

Neural-network designed on approximate reasoning architecture and its application to the pattern recognition

高木, 英行

Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.

KOUDA, Toshiyuki

Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.

KOJIMA, Yoshihiro

Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.

<https://hdl.handle.net/2324/4479713>

出版情報 : Proceedings of International Conference on Fuzzy Logic & Neural Networks, Iizuka '90. 1, pp.671-674, 1990. Fuzzy Logic Systems Institute

バージョン :

権利関係 :



NEURAL NETWORK DESIGNED ON APPROXIMATE REASONING ARCHITECTURE AND ITS APPLICATION TO THE PATTERN RECOGNITION

Hideyuki TAKAGI, Toshiyuki KOUDA and Yoshihiro KOJIMA

Central Research Laboratories,
Matsushita Electric Industrial Co., Ltd.

3-15, Yagumo-Nakamachi, Moriguchi, Osaka 570 JAPAN

[abstract] This paper describes how to construct the structured NN using the structure of fuzzy inference rules. The advantage of this structured NN is that it is easy to find out the internal part that causes poor performance of whole NN system by analysis of error according to the rule structure. This paper mainly shows the procedure to construct this structured NN and to improve it using pattern identification problem.

1. INTRODUCTION

1.1. Background

In recent years, researches on neural networks (NN) and fuzzy system have been accelerated. According to this advancement, examinations of new fusion technology that combines the advantage of NN and fuzzy system have also become enthusiastic. Most of these researches incorporate the learning function and non-linearity of NN into fuzzy system. However, the reverse researches incorporating fuzzy theory into NN are very few ⁽¹⁾.

1.2. Purpose of this paper

The simplest usage of NN is to let NN learn input/output relation. This method has the merit of easy automated identification knowledge acquisition by the learning algorithm. However, it is hard to analyze NN and to improve the capacity because the knowledge is implicitly

expressed in NN.

The solution is to incorporate the known knowledge structure to NN in advance and use the knowledge structure for analysis. This paper describes how to design NN with the fuzzy inference rules of IF-THEN format that is direct and high-level knowledge expression. The internal analysis according to IF-THEN format makes easier capture and modification of problems. This paper explains the principle that the above solution can improve the capability.

2. STRUCTURE OF NN DESIGNED ON APPROXIMATE REASONING ARCHITECTURE

The ground of this structural method is following. The first point is similarity of fuzzy partition of input space and identification of NN. NN makes partition of input space and identifies the classes. Because neuron output has sigmoid feature, this input space will not be divided crisply even if the two-valued supervised data is used for learning. This is similar to the membership function of fuzzy set. That is, pattern identification of NN is common to fuzzy partition of the input space. The second point is that approximate reasoning infers result according to own principle of each divided space (or rule).

From these two points, the following NN system can be constructed. Fig.1 shows the structure of NN_{mem} and NN_s . NN_{mem} corresponds to IF part of fuzzy inference rules that make fuzzy partition of the input data. NN_s corresponds to THEN part of s-th fuzzy inference rule, and each NN_s is independent for other rules. The point of NN-driven Fuzzy Reasoning ^{(2),(3)} is incorporating NN to generate membership function of fuzzy inference rules automatically. Although the purpose is different, Fig.1 shows the similar structure to the one shown as a practical form of NN-driven Fuzzy Reasoning.

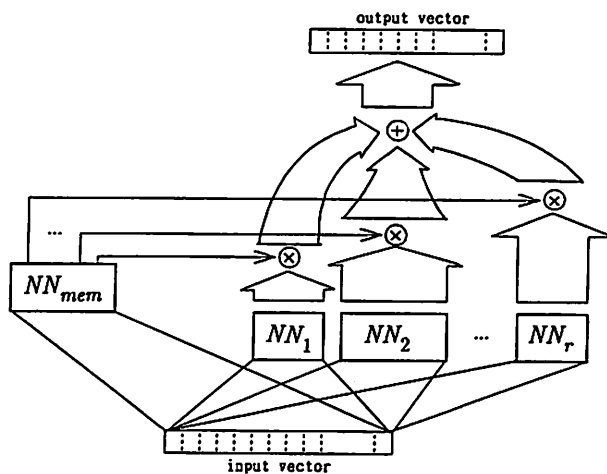


Fig.1 Neural-network designed on Approximate Reasoning Architecture

As an expansion, it is possible to replace NN corresponding to IF parts with another method for membership function generation. For example, this NN can be the method forming membership function based on fuzzy clustering ⁽⁴⁾. To the contrary, THEN part can be replaced with another reasoning method, or combination of IF and THEN parts replacements is also possible. Using these expansions, however, the form of structured NN is getting indefinite.

