A New Fuzzy Supervised Learning Algorithm

Kwang Baek KIM[a], Chang Keun YUK[b], and Eui Young CHA[b]

a. Major in Computer Engineering, Division of Computer Information Engineering, Silla University,
San 1-1, Kwaebop-dong, Sasang-gu, Pusan, 617-736, Korea
Tel:+82-51-309-5052 e-mail:gbkim@lotus.silla.ac.kr
b. Department of Computer Science, Pusan National University,
San 30, Jangjeon, Keumjung, Pusan 609-735, Korea
Tel:+82-51-510-2219 e-mail:eycha@harmony.cs.pusan.ac.kr

Abstract

In this paper, we propose a new fuzzy supervised learning algorithm. We construct, and train, a new type fuzzy neural net to model the linear activation function. Properties of our fuzzy neural net include : (1) a proposed linear activation function; and (2) a modified delta rule for learning algorithm. We applied this proposed learning algorithm to exclusive OR, 3 bit parity using benchmark in neural network and pattern recognition problems, a kind of image recognition.

Keywords Fuzzy Neural Net, Fuzzy Perceptron, Linear Activation Function, Smoothing Method, Edge Detection

1. Introduction

In the conventional single layer perceptron, it is inappropriate when a decision boundary for classifying input pattern does not composed of hyperplane. Moreover, the conventional single layer perceptron, due to its use of unit function, was highly sensitive to change in the weights, difficult to implement and could not learn from past data[1]. Therefore, it could not find a solution of the exclusive OR problem, the benchmark.

There are a lot of endeavor to implement a fuzzy theory to artificial neural network[2]. Goh et al.[3] proposed the fuzzy single layer perceptron algorithm, and advanced fuzzy perceptron based on the generalized delta rule to solve the XOR problem, and the classical problem[3]. This algorithm guarantees some degree of stability and convergency in application using fuzzy data, however, it causes an increased amount of computation and some difficulties in application of the complicated pattern recognition. However, the enhanced fuzzy perceptron has shortcomings such as the possibility being located in local minima and slowness learning time[4].

In this paper, we propose a new fuzzy single layer supervised learning algorithm. We construct, and train, a new type of fuzzy neural net to model the linear function. Properties of this new type of fuzzy neural net include : (1) proposed linear activation function; and (2) a modified delta rule for learning. We will show that such properties can guarantee to find solutions for the problems-such as exclusive OR, 3 bit parity, 4 bit parity and image pattern recognition which a simple perceptron and a simple fuzzy perceptron can not.

2. A new fuzzy supervised learning algorithm

The learning algorithm for our single layer perceptron will be proposed. Before we discuss the new learning algorithm, we introduce the proposed learning architecture. Fig.1 shows the architecture of the new learning algorithm.

2.1 A new fuzzy supervised learning algorithm

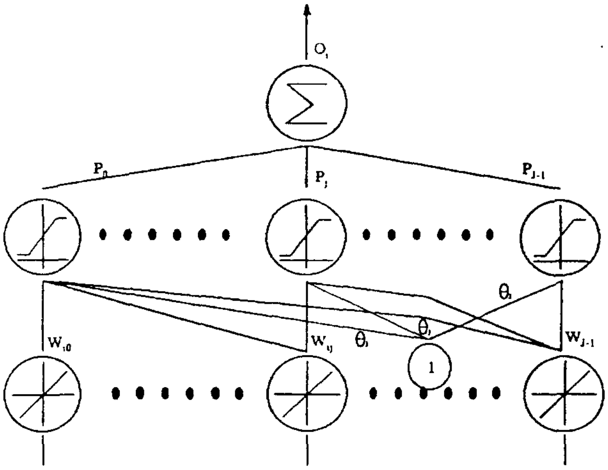

Fig. 1 A proposed fuzzy supervised learning architecture

A proposed learning algorithm can be simplified and divided into four steps. For each input, repeat step 1, step 2, step 3, and step 4 until error is minimized

Step 1 : Initialize weight and bias term.
Define W_{ij}, $(1 \leq i \leq I)$, to be the weight from input j to output i at time t, and θ_i to be the bias term in the output soma. Set $W_{ij}(0)$ to small random values, thus initializing all the weights and bias term.

Step 2 : Rearrange A_j according to the ascending order of membership degree m_j and add an item m_0 at the beginning of this sequence.

$$0.0 = m_0 \leq m_1 \leq \ldots \leq m_j \leq m_J \leq 1.0$$

Compute the consecutive difference between the items of the sequence.

$$P_k = m_j - m_{j-1}$$
$$\textit{Where } k = 0, \ldots, n$$

Step 3 : Calculate a soma (O_i)'s actual output.

$$O_i = \sum_{k=0}^{I-1} P_k * f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$$

Where $f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$ *is linear activation function*
Where $i = 1, \ldots, I.$

In the sigmoid function, if the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.0 and 0.25, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to $(\frac{1.0}{1.0+e^{-net}})$. If the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.25 and 0.75, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to 0.25. If the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.75 and 1.0, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to $(1-\frac{1.0}{1.0+e^{-net}})$.

