KNOWLEDGE LEARNING ON FUZZY EXPERT NEURAL NETWORKS*

H.C. Fu, J.J. Shann
Department of Computer Science and Information Engineering
Hsiao-Tien Pao
Department of Management Science
National Chiao-Tung University,
Hsinchu, Taiwan 300,
R.O.C.

Abstract

The proposed fuzzy expert network is an event-driven, acyclic neural network designed for knowledge learning on a fuzzy expert system. Initially, the network is constructed according to a primitive (rough) expert rules including the input and output linguistic variables and values of the system. For each inference rule, it corresponds to an inference network, which contains five types of nodes: Input, Membership-Function, AND, OR, and Defuzzification Nodes. We propose a two-phase learning procedure for the inference network. The first phase is the *competitive backpropagation (CBP)* training phase, and the second phase is the *rule-pruning phase*. The *CBP* learning algorithm in the training phase enables the network to learn the fuzzy rules as precisely as backpropagation-type learning algorithms and yet as quickly as competitive-type learning algorithms. After the *CBP* training, the *rule-pruning* process is performed to delete redundant weight connections for simple network structures and yet compatible retrieving performance.

1 Introduction

Recently, there are more and more research on implementing fuzzy expert systems in neural networks for the advantages of learning fuzzy expert rules from examples [7, 3, 2, 5, 8]. Most of the neural networks methodologies for learning knowledge can be divided into two categories: $backpropagation - type$ and $competitive - type$. Roughly speaking, backpropagation-type learning algorithms learn more precisely than competitive-type algorithms because they are based on gradient descent search, thus they take a long time and numerous training epochs to converge.

*This research is supported in part by the National Science Council under Grant NSC82-0408-E009-427.

 0-8194-1547-2/94/$6.00

In contrast, competitive-type learning algorithms learn much rapidly than backpropagation-type algorithms because they are based on unsupervised clustering, hence its the knowledge learning may not be precise enough. Therefore, an important issue that remains to be resolved in this field is how to learn knowledge both precisely and rapidly.

In this paper, we propose an event-driven, acyclic neural network for knowledge learning on a fuzzy expert system. The fuzzy rules considered are in the linguistic IF-THEN form. The *IF*-part, i.e., the antecedent, of a rule is the conjunction of several input linguistic variables [12], each associated with a linguistic value. The *THEN*-part, i.e., the consequence, of a rule contains only one output or intermediate linguistic variable associated with a linguistic value. The proposed learning procedure of this network consists of two phases. The first phase is a *competitive backpropagation (CBP)* training phase, and the second phase is a rule-pruning phase. The *CBP* learning algorithm in the training phase is designed to be a compromise between the advantages of backpropagation-type and competitive-type learning methodologies. After the *CBP* training phase, a rule-pruning process is executed to delete redundant weight connections and to better represent the inference rules.

This paper is organized as follows. The structure of the network and the functions of the nodes in the network are described in Section 2. The *CBP* learning algorithm is stated in Section 3. A pruning method for deleting redundant weight connections is described in Section 4. In Section 5, a simple exemplar problem is simulated and analyzed. Finally, conclusions and future research plans are presented in Section 6.

2 Network Structure and Node Functions

In general, the inference structure of a fuzzy expert system can be classified into two categories: (1) single level of inferences, and (2) multi-level of inferences. A single level inference expert system can be constructed by a five layered neural networks, and the inference knowledge can be learned from examples. The network structure for single-level inference as shown in Figure 1. Knowledge learning for single-level inference system have been presented in [9, 11]. For multi-level inference expert system, outputs of a defuzzification function can be inputs to a membership function of another inference rules. Implementing a multi-level inference system on neural networks, requires training examples and primitive knowledge of inference relations (rules). For instance, suppose an expert expert system is composed by the following rules:

$$
\begin{array}{lll}
\textit{Rule}1: & if\ \text{A}\ \textit{and}\ \text{B} & \textit{then}\ \text{F}, \\
\textit{Rule}2: & if\ \text{D}\ \textit{and}\ \text{E} & \textit{then}\ \text{G}, \\
\textit{Rule}3: & if\ \text{C}\ \textit{and}\ \text{F} & \textit{then}\ \text{H}, \\
\textit{Rule}4: & if\ \text{C}\ \textit{and}\ \text{G} & \textit{then}\ \text{I}.
\end{array}
$$