3. HOW TO DESIGN NN AND HOW TO IMPROVE IT

3.1. Design Method

This section explains how to design proposed NN structure in detail. In the following designing procedure, Fig.2 showing pattern identification is used as a helpful example for understanding.

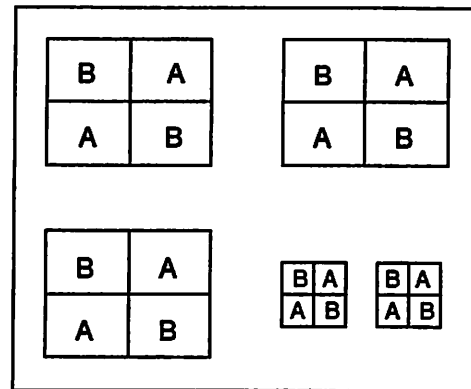


Fig.2 Sample Task of Pattern Identification

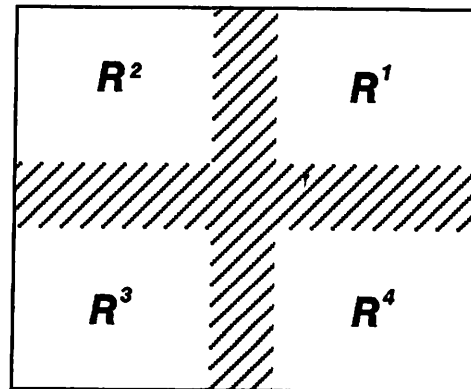


Fig.3 Clustered area of Fig.2

[Step 1] Clustering of training data

Cluster the D-dimensional feature vector (in Fig.2, 2-D coordinates) of the input data. An identification label is not included in the clustering variables. As a result of this, for example, in case of character recognition, the same identification label 'a' can be assigned to several classes because of difference of fonts. Let C^s be the s-th class ($1 \leq s \leq n$, wherein s is the class number

and n is the number of classes). The number of classes correspond to the number of inference rules. The input space shown in Fig.2 is clustered to $n = 4$, and is fuzzy partitioned as shown in Fig.3.

[Step 2] Training of NN corresponding to IF parts of fuzzy inference rules

The class number s of each C^s clustered in [Step 1] is used as the supervised data. Supervised data mentioned in [Step 1] and the D -dimensional input feature vector is used as the training data. Make NN learn to identify n classes. When hard clustering is applied, supervised data is crisp data that belongs to only one class. However, the internal acquired classes that trained NN express is not crisp set, but fuzzy set. That is, NN forms the continuous membership function internally. This is the reason that the proposed NN structure is based on the approximate reasoning architecture. Let NN_{mem} be this NN and use as the unification of all IF parts. Let R^s be the input subspace partitioned by NN_{mem} , where s is the class number. As to the recognition capability of NN, the fact that R^s is not crisp set, but fuzzy set enhances generalization. Namely, the sets of the training data and the checking data do not match completely because of variation (generally, it is impossible to avoid sampling noise because of practical finite number of training data). In this case, a design is required to recover the checking data that is close but doesn't belong to the set of the training data. Because R^s is fuzzy set, its class boundary provides width of $[0,1]$. Therefore, it is possible to recover the varied checking data.

[Step 3] Training NN_s corresponding to THEN part of s -th fuzzy inference rule

Train NN that identifies labels in classes for each C^s . In the case of Fig.2, four THEN parts identify a problem looking like Exclusive OR. Let NN_s be NN corresponding to THEN part of s -th inference rule NN_s .