Therefore, the proposed linear activation function expression is represented as follows :

$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = 1.0 \quad where(\sum_{j=k}^{I-1} W_{ij} + \theta_i) > 5.0$$

$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = \rho * (\sum_{j=k}^{I-1} W_{ij} + \theta_i) + 0.5$$
$$where -5.0 \leq (\sum_{j=k}^{I-1} W_{ij} + \theta_i) \leq 5.0, \ \rho \in [0.1, 0.4]$$

$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = 0.0 \quad where(\sum_{j=k}^{I-1} W_{ij} + \theta_i) < -5.0$$

The formulation of the activation linear function is following.

$$f((\sum_{j=k}^{I-1} W_{ij} + \theta_i)) = (\frac{1}{range*2}) * (\sum_{j=k}^{I-1} W_{ij} + \theta_i) + 0.5$$

where the range means monotonic increasing interval except for the interval between 0.0 and 1.0 of value of the $f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$.

Step 4 : Applying the modified delta rule. And we derive the incremental changes for weight and bias term.

$$\triangle W_{ij}(t+1) = \eta_i * E_i * \sum_{k=0}^{i} P_k * f(\sum_{j=k}^{l-1} W_{ij} + \theta_i) + \alpha_i * \triangle W_{ij}(t)$$

$$W_{ij}(t+1) = W_{ij}(t) + \triangle W_{ij}(t+1)$$

$$\triangle \theta_i(t+1) = \eta * E_i * f(\theta_i) + \alpha_i * \triangle \theta_i(t)$$

$$\theta_{ij}(t+1) = \theta_{ij}(t) + \triangle \theta_{ij}(t+1)$$

Where η_i *is learning rate* α_i *is momentum.*

Finally, we enhance the training speed by using the dynamical learning rate and momentum based on the division of soma.

if $(Inactivation_{totalsoma} - Activation_{totalsoma} > 0)$
then $\triangle \eta_i(t+1) = E^2$
$\eta_i(t+1) == \eta_i(t) + \triangle \eta_i(t+1)$

if $(Inactivation_{totalsoma} - Activation_{totalsoma} > 0)$
then $\triangle \alpha_i(t+1) = E^2$
$\alpha_i(t+1) == \alpha_i(t) + \triangle \alpha_i(t+1)$

2.2 Error criteria problem by division of soma

In the conventional learning method, learning is continued until squared sum of error is smaller than error criteria. However, this method is contradictory to the physiological neuron structure and takes place the occasion which a certain soma's output is not any longer decreased and learn no more[5]. The error criteria was divided into activation and inactivation criteria. One is an activation soma's criterion of output "1", the other is an inactivation soma's of output "0". The activation criterion is decided by soma of value "1" in the discriminant problem of actual output patterns, which means in physiological analysis that the pattern is classified by the activated soma. In this case, the network must be activated by activated somas. The criterion of activated soma can be set to the range of [0,0.1].

In this paper, however, the error criterion of activated soma was established as 0.05. On the other hand, the error criterion of inactivation soma was defined as the squared error sum, the difference between output and target value of the soma. The degree of activation and inactivation was equally set up on the basis that activation and inactivation is the same[5,6]. Fig.2 shows the proposed algorithm.

```
while ((Activation_no == Target_activated_no)
      &&(Inactivation_error <= Inactivation_area))
    do {
      for (i=0; i<Pattern_no; i++)
        for(j=0; j<Out_cell_no;j++) {
           Forward Pass;
           Backward Pass;
           if(Out_cell=Activation_soma&&|error|<=
                  Activation_area)
               Activation_number++;
           if (Out_cell=Inactivation_soma)
               Inactivation_error += error * error;
        }
    }
```

Fig.2 Learning Algorithm by division of soma

3. Simulation & Result

We simulated our method on IBM PC/586 with C++ language. In order to evaluate the proposed algorithm, we applied it to the exclusive OR, 3 bit parity using benchmark in neural network and pattern recognition problems, a kind of image recognition. In the proposed algorithm, the error criteria of activation and inactivation for soma was set to 0.09.

3.1 Exclusive OR and 3 bit parity

Here we set up initial learning rate 0.5 initial momentum 0.75 respectively. Also we set up the range of weight [0,1]. In general, the range of weights were [-0.5,0.5] or [-1,1].

As shown in Table 1 and Table 2 our model showed higher performance than fuzzy perceptron in convergence epochs and convergence rates of the three tasks.

