Multi-criteria Optimization of neural networks using multi-objective genetic algorithm

SENHAJI Kaoutar
Laboratory of modeling and scientific calculing
University Sidi Mohamed Ben Abdellah
Faculty of Sciences and Technics
FES, MOROCCO
kaoutar.senhaji@usmba.ac.ma

ETTAOUIL Mohamed
Laboratory of modeling and scientific calculing
University Sidi Mohamed Ben Abdellah
Faculty of Sciences and Technics
FES, MOROCCO
mohamedettaouil@yahoo.fr

***Abstract*— this paper propose a new multi-objective model optimization allow training the multi-layer perceptron neural network (MLPNN) and optimizing its architecture. More precisely, this model aims to satisfy two objectives: the first one is minimizing the perceptron error (training objective) and the second one is minimizing the sum of the absolute weights (optimizing architecture objective). As known, a multi-objective problem's optimal solution is a set called Pareto set, from which a single weight vector with best performance will correspond to a reduced number of weights; this set is a compromise between the two objectives. To solve the proposed model, we have chosen the NSGA II algorithm (Non-Dominated Sorting Genetic Algorithm II). This algorithm has shown to be very powerful for multi-objective optimization, thus for multi-objective learning.**

Keywords— multi-objective optimization, neural networks, multi-layer perceptron, NSGA II, multi-criteria training, Pareto set.

I. INTRODUCTION

The multi-layer Perceptron is an efficient neural network capable to approximate any continuous function or classify any data, as long as he have enough neurons number and the adequate weight, What is knowing as optimization and learning of the MLP's. The learning of a neural network is mainly based on the search for the adequate weights value allowing to have a better result without really worrying about the best network topology, thus intervenes the optimization of the neural network. Several works treat each problem individually on the other using classical learning methods [1], optimization methods [2] [3], unfortunately learning method that use only training data error do not necessarily yield good generalization models for noisy data since they don't control flexibility during the training process. The subject about training neural networks is a real main concern, so a few theory basing on a multi-objective optimization such as model MOBJ [4] [5] and LASSO (last absolute shrinkage and selection operator) [6] appears but proposes to solve the problem via mono-objective optimization. Our approach consists in modelling and solving the generalization problem of the MLP by a purely multi-objective methodology including two objectives: to minimize the general error of the perceptron and the sum of the absolute values of the weights, in this case the solution of the problem is a set of an undominated solutions by the Pareto concept [7], what is the set name from, "PARETO FRONT". We propose to solve the model by the NSGA II (Non-dominated Sorting Genetic Algorithm II) [21] one of the most efficient multi-objective genetic algorithms. This approach will make it possible to imply the generalization of the MLP's and reduce the topology.

This paper is organized as follows. In section II is a brief description of the multi-objective optimization. Section III discusses the problem in detail, section IV is about the multi-objective resolver, NSGA II, and finally a short conclusion.

II. MULTI-OBJECTIVE OPTIMIZATION

The multi-objective optimization (by Vilfredo Pareto) is the area looking for a balance between several objectives functions, in such a way that, no criteria can be improved without deteriorating, at least, one of the other criteria, this equilibrium is called the Pareto optimum [8].

A multi-objective problem can be presented as:

Finding the Vector $x^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$, which satisfies the m inequalities constraints and the p equalities constraints as follows:

$$g_i(x) \geq 0 \qquad i = 1, \ldots, m \tag{1}$$

$$h_i(x) = 0 \qquad i = 1, \ldots, p \tag{2}$$

By optimizing (minimizing or maximizing), the following functions vector:

$$f(x) = [f_1(x), f_2(x), \ldots, f_n(x)]^T \tag{3}$$

The comparison between the feasible solutions is made by the dominance process defined by

A solution x dominates a solution y if and only if

$$\forall i \in (1,2, \ldots, k): f_i(x) \leq f_i(y) \tag{4}$$

And

$$\exists j \in (1,2, \ldots, k): f_i(x) < f_i(y) \tag{5}$$

By using this concept, a solution x* is an undominated solution or Pareto optimum if there is no solution in the decision space that dominates it. Therefore, the Pareto optimum set of a multi-objective problem is the set of all Pareto optimum solution of the problem:

$$POS = \{x \in S / \exists\, y \in S, f(y) \prec f(x)\} \tag{6}$$

POS: Pareto optimum set;

The image of the optimal Pareto set in the objectives space is called "the Pareto front", the shape of this boundary takes different forms depending on whether the objectives should be minimized or maximized.