Without knowing primitive inference relations, a neural expert system can also be constructed by a single level inference structure, and training by many examples. Obviously, the resulting network will containing more connections and requires more training iterations to converge the

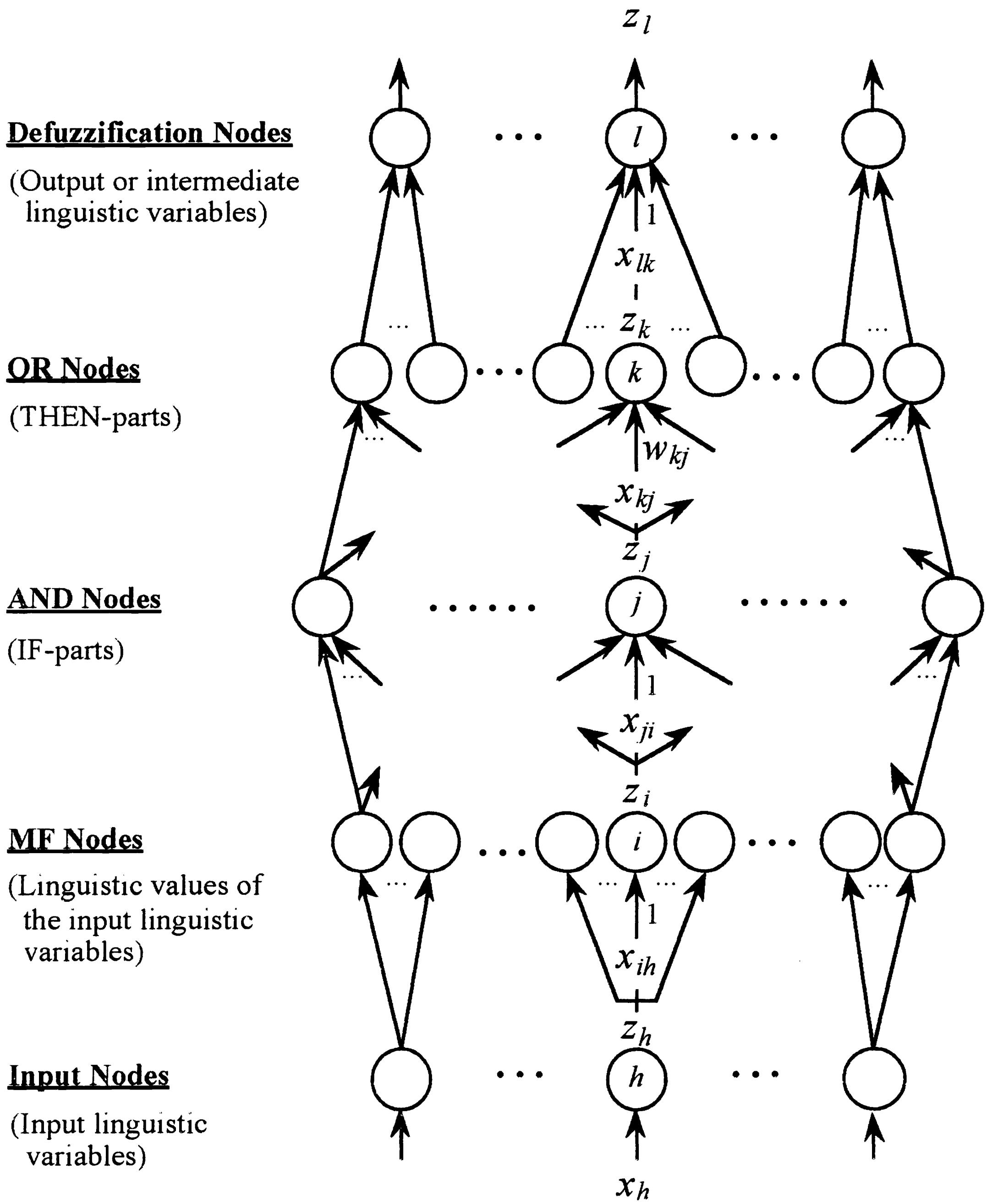
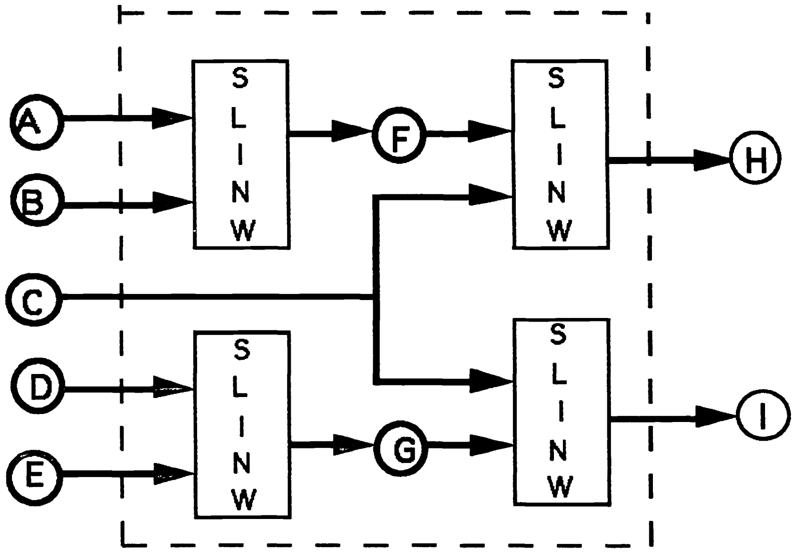


