Studies in artificial neural networking model

ABSTRACT:

"What does the reader do when he wishes to see in what the precise likeness or difference of two objects lies He transfers his attention as rapidly as possible, backwards and forwards, from one to the other. The rapid alteration of consciousness shakes out, as it were, the points of difference or agreement, which would have slumbered forever unnoticed if the consciousness of the objects compared had occurred at widely distant periods of time. H/hat does the scientific man do when he searches for the reason or law embedded in a phenomenon? He deliberately accumulates all the instances he can find which have any analogy to that phenomenon; and by simultaneously filling his mind with them all, he fre quently succeeds in detaching from the collection the peculiarity which he was unable to formulate in the one alone; even though that one had been preceded in his former experience by all those with which he now at once confronts it".

Introduction:

With the advent of more and more sophisticated devices, the amount of data in demand for processing is exponentially increasing. As for example, the recent space surveys are capable of bringing in Terra bytes of data for processing. Needless to say, the increased number of International conferences and journal titles on machine learning reveal the wider interest and research potential in this relatively new area of science. The last two decades particularly witnessed great enthusiasm for machine learning in connection to data mining. Although the original context of data mining refers to attempts made in extracting irrelevant (only statistically significant) information from insufficient data, it is now popular as a method for identifying the underlying physics in those systems. Usually, data mining involves the four steps.

- **Gathering of raw data:** This could be the data on rainfall recorded in a collection center - there may be errors in it.
- **Data Cleaning:** In this process, any bogus data is removed from the collected data. In the above example, this could happen due to the carelessness of the observer or the recording method.
- **Feature identification:** Not all data is required for processing. By careful examination, an expert can identify key parameters or features in the data that is relevant for a particular study based on the data.
  Data Mining: Extraction of decisive information - knowledge discovery - from the data. Neural Networks are best suited for this purpose due to their ability to handle large chunks of data.

It would be highly misleading to give the impression that machine learning and robots will replace humans in decision making. Here is a simple illustration. At a certain cash counter in a company, the delay in maturing a transaction was always the complaint of the customers. The company tried improving the networks with faster computers and newer .softwares - but in vain. Finally the real answer came. Somebody suggested the installation of big mirrors in front of each counter. The customer could now see himself and beautify himself while being in the queue. The customers stopped complaining about the delay in the queue!

NETWORK ARCHITECTURES

In actual practice, the training of a network may require a judicious selection of training data and a continuous monitoring of the training process by applying some stopping criteria [10]. A set of data known as validation data is used to estimate when it is best to stop the training process. It has also been shown that some pruning [16] of the connections also helps in improving the performance of networks. A simple criterion for pruning is to remove all connections that do not affect the performance of a network when it is temporally removed from the network. The reverse process is also possible, that is, to add a node or a connection if that would improve the performance of the network. Such networks are generally known as Adaptive networks [23].

NETWORKING MODELS

Perceptrons

Perceptrons are the simplest of the network models and was proposed by Rosenblatt in 1958 [25, 21] to classify linearly separable data by a learning process. The perceptron forms a network with a single node and a set of input connections along with a dummy input which is always set to 1 and a single output lead. The input pattern which could be a set of numbers is applied to each of the input connections to the node. The perceptron learning algorithm updates the strength of each connection to the node (also known as weight or gain) in such a way that the output from the node happens to be within some threshold value for each class represented by the input data patterns. Thus the perceptron equation for class label 0k is

$$C_k = w_o + w_1 I_1 + w_2 I_2 + w_n I_n \ldots\ldots \quad (1.1)$$

Fully connected networks

The most general form of a network is one in which there exists an independent connection between a given node and every other node in the network. It is also possible that the connection weights between two nodes for the forward and reverse connections are different. Such networks are called asymmetric fully connected networks. In spite of its structural simplicity, such networks are of little practical significance. The major reason for that is the large number of parameters. In a network with n nodes, there are n2 possible connection weights and as many connections. Having many parameters means more elements to memorize and thus if they are used, they are used as memory elements. A more practical version is a symmetric network that has the same weights in both the forward and reverse connections between nodes.

Layered networks

It is possible to consider nodes as situated on various layer& The layer that holds the input nodes is called the input layer and the layer that holds the output nodes is called the output layer. All other layers are called hidden layers. Between layers, interconnections are allowed only between nodes in the same layer and to the nodes in the following layers. Thus there is no connection in the reverse direction from a node in a layer closer to the output node to the layer closer to the input node. Such networks are called layered networks.

