Training Feed-forward Neural Networks using Asexual Reproduction Optimization (ARO) Algorithm

Mehdi Zekriyapanah Gashti
Department of Computer Engineering,
Payame Noor University Tehran, Iran
Email: gashti@pnu.ac.ir

Rouhollah Habibey
IUF Leibniz Research Institute for
Environmental Medicine Dsseldorf, Germany
Email: Rouhollah.habibey@iit.it

***Abstract*—Artificial neural networks have been increasingly used in many problems of data classification because of their learning capacity, robustness and extendibility. Training in the neural networks accomplished by identifying the weight of neurons which is one of the main issues addressed in this field. The process of network learning by back-propagation algorithm which is based on gradient, commonly fall into a local optimum. Due to the importance of weights and neural network structure, evolutionary neural networks have been emerged to obtain suitable weight set. This paper will concentrate on training a feed-forward networks by a modified evolutionary algorithm based on asexual reproduction optimization (ARO) in order to data classification problems. The idea is to use real representation (rather the binary) for adjusting weights of the network. Experimental results show a better result in terms of speed and accuracy compared with other evolutionary algorithms including genetic algorithms, simulated annealing and particle swarm optimization.**

Keywords*—*Neural Network, Asexual reproduction optimization, Real representation, Local optimum.

I. Introduction

Artificial Neural Networks (ANNs) have many application in machine learning and pattern recognition. Feed-forward back-propagation algorithm, which is main algorithm of network training, have been used for many years for this purpose. Feed-forward back-propagation algorithm which is also called Back-Propagation (BP) is applied in several area such as computer vision, robotic, medicine, etc. The back-propagation neural network is suitable method for modeling non-linearity functions which is use in many applications of detection and recognition such as face recognition, image classification and signal modeling. Back-propagation algorithm is based on gradient descent (GD) which try to minimize the error function (cost function) in weight space. Weights of the network are tuned using training data to minimize the cost function of the network output based on actual value of data labels. Training process is complete when cost function achieves best weights in the weight space.

There are several algorithms developed for training neural networks which are based on back-propagation method that their performance depends on various factors such as number of layers, the number of neurons for each layer, activation functions and initial weights [1] [2] [3].

There are several forms of back-propagation algorithms in the literatures so that the conventional back-propagation algorithm is one of them which based on the gradient descent. Back-propagation algorithm does not guarantee to find a global minimum and usually fall into a local optimum. This is because back-propagation algorithm for neural network training is indeed an optimization problem and use gradient descent for estimate the best solution which in most time it stuck in local minimum.

Furthermore, in the solution domain, there are several evaluation functions which are discontinuous, non-linearly separable and complex approximation. So, traditional algorithms based on derivatives do not work well for these functions. There are also some drawbacks to back-propagation algorithm; First, the convergence of back-propagation algorithm is slow and it usually fall into any local optimum on the error surface. Next, there is scaling problem for back-propagation algorithm. This algorithm works significantly when the training data are simple and it performance quickly drop off when problem complexity increase (because of complexity of the data or growing dimensionality of data). In order to overcome the back-propagation drawbacks, another method which is called evolutionary computation is extensively used as an alternative for training the neural network.

In the literatures, there are several methods for developing neural networks based on evolutionary algorithms. In [4], Asexual Reproduction Optimization (ARO) is applied in neural network training. The ARO algorithm used to find best initial weights for the network and then back-propagation algorithm is used for training network. Authors of [5], applied nature-inspired algorithm named Multi-Verse Optimizer (MVO) in training neural network. The method considered capabilities of MVO for utilization to find the optimal weights and biases, and high exploration rate in weight space. Paper [6] proposed a new algorithm based on evolutionary computing to develop concurrently topology and weights of neural network so that used a combination of Genetic Algorithm (GA) and Grammatical Evolution (GE). The GA algorithm is used for finding better weight while GE algorithm is used for choosing the network topology. In order to avoid over-fitting problems in the evolution process, a new adaptive penalty approach is applied to simplify neural network. Authors of [7] proposed a new classification method based on artificial bee colony (ABC) and particle swarm optimization (PSO) for recognizing abnormal brains images in the MRI scanning. The proposed method use feed-forward neural network architecture. In [8] presented an

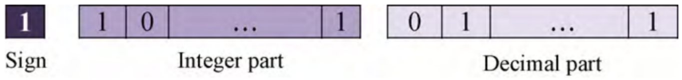

Fig. 1: A structure of ARO chromosome.

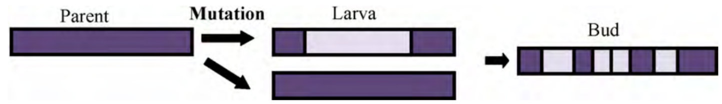

Fig. 2: The reproduction mechanism of ARO algorithm.

improved stochastic PSO (SPSO) to training neural network. The SPSO investigate completely in the solution space to find best global solution with hight confidence and it is also suitable for high dimension problems optimization. Paper [9] used feed-forward neural networks architecture and ABC algorithm for improving the network training in the classification tasks. Authors of [10] used a combination of ABC and Levenberq-Marquardt (LM) algorithm for training a feed-forward neural network in order to find optimal weights of network.

