

Parameter Optimization of Small Set Genetic Algorithm Multilayer Perceptron

You Zhining and Pu Yunming

*School of Computer Technology, Jimei University, Xiamen 361021, China.
Corresponding author: PU Yun-Ming, E-mail: yunmingpu@163.com*

Abstract

The SSGAMLP (Small Set Genetic Algorithm Multilayer Perceptron) model helps individual evolution by group evolution. With respect to the MLP, it has better generalization, it can get unknown feature expressions of more possibilities. The model still exist many problems need to be solved. The number of nodes in the hidden layers and the population size of MLP has a great influence on the performance of SSGAMLP. So this paper focuses on the optimization of that two parameters on SSGAMLP. In this paper, the models of several different experiments are designed. By comparing the experimental data, the relationship between the parameter selection and the model performance is obtained.

Keywords: *genetic evolutionary algorithm; multilayer perceptron; Parameter Optimization*

1. Introduction

In this paper, the SSGAMLP (Small Set Genetic Algorithm Multilayer Perceptron) model [1] is an improved genetic algorithm with selection of the MLP (Multilayer Perceptron) model as the genetic individual. The main point of this model is that reduce the size of individual training sample set S_i . S_{all} is the entire training data set. Each S_i is $1/N$ of S_{all} . One of Its' benefits is that the model complexity is reduced to N times. At the same time also brought a greater benefit is to improve the generalization of the model, because of the random selection of training sample subset S_i which is equal to the introduction of the random factor. The random subset S_i conducive to noise and improve the generalization of the model. Meanwhile, whether the nodes in the hidden layer of the MLP of individual SSGAMLP will affect the performance of the model? Whether the size of the population will affect the performance of the model? In this paper, we hope to design several groups of experimental programs of different numbers of hidden layer nodes and different sizes of population. By comparing the experimental data, we hope to obtain the relationship between parameter selection and model performance. Experiments using 0-9 handwritten digital MNIST data set and character recognition is with one of the challenging problems in pattern recognition, the practical applications include the bank check recognition and postal address recognition. Due to the style of writing, the construction of an effective character recognition system is still a challenging task. Character recognition includes three steps: preprocessing, feature extraction and classification. Using the accepted MNIST data set, the experimental results are credible. Neural evolution may have different implementations [2], such as co-evolutionary algorithm [3]. Evolutionary algorithms are usually used to reduce the mean squared error between the true value and expected output and [4]. Evolutionary computation can more effectively solve the inherent problems of neural networks, such as not according to the problem of structure adjustment and the problem of local optimal [5-10]. In character recognition, evolutionary computing is demonstrated for the neural network learning

[11-13], parameter optimization [14], structure optimization [15-16] is effective. An SSGAMLP model is one kind of neural evolution, SSGAMLP model is mainly to optimize the connection weights.

2. Algorithm Composition

The SSGAMLP model is composed of the MLP part and the GA algorithm. The superiority in the model is reflected in that the number and order for the training sample set of MLP can be changed from the GA algorithm. It is equivalent to the random factors, which can improve the generalization ability. The training of each individual MLP uses a random sample subset S_i , which is the biggest improvement to the classic GA algorithm in the SSGAMLP model. Fitness is to measure the degree of a species to adapt to living environment. On the survival environment, the species of high fitness will get more chances. And those species with low fitness to adapt to the living environment will get less chance. Fitness function is obtained by according to the error rates of the validation set S_{valid} . It has the advantage of better generalization. Crossover operation is to feature as a unit of exchange. It defined for each node all down connection weights and threshold for a feature, which conform to the feature extraction. According to this, genetic algorithms (GA) can choose the better feature representation.

One hidden layer of MLP and three hidden layers of MLP can be expressed in the following figure:

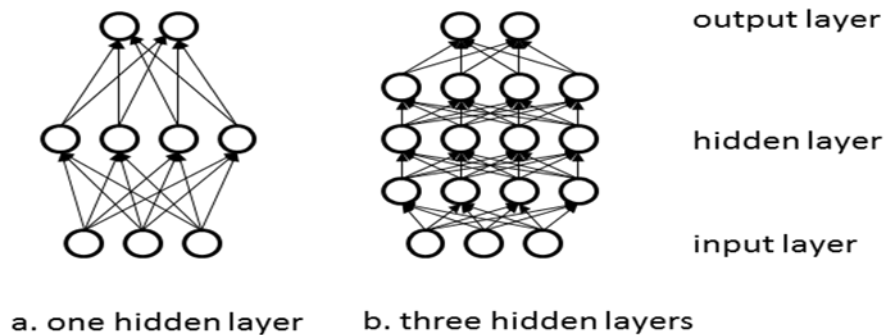


Figure 1. Multi Layer MLP Structure

Formulation of a hidden layer MLP can be expressed as follows: $f: R^D \rightarrow R^L$. Among them, D is the length of the input vector x , L is the length of the output vector $f(x)$. In this way, it can be expressed as follows:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x)))$$