A slightly different network in which there is no connection between nodes in the same layer is called an Acyclic Network. However it is still possible for the nodes in a layer to have forward connection to a node in any other layer. If this flexibility is removed and the node in a layer is allowed to make connections with nodes in the following layer only, the model is called a Feed Forward Network . Feed forward networks are the most popular ones in use and the term neural network itself is often used as a synonym for feed forward networks.

Modular neural networks

It is known that there are different kind of neurons in the cortex of the brain. Modular networks [131 allow different networks with sparse interconnection between networks (modules) to form a large complex network. Each module in such a network analyses a different aspect of the data stream and transfers the output to another module which might accept several such inputs to do further processing. Modularity can be either in the input layer where a module accepts a part of the input pattern to the network and passes it to the subsequent modules. Another possibility is to have modularity in the subsequent layers of the network such that the output from a node is processed by different modules in the hidden layers to associate some probability for their membership in any of the possible output class labels.

Optimizing The Transfer Function Of A BP Network

In this chapter we discuss a method to improve the performance of a standard back-propagation network. The method is applicable to any network using back- propagation for learning. As we discussed in the previous chapter, the sigmoid transfer function has many advantages. However, it also has the disadvantage that it builds the network assuming that the cost function has a sigmoid shape. This is not always the case and the result is slow convergence and reduced accuracy. The goal of this chapter is to introduce a simple method to modify the sigmoid function that it may adjust itself to the data it is trained with. This modification, although involves a bit more computations, is found to improve the learning ability and generalization ability of the network considerale.

Prediction of Rainfall In Kerala:

An Application of ABF Neural Network

In this section we present one application of the ABFN. As mentioned in the introduction, rainfall is a very irregular phenomenon over short intervals of time. But, taken over long periods, it follows some regular patterns in most places in the Indian peninsula. In this example, we try to predict the rainfall over a period of 47 years by training the network with the monthly rainfall received in the 40 previous years. An interesting consequence of the study is the observation that in spite of fears of global warming and such, the rainfall pattern in this corner of the globe remains more or less unaffected

Bayesian Theorem

The conditional probability of an event A assuming that the event B has occurred is denoted by P (AIB). By the definition of likelihood, it is the fraction of the whole that has resulted in a particular outcome. Thus the likelihood can be represented as:

$$\textit{likelihood} = P(A \cap B) \qquad (3.1)$$

In Venn diagram representation, conditional probability may be expressed as

$$F(A|B) = \frac{P(A \cap B)}{P(B)} \qquad (3.2)$$

Where P(B) represents the probability for the event B to happen. The derivation of Bayesian theorem is now straight forward as shown below.

$$P(A \cap B) = P(A|B)P(B) \qquad (3.3a)$$

But $P(A \cap B)$ is same as $P(B \cap A)$.

So, it is also correct to say that

$$P(B \cap A) = P(B|A)P(A) \quad (3.3b)$$

DIFFERENCE BOOSTING NEURAL NETWORK:

Boosting in DBNN

Our method differs from the AdaBoost algorithm in that, instead of a series of classifiers we use the same classifier throughout the training process. In each round the conditional probability $P(U_m|C_k)$ for each attribute of the misclassified examples is enhanced using a multiplicative weight function W_m. An error function is defined for each of the misclassified examples based on its distance from the computed probability of its nearest rival. The enhancement to the attribute is done such that the error produced by each example decides the correction to be applied to its associated weights. Since it is likely that more than one class would be sharing at least some of the same attribute values, this would lead to a competitive update of their attribute weights, until either the classifier figures out the correct class or the training round completes. However, for attributes that do not overlap, there will be a constant upward weight revision in each round. The net effect of this would be that the classifier will become more and more dependent on the differences in the examples rather than their similarities. This is analogous to the way in which the human brain differentiates between almost similar objects by sight, like for example, rotten tomatoes from a pile of good ones.

