DESIGN OF ARTIFICIAL BACK PROPAGATION NEURAL NETWORK FOR DRUG PATTERN RECOGNITION

Ms Jyoti C Chaudhari
Department of Computer Engg
K K Wagh College of Engg,
Nashik.
E-mail: aim_jyoti@yahoo.co.in

Abstract

In recent years considerable effort s has been devoted to applying pattern recognition techniques to the complex task of data analysis in drug research. Artificial neural network is a branch of artificial Intelligence (AI). This is one of the most successful techniques used to find out non-linear regression among the properties of any entity. The non-linear regression helps in deciphering the hidden relationships among the various properties of an entity. Artificial neural network methodology is a modeling method with great ability to adapt to a new situation or control an unknown system, using data acquired in previous experiment The neural network can be considered as a tool for molecular data analysis and interpretation. Analysis by neural networks improve the classification accuracy, data quantification and reduce the number of analogues necessary for correct classification of biological active compounds. The experimental results verify these characteristics and show that the back propagation model practical classifier for pattern recognition system. Artificial neural network are being developed for many medical applications system , The back propagation neural network is widely used in the field of pattern recognition because this artificial neural network can classify complex pattern and perform nontrivial mapping function. Neural network are used in pattern recognition because of their ability to learn and to store knowledge e Because of their parallel nature can achieve, artificial neural network can achieve very high computation rates which is vital in application like telemedicine. In this paper the emphasis is to use "Pattern Recognition Neural Network. Number of workers at various laboratories are working in this direction. In present study, an effort is being made to prepare the logical assembling of the various advanced methods which will be circulating around the Artificial Neural Network. It is reported that drug industries need the fast screening of chemical molecule to determine drug like properties in molecules. l help the drug design scientist to select the correct molecules for synthesis as otherwise it takes 13 to 14 years to finalize the molecule having drug like properties.

Key Words- Artificial Neural Network, Backprapogation neural network, Molecular mechanics techniques, Pattern recognition

1. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology

Which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

2. Back propagation learning algorithm

Back-propagation is a supervised learning technique used for training artificial neural networks. It was first described by Paul Werbos in 1974, and further developed by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams in 1986.

As the algorithm's name implies, the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes. So technically speaking, back-propagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. This gradient is almost always then used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "back-propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent.

The Back propagation algorithm has been widely used as a learning algorithm in feed forward multiplayer neural networks. The Back propagation is applied to feed forward ANN with one or more hidden layers.

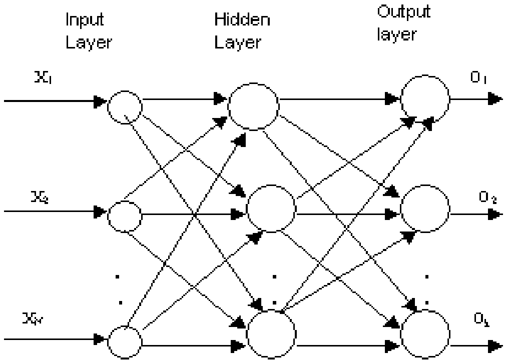


Fig 1. Feed forward Multiplayer Perceptron

Based on the algorithm, the network learns a distributed associative map between the input and output layers.

This algorithm is different than others in the way in which the weights are calculated during the learning phase of the network. In general, the difficulty with multiplayer perceptions is calculating the weight of hidden layers in an efficient way that result in the least output error; the more hidden layers there are , the more difficult it becomes. To updates the weights, one must calculate the error. At the output layer the error will easily measured; this is the difference between the actual and desired output at the hidden layers, however there is no direct observation of the error. Hence some other technique must be used to calculate an error at the hidden layers that will cause minimization of the output error, as this is the goal.

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (***EW***). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the ***EW***.

