International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-3, August 2013

Reinforcement-Clustering Technique based on POPTVR FNN for Pattern Classification

A.Mohan, V.Uday Kumar, B.Sateesh

Abstract-In ***general, a Fuzzy Neural Network (FNN) is characterized by its learning algorithm and its linguistic knowledge representation. However, it does not necessarily interact with its environment when the training data is assumed to be an accurate description of the environment under consideration. In interactive problems, it would be more appropriate for an agent to learn from its own experience through interactions with the environment, i.e. reinforcement learning. In this work, three clustering algorithms are developed based on the reinforcement learning paradigm. This allows a more accurate description of the clusters as the clustering process is influenced by the reinforcement signal, They are the Reinforce Clustering Technique I (RCT-I), the Reinforce Clustering Technique II (RCT-II), and the Episodic Reinforce Clustering Technique (ERCT).we have implemented, the integrations of the RCT-I, the RCT-II, and the ERCT within the pseudo-outer product truth value restriction (POPTVR), which is a Fuzzy neural network integrated with the truth restriction value (TVR) inference scheme in its five layered feed forward neural network. The three reinforcement-based clustering techniques applied to the POPTVR network are able to exhibit the trial-and-error search characteristic that yields higher qualitative performance.***

Index Terms—Clustering, Fuzzy Neural Networks

I. INTRODUCTION

Fuzzy NEURAL NETWORKS (FNNs) are an important paradigm in the development of hybrid intelligent systems for solving complex real-world problems such as pattern recognition, modeling, and forecasting. Some of the more commonly known FNNs include the adaptive network-based fuzzy inference system (ANFIS) and the Falcon adaptive resonance theory (Falcon-ART).Moreover, more advanced fuzzy architectures can be found in, for example, Yager rule network ,pseudo-outer product Yager (POP-Yager) FNN and Yager pattern classifying FNN. They combine the learning, fault tolerant, and parallel processing capabilities of neural networks, and the human inference and decision-making style of fuzzy inference systems to allow a better description of fuzzy attributes and rules. Conventionally, fuzzy membership functions can be derived manually by experts. Several attempts have been made to automatically generate fuzzy membership functions for the fuzzy sets using clustering techniques or learning approaches that organize the training data into the appropriate fuzzy sets.

Reinforcement learning is a learning process which maps situations to actions so as to maximize a numerical reward signal. The agent is not told of what actions to take but instead, it must discover which action yields the best reward by exploring them. In more difficult situations, the selected actions may not only affect immediate rewards but also the next situation and all subsequent rewards. Hence, reinforcement learning is characterized by trial-and-error search and delayed rewards. Much of the literatures in reinforcement learning have been inspired by the classical methods in dynamic programming.

The dominant approach for the last decade has been the value-function approach, in which all function approximation effort goes into estimating a value function. The action selection policy is implicitly represented as the greedy policy with respect to the estimated values. Although the value-function approach worked well in many applications, it has several limitations.

First, it is oriented towards finding deterministic policies whereas the optimal policy is often stochastic, selecting different actions with specific probabilities .Second, an arbitrarily small change in the estimated value of an action can cause it to be selected or left out.
These discontinuous changes are the key obstacle to establish convergence assurances for algorithm following the value-function approach. For example, Q-learning and dynamic programming have proven to be unable to converge to simple function approximators. An alternative approach is to explicitly represent the policy by its own function approximator, independent of the value function. Then, it is updated accordingly to the gradient of the expected reward with respect to the policy parameters. More recently, there are other works that have successfully employed recurrent neural networks to solve function approximation through dynamic programming.

In this work, we attempt to integrate reinforcement learning based on the optimal policy approach to generate fuzzymembership parameters in FNNs. When there exists a lot of input–output data from the observation of the system to be modeled, in the absence of other information about the system, then the determination of the structure of the FNN to describe the input–output relations becomes an important issue. In general, if only the input–output data are available for the system, then clustering algorithms are one of the most promising techniques employed to construct the fuzzy system. Conventional algorithms for FNN clustering are described in, for example. However,in general, such clustering algorithms do not interact with their environment by assuming that the training data have an accurate description of the environment.

Manuscript published on 30 August 2013.
***Correspondence Author(s)
A.Mohan, Computer Science And Engineering, Chaitanya Bharathi Institute of Technology,Hyderabad,Andhra Pradesh,India.
V.Uday Kumar, Computer Science And Engineering, Chaitanya Bharathi Institute of Technology,Hyderabad,Andhra Pradesh,India.
B.Sateesh, Computer Science And Engineering, Chaitanya Bharathi Institute of Technology,Hyderabad,Andhra Pradesh,India.

