A Global Optimization Approach to Neural Network Training

C. Voglis & I.E. Lagaris

Dept. of Computer Science, University of Ioannina

{voglis, lagaris}@cs.uoi.gr

Abstract

We study effective approaches for training artificial neural networks (ANN). We argue that local optimization methods by themselves are not suited for that task. In fact we show that global optimization methods are absolutely necessary if the training is required to be robust. This is so because the objective function under consideration possesses a multitude of minima while only a few may correspond to acceptable solutions that generalize well.

Keywords- Artificial neural networks, global optimization, multistart.

1. INTRODUCTION

The minimization of multimodal functions with numerous local and global minima is a problem that frequently arises in many scientific applications. In general the nature of some applications is such that it is necessary to detect all the global minimizers (e.g. *computation of Nash equilibria [9]* in game theory) or a set of minima with objective function value in a specific range (*energy values in the molecular conformation problem* [6]). Another interesting application that requires the computation of more than one global minimizer is the computation of *periodic orbits of nonlinear mappings* [8]. Neural network training is a problem of similar nature; i.e. the relevant objective function possesses a multitude of local (and possibly global) minima.

Consider the classical data fitting problem: *Given M points and associated values $(x_i, y_i),\ i = 1, 2, \ldots, M$, with $x_i \in R^d,\ y_i \in R$, draw a smooth hypersurface that is optimal in the least square sense.*

The traditional way of solving such problems is to assume a parametric model (e.g. an Artificial Neural Network) $N(x; p)$ and adjust the parameters p, so as to minimize the deviations, i.e. minimize the "Error":

$$E(p) = \sum_{i=1}^{M} [N(x_i; p) - y_i]^2 \tag{1}$$

Thus the problem of training an ANN is transformed into an optimization one

$$\min_{p} E(p) \tag{2}$$
$$s.t.\ \ p \in S \subset R^n$$

Unfortunately the terrain modelled by the error function can be extremely rugged and often has a multitude of local minima. Obviously, a method that can not escape from local minima has hardly any chance to find a solution to the problem. We must add that smaller neural networks generalize better, since they avoid over-fitting and this is the reason they are preferred for both classification and regression tasks. On the other hand, training smaller networks is more difficult since the error surface is heavily rugged and there exist only a few good solutions.

2. MULTISTART BASED ALGORITHM IN GLOBAL OPTIMIZATION

It is important to describe in brief the basic framework of a multistart based global optimization method. In Multistart, a point is sampled uniformly from the feasible region, and subsequently a local search is started from it. The weakness of this algorithm is that the same local minima may be found over and over again, wasting so computational resources.

The **"region of attraction"** of a local minimum associated with a deterministic local search procedure $\mathcal{L}$ is defined as:

$$A_i \equiv \{x \in S, \mathcal{L}(x) = x_i^*\} \tag{3}$$

where $\mathcal{L}(x)$ is the minimizer returned when the local search procedure $\mathcal{L}$ is started at point x.

Algorithm MA 1 Multistart framework

Initialize: Set k=1
Sample $x \in S$
$y_k = \mathcal{L}(x)$

Termination Control: If a stopping rule applies STOP.

Sample: Sample $x \in S$

Main step: If ($x \notin \cup_{i=1}^{k} A_i$) Then
$y = \mathcal{L}(x)$
$k = k + 1$
$y_k = y$
Endif

Iterate: Go back to the Termination Control step.

The algorithm described above, returns a set $Y = \{y_k\}$ of the recovered local minima. A lot of research is carried out on reducing the number of times, the local search procedure is applied in a way that minimizes the risk of missing a local minimum.