In this paper, it is proposed a feed-forward neural network training inspired by [001] that use an evolutionary algorithm named Asexual Reproduction Optimization (ARO) [11]. Proposed method used a modified evolutionary algorithm based on ARO in order to data classification problems. The main idea is to use a real representation of chromosome instead of binary one in order to adjusting weights of the network for increasing speed and accuracy of the network training.

II. Proposed Method

This work propose a new method for improving convergent speed and accuracy of neural network using Asexual Reproduction Optimization (ARO) algorithm. Applying the ARO algorithm to neural networks in order to training its weights, is proportionately straight-forward while in the back-propagation algorithm it is forward and backward process. However, the main goal here is data classification problems so that the trained network can be generalized to carry out well on unseen testing data. One of advantages of using ARO in neural network training is its free parameter that not need to be tuned, and also there is no mechanism for parent selection (selection operator). On the other hand, unlike other evolutionary algorithms which are population-based, the ARO algorithm is a single solution approaches can be convergence fast in the optimization process.

A. Asexual Reproduction Optimization

Asexual reproduction is a kind of reproduction that offspring occur from a single cell, and inherit the genes just from one parent. This form of reproduction happens for single-celled organisms like bacteria in which one cell yield two to four cells with equal chromosome number [11]. Inspired by asexual reproduction mechanism, ARO algorithm use a budding method in the way that each individual can be represented by a binary string. In ARO a vector $X = (x_1, x_2, ..., x_n); X \in R^n$ represents an individual where each x_i stands for a chromosome and consists of bits which is called genes. Base algorithm of ARO consist a binary string length of L in which it divided into three parts. First bit considered as sign of the chromosome, next part identify integer part by length of l_1 and last part represent decimal part of the chromosome by length of l_2. So, length of a chromosome in ARO is equal to $L = l_1 + l_2 + 1$ and length on an individual is equal to $n \times L$. Figure 1 shown a chromosome of ARO.

In the ARO, a solution in the search domain represented by an individual and the algorithm begins in a initiated random individual. Next, an offspring which is called bud reproduced by individual during reproduction mechanism (Figure 2). Finally, the parent and its offspring fall into a competition to survive for next generation based on fitness function which measures their performance. Consequently, if parent success in the competition then its offspring will be put aside.

In contrast, if the bud success in the competition then its parent will put aside and the bud is exchanged with its parent. The competition process repeats up to a defined iteration or when a ending condition is satisfied. In reproduction mechanism of ARO, larva is a substring which consists g bits ($g \sim uniform(1, L)$) that can be selected randomly. During merging mechanism, data of the larva and its parent will be combined and crossover operator happens during mutation. Probability of merging mechanism for exploration and exploitation rate defined as follows:

$$p = \frac{1}{1 + Ln(g)} \tag{1}$$

When reproduction mechanism is finished, fitness functions of parent and its bud will be compared and then parent or its bud which have best fitness value will be survived for next generation and another one will be removed.

B. Training Neural Networks by ARO

The feed-forward neural network comprises a set of nodes or neurons which are inked together. The ARO algorithm applied to a feed-forward neural network that consists of three layers and the ARO aim is to minimize the network cost function. A typical Mean Square Error (MSE) used as a cost function in the training phase. In the used feed-forward network, each weight considered as a chromosome in the ARO algorithm. By reproduction mechanism in the ARO algorithm, weights of the feed-forward network can be changed until reaching to optimum weight set. In contrast of [4], where in each individual consist of binary chromosome, in this work a real representation on a chromosome used in each individual so that for two real values x and y which are indicate a parent and larva, a linear combination can be taken like $\lambda x + (1-\lambda)y$ [11]. So, the chromosomes of each individual and weights in the feed-forward network are identical.

III. Experimental Results

Simulation of feed-forward neural network using ARO reported in this section and several experiments have been carried out to measure performance of proposed method. Accuracy and speed of data classification using proposed method have been compared with other evolutionary algorithms including genetic algorithms, simulated annealing and particle swarm optimization which are also applied to feed-forward network.

For the feed-forward network which consist of three layers, number of input layer is equal to feature number of dataset, number of hidden layer which is a hyper-parameter is selected by 10 fold cross-validation, and number of output layer is equal to dataset class. The sigmoid function is selected as activation function of the network nodes. Parameters of ARO algorithm and other evolutionary algorithms are considered equal for a fair comparison. Number of iteration is considered 500 and number of particle is 40 for all algorithms.

Dataset from UCI machine learning repository are shown in the table I for classification problem. The proposed algorithm is implemented in Matlab 2015a and runs on Linux OS system with 5.8 GB RAM and Intel Core i5 CPU.

TABLE I: The dataset used for classification from UCI machine learning repository.

