Journal of Theoretical and Applied Information Technology

www.jatit.org

BATTLEFIELD DECISION MAKING: A NEURAL NETWORK APPROACH

G.S. Gill[1] and J. S. Sohal[2]

[1]Lecturer, Institute of Instrumentation Engineering, Kurukshetra University, Kurukshetra, India-136118

[2]Formerly Dean, Faculty of Engineering, Guru Nanak Dev University, Amritsar, India.

E-mail: gaganssgill@yahoo.co.in

ABSTRACT

Neural networks (NNs) have been increasingly used in recent years for the solving complex nonlinear problems. NNs are seen as an attractive alternative to process based modeling approaches, as they are able to extract an underlying relationship from the data when knowledge of physical process is lacking. The paper evaluates the predictive power of a model, which emulates an army commander on the battlefield when encountered with various situations, using two different neural network configurations – the multilayer perceptron (MLP) and the probabilistic neural network (PNN). The proposed model may prove effective to defence scientists and commanders for Battlefield decision making, strategy development and resource management. Empirical results support the potential of PNN as a better classifier, in training data as well as test data, compared to MLP.

Keywords - *Neural networks, battlefield decision making, multilayer perceptron (MLP), probabilistic neural network (PNN).*

1. INTRODUCTION

Battlefield resource management & planning is one of the major challenges for defence analysts and scientists. The distribution of resources and deployment of optimal amount of manpower is a critical problem and choosing a winning combination is a problem faced by every commander during the war. The presence of science & technology on battlefield has given a transformed look to defence forces, which are now days highly sophisticated compared to what they had been two decades back.

The paper describes a model for battlefield resource management & planning and compares its performance with the two neural network configurations - the multilayer perceptron (MLP) and the probabilistic neural network (PNN). Neural Networks are programming paradigms that seek to emulate the microstructure of the brain, and are used extensively in artificial intelligence problems from simple pattern-recognition tasks, to advanced symbolic manipulation.

The model, currently in its developmental phase, adopts a neural networks approach to decide whether the army should advance for attack or retreat depending upon the circumstances prevailing. Although neural networks are now widely used in a growing number of fields such as political science [1], medicine [2] and engineering [3] there is little evidence of their application in the field of defence.

The data for the problem was generated from the various attributes that decide the future course of action during the war. An expert volunteered to generate the data according to the attributes presented to him. The expert was given a choice of answering, as the army commander, in terms of either ‘attack’ or ‘retreat’ when presented with two contrasting values of the attributes.

The database contained 32 samples, generated from 5 attributes, which were classified into two classes – attack and retreat. The attributes take either of the two values or states, as follows:

(1) Manpower Strength – Sufficient or Insufficient;
(2) Food & Ammunition - Sufficient or Insufficient;
(3) Infantry Support – Available or Unavailable;

(4) Air Support - Available or Unavailable;
(5) Casualty Rate - Low/Moderate or High.

These attributes identify a range of issues that might or might not become salient during war. All the major issues that motivate the army to proceed for an attack and those which force it to retreat were duly taken care of.

2. APPLIED NEURAL NETWORK STRUCTURES

2.1 Multilayer Perceptron

Multilayer perceptrons (MLPs) can act as universal approximators for a large class of nonlinear functions. Further, the learning and generalization properties of these networks have found many diverse applications [4]. Every practitioner of the MLP faces the same architecture selection problem: how many hidden layers to use and how many neurons to choose for each hidden layer? There is no foolproof recipe, at the present time, regarding the choice of number of hidden layers and neurons. The common practice is simply regarding the MLP as a magic black box and choosing a sufficiently large number of neurons such that it can solve the practical problem in hand [5].

A three-layer model with fixed learning rate was used. The input and hidden layers comprised of 5 elements each while the output layer had a single neuron. Connection weights were taken as 0.5 for, both, input to hidden layer and for hidden to output layer synapses. The back-propagation training algorithm was used to make an adaptation of connection strengths.

2.2 Probabilistic Neural Network

The PNN introduced by Specht is essentially based on the well-known Bayesian classifier technique commonly used in many classical pattern-recognition problems. Consider a pattern vector 'x' with 'm' dimensions that belongs to one of two categories K_1 and K_2. Let $F_1(x)$ and $F_2(x)$ be the probability density functions (pdf) for the classification categories K_1 and K_2, respectively. From Bayes' discriminant decision rule, 'x' belongs to K_1 if