Figure 1: The structure of a fuzzy expert network for single-level inference.

learning process. This paper will focus on the construction and training of multi-level inference expert networks. First, according to the primitive inference relation, an inference flow diagram can be constructed (see Figure 2). Based on the inference diagram, we can apply the single-



SLINW: S ingle L evel I nference N etwork

Figure 2: The structure of a fuzzy expert network for multi-level inference.

level inference structure to implement each individual inference rules. As proposed in [9, 11], the network contains five types of nodes: Input, Membership-Function, Fuzzy-AND, Fuzzy-OR, and Defuzzification Nodes. The semantic meaning, connectivity, and function of each type of nodes are described as follows:

Let P_i be the set of the indices of the nodes each of which has its output link connected to node i. The semantic meaning, connectivity, and function of each type of nodes in the proposed network are described as follows:

- **Input Nodes:**
  Each input node of the fuzzy expert network represents an input linguistic variable of the fuzzy expert system, and is used as a buffer to broadcast the input to the membership-function nodes of its linguistic values.

- **MF (Membership-Function) Nodes:**
  Each MF node represents the membership function of a linguistic value associated with an input or an intermediate linguistic variable. An MF node has one input link emitted from an input node or from a defuzzification node. The output link of the MF node are connected to the AND nodes which represent the IF-parts containing this linguistic value of the linguistic variable. For each input or intermediate linguistic variable, there is n MF nodes, where

n is the number of linguistic values associated with the linguistic variable. The output of an MF node is in the range of [0, 1] and represents the membership grade of the input or intermediate linguistic variable with respect to the membership function of a linguistic value of the variable.