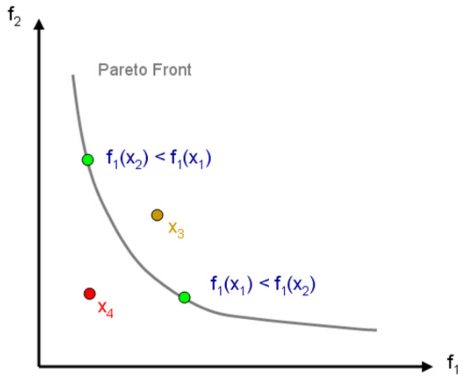

Figure 1. The shape of the Pareto front for a minimizing multi-objective problem

Many algorithm was proposed to solve the multi-objective problem by a constrained optimization method as the ellipsoidal algorithm [9]. Alternative multi-objective algorithms include Sliding Mode Multi-Objective algorithm [10], Levenberg-Marquardt based Multi-Objective [11] and Multi-Objective Evolutionary Algorithms [12] [13]. The research proved that the genetic algorithm is more suited to use for similar problem, since that, many variation was developed like VEGA [14] (non-aggregated non Pareto methods), MOGA [15], NSGA [16], NPGA [17] (based on Pareto non-elitist); SPEA [18], PAES [19], PESA [20], NSGA II [21](Pareto elitist).

III. Multi-objective Learning Approach

The generalization performance of a neural network is defined by its capability of learn from data set minimizing the influence of a stochastic variable associated with a noise [22]. This performance can be measured by the error value of the validation set compared to the training one. In general, the final model should provide the balance between the training and validation set error.

Since we treat the case of a multilayer perceptron, the local error, the error corresponding to a single example, the pair (x, d) such that:

x : Input vector;
d : Desired output vector;

Therefore, the local error is defined as:

$$\varphi_j(w) = \frac{1}{2}\sum_{i=1}^{N}\left(d_i^j - g(w, x_j)\right)^2 \qquad (7)$$

Such that:

N: is the size of the output;
d_i^j : the output i of the desired vector output d^j;
w : The network weights;
$g(w, x_i)$: The network output Associated to the input vector x_j;

Then the global error of the training set can be defined as:

$$E(w) = \frac{1}{k}\sum_{j=1}^{k}\varphi_j(w) \qquad (8)$$

Such that

k: is the size of the training set;

The topology of the network can also be pruned to improve generalization, to get it, we propose to eliminate the network weights of over determined networks, for a better balance between flexibility and rigidity of the multi-layer perceptron, presented the following formula:

$$\mathrm{J(w)} = \sum_{i=1}^{n}|C_i w_i| \qquad (9)$$

Such that

n : is the network weights vector size;

C_i Decision variable such that

$$\boldsymbol{C_i = \begin{cases} 1 \text{ if } w_i \text{ is kept in the perceptron architecture} \\ 0 \text{ if not} \end{cases}}$$

A. The problem formulation

The multi-objective model try to optimize two conflicting objectives: the error (formulated by the objective (9)) and the sum of the absolute weights (presented by the objective (8)), cost functions. The multi-objective model is presented by:

$$(P)\begin{cases} \min \sum_{j}^{k}\sum_{i}^{N}\left(d_i^j - g(w_i', x_j)\right)^2 & (10) \\ \min \sum_{i=1}^{n}\left|w_i'\right| & (9) \end{cases}$$

Such that:

$$w_i' = |C_i w_i|$$

IV. Non-dominated Sorting Genetic Algorithm II

The genetic algorithm (GA) is a meta-heuristic approach based in evolutionary population, spared from Darwin principle « the fittest survive » in nature, proposed by Holland [23]. Since then GA was used and developed to solve many problem, including the multi-objective optimization. One of the variation of GA multi-objective is NSGA [16], a popular non-domination based in genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally critiqued for its computational complexity, lack of elitism and for choosing the optimal parameter value for sharing parameter σshare. A modified version, NSGAII [21] was developed, one of the most efficient multi-objective evolutionary algorithm [24] [25], which has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori. It uses the crowding distance, which estimates the density of solutions in the objective space, and the crowded comparison operator, which guides the selection process towards a uniformly spread

Pareto-frontier [26]. NSGA-II is discussed in detail in this section.

A. *Algorithm*

A general NSGA II procedure is presented as follows:

Step 1: Create a random parent population P_t of size Z;
Step 2: Sort the random parent population based on non-domination concept;
Step 3: for each non-dominated solution assign a rank equal to its non-domination level, 1 is the best level, 2 is the next best level and so on;
Step 4: Create on offspring population Q_t using selection and production operators as following:

1) Selection : the comparison step applied to choose a parents solutions is defined by the rank value, if n solutions have the same rank value, the crowded comparison operator based on the crowding distance is applied, the solution with the best crowding distance Is kept;
2) Crossover : choose the parents solutions for crossover;
3) Mutation : Choose the parents solutions for mutation;

Step 5: create the mating pool R_t by combining the parent population P_t and the offspring population Q_t;
Step 6: sort the combined population R_t according to the fast non-dominated sorting procedure to identify all non-dominated fronts $(F_{r1}, F_{r2}, \cdots, F_{rm})$;
Step 7: Generate the new parent population P_{t+1} of size Z by choosing non-dominated solutions, starting from the first ranked non- dominated front F_{r1}. When the population size z is exceed, reject some of the lower ranked non-dominated solution;
Step 8: Repeat from step 3 until the stopping criteria is reached.