The steps as follows:

$$\mathrm{EA}_j = \frac{\partial E}{\partial y_j} = y_j - d_j$$

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.
2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$\mathrm{EI}_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = \mathrm{EA}_j y_j \left(1 - y_j\right)$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$\mathrm{EI}_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = \mathrm{EA}_j y_j \left(1 - y_j\right)$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to multilayer networks. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j W_{ij}$$

By using steps 2 and 4, we can convert the ***EAs*** of one layer of units into ***EAs*** for the previous layer. This procedure can be repeated to get the ***EAs*** for as many previous layers as desired. Once we know the ***EA*** of a unit, we can use steps 2 and 3 to compute the ***EWs*** on its incoming connections.

3. Pattern Recognition

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward (figure 3) neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input. In this case, the network gives the output that corresponds to a taught input pattern that is least different from the given pattern

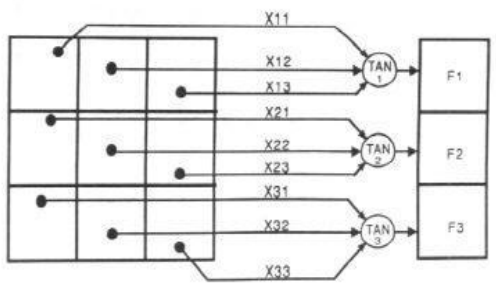


Fig2: Feed-forward ANN for pattern recognition.

4. 3D Molecular structure

Numerous theoretical methods in the field of computational chemistry fall back on the availability of 3D structures of compounds. Determine molecular structure without human interaction is an essential components of this pattern recognition technique. The efficiency of 3D structure based on high throughput screening tools. This can be done by molecular mechanics technique. Following figure is the sample of 3D molecule.

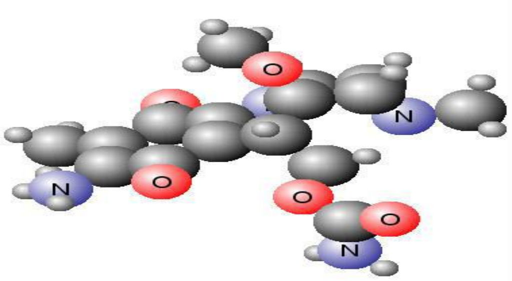


Fig 3: 3D structure of sample

5. Objective

To understand the ANN methodology to solve non-linear regration problems. To prepare program for back propagation neural network using the concept of system analysis and design. To test the prepared program using “X-OR” logic, and validate model. Study the anti-cancer drugs, and design the ANN network for it. To check the new molecules for anti-cancer properties.

6. Proposed Scheme

The stepwise process of using ANN to map anti-cancer drugs is as follows: a) Select the list of anti-cancer drugs having certain common geometry. In the present study “Alkaline type of anti-cancer drugs are selected” .b) Prepare 3D structure of the drugs c) find the Molecular Electrostatic Potentials of the molecules. d) Find out the different parameters of anticancered drugs like log p value, energy level, capacity, etc. Store the value of that parameter. Generally 40-50 drugs will be considered. That is used for learning the network. Prepare Back-Propagation ANN. e) Train the network. f) Use this trained network for the selection of unknown compounds as an anti-cancer agent. n) The prepared Back-propagation ANN can be used to select any new molecule as a anti-cancer molecules.

Te different parameter will be considered, for Eg. Learning rate can be 1, momentum term is 1, Error tolerance as 0.5, display iteration as 20, number of input layer, hidden layer, output layer.

7. Test for validation

Following are the procedures for testing the validation of software.

In this procedure we are going to explain that how the Artificial neural network (ANN) solve the logical base XOR problem.

1) The first step of program procedure is load the ANN software. In this step we have to specify the path where the complete program is saved.

2) The input provided to the software, is in the form given below

```
1 0 0 0
2 0 1 1
3 1 0 1
4 1 1 0
```

Here, the 1st column specifies the serial number.

The two other columns specify the input to the ANN software. The fourth column specifies the output. This is stored in a file, which is the input file. Next, we create a test file in which the 1st and last columns are not specified. Because, this file is to test the system. Without giving the output. This file will be the unknown parameter.

