helix, β- sheets and reverse turns constitutes the major task. NNs have also been applied to predict protein tertiary structure such as prediction of side-chain packing and structural class prediction. Chae and colleagues(9) have constructed a neural network that employs the information from atom-pair distance distributions of a large number of decoys to predict protein complex geometries. They have found that their neural network based scoring function achieves a reasonable performance in rigid-body unbound docking of proteins. Kakumani and colleagues (10) have proposed a two-stage protein secondary structure prediction technique, implemented using neural network models. The first neural network stage of the proposed technique associates the input protein sequence to a bin containing its corresponding homologues. The second stage predicts the secondary structure of the input sequence utilising a neural prediction model specific to the bin obtained from stage one. Punta and Rost(11) have used both supervised and unsupervised neural networks to the prediction of protein structure and function. They have focused on feed forward neural networks and described how these learning machines can be applied to protein prediction. Fuchs, Kirschner and Frishman(12) have used a neural network approach to predict helix-helix contacts and were able to obtain prediction of contacts between residues in transmembrane segments with nearly 26% accuracy. On their dataset consisting of 62 membrane proteins of solved structure, they gained an accuracy of 78.1%.

Prediction of signal peptides

Plewczynski and colleagues(13) have developed a neural network based method for detection of signal peptides in proteins. The method was trained on sequences of known signal peptides extracted from the Swiss-Prot protein database and was able to work separately on prokaryotic and eukaryotic proteins. Their method provided a significantly higher speed and portability. The accuracy of cleavage site prediction reached 73% on heterogeneous source data that contained both prokaryotic and eukaryotic sequences while the accuracy of discrimination between signal peptides and non-signal peptides was above 93% for any source dataset. Similar methods have been developed by Nielsen and colleagues(14) for the identification of signal peptides and their cleavage sites based on neural networks trained on separate sets of prokaryotic and eukaryotic sequences. They claim that their method performs significantly better than previous prediction schemes, and could easily be applied to genome-wide data sets.

Conclusion

It is evident in the literature that artificial intelligence techniques like Neural Networks are heavily used in the field of bioinformatics to solve hard problems. These methods have proved and established its value in the field of bioinformatics. Knowledge and ability to use neural networks method add definite advantage to bioinformaticians to solve many types of problems in the field of bioinformatics.

References

1. Jena RK, Aqel MM, Srivastava, Mahanti PK. *Soft Computing Methodologies in Bioinformatics.* Europian Journal of Scientific Research, 2009. **26**(2): p. 189-203.

2. Wu CH, McLarty JW. *Neural Networks and Genome Informatics*. Methods in computational Biology and Biochemistry, 1 ed. Vol. 1. Elsevier; 2000.

3. Ying Xu, Mural RJ, Einstein JR, Shah MB, Uberbacher EC. *GRAIL: a multi-agent neural network system for gene identification.* Proceedings of the IEEE 1996. **84**(10).

4. Snyder EE, Stormo G. *Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks.* Nuclic Acids Research, 1993. **21**(3).

5. Kalate RN, Tambe SS, Kulkarni BD. *Artificial neural networks for prediction of mycobacterial promoter sequences.* Comput Biol Chem, 2003. **27**(6): p. 555-64.

6. Reese MG, Eeckman FH. (1995) *NOVEL NEURAL NETWORK PREDICTION. SYSTEMS FOR HUMAN PROMOTERS AND. SPLICE SITES*. Available from : http://eprints.kfupm.edu.sa/53545/1/53545.pdf

7. Tikolea S, Sankararamakrishnan R. *Prediction of translation initiation sites in human mRNA sequences with AUG start codon in weak Kozak context: A neural network approach.* Biochemical and Biophysical Research Communications, 2008. **369**(4): p. 1166-1168.

8. Blekas K, Fotiadis DI, Likas A. *Motif-Based Protein Sequence Classification Using Neural Networks.* Jouranl of Computational Biology, 2005. **12**(1): p. 64-82.

9. Chae MH, Krull F, Lorenzen S, Knapp EW. *Predicting protein complex geometries with a neural network.* Proteins, 2010. **78**(4): p. 1026-39.

10. Kakumani R, Devabhaktuni V, Ahmad M. *A two-stage neural network based technique for protein secondary structure prediction.* Conf Proc IEEE Eng Med Biol Soc, 2008. **2008**: p. 1355-8.

11. Punta M, Rost B. *Neural networks predict protein structure and function.* Methods Mol Biol, 2008. **458**: p. 203-30.

12. Fuchs, A, Kirschner A, Frishman D. *Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks.* Proteins, 2009. **74**(4): p. 857-71.

13. Plewczynski D, Slabinski L, Ginalski K, Rychlewski L. *Prediction of signal peptides in protein sequences by neural networks.* Acta Biochim Pol, 2008. **55**(2): p. 261-7.

14. Nielsen H, Engelbrecht J, Brunak S, von Heijne G. *A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites.* Int J Neural Syst, 1997. **8**(5-6): p. 581-99.