ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **8**, August **2016** pp. 1865–1870

A FUZZY LOGISTIC NEURAL NETWORK FOR BINARY CLASSIFICATION

Dandan Chi, Degang Wang and Hongxing Li

School of Control Science and Engineering
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116024, P. R. China
wangdg@dlut.edu.cn

Received December 2015; accepted March 2016

Abstract. *In this paper, a kind of fuzzy logistic neural network (FLNN) is designed to solve the binary classification problem. The consequent of FLNN is chosen as logistic function. Accordingly, a hybrid learning strategy is applied to determining the parameters of FLNN. The constrained gradient descent algorithm is used to determine the center and width of membership function and maximum likelihood estimation method is applied to identifying the parameters of logistic function. Further, Gaussian particle swarm optimization (GPSO) algorithm is introduced to optimize the parameters of these two methods, including the learning rate and the momentum factor. Finally, FLNN is used to handle some UCI datasets. The simulation results demonstrate the validity of the proposed method.*

Keywords: Fuzzy logistic neural network, Gradient descent, GPSO, Maximum likelihood estimation

1. **Introduction.** As is well known, binary classification problem is the most essential classification problem. Many practical problems can be transferred into a binary classification problem, such as medical diagnosis [1], financial analysis [2,3] and weather forecast [4]. Accordingly, how to improve the classification accuracy for the binary classification problem has gained much more attention by scholars.

Many scholars apply various intelligent methods to solving binary classification problem. In [5], a belief rule-based classification system based on belief function theory and fuzzy rule-based classification system is introduced. In [6], a genetic programming model modified by particle swarm optimization is presented to investigate the classification problem. Meanwhile, some scholars utilize different neural networks to handle binary classification problem. In [7], a hybrid algorithm based on the q-Gaussian RBF neural network is established for classification problem. In [8], a Bayes classifier is introduced to handle the imbalanced datasets. In [9], a combination model of neural network and interval neutrosophic set is proposed to be the classifier. In addition, in [10,11], scholars study bankruptcy prediction problems by combining neural network and regression technology.

Although some fuzzy neural networks have successfully been proposed to solve the classification problem, how to improve the model structure and classification accuracy of fuzzy neural network is still an interesting question. In this paper, we propose a kind of fuzzy logistic neural network (FLNN) by utilizing logistic function as the consequent. Different from the fuzzy neural networks with polynomial consequents, the use of logistic function in this paper can enhance the interpretability of classification problem. Since FLNN can be seen as a nonlinear combination of several logistic models, it can achieve higher classification accuracy than single logistic model. Further, a hybrid learning strategy is proposed for FLNN. The basic idea is that the constrained gradient descent algorithm and maximum likelihood estimation method are used to learn the parameters of FLNN and Gaussian particle swarm optimization (GPSO) algorithm is introduced to optimize the parameters of these two learning methods. The constrained gradient descent algorithm can ensure the parameters converge to the specified range and the parameter optimization process by GPSO algorithm avoids the network falling into local optimum to a certain extent. Numerical simulations show that the proposed method can effectively improve the accuracy of FLNN. This paper is organized as follows. In Section 2, the structure and mathematical representation of FLNN are introduced. In Section 3, the hybrid learning method for FLNN is investigated. In Section 4, some numerical examples are provided to demonstrate the validity of the proposed method. Section 5 provides conclusions and addresses the future work.

2. **Basic Structure of Fuzzy Logistic Neural Network.** In this section, we will introduce the basic structure and mathematical representation of the proposed FLNN.

Suppose that $(\mathbf{x}_l, y_l) \in \mathbb{R}^n \times \mathbb{R}$ $(l = 1, \ldots, N)$ are a group of training samples, where $\mathbf{x}_l^T = [x_1^l, \ldots, x_n^l]$ is the input variable, and y_l is the output variable. The structure of FLNN is shown in Figure 1. The function of each layer is as follows.

