

An ANN Model to Classify Multinomial Datasets with Optimized Target Using Particle Swarm Optimization Technique

Nilamadhab Dash, Rojalina Priyadarshini and Rachita Misra

Abstract In this paper we propose a particle swarm based back propagation neural network model which uses an optimized target to maximize the classification accuracy of the classifier. By using Particle swarm optimization technique an optimized target for each class was determined and there after the artificial neural network is used to classify the data using these targets. For this, some of the bench mark classification datasets are used, which are taken from UCI learning repository. An extensive experimental study has been carried out to compare the proposed method and existing method on the same datasets and a comparative analysis is done by taking several parameters like percentage of accuracy, time of response and complexity of the algorithm. During this study we have examined the performance improvement of the proposed PSO and BPN combined approach over the conventional BPN approach to generate classification inferences from the training and testing results.

Keywords Multinomial classification · Back propagation neural network · Normalization · Particle swarm optimization

1 Introduction

Data Classification and prediction are two of the prime tasks in Data mining. They continue to play a vital role in the area of data processing, financial analysis, stock market predictions, weather forecasting, disease predictions, pattern recognition, bioinformatics, image processing... etc. [1]. Clustering and classifications in Data

N. Dash (✉) · R. Priyadarshini · R. Misra
Department of IT, C.V. Raman College of Engineering, Bhubaneswar, Odisha, India
e-mail: nilamadhab04@gmail.com

R. Priyadarshini
e-mail: priyadarshini.rojalina@gmail.com

R. Misra
e-mail: rachita03@yahoo.com

Mining are applied in various domains to give a meaning to the available data and also give some useful prediction results which can be applied to many of the crucial problem areas of the real world. Classification is the task of dividing different data from their known features to a particular group. Depending on the number of classes the classifier is a binary or multinomial classifier. Binary classifiers are applied to the data having two classes. In machine learning, multiclass or multinomial classification is the problem of classifying the given instances into more than two classes. While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies. Multiclass classification should not be confused with multi-label classification, where multiple classes are to be predicted for each problem instance. In this paper we have used artificial neural network as a classifier but tried to improvise the model by using particle swarm optimization technique with it. For measuring the performance the classification accuracy is taken as the prime criteria. The architectural complexity is taken care of by optimizing the number of nodes in the hidden layer. In this work we have taken some classification datasets from the UCI learning repository. They are: Iris, seeds and wine datasets.

The rest of the paper is sequentially arranged in the following order. Section 2 comprises the details of the dataset and pre-processing of the dataset. In Sect. 3, Particle Swarm optimization is briefly described, which is used to model the target output of the classifier. Section 4 describes the results, which contains the evaluation of the proposed model on the basis of different criteria. Finally Sect. 5 details the conclusion and future work.

2 Datasets and Preprocessing

In this work we have used 3 bench mark datasets, taken from UCI learning repository for verification and validation of the proposed model [2]. A brief of the datasets are as follows.

2.1 *Iris Flower*

The dataset consists of 150 samples which consists of a set of Iris flowers, where the goal is to predict three classes, setosa, versicolor and virginica. Based on sepal (green covering) length and width, and petal (the flower part) length and width.

2.2 *Seed*

This dataset consist of experimental high quality visualization of the internal kernel structure which is detected using a soft X-ray technique. The images were recorded

on 13×18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agro physics of the Polish academy of science Lubin. The data set can be used for the tasks of classification and cluster analysis. The dataset contains the following feature attributes, (1) area A, (2) perimeter P, (3) compactness $C_{pi} * A/P^2$, (4) length of kernel, (5) width of kernel, (6) asymmetry coefficient (7) length of kernel groove. All of these parameters were real-valued continuous.

2.3 Wine

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The data consists of (1) Alcohol (2) Malic acid (3) Ash (4) Alcalinity of ash (5) Magnesium, (6) Total phenols, (7) Flavanoids (8) Nonflavanoid phenols, (9) Proanthocyanins, (10) Color intensity, (11) Hue (12) OD280/OD315 of diluted wines, (13) Proline. All attributes are continuous.

2.4 Normalization

Normalization of input data is used for ranging the values to fall within an acceptable scope, and range [1]. These procedures help in obtaining faster and efficient training. If the neurons have nonlinear transfer functions (whose output range is from -1 to 1 or 0 to 1), the data is normalized for efficiency. As our outputs are falling within these ranges, each feature in each dataset is normalized using column normalization. The normalized data are used as the inputs to the machines.

