Volume 2, No. 2, Mar-Apr 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Probabilistic Neural Networks for Rule Based Systems

Nabil Hewahi
Professor of Computer Science,
Islamic University of Gaza,
Palestine
nhewahi@iugaza.edu.ps

Abstract: In this paper we introduce a theoretical approach for representing rule based system using Probabilistic Neural Networks (PNN). The proposed scheme is inspired from the statistical algorithm kernel discriminant analysis where there are four layers and works with feed forward network. This proposed approach is implemented in an algorithm called PNN-RBS. The purpose of the algorithm is to represent rule based systems using probabilistic neural networks. This mechanism is used to have a rule based system machine learning approach and to be able to produce results for unknown cases in the knowledge base. Also the approach should be capable of adding or removing new rules without retraining for the neural network.

Keywords: Probabilistic Neural Networks, Neural Networks, Machine Learning

I. INTRODUCTION

Probabilistic neural network (PNN) is one method that uses supervised learning based on feed forward network. PNN is defined as an implementation of statistical algorithm called Kernel discriminate analysis in which the operations are organized into multilayered feed forward network. The algorithm has four main layers, the first layer is the input layer, the second layer is the pattern layer, the third layer is summation layer and the fourth layer is the output layer. The input layer contains n neurons where each neuron represents one input/attribute. In the pattern layer, there are m neurons where m is the number of patterns/examples in the training set and each neuron represents one example. The summation layer contains z neurons where each neuron represents one output class and the output layer generally has one neuron which represents the output of the neural network [10][11]. The main advantages of using PNN are:

a) Fast during the training process compared to backpropagation algorithm.
b) Guaranteed to converge to optimal solution as the training set increases.
c) No local minima issues.
d) No extensive retraining is required if new samples are added or removed.

The main disadvantages of using PNN are:

a) Large memory requirements.
b) Not general as backpropagation algorithm. [3]

Because in many cases in rule based systems, we might have incomplete or partially incorrect rules, and also in many situations we may add or remove rules to the system. This addition or removing rules to the system may cause some changes in producing the output which means given some input i at time t, the system may produce output o1, but at time t1 with the input i, the system may produce output o2. In other cases with missing inputs or not known previously inputs, the system should be able to produce correct /proper output. Our main objective in this paper is to have a neural network structure that represent rule based systems to facilitate producing correct/proper output in cases of:

a) The given input is exactly as the trained set.
b) The given input is close or far from the input in the trained set.
c) Capability of adding new patterns/examples to the previously trained set and changing the produced output accordingly.

Because PNN has many features that can help us in achieving our goals, we shall use PNN to represent our rule based systems.

A. Probabilistic Neural Network

The algorithm of PNN is as below [3] [10] [11]

a) Make n neurons in the input layer where each neuron represents one attribute and n is the number of attributes..
b) Make for each pattern a neuron in the pattern layer. The patterns with the same output/class are arranged as neighbors .
c) Connect each input in the input layer with each neuron in the pattern layer.
d) Connect the neurons in the pattern layer with the same output/class to one node in the summation layer. This node represents their class.
e) Connect all the neurons in the summation layer to the neuron in the output layer. The output layer has one neuron which represents the output of neural network.

The PNN calculation is as below:

Calculate Y_{ij} (probability distribution function) for each neuron in the pattern layer, where i is the output number and j is the example number with the ith output. For example

$$Y_{11}(X) = e^{\frac{\|X-X_{11}\|^2}{\sigma^2}}$$

is the Gaussian function (G), where σ is a smoothing parameter and X is the unknown input which we want to know its output. X_{11} is the first related example to the first output/class . Similarly calculate Y_{12} for the next neurons of the same output with the same mechanism. For example

$$Y_{12}(X) = e^{-\frac{\|X-X_{12}\|^2}{\sigma^2}}$$

In the summation layer, calculate $g_i(X)$ for all the neurons related to one output/class

$$g_i(X) = \frac{1}{n_i}\sum_{k=1}^{n_i} e^{-\frac{\|X-X_{ik}\|^2}{\sigma^2}}$$

Where n_i is the number of examples related to ith class/output. This actually the average of the Y_s related to one class. This means we will have k number of g_s, where k is the number of distinct classes.

