International Journal of Computational Intelligence Research.
ISSN 0973-1873 Vol.2, No. 1 (2006), pp. 72-75


# A Classification Approach Based on Evolutionary Neural Networks*

**Jing Zhong [1,2], Yan Fu [2] and Jun-lin Zhou [2]**

[1] Department of Mathematics and Computer Science, Chongqing Three Gorges University,
Shalong Road 780, Wanzhou, Chongqing 404000, China
*zhongandy@sohu.com*

[2]Department of Computer Science and Engineering, University of Electronic Science and Technology of China,
Jianshe North Road 4, Chendu, 610054, China
*{fuyan, jlzhou}@ uestc.edu.cn*

***Abstract*: Classification is important in data mining and machine learning. In this paper, a classification approach based on evolutionary neural networks (CABEN) is presented, which establishes classifiers by a group of three-layer feed-forward neural networks. The neural networks are trained by an improving algorithm synthesizing modified Evolutionary Strategy and Levenberg-Marquardt optimization method. The class label of the identifying data can first be evaluated by each neural network, and the final classification result is obtained according to the absolute-majority-voting rule. Experimental results show that the algorithm is effective for the classification, and has the better performance in classification precision, comparing with Bayesian and decision trees, especially for the complex classification problems with many classes.**

***Keywords***: classification, evolutionary neural networks, Levenberg-Marquardt, absolute-majority-voting**.**

## I. Introduction

In the past decades, data are being collected and accumulated at a dramatic pace. Therefore, there is an urgent need for a new generation of computational techniques and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of data [1]. This arouses many researchers to study into the area of data mining. One of the data mining functionalities, which plays an important role in business decision-making tasks, is classification. Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [2]. A variety of techniques have been applied to deal with the classification problems, such as neural networks, decision trees, and statistical methods. However, many previous research works show that neural network classifiers have a better performance, lower classification error rate, and more robust to noise than the other two methods mentioned above. The proposed classification based on evolutionary neural networks described in this paper employs modified Evolutionary Strategy and Levenberg-Marquardt optimization method. The performance of the proposed network is evaluated against Bayesian and decision trees.

This paper is organized as follows. Following this introduction, section 2 presents the architecture of the evolutionary neural networks. The learning algorithm is described in section 3. In section 4, the experimental results are demonstrated and discussed. Finally, section 5 is the conclusions.

## II. The Proposed Model

The architecture of the evolutionary neural networks classifier is a group of three-layer feed-forward neural networks as shown in Figure 1. Firstly abstract the input data set and the label set according to the problem. Then construct the model during training the networks by the improved method. Lastly input the identifying data in the trained model, evaluated by each neural network, and identify the final classification according to the absolute-majority-voting rule.

The characteristic of the proposed model is that the neural networks are trained by an improving algorithm synthesizing modified Evolutionary Strategy and Levenberg-Marquardt optimization method. CABEN can improve the whole performance, because the evolutionary strategy has the ability of fine global search, and LM method has the advantage of

* Supported by the Natural Science Foundation under grant NO. 10476006

short time in learning.

## III. The Learning Algorithm

In CABEN, two phases of learning are involved: Phase1 (Train the networks to establish the model) and Phase 2 (Classify the identifying data ).

### *A. Train the Neural Networks Model*

In this phase, CABEN employs the Evolutionary Strategy to train the neural networks, which include individual presentation, initial population, mutation, fitness evaluation.

*1) Individual Presentation*

Each neural network is a three-layer feed-forward neural network, where *r*, *s*, *h* denote the number of input, output and hidden nodes, $w_{ij}^1 (i=1,\ldots,r,\ j=1,\ldots,h)$ are the weights of input to hidden, $w_{ij}^2 (i=1,\ldots,h,\ j=1,\ldots,s)$ are the weights of hidden to output, $\alpha_i (i=1,\ldots,h)$ , $\beta_i (i=1,\ldots,s)$ denote the thresholds of hidden and output nodes. The individual is presented by the ordered vectors of all weights and thresholds of each neural network. Multi-neural networks have the same structures, so they have the same number nodes.