Hence, they are inadequate for learning via interaction to organize the training data into the appropriate fuzzy sets.

Therefore, the pseudo-outer product truth value restriction (POPTVR) FNN that employs reinforcement learning in the clustering of fuzzy memberships is proposed in this work.

II. PROBLEM SPECIFICATION

The POPTVR integrates the truth restriction value (TVR) inference scheme into a feed forward-type connectionist network. Each layer in the FNN performs the TVR fuzzy inference.

Three types of clustering algorithms based on reinforcement learning are developed together with the Reinforce algorithm in the POPTVR FNN architecture while retaining its TVR fuzzy inference scheme. These three clustering algorithms are, namely, Reinforce Clustering Technique I (RCT-I), Reinforce Clustering Technique II (RCT-II), and Episodic Reinforce clustering Technique (ERCT). The integrations of the RCT-I, the RCT-II, and the ERCT into the POPTVR.Three classification tasks were selected for benchmarking the reinforcement-based FNNs against other types of FNNs. The results are promising and show that the reinforcement-based clustering techniques are able to produce higher qualitative performance, especially when the data characteristics are highly nondistinguishable.

III. IMPLEMENTATION

The tasks of selecting an appropriate reinforcement learning rule and the derivation of the new REINFORCE learning rule have been accomplished in the first two subsections. In this section, the architectures and the learning sequences of the novel clustering networks will be examined. Altogether, three learning sequences will be described; they are, namely:

RCT-I
RCT-II
ERCT.

3.1 REINFORCE clustering technique I (RCT-I)

RCT-I is based on the competitive principle of the Kohonen network. In the Kohonen network, the strongest node that emerged as the winner is updated by the learning rule. Similarly, in RCT-I architecture, as shown in Fig. 3.1, it is based on the same principle. For each input vector X the s_i=||x-w_i|| for each output node is computed .Figure 3.2 Architecture of RCT-I.

The winning output node is identified as the smallest value of s_i then; its corresponding probability pi is computed and y_i will be generated based on the comparison against the threshold probability p_t.Finally, based on the value of y_i, the winning weight vector will be updated using the appropriate learning equations. In RCT-I, the reinforcement signal r in the two learning equations is assumed to be discrete, that is r ϵ {0, 1}. The number of clusters is also a known *a* priori.

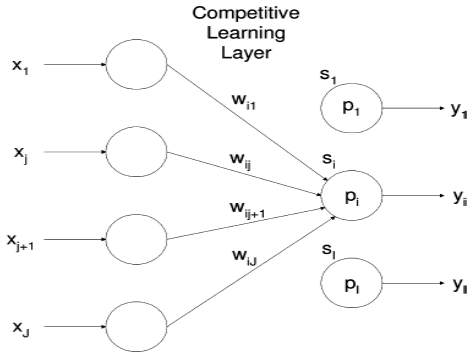
In Fig 3.1the input vector is represented by X=[$x_1, \dots . x_j, x_{j+1}, \dots . x_j$] The link weights from J number of input nodes to an output node i is represented by the weight vector .$w_i = [w_{i1}, \dots \dots w_{i,j,} w_{i,j+1}, . . w_{i,j}]$The output vector is represented by Y= $[y_1, . . y_i, \dots .. y_I]$ In addition, for an output node i, there are two scalar values:

1) s_i represents the Euclidean distance between input and weight vectors at node ;

2) p_i represents the probability of the output node I With reference to Fig 3.1 the learning

3.2 REINFORCE clustering technique (RCT-II)

RCT-II shares the same learning equations as RCT-I. However, it has a different architecture and learning sequence from RCT-I. The architecture of RCT-II is illustrated in Fig.3.2.1



Fig(3.2.1)

For each output node i , there is a set of k intermediate nodes that is linked to it. Basically, each set of intermediate nodes can be viewed as a set of linguistic labels (Li1…Lik,…Lik) for each output node Lik where is the output value of the k th linguistic label node of output node i . Among all output nodes, only one output node can be active (i.e. yi=1,) for selecting which set of k linguistic nodes to compute. Suppose a set of linguistic labels is selected, which is linked to this active output node , their respective Euclidean distances and corresponding probabilities p_i will be computed in the same manner as RCT-I.