Including: $b^{(1)}$ is the offset vector of the first hidden layer. $b^{(2)}$ is the offset vector of the output layer. $W^{(1)}$ is the weight vector of the first hidden layer. $W^{(2)}$ is the weight vector of the output layer. G and s is the activation function. the vector $h(x) = \mathfrak{z}(x) = s(b^{(1)} + W^{(1)}x)$ composed of hidden layer. $W^{(1)} \in R^{D \times D_h}$ is the weights matrix connected from the input layer to the hidden layer. Each column vector $W_{-i}^{(1)}$ on behalf of the from the input node to the i -th hidden layer node weights. Activate

function s usually choose $\tanh(a) = (e^a - e^{-a}) / (e^a + e^{-a})$ or $\text{sigmoid}(a) = 1 / (1 + e^{-a})$, and the model selection of sigmoid function as the activation function. In addition, a practical MLP can have a lot of hidden layer stacked.

The output vector can be obtained: $\sigma(x) = G(b^{(2)} + W^{(2)}h(x))$

The function G can be defined as a softmax function:

$$P(Y = i | x, W, b) = \text{softmax}_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

This function can obtain the probability value of each classification (such as the MNIST data set, that is, the classification number 0-9), so we can use the argmax function to get the class label with the maximum probability value to achieve the classification:

$$y_{\text{pred}} = \text{argmax}_i P(Y = i | x, W, b)$$

Similarly, the MLP of n hidden layers can be expressed as follows: $f: R^D \rightarrow R^L$, where D is the length of the input vector X , L is the length of the output vector $f(x)$, so that you can have the following vector expression:

$$f(x) = G(b^{(n+1)} + W^{(n+1)}(s(b^{(n)} + W^{(n)}(\dots s(b^{(1)} + W^{(1)}x))))))$$

Among them: $b^{(1)}, b^{(2)}, \dots, b^{(n)}$ and $W^{(1)}, W^{(2)}, \dots, W^{(n)}$ are the offset vector and weight vector of first, second, ... n -th hidden layer. $b^{(n+1)}$ and $W^{(n+1)}$ are offset vector and weight vector of output layer. Other variables and functions as above[17].

GA algorithm simulating biological natural genetic and evolutionary process, and in 1975 by JH Holland first introduced, is mainly used to utilize solved optimization problems. A typical application of the genetic algorithm is efficiently searching a large space of possible solutions.

GA algorithm can be defined as an eight tuple:

$$GA = (M, F, s, c, m, pc, pm, n)$$

Where M is the size of the population, F for the individual adaptation degree evaluation function, s is the selection operator, c for the crossover operator, m for mutation operator, pc proportion crossover operator and pm for the proportion of variation, where n is the sample training fractions.

The algorithm of SSGAMLP:

A) Set the initial value, generation $g = 0$.

B) If g is greater than the maximum evolution generation g_{max} , then output the best individual, and quit, if not then continue.

i) $M*pc$ times to select two individuals from the population, then cross, then generate $2*M*pc$ MLP individuals in the pool.

ii) $M*pm$ times to choose an individual, then mutate, and generate $M*pm$ MLP individuals in the pool.

iii) $2M-2*M*pc-M*pm$ times to select from a population in the individual, then directly put into the pool.

iv) For each MLP individual in the pool repeats the following:

- ◆ Adjusting sample order of train set.
- ◆ Training individual MLP model with the adjusted train set.
- ◆ Calculating the error rate misclass with MLP model trained in the valid set.
- ◆ According to the misclass, calculate individual fitness and record the best individual.

v) generating a new generation of population from the pool.

vi) goto B)

3. Experimental Design

In this paper, the experimental data set using MNIST data set. It is composed of handwritten digital images. It is divided into 60000 training data and 10000 test data. The training data is further divided into 50000 training data and 10000 validation data. In order to facilitate the selection of model parameters. All the images have been standardized processing. The size of each image is 28*28. In the original data, the pixels of the image are stored in a common grayscale image (0~255).

The experiment includes the following groups of programs:

A. An MLP contains 2 hidden layers and its hierarchical structure of $structure = [[784, 800],[800, 800],[800, 800],[800, 10]]$. meaning is 784 input layer nodes. The number of nodes in the first hidden layer is 800. The number of nodes in the second hidden is 800. The number of nodes in output layer is 10. Define it as MLP1. We randomly generate 100 MLP1.

B. An MLP contains 2 hidden layers and its hierarchical structure of $structure = [[784, 200],[200, 200],[200, 200],[200, 10]]$. meaning is 784 input layer nodes. The number of nodes in the first hidden layer is 200. The number of nodes in the second hidden is 200. The number of nodes in output layer is 10. Define it as MLP2. We randomly generate 100 MLP2.

C. An MLP contains 2 hidden layers and its hierarchical structure of $structure = [[784, 50],[50, 50],[50, 50],[50, 10]]$. meaning is 784 input layer nodes. The number of nodes in the first hidden layer is 50. The number of nodes in the second hidden is 50. The number of nodes in output layer is 10. Define it as MLP3. We randomly generate 100 MLP3.

