MACROJOURNALS

The Journal of **Macro**Trends in Technology and Innovation

Artificial Neural Network Trained by Local and Global Optimization Methods for Classification of Car Evaluation Dataset

Emre Çomak and Gökhan Uçkan

Department of Computer Engineering, Pamukkale University, 20020 Denizli, TURKEY

Abstract

One of the most commonly studied topics in machine learning is the classification problem. In this paper, ANN classifier was trained by two alternative methods, i.e. BP (Back Propagation) and PSO (Particle Swarm Optimization) over car evaluation dataset. In addition, PSO was implemented in two different modes, i.e. linearly increasing and decreasing inertia weight modes. Finally, totally three classifiers' experimental results in this study and SVM (Support Vector Machine) trained by PSO classifiers' result were compared on car evaluation dataset. According to experimental results, PSO with linearly decreasing inertia weight method provided an important improvement to ANN training with respect to BP method.

Keywords: *ANN (Artificial Neural Network), PSO (Particle Swarm Optimization), car evaluation dataset.*

1. Introduction

Classification problems place in an important position in machine learning researches. Artificial Neural Network (ANN) is a machine learning technique can be run in supervised or unsupervised training mode. ANN has been widely used as a solver in many classification, clustering and approximation problems **[1].**

Particle Swarm Optimization (PSO) is also one of the machine learning techniques can be used especially in optimization and searching problems **[3].** PSO creates a model simulating natural behavior of a bird flock or a fish school when they search optimal food source. It can be a strong candidate solver for any optimization problem including the training process of some classifiers. Lin S. W. et al. **[2]** introduced a new hybrid methodology using PSO and Support Vector Machine (SVM) for classification of some data sets from UCI **[4]** database. Tuning of SVM training parameters and feature selection were conducted by PSO in this study.

ANNs can be constructed on different architectures with respect to their neuron hierarchy and connection orders between neurons. The most commonly used one among these architectures is feed-forward, fully-connected ANNs, i.e. multilayer perceptrons (MLPs) with the back-propagation (BP) training approximation. BP is a gradient based method implemented in an iterative way. It has two fundamental risks. The fist of them is local optimal problem arising from the local solution idea in BP. The second one is dependence between initial parameter values of BP and classification performance of ANN with BP **[5].**

In training of ANNs; BP can search a local area in a search space constructed on weight and training error values while PSO can search the global area entirely in the same search space. The first risk in BP we explained briefly above paragraph was eliminated entirely in PSO, while the second one was eliminated partially in PSO. Many researchers have applied PSO for training of ANNs in different problems **[6, 7, 8, 9, 10].**

In this paper; in order to train an ANN one local (i.e. BP) and one global (i.e. PSO) optimization methods were implemented. In addition, the effects of these local and global methods on classification performance of ANN were researched. These systems were experimented on car evaluation classification problem obtained from UCI machine learning dataset **[4].** An MLP architecture with 6 input, 5 hidden and 1 output layer neurons was constructed as an ANN model. Tangent sigmoid and linear activation functions were used in hidden layer and output layer, respectively. Since classification performance of ANN depends on several criteria such as ANN architecture, initial parameters' values, problem characteristics, used activation function(s) and training method. Only training method (and initial parameters' values according to training method) was changed in this study, while the other criteria were not changed for a fair comparison between BP and PSO methods. Besides, PSO was run in two different modes, i.e. PSO with linearly increasing and linearly decreasing inertia weight. Thus, two types of PSO were also compared with each other in experimental results.

Remaining of the paper is organized as follows. Section 2 involves methods. Experimental tunings and results are presented in Section 3. Finally, Section 4 includes the conclusion and discussion parts.

2. A Multilayer Perceptron Artificial Neural Network (MLP-ANN)

MLP is a type of ANN, which is constructed on three or more layers and connections feedforward direction way. Each layer is fully connected to the next layer from input to output. MLPs are trained by supervised learning style. As a distinct from linear perceptron, MLP manages to separate different classes in a linearly inseparable space by using non-linear activation functions. Many gradient based methods such as gradient descent with momentum, scaled conjugate gradient (SCG), resilient propagation (RPROP), BFGS quasi-Newton, and Levenberg-Marquardt (LM) and also, evolutionary computing methods such as genetic algorithm (GA), particle swarm optimization (PSO), and artificial immune system (AIS) can be used to train MLPs.

In this study, BP and PSO methods were used as the training methods. More information about these methods are clarified in the following parts of the paper.

