MEMETIC ALGORITHMS FOR NEURAL NETWORK TRAINING ON MEDICAL DATA

Y.G. Petalas and M.N. Vrahatis

Department of Mathematics, University of Patras, GR–26110 Patras, Greece,
University of Patras Artificial Intelligence Research Center(UPAIRC)

email:{petalas,vrahatis}@math.upatras.gr

Abstract: During the last decade Computational Intelligent methods have been employed to address problems arising in the field of Biomedicine. Artificial Neural Networks constitute one of the most widely used Computational Intelligence methods. The supervised training of Artificial Neural Networks amounts to the global minimization of the network error function. Memetic Algorithms (MAs) comprise a family of population–based, heuristic, search algorithms designed to perform global optimization. MAs have been successfully applied in difficult optimization problems with considerable success. In this contribution, we propose a Memetic Algorithm as a neural network training method. The performance of the proposed algorithm is evaluated on problems from the field of Biomedicine.

1 Introduction

Biomedical engineering is an interdisciplinary field which not only integrates fundamental scientific knowledge with medicine and biology, but also stimulates the interaction of engineers from all the traditional disciplines. From its early days it focused on the development of medical devices. Nowadays, biomedical engineering has developed into a field of extreme breadth and diversity. Biomedical engineers may work to develop improved materials for implantable artificial organs, write software to analyze medical data, or study biological fluid dynamics.

In this work we focus on the analysis of medical data using computational intelligent techniques for diagnosis. The medical data were obtained from [3]. They consist of two datasets concerning benign and malignant breast cancer and the angiographic disease status of the heart. Both of them are classification tasks.

To address these classification tasks we employ Artificial Neural Networks. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the information processing functions of biological nervous systems, such as the brain. The key element of this paradigm is the novel structure of the system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

An important issue in Neural Network training is the global minimization of the error function, i.e. the minimization of the discrepancy between the desired and the computed output of the Network. The aim is to determine a set of weights which minimizes this error function. Training Neural Networks is a difficult task because the error function is complex and further it is characterized by a multitude of local minima. We propose the use of a new Memetic Algorithm for Neural Network training. MAs are population-based heuristic search methods for global optimization.

The rest of the paper is organized as follows: Section 2 is devoted to MAs, in section 3 we give a short description of our method, in Section 4 we present experimental results, and the paper ends with concluding remarks in Section 5.

2 Memetic Algorithms

Memetic Algorithms are population-based heuristic search algorithms for global optimization.

MAs are inspired by models of adaptation in natural systems that combine evolutionary adaptation of individuals with individual learning within a lifetime. Individual learning in this context is usually a local search method. MAs which use local search are called Local Based Memetic Algorithms (LS-based MAs) [1] and they have been successfully applied in difficult optimization problems, and especially in combinatorical optimization problems. We give below a general description of an LS-based MA:

Begin
Population Initialization
LocalSearch
Evaluation
Repeat

Recombination
Mutation
LocalSearch
Evaluation
Selection

Until termination criterion is satisfied
Return best solution
End

Next we explain the pseudocode of the LS-based MA. First the population is initialized usually with some random values. LocalSearch is the Local Search method that is executed by an individual. The Evaluation function assigns to the individual the value of the objective function of the optimization problem under consideration. The Recombination function recombines two selected individuals from the population to produce a new individual. The Mutation function applies a mutation operator to a selected individual from the population. The Selection function chooses which individuals will continue to the next generation. The termination criterion can be a number of generations, or the value of an error goal that has to be reached.

In a previous work [2] we have used an LS-based MA which used the Differential Evolution (DE) algorithm [5] as the global search method and Random Walk with Direction Exploitation as the Local Search Method [4]. In this contribution, we propose an LS-based MA which uses the DE algorithm as the global search method and in each iteration we apply as a local search method a local version of the DE algorithm.

3 Method Description

In the following subsections we will give a short description of the DE algorithm and the local version of the DE algorithm.

3.1 Differential Evolution

DE is a population based algorithm for global optimization. It has been successfully applied in various optimization problems. The ith individual of the population at the gth generation is denoted as u_g^i. A brief description of the workings of the algorithm is provided immediately below.

Step 1. Initialize the individuals of the population with random values. Set values to the mutation factor F and the crossover factor CR. These constants take values in the interval $[0, 1]$.