In general, the most commonly used fuzzy-set membership functions are in the shape of trapezoid, triangle, or bell. For bell-shaped membership functions, the operation of an MF node i with an input link emitted from an input or defuzzification node h is defined as follows:

$$z_i = \exp^{-\frac{(x_{ih}-c_i)^2}{2\sigma_i{}^2}} , \tag{1}$$

where c_i is the centroid, σ_i is the variance, $x_{ih} = z_h$, and z_h is the output value of node h. The weight of the input link of an MF node is unity.

- **AND Nodes:**

  Each AND node represents an IF-part for the possible rules of the fuzzy expert system. An AND node has several input links each of which is emitted from an MF node for the linguistic value of an linguistic variable containing in the IF-part. The output link of the AND node is connected to all the OR nodes which represent the THEN-parts of the fuzzy rules with this IF-part.

  In fuzzy set theory, the most commonly used operator for fuzzy intersection is the *min*-operator suggested by Zadeh [13]. Therefore, the operation performed by an AND node j is defined as follows:

$$z_j \;=\; \min_{i \in P_j}(x_{ji}) \;=\; MIN_j \,, \tag{2}$$

  where $x_{ji} = z_i$. The weights of the input links of the AND node, w_{ji}'s, are unity.

- **OR Nodes:**

  Each OR node represents a THEN-part for the possible rules of the fuzzy expert system. The operation performed by an OR node is to integrate fuzzy rules with the same consequence. An OR node may have several input links each of which is emitted from an AND node, and its output link is connected exactly to one defuzzification node which represents an output or an intermediate linguistic variable. The weight w_{kj} of the link connected from an AND node j to an OR node k represents the weight of a fuzzy rule which regards node j as the IF-part and node k as the THEN-part, respectively.

  In fuzzy set theory, the most commonly used operator for fuzzy union is the *max*-operator suggested by Zadeh [13]. Therefore, the operation performed by an OR node k is defined as follows:

$$z_k \;=\; \max_{j \in P_k}(x_{kj} w_{kj}) \;=\; MAX_k \,, \tag{3}$$

  where $x_{kj} = z_j$. The weights of the input links of an OR node, w_{kj}'s, are learnable positive real numbers.

- **Defuzzification Nodes:**

  Each defuzzification node represents either an intermediate linguistic variable or an output linguistic variable, and performs the defuzzification concerning all the linguistic values of the linguistic variable. A defuzzification node has several input links each of which is emitted from an OR node. The output link of a defuzzification node may be connected to the MF nodes of its linguistic values when the defuzzification node represents an intermediate linguistic variable, or may be one of the outputs of the network when the defuzzification node represents an output linguistic variable.

  Suppose that the correlation-product inference and the fuzzy centroid defuzzification scheme [4] are used. The function of a defuzzification node l is defined as follows:

$$z_l = \frac{\sum_{k \in P_l}(x_{lk} a_k c_k)}{\sum_{k \in P_l}(x_{lk} a_k)}, \tag{4}$$

  where a_k and c_k are the area and centroid of the membership function for a linguistic value of an linguistic variable in the THEN-part represented by OR node k, respectively, and $x_{lk} = z_k$. For a bell-shaped membership function, $a_k = \sqrt{2\pi}\sigma_k$, where σ_k is the variance of the membership function. The weights of the input links of a defuzzification node are unity.

3 The Competitive Backpropagation Learning Algorithm

We propose a two-phase learning procedure for our fuzzy expert network. The first phase is a *competitive backpropagation (CBP)* training phase, and the second phase is a rule-pruning phase. The basic concept of the *CBP* learning algorithm [9, 11] is to choose *competitive* operations, e.g., Eq.(2) and (3) , for the nodes in a neural network and to train the network by the process of the *backpropagation* learning algorithm.

In the first phase of the learning procedure, the process of backpropagation learning method is performed to minimized the error function: $E = \frac{1}{2}\sum_{l=l_1}^{l_M}(T_l - z_l)^2$, where M is the number of output linguistic variables, and T_l and z_l are the target and the actual output values of a defuzzification node l which represents an output linguistic variable. Since the proposed fuzzy expert network is acyclic, it is guaranteed that the network has stable forward activation and backward error propagation [6].

