A Genetic Fuzzy Neural Network for Pattern Recognition

Abraham Kandel*  
Department of Electrical Eng. Systems  
Tel-Aviv University  
Tel-Aviv 69978, ISRAEL

Yan-Qing Zhang and Horst Bunke†  
Department of Computer Science and Eng.  
University of South Florida  
Tampa, Florida 33620-5350 USA

Abstract

In this paper, a genetic fuzzy neural network for pattern recognition is proposed by applying genetic algorithms to the Kwan-Cai fuzzy neural network. A genetic-guided self-organizing learning algorithm is capable of reducing the number of fuzzy neurons and increasing recognition rates for the fixed number of output neurons. The simulations have indicated that the genetic fuzzy neural network can effectively recognize various distorted patterns with good recognition rates.

1 Introduction

Pattern recognition is an important application of both neural network techniques and fuzzy sets theory [1-10]. Various crisp neural networks such as the Hopfield network [8], the Hamming net [8], the Carpenter/Grossberg ART [2], Martin-Pittman BP-based neural net [9] and the Fukushima Neocognitron [3,4] have been used in pattern recognition. On the other hand, fuzzy sets theory is able to deal with uncertainty and fuzziness in pattern recognition, therefore fuzzy logic is a useful tool for simulating the human brain's perception and decision. In recent years, fuzzy neural networks have been used in pattern recognition [1,6,7]. Kosko proposed a Fuzzy Associate Memory (FAM) using fuzzy matrices to represent fuzzy mappings [6]. Kwan and Cai designed a fuzzy neural network to recognize the 26 English letters and the 10 Arabic numerals [7]. However, the Kwan-Cai Fuzzy Neural Network (FNN) has two disadvantages which are (1) the important parameter $\alpha$ is defined subjectively and (2) the learning algorithm may greatly increase the number of fuzzy neurons if many training patterns are not similar. To solve these problems, we have developed a genetic fuzzy neural network which is capable of using genetic algorithms to optimize the parameter $\alpha$ and reducing the number of fuzzy neurons. The simulations have indicated that the genetic fuzzy neural network can recognize various distorted patterns with high recognition rates.

2 Structure of a Genetic Fuzzy Neural Network

![Diagram of a Genetic Fuzzy Neural Network](image)

A Genetic Fuzzy Neural Network (GFNN) is a genetic-algorithms-based Kwan-Cai FNN, and consists of 4 layers (see Fig. 1). Each pattern has $N_1 \times N_2$ pixels and there are $K$ training patterns. For clarity, the functions of fuzzy

*On leave from Department of Computer Science and Engineering, University of South Florida, USA.
†On leave from Institut Fur Informatik und angewandte Mathematik, University of Bern, Bern, Switzerland.
neurons in different layers are described layer by layer as follows:

**Layer 1: Input Layer**

\[ N_1 \times N_2 \text{ input neurons in the first layer are oval nodes with simple normalization functions defined by} \]

\[
O_{ij}^{[1]} = \frac{x_{ij}^k}{\max_{i=1}^{N_1}(\max_{j=1}^{N_2}(\text{max}_k(x_{ij}^k)))},
\]

where \( x_{ij}^k \) is the \((i,j)\)th pixel value of the \(k\)th pattern for \(i = 1,2,...,N_1\), \(j = 1,2,...,N_2\) and \(k = 1,2,...,K\).

**Layer 2: Genetic Fuzzification Layer**

\( N_1 \times N_2 \) Genetic fuzzification neurons in the second layer are square nodes with either the triangular fuzzification functions used by Kwan and Cai [7] or the new Gaussian fuzzification functions.

The triangular fuzzification functions are given by:

\[
\gamma_{pqm} = \max_{i=1}^{N_1}(\max_{j=1}^{N_2}(O_{ij}^{[1]} \cdot \alpha))^2
\]

where \( f_{pq} = \max_{j=1}^{N_2}(O_{ij}^{[1]} e^{-\beta[(p-i)^2+(q-j)^2]}) \)

and \( \beta \) is parameters to adjust centers and widths of fuzzification functions for the second layer.

**Layer 3: Fuzzy Clustering Layer**

\( M \) fuzzy neurons in the 3rd layer are trapezoidal nodes with a min function defined by

\[
O_m^{[3]} = \min_{p=1}^{N_1}(\min_{q=1}^{N_2}(O_{pqm}^{[2]})),
\]

where \( m = 1,2,...,M \). \( M \) will be determined by the learning algorithm.

**Layer 4: Output Layer**

\( M \) output neurons in the 4th layer are triangular nodes with the functions defined by

\[
O_m^{[4]} = \begin{cases} 
0 & \text{for } \max_{m=1}^{M}(O_{pqm}^{[3]}) < \max_{m=1}^{M}(O_{pqm}^{[2]}) \\
1 & \text{for } \max_{m=1}^{M}(O_{pqm}^{[3]}) = \max_{m=1}^{M}(O_{pqm}^{[2]}),
\end{cases}
\]

where \( m = 1,2,...,M \).

