Genetic algorithm for neural network architecture optimization

Conference Paper · May 2016

DOI: 10.1109/GOL.2016.7731699

4 authors, including:

Mohammed Amine Janati Idrissi
Faculté des Sciences et Techniques Fès
7 PUBLICATIONS **71** CITATIONS

SEE PROFILE

Hassan Ramchoun
Faculty of sciences and technology fez
11 PUBLICATIONS **74** CITATIONS

SEE PROFILE

Youssef Ghanou
Université Moulay Ismail
41 PUBLICATIONS **200** CITATIONS

SEE PROFILE

Genetic Algorithm for Neural Network Architecture Optimization

JANATI IDRISSI
Mohammed Amine
Modelling and Scientific
Computing Laboratory,
Faculty of Sciences and
Technology, Sidi Mohammed
ben Abdellah University Fez,
MOROCCO
medamine.janatidrissi@gmail.com

RAMCHOUN Hassan
Modelling and Scientific
Computing Laboratory,
Faculty of Sciences and
Technology, Sidi Mohammed
ben Abdellah University
Fez, MOROCCO
ramchounhassan@gmail.com

GHANOU Youssef
Team TIM,
High School of Technology,
Moulay Ismail University,
Meknes, Morocco
youssefghanou@yahoo.fr

ETTAOUIL Mohamed
Modelling and Scientific
Computing Laboratory,
Faculty of Sciences and
Technology,
Sidi Mohammed ben Abdellah
University
Fez, MOROCCO
mohamedettaouil@yahoo.fr

Abstract— **The optimization of architecture of feed-forward neural networks is a complex task of high importance in supervised learning because it has a great impact on the convergence of learning methods. In this paper, we propose a multi-objective mathematical formulation in order to determine the optimal number of hidden layers, the number of neurons in each layer and good values of weights. We will solve our mathematical modeling using a hybridation of the famous genetic algorithm and the back-prop training algorithm. For evaluating our approach, we apply it to benchmark classification problems data-iris, seed, and wine. The obtained results demonstrated the effectiveness of the proposed approach.**

Keywords—Artificial neural networks, Neural network architecture, Genetic algorithms, Supervised Training, Feed forward neural network, Multi-objective optimization.

I. INTRODUCTION

ANN is a computational network, which attempt to simulate, in a gross manner, the networks of nerve cell (neurons) of the biological (human or animal) central nervous system (Graupe, 2007). ANN have interesting properties, such as adaptability, ability to learn from examples, and the ability to generalize. With these very interesting properties ANNs are one of the powerful tools effectively employed in various fields (Anderson 2006) such as time series prediction, classification, pattern recognition, system identification and control, function approximation, signal processing and so on.

Learning or training of ANN is equivalent to finding the values of all weights such that the desired output is generated to corresponding input, it can be viewed as the minimization of error function computed by the difference between the output of the network and the desired output of a training observations set [4]

However, some critical points of ANN structure should be carefully analyzed. The definition of the appropriate network architecture such as number of hidden layers and neurons in each hidden layer is a crucial factor. The number of neurons in the input and output layers is known, that is to say, determined by the given problem. However, this is not the case for the number of neurons in the hidden layers. In general, the choice of architecture is done empirically, so that it is satisfactory according to some criteria, for example, minimization of a quadratic cost.

The architecture of a multilayer neural network can be formulated as a mathematical model under constraints [7][8].where each solution is an architecture. There are several optimization methods to solve this problem, like tabu search, simulated annealing and the famous genetic algorithm.

In this work, we formulate the architecture design as a multi-objective optimization problem, where each solution represents an architecture. In addition, for solving this problem, we use the famous genetic algorithm.

For evaluating the proposed approach, we have used classical benchmark classification problems (Data Iris, Seed and Wine) obtained from UCI Repository of Machine Learning Databases [1].

The contribution is structured as follows. Section 2 describes the genetic algorithm. Then, we start describing 3. Proposed model in Section 3. Section 4 deals with solving the obtained model. In section 5, we present results and discussions. Finally, Section 6 is devoted to conclusions.

II. GENETIC ALGORITHM

GA is a global search procedure that searches from one population of points to another. It belongs to a class of probabilistic methods called “evolutionary algorithms” based on the principles selection and mutation. GA was introduced by J.HOLLAND [5] that it’s based on natural evolution theory of Darwin. This algorithm has shown its performance in fairly difficult problems [10][11].They use a vocabulary similar to that of natural genetics. However, natural processes to which they refer are much more complex. So, we talk about individual in a population. The individual is represented by a chromosome composed of genes, which contain the hereditary characteristics of the individual. The principles of selection, crossover, and mutation are inspired by the natural processes of the same name.