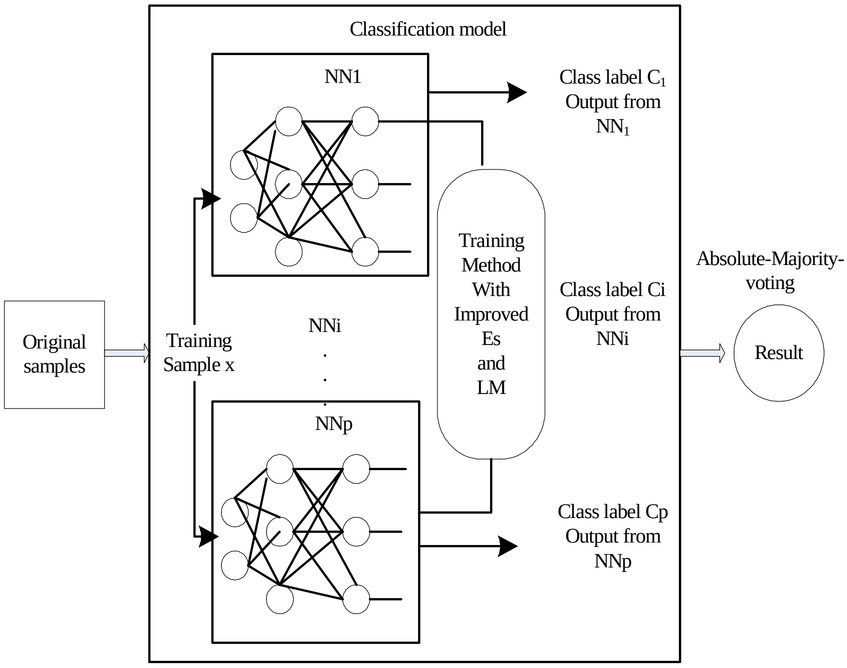


**Figure 1.** Framework of CABEN

*1) Initial Population*

Randomly produce P initial father samples ( $X_1^1, X_2^1, \ldots, X_P^1$ ), where P is the number of the neural networks.

*2) Mutation*

Mutation operator is the main way to form the newly generation. To any $X_i^k = (x_i^k, \sigma_i^k)$ , the newly offspring is $(X_i^k)'$, where *k* is the evolution generation.

$x_{ij}^k (j=1,2,\cdots L)$ , the component of $x_i^k$ in $X_i^k$ forms the newly individual $(x_{ij}^l)'$ by mutation as the follows:

$$(x_{ij}^k)' = x_{ij}^k + \sigma_i^k(j)\cdot N(0,1) \quad (1)$$

where L is the length of $\mathbf{x}_{ij}^k$ .

$\sigma_i^k(j)(j=1,2,\ldots,L)$ , the component of $\sigma_i^k$ forms the newly individual $(\sigma_i^k)'$ by mutation as the follows:

$$(\sigma_i^k(j))' = (B + A\cdot(f(X_i^k))^\alpha)\cdot\sigma_0(j) \quad (2)$$

where $i=1,\cdots P,\ j=1,\cdots L, k=1,\cdots k_{\max}$ , $B=0.5\sim1$, $\alpha=1 or 2$, $f(X_i^k)$ is the $i^{th}$ father sample's fitness, $A=C/\min\{(f(X_i^k))^\alpha\}$, $C=0.5\sim2$, $\sigma_0(j)$ is the $j^{th}$ component of $\sigma_0$ , here $\sigma_0(j)=1,\quad j=1,2,\cdots L$ .

*3) Fitness Evaluation*

In this evolution strategy, direct suppose each individual fitness the objective function, which is the problem to be optimized. The fitness function used in this research is as follows:

$$\left.\begin{aligned} f(X_k^i) &= 1/(E(i)+1) \\ E(i) &= \left\|Y-\overline{Y}\right\|_2 \end{aligned}\right\} \quad (3)$$

Where $X_k^i$ is the $i^{th}$ father sample of the $k^{th}$ generation, Y is the expect output, $\overline{Y}$ is the fact output, $E$(i) is the error's $L_2$ norm. The individual is much better while the fitness is larger.