In the output layer, find the maximum of g_s obtained in the summation layer. The class with the maximum g is considered to be the output class.

$$MAX(g_1,g_2,\ldots,g_i)$$

Figure 1 shows the structure of the neural network used in PNN.

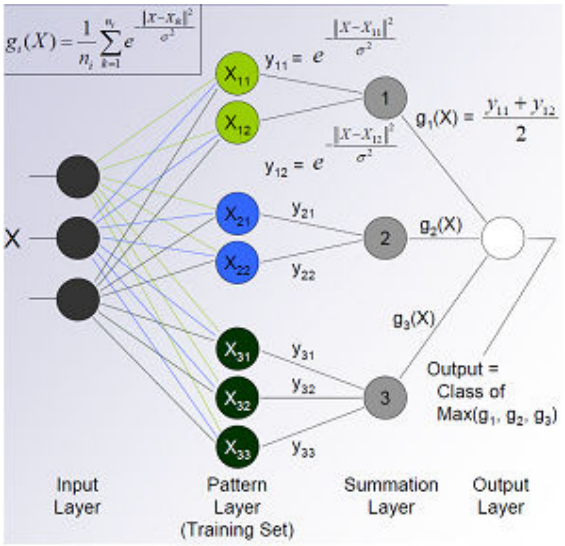

Figure 1. The structure of the neural network used in PNN[3]

PNN is now among the neural network structures that the researchers pay attention. In the recent years there are many research applications that uses PNN. Araghi et.al [1] used PNN as a classifier for ships. Grim et.al [7] used PNN to play a Tic-Tac-Toe game. Some other trails are to explore some modified PNN in various applications, Georgiou et.al [5] used PNN in bioinformatics task. The model incorporates the Particle Swarm Optimization algorithm to optimize the spread parameter of the probabilistic neural network, enhancing thus its performance. Stevens and Sundareshan used PNN[12] for sensor configuration for managing wireless ad hoc networking. El Emary and Ramakrishnan [4] devoted their study to application of PNN in various classification problems like classification brain tissues in multiple sclerosis, classification image texture, classification of soil texture and EEG pattern classification. They compared their results with other classification methods**.** Georgiou et. al [6] tried to improve the performance of PNN by incorporating a fuzzy class membership function for the weighting of its pattern layer neurons. For each neuron of the pattern layer, a fuzzy class membership weight is computed and it is multiplied to its output in order to magnify or decrease the neuron's signal when applicable[6].

It is to be noted that there are no trails concerning with incorporating rule based system with PNN.

B. Rule Based Systems

There are some attempts to represent rules in neural networks to make the system able to infer results that are not given in the set of rules. This can be done after representing the rules in a neural networks and training the neural networks by giving training examples. Some of these attempts are NEURULES [8], VPNL[2], KBANN [13][14], GRSNL[9]. It is worthy to mention that none of the known methods to represent rules using neural networks used PNN is existing. Moreover, none of these approaches considered the changes that might occur in rule based systems by adding or removing new rules without an extensive change in the neural network structure.

In our case we consider possible change in the knowledge base, the backpropagation algorithm will not be proper because adding new knowledge or rules means extensive training and that is why we prefer to use PNN. Another assumption we consider is that we don't have other training examples more than the knowledge base. In our approach we shall consider rule based systems with standard rule which is represented as If <C> Then <A>, where C is the condition and A is the action. If C is matched then A is achieved/obtained/executed. Some rules might initialize other rules whereas others might not initialize any other rule. For example

Given the inputs A,X,R,M,T and the following set of rules

Rule 1: IF A and X Then G
Rule 2: IF G and M Then Z
Rule 3: IF R and M Then G
Rule 4: IF M Then Y
Rule 5: IF Y and T Then Z

G can be obtained because A and X are given, and Z can only be produced after obtaining G. This means Rule 1 or Rule 3 initializes Rule 2. It is also to be noted that G can be obtained by either Rule 1 or Rule 3 and Z can be obtained by Rule 2 or Rule 5.