Let N_i be the set of the indices of nodes each of which has an input link emitted from node i. The definition of the Delta values for **OR**, **AND** and **MF** nodes, the gradients of E with respect to the learnable weights, and the adjustments of learnable weights are described as follows:

- **The definition of Delta values:**

  For a defuzzification node l, the definition of the Delta value, δ_l, is defined as follows:

$$\delta_l = \frac{\partial E}{\partial z_l}$$

$$= \begin{cases} -\,(T_l - z_l)\ , & if\ node\ l\ is\ an\ output\ linguistic\ variable\ , \\ -\,\{\sum_{i\in N_l}[\delta_i\ \frac{z_i(z_l-c_i)}{\sigma_i{}^2}]\}\ , & if\ it\ is\ an\ intermediate\ linguistic\ variable\,. \end{cases} \tag{5}$$

For an OR node k, the definition of the Delta value is defined as follows:

$$\begin{aligned} \delta_k &= \frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial z_l}\frac{\partial z_l}{\partial z_k} \\ &= \delta_l\,\frac{\sigma_k(c_k - z_l)}{\sum_{k'\in P_l}(x_{lk'}\sigma_{k'})}\,, \end{aligned} \tag{6}$$

where the output link of node k is connected to defuzzification node l only.

For an AND node j, the definition of the Delta value is defined as follows:

$$\begin{aligned} \delta_j &= \frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial z_j} \\ &= \sum_{k\in N_j} \delta_k \cdot \begin{cases} w_{kj}\ , & if\ x_{kj}w_{kj} = MAX_k\ , \\ 0\ , & otherwise\ . \end{cases} \end{aligned} \tag{7}$$

For an MF node i, the definition of the Delta value is defined as follows:

$$\begin{aligned} \delta_i &= \frac{\partial E}{\partial z_i} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial z_i} \\ &= \sum_{j\in N_i} \delta_j \cdot \begin{cases} 1\ , & if\ x_{ji} = MIN_j\ , \\ 0\ , & otherwise\ . \end{cases} \end{aligned} \tag{8}$$

- **The gradient of E with respect to the weight of a fuzzy rule:**
  For the weight w_{kj} of a link connected from AND node j to OR node k, the gradient of E with respect to w_{kj} is evaluated as follows:

$$\begin{aligned} \nabla E_{w_{kj}} &= \frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial w_{kj}} \\ &= \delta_k \cdot \begin{cases} x_{kj}\ , & if\ x_{kj}w_{kj} = MAX_k\ , \\ 0\ , & otherwise\ . \end{cases} \end{aligned} \tag{9}$$

- **The adjustment of a learnable weight:**
  The adjustment of a learnable weight w_{kj}, which is based on the gradient descent search, can be described as follows:

$$w_{kj}(t+1) = w_{kj}(t) - \beta\,\nabla E_{w_{kj}}\ ,$$

  where β is the learning rate.

The knowledge of the fuzzy rules learned by the proposed *CBP* algorithm is distributed over the weights of the links between AND and OR nodes.

4 The Rule-Pruning Method

In general, after an idea fuzzy rule learning, a sound rule base is expected to be obtain. A sound rule base is defined as that for an output linguistic variable, at most one consequence can be implied from each possible antecedent in such a rule base. However, since the knowledge of fuzzy rules learned by the *CBP* training is distributed over the weights on the input links of OR nodes, there may be rules with identical antecedents and output linguistic variables but with different output linguistic values. Therefore, a pruning method is required to determine which rule in such a group of rules should remain while the others are pruned to form a sound rule base.