For a given optimization problem, an individual represents a point in the research space. We associate to it the value of the criterion to be optimized, which is its adaptation. Then we generate iteratively populations of individuals on which selection crossover and mutation process are applied. The selection aims to promote the best elements of the population for the given criterion (most suitable), crossover and mutation ensure the exploration of the state space.

It starts by generating a random population of individuals. To pass from one generation to the next generation, the following operations are performed. First, the population is reproduced by selection, where good individuals reproduce better than the bad. Then, applying a cross to the pairs of individuals (parents) of a proportion of the population (probability Pc) to produce new (child). A mutation operator is also applied to a certain proportion of the population (probability Pm, usually much smaller than Pc). Finally, new individuals are evaluated and integrated into the population of the next generation. Several judgment criteria of the algorithm are possible: the number of generations can be fixed a priori (time constant) where the algorithm can be stopped when the evolution of the population is not fast enough. For using a genetic algorithm to an optimization problem, we must have an individual coding principle, a generation mechanism of the initial population and operators to diversify the population over generations and explore the search space.

III. PROPOSED MODEL

The problem of neural architectures optimization is to find the optimal number of hidden layers in the ANN, the number of neurons within each layer, in order to maximize the performance of artificial neural networks [2][3][6] [9]. In this work, we assign to each neuron a binary variable, which takes the value 1 if the neuron is active and zero otherwise and a binary variable for each hidden layer, which take the value 1 if we use this hidden layer and zero otherwise.

Notation

N: Number of hidden layers

n_i: Number of neurons in i^{th} hidden layer.

X : Input data of neural network.

Y : Calculated output of neural network.

d : Desired output.

W : Weights.

F : Activation function.

U_i: Binary variable, which take the value 1 if the i^{th} hidden layer is used, and zero otherwise. where $i = 1, \dots, N$, $j = 1, \dots, n_i$.

$$U = (U_1, \dots, U_N)\ , V = (V_{ij})$$

V_{ij}: Binary variable, which take the value 1 if the i^{th} neuron of the i^{th} hidden layer is used, and zero otherwise where $j = 1, \dots, n_i$. $i = 1, \dots, N$

A. Objectives functions

Our mathematical model is bi-objective, the first is to optimize the network architecture by looking for the smallest possible number of hidden layers, and in each layer a minimum number of neurons. Secondly minimize the mean square error of the network.

$$\min \begin{cases} \|F(X, W, U, V) - Y\| \\ \dfrac{\sum_{i=1}^{N} U_i}{M_U} + \dfrac{\sum_{i=1}^{N}\sum_{j=1}^{n_i} V_{ij}}{M_V} \end{cases} \qquad (1)$$

Where M_U and M_V are the maximum of used hidden layer and neurons respectively.

B. Constraints

The first Constraint guarantee the existence of at least one hidden layer

$$\sum_{i=1}^{N} U_i \geq 1 \qquad (2)$$

The second constraint models the fact that if we do not use a hidden layer, thus automatically, not using their neurons.

$$\sum_{j=1}^{n_i} V_{ij} \leq MU_i \qquad i = 1, \dots, N \qquad (3)$$

Where M is a big positive scalar.

Remark:
In the case where, for a i_0 $V_{i_0 j} = 0, for\ all\ j\ in\{1, \dots, n_0\}$ we have at once $U_{i_0} = 0$.

IV. SOLVING THE OBTAINED MODEL

In this section, we use the genetic algorithm to solve the architecture optimization problem. To this end, we have coded individual by tree chromosomes; moreover, the fitness of each individual depends on the value of the objective function.

Our mathematical model has two objective functions, which they are not on the same scale, so we use the scalarisation method to transform our multi objective problem into a problem with a single objective function, so the new objective function is:

$$\alpha_1 \frac{\|F(X,W,U,V)-Y\|}{f^*} + \alpha_2 \frac{\frac{\sum_{i=1}^{N} U_i}{M_U} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i} V_{ij}}{M_V}}{g^*} \qquad (4)$$

Where $f^* = \max(\|F(X,W,U,V) - Y\|)$, $\alpha_1 + \alpha_2 = 1$.

In general, the determination of f^* and g^* for a multi objective problems is very expensive, but in our case g^* is a maximum number of neurons supposed sufficient for a learning dataset. Moreover, since the output of a neuron is binary, so the maximum square error for a neuron is 1, therefore the maximum square error of a neural network (f^*) is the number of neurons of the output layer.