The winning node is accordingly derived and its corresponding weight vector will be updated using the learning equations (6) and (7). In RCT-II, the reinforcement signal (r) in the two learning equations can be either discrete, i.e., r ϵ {0, 1}.,or continuous, i.e. r ϵ {0, 1}. The number of clusters is known a priori. To further clarify, when discrete reinforcement signal is used, that is, r ϵ {0, 1}.only the winning linguistic node is updated. When continuous reinforcement signal is used, i.e. r ϵ {0, 1}.both the winning linguistic node and its neighboring nodes will be updated. The strength of the reinforcement signal for a particular node will depend on the lateral distance between the winner and itself over the Gaussian distribution of rik=$e^{(-|wik-wiq|)/\sigma}$ The winner will have rik=1 while those farthest away from the winner will have rik~0. With reference to Fig.3. 2.2

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

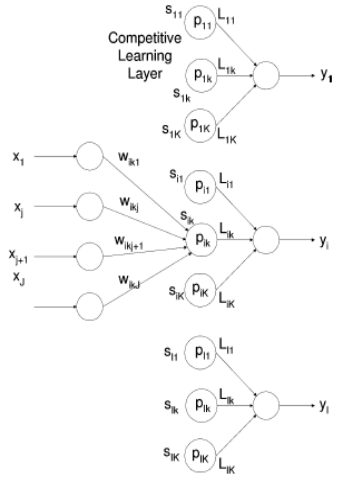


Figure 3.2.2:Architecture of RCT-II

3.3 ERCT (Episodic clustering technique)

The architecture of the ERCT is the same as that of RCT-II. However, the update of the weight vectors wik is different. The ERCT is characterized by a parameter called the episode$\bar{\varepsilon}$. For a learning cycle, the ERCT is updated by exactly the number of episodes $\bar{\varepsilon}$ defined by the user. Then, for each episode, the weight updates for 'T 'number of data vectors will be accumulated. 'T' is defined as [N/M] where N is the total number of data vectors and M is the number of episodes. At the end of each episode, the weight vectors will be updated by multiplying the accumulated weight updates $\sum_{p=0}^{T} wiq$ by a factor r, where 'r' is a reinforcement signal. The reinforcement signal 'r' can be computed dependent upon the classification result by running the ERCT algorithm. A normalized classification result a(i.e.a=0[1]) can be derived in each classification test by ERCT with respect to each data vector. Then, the reinforcement signal 'r' is computed as 1-a. Initially, the normalized classification result will be low,'r' hence will be large. This means that a corrective action is needed and hence 'r' a larger value is able to make significant correction. However, as the number of training cycles is increased, the normalized classification result should be higher, hence value will become lower. This also means that the ERCT network needs fine tuning and hence a smaller is desirable. The rationale behind this is that a stronger reinforcement signal 'r' is needed initially to correct the ERCT network but it should decrease as the network stabilizes towards the end of its training cycles.

IV. POPTVR ARCHITECTURE AND LEARNING ALGORITHM

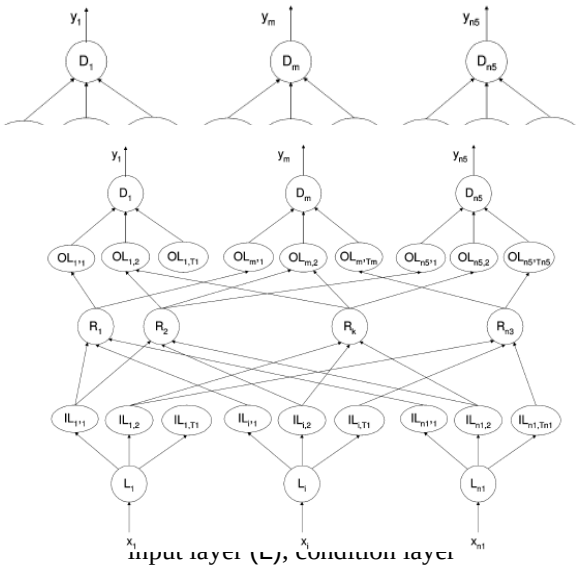
The POPTVR [38] is a type of FNNs that integrates the TVR in fuzzy logic with a feed forward connectionist network. In a nutshell, each layer in the FNN performs a respective step in the TVR fuzzy inference. First, this section presents the basic architecture and learning algorithm of POPTVR. The limitations of the POPTVR will then be addressed, and hence the proposed REINFORCE-based POPTVR will be described to show how to overcome the limitations.

4.1 The Architecture of POPTVR

POPTVR has a total of five layers as shown in Fig. 4.1 The five layers are, namely, the input layer, the condition layer, the rule-base layer, the consequence layer, and the output layer. Each layer performs its appropriate fuzzy operations and the five layers together perform the TVR fuzzy inference.

The input and the output of the POPTVR are represented as vectors. X=[x1,..x2,..Xi,..xn1] ^{T} and Y=[y1,y2,…yn5] ^{T} Where n1 and n5 denote the number of input and output variables, respectively. Both the input and output vectors are nonfuzzy.