3) After loading the ANN system we have to give input file for learning. If the software is validated positively, it can be used to determine the anticancer properties in unknown compounds If the output result is validated then system is being trained successfully. This trained system is used to find out anticancer activity in unknown molecules

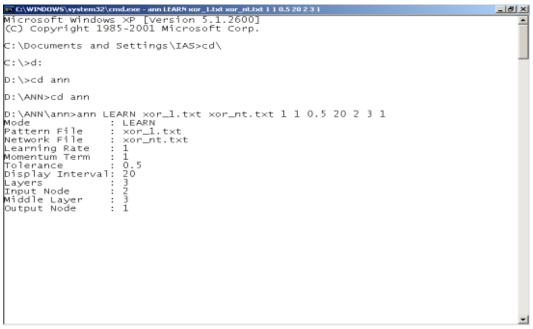
The finally designed computer software will be tested with a known system and if found correct then use for identifying anticancer property in newly or existing chemical compounds.

If the output result is validated then system is being trained successfully. This trained system is used to find out anticancer activity in unknown molecules.

8. Experimental Result

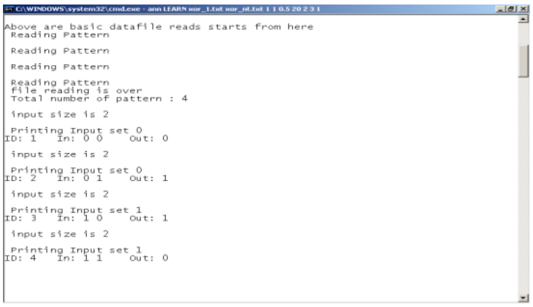
To solve the XOR problem we pass the parameters to software as

C :> ann LEARN xor_1.txt xor_nt.txt 1 1 0.5 2 0 2 3 1


C:\WINDOWS\system32\cmd.exe - ann LEARN xor_1.txt xor_nt.txt 1 1 0.5 20 2 3 1

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\IAS>cd\

C:\>d:

D:\>cd ann

D:\ANN>cd ann

D:\ANN\ann>ann LEARN xor_1.txt xor_nt.txt 1 1 0.5 20 2 3 1
Mode             : LEARN
Pattern File     : xor_1.txt
Network File     : xor_nt.txt
Learning Rate    : 1
Momentum Term    : 1
Tolerance        : 0.5
Display Interval: 20
Layers           : 3
Input Node       : 2
Middle Layer     : 3
Output Node      : 1
```

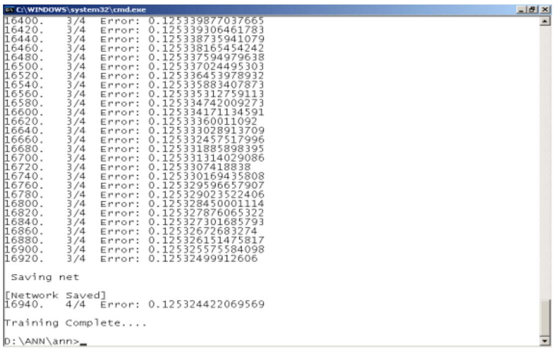
When this input is given to ANN software the following output is produce by ANN software.


C:\WINDOWS\system32\cmd.exe - ann LEARN xor_1.txt xor_nt.txt 1 1 0.5 20 2 3 1

