A CONCURRENT FUZZY NEURAL NETWORK APPROACH FOR A FUZZY GAUSSIAN NEURAL NETWORK

I. F. Iatan

Department of Mathematics and Computer Science, Technical University of Civil Engineering of Bucharest (iuliafi@yahoo.com)

Abstract. The aim of the paper is to introduce a concurrent fuzzy neural network approach, representing a winner-takes-all collection of fuzzy Gaussian modules. Our proposed model will be applied for the pattern classification. The fuzzy neural model consists of a set of M fuzzy neural networks, one for every class, each network having a single output. The output value corresponding to the \( k \), \( k = \frac{1}{M} \), neural network is equal to 1 for those patterns belonging to the class \( k \) and 0 for the others patterns from the training set. After we have trained the \( M \) fuzzy neural networks, we shall save the weights in \( M \) files, in order to be used in the test stage of the respective networks. We have applied this model as a classifier, in a cascade having the following processing stages: the application of a pattern to the input of each of the \( M \) networks; computing the output of the respective network and then taking the maximum of those \( M \) outputs. The results of computer simulation will be given.

Keywords: Concurrent neural network, Fuzzy Gaussian Neural Network, Face recognition, Principal Component Analysis (PCA), Discrete Cosine Transform (DCT).

1. FUZZY GAUSSIAN NEURAL NETWORK (FGNN)

The four-layer structure of the Fuzzy-Gaussian Neural Network (FGNN) is shown in [6]. It is a special type of neural network, by special type of FGNN understanding that it has so special the connections (between the second and third layers) and the operations with the nodes, too.

It represents a modified version of Chen and Teng fuzzy neural network, by transforming the function of approximation into a function of classification. The change affects only the equations of the fourth layer, but the structure diagram is similar.

The FGNN keeps the advantages of the original fuzzy net described by Chen and Teng [3] for identification in control systems:

(a) its structure allows us to construct the fuzzy system rule by rule;
(b) if the prior knowledge of an expert is available, then we can directly add some rule nodes and term nodes;
(c) the number of rules do not increase exponentially with the number of inputs;
(d) elimination of redundant nodes rule by rule.

Its construction is based on fuzzy rules of the form:
∀_j: If x_1 is A_1^j and x_2 is A_2^j ... and x_m is A_m^j, then y_1 is β_1^j, ..., y_M is β_M^j,

where:
- m is the dimension of the input vectors (number of retained features),
- j, j = 1, K is the rule index,
- M is the number of the output neurons (it corresponds to the number of classes),
- X = (x_1, ..., x_m) is the input vector, corresponding to the rule ∀_j,
- A_i^j, i = 1, m are some fuzzy sets corresponding to the input vector,
- Y = (y_1, ..., y_M) is the vector of the real outputs, corresponding to the rule ∀_j,
- β_i^j, i = 1, M are some fuzzy sets corresponding to the output vector.

The j-th fuzzy rule is illustrated in Fig. 1.

![Figure 1. The j-th component of FGNN.](image)

FGNN consists in four layers of neurons and each of its neuron performs two actions using two different functions:

1. the first function is the aggregation function \( g^k( ) \) which computes the net input

\[
\text{Net input} = g^k(x^k; W^k)
\]

where: the superscript indicates the layer number, therefore \( k = 1, 4 \), \( x^k \) is input vector of the \( k \)-th layer, \( W^k \) represents weight vector corresponding to the \( k \)-th layer of FGNN;

2. the second function is the nonlinear activation function, denoted \( f^k( ) \), which gives the output:

\[
\text{Output} = O_i^k = f^k(g^k)
\]

where \( O_i^k \) is the \( i \)-th output of the respective neuron, from the layer \( k \).
As in the case of the other neural networks, the FGNN input layer is a transparent layer, without a role in the data processing; the neurons of the first layer only transmit the information to the next level.

The neurons of the second layer (the linguistic term layer) of the FGNN are membership neurons, resulting by the fuzzification of the first layer neurons. Each neuron of this level performs a Gaussian membership function. The number of neurons characterizing this level is \( m \times K \). Each input is transformed by this layer into a fuzzy membership degree.