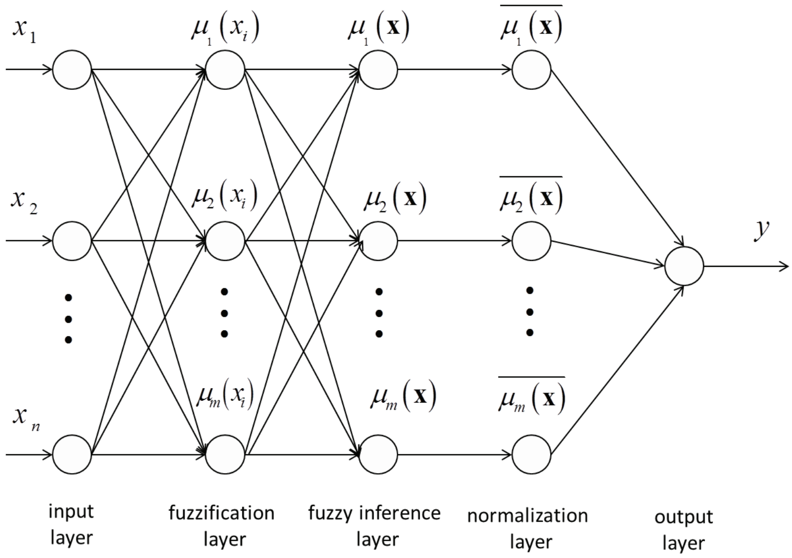

Figure 1. Fuzzy logistic neural network structure chart

The first layer is the input layer. Each node of this layer represents different attribute value of the input sample x_i $(i = 1, \ldots, n)$. The second layer is the fuzzification layer. Each node is the membership function of the input variable. In this paper, we choose the Gaussian function as the membership function, i.e., $\mu_j(x_i) = \exp\left(-\left(\frac{x_i - c_{ij}}{b_{ij}}\right)^2\right)$, where c_{ij} is the center, b_{ij} is the corresponding width, $i = 1, \ldots, n$, $j = 1, \ldots, m$. The third layer is the fuzzy inference layer. Each node of this layer is the product of the memberships of the second layer, i.e., $\mu_j(\mathbf{x}) = \prod_{i=1}^n \mu_j(x_i)$ $(j = 1, \ldots, m)$, where m is the number of the fuzzy rules. The fourth layer is the normalization layer. The function of this layer is to compute the normalization results of the output values in the third layer. The value of each node in the fourth layer is taken as $\overline{\mu_j(\mathbf{x})} = \frac{\mu_j(\mathbf{x})}{\sum_{j=1}^m \mu_j(\mathbf{x})}$, $j = 1, \ldots, m$. The fifth layer is the output layer. The output of FLNN can be expressed as $y = \sum_{j=1}^m \overline{\mu_j(\mathbf{x})} \cdot F_j(\mathbf{x}'\boldsymbol{\beta})$, where $F_j(\mathbf{x}'\boldsymbol{\beta}) = \frac{1}{1+\exp(-(\beta_{0j}+\beta_{1j}\cdot x_1+\ldots+\beta_{nj}\cdot x_n))}$ is chosen as the activation function of the neural network consequent.

From the construction of FLNN, we can find that the consequent of the neural network is chosen as logistic function $F_j(\mathbf{x}'\boldsymbol{\beta})$. That is to say that the output of the neural network can be seen as the weighted sum of several logistic functions.

3. **Fuzzy Logistic Neural Network Algorithm.** In this section, we will introduce the identification algorithm for FLNN. Parameters to be identified include the antecedent parameters c_{ij} and b_{ij}, and the consequent parameters β_{ij}, where $i = 0, \ldots, n$; $j = 1, \ldots, m$.

The main phases of the proposed FLNN are shown as follows. Firstly, the antecedent parameters and consequent parameters are respectively determined by the constrained gradient descent algorithm and maximum likelihood estimation method. Then, let the prediction accuracy of the testing samples be the fitness value of the objective function in GPSO algorithm. We adjust the learning rate and the momentum factor dynamically through iterative updating, in order to promote the prediction accuracy of FLNN.

In the following, we will introduce the detailed phases of the proposed method. Firstly, we give the identification algorithm of the antecedent parameters. The parameters of antecedent membership function are initialized by K-means clustering algorithm. Then, based on the initial values, we use a hybrid gradient descent algorithm combining the constrained gradient descent algorithm [12] and Gaussian particle swarm optimization (GPSO) algorithm [13] to obtain the values of c_{ij} and b_{ij}. The error cost function is defined as $E = \frac{1}{2}\left(y_d - \hat{y}_d\right)^2$ where y_d is the desired output and $\hat{y}_d$ is the actual output. Using the constrained gradient descent method [12], we have that

$$
\begin{aligned}
c_{ij}(k) &= c_{ij}(k-1) - \lambda \tfrac{\partial E}{\partial c_{ij}} + \alpha(c_{ij}(k) - c_{ij}(k-1)) + \Delta c_{ij}(k), \\
b_{ij}(k) &= b_{ij}(k-1) - \lambda \tfrac{\partial E}{\partial b_{ij}} + \alpha(b_{ij}(k) - b_{ij}(k-1)) + \Delta b_{ij}(k),
\end{aligned}
$$