3. LOCATING MINIMA IN NEURAL NETWORK TRAINING

The main purpose of this work is to demonstrate the need of a multistart based global optimization method for training NNs. In the literature so far we can distinguish three major classes of methods:

- Local optimization procedures: All methods that attempt to find a local minimum of the error function $E(p)$. (Gradient descent, conjugate gradient, quasi-Newton methods, Levenberg-Marquardt)

- Global optimization procedures (single global minimum): These methods employ probabilistic or deterministic strategies, to overcome local minima and locate a single global optimum. (Trajectory methods [1], covering methods[11], evolutionary algorithms [10,11], simulated annealing [4,11])

- Global optimization procedures (all global minima): These methods use global strategies to locate all the existing global minima. (Interval methods [12], Particle swarm [2, 3])

In this section we are going to present real cases in neural network training that the above mentioned procedures will not perform optimally. Note that:

- Local minima are often poor solutions to the training problem. Thus, local optimization methods are out of question.

- The various global minima present different interpolation behavior, i.e they do not generalize in a similar way.

- There are cases where some local minima generalize better than a global minimum.

We list the two regression problems used to illustrate our points.

Problem A: We used 40 evenly distributed points in $[0, 20]$ and their corresponding function values, to construct the training data set $\mathcal{D}_A = (x_i, f(x_i)),\ i = 1, \ldots, 40$ and 500 points for the test set $\mathcal{T}_A$, with $f(x) = x\sin(x)$

Problem B: We used 40 evenly distributed points in $[0, 10]$ and their corresponding function values, to construct the training data set $\mathcal{D}_B = (x_i, f(x_i)),\ i = 1, \ldots, 40$ and 500 points for the test set $\mathcal{T}_B$, with $f(x) = x \sin(x^2)$.

3.1 Quality of local minima

Consider a feedforward artificial neural network, with sigmoid activation functions in the hidden layer. This model can be written as:

$$N(x; p) = \sum_{i=1}^{n_h} p_{3i-2}\sigma(p_{3i-1}x + p_{3i}) \tag{4}$$

$$\text{with } \sigma(z) = \frac{1}{1 + e^{-z}}$$

where the weight parameters p are numbered as shown in Figure 1.

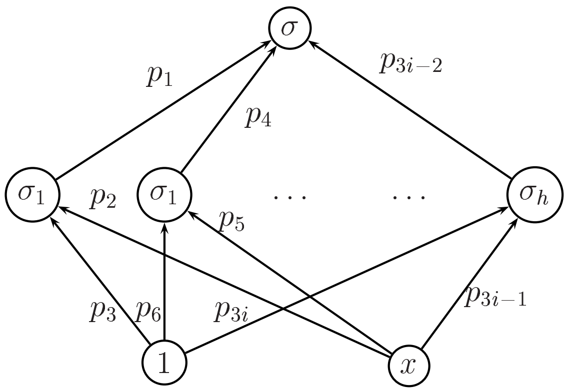


Figure 1: Neural network: Labelling of parameters

One easily realizes that problem (2) may be solved with a large number of different values for the parameters.

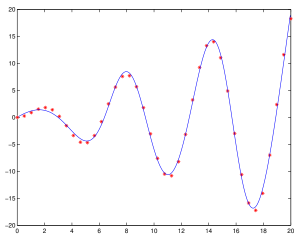
In order to evaluate the quality of local minima for problem A, we optimized the error function using a multistart-based global optimization method [7]. Our goal was to find as many local minima as possible. Using six nodes in the hidden layer ($n_h = 6$), we found more than $10,000$ minima. In Table 1 we present only a small subset of 20 minima.