After define the training, a description of the algorithm 1 used in CABEN is given below:

*Algorithm 1: CABEN training*

Input: the training data sets.

Output: the P neural networks model, (P>1).

Step 1: Initial parameters: Define the size (P) of the population, the maximum of evolution generation ($k_{max}$), the first value of the root mean squared vector ($\sigma_0$).

Step 2: Randomly form P initial father samples ($X_1^k, X_2^K, \ldots, X_P^k$), calculate each fitness $f(X_i^k)$ using Equation (3).

Step 3: Mutate any father sample $X_i^k$ by using the formula (1), mutate $\sigma_i^k(j)$ by using the formula (2).

Step 4: Calculate P fitness values of offspring $f((X_i^k)')$ using Equation (3).

Step 5: Selection operator employs the evolution strategy $(\mu+\lambda)-ES$, being $\mu = P, \lambda = P$, while the individual set is: $select(\max_{i=1,\ldots P}(f(X_i^k)'))$.

Step 6: $k$+1, if $k$>$k_{max}$, then skip to step 7, otherwise back to step 3.

Step 7: Construct P neural networks with the same structure according to $X_1^k, X_2^K, \ldots, X_P^k$. The weights and thresholds of $i^{th}$ neural network are: the weights of input to hidden are $x_{ij}^k(j=1,\ldots r*h)$, the weights of hidden to output are $x_{ij}^k(j=r*h+1,\ldots,r*h+h*s)$, the thresholds of hidden nodes are $x_{ij}^k(j=r*h+h*s+1,\ldots,r*h+h*s+h)$, the thresholds of output nodes are $x_{ij}^k(j=r*h+h*s+h+1, \ldots, r*h+h*s+h+s)$.

Step 8: Setup the parameters of LM, the maximum training number ($t_{max}$), prediction error (*err*).

Step 9: Continue to train the networks by LM, if a predetermined number of iteration is reached or the end condition is satisfied, stop the algorithm.

### B. Classify

After established the neural networks model in Phase 1, in this phase we classify the identifying data according to the algorithm 2.

*Algorithm 2: Classify Process*

Input: the identifying data $x$.

Output: the class label C which belongs to $x$.

Step 1: Input the identifying data into the trained model $NN_1, NN_2, \ldots, NN_p$

Step 2: for $i$=1 to P

Calculate the $i^{th}$ neural network output, which is each label's output $c_{i1}, c_{i2}, \ldots c_{it}$ ($t$ is the class number) ;

Identify the classify result of the $i^{th}$ neural network $C_i = \max(c_{i1}, c_{i2}, \ldots c_{it})$.

Step 3: Identify $x$'s class label according to the absolute-majority-voting rule in each classification result $C_i$.

## IV. Experimental Results

To test the performance of the proposed approach, the experiments have been conducted on 5 benchmark data sets from UCI Machine Learning Database[9] as shown in Table 1.

| | Data set | Samples number | Attributes number | Classes number |
|---|---|---|---|---|
| 1 | Cleveland | 296 | 13 | 2 |
| 2 | Glass | 214 | 9 | 6 |
| 3 | Iris | 150 | 4 | 3 |
| 4 | Lymphography | 148 | 18 | 4 |
| 5 | Vehicle | 846 | 18 | 4 |

*Table 1.* Description of experimental data sets.

Employ the method (5 fold cross validation) to the experiment on the data sets above for 5 times. The experimental results denote the classification precision. Calculate the averages of the 5 experiments, then compare to Bayesian classifiers, Bayesian networks, C4.5, the results are shown in Table 2.