Step 2. (Mutation Step) Mutate each individual u_g^i (called the target individual), of the population to form a trial individual v_{g+1}^i, by applying one of the following operators,

$$v_{g+1}^i = u_g^{r1} + F(u_g^{r2} - u_g^{r3}) \qquad \textbf{DE1}$$
$$v_{g+1}^i = u_g^{r1} + F(u_g^{r1} - u_g^{r2}) \qquad \textbf{DE2}$$
$$v_{g+1}^i = u_g^{\text{best}} + F(u_g^{r1} - u_g^{r2}) \qquad \textbf{DE3}$$
$$v_{g+1}^i = u_g^i + F(u_g^i - u_g^{\text{best}}) + +F(u_g^{r1} - u_g^{r2}) \qquad \textbf{DE4}$$
$$v_{g+1}^i = u_g^{\text{best}} + F(u_g^{r1} - u_g^{r2}) + +F(u_g^{r3} - u_g^{r4}) \qquad \textbf{DE5}$$

where $r1, r2, r3, r4$ are random integers satisfying $r1 \neq r2 \neq r3 \neq r4 \neq i \neq best$. The index $best$ is used to represent the individual with the lowest function value in the current population.

Step 3. (Crossover step) For each element of the trial individual, v_{g+1}^i, obtain a random value r in the interval $[0, 1]$. If $r \leqslant CR$, $u_{g+1}^i = v_{g+1}^i$, else $u_{g+1}^i = u_g^i$.

Step 4. (Selection step) For each individual of the population u_{g+1}^i evaluate its function value. If this value is lower than that of the target individual u_g^i, then the individual u_{g+1}^i replaces the target individual in the next generation. Otherwise the target individual is retained in the next generation. If the termination criterion is not satisfied go to Step 2.

3.2 Local Differential Evolution

We use as a local search method a local version of the DE algorithm. Assume u^i to be an individual of the population which has been selected to execute a local search method for some iterations. A number of random vectors is generated and added to u^i to produce new temporary individuals. Then u^i acts as the target vector and all the steps of the DE algorithm are applied using the produced individuals.

3.3 The proposed method

The resulting memetic algorithm combining the aforementioned methods is,

Step 1. Initialize the individuals of the population with random values. Set values to the mutation factor F and the crossover factor *CR*. These constants take values in the interval $[0, 1]$.

Step 2. For each individual u_i of the population perform a local DE algorithm for some iterations.

Step 3. Perform the mutation step of the DE.

Step 4. Perform the crossover step of DE.

Step 5. Perform the selection step of DE. If the termination criterion is not satisfied go to Step 2.

4 Experimental Results

We give below some implementation details for the test problems and we present the experimental results in Tables.

Cancer1: For the problem Cancer1 an architecture 9–4–2–2 was used for the ANNs. For all the DE operators various population sizes were tested and we concluded that a population size of 40 individuals gives the best performance. When the DE was equipped with a local search method a population of 8 individuals was used. Only the two individuals with the lowest error function value executed local search. The local search method was executed for 5 iterations with mutation factor equals 0.6 and crossover factor 0.8; the DE2 operator was used. The termination criterion was an error goal of 0.035 or 500 generations.

Heart1: For the problem Heart1 for the ANN an architecture 35–8–2 was used. For all the DE operators various population sizes were tested and a population of 40 individuals produced the best performance. For the MA a population of 10 individuals was used. As before, only the two individuals with the lowest error function value executed local search. The local search method was executed for 8 iterations with mutation factor equals 0.5 and crossover factor 0.7; the DE2 operator was used. The termination criterion was the error goal 0.1 or 500 generations.

We measured the function evaluations and classification error and also calculated the average (Mean), standard deviation (Stdev), maximum (Max) and minimum (Min) values for each problem over 50 experiments.

