APPLICATION OF NEURAL NETWORKS IN PREDICTIVE DATA MINING

Saratha Sathasivam *

Neural Networks represent a meaningfully different approach to using computers in the workplace. A neural network is used to learn patterns and relationships in data. The data may be the results of a market research effort, or the results of a production process given varying operational conditions. Regardless of the specifics involved, applying a neural network is a substantial departure from traditional approaches. In this paper we will look into how neural networks is used in data mining. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications. Therefore, we will consider how this technique can be used to classify the performance status of a departmental store in monitoring their products.

Neural networks, data mining, prediction

1.0 Introduction

Neural Networks 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 unique structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in harmony to solve specific problems. Neural Networks, like people, learn by example. A neural network is configured for a specific application through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons; neural networks use the same as well. Neural networks have demonstrated their success in many applications due to their ability to solve some problems with relative ease of use and the model-free property they enjoy. These can solve problems without the need to understand or learn the analytical and statistical properties of neither the problem nor the solution steps.

There is also a strong potential for using neural networks for database mining that is, searching for patterns implicit within the explicitly stored information in databases. Most work in this area is applying neural networks, such as the Hopfield-Tank network for optimization and scheduling. One particular example of use of neural networks in data mining is as follows: Classification is one of the data mining problems receiving great attention recently. For that the approach of symbolic classification rules using neural networks has been appreciated. With the proposed approach, concise symbolic rules with high accuracy can be extracted from a neural network.

In this paper, the data mining based on the reverse analysis algorithm is analyzed in detail. In section 2, we will discuss data mining. Later in section 3, we will focus on Hopfield neural network. Next, we will explained on how rules can be extracted from a data base. Following that, algorithm of reverse analysis algorithm is discussed. Finally, conclusion and discussion occupied the last section.

2.0 Data Mining

Data mining automates the process of sifting through historical data in order to discover new information. This is one of the main differences between data mining and statistics, where a model is

* Saratha Sathasivam, School of Mathematical Sciences, Universiti Sains Malaysia. Email: *saratha@cs.usm.my*

usually devised by a statistician to deal with a specific analysis problem. It also distinguishes data mining from expert systems, where the model is built by a knowledge engineer from rules extracted from the experience of an expert.

Data mining tools can sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions, performance bottlenecks in a network system and identifying *anomalous data* that could represent data entry keying errors. The ultimate significance of these patterns will be assessed by a domain expert - a marketing manager or network supervisor - so the results must be presented in a way that human experts can understand (Michael and Gordon 2004, Michalski and Kubat 1998).

Neural networks are an approach to computing that involves developing mathematical structures with the ability to learn. The methods are the result of academic investigations to model nervous system learning. Neural networks have the remarkable ability to derive meaning from complicated or imprecise data and 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. Neural networks have broad applicability to real world business problems and have already been successfully applied in many industries.

3.0 Hopfield Network

The Hopfield model (Hopfield 1982, 1984, 1985) is a standard model for associative memory. The Little dynamics is asynchronous, with each neuron updating their state deterministically. The system consists of *N* formal neurons, each of which is described by an Ising variable $S_i(t),(i=1,2,....N)$. Neurons then are bipolar, $S_i \in \{-1,1\}$, obeying the dynamics $S_i \to \mathrm{sgn}(h_i)$, where the field, $h_i = \sum_j J_{ij}^{(2)} V_j + J_i^{(1)}$, *i* and *j* running over all neurons *N*, $J_{ij}^{(2)}$ is the synaptic strength from neuron *j* to neuron *i*, and $-J_i$ is the threshold of neuron *i*.

Restricting the connections to be symmetric and zero-diagonal, $J_{ij}^{(2)} = J_{ji}^{(2)}$, $J_{ii}^{(2)} = 0$, allows one to write a Lyapunov or energy function,

$$E = -\frac{1}{2}\sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \tag{1}$$

which monotone decreases with the dynamics.

The two-connection model can be generalized to include higher order connections. This modifies the "field" to be

$$h_i = ...+\sum_j \sum_k J_{ijk}^{(3)} S_j S_k + \sum_j J_{ij}^{(2)} S_j + J_i^{(1)} \tag{2}$$

where "….." denotes still higher orders, and an energy function can be written as follows:

$$E = ...-\frac{1}{3}\sum_i \sum_j \sum_k J_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \tag{3}$$

provided that $J^{(3)}_{ijk} = J^{(3)}_{[ijk]}$ for i, j, k distinct, with [...] denoting permutations in cyclic order, and $J^{(3)}_{ijk} = 0$ for any i, j, k equal, and that similar symmetry requirements are satisfied for higher order connections. The updating rule maintains

$$S_i(t+1) = \mathrm{sgn}[h_i(t)] \tag{4}$$

4. The Logic of Hebbian Learning

For two-neuron connections, a Hebbian-like learning is given by (Wan Abdullah, 1993)