where λ is the learning rate, α is the momentum factor, and

$$
\Delta c_{ij}(k) = \begin{cases} c_2 - c_{ij}(k-1), & c_{ij}(k-1) > c_2, \\ E(k)\Delta c_{ij}(k-1), & c_1 \le c_{ij}(k-1) \le c_2, \\ c_1 - c_{ij}(k-1), & c_{ij}(k-1) < c_1. \end{cases}
$$

$$
\Delta b_{ij}(k) = \begin{cases} b_2 - b_{ij}(k-1), & b_{ij}(k-1) > b_2, \\ E(k)\Delta b_{ij}(k-1), & b_1 \le b_{ij}(k-1) \le b_2, \\ b_1 - b_{ij}(k-1), & b_{ij}(k-1) < b_1. \end{cases}
$$

Compared to the traditional gradient descent algorithm, the constraints added to the gradient optimization process can avoid search divergence. In addition, it should be noted that the selection of learning rate and momentum factor has great influence on the convergence precision and speed of the network. To solve this problem, in [14,15] some authors apply different methods to optimize the learning rate and the momentum factor. Based on these ideas, in this paper, we set the learning rate and the momentum factor of the constrained gradient descent algorithm as particles to be optimized, and apply GPSO algorithm [13] to tune them. The classification accuracy of the network is the fitness value of GPSO. Then we implement the parameter optimization by updating the positions of the particles as follows:

$$
V_{id}^{k+1} = \omega V_{id}^k + r_1 \cdot G\left(\mu, \sigma^2\right)\left(P_{id}^k - X_{id}^k\right) + r_2 \cdot G\left(\mu, \sigma^2\right)\left(P_{gd}^k - X_{id}^k\right), \ X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1},
$$

where ω is the inertia weight, r_1, r_2 are the acceleration factors, $G\left(\mu, \sigma^2\right)$ is the Gaussian random variable, V_{id} is the velocity of the particles, X_{id} is the position of the particles, P_{id} is the individual optima, and P_{gd} is the global optima. In this way, we can use hybrid learning strategy to obtain the parameters of the membership function. Compared to the constrained gradient descent algorithm, the proposed hybrid method can effectively improve the performance of FLNN.

Next, we will investigate how to identify the consequent parameters of FLNN. In this paper, we apply the maximum likelihood estimation method to compute the consequent parameters β_{ij} ($i = 0, \ldots, n$; $j = 1, \ldots, m$). Let $Y_l \in \{0, 1\}$ ($l = 1, \ldots, N$) be the final output of the samples, then the probabilities of $Y_l = 1$ and $Y_l = 0$ are given as follows:

$$p(Y_l = 1) = \frac{\sum_{j=1}^{m} \mu_j(\mathbf{x}) \cdot F_j(\mathbf{x}'\boldsymbol{\beta})}{\sum_{j=1}^{m} \mu_j(\mathbf{x})}, \quad p(Y_l = 0) = 1 - \frac{\sum_{j=1}^{m} \mu_j(\mathbf{x}) \cdot F_j(\mathbf{x}'\boldsymbol{\beta})}{\sum_{j=1}^{m} \mu_j(\mathbf{x})}.$$

Accordingly, the log-likelihood function can be expressed as:

$$\ln L = \sum_{l=1}^{N} \left\{ Y_l \cdot \ln \left[\frac{\sum_{j=1}^{m} \mu_j(\mathbf{x}) \cdot F_j(\mathbf{x}'\boldsymbol{\beta})}{\sum_{j=1}^{m} \mu_j(\mathbf{x})} \right] + (1 - Y_l) \cdot \ln \left[1 - \frac{\sum_{j=1}^{m} \mu_j(\mathbf{x}) \cdot F_j(\mathbf{x}'\boldsymbol{\beta})}{\sum_{j=1}^{m} \mu_j(\mathbf{x})} \right] \right\}.$$

To obtain the maximum of the log-likelihood function, we use the corresponding parameter-adjusting method as follows:

$$\beta_{ij}(k) = \beta_{ij}(k-1) - \lambda' \cdot \frac{\partial \ln L}{\partial \beta_{ij}} + \alpha' \cdot (\beta_{ij}(k) - \beta_{ij}(k-1)), \quad i = 0, \ldots, n; \ j = 1, \ldots, m$$

where λ' is the learning rate, and α' is the momentum factor. Similarly, we also use GPSO algorithm to determine the λ' and α'. Accordingly, using the hybrid maximum likelihood estimation method, we can obtain the consequent parameters.