```
Above are basic datafile reads starts from here
 Reading Pattern

 Reading Pattern

 Reading Pattern

 Reading Pattern
 file reading is over
 Total number of pattern : 4

 input size is 2

 Printing Input set 0
ID: 1    In: 0 0     Out: 0

 input size is 2

 Printing Input set 0
ID: 2    In: 0 1     Out: 1

 input size is 2

 Printing Input set 1
ID: 3    In: 1 0     Out: 1

 input size is 2

 Printing Input set 1
ID: 4    In: 1 1     Out: 0
```

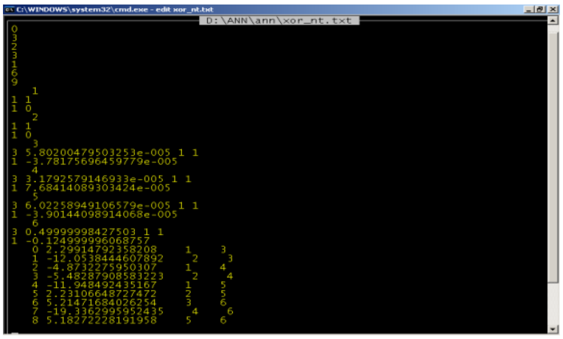
Now the network will learn. and note the error generated. And the network will be saved.

The error is 0.1253, this error is being tolerated as 0.5 as we have shown given earlier in the syntax.



```
16400.   3/4   Error: 0.125339877037665
16420.   3/4   Error: 0.125339306461783
16440.   3/4   Error: 0.125338735941079
16460.   3/4   Error: 0.125338165454242
16480.   3/4   Error: 0.125337594979638
16500.   3/4   Error: 0.125337024495303
16520.   3/4   Error: 0.125336453978932
16540.   3/4   Error: 0.125335883407873
16560.   3/4   Error: 0.125335312759113
16580.   3/4   Error: 0.125334742009273
16600.   3/4   Error: 0.125334171134591
16620.   3/4   Error: 0.12533360011092
16640.   3/4   Error: 0.125333028913709
16660.   3/4   Error: 0.125332457517996
16680.   3/4   Error: 0.125331885898395
16700.   3/4   Error: 0.125331314029086
16720.   3/4   Error: 0.1253307418838
16740.   3/4   Error: 0.125330169435808
16760.   3/4   Error: 0.125329596657907
16780.   3/4   Error: 0.125329023522406
16800.   3/4   Error: 0.125328450001114
16820.   3/4   Error: 0.125327876065322
16840.   3/4   Error: 0.125327301685793
16860.   3/4   Error: 0.12532672683274
16880.   3/4   Error: 0.125326151475817
16900.   3/4   Error: 0.125325575584098
16920.   3/4   Error: 0.12532499912606

 Saving net

[Network Saved]
16940.   4/4   Error: 0.125324422069569

Training Complete....

D:\ANN\ann>_
```

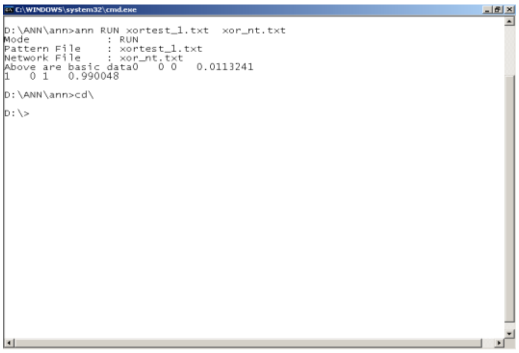
After getting learned the network file will be created automatically by the software. It will create some weight which will be later used to test the network.



```
0
3
2
3
1
6
9
   1
1 1
1 0
   2
1 1
1 0
   3
3 5.80200479503253e-005 1 1
1 -3.78175696459779e-005
   4
3 3.1792579146933e-005 1 1
1 7.68414089303424e-005
   5
3 6.02258949106579e-005 1 1
1 -3.90144098914068e-005
   6
3 0.49999998427503 1 1
1 -0.124999996068757
   0 2.29914792358208     1    3
   1 -12.0538444607892     2    3
   2 -4.8732275950307     1    4
   3 -5.48287908583223     2    4
   4 -11.948492435167     1    5
   5 2.23106648727472     2    5
   6 5.21471684026254     3    6
   7 -19.3362995952435     4    6
   8 5.18272228191958     5    6
```

This is the network file created. It stores some weight between input node and hidden node. And also between hidden and output node. In this way the network is created.

While testing the network a testing file is being given as input