2.1 BP Method

Back-propagation is a gradient based method can be implemented in supervised learning strategy **[11]**. Thus, it is commonly used for training of MLPs.
As illustrated in Figure 1, at least one hidden layer has to be in an MLP structure. Hidden layers are labeled by using an "H" as an initial character. We must prefer at least one non-linear activation function in one of the hidden layers in order to classify non-linear surfaces in the input space. BP method conducts ANN training as an optimization problem in that we try to minimize the mean square error between ANN output and target output values for all

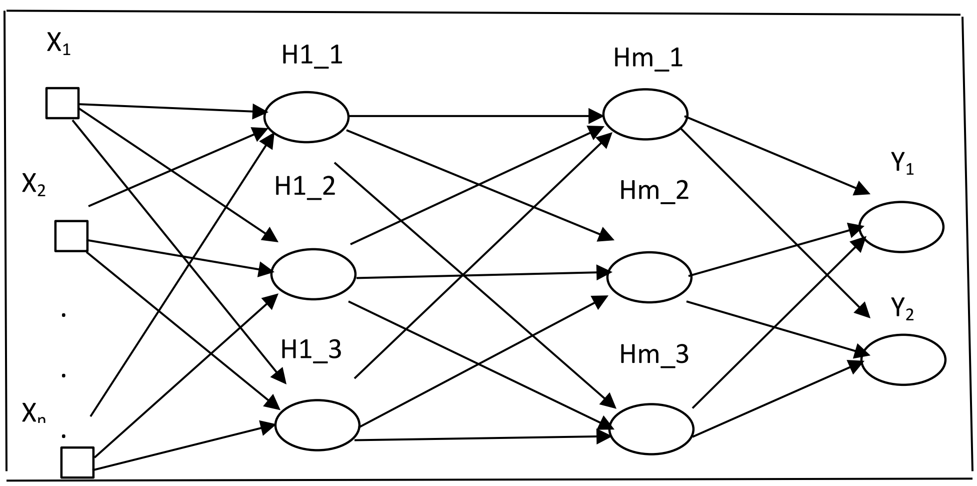

Figure 1 An Example of MLP Structure

training dataset. The name of "backpropagation" arises from direction of the error which is propagated from output to input layer. Pseudo code of BP is as follow:
- Initialize weight, bias and threshold values randomly,
- Compute the ANN output for each training datum separately,
- Compute the mean square error with respect to difference between ANN output and target output values,
- Calculate the local gradients of total error value with respect to weights,
- Change weights and threshold values with respect to previous weight value and local gradients multiplied by learning rate parameter.

BP method cannot exhibit a global searching behavior in an error space. The reason of this is resulted from BPs' original algorithm which directs itself with respect to only one candidate solution point. In addition to this reason, BP method starts its algorithm steps from a set of randomly determined training parameter values. This reason and property of BP expose it to local optimal problem. Detailed discussion on BP can be researched in **[12]**.

2.2 PSO Method

PSO is an optimization and searching method inspired from social and biological behavior of the natural swarms **[3]**. A swarm is comprised by N particles and each particle is a randomly generated candidate for the optimal solution in PSO. Each particle is determined as a vector with D independent values, i.e. dimensions. This dimension value depends on natural of the problem. According to these assumptions, position of the i_{th} particle in a swarm is represented by $x_i = (x_{i1}, x_{i2},…,x_{iD})$ while its' velocity equation is represented by $v_i = (v_{i1}, v_{i2},…,v_{iD})$. Furthermore, $p_i = (p_{i1}, p_{i2},…,p_{iD})$ term is used for position of the best individual performance of i_{th} particle while $p_g = (p_{g1}, p_{g2},…,p_{gD})$ term is used for the global (namely, all neighbors of the i_{th} particle or swarm entirely) best position. In each iteration of PSO; x, v, p_i and p_g vectors are updated with respect to following equations.

$$v_{id} = w * v_{id} + c_1 * rand1(\ ) * (p_{id} - x_{id}) + c_2 * rand2(\ ) * (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

Where i=1, 2, … ,N and d=1, 2, … ,D. Cognitive and social constants affecting magnitude of the velocity are presented as c_1 and c_2. Functions named as rand1 and rand2 generate numbers randomly and uniformly. The variable of w is an inertia weight parameter tuning a balance between global and local searching in PSO method.

Traditional PSO begins its' solution by setting a randomly created population and velocities. In each passing from current iteration to the next, equations (1) and (2) are computed iteratively. In case of some particles exceed the search space, V_{max} value can be set to avoid this exceeding.

3. Car Evaluation Dataset and Experimental Results

In this study, the classification performances of two ANN types i.e. ANN trained by BP and ANN trained by PSO were compared on car evaluation dataset. Section 3.1 includes description about this dataset and Section 3.2 includes the experimental results.

3.1 Car Evaluation Dataset

This dataset consists of six attributes; buying price, price of the maintenance, number of doors, person capacity, the size of luggage boot, and safety of the car. The first three attributes consist of four levels of value while the last three ones consist of three levels of value. This dataset consists of 1728 data whose 1211 number belongs to class1, 385 to class2, 70 to class3, and remaining 66 to class4.