The parameters setup in experiment are: P=6, $k_{max}$=20,• =1,B=0.5,C=0.5, the hidden nodes respectively are 5,8,5,7,7 in the data sets Cleveland, Glass, Iris, Lymphography and Vehicle. The last column in Table 2 denotes the average difference between CABEN and the former 3 methods, it is:

$$((|CABEN|-|NB|)+(|CABEN|-|BN|)+(|CABEN|-|C4.5|))/3$$

where || represents the test set's precision to the corresponding method.

| | NB | BN | C4.5 | CABEN | Average difference |
|---|---|---|---|---|---|
| 1 | 82.76 | 81.39 | 73.31 | 84.16 | 4.99 |
| 2 | 69.66 | 55.57 | 69.62 | 73.09 | 9.27 |
| 3 | 93.33 | 94.00 | 94.00 | 95.60 | 1.62 |
| 4 | 79.72 | 75.03 | 77.03 | 86.49 | 9.23 |
| 5 | 58.28 | 61.00 | 69.74 | 86.10 | 23.09 |

*Table 2.* Comparison with Bayesian and decision trees methods (testing precision:%)

It is apparent, for the data sets Glass, Lymphography and Vehicle, CABEN's average difference outperforms them by a wide margin, which indicates CABEN has the best classification precision among the compared methods. Meanwhile we observe the classes number are 4 or 6, indicating the complex classification problem with many classes, otherwise, CABEN has the better performance comparing with Bayesian and decision trees.

## V. Conclusions

In this paper, the classification based on evolutionary neural networks, which applies the modified Evolutionary Strategy and Levenberg-Marquardt optimization method to the artificial neural network, is proposed and its performances are compared with Bayesian and decision trees. The results of the proposed evolutionary neural networks are the best among those of the compared methods.

## Acknowledgment

This work are supported by the Natural Science Foundation under grant NO. 10476006. The authors would also like to thank the anonymous referees for their helpful suggestions on improving the presentation.

## References

[1] Fayyad, U., Piatetsky-Shapiro, G., Smyth P. "Knowledge Discovery and Data Mining:Towards a Unifying Framework". In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996.

[2] Han, J., Kamber, M. *Data Mining: Concepts and Techniques*, Academic Press, 2001.

[3] Guo CH, Tang HW. "Global convergence properties of evolution strategies". *Mathematica Numerica Sinica*, 23(1),pp. 105-110, 2001.

[4] Arulampalam G, Bouzerdoum A. "A generalized feedforward neural network architecture for classification and regression". *Neural Networks*, 16(5-6),pp.561-568,2003.

[5] Ajith A. "Meta learning evolutionary artificial neural networks". *Neurocomputing*, ,56c,pp.1-38,2004.

[6] Buhrman H, de Wolf R. "Complexity measures and decision tree complexity: A survey". *Theoretical Computer Science*,288(1),pp.2-43, 2002.

[7] Ajith A. "Meta learning evolutionary artificial neural networks". *Neurocomputing*, 56c, pp. 1-38, 2004.

[8] Plagianakos VP, Vrahatis MN. "Parallel evolutionary training algorithms for hardware-friendly neural networks". *Natural Computing*, 1(2-3), pp.307-322, 2002.

[9] Blake CL, Merz CJ. "UCI repository of machine learning databases".*http://www.ics.uci.edu/~mlearn.MLRepository.html.* 1998.

## Author Biographies

**Jing Zhong** was born in Lixian, Sichuan Province, China, 1976. She is a master student at the Software Department, University of Electronic Science and Technology, Chendu, China. Her research interests mainly include: Neural Networks, Evolutionary algorithm.

**Yan Fu** was born in Chendu, Sichuan Province, China, 1962. She got the master degree of Computer Software and Theory at School of Computer Science and Engineering in UESTC in 1988. Now, she is a professor in UESTC and the research interests are in the areas of Artificial Intelligence, Data Mining.

**Jun-lin Zhou** was born in Xinyang, Henan Province, China, 1981. He got the master degree of Computer Software and Theory at School of Computer Science and Engineering in UESTC in 2005. Now, he is a PhD student in UESTC. His research interests are in Neural Networks, Self Organization Map.