The classifier network

The network presented here may be divided into three units. The first unit computes the Bayes' probability and the threshold window function for each of the training examples. If there are *M* number of attributes with values ranging from m_{min} to m_{max} and belonging to one of the *K* discrete classes, we first construct a grid of equal sized

bins for each *k* with columns representing the attributes and rows their values. Thus a training example S_i belonging to a class *k* and having one of its attributes 1 with a value in will fall into the bin B_{klm} for which the Euclidean distance between the center of the bin and the attribute value is a minimum. The number of bins in each row should cover the range of the attributes from m_{min} to m_{max}. The training process is simply to distribute the examples in the training sets into their respective bins. After this, the number of examples in each bin i for each class *k* is counted and this gives the probability $P(U_m|C_k)$ of the attribute *m* with value $U_m \equiv i$ for the given $C_k = k$.

Experimental results:

Wisconsin breast cancer databases

The Wisconsin breast cancer database represents a reasonably complex problem with 9 continuous input attributes and two possible output classes. This data set was donated by William H. Wolberg of the University of Wisconsin hospitals [53]. The dataset consists of 683 instances and we divided it into a training set of 341 examples and a test set of 342 examples each. The problem is to find whether the evidences indicate a benign or malignant neoplasm. Wolberg used 369 instances of the data (available at that point in time) for classification and found that two pairs of parallel hyperplanes are consistent with 50% of the data. Accuracy on the remaining 50% of the dataset was 93.5%. It is also reported that three pairs of parallel hyperplanes were found to be consistent with 67% of data and the accuracy on remaining 33% was 95.9% The input attributes are are shown in table 4.3

Taha and Ghosh [54] used all the 683 instances to test a hybrid symbolic-connectionist system. Using a Full Rule Extraction algorithm, they report a recognition rate of 97.77%. The network described in this work, using 8 bins

for each attribute, converged in 87 iterations to produce a classification accuracy of 97.95% on the independent test set. Only seven out of 342 examples were misclassified.

Table 4.3: The parameters used for Wisconsin breast cancer detection problem.

Attribute	Type
Clump Thickness	Continuous
Uniformity of Cell Size	Continuous
Uniformity of Cell Shape	Continuous
Marginal Adhesion	Continuous
Single Epithelial Cell Size	Continuous
Bare Nuclei	Continuous
Bland Chromatin	Continuous
Normal Chromatin	Continuous
Mitoses	Continuous

Thyroid databases

The thyroid database was donated by Randolf Werner in 1992. It consists of 3772 learning examples and 3428 testing examples readily usable as ANN training and test sets. In the repository, these datasets are named : pub/machine-learningdatabases/thyroid disease/ann*. Each example has 21 attributes, 15 of which are binary and 6 are continuous. The problem is to determine whether a patient referred to the clinic is hypothyroid. The output from the network is expected to be one of the three possible classes, namely: (i) normal (not hypothyroid), (ii)

hyperfunction and (iii) subnormal function. In the dataset, 92 percent of the patients are not hyperthyroid and thus any reasonably good classifier should have above 92% correct predictions. Schiffmann et at., [55] used this dataset to benchmark 15 different algorithms. Fourteen of the networks had a fixed topology of 3 layers with 21 input nodes, 10 hidden nodes and 3 output nodes. The network was fully interconnected. The other network used was a cascade correlation network with 10 and 20 units each. Using a SPARC2 CPU, the reported training time on the dataset varied from 12 to

4.4.3 Pima Indians diabetes database

The Pima Indian diabetes database, donated by Vincent Sigillito, is a collection of medical diagnostic reports of 768 examples from a population living near Phoenix, Arizona, USA. The paper dealing with this data base [56] uses an adaptive learning routine that generates and executes digital analogs of perceptron-like devices, called ADAP. They used 576 training instances and obtained a classification of 76% on the remaining 192 instances. The samples consist of examples with 8 attribute values and one of the two possible outcomes, namely whether the patient is tested positive for diabetes (indicated by output one) or not (indicated by two). The database now available in the repository has 512 examples in the training set and 256 examples in the test set. The attribute vectors of these examples are shown in table 4.5

Table 4.5: Attribute vectors used for the Pima Indians diabetes database study

Attribute	**Type**
Number of times pregnant	Continuous
Plasma glucose concentration	Continuous
Diastolic blood pressure (mm Hg)	Continuous
Triceps skin fold thickness (mm	Continuous
2-Hour serum insulin (mu U/ml)	Continuous
Body mass index [weight in kg/(height in m)^2	Continuous
Diabetes pedigree function	Continuous
Age (years)	Continuous

Conclusion:

Bayes' rule on how the degree of belief should change on the basis of evidences is one of the most popular formalism for brain modeling. In most implementations, the degree of belief is computed in terms of the degree of agreement to some known criteria. However, this has the disadvantage that some of the minor differences might be left unnoticed by the classifier. We thus devise a classifier that pays more attention to differences rather than similarities in identifying the classes from a dataset. In the training epoch, the network identifies the apparent differences and magnifies them to separate out classes. We applied the classifier on many practical problems and found that this makes sense. To illustrate some of the features of the network, we discuss four examples from the UCI repository. The highlights of our network are:

1. In all the examples the classification accuracy in both training and testing sets are comparable. This means that the network has successfully picked up the right classification information avoiding any possible over-fitting of data.
2. Unlike back propagation or its variant, the network converges to the same accuracy irrespective of initial conditions.
3. The training time is less compared to other networks and the accuracy is more.
4. The network topology may be optimized using parallel computation and the network is well suited for parallel architecture offering high throughput.

REFERENCES:

[1] R. Rojas, *Neural Networks: A Systematic Introduction, Springer-Verlag*, Berlin (1996).

[2] E. Lorenz, *Predictability: Does the Flap of a Butterfly's Wings in Brazil Set Off a Tornado in Texas*?, AAAS Convention Of the Global Atmospheric Research Program, MIT, Dec. 29, (1972).

[3] Kathleen T. Alligood, Tim D. Sauer and James A. *Yorke, Chaos: An Introduction to Dynamical System*, Springer-Verlag, New York, (1997).

[4] Nigel P. Cook, *Introductory Digital Electronics*, Prentice Hall, (1997).

[5] Robert Groth, *Data Mining: Building Competitive Advantage*, Prentice Hal, (2000).

[6] Safavian, S. R., and D. Langrebe, *A survey of Decision Tree Classifier Methodology*, IEEE Transactions on Systems, Man and Cybernetics 21:660-674, (1991).

[7] F. Murtagh, *Multidimensional Clustering Algorithms*, Physica-Verlag, Heidelberg, (1985), (ISBN 3 7051 0008 4).

[8] E. Gelenbe, *Learning in the Recurrent Random Neural Network*, Neural Computation, Vol. 5, No. 1, pp 154-164, (1993).

[9] A. M. Arbib, The Handbook of Brain Theory and Neural Networks. Cambridge, MA: MIT Press, (1995).

[10] R. Y. Shao, S. Lin, and M. P. Fossorier, *Two Simple Stopping Criteria for Turbo Decoding*, IEEE Transactions on Communications, 47(8):1117—1120, (Aug. 1999).

[11] Simon Haykin, Neural Networks: A *Comprehensive Foundation*, Prentice Hall Press, (1998).

[12] M. Marchand, M. Golea, P. Rujan, *A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons*, Europhysics Letters, 11(6), 487—492, (1990).

[13] Frederic Gruau, Automatic Definition of Modular Neural Networks, Adaptive Behavior, 3:151-183, (1995).

[14] F. J. Pineda, *Generalization of Back-Propagation to Recurrent Neural Networks*, Physics Review Letters 59, 2229—2232, (1987).

[15] M. M'ezard and J. P. Nadal. *Learning in Feedforward Layered Networks*: The Tiling Algorithm, J. of Physics: A, 22(12):2191—2203, (1989).

[16] B. Hassibi, D. G. Stork, G. J. Wolff, *Optimal Brain Surgeon and General Network Pruning*, IEEE International Conference on Neural Networks, Volume 1, pages 293—299, (1993).

[17] Y. Le Gun, J. S. Denker, S. Solla, *Optimal Brain Damage*, Advances in Neural Information Processing Systems, 2:598—605, (1990).

[18] Scott E. Fahlmann and Christian Lebiere, *The Cascade-Correlation Learning Architecture*, Technical Report CMU-CS-90-100, Carnegie Mellon University, (1990).

[19] A. Sankar, R.J. Mammone, *Optimal Pruning of Neural Tree Networks for Improved Generalization*, in "IJCNN-91-SEATTLE: International Joint Conference on Neural Networks", IEEE Press, Seattle, WA, pp. II: 219—224, (1991).

[20] M. Freari, *The Upstart Algorithm: A Method for Constructing and Training Feed forward Neural Networks*, Neural Computation, 2(2): 198 – 209, (1990).

[21] Principe, Euliano, Lefebvre *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley and Sons, (1999).

