Multilayer neural network with a fuzzy controlled learning method for optical pattern training

Xin Lin,* MEMBER SPIE
National Agriculture Research Center
3-1-1 Kannondai
Tsukuba 305-8666
Japan

Masahiko Mori
Electrotechnical Laboratory
1-1-4 Umezono
Tsukuba 305-8568
Japan

Abstract. A fast learning method for neural networks learning system using fuzzy controlling is proposed. By the fuzzy controlled theory, the learning rate and the nonlinear gain of the output function in the backpropagation algorithm are adjusted based on training error and training time. The effectiveness of this learning method is demonstrated in optical pattern training. © 2000 Society of Photo-Optical Instrumentation Engineers. [S0091-3286(00)02210-8]

Subject terms: pattern training; fuzzy controlling; pattern recognition; neural network.

Paper 990501 received Dec. 15, 1999; revised manuscript received Apr. 4, 2000; accepted for publication May 5, 2000.

1 Introduction

One of the major functions of human brains is storing learned knowledge for information processing and pattern recognition. The motivation for the development of neural networks has been to mimic the operation of human brains for intelligent information processing. Specifically, the multilayer neural network that employ the backpropagation (BP) algorithm for learning has been widely applied to pattern recognition and its effectiveness has been shown. However, time for training an artificial neural network using a standard backpropagation (SBP) algorithm is too long to be accepted in some cases. To accelerate learning process of SBP, a number of improved learning approaches have been proposed, for example, BP with momentum term, BP with a modified error function, second order BP, and so on.

Our investigation reveals that one of the reasons for slow training is that the learning rate is a constant in SBP, because a fixed learning rate cannot be established a priori due to a primary problem of overshooting the goal. Another reason is that the nonlinear gain of the output function of the neural network in SBP is not changeable. Thus the network has unchangeable responses to all input neurons during the whole learning. In fact, the quality of training artificial neural networks depends on the expertise and knowledge of experts, and they are what’s fuzzy in nature. Fuzzy sets were introduced by Zadeh in 1965 as a means for representing, manipulating, and utilizing information that possesses nonstatistical uncertainty. In recent years, fuzzy set theory and fuzzy logic have been widely applied to optical image processing, optical associative memory, and multilayer neural networks since it is extremely flexible to accommodate various and minute variations in data.

In this paper, the fuzzy technique is applied to help expedite the training speed of the neural network for optical pattern training. Two crucial components in a BP algorithm, the learning rate and the nonlinear gain of the output function, are adjusted based on the training error and the training time, and the fuzzy controlling functions for learning rate and nonlinear gain are obtained based on the fuzzy rules and inferences. The numerical simulation results indicate that the proposed fuzzy controlled learning method works well for pattern training.

2 BP Algorithm for Optical Pattern Recognition

The multilayer pattern recognition neural network is a feedforward system with one or more hidden layers between the input and the output neural layers. Figure 1 shows a three-layer neural network. The input to the j’th neuron in the l’th layer can be written as

\[ X_j(l) = \sum_i W_{ji} Y_i(l-1) - \theta_j, \]

where \( Y_i(l-1) \) is the output of the i’th neuron in the \((l-1)’\)th layer, \( W_{ji} \) is the interconnection weight from the i’th neuron in layer \((l-1)’\) to the j’th neuron in layer l, and \( \theta_j \) is a thresholding value, that can be implemented by adding an extra bias neuron as shown in Fig. 1. In the input layer, the output of the i’th neuron is

\[ Y_i(1) = X_i(1) = x_i. \]

For neurons in other layers, the output is a monotonic nonlinear function (for instance, a sigmoidal function) of its total input and is given by

\[ Y_j(l) = f(X_j(l)) = \frac{1}{1 + \exp[-\alpha X_j(l)]}, \quad l > 1, \]

where \( \alpha \) is the nonlinear gain, and it is a positive control constant in SBP. Assume that the training set consists of P patterns. The error BP is an iterative gradient algorithm designed to minimize the mean square error \( \varepsilon_p \) (usually
called the training error) between the actual output $Y_{k,p}$ and the desired output $T_{k,p}$ (for $p$th training pattern),

$$e_p = \frac{1}{2} \sum_{k=1}^{N} (Y_{k,p} - T_{k,p})^2,$$

where $N$ denotes the number of neurons in the output layer of the neural network. Here the mean square error is applied after each pattern presented. A detailed discussion for the mathematical rigor of this approach can be found in Ref. 14.