3 Basic Principles of PSO

Particle swarm optimization (PSO) is a population based stochastic search and optimization technique, which was introduced by Kennedy and Eberhart in 1995 [3]. It is a multi-objective optimization method to find optimal solution to the problems having multiple objectives [4]. It is a technique, loosely modeled on the collective behavior of groups, such as flocks of birds and schools of fish. It mainly shows a natural behavior of a group of objects, search for some target (e.g., food). It is computer simulation of the coordinated behavior of a swarm of particles moving to achieve a common goal [5]. The goal is to reach to the global optimum

of some multidimensional and possibly nonlinear function or system [6]. PSO's working principle is quiet similar to any other evolutionary algorithms like genetic algorithm. Initially particles are randomly distributed over the search space. So each particle gets a virtual position that represents a possible potential solution to optimization problem. It is an iterative process where in each iteration every particle moves to a new position, navigating through the entire search space. Each particle keeps track of its position in the search space and its best solution so far achieved. The personal best value is called as pBest and the ultimate goal is to find the global best called as gBest [7].

The standard PSO algorithm broadly consists of the following computational steps

- (i) Initialize particles with random positions and velocities;
 For each value of k [where 'k' represents the number of particles]
 Do
 - a) Evaluate fitness of each particle's position (p)
 - b) If fitness(p) better than fitness(pbest) then pbest = p
 - c) Set best of pBests as gBest
 - d) Update particles velocity and position
- End of for
- (ii) Stop: giving gBest, optimal solution.

Here, a particle refers to a potential solution to a problem in d-dimensional design space with k particles. Each particle is characterized by Position vector..... $x_i(t)$ and Velocity vector..... $v_i(t)$

Each particle has Individual knowledge pbest, its own best-so-far position, Social knowledge gbest, pbest of its best neighbour. The equations for velocity and position updates are given below

$$v(t + 1) = (w * v(t)) + (c1 * r1 * (p(t) - x(t))) + (c2 * r2 * (g(t) - x(t))) \quad (1)$$

$$x(t + 1) = x(t) + v(t + 1) \quad (2)$$

The first equation updates a particle's velocity. The term $v(t + 1)$ is the velocity at time $t + 1$. The new velocity depends on three terms. The first term is $w * v(t)$. The w factor is called the inertia weight and is just a constant between 0 and 1. Here the value of w is taken as 0.73, and $v(t)$ is the current velocity at time t. The second term

is $c1 * r1 * (p(t) - x(t))$. The $c1$ factor is a constant called the cognitive (or personal or local) weight. The $r1$ factor is a random variable in the range $[0, 1]$ —that is, greater than or equal to 0 and strictly less than 1. The $p(t)$ vector value is the particle’s best position found so far. The $x(t)$ vector value is the particle’s current position. The third term in the velocity update equation is $(c2 * r2 * (g(t) - x(t)))$. The $c2$ factor is a constant called the social, or global, weight. The $r2$ factor is a random variable in the range $[0, 1]$. The $g(t)$ vector value is the best known position found by any particle in the swarm so far. Once the new velocity, $v(t + 1)$ has been determined, it is used to compute the new particle position $x(t + 1)$.

In case of a neural network, a particle’s position represents the values for the network’s weights and biases. Here, the goal is to find a position/weights so that the network generates computed outputs that match the outputs of the training data.

3.1 PSO for Target Optimization

In case of supervised learning, during the training process of neural network, three parameters are mostly required, the input, weight and target [8]. In practical approach the target of a neural network is either randomly chosen or depends on the feature of input dataset. e.g., if the input dataset has three different classes, three target outputs are generated. In case of unsupervised learning, the target does not exist at all, rather it is needed to be explored by the network itself. If any of the previously said problem occurs, the efficiency of the neural network reduces drastically due to the following reasons

1. The neural network may take longer time to get trained as randomly chosen target outputs may not be the optimal one.
2. Extensive computations are required as additional mapping is required in order to match up with the input values to the randomly chosen target values.
3. Usually for classification tasks, the number of output neurons in a neural network depends on the number of classes present in the dataset [9], whereas the number of input neurons depends on the number of features per input data. This particular architecture fails to maintain the relevance between the input and the randomly chosen target. (Feature to number of class mapping is done instead of feature to feature mapping which is more accurate). These problems are addressed by using the proposed PSO based technique, which generates a nearly optimized target by analyzing the input dataset of the neural network. The steps for designing the classifier is diagrammatically shown in Fig. 1.