Fitness function

The fitness function is the inverse of the new objective function (4). The higher fitness, the lower the objective function is it.

V. EXPERIMENTAL RESULT

For evaluating the performance of our approach, we choose tree known problems from the UCI repository of machine learning databases [1]: Databases "Iris" , "Seeds" and "Wine". We developed the programs in C++ environment and tested in the Intel (R) Core (TM) i3-2370M CPU @ 2.4GHz (CPUs) and 4 Go of RAM.

Let NLB (Number the learning base), CC (Data Correctly Classified) and MC: Data Misclassified.

In experiments, for each topology, ten runs were performed with 20 distinct random weight initializations. Moreover, we have trained a fully connected multilayer perceptron by using a gradient descent with momentum backpropagation. The learning rate was set to 0.1, and the momentum term to 0.6.

We realize a series of experiments by reducing the size of neural networks, where we have test our methods for instance of the database data Iris Table 1, Seeds, Table 2, and Wine Table 3.

We observe that the proposed method allows to properly classifying the training data. The results obtained by our approach are good, because most of the training data on the test sets were correctly classified.

Table 1. Accuracy Test Using the Proposed Approach for Database Iris

	NLB	CC	MC	Accuracy (%)
Setosa	25	24	1	96%
Versicolor	25	25	0	100%
Virginica	25	23	2	92%
Total	75	72	3	96%

Table 2. Accuracy Test Using the Proposed Approach for Database Seeds

	NLB	CC	MC	Accuracy (%)
Canadian	45	44	1	97.78%
Kama	45	42	3	93.33%
Rosa	45	44	1	97.78%
Total	135	130	5	96.3%

Table 3. Accuracy Test Using the Proposed Approach for Database Wine

	NLB	CC	MC	Accuracy (%)
Class 1	30	29	1	96.66%
Class 2	36	34	2	94.44%
Class 3	24	23	1	95.83%
Total	90	86	4	95.55%

VI. CONCLUSION

In this work, we have analyzed the feed-forward neural networks architecture optimization problem with the use of a methodology entirely based on the genetic algorithm and back prop training algorithm. This methodology was inspired by a similar process described in [8], where the authors had optimize only the number of hidden layers and in [7] the authors had optimize the number of neurons in each layer without optimizing the number of layers.

VII. REFERENCES

[1] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases. Irvine, CA: Dept. Inf. Comput. Sci., Univ. California,1998.

[2] E. Egrioglu, C, Hakam Aladag, S. Gunay, "A new model selection straegy in artificiel neural networks", Applied Mathematics and Computation (195), 2008, pp. 591-597.

[3] H.Ramchoun, M.A. Janati Idrissi, Y.Ghanou, M. Ettaouil " Multilayer Perceptron: Architecture Optimization and Training" International Journal of Artificial Intelligence and Interactive Multimidia Vol. 1 N0 1 (2016).

[4] INCI CABAR, SIRMA YAVUZ2, OSMAN EROL1, 'Robot Mapping and Map Optimization Using Genetic Algorithms and Artificial Neural Networks', WSEAS TRANSACTIONS on COMPUTERS, Issue 7, July 2008.

[5] J. Holland, 'Adaptation in natural and artificial systems'. Ann Arbor, MI: University of Michigan Press, 1992.

[6] M. Duran, I. E. Grossmann. "An outer approximation algorithm for a class of mixed integer nonlinear programs". Mathematical Programming, 1986, pp. 307-339.

[7] M. Ettaouil and Y. Ghanou, "Neural architectures optimization and Genetic algorithms", Wseas Transactions On Computer, Issue 3, Volume 8, 2009, pp. 526-537.

[8] M. Ettaouil M. Lazaar and Y. Ghanou "Architecture optimization model for the multilayer perceptron and clustering" Journal of

Theoretical and Applied Information Technology 10 January 2013. Vol. 47 No.1.

[9] M. Ettaouil, Y.Ghanou, K. Elmoutaouakil, M. Lazaar, “A New Architecture Optimization Model for the Kohonen Networks and Clustering”, Journal of Advanced Research in Computer Science (JARCS), Volume 3, Issue1, 2011, pp. 14 - 32.

[10] R.E.Dorsey,W.J. Mayer, Genetic algorithms for estimation problems with multiple optima, non-differentiability ,and other irregular features, Journal of Business and Economic Statistics 13 (1) (1995) 53-66.

[11] Y. Chang, C. Low, ‘Optimization of a passive harmonic filter based on the neural-genetic algorithm with Fuzzy logic for a steel manufacturing plant’, Expert systems with Application, 2059-2070,2008.