The physical meaning of the weights on the input links of OR nodes is explained in the following. After the *CBP* training, the learned weights w_{kj} on a set of links from an AND node j to the OR nodes of an output linguistic variable F_l (see Figure 3) are interpreted as the weights (or certainty factors) of a set of fuzzy rules which contain the same *IF*-part and are related to the same output linguistic variable with different linguistic values.

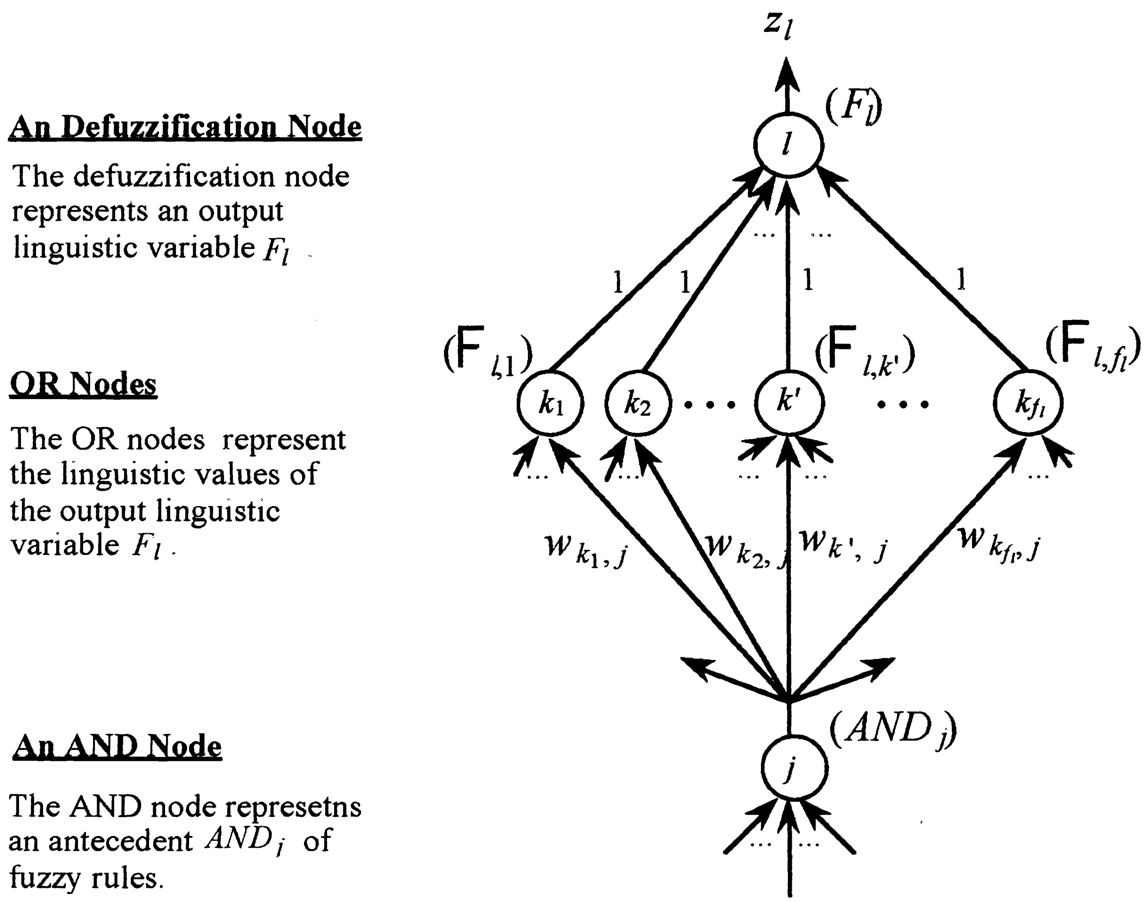


Figure 3: The diagram of the possible fuzzy rules with identical antecedent AND_j for an output linguistic variable F_l.