Table 1. Comparison of step number

Epoch No	Fuzzy Perceptron	Proposed Algorithm
Exclusive OR	8 (converge)	4 (converge)
3 bit parity	13 (converge)	8 (converge)
4 bit Parity	0 (not converge)	15 (converge)

Table 2. Convergence rate in initial weight range

Appiled Problem	Fuzzy Perceptron	Proposed Algorithm	Initial Weight Range
exclusive OR	100%	100%	[0.0, 1.0]
	89%	99%	[0.0, 5.0]
3 bit parity	83%	97%	[0.0, 1.0]
	52%	96%	[0.0, 5.0]
4 bit parity	0%	96%	[0.0, 1.0]
	0%	95%	[0.0, 5.0]

3.2 Image pattern recognition

The procedure of image pre-processing is presented in Fig.3 and the example images we used are shown in Fig.4

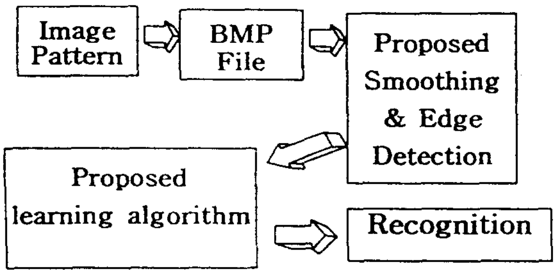

Fig.3 Preprocessing diagram

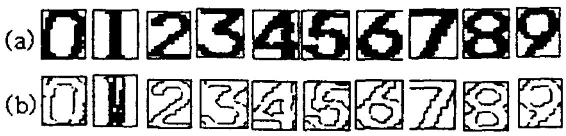

Fig.4 (a) Image pattern and (b) training image pattern by edge detection

We carried out image pre-processing in order to prevent high computational load as well as loss of information. If the original images were used as training patterns, it requires expensive computation load. In contrast skeleton method causes loss of important information of images. To overcome this trade-off, we used edge information of images.

The most-frequent value method we had developed was used for image pre-processing. This method was used because blurring of boundary when a common smoothing method was used. Thus, it degrades both color and contour lines[7,8]. The new method replaced a pixel's value with the most frequent value among specific neighboring pixels. If the difference of absolute value between neighborhood is zero in a given area, the area was considered as background. Otherwise, it was considered as a contour. This contour was used as a training pattern.

The input units were composed of 32 * 32 array for image patterns. In simulation, the fuzzy perceptron was not converged, but the proposed method was converged on 70 step at image patterns. Table 3 is shown the summary of the results in training epochs between two algorithms.

Table 3. The comparison of epoch number

Image Pattern	Epoch Number
fuzzy perceptron	0 (not converge)
proposed algorithm	70 (converge)

4. Conclusions

The study and application of fusion fuzzy theory with logic and inference and neural network with learning ability have been actually achieving according to expansion of automatic system and information processing, etc.

We have proposed a fuzzy supervised learning algorithm which has greater stability and functional varieties compared to the conventional fuzzy perceptron, and a new learning algorithm for the network with enhanced fuzzy learning algorithm that for the first time, allows automatic extraction of fuzzy relations from data.

The proposed network is able to extend the arbitrary layers and has high convergence in case of two layers or more. When we considered only the case of the single layer, the networks had the capability of high speed during the learning process and rapid processing on huge image patterns.

In the future study direction, we will develop a novel fuzzy learning algorithm and apply to the face recognition.

References

1. F. Rosenblatt, "The perceptron : A perceiving and recognizing automaton," Cornell Univ., Ithaca, NY, Project PARA Cornell Aeronaut Lab, Rep., 85-460-1, 1957

2. M.M. Gupta and J. Qi, "On Fuzzy Neuron Models," IJCNN, Vol.2, pp.431-435, 1991.

3. TH Goh, PZ Wang, and HC Lui, "Learning Algorithm for Enhanced Fuzzy Perceptron," IJCNN. Vol2, pp.435-440, 1992.

4. Kwang Baek Kim, Eui Young Cha, "A New Single Layer Perceptron using Fuzzy Neural Controller," Simulators International XII, Vol.27, No.3, pp.341-343, 1995.

5. Kwang Baek Kim, Ji Ae Jun, and Eui Young Cha, "The Solution Method of Learning Time and Local Minima of Error Backpropagtion by the Nervous System," JCEANF, pp.592-601, 1992.

6. Judith E.Dayhoff, "Biological Synapse," Neural Network Architectures, pp.136-162, 1989.

7. D. G. Lowe, "Organization of Smooth Iamge Curves at Multiple Scales,", International journal of computer vision, 1:119-130, 1989.

8. R. B. Paranjape, R. N. Rangayyan, W. M. Morrow, and H. N. Nguyen, Adaptive neighborhood image processing, CVGIP Graphical Models and Image Processing, 54(3):259-267, 1992.