Modified particle swarm optimization algorithm and its application in neural network

Qiulin Xu

College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China

zsb@sdust.edu.cn

Email:1060170926@qq.com

Abstract: Intelligent optimization algorithm solves the optimization problem by simulating some natural phenomena and provides a new method for optimization theory. As a swarm based stochastic optimization algorithm, particle swarm optimization (PSO) is widely used in various optimization problems due to its simple model and fast convergence speed. However, particle swarm optimization algorithm still has some problems to be solved, such as premature phenomenon and balancing global exploration and local exploitation capabilities. In order to solve these problems, a nonlinear inertial weights and adaptive position updating strategy based on chaos is proposed. The algorithm can effectively avoid premature convergence of particle swarm, and has better convergence and faster convergence rate. In this paper, the modified particle swarm optimization algorithm (MPSO) is used as a learning algorithm for neural network optimization, and the MPSO is used to optimize the connection weights and thresholds of the neural network. In this paper, three sets of standard classification data sets are used for testing. Experimental results and statistical analysis verify the good performance of the MSPO.

1. Introduction

Optimization problem is based on mathematical knowledge and has been widely used in scientific engineering and other fields. Optimization method and its application in complex system is a challenging subject. There are many complex problems in engineering practice, and the traditional optimization methods are difficult to deal with high dimensional or nonlinear optimization problems. Due to the limitations of traditional algorithms, a large number of swarm intelligence algorithms have been designed and become one of the hot research fields of algorithms. Many experiments show that swarm intelligence algorithm is more effective in solving various complex optimization problems. At present, the widely used swarm intelligence algorithms mainly include ant colony algorithm[1], particle swarm optimization algorithm[2], frog hop algorithm[3] and so on. PSO was first proposed by Eberhart and Kennedy in 1995. It is an intelligent algorithm based on swarm search, which simulates the foraging behavior of birds to search for the optimal solution. The improvement research of PSO mainly includes the following aspects: improving weight, updating formula and combining with other optimization algorithms. Shi and Eberhart[4] proposed an improved method of inertial linear weight decreasing. Deep[5] improved the update formula by changing the individual optimal position and the global optimal position. Garg[6] combines particle swarm optimization algorithm with genetic algorithm. All the above improved methods improve the performance of the algorithm.
Artificial neural network (Ann) is a model that simulates human brain learning and memory. After decades of development, it has become an important research field of ARTIFICIAL intelligence. With the continuous development of optimization theory, the combination of intelligent algorithm and neural network has been widely studied. PSO can be used to optimize the parameters of neural network, which can enhance the learning ability of neural network. Therefore, the modified particle swarm optimization algorithm and its application in neural networks are important research directions in the field of industrial production. In this paper, through the improvement of PSO, a better optimization model is obtained.

2. Particle swarm optimization algorithm

SPO is an intelligent algorithm proposed by Eberhart and Kennedy in 1995, which solves optimization problems by simulating the foraging behavior of birds. PSO does not need gradient information of optimization problem, but seeks the optimal solution through mutual cooperation among individuals and information sharing among groups. Suppose that the search space of optimization problem is \( D \) dimensional, and \( N \) is number of particle swarm, and the position and velocity of \( i \) th particle of the swarm at time \( t \) are recorded as \( \{X_{i1}(t), X_{i2}(t), \ldots, X_{iD}(t)\} \) , and \( \{V_{i1}(t), V_{i2}(t), \ldots, V_{iD}(t)\} \) , \( i = 1, 2, \ldots, N \), respectively. At each iteration, the velocity and position of particle \( i \) will be updated according to the following equations

\[
V_i(t+1) = \omega(t)V_i(t) + c_1 \cdot r_1(p_{best_i}(t) - X_i(t)) + c_2 \cdot r_2(g_{best}(t) - X_i(t))
\]

\[
X_i(t+1) = X_i(t) + V_i(t+1)
\]

where \( \omega \) is the inertia weight, \( c_1 \) and \( c_2 \) are cognitive and social scaling parameters respectively, \( r_1 \) and \( r_2 \) are two random numbers which are uniformly distributed in the interval \((0,1)\). The past optimal position of \( i \) th particle and the global optimal position are defined as Equation(3) and (4), respectively.

\[
p_{best_i}(t) = \arg \min \{ \text{fit}(X_i(1)), \text{fit}(X_i(2)), \ldots, \text{fit}(X_i(t)) \}
\]

\[
g_{best}(t) = \arg \min \{ \text{fit}(p_{best_1}(t)), \text{fit}(p_{best_2}(t)), \ldots, \text{fit}(p_{best_N}(t)) \}
\]

Obviously, the particle positions of swarm are varied with the increase of the number of iterations through Equation(1) and (2).