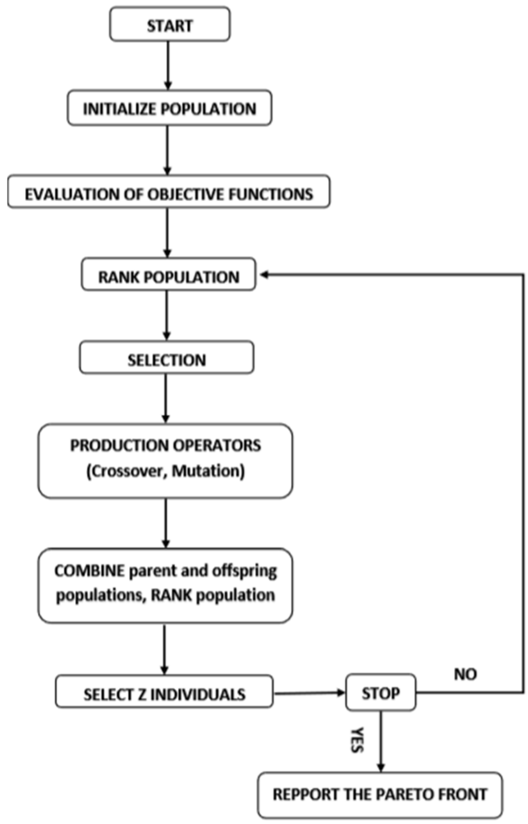

Figure 2. Flow chart of NSGA-II

B. *Crowding distance*

The crowding distance serves as an estimate of the perimeter of the cuboid (figure 3.) formed by using the neighbours as the vertices; an estimate of the density of solutions surrounding a particular solution in the population.

The algorithm used to calculate the crowding distance of each point in the set F_r is given by:

Step 1: For each solution in the set, F_r assign 0 to the crowding distance corresponding;

Step 2: for each objective function $f_m, m = 1,2, \dots, M$, sort the set in worse order of f_m;

Step 3: assign a large distance to the boundary solutions $d^r(1) = d^r(l) = \infty$ (1 is the first solution and l is the last one in the front F_r. On the other hand, for all other solutions $i = 2, \dots, l-1$ assign:

$$d^r(i) = d^r(i) + \frac{\left(f_m^r(i+1) - f_m^r(i-1)\right)}{f_m^{max} - f_m^{min}}$$

$f_m^r(i+1)$ is the mth objective function value of the $(i+1)$ solution in the set F_r;

$f_m^r(i-1)$ is the mth objective function value of the $(i-1)$ solution in the set F_r;

f_m^{max} is the maximum value of the mth objective;

f_m^{min} is the minimum value of the mth objective;

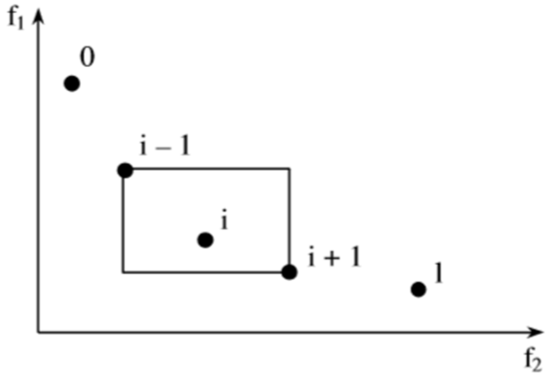

Figure 3. The crowding distance calculation

C. Crowding-comparison operator

The crowded-comparison operator leads the selection process towards a uniformly spread out Pareto-optimal front. Every individual in the population has two attributes: non-domination rank i_{rank} and crowding distance i_{Cr}, so the crowding-comparison distance can be define as:

$$\boldsymbol{f}\ (i_{rank} < j_{rank})\ or\ ((i_{rank} = j_{rank})\ and\ (i_{Cr} > j_{Cr}))$$

Between two individuals with differing non-domination ranks, the individual with the lower rank is selected. If the both individuals belong to the some front, then the individual with larger crowding distance is preferred.