Initially, small random values are assigned to all interconnection weights $W_{ij}$. Each pattern in the training set is then presented to the neural network one by one. The weights are corrected according to the difference between the actual output and desired output. The weights learning rule is obtained by applying the gradient descent method:

$$W_{ij}(t+1) = W_{ij}(t) - \eta \frac{\partial e_p}{\partial W_{ij}},$$

where $\eta$ is the learning rate. In SBP, the learning rate is a positive constant. It decides the size of the correction of the interconnection weights. Note that $t$ (usually called the training time) denotes the number of the iteration currently in progress, and is incremented by 1 for each training sweep through the whole set of input-output cases. The training process is repeated until the desired output can be obtained for each pattern in the training set.

In optical pattern recognition applications, the images are almost two-dimensional (2-D) structures. Therefore the preceding mathematical model should be extended from one to two dimension. Such an extension is straightforward and will not be repeated here. We have constructed a very simple optical network for pattern learning by use of the SBP is shown in Fig. 2 in a previous paper. The optical system uses an electrically addressed liquid crystal device (LCD) to display the multiple image of an input pattern, a Pockels readout optical modulator (PROM) for the weight tensor is stored, a blue LED array for the error-signal matrix is displayed, and a photodetector array (PDA) for the resultant tensor image is detected and summed locally. The learning rate is determined by the intensity and illumination time of the blue LED array. For such optical learning networks, we can adjust the intensity of the blue LED array according to the calculation of a fuzzy membership function with the training error and the training time.

3 Principles of Fuzzy Controlling

A fuzzy system can be described as a set of fuzzy logical rules. Fuzzy logical rules must be understood as propositions associated with possibility distributions. The construction of a fuzzy system includes steps to (1) identify system parameters and define fuzzy sets memberships, (2) determine fuzzy rules, (3) select inference strategy, and (4) determine defuzzification strategy. In this paper, the proposed fuzzy learning system is built as follows.

3.1 System Parameters (i.e., System Inputs and Output) and Memberships

In this fuzzy learning system, two learning parameters, the learning rate and nonlinear gain, are adjusted based on the training error and training time. Thus, the elements of the fuzzy set in the input space of the system are training error and training time, and in the output space they are the learning rate or the nonlinear gain. The mapping between elements of the universe and their corresponding degrees of membership in the fuzzy sets is referred to as the membership function of the fuzzy set.

3.2 Fuzzy Controlling Rules

The general form of fuzzy rules is "if $x$ is $A$ then $y$ is $B$," where $x$ and $y$ are the parameters in input and output space of a system, respectively, and $A$ and $B$ are fuzzy sets that are defined by the membership functions. The "$x$-$A$" is called the premise part of the fuzzy rule, and "$y$-$B$" is called the consequent part of the fuzzy rule. In our fuzzy learning system, the premise part has two fuzzy sets, training error $e_p$ and training time $t$, and the consequent part has one fuzzy set, the learning rate or the nonlinear gain. The premise part is defined by "large" or "small," which are imprecise, fuzzy subsets based on the learning states of neural network. Associating these fuzzy subsets with the consequent part to make the fuzzy rules.
Table 1 Fuzzy rules for the learning rate $\eta(e_p,t)$.

<table>
<thead>
<tr>
<th>Training Time t</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>Medium</td>
<td>Large</td>
<td>Large</td>
</tr>
<tr>
<td>Medium</td>
<td>Small</td>
<td>Medium</td>
<td>Large</td>
</tr>
<tr>
<td>Long</td>
<td>Small</td>
<td>Small</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 2 Fuzzy rules for the nonlinear gain $\alpha(e_p,t)$.

<table>
<thead>
<tr>
<th>Training Time t</th>
<th>Small</th>
<th>Medium</th>
<th>Small</th>
<th>Medium</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>Medium</td>
<td>Large</td>
<td>Small</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>Large</td>
<td>Medium</td>
<td>Small</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long</td>
<td>Large</td>
<td>Large</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

First, we discuss the fuzzy rules of the learning rate. In the BP algorithm, the corrections of the interconnection weights depend on the learning rate and the training error. When the training error is very large, the learning rate should be large, indicating that the weights are away from the desired value. When the training error is very small, the learning rate should become small, showing that the weights are modified close to the desired value. In addition, a small learning rate can prevent emmation and oscillation of the networks. On the other hand, if the training time is quite short, the learning rate should be large to promote the training speed. If the training time is quite long, the learning rate should become small so that a better convergence state is reached in the final stage. The fuzzy rules for the learning rate are tabulated in Table 1. Each rule describes a partial behavior of the learning system and represents a fuzzy relation from the input space to the output space.