D. An SSGAMLP contains individual of MLP1 and $GA = (100, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP1_100.

E. An SSGAMLP contains individual of MLP2 and $GA = (100, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP2_100.

F. An SSGAMLP contains individual of MLP3 and $GA = (100, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP3_100.

G. An SSGAMLP contains individual of MLP1 and $GA = (20, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP1_20.

H. An SSGAMLP contains individual of MLP3 and $GA = (20, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP2_20.

I. An SSGAMLP contains individual of MLP3 and $GA = (20, F, s, C, m, 0.5, 0.45, 5)$. This model is defined as SSGAMLP3_20.

J. In this paper, the model and co-evolutionary algorithm were compared. The co-evolutionary algorithm is using global optimization algorithm (such as PSO or BA algorithm) effect on MLP or RBF neural network. The relevant algorithm description and experimental conditions of reference [3]. The SSGAMLP includes MLP($GA = (100, f, s, cp, mp, 0.5, 0.45, n)$) with $structure = [[784, 800],[800, 800],[800, 10]]$. This model is defined as SSGAMLP_800_100. Another of SSGAMLP includes MLP($GA = (30, f, s, cp, mp, 0.5, 0.45, n)$) with $structure = [[784, 15],[15, 15],[15, 10]]$. This model is defined as SSGAMLP_15_30. The experimental data set using the MNIST data set, including 8000 training data and 2000 test data. These experimental conditions equivalents to the literature [3].

4. The Experimental Results and Analysis

Experiment platform is a GPU server. The GPU chip is nVIDIA GeForce GTX 770, the number of CUDA Stream processor is 1536, GPU memory size is 2G, 8G server memory, operating system is 64-bit Windows 7, the algorithm using CUDA library, make full use of the GPU parallel computing ability.

The experimental results of 100 MLP1s, 100 MLP2s and 100 MLP3s are showed in Table 1. Valid_y_misclass is the error rate on the valid dataset. Table 2 shows

experimental results of the SSGAMLP1_100. Figure 2 is a line chart of Table 2. Table 3 shows experimental results of the SSGAMLP2_100. Figure 3 is a line chart of table 3. Table 4 shows experimental results of the SSGAMLP3_100. Figure 4 is a line chart of Table 4. Figure 5 shows the landscape contrasting in the best individuals of SSGAMLP1_100,SSGAMLP2_100 and SSGAMLP3_100.

Table 1. The Experimental Results of 100 MLP1s, 100 MLP2s and 100 MLP3s

	worst	best	average
training results of 100 MLP1s	0.0558	0.0431	0.0487
training results of 100 MLP2s	0.05	0.0394	0.0449
training results of 100 MLP3s	0.0587	0.0453	0.0516

Table 2. The Experimental Results of the SSGAMLP1_100

Gen	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valid_y_misclass															
average	0.0278	0.0267	0.0259	0.0252	0.0247	0.0244	0.0239	0.0237	0.0234	0.0231	0.0230	0.0228	0.0228	0.0228	0.0228
worst	0.0323	0.0314	0.0300	0.0287	0.0279	0.0274	0.0269	0.0261	0.0254	0.0255	0.0252	0.0252	0.0252	0.0249	0.0251
Best	0.0235	0.0238	0.0228	0.0227	0.0228	0.0223	0.0219	0.0215	0.0218	0.0212	0.0213	0.0213	0.0212	0.0213	0.0213
Gen	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
Valid_y_misclass															
average	0.0228	0.0226	0.0227	0.0226	0.0224	0.0225	0.0225	0.0223	0.0223	0.0223	0.0223	0.0222	0.0223	0.0221	
worst	0.0247	0.0242	0.0251	0.0247	0.0247	0.0247	0.0248	0.0244	0.0244	0.0240	0.0251	0.0247	0.0242	0.0242	
best	0.0211	0.0211	0.0212	0.0211	0.0210	0.0212	0.0210	0.0209	0.0208	0.0208	0.0205	0.0208	0.0208	0.0208	

Table 3. The Experimental Results of the SSGAMLP2_100

Gen	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valid_y_misclass															
average	0.0375	0.0332	0.0309	0.0294	0.0281	0.0269	0.0262	0.0256	0.0253	0.0253	0.025	0.0248	0.0247	0.0246	0.0236
worst	0.0507	0.0392	0.0369	0.0357	0.0328	0.0312	0.0314	0.0284	0.0277	0.0277	0.0273	0.0276	0.028	0.0271	0.0268
Best	0.0312	0.0289	0.0267	0.0256	0.0241	0.0232	0.0236	0.0236	0.0231	0.0233	0.0233	0.0226	0.0223	0.022	0.0211
Gen	16	17	18	19	20	21	22	23	24	25	26				
Valid_y_misclass															
average	0.0234	0.023	0.0229	0.0226	0.0224	0.022	0.0217	0.0216	0.0215	0.0213	0.0212				
worst	0.0268	0.0257	0.0262	0.0256	0.0248	0.024	0.0239	0.0243	0.0242	0.0239	0.0236				
best	0.0214	0.021	0.021	0.0207	0.0205	0.0204	0.0204	0.0199	0.0198	0.0194	0.0193				