```
D:\ANN\ann>ann RUN xortest_1.txt  xor_nt.txt
Mode           : RUN
Pattern File   : xortest_1.txt
Network File   : xor_nt.txt
Above are basic data0   0 0   0.0113241
1   0 1   0.990048

D:\ANN\ann>cd\

D:\>
```

The value came after testing the network is very close to the expected. That is the output of

0 0 should be 0 and

0 1 should be 1

By using ann output came

0 0 0.113

0 1 0.990

This output is nearly exact to the real one.

9. Conclusion

Hence, we have concluded that, any unknown sample of data can be found out by the use of unsupervised, back propagation neural network, also. Can find the value very near to accurate. So, we performed the learning and testing of network using ANN and test the result using X-OR logic. And got the result for it successfully.

10. References

[1] R.P.Lippman, “An introduction to computing with neural nets,”IEEE ASSP Magazine, pp 4-22.

[2] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003

[3] J.-K. Kamarainen, V. Kyrki, and H. K¨alvi¨ainen. Invariance properties of gabor filter based features - overview and applications. IEEE Transactions on Image Processing

[4] L. M. Harrison, O. David, and K. J. Friston, “Stochastic models of neuronal dynamics,” Philosophic Transactions of the Royal Society B, vol. 360, pp. 1075–1091, 2005.

[5] E. Mizutani, S.E. Dreyfus, and K. Nishio. On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application. In Proc. of the IEEE International Conference on Neural Networks (vol.2), pages 167–172, Como, Italy, July 2000.

[6] D.Palmer-Brown and M.Kang, “ADFUNN:An Adaptive Function Neural Network,” to appear in the 7th International Conference on Adaptive and Natural Computing Algorithms,Coimbra(Portugal), 2005

[7] M.Kang and D.Palmer-Brown, “An Adaptive Function Neural Network (ADFUNN) for Phrase Recognition,”, the International Joint Conference on Neural Networks (IJCNN05),Montral, Canada, 2005.

[8] J. M. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann,F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, *et al.*,“Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,” *Nature Medicine*, vol. 7, pp. 673- 679, 2001.

[9] D.Palmer-Brown and M.Kang, “ADFUNN:An Adaptive Function Neural Network,” to appear in the 7th International Conference on Adaptive and Natural Computing Algorithms,Coimbra(Portugal), 2005.

[10] J. Timmis and M. J. Neal, “A resource limited artificial immune system for data analysis,” Research and Development in Intelligent Systems XVII, pp. 19–32, December 2000.

[11] A. Pujol, H. Wetzel, and J.J. Villanueva. Learning and caricaturing the face space using self-organization and hebbian learning for face processing. In *International Conference on Image Analysis and Processing*, pages 273–278, 2001.

[12] X. Ma, R. Salunga, J. T. Tuggle, J. Gaudet, E. Enright, P. McQuary, T. Payette, M. Pistone, K. Stecker, B. M. hang, et al., “Gene expression profiles of human breast cancer progression,” Proc. Natl. Acad. Sci. USA, vol. 100, pp. 5974-5979, 2003.

[13] Kewley, R.H.; Embrechts, M.J.; Breneman, C. IEEE T Neural Network, **2000**, 11, 668.

[14] Arciniegas, F.; Bennett, K.; Breneman, C.; Embrechts, M.J.Molecular database mining using self-organizing maps for thedesign of novel pharmaceuticals. ANNIE Conference, St. Luis, Missouri, November **2001**, ASME: **2001**.

[15] X. Fu, and L. Wang, "Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance," IEEE Trans. Systems, Man, Cybernetics-Part B: Cybernetics,
vol. 33, pp. 399-409, 2003.

[16] N.J. Pizzi, "Classification of biomedical spectra using stochastic feature selection,"

[17] M. E. Tipping, The Relevance Vector Machine. Advances in Neural Information Processing Systems, 2000.