$$\Delta T_{ij} = \alpha\,(2V_i - 1)(2V_j - 1) \tag{5}$$

(or, for bipolar neurons, $\Delta T_{ij} = \alpha S_i S_j$), where α is a learning rate. For connections of other orders, we can generalized this to

$$\Delta T_{ij....m} = \alpha(2V_i - 1)(2V_j - 1)..(2V_n - 1) \tag{6}$$

Assume we have events $A, \bar{A}, B, \bar{B}, C, \bar{C}, ...$ occurring randomly but equally probably: V_a ,etc. are randomly 0 or 1 with equal probabilities. As such, there would be no nett change in connection strengths, because $\Delta T_{ij...n}$ has equal probability of being positive as well as being negative.

Now say for example $\bar{D}$ does not occur. This would result in ΔT_D being positive, which is equivalent to, according to our analysis in the previous section, the assertion $D \leftarrow .$ being learnt. In this case, our system has learnt a rule which corresponds to only *D* occurring.

For the case of *C* occurring when *D* occurs, example *CD* occurs, $C\bar{D}$ occurs, $\bar{C}D$ does not occur, $\bar{C}\bar{D}$ occurs, there is a nett increase in $\Delta T_{[CD]}$ and a nett decrease of the same magnitude in ΔT_D. This is equivalent to the rule $C \leftarrow D.$ being learnt, which the rule giving events like those is observed. However, there is also an increase in ΔT_C which means that $C \leftarrow .$ has also been learnt.

To clarify further, let us look at the case where *A* occurs if *B* and *C* both do. Then the following table summarizes what happens:

	$\Delta T_{[ABC]}$	$\Delta T_{[AB]}$	$\Delta T_{[AC]}$	$\Delta T_{[BC]}$	ΔT_A	ΔT_B	ΔT_C
ABC occurs	+	+	+	+	+	+	+
$AB\bar{C}$ occurs	–	+	–	–	+	+	–
$A\bar{B}C$ occurs	–	–	+	–	+	–	+
$A\bar{B}\bar{C}$ occurs	+	–	–	+	+	–	–
$\bar{A}B\bar{C}$ occurs	–	–	–	+	–	+	+
$\bar{A}B\bar{C}$ occurs	+	–	+	–	–	+	–
$\bar{A}\bar{B}C$ occurs	+	+	–	–	–	–	+

$\bar{A}\bar{B}\bar{C}$ occurs	–	+	+	–	–	–	–
nett	+	+	+	-	+	-	-
factor	x 6(-1/3)	x 2(-1/2)	x 2(-1/2)	x 2(-1/2)	x 1(-1)	x 1(-1)	x 1(-1)

The net change is multiplied by the number of terms in the energy giving the same contribution (the various permutations of the subscripts) and the factor associated with each term in the energy function. The system "correctly" learns $A \leftarrow B, C.$ but also the extra rules $A \leftarrow B.$, $A \leftarrow C.$, $A \leftarrow .$ and so on.

If we use bipolar neurons, the energy change to include the clause $C \leftarrow D.$ is

$$\Delta E = \frac{1}{4}(1 - S_c)(1 + S_D) \tag{7}$$

Thus the events {CD, $C\bar{D}$, $\bar{C}\ \bar{D}$} correctly give the corresponding change in energy without the spurious clause as with binary neurons. This may be expected as the change in variable is effectively an overall change in the neural threshold values.

For $A \leftarrow B, C.$ the change in energy with bipolar neurons is

$$\Delta E = \frac{1}{8}(1 - S_A)(1 + S_B)(1 + S_C) \tag{8}$$

The collection of events {ABC, $AB\bar{C}$, $A\bar{B}C$, $A\bar{B}\bar{C}$, $\bar{A}B\bar{C}$, $\bar{A}\bar{B}\bar{C}$, $\bar{A}\bar{B}\bar{C}$} yields the learning of $A \leftarrow B, C.$ plus the extra energy term $-S_A S_B S_C$ which causes the system to have a liking for all *A*, *B*, and *C* to be true.

5. Reverse Analysis Method

In this paper, we define a machine learning method: Reverse Analysis Method (Saratha and Wan Abdullah 2008, Saratha 2007), that uses Hopfield neural network to discover trends in datasets. They are few methods of rules extraction strategy such as subset method (Bochereau and Bourgine, 1990), M of N method (Towell and Shavlik 1993), RULEX method (Andrews and Geva 1994) and few other methods. However by using Reverse Analysis method we can learn the inherent relationships among the data. In another words, by using this method we can extract common trends that exists in the dataset as well as to predict trends. The extraction phase can be regulated through several control parameters, such as number of trials, number of energy relaxation loops etc. Furthermore, this method is additive and need less computation effort as illustrated in previous section.