[Step 4] Desition of final output

Combine NN_{mem} and $NN_1, \dots, NN_n$ trained in [Step 2] and [Step 3] as shown in Fig.1. The multiplier weights the recognition vector r_s that is output from NN as THEN part of each rule by the element of the membership vector $m = (m_1, m_2, \dots, m_n)$. The total of the recognition vectors r_s and m can be the final recognition vector r .

$$r = \frac{1}{n} \sum_{s=1}^n m_s \cdot r_s$$

3.2. How to improve the capability

[Step 5] Error analysis

Higher identification capability than the one of the structured NN designed in [Step 4] can be obtained by error analysis. In case of Fig.2, there are many output errors in 4th inference rule. Therefore, the point is the improvement on the identification capability of NN_4 as THEN part, not IF part.

[Step 6] The improvement for higher capability

The analysis by clustering the data in area C^4 as 4th inference rule shows that it is divided into two subclasses (Fig.4). In this case, when 4th inference rule is satisfied, design NN to make inference on applying two IF-THEN rules shown in Fig.4. Thus five THEN parts, NN_1 , NN_2 , NN_3 , NN_{41} and NN_{42} are generated for 4+2 rules. Each THEN part only has to identify the simple problem looking like as EXOR, and so identification capability of the structured

NN can be improved.

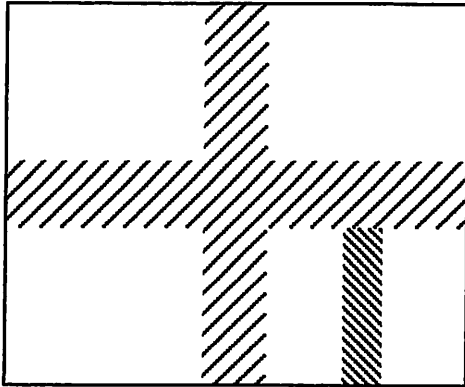


Fig.4 Subclustered area for improvement of Fig.3

4. EXPERIMENT

Ten random coordinate data are prepared for each class in Fig.2. This is, the number of total training data is two hundreds. The task is to identify A and B.

First, ordinary one NN is supervised to identified it. Let $NN^{ordinary}$ be this NN. The performance of $NN^{ordinary}$ was 50%. This performance means that $NN^{ordinary}$ can't identify A and B completely.

Second, structured NN that is designed according to [Step 1] - [Step 4] is supervised. The result of clustering was four classes as meeting our expectation. The number of synaptic weights is adjusted to one of $NN^{ordinary}$. Let NN^{new1} be this NN. The performance of NN^{new1} was 85%.

Third, NN^{new1} is improved according to [Step 5] and [Step 6]. The analysis shows that the error rate 15% is caused by NN_4 . The training data belonging to C^4 is clustered. The result of clustering was two classes as meeting our expectation, too. New NN_{41} and NN_{42} are added to NN^{new1} as mentioned in [Step 6]. Let NN^{new2} be this NN. The performance of NN^{new2} became to 94%. The following table shows the comparison of identification capability for train-

ing data of them.

$NN^{ordinary}$	NN^{new1}	NN^{new2}
50%	85%	94%

5. CONCLUSION

This paper has proposed the NN structural method based on the approximate reasoning rules, and has explained the advantages of this method using an example of virtual pattern. The point of the proposed method is that it can analyze internal of NN according to the rule structure, and can find out the point for improvement the identification capability. As a result of error analysis, the potential method for improvement depends on the distribution pattern. That is, it does not depend on only the number of partitions explained in [Step 6] of 3.2. From now on, we are going to evaluate the capability through concrete application.

References

- (1) H.Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future directions -", Int. Conf. on Fuzzy Theory & Neural Networks (IIZUKA-90), plenary session (1990)
- (2) H.Takagi & I.Hayashi, "Artificial Neural Network-Driven Fuzzy Reasoning", Int. J. Approx. Reas. (accepted in 1989, will appear in 1990)
- (3) H.Takagi & I.Hayashi, "Artificial Neural Network-Driven Fuzzy Reasoning", Int. Worksh. on Fuzzy Syst. Applic. (IIZUKA-88), pp.183-184 (1988)
- (4) K.Hirota & Y.Yoshida, "Identification of Fuzzy Control Rule Based on Fuzzy Clustering Method", 5th Fuzzy Syst. Symp., pp.253-258 (1989) (Japanese)