	Min.	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	11.93	-626.79	0.49	-4.05	400.	-0.85	6.11	439.16	-0.97	15.27	526.32	0.26	-3.36
2	13.13	-595.56	-0.26	3.53	400.	1.00	-15.79	504.03	0.82	-5.98	276.26	-0.56	4.46
3	59.37	-730.46	-0.54	6.04	400.	0.87	-8.98	568.78	0.96	-18.08	501.58	-0.28	4.46
4	60.26	-229.63	0.89	-13.88	99.48	0.37	-1.81	0.00	1.03	-10.39	756.47	0.24	-3.36
5	62.42	-429.90	-0.42	8.15	400.	-1.77	12.18	193.51	1.07	-13.74	298.31	-0.41	6.14
6	64.22	-613.04	0.28	-4.57	400.	-1.02	19.17	433.43	-0.99	10.17	296.87	0.55	-6.20
7	70.87	-373.12	-0.43	8.59	400.	1.94	-13.30	125.49	1.11	-14.25	222.39	-0.42	6.22
8	80.20	-577.92	-0.57	6.44	400.	-0.94	17.86	406.32	-1.01	10.42	303.61	-0.34	4.89
9	80.31	-286.48	-0.28	4.87	400.	-1.25	20.00	0.00	1.29	-9.11	58.09	-0.78	6.04
10	84.20	-628.36	-0.29	4.69	400.	0.90	-14.27	305.83	-1.30	13.07	513.88	-0.09	2.53
11	84.23	-630.99	-0.29	4.72	400.	0.90	-14.21	153.35	1.30	-13.06	750.10	-0.09	2.65
12	85.34	-585.91	0.06	-1.95	400.	0.98	-15.53	255.16	-1.37	13.72	478.08	0.27	-4.38
13	86.38	-188.04	0.98	-15.44	0.00	0.08	-0.50	257.18	-1.37	13.74	386.63	0.29	-4.60
14	291.60	-240.72	0.10	-4.30	0.00	0.65	-7.33	0.00	1.08	-16.88	800.00	0.43	-6.11
15	334.71	-272.36	-0.25	5.96	400.	1.38	-18.17	142.09	-0.60	8.59	0.00	0.25	-5.96
16	353.93	-270.56	0.63	-9.05	400.	1.36	-18.06	202.06	0.32	-6.61	324.18	-19.58	-20.00
17	353.93	-270.56	0.63	-9.05	400.	1.36	-18.06	200.80	-9.01	-20.00	1.26	0.32	-6.61
18	354.72	-253.81	-4.02	-6.56	2.04	0.63	-7.44	0.00	1.10	-17.15	800.00	0.49	-7.00
19	354.85	-251.78	0.63	-7.45	0.00	-4.27	-19.58	0.00	1.10	-17.15	800.00	0.49	-7.01
20	828.76	-580.81	-18.01	-20	101.91	1.19	-20	-332.60	1.24	-20	800	1.02	-17.15

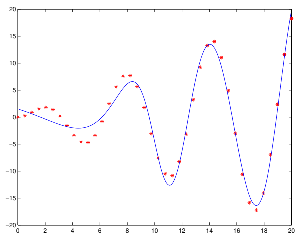
Table 1: A set of 20 selected local minima (Problem A)

We have sorted these solutions in ascending order of the $E(p)$ value. Intuitively we can assume that the quality of the approximation is inversely proportional to the Error Function value achieved for each minimum. This is shown in the Figure 2, where we plot 4 found solutions. Notice that only the first set of parameters (Solution 1) managed to approximate the target function accurately.

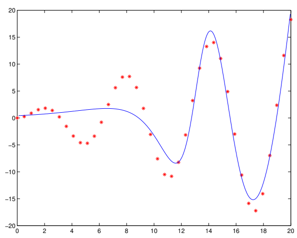
Consequently the probability that a local search method recovers such a solution, is rather small considering the large number of existing local minima.



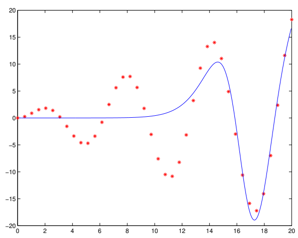
(a) Solution 1



(b) Solution 8



(c) Solution 17



(d) Solution 20

Figure 2: Quality of local minima found.

j	Train Error	Test Error	$\|\|\vec{p}_1 - \vec{p}_j\|\|_2$
1	0.0013	4.38	0.0
2	0.0012	0.28	6246.69
3	0.0014	4.0	7017.74
4	0.0016	0.1	6162.54