As an example, Figure 3 can be interpreted as the following fuzzy rules:

$$\begin{array}{lll}
R_{j,l,1}: & \text{If } AND_j, \text{ then } F_l \text{ is } \mathcal{F}_{l,1}. & (w_{k_1,j}) \\
R_{j,l,2}: & \text{If } AND_j, \text{ then } F_l \text{ is } \mathcal{F}_{l,2}. & (w_{k_2,j}) \\
\vdots & & \\
R_{j,l,k'}: & \text{If } AND_j, \text{ then } F_l \text{ is } \mathcal{F}_{l,k'}. & (w_{k',j}) \\
\vdots & & \\
R_{j,l,f_l}: & \text{If } AND_j, \text{ then } F_l \text{ is } \mathcal{F}_{l,f_l}. & (w_{k_{f_l},j})
\end{array}$$

where AND_j is the antecedent represented by an AND node j. The effects of these rules are that when the antecedent AND_j holds, each of the rules is activated to a certain degree represented by the weight value (the certainty factor) associated with that rule.

For the output linguistic variable F_l, at most one of the rules listed above could exist in a sound rule base. In the rule pruning phase, the fuzzy rules with identical antecedents and output linguistic variables are combined into one rule. Then, the redundant rules are deleted. An evaluation equation is proposed here based on the concept of the centroid of gravity, which is also the basis of the defuzzification scheme described in Section2. The centroid of these rules is evaluated according to the areas and centroids of the membership functions (of the linguistic values for the output linguistic variable) and the weights of these rules.

The equation for determining the remaining rule in the group of fuzzy rules represented by the links from an AND node j to the OR nodes of an output linguistic variable F_l is defined as follows [11]:

$$C_{lj} = \frac{\sum_k (w_{kj}\sigma_k c_k)}{\sum_k (w_{kj}\sigma_k)}\,, \tag{10}$$

where $k \in N_j$ and $k \in P_l$.

We divide the space of the output linguistic variable F_l into several nonoverlapped intervals corresponding to the membership functions of the linguistic values of F_l. The computed value of C_{lj} will be in one of the intervals. Then we prune all the other rules (represented by weight connections) by setting their weights equal to zero.

For each AND node, the pruning process is performed to delete its redundant output links connected to the OR nodes associated with an output or intermediate linguistic variable. After the pruning phase, a sound rule base is obtained.

5 Simulation and Comparison

A general purpose simulator of the proposed fuzzy expert network has been implemented. Several simple fuzzy expert systems were simulated to observe the learning ability of the fuzzy expert network. For an exemplar fuzzy expert system, there are five linguistic variables A, B, C, D, and E. Their linguistic values are defined as:

1. A has three linguistic values A_1, A_2, and A_3.

2. B has three linguistic values B_1, B_2, and B_3.

3. C has three linguistic values C_1, C_2, and C_3.

4. D has four linguistic values D_1, D_2, D_3, and D_4.

5. E has four linguistic values E_1, E_2, E_3, and E_4.

The inference relationship between these variables is:

1. C is inferred from A and B, and

2. E is inferred from C and D.

Therefore, A, B, and D are input linguistic variables, C is an intermediate linguistic variable, and E is an output linguistic variable. The target fuzzy rule bases are shown in Table 1.

Table 1: The target rule base for the exemplar problems

B \ A	A_1	A_2	A_3
B_1	C_3	C_3	C_3
B_2	C_2	C_3	C_3
B_3	C_1	C_2	C_3

D \ C	C_1	C_2	C_3
D_1	E_3	E_4	E_4
D_2	E_2	E_3	E_4
D_3	E_1	E_2	E_3
D_4	E_1	E_1	E_2

From the simulation results of this fuzzy expert system, after 220 *CBP* training epochs, the error rate was about 6%. Then the pruning process was performed to delete the redundant links with a slightly increased error rate, 7%. Table 2 shows the learned rule-base for the exemplar fuzzy expert system.