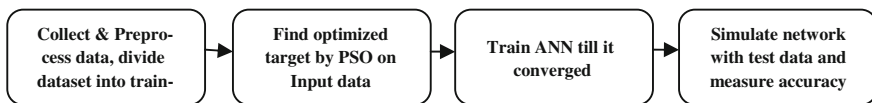


Fig. 1 Proposed model of PSO based classifier

3.2 Training by BPNN and Simulation

Training in an artificial neural network is necessary to make the classifier learn by itself. It is the ability to approximate the behavior adaptively from the training data while generalization is the ability to predict the training data [9]. A BPN network learns by example. It gives us the desired output for a particular input if, provided with the known input by changing the network's weights so that, when training is finished, it will give the correct output. The change in weight takes place according to the error produced. The data sets are divided into two parts. i.e., training set (known data) and testing set (Unknown data) which is not used in the training process, and is used to verify the machine and then we have simulated our results with these datasets [10].

The training process of this work follows the following steps

1. On the input data, particle swarm optimization is applied to find out the optimized target. In other words the optimized inputs for the classifier are found out with the help of PSO technique.
2. The training data are prepared by normalizing the input and the optimized target that range from 0 to 1.
3. The artificial neural network is trained by previously got inputs till it gets converged.

The convergence criterion for the network is taken as minimum error condition.

Figures 2 and 3 show the convergence of different networks used to train the said datasets.

Fig. 2 Convergence graph of iris

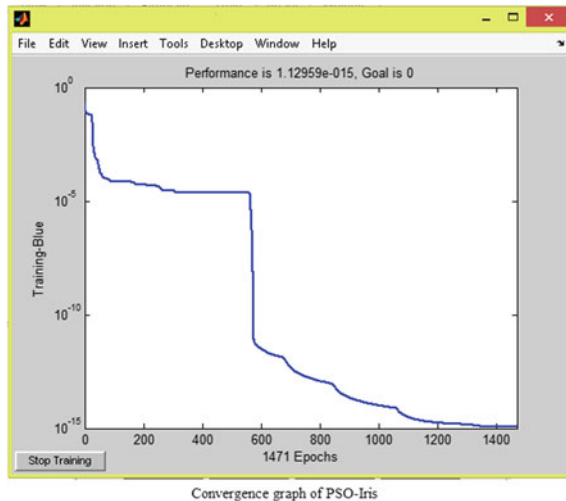
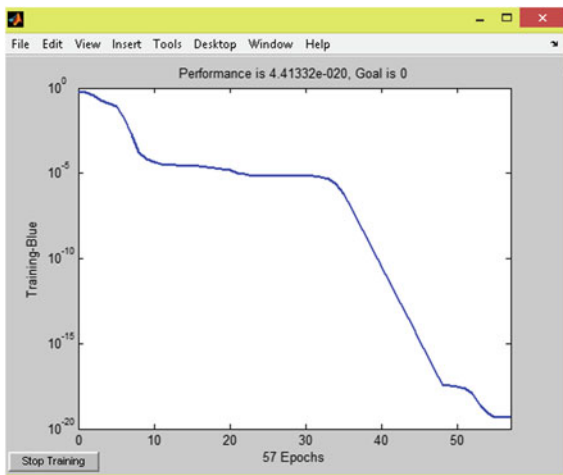


Fig. 3 Convergence graph of seed



In the proposed work the architecture of the network varies according to the dataset. This is due to the fact that, the output of the neural network depends on the number of features present in the dataset. The output of the neural network depends on the number of features present in the dataset. For example, a $[4 * 5 * 4]$ network having 4 input data is used to train the iris dataset containing four features per data.

4 Results and Observations

The process of classification was carried out on the previously mentioned datasets by taking the conventional ANN with back propagation learning algorithm as well as the proposed approach. In both the cases the classification accuracy was taken as the most vital factor for performance evaluation. Number of misclassification is calculated by measuring the Euclidean distance between the target and actual output. Percentage of misclassification is ratio of incorrectly predicted class and total number of data present in the testing samples multiplied by 100. Table 1 shows the overall comparison between the PSO-BPNN and BPNN approach. It is clear from the result that the proposed approach shows significant improvement over BPNN. Tables 2, 3 and 4 give the simulation accuracy of the proposed work for different datasets up to 4 decimal places.