Table 4. The Experimental Results of the SSGAMLP3_100

Gen	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valid_y_misclass															
average	0.0445	0.0407	0.0383	0.0366	0.0355	0.034	0.0333	0.0326	0.0323	0.0321	0.0319	0.0319	0.0317	0.0318	0.0316
worst	0.0542	0.0503	0.0468	0.0432	0.0397	0.0382	0.0368	0.0368	0.0362	0.0354	0.0355	0.0351	0.0355	0.0347	0.0348
Best	0.0399	0.0352	0.0342	0.0326	0.0315	0.03	0.0301	0.0297	0.0297	0.0297	0.0298	0.0296	0.0294	0.0287	0.0295
Gen	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Valid_y_misclass															
average	0.0313	0.0311	0.0308	0.0304	0.0302	0.0299	0.0296	0.0293	0.0293	0.0292	0.0291	0.0289	0.0288	0.0289	0.0287
worst	0.0346	0.0337	0.0341	0.0328	0.0335	0.0328	0.0333	0.0316	0.032	0.0326	0.0317	0.0322	0.0322	0.0315	0.0317
best	0.0288	0.0286	0.028	0.028	0.0272	0.0272	0.027	0.027	0.0268	0.027	0.0269	0.027	0.0266	0.0265	0.0263
Gen	31	32	33	34	35	36	37	38	39	40					
Valid_y_misclass															
average	0.0288	0.0285	0.0283	0.028	0.0281	0.0279	0.0279	0.028	0.0279	0.0279					
worst	0.0311	0.0312	0.0316	0.0315	0.031	0.0313	0.031	0.031	0.0307	0.0315					
best	0.0264	0.0258	0.0257	0.0257	0.0257	0.0257	0.0256	0.0257	0.0258	0.0257					