This method consists of these following steps:

i) Enumerate number of neurons and patterns in the database.
ii) Initialize number of trials, energy relaxation loops, number of patterns.
iii) Extract the events from the database and represent in binary/bipolar pattern, where 0 indicates false state and 1 indicates true state (for bipolar -1 represent false state and 1 represent true state).
iv) Calculate the connection strengths for the events using Hebbian learning as described in section 3.
v) List out all the connection strengths obtained for third order connections($\Delta T_{[ABC]}, \Delta T_{[AB]}, \Delta T_{[AC]}, \Delta T_{[BC]}, \Delta T_A, \Delta T_B, \Delta T_C$), second order

connections($\Delta T_{[AB]}, \Delta T_{[AC]}, \Delta T_{[BC]}, \Delta T_A, \Delta T_B, \Delta T_C$) and first order connections ($\Delta T_A, \Delta T_B, \Delta T_C$).

vi) Capture all the nonzero values (connection strengths) for third order connection.

vii) By using the method in section 3, list out all the corresponding clauses for (vi).

viii) Calculate the connection strengths for the extracted clauses in step (vii) and deduct the value of the corresponding clauses connection strengths from (v).

ix) Repeat the similar steps to extract the clauses corresponding to the first order and second order connections.

6. Simulation

In this section we will look into application of Reverse Analysis method in monitoring stock in a departmental store. Following table 1, shows customers purchase trend in ABC store (Saratha 2010).

Table 1: Customers daily purchased from ABC store

	Bread	Jam	Butter	Sausages	Burger
Lee	√				
Michael		√	√	√	
Jonanthan	√				√
Leu		√			√
Susila	√		√	√	

By using Reverse Analysis method approached we was discussed previously, we obtained the following rules:

Bread ← Jam, Butter
Burger ← Sausages, Butter

From the rules, we can interpret that a customer who purchased butter and jam has a high probability of purchasing bread also. Meanwhile, a customer who purchased sausages and butter has a high probability of purchasing burger.

The logical rules that were induced by using Reverse Analysis method can help the departmental store in monitoring their stock according to the customers demand. Significant patterns or trends in the data set have been identified by using reverse analysis. The departmental store can apply the patterns to improve its sales process according to customers shopping trends. Furthermore, the knowledge obtained may suggest new initiatives and provide information that improves future decision making.

7. Conclusion

The goal of data mining is to unearth relationships in data that may provide useful insights. A data mining application is an implementation of data mining technology that solves a specific business or research problem. By using Reverse Analysis algorithm, we manage to unearth the underlying relationship in a departmental store data set. We have induced important logical connections in the data sets in represent it a clausal manner, which is easier to interpret. By using the induced information, the stock in the departmental store can be monitor closely with the customers demands.

8. Acknowledgement

This research is partly financed by short term grant (304/ PMATHS/631006) from Universiti Sains Malaysia.

References

Andrews, R & Geva, S. 1994. Rule extraction from a constrained error back ropagation MLP. In *Proc 5th Australian Conference on Neural Networks, Brisbane Queensland*, pp 9-12.

Bochereau, L & Bourgine, P. 1990. Extraction of semantic features and logical rules fom a multilayer neural network. In *International Joint Conference on Neural Networks, Washington*, pp 579-582.

Hopfield, J.J. 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings. Natl. Acad. Sci. USA.* , **79(8)**, pp 2554-2558.

Hopfield, J.J. 1984. Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons. *Proceeding. Natl. Acad. Sci. USA.* , **81(10)**, pp 3088-3092.

Hopfield, J.J. and Tank, D.W. 1985. Neural Computation of Decisions in Optimization Problems. *Biol. Cyber,* **52,** pp 141-152.

Michael, B & Gordon, L. 2004. *Data Mining Techniques*. New York: John Wiley & Sons, Ltd.

Michalski, R. S., Bratko, I., & Kubat, M. 1998. *Machine Learning and Data Mining:Methods and Applications*. New York: John Wiley & Sons, Ltd.

Saratha Sathasivam. & Wan Abdullah, W.A.T. 2008. *Logic Learning in the Hopfield Networks*, Modern Applied Science, 2(3), pp 57-62.

S. Sathasivam. 2007. *Logic Mining in Neural Network.* PhD Thesis, Malaysia.

Saratha Sathasivam. 2010. Extracting rules by using neural-symbolic approach. Chapter 9 in Mathematics and Economic Book by School of Mathematical Sciences, Universiti Sains Malaysia, pp 87-94.

Towell, G. & Shavlik, J. 1993. The Extraction of Refined Rules From Knowledge Based Neural Networks, *Machine Learning,* **131**, pp 77-101.

Wan Abdullah, W.A.T. 1993. The Logic of Neural Networks. *Phys. Lett. A.*, **176**. pp 202-206.