### 3 A Genetic-Algorithms-Based Self-Organizing Learning Algorithm

By adding genetic algorithms into the Kwan-Cai learning algorithm [7], we propose a genetic-algorithm-based self-organizing learning algorithm which is able to reduce \( M \) by adjusting \( \alpha \). The total number of fuzzy neurons in a GFNN is \((1+M)N_1N_2 + 2M\). \( N_1 \) and \( N_2 \) are fixed, therefore we can only try to reduce \( M \) to reduce the complexity of the GFNN. The genetic-algorithm-based self-organizing learning algorithm is given below.

**Step 1:** Choose values of \( \alpha(\alpha \geq 0) \) and \( \beta \) for the second layer of the GFNN.

**Step 2:** \( m = 0 \) and \( k = 1 \).

**Step 3:** \( m = m + 1 \). Calculate:

\[
\gamma_{pqm} = \max_{i=1}^{N_1}(\max_{j=1}^{N_2}(O_{ij}^{[1]} \cdot \alpha)),
\]

where \( \theta = e^{-\beta[(p-i)^2+(q-j)^2]} \)

for \( p = 1,2,...,N_1 \) and \( q = 1,2,...,N_2 \).

**Step 4:**

If \( m < M_{max} \)

Then goto Step 5.

Else use genetic algorithms to adjust \( \alpha \):

The fitness function is given by:

\[
F = \sum_{k=1}^{K} \sum_{m=1}^{M_{max}} |O_m^{[4]} - y_k|,
\]

where \( O_m^{[4]} \) are the outputs of the 4th layer of the GFNN and \( y_k \) are the desired outputs for the \( k \)th training pattern.

Finally goto Step 6.

(Note: \( M_{max} \) is the maximum number of output neurons given by a user.)

**Step 5:** \( k = k + 1 \).

If \( k > K \)

Then goto Step 6.

Else calculate:

\[
\sigma = 1 - \max_{j=1}^{m}(O_{jk}^{[3]}),
\]

where \( O_{jk}^{[3]} \) is the output of the \( j \)th neuron in the 3rd layer for the \( k \)th training pattern.

**Step 6:** End.
4 Simulations

We use 10 kinds of the distorted 26 English letters and 10 Arabic numerals to verify the new GFNN. For example, 10 kinds of distorted patterns for “H” and “4” are shown in Figs. 2, 3, 4, 5 and 6. In the simulations, $N_1 = N_2 = 16$, $M_{max} = 36$, $K = 72$, $\beta = 0.3$, the initial $\alpha = 2.0$. We used 36 LA & 36 SM, 36 TT & 36 SQ, and 36 SH & 36 DC distorted patterns to train a GFNN, respectively, then used 360 distorted patterns to check recognition rates. In order to compare the GFNN with the Kwan-Cai FNN, we adopted the triangular fuzzification function (Eqn. 2) in the simulations. By using genetic algorithms, we finally get $\alpha = 1.684$ for the trained GFNN. The simulation results are given in Tables 1, 2 and 3. As a result, the new GFNN is better than the Kwan-Cai FNN because the GFNN can optimize $\alpha$ by using genetic algorithms.

![Figure 2: LA and SM distorted patterns (LA: larger, SM: smaller)](image)

![Figure 3: TT and SQ distorted patterns (TT: taller and thinner, SQ: squished)](image)

![Figure 4: SH and DC distorted patterns (SH: shaking, DC: disconnected)](image)

![Figure 5: HL and HS distorted patterns (HL: half-part larger, HS: half-part shifted)](image)

<table>
<thead>
<tr>
<th>Training Patterns</th>
<th>36 LAs and 36 SMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kwan-Cai FNN</td>
<td>87.78%</td>
</tr>
<tr>
<td>Our GFNN</td>
<td>94.44%</td>
</tr>
</tbody>
</table>

**Table 1: Comparison between the Kwan-Cai FNN's Recognition Rate and our GFNN's Recognition Rate for $T_f = 0.52$**

<table>
<thead>
<tr>
<th>Training Patterns</th>
<th>36 TTs and 36 SQs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kwan-Cai FNN</td>
<td>87.78%</td>
</tr>
<tr>
<td>Our GFNN</td>
<td>93.05%</td>
</tr>
</tbody>
</table>

**Table 2: Comparison between the Kwan-Cai FNN's Recognition Rate and our GFNN's Recognition Rate for $T_f = 0.48$**

<table>
<thead>
<tr>
<th>Training Patterns</th>
<th>36 SHs and 36 DCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kwan-Cai FNN</td>
<td>87.78%</td>
</tr>
<tr>
<td>Our GFNN</td>
<td>89.17%</td>
</tr>
</tbody>
</table>

**Table 3: Comparison between the Kwan-Cai FNN's Recognition Rate and our GFNN's Recognition Rate for $T_f = 0.515$**
5 Conclusions

By applying genetic algorithms to the Kwan-Cai fuzzy neural network, we have designed a more adaptive genetic fuzzy neural network for pattern recognition. A genetic-algorithms-based self-organizing learning algorithm is capable of reducing the total number of fuzzy neurons and increasing recognition rates for a fixed number of output neurons. The simulations have indicated that the genetic fuzzy neural network is effective for recognizing various distorted patterns with good recognition rates.

Acknowledgments

The authors would like to thank Dr. Cai for sending us the distorted patterns and Adam for his help.

References