Table 2: An example of 4 global minima

3.2 Quality of the global minima

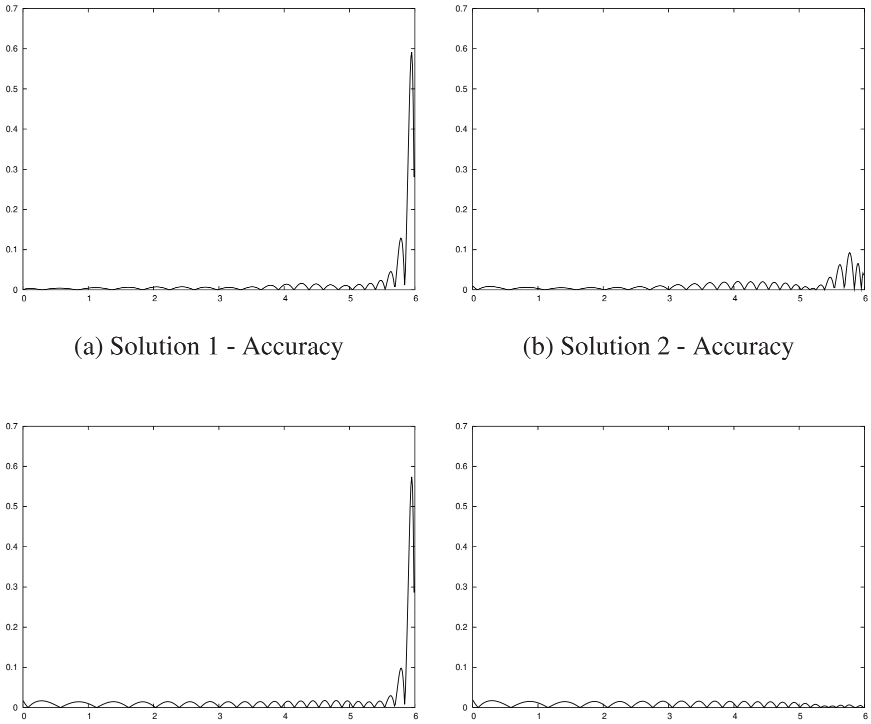
In the past, many researchers used global optimization algorithms to search for a single global minimum. Such strategies perform better than local techniques, however they do not take in account the existence of many *global* minima. If this is the case (which is quite common), multistart based global optimization algorithms, that recover all the global minima of the problem can be used to identify (by means of a test set) the best parameter values.

To illustrate this we solved Problem B, using a neural network with 15 nodes in the hidden layer. Four global minima were recovered, presented in Table 2. Equivalent solutions generated by node-permutation were excluded. As can be realized by inspecting Table 2, the four solutions perform quite differently in the test set, yielding a clear winner (Solution #4, with test error 0.1). In Figure 3 we plot for each point in the test set, the accuracy of the approximation, i.e. the quantity $|f(x_i) - N(x_i; p)|$.

3.3 Finding Multiple Minima vs. Overtraining

Overtraining is a frequently encountered problem in NN training. This happens when the model parameters are extremely well tuned to the training data and interpolate inaccurately in nearby points.

Such kind of solutions may correspond to global minima as well. So it is important to maintain as candidate solutions global and local minima as well.



(a) Solution 1 - Accuracy

(b) Solution 2 - Accuracy

(c) Solution 3 - Accuracy

(d) Solution 4 - Accuracy

Figure 3: Quality of local minima found.

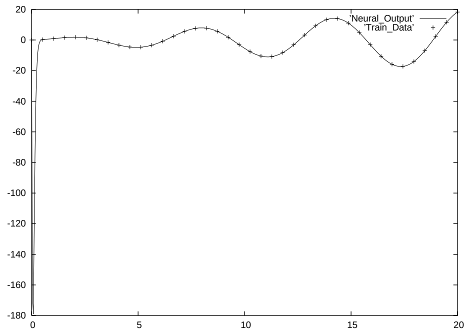
	Train Error	Test Error	$\|\|p_1 - p_j\|\|_2$
1	1.10E-4	1.7E+5	0.0
2	3.02E-2	0.78	3914.18
3	7.9E-2	0.89	3937.74
4	0.22	2.13	1167.21

Table 3: An example of overtraining.