Finally, we summarize the main steps of the identification method for FLNN.

Step 1. Initialize the parameters, including the number of fuzzy rules, the center, the width, the learning rate and the momentum factor, the number of population and the maximum number of iteration.

Step 2. Compute c_{ij} and b_{ij} by constrained gradient descent algorithm.

Step 3. Identify parameters β_{ij} by maximum likelihood estimation method.

Step 4. Compute the output of FLNN.

Step 5. Use GPSO to optimize the learning rate and the momentum factor.

Step 6. If the number of iteration is less than the maximum number, then go to Step 2; else stop.

4. Numerical Simulations.

In order to verify the performance of the proposed method, we use FLNN to classify six UCI datasets, i.e., Mammographic Mass, Pima Indian Diabetes, Wisconsin Breast Cancer, Bupa Liver Disorders, Heart-Statlog and Australian. The descriptions of these datasets are shown in Table 1.

TABLE 1. Basic information of UCI datasets

Dataset	*Features*	*Classes*	*Instances*
Mammographic Mass (Mammo)	*5*	*2*	*961*
Pima Indian Diabetes (PID)	*8*	*2*	*768*
Wisconsin Breast Cancer (WBC)	*9*	*2*	*683*
Bupa Liver Disorders (BLD)	*6*	*2*	*345*
Heart-Statlog (HS)	*13*	*2*	*270*
Australian (Aus)	*14*	*2*	*690*

Firstly, we use Mammo, PID and WBC datasets to compare the classification accuracy with other algorithms, including fuzzy RBF neural network (FNN), logistic regression (LR), mfCLL ([16]) and TAN-fCLL ([16]). In the numerical simulation, we use 5-fold cross-validation. For each dataset, we consider the average result of the five partitions. The parameters of FLNN are set as follows. The number of fuzzy rules is taken as $m = 4$. The numbers of iterations for constrained gradient descent algorithm and maximum likelihood estimation method are respectively taken as 100. The population size of GPSO is taken as 10. The number of iteration for GPSO is taken as 10. The simulation results are shown in Table 2. The accuracy is annotated with the standard deviation, which

Table 2. Simulation results about Mammo, PID and WBC

Dataset	*FLNN*	*FNN*	*LR*	*mfCLL ([16])*	*TAN-fCLL ([16])*
Mammo	***84.22***	*83.25*	*82.89*	*83.86*	*82.89*
	$\pm$***1.18***	$\pm$*1.20*	$\pm$*1.21*	$\pm$*1.19*	$\pm$*1.21*
PID	***78.65***	*76.16*	*77.46*	*78.39*	*78.52*
	$\pm$***1.48***	$\pm$*1.54*	$\pm$*1.51*	$\pm$*1.49*	$\pm$*1.48*
WBC	***97.66***	*96.64*	*96.93*	***97.66***	***97.66***
	$\pm$***0.58***	$\pm$*0.66*	$\pm$*0.66*	$\pm$***0.58***	$\pm$***0.58***

Table 3. Simulation results about BLD, HS and Aus

Dataset	*FLNN*	*PMC ([17])*	*DGC + ([18])*	*ε-PSVM ([19])*
BLD	***74.74***	*67.25*	*67.44*	*69.33*
HS	***85.93***	*83.70*	*84.52*	*85.07*
Aus	***87.54***	*86.96*	*83.74*	*86.52*

is computed according to the binomial formula $\sqrt{acc \times (1 - acc)/N}$, where acc is the classification accuracy and N is the size of the dataset.

Then we use BLD, HS and Aus to compare the classification accuracy with other algorithms, including PMC ([17]), DGC + ([18]) and ε-PSVM ([19]). In the numerical simulation, we use 10-fold cross-validation. For each dataset, we consider the average result of the ten partitions. The number of fuzzy rules is taken as $m = 3$. The numbers of iterations for constrained gradient descent algorithm and maximum likelihood estimation method are respectively taken as 100. The population size of GPSO is taken as 20. The number of iteration for GPSO is taken as 10. The simulation results are shown in Table 3.

From Table 2 and Table 3, we can find that the proposed FLNN can achieve higher accuracy. These facts mean that the proposed FLNN is suitable for solving the binary classification problem.