CONCLUSION

The generalization problem of a neural network is actually a main concern for many researcher; a good generalization can lead to a better-supervised learning. In this paper, we try to improve the generalization performance of a multi-layer perceptron by a multi-objective modelization with two objectives: to minimize the general error of the perceptron and the sum of the absolute values of the weights. We have opted for the Non-dominated Sorting Genetic Algorithm II (NSGA II) resolver, a very popular variation of genetic algorithm for his efficiency, who use the crowding distance for better sorting algorithm.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagation error," *Nature,* vol. 323, pp. 533-536, 1986.

[2] H. Ramchoun, M. A. Janati Idrissi , Y. Ghanou and M. Ettaouil, "Multilayer perceptron: Architecture optimization and training," *International Journal of Artificial Intelligence and Interactive Multimidia,* vol. 4, no. 1, pp. 26-30, 2016.

[3] E.-s. Abdelatif, . F. Harchli and M. Ettaouil , "Optimization of the organized Kohonen map by a new model of preprocessing phase and application in clustering," *Journal of Emerging Technologies in Web Intelligence,* vol. 6, no. 1, pp. 80-84, 2014.

[4] R. d. A. Teixeira, A. Pàdua Braga, . R. H.C. Takahashi and . R. R. Saldanha, "Improving generalization of MLPs with multi-objective optimization," *Neurocomputing ,* vol. 35, no. 1, pp. 189-194, 2000.

[5] R. A. Teixeira, A. P. Braga, R. H. Takahashi and R. R. Saldanha, "Recent advances in the MOBJ algorithm for training artificial neural networks," *International journal of neural systems,* vol. 11, no. 03, pp. 265-270, 2001.

[6] M. A. C. a. A. P. Braga, "Optimization of neural networks with multi-objective lasso algorithm," in *In The 2006 IEEE International Joint Conference on Neural Network Proceedings*, Vancouver BC, Canada, 16-21 July 2006.

[7] V. Charkong and Y. Y. Haimes, Multiobjective decision Making : Theory and Methodology, vol. 8, New York: North-Holland (Elsivier), 1983.

[8] Y. Collette and Patrick Siarry, multiobjective optimization : Algorithmes, Editions Eyrolles, 2011.

[9] R. G. Bland, D. Goldfarb and . M. J. Todd, "The ellipsoid method: a survey," *Operations Research,* vol. 29, no. 6, p. 1039–1091, 1981.

[10] M. A. Costa, A. P. Braga, . B. R. Menezes, . R. A. Teixeira and G. G. Parma, "Training neural networls with a multi-objective sliding mode control algorithm," *Neurocomputing,* vol. 51, p. 467–473, 2003.

[11] M. A. Costa, . A. P. Braga and . B. R. Menezes, "Improved Generalization Learning with Sliding Mode Control and the Levenberg-Marquadt Algorithm," in *Proceedings of VII Brazilian Symposium on Neural Networks*, 2002.

[12] C. A. C. Coello, D. A. Van Veldhuizen and G. B. Lamont, Evolutionary algorithms for solving multi-objective problems, vol. 242, New York: Kluwer Academic, 2002.

[13] C. C. Coello, G. B. Lamont and D. A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems, pringer Science & Business Media, 2007.

[14] S. J. David, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st international Conference on Genetic Algorithms*, 1985.

[15] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization," in *Proceeding of the 5th International Conference on genetic Algorithms ICGA*, June 1993.

[16] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation,* vol. 2, no. 3, pp. 221-248, 1994.

[17] J. H. rey, N. Nafpliotis and D. E. Goldberg, "Multiobjective optimization using the niched pareto genetic algorithm," in *IlliGAL report no. 93005*, 1993.

[18] E. Zitzler and T. Lathan, "An Evolutionary Algorithm for multiobjective Optimization : the strenght Pareto Approach," *TIK-Report,* no. 43, 1993.

[19] J. Knowles and D. Corne, "The Pareto Archived Evolution Strategy : A New Baseline Algorithm for multiobjective optimization," in *Evolutionary Computation, Proceedings of the 1999 Congress on. IEEE*, washigton, july 1999.

[20] C. D. W., J. D. Knowles and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *In International Conference on Parallel Problem Solving from Nature*, Berlin Heidelberg, September 2000.

[21] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Nondominated Sorting Genetic Algorithm for multiobjective optimization : NSGA II," in *International Conference on Parallel Problem Solving From Nature*, Berlin Heidelberg, September 2000.

[22] D. H. Wolpert, "Stacket generalization," *Neural networks,* vol. 5, no. 2, pp. 241-259, 1992.

[23] J. Holland, "Adaptation in neural and artificial systems," *Ann Arbor: The University of Michigan Press,* 1975.

[24] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation,* vol. 8, no. 2, pp. 173-195, 2000.

[25] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *International Conference on Parallel Problem Solving from Nature*, Berlin Heidelberg, September 1998.

[26] F.-A. Fortin and M. Parizeau, "Revisiting the NSGA-II crowding-distance computation," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, July 2013.