The third layer of the FGNN is called the rule layer. The connections between the membership neurons of the second layer and the rule neurons that characterize the third layer of the FGNN indicate the premise of the fuzzy rules. This layer computes the antecedent matching by the product operation [6].

The last layer of the FGNN is the output layer, which contains the output neurons. The conclusion (the consequence) of the rules is evidenced by the connections between the neurons of the third layer and the neurons of the output layer. This level performs the defuzzification of its inputs, providing \( M \) non-fuzzy outputs.

The FGNN parameters have a physical significance, in the meaning that \( m_{ij}, i = 1, m, j = 1, K \) represents the average and \( \sigma_{ij}, i = 1, m, j = 1, K \) is the variance of the membership functions corresponding to some fuzzy sets, \( m \) being the number of the neurons from the input layer of the FGNN and \( K \) representing the number of the fuzzy considered rules.

These parameters of the FGNN one initialize according to an on-line initialization algorithm (see [6]) and they will be refined during the training algorithm.

The training algorithm is of type back-propagation (BP), in order to minimize the error function (like in the case of other fuzzy neural networks as the Neuro Fuzzy Perceptron, the Fuzzy Nonlinear Perceptron based on Alpha Level Sets).

The refining corresponding to the FGNN parameters, which takes place during the training algorithm can be divided into two phases, depending on the parameters of the rule premises and consequences as follows:

a) in the part of the rule premise, one refine the means and the variances of the Gaussian functions;

b) in the rule consequences, the weights of the last layer has to be refined as the others being equal to 1.

The advantage of the FGNN consists in the fact that, for certain values of the overlapping parameters one achieve very good recognition rates of the test lot.

The FGNN disadvantage is that it requires a large number of term neurons (on the second layer), namely \( m \times K \) neurons, for \( m \) inputs and \( K \) fuzzy rules.

The concurrent fuzzy Gaussian neural networks (introduced in the following paragraph) are designed to improve the recognition rates of the test lot, that have been obtained using the simple variant of this network.
2. CONCURRENT FUZZY GAUSSIAN MODULES

Our fuzzy neural model, entitled Concurrent Fuzzy Gaussian Neural Network= CFGNN (see Fig. 3) consists of a set of $M$ fuzzy neural networks, by the type FGNN (Fuzzy Gaussian Neural Network) , one for every class, each network from the Fig. 2 having a single output.

$X = (x_1, \ldots, x_m)$ represents the vector which one applies to the FGNN input, $m$ being the number of the neurons corresponding to the input layer;

The weight between the $(i-1)k + j$ -th neuron of the second layer and the neuron $j$ of the third layer, where $K$ is the number of the neurons from the third layer;

$W_{kj} = 1, K$ is the connection from the neuron $j$ ($j = 1, K$) from the third layer and the neuron from the last layer of the FGNN module;

$y^k_p$ is the output of the $k$-th module, when the input vector has the index $p$, ($k = 1, M$ where $M$ is the number of the neuro- fuzzy modules equal to the number of classes).
The ideal output of the $k$-th FGNN module is equal to 1, for the training pictures belonging to the class $k$ (the number of the pictures from each class being $N_1 = \cdots = N_k = \cdots N_M = N_c$) and is equal to 0, for the rest of the training lot.

After that the $M$ networks have been trained in a number of epochs, the weights have been "frozen" and they have been saved in $M$ files to be used in the test stage of the respective networks.

The test lot comprises $N_t = M \cdot N_c/2$ pictures, that are different from those used in training (every $N_c/2$ pictures in each class).

The classification phase consists of the following steps:

a) one applies the test picture to the input of each of the $M$ modules and computes the corresponding outputs;

b) evaluate the maximum of the $M$ outputs computed at step (a);

c) associate to the input image the class label of the neuro-fuzzy module that leads to the maximum output.

3. EXPERIMENTAL RESULTS

For the task of face recognition, we have used a processing cascade having two stages:

(a) Feature extraction using either the Principal Component Analysis (PCA) or the Discrete Cosine Transform (DCT);

(b) Pattern classification using both FGNN and CFGNN.

