ALTERNATIVES TO GRADIENT-BASED NEURAL TRAINING.

Włodzisław Duch
Department of Computer Methods, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
E–mail: duch@phys.uni.torun.pl

Abstract

Neural networks are usually trained using local, gradient-based procedures, and the best architectures are selected by experimentation. Gradient methods frequently find suboptimal solutions being trapped in local minima. Genetic algorithms are frequently used but do not guarantee optimal solutions and are computationally expensive. Several new global optimization methods suitable for architecture optimization and neural training are described here. Multistart initialization methods are also offered as an alternative to global minimization.

I. INTRODUCTION

SOFT computing methods compete with traditional pattern recognition and statistical methods in many applications. For neural networks with predetermined structure, for example Multilayer Perceptrons (MLPs) with fixed architectures, finding an optimal set of parameters (weights and thresholds) requires a solution of a non-linear optimization problem. Such problems in general are NP-complete and the chance to find the best solution using typical, gradient-based learning techniques starting from a large, multi-layered network, is minimal.

There are many global optimization (GO) methods suitable for use in minimization of the neural cost functions and optimization of neural architectures. Selection of the minimization procedure may lead to great improvement in the quality of the network and in the speed of convergence of the learning algorithm itself. Global minimization [1] replacing the gradient-based backpropagation algorithms (BP – this abbreviation is used here as a synonym of any gradient-based training method) may find solutions to hard problems using smaller, compact neural networks. For example, Shang and Wah [2] have found good solutions to the two-spiral benchmark problem using just 4-6 hidden neurons, while previous smallest MLPs (build using the cascade correlation algorithm [3]) used 9 neurons, 75 weights, and the training process was very sensitive to initial conditions. Except for genetic algorithms which are frequently combined with neural networks only a few global optimization methods have been applied so far to training and optimization of neural architectures.

The direct approach to finding optimal network structures is to use 'educated guesses' for good structures, and then select the best ones. GO methods, such as genetic algorithms (GA) [4] or simulated annealing (SA) [5], may guide us in this process. This strategy is based on an assumption (rarely spelled out explicitly) that the quality of the network, measured by the error on the training or sometimes on the validation set, is a smooth function of network topology and its adaptive parameters. Perhaps the most obvious, although rarely used method to find optimal solution, is based on good initialization, followed by gradient

optimization [6], [7]. Initialization should bring adaptive parameters into the vicinity of the global minimum. In a long series of computer experiments Schmidhuber and Hochreiter [8] observed that repeating random initialization ("guessing" the weights) many times leads to faster convergence than using sophisticated versions of gradient methods. Gradient learning procedures were not able to compensate for bad initial values of weights and biases, getting stuck in local minima. Therefore a good strategy is to abandon training as soon as it slows down significantly and start again from random weights. A better strategy is to use good initialization followed by a short gradient descent, or a global minimization method to solve the non-linear optimization problem.

In this paper alternatives to the standard gradient-based backpropagation training techniques are described and several novel global optimization methods suitable for neural network training and architecture optimization introduced. While most global minimization methods are computationally quite expensive there are some that are competitive with standard BP and should be used at least for initialization of gradient learning.

II. GLOBAL MINIMIZATION METHODS

THE problem of unconstrained global minimization is stated as follows: given a vector of initial parameters P, including such information as weights, biases, other adaptive parameters, structure of the network, and a function $E(P)$ evaluating the quality of this vector for some dataset $D = \{X^{(i)}, Y^{(i)}\}$ (this may be either training or validation dataset), where $X^{(i)}$ are input vectors and $Y^{(i)}$ are the desired target vectors, generate new vectors of parameters $P^{(k)}$ until global minimum of $E(P)$ function is found.

Since most global minimization procedures are relatively expensive hybrid methods are advocated here. The simplest method is based on Monte Carlo (MC) approach. New vector of parameters P is randomly generated by changing a single parameter or a group of parameters. For optimization of neural network structures the change may involve adding or deleting one connection or one neuron with some connections. After a short gradient-based training resulting networks are evaluated on a validation set. Using gradient training in connection with MC optimization of architecture does not guarantee that globally optimal solution will be found (but the use of genetic algorithms does not guarantee it either), but is relatively fast and may create several models useful for creation of an ensemble. If the change of the quality of neural solutions is not a smooth function of the parameters defining neural architectures MC approach will work on average as well as any other, more sophisticated method, such as GAs.

To improve results of BP training MC search in the space of quantized weights and biases is used in the first phase of training, followed by short gradient-based training (without quantization) of the most promising networks that have been selected by MC. Thus MC is used both for selection of architectures and initial learning phase. A pool of the most promising candidate networks may be selected before BP learning starts. After a selection of small number of the best neural architectures MC step may be repeated, this time with relatively small variation of architectures allowed.