II. THE PROPOSED APPROACH

We propose here a neural network structure mechanism for rule based systems. The proposed neural network is based on PNN. Our proposed algorithm is called Probability Neural Network for Rule Based System (PNN-RBS). The structure considers all variations of the rule based systems mentioned in the introduction. The structure may have the following layers:

1. Input layer: It contains a neuron for every input attribute in the rules. The input values can be 1, 0 or -1, where 1 represents true input/attribute, -1 represents

false input/attribute, and 0 means input/attribute is unknown.

2. The Pattern layer: Pattern layer represents some of the rules if not all the rules where each neuron represents one rule. Other remaining rules can be in intermediate layers. Each neuron in the input layer is connected to the neurons in the pattern layer. The neurons in pattern layer are connected to their related neurons in intermediate layer. This means rules in the pattern layer are initializing rules in the intermediate layer.
3. Intermediate layer: It is a layer which contains a single neuron for every intermediate rule that is initiated by one or more rule existing in pattern layer, pervious intermediate layer or summation layer. We might have more than one intermediate layer.
4. The summation layer: It is the layer which has a neuron for every group of nodes in the intermediate layer or pattern layer with the same output/action. For example three nodes with X action in the intermediate layer are connected to a node in the summation layer to represent the whole three Xs. We might have more than one summation layer.
5. Output layer: It contains one node which represents the output of the given inputs.

The proposed neural network structure is given in Figure 2. The square nodes represent rules with the same output/action.

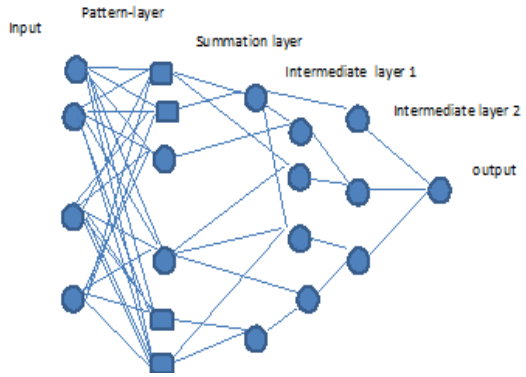

Figure 2. Shows the NN structure of the proposed rule based system with two pattern layers.

To make the idea more clear let us consider the following Figure 3 related to the example below:
Rule 1: IF A and B Then Z (to be considered as Z2 in Table 1)
Rule 2: IF C and D Then R (to be considered as R1 in Table 1)
Rule 3: IF G and F Then Z (to be considered as Z1 in Table 1)
Rule 4: IF M Then F
Rule 5: IF E and N then R (to be considered as R2 in Table 1)
Rule 6: IF Z and F then Y
Rule 7: IF R and M Then T
Rule 8: IF G and A Then H
Rule 9: IF X and B Then E
Based on the above rules, we shall have:
Input layer : Contains 8 nodes, A,B,C,D,G,M, N and X
Pattern layer : Contains 3 nodes, Rule 4, Rule 8, Rule 9
Intermediate layer: Contains 3 nodes Rule 1, Rule 2, Rule 3, Rule 5
Summation layer : It has two nodes, Z and R .
Output Layer: One node represents the output.

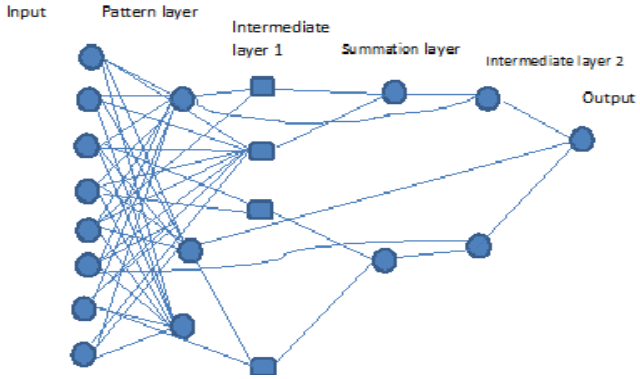

Figure 3. The structure of the rules given in the rule based example.