For experimenting the presented FKCSN model, we have used "The ORL Database of Faces" provided by the AT&T Laboratories from Cambridge University with 400 images, corresponding to 40 subjects (namely, 10 images for each class).

We have divided the whole gallery into a training lot (200 pictures) and a test lot (200 pictures). Each image has the size of 92 x 112 pixels with 256 levels of grey. For the same subject (class), the images have been taken at different hours, lighting conditions, and facial expressions, with or without glasses. For each class, one shows five images for the training...
and five images for the test lot (see Fig. 4).

Figure 4. The ORL face database.

We have implemented the software corresponding to the presented CFGNN face recognition application using Microsoft Visual C++.

To achieve the selection stage of feature selection with PCA and DCT respectively, the original images (portraits) of size 92 x 112 have been reduced to size 46 x 56. We considered the situation of 40 classes of subjects and 400 images (each of 40 subjects appear in 10 hypostases). From these images have resulted the vectors $X_i, i = 1, 400$, having the size $45 \times 56 = 2576$. The experimental results are given in Table 1, which provides the perfor-
mances of the FGNN and CFGNN systems, to face recognition (grouped in 40 classes), with the stage of feature selection (using PCA / DCT).

Table 1. Comparison between the performances of the FGNN and CFGNN systems experimented to the face recognition \((M = 40)\) with the feature selection

- \(m\) is the feature number selected with PCA / DCT,
- \(L\) denotes the number of the neurons from the hidden layer,
- \(\alpha_i, i = 1, m, 0 < \alpha_i < 1\) are some overlapping factors (see [6]),
- \(R_1\) is the recognition rate (%) for the training lot (obtained at the end of the training),
- \(R_2\) is the recognition rate (%) for the test lot (corresponding to the last epoch of training).

<table>
<thead>
<tr>
<th>The used model</th>
<th>(m)</th>
<th>The used transformation</th>
<th>(\alpha_i)</th>
<th>(R_1)</th>
<th>(R_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGNN</td>
<td>100</td>
<td>PCA</td>
<td>0.999999</td>
<td>99%</td>
<td>77%</td>
</tr>
<tr>
<td>CFGNN</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FGNN</td>
<td>160</td>
<td></td>
<td>0.999999</td>
<td>100%</td>
<td>84%</td>
</tr>
<tr>
<td>CFGNN</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FGNN</td>
<td>100</td>
<td>DCT</td>
<td>0.9999991</td>
<td>100%</td>
<td>91%</td>
</tr>
<tr>
<td>CFGNN</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FGNN</td>
<td>160</td>
<td></td>
<td>0.9999994</td>
<td>100%</td>
<td>91%</td>
</tr>
<tr>
<td>CFGNN</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the Table 1 is evaluated that the best recognition performance (91.50%) is achieved using a cascade of DCT (both for \(m = 100\) and \(m = 160\)) - CFGNN.

We can also noticed that the use of the Concurrent Fuzzy Gaussian Neural Network causes an increase of the recognition rates for the training lot and the test lot (both in case of feature selection using PCA and DCT) compared with the those obtained using the simple variant of FGNN.

3. CONCLUSIONS

The aim of this paper is to introduce a model of a concurrent fuzzy neural network for the Fuzzy Gaussian Neural Network (FGNN). The Concurrent Fuzzy Gaussian Neural Network (CFGNN) will be applied as a classifier, in a cascade having the following processing stages: the application of a pattern to the input of each of the \(M\) networks; computing the output of the respective network and then taking the maximum of those \(M\) outputs.

From the achieved results by computer simulation, we have highlighted the performances of the concurrent model, in comparison with the simple model of the fuzzy Gaussian network.
Acknowledgements

The author would like to thank Prof. dr. Victor Neagoe, from the Faculty of Electronics, Telecommunications and Information Technology, Polytechnic University of Bucharest, Romania for his comments and suggestions.

This work was supported by the strategic grant POSDRU/89/1.5/S/58852, Project "Postdoctoral programme for training scientific researchers" cofinanced by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007-2013.

4. REFERENCES