Next, we discuss the fuzzy rules of the nonlinear gain. We use a sigmoid function $f(x)=1/(1+e^{-ax})$ as the output function of the neural networks. The nonlinear gain $\alpha$ makes the nonlinear effectiveness of the function. Figure 3 shows the sigmoid functions with different nonlinear gain $\alpha$. We can see that when the nonlinear gain is small, the sigmoid function close to a linear function. Thus if the training error is very large and training time is quite short, the smaller nonlinear gain can relax all input neurons, and the initial weights can be adjusted quickly and easily. When the nonlinear gain is large, the sigmoid function approaches a step function. Thus if the training error is very small and the training time is quite long, the nonlinear gain should be large, such that the weights are convergent toward the desired value. The fuzzy rules for the nonlinear gain are tabulated in Table 2.

3.3 Fuzzy Inferences and Membership Functions

Once the fuzzy rules are obtained, the next problem is how to utilize these rules in a given situation to deduce, infer and use the coefficients for the data and/or knowledge. That is what the fuzzy inference does, i.e., the results of fuzzy inference can be obtained based on the fuzzy rules and membership functions. The form of the membership function is determined according to the complexity of the problem. In the fuzzy control theory, it can be a classification linear function, such as a triangle or a trapezoid function, or a curve function, such as a Gaussian function. The linear function is simple for analyzing and calculating and can obtain well enough results. The triangle and the trapezoid function are used in our method. Figure 4 shows the membership functions for the control parameters, the training error $e_p$, the training time $t$, the learning rate $\eta(e_p,t)$, and the nonlinear gain $\alpha(e_p,t)$. The values of the abscissa are summarized in a table below the figure. They are obtained with the help of the trial-and-error method based on the structure of the networks and heuristic knowledge. The range of the abscissa determines the region of fuzzy controlling and it also affects the computed results. But an adaptive control range can give universal correctness for similar problem.

3.4 Defuzzification Strategy

Defuzzification is a mapping from a fuzzy space into a crisp space. In the fuzzy set theory, the centroidal defuzzification technique is used generally. It utilizes all the information in the fuzzy distribution to compute the centroid. For a discrete universe, the fuzzy output $y$ is divided into $n$
sections, and the median of the section is defined as \( c_i \) (see Fig. 5), then the crisp output of the fuzzy inference \( y \) can be written as:

\[
y = \frac{\sum_{i=1}^{n} m(c_i) c_i}{\sum_{i=1}^{n} m(c_i)}.
\]  

(6)

where \( m(c_i) \) is the value of the membership function corresponding to centroid \( c_i \), and \( y \) is a crisp value and its universe is a crisp set. For our fuzzy learning method, the outputs of the fuzzy inferences are the crisp universes of the learning rate or the nonlinear gain, and they are obtained with the help of Eq. (6) and the membership functions in Fig. 4. Using these crisp values for the learning rates and the nonlinear gains, the adjusting functions for networks training are given as shown in Fig. 6. Figures 6(a) and 6(b) show the adjusting learning rate and nonlinear gain functions, respectively. Both functions cannot be described easily by simple mathematical formulas. Using two adjusting functions, the training of the neural networks can be controlled automatically based on the training error and training time.

4 Computer Simulations

To demonstrate the effectiveness of the proposed fuzzy learning method for pattern training, we conducted computer simulations using a three-layer neural network that consists of an input layer, a hidden layer and an output layer, as shown in Fig. 1. In principle, there is no limitation to the scale of this neural network and a network with 512×512 input neurons can be constructed to handle the 2-D images acquired by image sensors such as a CCD and focal-plane arrays. There is no limitation to the classes of recognized pattern if we use a large-scale multilayer network. However, for a feasibility check, we implemented only a small-scale network with 17×17 input neurons.

Three sets of 2-D binary patterns are used for network training, as shown in Fig. 7. Each set consists of 10 patterns with 17×17 pixels as the 10 classes. The input layer contains 17×17 neurons, which is the same as the number of pixels in the input pattern. Determining the number of neurons in the hidden layer is difficult because there is no deterministic method of obtaining an optical value. Here, the hidden layer has 51 neurons (50+1 bias neurons) that are obtained with using a trial-and-error simulation and a heuristic law, \( N > m(m - 1)/2 \), where \( N \) is the number of neurons in the hidden layer and \( m \) is pattern classes. The output layer has 10 neurons, which corresponds to the number of classes.

During training, the desired output for the neuron corresponding to the right class is 1, whereas the desired output

![Fig. 5 Example of centroidal defuzzification for learning rate.](image)

![Fig. 6 Fuzzy adjusting functions for (a) the learning rate and (b) the nonlinear gain.](image)
for all the other neurons should be 0. The initial interconnection weights are set to random values in the range $[-0.5, 0.5]$, and are then updated by Eqs. (1) to (5) [Eqs. (1) to (5) are extended from one to two dimensional]. The learning rate $\eta$ in Eq. (5) and the nonlinear gain $\alpha$ in Eq. (3) are adjusted based on the two fuzzy adjusting functions that are shown in Figs. 6(a) and 6(b). The training process is repeated until the convergence criterion $\varepsilon_p \leq 0.015$. After training, the network is tested to recognize some patterns that are not in the training pattern sets. A detailed discussion for pattern recognition processing can be found in Ref. 4, which is our previous work.