6 Concluding Remarks

A fuzzy expert network for rule learning of a fuzzy expert system is proposed and presented. The learning procedure of the network is divided into two phases. The first one is a competitive backpropagation (*CBP*) training phase, and the second one is a rule-pruning phase. In the first phase, the *CBP* learning algorithm enables the network to acquire the knowledge of fuzzy rules

Table 2: The retrieved rule base after 220 *CPB* learning epochs, and rule-pruning processes. The * symbol indicates the difference between target rule-base and learned rule-base.

B \ A	A_1	A_2	A_3
B_1	C_3	C_3	C_3
B_2	*C_3	*C_2	C_3
B_3	C_1	C_2	*C_2

D \ C	C_1	C_2	C_3
D_1	*E_2	E_4	E_4
D_2	E_2	E_3	E_4
D_3	E_1	E_2	*E_2
D_4	E_1	E_1	*E_1

quickly and precisely. The main reason for these advantages is that the gradient descent search approach in the *CBP* algorithm enables the network to learn more precisely than typical competitive learning algorithms, while the dedicated structure of the network and the competitive characteristics of the *CBP* algorithm enable the network to converge much more rapidly than conventional backpropagation learning algorithms. The knowledge of fuzzy rules learned in the *CBP* training phase is distributed over the learnable weights of the network. Therefore, in the second phase of the learning procedure, a pruning process is performed to convert the distributed knowledge of fuzzy rules learned by the *CBP* training into itermizable rule bases.

In the near future, we plan to further study on the learnability of the parameters of membership functions and on the choosing of the competitive operations performed by the nodes in a fuzzy expert network.

References

[1] H.C. Fu and J.J. Shann, "Fuzzy Expert Networks," in *the Proc. of ICANN'93*, Sep. 13-16, 1993, Amsterdan, Netherlands.

[2] S.I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-Propagation Algorithm," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 801-806, Sep., 1992.

[3] S.G. Kong and B. Kosko, "Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer," *IEEE Trans. on Neural Networks*, Vol. 3, No. 2, pp. 211-223, March, 1992.

[4] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamic Systems Approach to Machine Intelligence* , Prentice-Hall, Inc., New Jersey, 1992.

[5] R. Krishnapuram and J. Lee, "Fuzzy-Set-Based Hierarchical Networks for Information Fusion in Computer Vision," *Neural Networks*, Vol. 5, pp. 335-350, 1992.

[6] R.C. Lacher, S.I. Hruska, and D.C. Kuncicky, "Back-Propagation Learning in Expert Networks," *IEEE Trans. on Neural Networks*, Vol. 3, No. 1, Jan., 1992.

[7] C.T. Lin and C.S.G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," *IEEE Trans. on Computers* , Vol. C-40, No. 12, Dec., 1991.

[8] D. Nauck and R. Kruse, "A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation," in *Proceedings of International Conference on Neural Networks (ICNN'93)*, pp. 1022-1027, 1993.

[9] J.J. Shann and H.C. Fu, "A Fuzzy Neural Network for Knowledge Learning," in *the Proc. of Fifth IFSA World Congress*, July 4-9, 1993, Seoul, Korea

[10] J.J. Shann and H.C. Fu, "Backpropagation Learning for Acquiring Fine Knowledge of Fuzzy Neural Networks," in *the Proc. of WCNN'93*, July 11-15, 1993, Portland, Oragan, U.S.A.

[11] J.J. Shann and H.C. Fu, "A Fuzzy Neural Network for Acquiring Fuzzy Rule Bases," to be appeared in *the Proc. of International Symposium on Artificial Neural Networks (ISANN'93)*, Dec. 20-22, 1993, Hsinchu, Taiwan, R.O.C.

[12] L.A. Zadeh, "The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning I, II, III," *Information Sciences*, Vol. 8, pp. 199-249, pp. 301-357, Vol. 9, pp. 43-80, 1975.

[13] L.A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.