5. **Conclusions.** In this paper, FLNN is presented to deal with binary classification problem. Gaussian function is chosen as the membership function and logistic function is used to describe the consequent of the neural network. The proposed FLNN can enrich the structure of fuzzy neural network. And the numerical experiments show that hybrid learning strategy can make FLNN achieve higher classification accuracy. In the future research, we will apply FLNN to solving multi-classification problem.

Acknowledgment. This work is supported by the Fundamental Research Funds for the Central Universities (DUT14QY30), the National Natural Science Foundation of China (61374118) and the Special Fund for NHFPC Scientific Research in the Public Welfare (201502023).

REFERENCES

[1] A. Castano, F. N. Francisco and P. A. Gutierrez, Permanent disability classification by combining evolutionary generalized radial basis function and logistic regression methods, *Expert Systems with Applications*, vol.39, pp.8350-8355, 2012.

[2] Y. Jin, S. Gai, P. Liu and X. Tang, Contourlet-based feature extraction for bank note classification using GMM, *ICIC Express Letters*, vol.5, no.4(B), pp.1495-1500, 2011.

[3] S. Gai, G. Yang and S. Zhang, Banknote classification by using contourlet and quaternion wavelet features, *ICIC Express Letters*, vol.6, no.9, pp.2295-2300, 2012.

[4] H. Zhu, X. Zhao, P. Liu, J. Liu and X. Tang, An autocorrelation function-based classification of different weather conditions, *ICIC Express Letters*, vol.5, no.11, pp.4195-4201, 2011.

[5] L. M. Jiao, Q. Pan and T. Denoeux, Belief rule-based classification system: Extension of FRBCS in belief functions framework, *Information Sciences*, vol.309, pp.26-49, 2015.
[6] J. Hajira and A. R. Baig, GPSO: A framework for optimization of genetic programming classifier expressions for binary classification using particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.8, no.1(A), pp.233-242, 2012.
[7] F. N. Francisco and H. M. Ce'sar, Parameter estimation of q-Gaussian radial basis functions neural networks with a hybrid algorithm for binary classification, *Neurocomputing*, vol.75, pp.123-134, 2012.
[8] A. Adam, M. I. Shapiai, Z. Ibrahim, M. Khalid and L. W. Jau, Development of a hybrid artificial neural network – Naive Bayes classifier for binary classification problem of imbalanced datasets, *ICIC Express Letters*, vol.5, no.9(A), pp.3171-3175, 2011.
[9] P. Kraipeerapun and C. C. Fung, Binary classification using ensemble neural networks and interval neutrosophic sets, *Neurocomputing*, vol.72, pp.2845-2856, 2009.
[10] C. B. Cheng, C. L. Chen and C. J. Fu, Financial distress prediction by a radial basis function network with logit analysis learning, *Computers and Mathematics with Applications*, vol.51, pp.579-588, 2006.
[11] P. A. Gutierrez, M. J. Segovia-Vargas and S. Salcedo-Sanz, Hybridizing logistic regression with product unit and RBF networks for accurate detection and prediction of banking crises, *Omega*, vol.38, pp.333-344, 2010.
[12] K. Qian, T. Z. Wang and B. Ma, A new model of local connection BP network and its application, *Systems Science and Mathematics*, vol.34, no.7, pp.792-804, 2014.
[13] H. Melo and J. Watada, Gaussian-PSO with fuzzy reasoning based on structural learning for training a neural network, *Neurocomputing*, vol.172, pp.405-412, 2016.
[14] P. Tahmasebi and A. Hezarkhani, A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation, *Computers & Geosciences*, vol.42, pp.18-27, 2012.
[15] S. K. Oh, W. D. Kim and W. Pedrycz, Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization, *Fuzzy Sets and Systems*, vol.163, pp.54-77, 2011.
[16] A. M. Carvalho and P. Adão, Hybrid learning of Bayesian multinets for binary classification, *Pattern Recognition*, vol.47, pp.3438-3450, 2014.
[17] N. K. Sreeja and A. Sankar, Pattern matching based classification using ant colony optimization based feature selection, *Applied Soft Computing*, vol.31, pp.91-102, 2015.
[18] A. Cano, A. Zafra and S. Ventura, Weighted data gravitation classification for standard and imbalanced data, *IEEE Trans. Cybernetics*, vol.43, no.6, 2013.
[19] G. Y. Zhu, D. Huang and P. Zhang, ε-proximal support vector machine for binary classification and its application in vehicle recognition, *Neurocomputing*, vol.161, pp.260-266, 2015.