$$\frac{F_1(x)}{F_2(x)} > \frac{L_1 P_2}{L_2 P_1} \qquad (1)$$

Conversely, 'x' belongs to K_2 if

$$\frac{F_1(x)}{F_2(x)} < \frac{L_1 P_2}{L_2 P_1} \qquad (2)$$

where L_1 is the loss or cost function associated with misclassifying the vector as belonging to category K_1 while it belongs to category K_2, L_2 is the loss function associated with misclassifying the vector as belonging to category K_2 while it belongs to category K_1, P_1 is the prior probability of occurrence of category K_1, and P_2 is the prior probability of occurrence of category K_2. In many situations, the loss functions and the prior probabilities can be considered equal. Hence the key to using the decision rules given by equations (1) and (2) is to estimate the probability density functions from the training patterns. In the PNN, a nonparametric estimation technique known as Parzen windows is used to construct the class-dependent probability density functions (pdf) for each classification category required by Bayes' theory. This allows determination of the chance a given vector pattern lies within a given category. Combining this with the relative frequency of each category, the PNN selects the most likely category for the given pattern vector. Both Bayes' theory and Parzen windows are theoretically well established, have been in use for decades in many engineering applications, and are treated at length in a variety of statistical textbooks. If the j^{th} training pattern for category K_1 is x_j, then the Parzen estimate of the pdf for category K_1 is given by equation (3) as:

$$F_1(x) = \frac{1}{(2\pi)^{m/2}\sigma^m n}\sum \exp\left[-\frac{(x-x_j)^T(x-x_j)}{2\sigma^2}\right] \qquad (3)$$

where, n is the number of training patterns, m is the input space dimension, j is the pattern number, and σ is an adjustable smoothing parameter. However, the choice of σ in general has been found to be not too sensitive to variations in its value [2].

Probabilistic neural networks (PNN) can be used for classification problems as these networks generalize well. A PNN is guaranteed to converge to a Bayesian classifier providing it is given enough training data. The only factor that needs to be selected for training is the smoothing factor, which is the deviation of the Gaussian functions - too small deviations cause a very spiky approximation that cannot generalize well; too large deviations smooth out details.

3. TEST RESULTS

Two different neural network structures, multi layer perceptron and probabilistic neural network were applied to the database to show the performance of neural networks on the data. The spread value of PNN was chosen as 1 and, the learning rate of MLP was 0.5.

Out of 32 samples, 24 were used for training and 8 were used for testing the predictive capability of the trained network. The classification results of the training data by PNN and MLP are given in the Tables I and II.

TABLE I

CLASSIFICATION OF TRAINING DATA BY PNN

Class	Attack	Retreat
True	5	19
False	0	0

TABLE II

CLASSIFICATION OF TRAINING DATA BY MLP

Class	Attack	Retreat
True	4	19
False	1	0

PNN gives the better classification accuracy with 24 correct classifications while MLP has the lower accuracy with 23 correct classifications for the training data.

A total of 8 samples were applied to the networks as test data; that is, 25% percent of the database was used for testing. 4 samples, which belonged to attack class data, and 4 samples, which belonged to retreat class, were chosen for the test. The results for PNN and MLP are shown in the Table III and IV.

TABLE III

CLASSIFICATION OF TEST DATA BY PNN

Class	Attack	Retreat
True	4	4
False	0	0

TABLE IV

CLASSIFICATION OF TEST DATA BY MLP

Class	Attack	Retreat
True	3	4
False	1	0

For the test set PNN gives the best classification accuracy with 8 correct classifications while MLP has the lower accuracy with 7 correct classifications. Overall classification performances were 100% for PNN and 87.5 % for MLP.

4. CONCLUSION

How defense analysts can use neural networks as a tool is shown in this paper. In this work, the performance of a probabilistic neural network and multilayer perceptron were investigated for battlefield management and forecasting problem. PNN proved to be a better classifier in training data as well as with test data compared to MLP since it shows the future performance of the network.

This study is based on only 5 attributes, which decide the future course of action taken by the Commander in the battlefield. So a little claim for generality can be made for any conclusions. Before any generalization could be drawn from this sort of study a much larger number of attributes would have to be examined on similar terms. Further, the attributes were analyzed at qualitative level. The performance of the model can be improved by analyzing the attributes at quantitative levels.

REFERENCES

[1] R.Eisinga, P. Franses, and V.D. Dijk, “Timing of Vote Decisions in First and Second Order Dutch Elections 1978-1995: Evidence from Artificial Neural Networks”, Political Analysis, 7, 117-142, 1998.

[2] Tuba Kiyan and T Yildirim, “Breast Cancer Diagnosis Using Statistical Neural Networks”, Journal of electrical and electronics engineering-Istanbul University, vol. 4, no. 2, pp1149-1153, 2004.

[3] Dalibor Zivkak, “Electric load forecasting using ANN”, 1999.

[4] S. Haykin, “Neural Networks—A Comprehensive Foundation”, New York: Proc. Conc., 1994.

[5] Cheng Xiang, Shenqiang Q. Ding, and Tong Heng Lee, “Geometrical interpretation and architecture selection of MLP”, IEEE transactions on neural networks, vol. 16, no. 1, pp 84-96, January 2005.