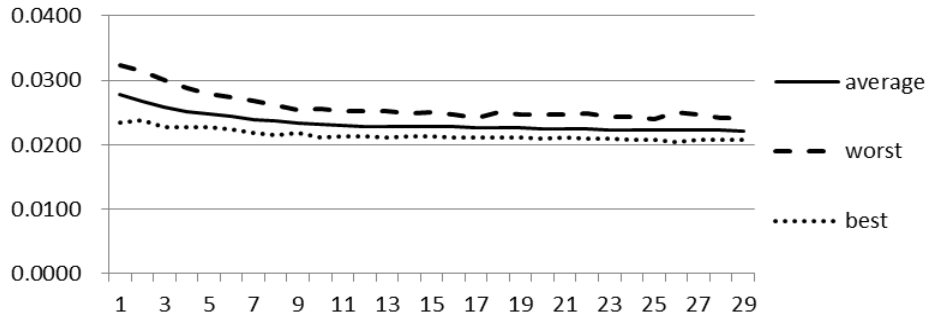


Figure 2. Line Chart of Table 2

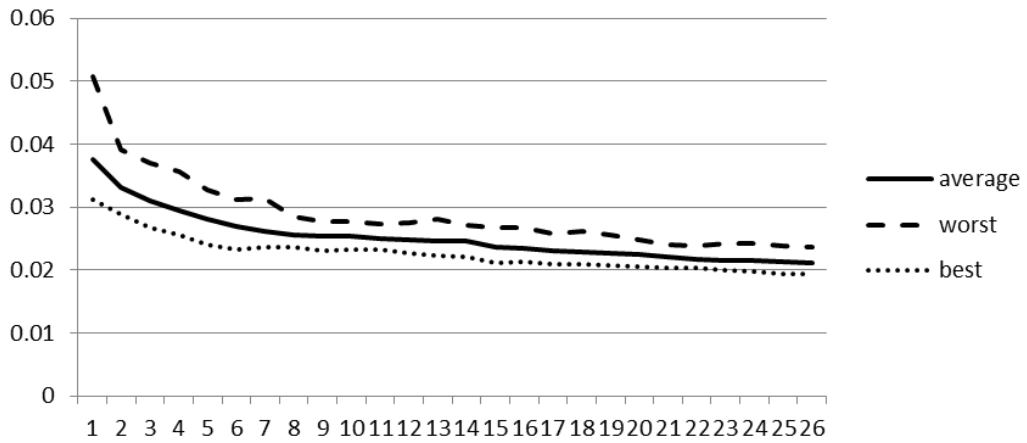


Figure 3. Line Chart of Table 3

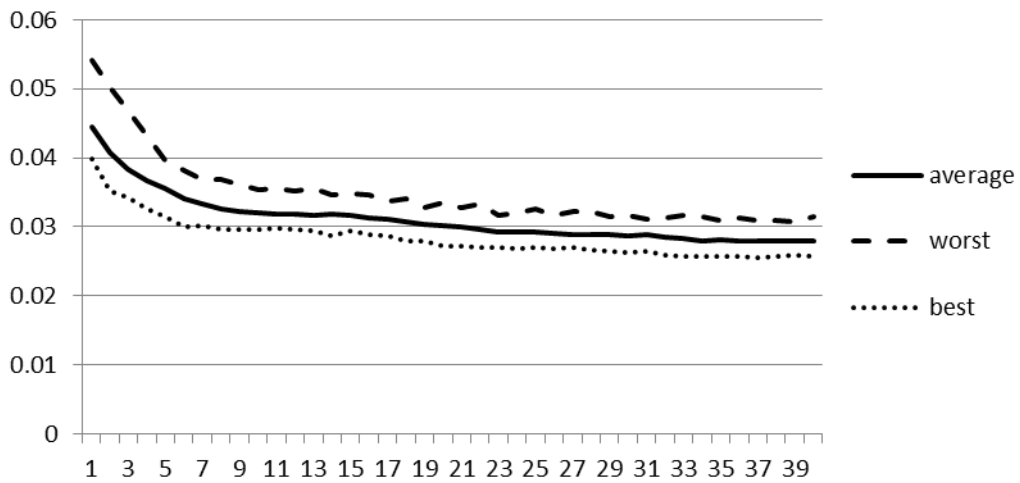


Figure 4. Line Chart of Table 4

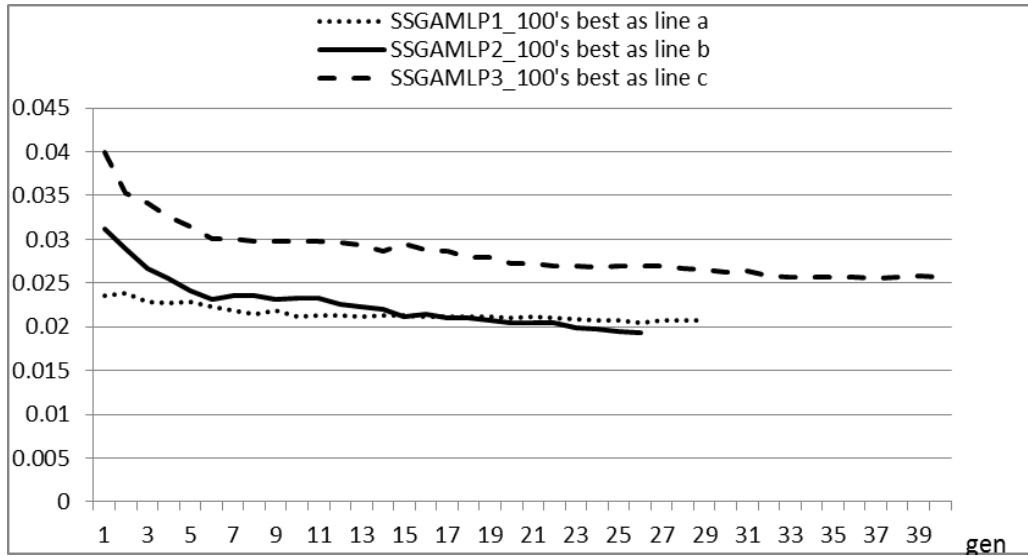


Figure 5. The Landscape Contrasting in the Best Individuals of SSGAMLP1,SSGAMLP2 And SSGAMLP3

For space considerations, we have omitted the table shows. Only the line chart of the experimental results which population scales size is 20 . Figure 6 is a line chart of the experimental results of the SSGAMLP1_20. Figure 7 is a line chart of the experimental results of the SSGAMLP2_20. Figure 8 is a line chart of the experimental results of the SSGAMLP3_20. Figure 9 shows the landscape contrasting in the best individuals of SSGAMLP1_100, SSGAMLP2_100, SSGAMLP3_100SSGAMLP1_20, SSGAMLP2_20 and SSGAMLP3_20.