3.2 Experimental Results

ANN was trained by BP and PSO methods separately to classify car evaluation dataset. BP was set with 1000 maximum iteration, 10^{-10} gradient, and 10^{10} mutation parameters. Besides, PSO was constructed with following parameter values;
- population size = 25,
- c1 (cognitive) = 2,
- c2 (social) = 2,
- dimension = 40,
- upper bound = 100,
- lower bound = -100,
- maximum iteration number = 2000 and
- error goal = 10^{-6}.

Since a fully connected MLP model with 6 input (with extra 1 bias), 5 hidden and 1 output neurons was implemented, 40 weights between all neurons were created. Thus, dimension of each particle were determined as 40 in PSO to be able to represent these connection weights. PSO was run in both decreasing and increasing inertia weight modes, and the application results of these modes were also compared in addition to BPs' results.

Since PSO was initialized with randomly created population, PSO was run 50 times in all modes for a fair comparison. After that average and standard deviation values of ANN with PSO were computed. ANN trained by BP was also run 50 times, and average and standard deviation values were computed again. To determine a dividing between training and test datasets, 10 fold cross validation approximation was preferred in both BP and PSO based training methods.
All results explained in above paragraph are presented in Table1.

Table1 Training and Test Errors of ANN with Different Training Methods and SVM with PSO

Method	**Average Fitness**	**Standard Deviation**
ANN + BP	0.0765*, 77.73%	0.0004*, 0.0159
ANN + linearly increasing inertia weight PSO	0.8489*, 75.83%	0.0018*, 0.0307
ANN + linearly decreasing inertia weight PSO	0.5627*, 81.32%	0.0012*, 0.0084
SVM + PSO **[1]**	99.89%	0.161

* reflects Mean Square Error (MSE) for training dataset

4. Conclusion and Discussion

When the experimental results in section 3.2 are compared, SVM based on PSO training reflects the best average classification performance for the car evaluation dataset. In addition; ANN based on linearly decreasing inertia weight PSOs' classification performance for test dataset overcomes ANN with BPs' one, however ANN based on BP method reflects better classification performance in training dataset. These results show that generalization learning ability of ANN with linearly increasing/decreasing inertia weight PSO is better than ANN with BP method.

The effects of two PSO modes on ANN classification performance were researched in this study and an important improvement in test classification performance was surveyed according to experimental results. Improvement versions of PSO method were also introduced in the literature. Only an assessing research study was introduced in this study. By using a suitable improved PSO method, the classification performance of ANN trained by PSO might also be increased.

Acknowledgement

This work is supported by Pamukkale University Scientific Research and Projects Unit.

References

[1] N. B. Karayiannis & G.W. Mi, "Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques," IEEE Trans. Neural Networks, vol. 8, pp. 1492–1506, Nov. 1997.

[2] S. W. Lin, K. C. Ying, S. C. Chen, Z. J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," Expert Systems with Applications, vol. 35, pp. 1817-1824, 2008.

[3] J. Kennedy, R. C. Eberhart, "Particle swarm optimization," In Proceedings of the IEEE Conference on Neural Networks. Australia, p. 1942-1948, 1995.

[4] S. Hettich, C. Blake, & C. Merz, "UCI repository of machine information and computer sciences," available at http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[5] J. F. Kolen, & J. B. Pollack, "Back propagation is sensitive to initial conditions," Complex Systems, 4, pp. 860–867, 1990.

[6] Y. Lu, N. Zeng, Y. Liu, N. Zhang, "A hybrid Wavelet Neural Network and Switching Particle Swarm Optimization algorithm for face direction recognition," Neurocomputing, vol. 155, pp. 219-224, 2015.

[7] A. Suresh, K. V. Harish, N. Radhika, "Particle swarm optimization over back propagation neural network for length of stay prediction," Procedia Computer Science, vol. 46, pp. 268-275, 2015.

[8] M. Carvalho, & T. B. Ludermir, "Particle swarm optimization of neural network architectures and weights," In Proceedings of the 7th international conference on hybrid intelligent systems, Washington DC, pp. 336–339, 2007.

[9] J. Yu, L. Xi, S. Wang, "An improved particle swarm optimization for evolving feedforward artificial neural networks," Neural Processing Letters, vol. 26(3), pp. 217–231, 2007.

[10] T. Ince, S. Kiranyaz, J. Pulkkinen, M. Gabbouj, "Evaluation of global and local training techniques over feed-forward neural network architecture spaces for computer-aided medical diagnosis," Expert Systems with Applications, vol. 37, pp. 8450-8461, 2010.

[11] Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representation by error propagation. In Parallel distributed processing: Explorations in the microstructure of cognition, pp. 318–362. Cambridge, MA: MIT Press.
[12] Y. Chauvin, & D. E. Rumelhart, “Back propagation: Theory, architectures, and applications,” Associates Publishers, Hillsdale, NJ, USA: Lawrence Erlbaum, 1995.