[22] Anthony Zaknich, *Artificial Neural Networks*: An Introductory Course available online at http:// www. maths. uwa. edu. au / rkea1ley / annall / index. html

[23] Kohonen, Teuvo, *Self-Organization and Associative Memory*, Springer Verlag, (1989).

[24] S. Amari. *Theory of adaptive pattern classifiers*, IEEE Trans. on Elect. Comput., 16(3):299 – 307, (1967).

[25] Rosenblatt Frank, The Perceptron: *A Probabilistic Model for Information Storage and Organization in the Brain*, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408, (1958).

[26] B. Widrow, *Adaline and Madaline*, In First International Conference on Neural Networks, pages 143—158, IEEE Computer Society Press, (1987).

[27] S.I. Gallant, *Perceptron Based Learning Algorithms*, IEEE Transactions on Neural Networks, 1:179 – 191, (1990).

[28] M. Widrow Lehr, *30 Years of Adaptive Neural Networks*: Perceptron, Adaline, Madaline and Backpropagation, in Proc. IEEE, vol. 78, no. 9, Sept. (1990).

[29] J. C. David MackKay, *Bayesian Methods for Adaptive Models*, PhD thesis, California Institute of Technology, (1992).

[30] Harold Jeifreys, *Theory of Probability*, Oxford University Press, (1939).

[31] David Heckerman, *A Tutorial on Learning with Bayesian Networks*, Technical Report MSR-TR-95-06, (1996).

[32] T. J. Loredo, in *Maximum Entropy and Bayesian Methods*, (Ed. P.]?. F'ougere), Kiuwer Academic Publishers, Dordrecht, (1990).

[33] David B. Fogel, *Using Evolutionary Programming to Create Neural Networks that are Capable of Playing Tic- Tac- Toe*, In Proceedings of International Conference on Neural Networks, (1993).

[34] Giulio D'Agostini, *Probability and Measurement Uncertainty in Physics - a Bayesian Primer*, Universita "La Sapienza" and INFN, Roma,Italy, (1995).

[35] Honavar. V and L. Uhr*, Generative Learning Structures and Processes for Generalized Connectionist Networks*, Information Sciences, 70:75-108, (1993).

[36] International Organization of Standardization (ISO), *Guide to the expression of uncertainty in measurement, Geneva*, Switzerland, (1993).

[37] E. T Jaynes, Probability Theory: *The Logic of Science*, Wayman Crow Professor of Physics, Washington University, St. Louis, USA, (1995), available on line from http://ayes.wustl.edu/etj/prob.html

[38] D. B. Rumelhart, 0. B. Hinton, and R. J. Williams. Learning Representations by Back-Propagation of Errors. Nature, 323:533—536, (1986).

[39] D. E. Rumeihart, 0. B. Hinton, R. J. Williams, Learning Internal Representations by Error Propagation, *in Parallel Distributed Processing* (D.E. Rumelhart, J.L, McClelland, eds.), MIT Press, 318-362, (1986).

[40] Fahlman, S. B. (1988) *Faster-Learning Variations on Back-Propagation*: An Empirical Study in Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann, (1988).

[41] M. F. Mller, *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*, Neural Networks 6, 525 (1993).

[42] D. E. Rumeihart, R. Durbin, R. Golden, Y. Chauvin, Backpropagation: The Basic Theory, *in Backpropagation: Theory, Architectures and Applications* (Y. Chauvin and D.E. Rumelhart, eds.), Lawrence Erlbaum, (1993).

[43] R. 0. Duda, P.E Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, (1973).

[44] P. Langley, W. Iba, K. Thompson, An Analysis of Bayesian Classifiers, in *Proceedings of Tenth National Conference on Artificial Intelligence*, Menlo Park, CA:AAAI Press,223-228, (1992).

[45] C. Elkan, Boosting and Naive Bayesian Learning, *Technical Report No. C597-557*, University of California, SanDiego, (1997).

[46] C. lvi. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, (1999).

[47] M. Minsky, S. Papert, *Perceptrons: an Introduction to Computational Geometry*, MIT Press, (1969).

[48] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting, *in Proceedings of Second European Conference on Computational Learning Theory*, 23-37, (1995).

[49] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting, *Journal of Computer and System Sciences*, 55(1), 119-139, (1997).

[50] J. Gama, Iterative Naive Bayes, Discovery Science99, LNAI, 1721 Springer Verlag, 80-91, (1999).