To illustrate Figure 3 and explain our proposed algorithm we shall explain our algorithm through Figure 3. Table 1 represents the data items of the structure of Figure 3.

Table 1. Representation of the data items for the rule based system example given in Figure 3.

Input layer	Pattern Layer 1	Intermediate Layer 1	Summation Layer	Intermediate layer 2	Output layer
A B C D G M N X	F E H	Z1 Z2 R1 R2	Z (two Zs in one) R (two Rs in one)	Y E T	Either Max function or Sort function

The proposed computation process will be as below

a. Insert the values for the given inputs/attributes as -1,0 or 1.
b. Calculate the values for the pattern layer by using Gaussian formula G as given before. For example in pattern-layer 1, we have three Gaussian values, G(F),G(E), and G(H).
c. Use the following formula to calculate each node in intermediate layer 1, for example
   Z1 = Min(G(F) , input(G))
   Z2 = G(Z2)
   R1 = Min(input(C), input (D))
   R2 = Min(input(N), G(F))
   Where input(D) value could be 0,1 or -1, and G(F) is the Gaussian function computed at node F. It is to be noted that the node Z2 is Gaussian value of Z2.
d. Compute the nodes in the Summation layer as below
   Z = (Z1+ Z2)/2
   R = (R1 + R2)/2
e. We have another intermediate layer (layer 2) before the output layer and its node values can be computed as below
   Y= Min (G(F), Z)
   E= G(E)
   T = (R+ input(M))
f. The output layer is
   Max (Y,E,T), The node with the higher value will be considered the output of the network. This is in case the structure developed for a system that produces one output. In case the developed system may produce more than one output, the used function is Sort(Y,E,T), which produces three outputs sorted according to the most relevant to the given input. If the value of a certain node is below a threshold t, it can be neglected and not considered in the Sort function. For example if E value is below t, then the output is Sort(Y,T). It is also to be noted that E is considered in both the layers pattern layer and intermediate layer 2 because it can be an output for the rule based system/neural network.

III. DISCUSSION

One of the main advantages of the proposed approach is that adding new rules will not force us to make training, we add the rules to the neural network structure and the same process will be done. For example if we add the following two rules

Rule 10: IF C and F Then Z (to be considered Z3 in Table 2)
Rule 11: IF J THEN P

The new neural network structure will have j as new input in the input layer. Also P will be considered to be one of the outputs because it does not initiate any other rule. Another value will be considered say Z3 to compute the value of Z. The new neural network structure is shown in Figure 4 and the representation of the data item of the neural network is given in Table 2. Based on the new structure, in the intermediate layer 1, Z3 = Min(G(F), input (C)) and in the summation layer Z=(Z1+Z2+Z3)/3. In the pattern layer, we additionally compute G(P) and in the output layer we perform either the Max(Y,E,T,P) or Sort (Y,E,T,P). Some times and in worst cases we may add new layers such as intermediate layer or summation layer.

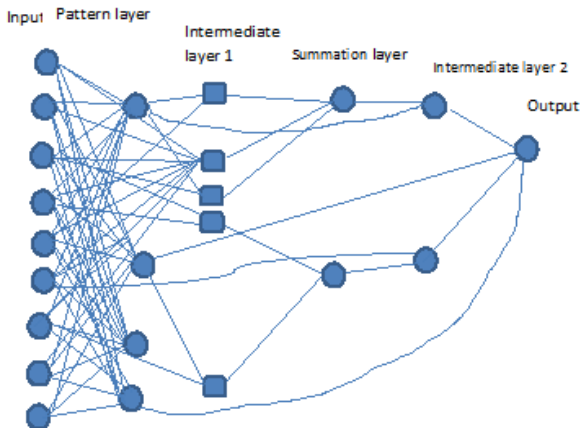

Figure 4. The structure of the rules after adding two new rules.

Table 2. Representation of the data items for the rule based system example after adding two rules and given in Figure 4.