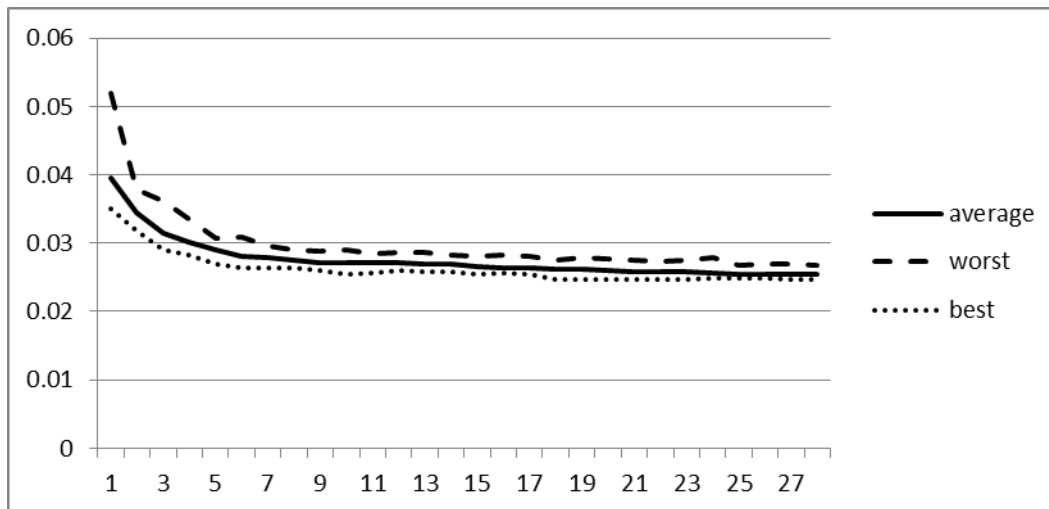


Figure 6. A Line Chart of the Experimental Results of the SSGAMLP1_20

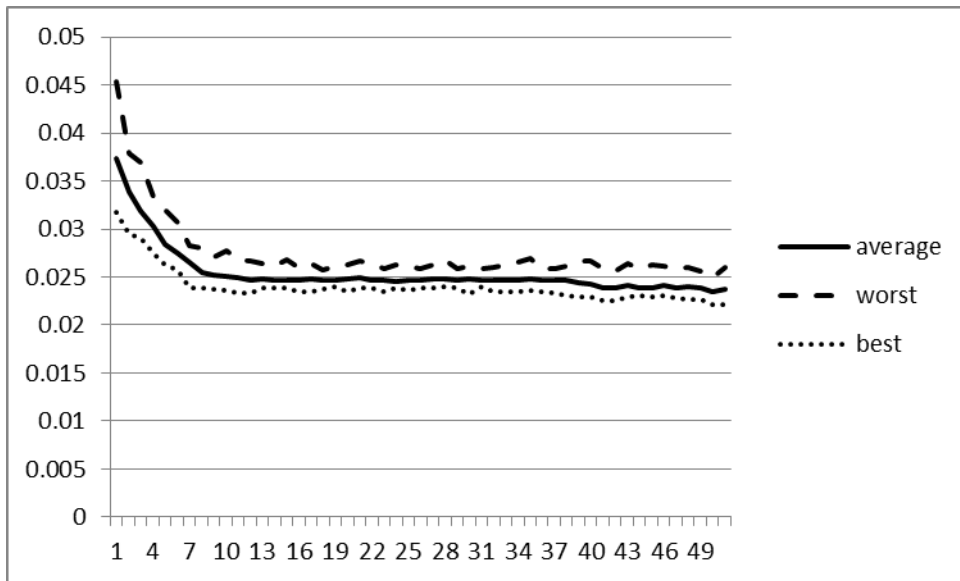


Figure 7. A Line Chart of the Experimental Results of the SSGMLP2_20

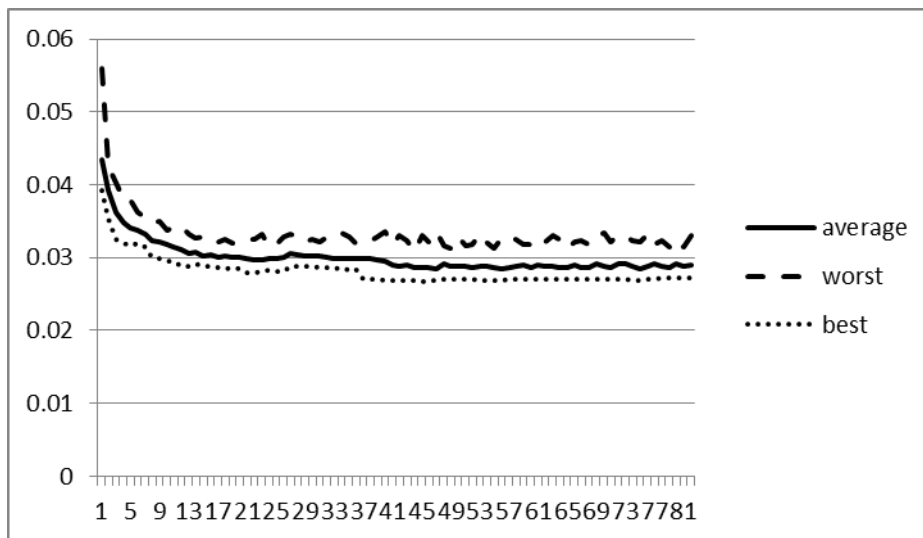


Figure 8. A Line Chart of the Experimental Results of the SSGMLP3_20

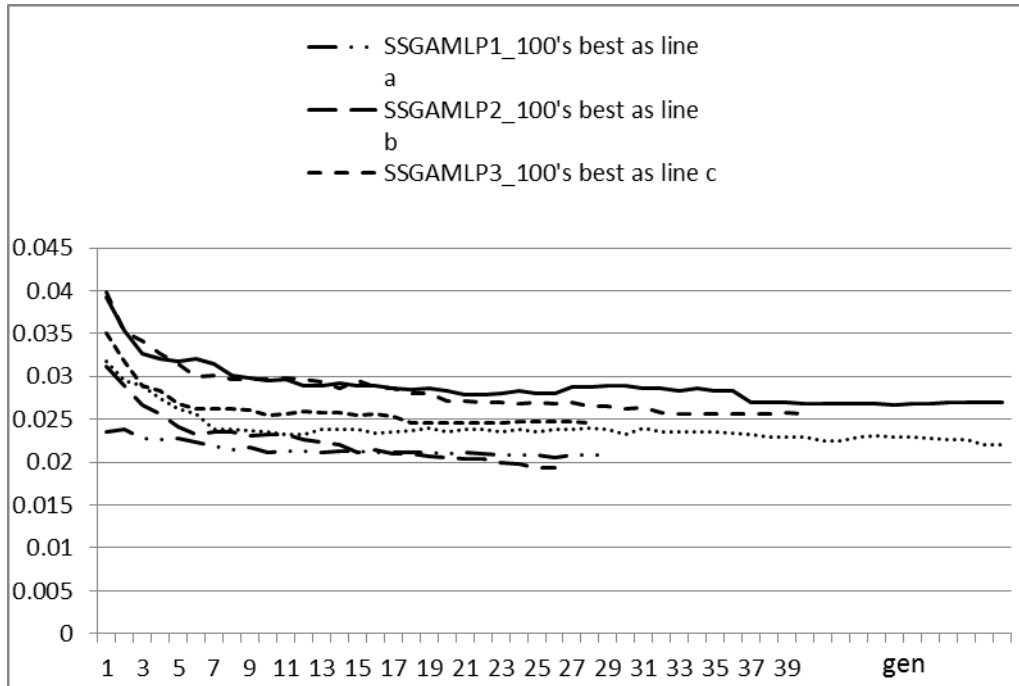


Figure 9. The Landscape Contrasting in the Best Individuals of SSGAMLP1_100, SSGAMLP2_100, SSGAMLP3_100, SSGAMLP1_20, SSGAMLP2_20 and SSGAMLP3_20

For space considerations, we only shows the line chart of results of the SSGAMLP_800_100. Figure 10 is a line chart of the experimental results of the SSGAMLP_800_100.

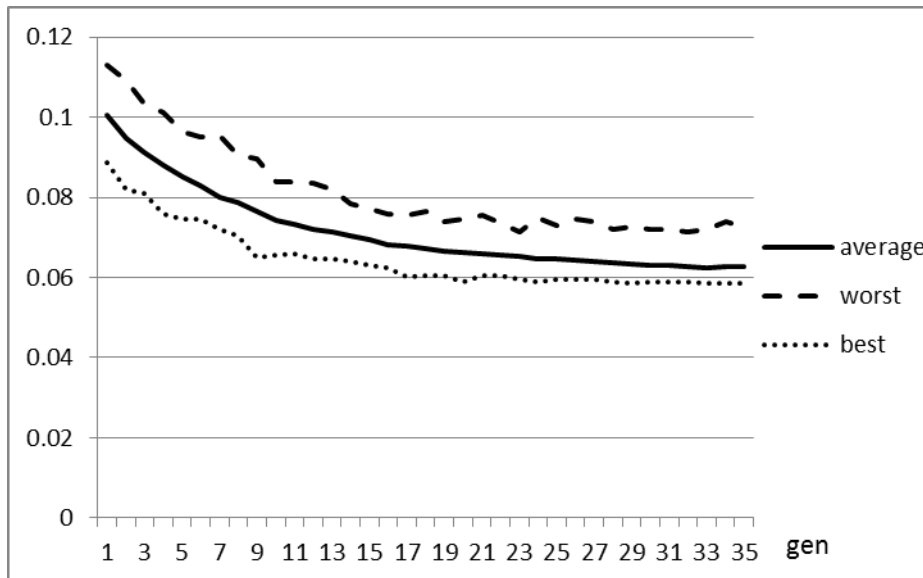


Figure 10. A Line Chart of the Experimental Results of the SSGAMLP_800_100

Table 5. Compared with the Model of Co-Evolutionary and SSGAMLP Model

model	recognition rate(%)
SSGAMLP_800_100	94.150
SSGAMLP_15_30	92.100
co-evolutionary RBF	88.790
co-evolutionary MLP	86.000
Statistical classifiers based on PSO(Zoning)	80.428
Statistical classifiers based on BA(Zoning)	92.857

According to the above experimental data, draw the following conclusions:

(1) In Table 1, three kinds of MLPs which are same layers number but different number of hidden nodes were compared. The experimental result showed that the number of hidden nodes more, better training results.

(2) From Figure 2,3,4,6,7,8, we can see that the performance of model was improved with the increase of generations continuously.

(3) From Figure 5 and Figure 9, in the same population size of the case, the number of hidden nodes increasing, the performance of the model is better.