The symbol n3 in the rule-base layer denotes the number of fuzzy rules in the POPTVR. Ti denotes the number of fuzzy labels of the th input node of the input layer. Hence, there are a total of $\sum_{i=1}^{n1} T_i$ nodes in the condition layer. Similarly Tm denotes the number of fuzzy labels of the m th node in the output layer. In addition, each input linguistic label is denoted by $IL_{i,j}$ where i is the i th element of the input vector x, and j is the j th input linguistic label of input node i.



I s

input layer (L), condition layer (IL), rule-base layer (R), consequence layer (OL), and output layer (D).

i). The Input Layer

The neurons in the input layer are known as the linguistic nodes. They represent linguistic variables such as weight or height or others. The linguistic nodes propagate the nonfuzzy input vector to the second layer. For each linguistic node, its net input f_i^I and net output o_i^I of the ith linguistic node Li are described, respectively, by

$$f_i^I = x_i \qquad (1)$$

$$o_i^I = f_i^I \qquad (2)$$

Where x_i is the th element of the input vector x.

ii).The Condition Layer

The neurons in the condition layer are known as the input label nodes. They represent fuzzy labels such as small, medium, and large of the corresponding input linguistic variable. The input label nodes in the condition layer form the antecedents of the fuzzy rules in POPTVR.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Each input label node $IL_{i,j}$ denotes the jth input label of the linguistic node lin the input layer. Each input label node is represented by a bell-shape membership function as shown in Fig.4.1The node function is chosen such that it ensures the differentiability to maintain the compatibility with the back propagation algorithm [21]. For each input label node $IL_{i,j}$ its net Input and net output $o_{i,j}^{II}$ are described, respectively, as

$$f_{i,j}^{II}=\frac{(o_i^I-c_{i,j}^{II})^2}{\sigma_{i,j}^{II}} \quad (3)$$

$$o_{i,j}^{II}=c_{i,j}^{II} \quad (4)$$

Where $c_{i,j}^{II}$ the centroid of the membership is function for the input label node $IL_{i,j}$ and $\sigma_{i,j}^{II}$ is the width of the membership function for the input label node .$IL_{i,j}$

iii). The Rule-Base Layer:

The neurons in the rule-base layer are known as the rule nodes. They represent rules like "if height is short then weight is light." For each rule node, the net input f_k^{III} and net output o_k^{III} of the kth rule node R_k are described, respectively, by

$$f_k^{III} = \min(o_{i,j}^{II}) \quad (5)$$

$$o_k^{III} = f_k^{III} \quad (6)$$

Where o_k^{III} is the net output of the input label node .$IL_{i,j}$ This node forms one of the antecedent conditions to the rule node. R_k.

iv) . The Consequence Layer

The neurons in the consequence layer are known as output label nodes. They represent fuzzy labels such as light, medium, and heavy of the corresponding output variables. The output label $OL_{l,m}$ represents the lth label of the defuzzification node D_m in the output layer. For each output label node,$OL_{l,m}$ The net input $f_{l.m}^{Iv}$ and net output $o_{l.m}^{Iv}$ are described, respectively, by

$$f_{l.m}^{Iv}=\sum_k o_k^{III} \quad (7)$$

$$o_{l.m}^{Iv}=\min(1, f_{l.m}^{Iv}) \quad (8)$$

Where o_k^{III} is the net output of the rule node. R_k The summation is only for the rule nodes which take the output label node$OL_{l,m}$ as one of their consequences.

v). The Output Layer

The neurons in the output layer are known as the defuzzification nodes. They represent the output variables such as height, weight, or others. For each defuzzification node, D_m the net input f_m^V and net output o_m^V are given as

$$f_m^V=\sum_{l=1}^{T_m}\frac{c_{l,m}^{IV}o_{l,m}^{IV}}{\sigma_{l,m}^{IV}} \quad (9)$$

()

$$o_m^V=f_m^V\frac{f_m^V}{\sum_{l=1}^{T_m}\frac{o_{l,m}^{IV}}{\sigma_{l,m}^{IV}}} \quad (10)$$

Where $c_{l,m}^{IV}$ the centroid of the membership function for the output is label node $OL_{l,m}$ and $\sigma_{l,m}^{IV}$ is the width of the membership function for the output label node. $OL_{l,m}$.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The pattern classification capability of the original POPTVR and the newly developed modified POPTVR .In this study, iris data sets are used for benchmarking. Hence, these data sets should serve as a reasonable platform to judge how well the network of interest can tackle the pattern classification problem. The back propagation algorithm in the original POPTVR is abandoned in modified POPTVR.Hence, for the purpose of fair comparison; a slightly modified version of the POPTVR is included in thebenchmarking in addition to the original POPTVR. The modified version of the POPTVR abandons the back propagation algorithm and empirically tunes its generalization accuracy using some fixed width constants, such as $\sigma_{i,j}^{II}$ and. $\sigma_{l,m}^{IV}$ Hence, a total of five differen architectures—POPTVR, the modified POPTVR.In addition, three sets of training and test data will be generated from each of the data set.