3. Modified particle swarm optimization

In order to avoid the premature phenomenon of the PSO algorithm and balance the local exploration and global exploration capabilities, we have improved the algorithm from the following two aspects

3.1. Chaotic-based inertia weight

Inertia weight \( \omega \) as one of the parameters of particle swarm optimization algorithm, can provide dynamic adjustment ability for particles in different environments, so as to achieve the balance of exploration and development. Therefore, it plays an important role in PSO. The linear method of inertia weight is generally adopted, but the nonlinear inertia weight has strong fitting and simulation capability. Logic is a chaotic map, which generates random numbers between 0 and 1. It’s defined by Equation(5). This paper introduces logical chaos into inertia weight and constructs a nonlinear inertia weight defined by Equation(6). The simulation diagram is shown in Fig.1.

\[
r(t+1) = 4r(t)(1-r(t)), r(0) = r_{n}\left\{0,0.25,0.5,0.75,1\right\}
\]
3.2. Adaptive position updating strategy

During iteration, different location update strategies have different exploration and exploitation capabilities. In order to balance global exploration and local exploitation, an adaptive location updating mechanism is proposed. Using this mechanism, the particle can choose the position updating strategy according to the corresponding conditions, so as to better balance exploration and exploitation. As mentioned earlier, the adaptive policy location updating strategy is defined by Equation (7) and (8).

\[
\omega(t) = r(t) \cdot \omega_{\text{max}} - \frac{(\omega_{\text{max}} - \omega_{\text{min}}) \cdot t}{T_{\text{max}}}
\]

\[
X_i(t+1) = \begin{cases} X_i(t) + V_i(t+1) & p_i < \text{rand} \\ \omega(t)X_i(t)+(1-\omega(t))V_i(t+1)+g_{\text{best}(t)} & \text{otherwise} \end{cases}
\]

In the above adaptive strategy, in each iteration, the particle can obtain an estimated value recorded as \( p_i \) according to the ratio between the current fitness of particle \( i \) and the average fitness. Adjust the update formula according to the value of \( p_i \), which can effectively balance the global exploration and local exploitation capabilities.

4. Test Results and Discussions

The neural network used in this experiment is a two-layer perceptron network. In this experiment, two sets of data sets of standard classification problems were selected as test examples, namely Iris and Wine classification problems data sets. The corresponding structural domain information of the neural network is shown in Table 1.

| Classification problem | The number of samples | The network structure | Number of weights |
|------------------------|-----------------------|----------------------|------------------|
| Iris                   | 150                   | 4-10-3               | 83               |
| Wine                   | 178                   | 13-5-3               | 88               |
For the purpose of testing and comparing, besides MPSO, basic PSO, standard genetic algorithm (GA), gradient descent method (GD) and BP algorithm are also used to optimize the perceptron weight, and the classification performance of these algorithms is analyzed and compared. The classification results are shown in Table 2 and Table 3.

### Table 2. Comparison of error rates of various optimization algorithms on Iris classification problems.

| The algorithm name | Training set error rate | Test set error rate |
|--------------------|-------------------------|---------------------|
| MPSO               | 0.71±0.29               | 3.24±1.05           |
| PSO                | 1.14±0.45               | 2.13±1.71           |
| GA                 | 3.21±0.55               | 5.25±2.34           |
| GD                 | 2.57±0.87               | 5.89±1.46           |
| BP                 | 2.97±0.58               | 4.47±2.64           |

### Table 3. Comparison of error rates of various optimization algorithms on Wine classification problems.

| The algorithm name | Training set error rate | Test set error rate |
|--------------------|-------------------------|---------------------|
| MPSO               | 1.91±0.12               | 1.21±1.85           |
| PSO                | 2.14±0.45               | 2.82±1.72           |
| GA                 | 3.61±0.65               | 4.21±2.78           |
| GD                 | 4.56±0.87               | 4.82±1.26           |
| BP                 | 2.97±0.58               | 7.98±1.42           |

This section presents the neural network model based on MPSO algorithm, and compares its performance with other learning algorithms. The results show that the neural network model based on MPSO algorithm has strong generalization ability and high classification accuracy.

### 5. Conclusion

Based on the above experimental results and discussion, the following conclusions are drawn. PSO is improved by chaotic inertia weight and adaptive position updating strategy. The modified PSO is applied to neural network optimization. By comparing with several traditional algorithms, MPSO is proved to have better convergence and significantly improves the classification accuracy of the classification set.

### References

[1] Dorigo M, Maniezzo V. (1992) Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life. 134-142.

[2] Kennedy J, Eberhart R C. (1995) Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. 1942-1948.

[3] Eusuff M M, Lansey K E . (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management. 129(3):210-225.

[4] Shi Y, Eberhart R C. (1998) A modified particle swarm optimization. Proceedings of the IEEI International Conference on Evolutionary Computation.69-73.

[5] Deep K, Bansal J C. (2009) Mean particle swarm optimization for function optimization. International Journal of Computational Intelligence Studies. 1(1):72-92.

[6] Grag H. (2016) A hybrid PSO-GA algorithm for constrained optimization problems. Applied Mathematics and Computation.274(11):292-305.

[7] Kennedy J. (2000) Stereotyping:improving particle swarm performance with cluster analysis. In:Proceedings of the IEEI Conference on Evolutionary Computation. Piscataway. 1507-1512.