(4) From Figure 9, SSGAMLP1_100(line a) and SSGAMLP1_20(line d) have the same number of hidden layer nodes but their different population size. The performance of SSGAMLP1_100 is better than SSGAMLP1_20, Similarly SSGAMLP2_100 and SSGAMLP2_20, SSGAMLP3_100 and SSGAMLP3_20. The conclusion is that the population size is larger, the performance of the model is better. The number of hidden nodes and population size have an effect on the performance of the model together. The effect of only increase the number of hidden layer nodes or population size is less than the effect of increase both them. The computational complexity of SSGAMLP2_100(line b) is only 1/4 of SSGAMLP1_100(line a), but after a few generations, can obtain similar performance of SSGAMLP1_100.

(5) From Table 5, under the roughly the same experimental conditions , The SSGAMLP model can achieve better performance than co-evolutionary RBF, co-evolutionary MLP, Statistical classifiers based on PSO(Zoning)and Statistical classifiers based on BA(Zoning) model .

5. Summary

MLP has the over fitting problem. The generalization performance is a very important aspect of MLP. The Train set into n subsets. The each individual of SSGAMLP model randomly selects subset. There are two advantages of learning in the subset. The first advantage is to reduce the computational complexity of the individual, because the training set is reduced. So that each individual training time is reduced. The second advantage are to improve the generalization performance, because each individual training is based on the random subset, but not in the whole Train data set, equivalent to the introduction of random factor. The number of hidden nodes or population size increasing, the performance of the model is better. The next research is to combine deep learning and genetic algorithm to improve the generalization performance of deep learning model.

Acknowledgments

The research work was supported by Natural Science Foundation of Fujian Provincial under Grant No. 2013J01243 and Science Project Foundation of Fujian Provincial under Grant No. 2013H0032.

References

- [1] Y. Zhining and P. Yunming, "Random sample of genetic algorithm with MLP model", *Computer Engineering and Applications*, vol. 51, no. 21, (2015), pp. 121-127.
- [2] Y. Zhining and P. Yunming, "The Genetic Convolutional Neural Network Model Based on Random Sample", *International Journal of u- and e- Service, Science and Technology*, vol. 8, no. 11, (2015), pp. 317-326.
- [3] N. S. Boukerram, "A Handwritten characters recognition based on nature-inspired computing and neuro-evolution", *Application Intell.*, vol. 38, (2013), pp. 146-159.
- [4] A. Abraham and B. Nath, "Optimal design of neural nets using hybrid algorithm", *Proceedings of 6th Pacific Rim International Conference on Artificial Intelligence*, (2000), pp. 510-520.
- [5] J. Kang and S. Kim and S. Y. An, "A new approach to simultaneous localization and map building with implicit model learning using neuro evolutionary optimization", *Application Intell*, vol. 36, no. 1, (2012), pp. 242-269.
- [6] J. Kessler, K. Rasheed and I. B. Arpinar, "Using genetic algorithms to reorganize superpeer structure in peer to peer networks", *Application Intell*, vol. 26, no. 1, (2007), pp. 35-52.
- [7] S. A. Khan and A. P. Engelbrecht, "A fuzzy particle swarm optimization algorithm for computer communication network topology design", *Application Intell*, vol. 36, no. 1, (2012), pp. 161-177.
- [8] K. Kim and H. Ahn, "Simultaneous optimization of artificial neural networks for financial forecasting", *Application Intell*, vol. 36, no. 4, (2012), pp. 887-898.
- [9] B. J. Ross and E. Zuviria, "Evolving dynamic Bayesian networks with multi-objective genetic algorithms", *Application Intell*, vol. 26, no. 1, (2007), pp. 13-23.
- [10] I. Valova, G. Milano and K. Bowen, "Bridging the fuzzy, neural and evolutionary paradigms for automatic target recognition", *Application Intell*, vol. 35, no. 2, (2011), pp. 211-225.
- [11] R. Ghosh and M. Ghosh, "An intelligent offline handwriting recognition system using evolutionary neural learning algorithm and rule based over segmented data points", *J Res Practice Information Technol*, vol. 37, no. 1, (2005).
- [12] R. Kaur and B. Singh, "A hybrid neural approach for character recognition system", *International Journal Computer Science Information Technology*, vol. 2, no. 2, (2011), pp. 721-726.
- [13] M. Mangal and M. P. Singh, "Handwritten English vowels recognition using hybrid evolutionary feed-forward neural network", *Malays Journal Computer Science*, vol. 19, no. 2, (2006), pp. 169-187.
- [14] L. S. Oliveira, A. de S. Britto and R. Sabourin, "Improving cascading classifiers with particle swarm optimization", *Proceedings of ICDAR'2005*, (2005), pp. 570-574.
- [15] R. Kala, H. Vazirani and N. Khawalkar, "Evolutionary radial basis function network for classificatory problems", *International Journal Computer Science Application*, vol. 7, no. 4, (2010), pp. 34-49.
- [16] Y. Li, J. Li and M. Li, "Character recognition based on hierarchical RBF neural networks", *Proceedings of ISDA*, (2006), pp. 127-132.
- [17] X. X. Fen and L. Bo, "Pattern Recognition Using an Improved Back-propagation ANN", *WANG De-jun. Journal of Harbin University of Science and Technology*, vol. 9, no. 5, (2004), pp. 63-65.