Table 1: FE Cancer1 problem

Algorithm	F	*CR*	**Max**	**Min**	**Mean**	**Stdev**
DE1	0.6	0.9	7241	1521	3081.00	1043.53
DE2	0.2	0.8	9121	2241	4699.40	1573.30
DE3	0.6	0.7	2001	601	1031.40	292.14
DE4	0.7	0.8	2001	561	1085.80	304.38
DE5	0.5	0.6	4441	841	1720.20	856.62
LDE1	0.6	0.9	5895	695	1084.56	715.63
LDE2	0.2	0.8	1655	715	1068.96	200.71
LDE3	0.6	0.7	1078	394	**611.08**	157.57
LDE4	0.7	0.8	**1015**	**315**	651.68	**148.76**
LDE5	0.5	0.6	1141	415	724.64	167.24

In Tables 1, 3 the function evaluations performed by the algorithms on the two test problems are presented. The algorithms that use a combination of global and local search (LDE) clearly outperform the ones which use only the global search (DE). Tables 2 and 4 exhibit the results of the classification error achieved by the algorithms. For the heart problem there was not a significant difference between the algorithms, but for the Cancer problem the LDE algorithms had a better performance than the standard DE.

Table 2: CE Cancer1 problem

Algorithm	*F*	*CR*	Max	Min	Mean	Stdev
DE1	0.6	0.9	4.02	**0.00**	2.23	0.83
DE2	0.2	0.8	4.02	**0.00**	2.25	0.85
DE3	0.6	0.7	4.02	0.57	2.09	0.73
DE4	0.7	0.8	4.60	0.57	2.07	0.93
DE5	0.5	0.6	4.02	0.57	2.34	**0.65**
LDE1	0.6	0.9	**3.45**	0.57	1.80	0.69
LDE2	0.2	0.8	4.60	**0.00**	1.92	0.93
LDE3	0.6	0.7	**3.45**	0.57	**1.69**	0.76
LDE4	0.7	0.8	4.02	**0.00**	2.04	0.87
LDE5	0.5	0.6	**3.45**	0.57	1.87	0.72

Table 3: FE Heart1 problem

Algorithm	*F*	*CR*	Max	Min	Mean	Stdev
DE1	0.3	0.8	20000	6081	11128.90	3775.77
DE2	0.1	0.8	20000	7161	14304.90	4042.32
DE3	0.5	0.6	7201	2321	3669.00	805.51
DE4	0.6	0.8	5401	2121	3458.60	639.46
DE5	0.4	0.7	5241	2601	4005.00	812.39
LDE1	0.3	0.8	4836	2428	3374.18	499.19
LDE2	0.1	0.8	4584	2316	3226.36	490.32
LDE3	0.5	0.6	5704	1308	2249.38	641.97
LDE4	0.6	0.8	**3100**	**1280**	**2048.96**	**400.68**
LDE5	0.4	0.7	3389	1364	2059.88	406.29

Table 4: CE Heart1 problem

Algorithm	*F*	*CR*	Max	Min	Mean	Stdev
DE1	0.3	0.8	23.48	18.26	**19.96**	1.39
DE2	0.1	0.8	**22.61**	17.39	20.13	1.53
DE3	0.5	0.6	23.48	17.39	20.84	1.29
DE4	0.6	0.8	23.91	18.26	20.92	1.25
DE5	0.4	0.7	23.04	17.83	20.48	1.42
LDE1	0.3	0.8	24.78	17.39	20.99	1.53
LDE2	0.1	0.8	23.48	**16.52**	20.90	1.36
LDE3	0.5	0.6	23.04	17.83	20.78	**1.23**
LDE4	0.6	0.8	23.04	17.83	20.89	1.28
LDE5	0.4	0.7	23.91	17.83	20.77	1.46

5 Conclusions

A new memetic algorithm was proposed. The algorithm uses as global search method the DE algorithm and as local search method a local version of the DE algorithm. The performance of the algorithm has been tested in problems from the field of medicine and the results are promising. The proposed algorithms achieve a better performance than the standard DE algorithms.

Acknowledgment

We acknowledge the partial support of the "Pythagoras" research grant awarded by the Greek Ministry of Education and Religious Affairs and the European Union.

References

[1] D. Corne, M. Dorigo, and F. Glover. *New Ideas in Optimization*. McGraw-Hill, 1999.

[2] Y. G. Petalas and M. N. Vrahatis. Memetic algorithms for neural network training in bioninformatics. In *European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systemsm, (EUNITE 2004), June 10-12, 2004, Aachen, Germany.*

[3] L. Prechelt. Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, 1994.

[4] S.S. Rao. *Optimization theory and Applications*. Wiley Eastern Limited, 1992.

[5] R. Storn and K. Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.