Dataet	Sample	Features	Class
Australian	690	14	2
German	100	24	2
Madelon	4400	500	2
Waveform	500	21	3

Tables II to V shown results of comparison between proposed method and other methods. As it can be seen from the tables, accuracy and speed of proposed method are higher than others. From the tables, average accuracy measure on all dataset for proposed method on the test accuracy is 83.86 and outperform other methods.

TABLE II: Comparison results of evolutionary algorithms on Australian dataset.

Algorithm	Time	Train Accuracy	Test Accuracy
Real ARO	8.19	81.91	80.96
Binary ARO	10.19	81.42	80.26
PSO	15.09	79.26	78.24
GA	12.29	79.49	77.16
SA	251.36	76.25	75.37
BP	0.016	56.91	54.73

TABLE III: Comparison results of evolutionary algorithms on German dataset.

Algorithm	Time	Train Accuracy	Test Accuracy
Real ARO	4.64	85.91	84.89
Binary ARO	7.22	85.05	84.79
PSO	9.68	81.62	80.31
GA	18.21	82.63	79.80
SA	98.06	79.64	78.89
BP	0.09	65.21	62.65

During optimization process, however the back-propagation algorithm is faster than other evolutionary algorithm, by it fall into local minimum and then leads to premature convergence, so it performance is decreased.

TABLE IV: Comparison results of evolutionary algorithms on Madelon dataset.

Algorithm	Time	Train Accuracy	Test Accuracy
Binary ARO	9.82	87.54	86.94
Real ARO	13.18	86.42	85.65
PSO	21.93	83.85	81.64
GA	18.69	84.65	82.64
SA	483.36	79.25	77.05
BP	0.12	68.62	65.84

TABLE V: Comparison results of evolutionary algorithms on Waveform dataset.

Algorithm	Time	Train Accuracy	Test Accuracy
Real ARO	12.96	83.66	82.91
Binary ARO	16.61	82.95	82.16
PSO	20.33	81.32	79.39
GA	18.21	82.65	80.85
SA	418.03	77.57	75.24
BP	0.09	57.96	56.76

IV. Conclusion

An improved ARO algorithm have presented in this work for training neural network in data classification. Back-propagation algorithm which is based on gradient have many drawbacks such as falling into a local minimum. A feed-forward neural network architecture used in this work and a modified ARO algorithm applied for train its weights in the way that it use real representation of weights in each chromosome instead of binary for adjusting the network weights. The performance of the proposed method compared with other evolutionary algorithms including genetic algorithms, simulated annealing and particle swarm optimization so that it achieves best results in terms of speed and accuracy. For future work, the real version of ARO can be used for others neural network topologies and also in real time systems which are needed to learn fast.

References

[1] Lin Wang, Yi Zeng, and Tao Chen. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Applications*, 42(2):855–863, 2015.

[2] Ehsan Valian, Shahram Mohanna, and Saeed Tavakoli. Improved cuckoo search algorithm for feedforward neural network training. *International Journal of Artificial Intelligence & Applications*, 2(3):36–43, 2011.

[3] Nazri Mohd Nawi, Abdullah Khan, and Mohammad Zubair Rehman. A new back-propagation neural network optimized with cuckoo search algorithm. In *International Conference on Computational Science and Its Applications*, pages 413–426. Springer, Berlin, Heidelberg, 2013.

[4] Sajad Ahmadian and Ali Reza Khanteymoori. Training back propagation neural networks using asexual reproduction optimization. In *Information and Knowledge Technology (IKT), 2015 7th Conference on*, pages 1–6. IEEE, 2015.

[5] Hossam Faris, Ibrahim Aljarah, and Seyedali Mirjalili. Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence*, 45(2):322–332, 2016.

[6] Fardin Ahmadizar, Khabat Soltanian, Fardin AkhlaghianTab, and Ioannis Tsoulos. Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. *Engineering Applications of Artificial Intelligence*, 39:1–13, 2015.

[7] Shuihua Wang, Yudong Zhang, Zhengchao Dong, Sidan Du, Genlin Ji, Jie Yan, Jiquan Yang, Qiong Wang, Chunmei Feng, and Preetha Phillips. Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection. *International Journal of Imaging Systems and Technology*, 25(2):153–164, 2015.

[8] Xin Chen and Yangmin Li. Neural network training using stochastic PSO. In *Neural Information Processing*, pages 1051–1060. Springer, 2006.

[9] John A Bullinaria and Khulood AlYahya. Artificial bee colony training of neural networks. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*, pages 191–201. Springer, 2014.

[10] Celal Ozturk and Dervis Karaboga. Hybrid artificial bee colony algorithm for neural network training. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 84–88. IEEE, 2011.

[11] Alireza Farasat, Mohammad B Menhaj, Taha Mansouri, and Mohammad Reza Sadeghi Moghadam. ARO: a new model-free optimization algorithm inspired from asexual reproduction. *Applied Soft Computing*, 10(4):1284–1292, 2010.