Table 5.1: Test Results Of POPTVR, The Modified POPTVR, For All The Three Test Case.

Percentage of Classification(%)	POPTVR	Modified POPTVR
Test1 Classification(%)	93.34	96.00
Test 2 Classification(%)	93.86	96.69
Test 3 Classification(%)	92.58	94.50
Average classification(%)	93.10	95.71
Standard deviation (%)	0.53	0.8

Table 5.1 tabulates the average classification test results and the standard deviation of the three different test cases According to the results shown in Table 5.1.it can be concluded that POPTVR and the modified POPTVR yield an average classification accuracy of 93.10 and 95.71 respectively. Their standard deviations in classification accuracy among the three classes are 0.55% and 0.8, respectively.

It is noted that POPTVR has the lowest classification results, however, it is no conclusive that the back propagation algorithm is unsuitable for classification since the POPTVR has a lower standard deviation than its modified counterpart and ties with modified POPTVR.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VI. CONCLUSION AND FUTURE ENHANCEMENT

In This work, the Reinforce algorithm has been adapted into the Kohonen network to form the REINFORCE clustering techniques, namely, RCT-I, RCT-II, and ERCT. This class of reinforcement-based clustering techniques replaces the self-organizing clustering algorithm in the POPTVR FNN. Basically, the REINFORCE algorithm often avoids the use of the value function approach which has inherent limitations. Instead, it explicitly represents the policy by its own function approximator, which is independent of the value function.

The parameters are updated according to the gradient of the expected reward with respect to the policy parameters. From the experimental results, with the trial-and-error search, characteristic of reinforcement learning integrated as part of the clustering techniques has led to superior performance. Moreover, RCT-I, RCT-II, and ERCT are able to yield good convergence results in contrast to many supervised networks such as back propagation in the POPTVR network.

Further studies may address regression in addition to the current studies in pattern classification. A further investigation will also be carried out since there is neither mathematical proof nor experimental results to support that the use of the reinforcement baseline necessarily leads to faster convergence and more qualitative reinforcement actions.

REFERENCES

1. Wong WC, ChoSY, QuekC. "POPTVR fuzzy neural network for pattern classification". IEEE Trans Neural Netw. 2009 Nov;20(11):1740-55. doi: 10.1109/TNN.2009.2029857. Epub 2009 Sep 18.
2. C. Quek and R. W. Zhou, "The POP learning algorithms: Reducing work in identifying fuzzy rules," Neural Network., vol. 14, no. 10, pp. 1431–1445, 2001.
3. R. W. Zhou and C. Quek, "POPFNN: A pseudo outer-product based fuzzy neural network," *Neural Netw.*, vol. 9, no. 9, pp. 1569–1581, 1996.
4. G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbo generator," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 764–773, May 2002.

AUTHOR PROFILE

A.Mohan, Assistant Professor,C.S.E Deptchaitanya Bharathi Institute Of Techmology,He Has Done His Masters In Computer Science In Chaitanya Bharathi Institite Of Technology.He Is Having 3 Years Of Teaching Experience.He Publishes one paper on Datamining in National Conference held at C.B.I.T. 2013.His area of interest is FUZZY LOGIC,DATAMINING,NEURAL NETWORKS

V.Udaykumar, Assistant Professor, C.S.E, Dept, Chaitanya Bharathi Institute Of Technology.He Has Done His Masters In Chaitanya Bharathi Institute Of Technology, Hyderabad In 2008. He also has publishesd two papers in INTERNATIONAL JOURNALS in DATAMINING.His area of interest is DATAMINING,ADVANCED DATABASES

B.Sateesh, Assistantprofessor, C.S.E, Dept, Chaitanya Bharathi Institute Of Technology.He Has Done His Masters In Chaitanya Bharathi Institute Of Technology, Hyderabad In 2008.He Is Having 5 Years Of Experience In Teaching Field. He Publishesdd One Paper On Datamining In National Conference Held At C.B.I.T. 2013.His Area Of Interest Is Datamining And Advanced Databases

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)