Table 1 Performance comparison of back propagation neural network and the proposed approach

Dataset	Number of classes	Number of samples	Number of features	Number of training samples	Number of testing samples	Number of misclassifications		Classification accuracy	
						BPNN	PSO-BPNN	BPNN (%)	PSO-BPNN (%)
Iris	3	150	4	120	30	1	11	78	98
Seed	3	210	7	150	60	9	17	72	85
Wine	3	178	13	150	28	1	6	79	98

Table 2 Simulation accuracy of wine dataset

No. of classes	Number of simulation result accurate up to				Total no. of test samples per class
	4 decimal places	3 decimal places	2 decimal places	1 decimal places	
Class 1	09	09	09	09	09
Class 2	10	10	11	11	11
Class 3	08	08	08	08	08
Percentage of accuracy	97 %	97 %	100 %	100 %	Total = 28

Table 3 Simulation accuracy of seed dataset

No. of classes	Number of simulation result accurate up to				Total no. of test samples
	4 decimal places	3 decimal places	2 decimal places	1 decimal places	
Class 1	16	16	17	20	20
Class 2	15	15	16	20	20
Class 3	17	18	18	20	20
Percentage of accuracy	80 %	83 %	85 %	100 %	Total = 60

Table 4 Simulation accuracy of iris dataset

No. of classes	Number of simulation result accurate up to				Total no. of test samples
	4 decimal places	3 decimal places	2 decimal places	1 decimal places	
Class 1	10	10	10	10	10
Class 2	10	10	10	10	10
Class 3	09	09	09	09	10
Accuracy %	98 %	98 %	98 %	98 %	Total = 30

5 Conclusion and Future Work

Though the role of back propagation neural network is inevitable in the field of classification, there is a certain need to assess its efficiency in terms of learning time, simulation accuracy and flexibility. In the proposed work we have tried to improve the classification accuracy, and got some promising results which verify that the proposed method shows a remarkable improvement over back propagation machine classifier alone. The particle swarm optimization technique also played a vital role to provide the optimized target that made the learning process easier and efficient. Considering the inspiring results obtained from the proposed work the future objectives are (1) to apply the proposed method on some real life problems with

some benchmark datasets mainly in the area of computational biology and bioinformatics (2) to reduce the architectural and computational complexity of the network in terms of number of hidden neurons along with training and simulation time.

References

1. Dehuri, S., Cho, S.B.: A Comprehensive survey on functional link neural networks and adaptive PSO-BP learning for CFLNN. *Neural Comput. Appl.* (2009)
2. Bache, K., Lichman, M.: UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. University of California, School of Information and Computer Science, Irvine (2013)
3. Ince, T., Kiranyaz, S., Pulkkinen, J., Gabbouj, M.: Evaluation of global and local training techniques over feed-forward neural network architecture spaces for computer-aided medical diagnosis. *Expert Syst. Appl.* **13**, 8450–8461 (2010)
4. Dehuri, S., Chob, B.: Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: a review and state-of-the-art. *Comput. Sci. Rev.* **3**, 19–40 (2009)
5. Dehuri, S., Royb, R., Cho, S.B., Ghosh, A.: An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification. *J. Syst. Softw.* **85**, 1333–1345 (2012)
6. Kennedy, J., Eberhart, R.: Particle swarm optimization. *IEEE Trans.* 1942–1948 (1995)
7. Devi, S., Jagadev, A.K., Dehuri, S., Mall, R.: Knowledge discovery from bio-medical data using a hybrid PSO/Bayesian classifier. *Tech. Int. J. Comput. Sci. Commun. Technol.* **2**(1), 364–371 (2009)
8. Zhang, G.P.: Neural networks for classification: a survey. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **30**(4), 451–463 (2013)
9. Haykin, S.: *Neural networks—a comprehensive foundation*. Prentice Hall, Englewood Cliffs (1999)
10. Priyadarshini, R., Dash, N., Swarnkar, T., Misra, R.: Functional analysis of artificial neural network for dataset classification. *Int. J. Comput. Commun. Technol.* **1**(2–4), 49–54 (2010)
11. Zhang, G.P.: Avoiding pitfalls in neural network research. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **37**(1), 3–163 (2007)
12. van der Merwe, D.W., Engelbret, A.P.: Data clustering using particle swarm optimization. **1**, 215–220 (2003) (IEEE)