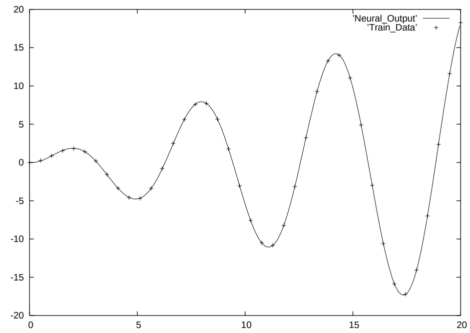
To illustrate the above, we solved Problem A, using TML and a neural network consisting of 10 hidden nodes. In Table 3 we present the four best solutions found. It is remarkable that the solution with the lowest error in the training set displays the worst error in the test set. The first two solutions are shown in Figure 4. Notice the large oscillation near zero, a characteristic sign of overtraining.

4. CONCLUSIONS

The plethora of local minima inherent in the objective function that results in the case of



(a) Solution 1. Overtrained solution - Error in test set = 169953. Note the behavior around $x = 0$



(b) Solution 2. Quality solution - Error in test set = 0.78

Figure 4: An example of overtraining.

neural network training, renders necessary the use of global optimization methods. Since generalization is a very important property that can be verified only a-posteriori (i.e. by using a test data set), multistart-based methods that recover all the local (and global minima) should be preferred.

In this article we presented a framework that can be used for achieving robust neural network training. Almost any multistart-based method may be used since most variations aim in collecting all the local minima inside the feasible region.

The test problems presented here are constructed for illustration purposes, however they are typical and represent the difficulties of real world problems.

REFERENCES

1. Shang, Y. & Wah, B. W. (1996). Global optimization for neural network training, *IEEE Computer*, v. 29, no. 3, pp. 45–54.
2. Parsopoulos, K. .E & Vrahatis, M. N., (2004). On the computation of all global minimizers through particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, v. 8, no. 3, pp. 211–224.
3. Plagiannakos, V. P., Magoulas, G. D., Androulakis, G. S., & Vrahatis, M. N. (1999). Global search methods for neural network training, *Proceedings of the 3rd IEEE-IMACS World Multiconference on Circuits, Systems, Communications and Computers*, vol. 1, pp. 3651–3656.
4. Corana, A., Marchesi, M., Martini, C., & Ridella, S. (1987). Minimizing multimodal functions of continuous variables with the Simulated Annealing algorithm, *ACM Trans. Math. Soft.*, vol. 13, pp. 262–280.

5. Gori, M. & Tesi, A. (1992). On the problem of local minima in backpropagation, *IEEE Trans. Patterns Analysis and Machine Intelligence*, vol. 14, pp. 76–85.
6. Saunders, M., Houk, K. N., Wu, Yun-Dong., Still, W. C., Lipton, M., Chang, W. G. (1990). Conformations of Cycloheptadecane. A comparison of methods for conformational searching, *J. Am. Chem. Soc.*, vol. 112, pp. 1419–1427.
7. Lagaris, I. E., Papageorgiou, D. G., & Theos, F. V. (2004). PANMIN: Sequential and parallel global optimization procedures with a variety of options for the local search strategy, *Computer Physics Communications*, vol. 159, pp. 63-69.
8. Vrahatis, M. N. (1995) An efficient method for locating and computing periodic orbits, *J. Comp. Phys.*, vol. 119, pp. 105–119.
9. Floudas, C. et. al. (1999) Handbook of Test Problems in Local and Global Optimization. Dordrecht: Kluwer Academic Publishers.
10. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley: Reading, MA.
11. Torn, A. & Zilinskas, A. (1987). Global optimization. Springer–Verlang.
12. Hansen, E. (1992) Global Optimization Using Interval Analysis. New York:Dekker.