5 Results and Discussion

We compared the proposed learning method with fuzzy adjusting functions with a SBP learning system using the same learning pattern sets, as shown in Fig. 7. In SBP, the nonlinear gain is a constant, which is defined as $\alpha = 1$ to guarantee that the network not only has enough nonlinearity but also has an effective activation area. The learning rate is also a constant. To choose an optimal value of the learning rate $\eta$, we tested $\eta$ for values from 0.1 to 2 with a step of 0.1 using the trial-and-error method. We found that with $\eta = 1.2$, the network is emanative, namely, it cannot converge to a desired output. We also found that training processing is very slow when $\eta \leq 0.5$. We select $\eta = 0.9$ for network training because the convergence speed is faster and network working is stable for $\eta$ close to 1. The convergence criterion is $\varepsilon_p \leq 0.015$, which is the same as in the fuzzy learning method. The training times of both learning methods have small changes with the initial interconnection weights because the values of initial weights are set at random. Therefore we measured the training times with 100 trials for each pattern set, and determined the best time, the worst time, the average time $\bar{t} = \frac{1}{M} \sum_{m=1}^{M} t_m$ ($M$ is total test time and $t_m$ is trial value), and the standard deviation

$$\sigma = \sqrt{\frac{\sum_{m=1}^{M} (t_m - \bar{t})^2}{M(M-1)}}.$$  Here the "time" denotes one "training time," namely a training cycle.

Computer simulation results for the two learning methods are tabulated in Table 3. When the network is trained by the three pattern sets, as shown in Fig. 7, the fuzzy learning method converged after 439, 488, and 459 average training times, respectively. In contrast, the SBP learning method, the three-layer network of the same size and training pattern sets required 2999, 3079, and 3031 average training times to converge, respectively. The standard deviations of the average values for two learning method are about 0.3.

These results are plotted as histograms for the first pattern set (the number set), as shown in Figs. 8 and 9. Figure

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Fuzzy Learning Method</th>
<th>SBP Learning Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best Time</td>
<td>Worst Time</td>
</tr>
<tr>
<td>Number set</td>
<td>430</td>
<td>446</td>
</tr>
<tr>
<td>Small letter set</td>
<td>480</td>
<td>498</td>
</tr>
<tr>
<td>Capital letter set</td>
<td>450</td>
<td>469</td>
</tr>
</tbody>
</table>
8 is for the fuzzy learning method in Fig. 8(a) and the SBP learning method in Fig. 8(b). The abscissa is the training time for each test and the ordinate is the number of trials. We can see that the training times are contained in the range [430, 446] for the fuzzy method and in the range [2992, 3006] for the SBP method. The average values are about 439 and 2999, respectively. Figure 9 is the learning curve with the training error against the training time for two learning methods. We can see that the learning speed is much improved in the fuzzy learning method.

6 Conclusion

We described a method for fast pattern training using the fuzzy adjusting functions. The proposed fuzzy learning method was proven to converge faster than the SBP learning method. The learning rate in an optical learning neural network, as shown in Fig. 2, can also be controlled by the proposed fuzzy adjusting function. We are doing this experiment and some initial results have been obtained. In practice, an optical implementation of the fuzzy set was also recently investigated.

However, several issues are currently under investigation, including the selection of the fuzzy membership function for different control problems for pattern recognition and the optimization of the structure and parameters of the network. On the other hand, much work has been done to try to improve the training performance of the SBP algorithm. A comparison of our fuzzy approach with other techniques applied to the same learning and recognition problem will be assessed in future studies.

References


Xin Lin received her PhD degree in optoelectronics from the Shizuoka University, Japan, in 1996. From 1997 to 1999, she held a domestic research fellowship at the Electrotechnical Laboratory. She became a researcher with the National Agriculture Research Center in 2000. Her research interests include optical metrology, optical information processing, pattern recognition, neural networks, and biomedical optics. She is a member of SPIE, the OSA, and the Japanese Society of Applied Physics.

Masaikho Mori graduated in 1984 from Keio University, Tokyo, Japan, where he received his MS degree in electrical engineering in 1986, and he joined the Electrotechnical Laboratory, where he performs research on the optical properties of semiconductor materials, optoelectronic devices and optical information processing systems. His research interest include optical neural networks and ultrashort optical pulse controlling. He is a member of the Japanese Society of Applied Physics.