Input layer	Pattern Layer 1	Intermediate Layer 1	Summation Layer	Intermediate layer 2	Output layer
A B C D G M N X J	F E H	Z1 Z2 Z3 R1 R2	Z (two Zs in one) R (two Rs in one)	Y E T O	Either Max function or Sort function

IV. CONCLUSION

In this paper we presented a representation of rule based systems using PNN by proposing an algorithm called PNN-RBS. This representation has been used to make adaptive rule based systems to be able to give conclusions about cases which are not provided in the knowledge base. Moreover, adding and removing new rules will not cost the system retraining. The proposed structure is similar to that of PNN with addition to a layer which is called intermediate layer. The proposed structure might have many intermediate layers and summation layers. The number of intermediate layers and summation layers is changeable based on the represented rules. The proposed algorithm is simple and easy to implement. One of the future directions is applying this proposed algorithm and structure on many rule based system domains.

V. REFERENCES

[1] L. F. Araghi, H. Khaloozade , M. R. Arvan , "Ship Identification Using Probabilistic Neural Networks (PNN)", Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, Vol III, MECS 2009, Hong Kong, March 18 - 20, 2009.

[2] K. Bharadwaj and J. Silva, Towards integrating hierarchical censored production rule (HCPR) based system and neural networks, in F.Oliveira (Ed.), Lecture Notes in Artificial Intelligence, 1515, pp.121-130, 1998.

[3] V. Cheung and K. Cannons," An Introduction to Probabilistic Neural Networks", Signal & Data Compression Laboratory, Electrical & Computer Engineering, University of Manitoba, Winnipeg, Manitoba, Canada, www.psi.toronto.edu/~vincent/research/presentations/PNN.pdf

[4] I. El Emary and S. Ramakrishnan, "On the Application of Various Probabilistic Neural Networks in Solving Different Pattern Classification Problems", World Applied Sciences Journal, 4 (6), pp. 772-780, 2008

[5] V. Georgiou, N. Pavlidis and K. Parsopoulos, P. Alevizos, M. Vrahatis, "Optimizing the Performance of Probabilistic Neural Networks in a Bionformatics Task", Proceedings of the EUNITE 2004, pp. 34-40, 2004.

[6] V. Georgiou, P. Alevizos and M. Vrahatis, "Fuzzy Evolutionary Probabilistic Neural Networks", Artificial Neural Networks in Pattern Recognition, Lecture Notes in Computer Science, 2008, Vol. 5064/2008, pp. 113-124, 2008.

[7] J. Grim, P. Somol and P. Pudil, "Probabilistic Neural Network Playing and Learning Tic-Tac-Toe", Pattern Recognition Letters-Special issue: Artificial Neural Networks in Pattern Recognition, Vol. 26, issue 12, pp. 1866-1873, Sept. 2005.

[8] I.Hatzilygeroudis, I. and J. Prentzas, " Neurules: improving the performance of symbolic rules", Inter. Journal on Artificial Intelligence Tools, 9, pp. 113-130, 2000.

[9] N. Hewahi, "Principles on integrating general rule structure (GRS) based systems and neural networks", Proc. Of the Inter. Conference on Artificial Intelligence, IC-AI '04, pp. 174-180, 2004.

[10] D. Specht, "Enhancements to Probabilistic Neural Networks", International Joint Conference on Neural Networks, vol. I, pp. 761-768, June 1992.

[11] D. Specht, "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory", IEEE International Conference on Neural Networks, vol. I, pp. 525-532, July 1998.

[12] J.Stevens and K.Sundareshan "Probabilistic Neural Network-Based Sensor Configuration in a Wireless Ad Hoc Network", Technical report, Arizona university tucson department of electrical and computer engineering, Dec. 2004.

[13] G. Towell and J. Shavlik, "Interpretation of artificial neural networks: mapping knowledge base neural networks into rules", Advances in Neural Information Processing Systems, Vol. 4, Denver, Co: Morgan Kaufmann, pp. 977-984, 1991.

[14] G. Towel. and J. Shavlik, "Knowledge base artificial neural networks", Artificial Intelligence, 70, pp.119-165, 1994.