Several improvements of this basic schemes are proposed: improvements in the Monte Carlo procedure in which a whole ensemble of energy functions is defined [9] and the minimum is searched for on all these energy landscapes, various versions of simulated annealing [10] (SA), rescaling of the error functions in SA or MC [11], modifications of the Alopex algorithm [12] based on SA, excluding already explored areas of the parameter space [13]. Various versions of multisimplex [14], Random Line Generation [15] and particle swarm optimization [16] methods are presented. Deterministic methods of global exploration of input space are also discussed: trajectory based method, such as NOVEL [2], smoothing algorithm [17], branch and bound methods and interval methods [18].

GO methods described here do not require calculation of gradients and may be used for optimization of weights, biases and also slopes of the sigmoidal functions. In some cases networks with large slopes (>10) give significantly better results. In BP methods transition to very large weights and biases, corresponding to high slope values, is not numerically stable. Some modifications of the existing GO methods that have not yet been used for neural networks are listed below:

1. Selecting up to K contributions to the error function from randomly selected training vector and its $K-1$ nearest neighbors is a modification of the Dittes Monte Carlo procedure [9]. It may be used in any SA scheme.
2. Parallel multi-simulated annealing procedure (a 'population' of SA runs created during a single run) may be used for initialization of the gradient descent searches around promising values found after a fixed number of function evaluations. The list of hyperboxes containing the local minima found by gradient procedure should be kept to avoid repetitions.
3. SA or GA may be combined with "rough quantization" approach, i.e. at the beginning only a few bits per weight are allowed and the changes are relatively large since they have at least the size corresponding to the flipping of the least significant bit. For example, a sign plus a single bit per weight gives the possibility to have 3 values, $0, \pm 1$ and in our experience this may already be sufficient to get quite good results [19]. Annealing is than equivalent to increasing the resolution of parameters around the most promising minima.
4. The Alopex algorithm may be combined with the "rough quantization" of adaptive parameters in the global search phase.
5. From the formal point of view genetic algorithms define specific prescription allowing to make changes of adaptive parameters, and therefore they may be combined with the extension of Monte Carlo approach proposed by Dittes [9].

III. ALTERNATIVES TO GLOBAL OPTIMIZATION

TWO alternatives to global optimization methods are discussed here: initialization methods for optimization of neural architectures and ensemble methods for combining results of many models.

Good initialization of network structure and adaptive parameters may bring the neural model close enough to the global minimum to make the local minimization techniques sufficient for finding an optimum solution. Recently we have proposed several initialization

methods based on clusterization [6] and statistical discriminant analysis [7]. The method works for single or more hidden layers and could be used with various parameters for initial clusterization (or simply to provide multistart parameters (one may also add some random numbers to the proposed initial weights) and to set up different structures of networks. What is of primary importance is that – as already has been mentioned – in small networks with small number of parameters it is very hard to find globally optimal set of weights and therefore if multistart gradient methods are used a good initialization is needed. The multistart gradient method is relatively fast comparing to most global minimization methods and with a proper starting point may be an alternative to global minimization.

The methods presented in [6], [7] are easily extended to any architecture containing in the first hidden layer sufficient number of the hidden neurons to account for all the clusters; parameters of other layers should be set up in such a way that the second hidden layer is treated as output nodes and further layer just pass information. In short our minimal architecture is embedded in more complex architecture, and all extra connections have small random weights, while the output from the embedded three-layered network is passed to the final output through the extra hidden layers. In this way the same initialization by prototypes may be used in more complex architectures. Since the initial network should be similar to the network with globally optimal architecture and parameters multistart local optimization of such networks may be an inexpensive, but interesting alternative to global minimization. The weights may be arbitrarily large (the norm of the weight simply changes the slope of the sigmoidal function), and the resulting networks are quite small. The results of optimization through initialization have not yet been compared with those obtained by optimization using global minimization techniques.

It is also not clear whether global optimization methods are competitive to mixture of experts. There are many methods for combining results of various models obtained for example by using systematically generated MLP networks of various complexity or using completely different models, such as various neural networks with several transfer functions combined with machine learning methods. In the simplest approach linear combination of predictions of these models is used to form an ensemble (cf. [20]):

$$M(\mathbf{X},\alpha) = \frac{1}{K}\sum_{k=1}^{K} \alpha_k M_k(\mathbf{X},\mathbf{W}_k) \tag{1}$$

$$\sum_{k=1}^{K} \alpha_k = 1; \qquad \alpha_k \geq 0 \tag{2}$$

There are no indication in the literature that non-linear combination of models has advantages. The parameters α of the ensemble are optimized using the least squares method. In probabilistic (Bayesian) interpretation these coefficients are treated as probabilities that the data has been generated by M_k models.

More sophisticated methods are based on a resampling approach to combine results. Several such resampling schemes were developed in statistics and machine learning (cf. Breiman [21] or Diettrich [22]) and should perform very well also for neural networks.

Ensemble methods may easily be combined with initialization methods. For example setting different granularity in initial clusterization procedure will produce a series of methods with various bias-variance tradeoffs. Since generation of these models is computationally inexpensive it may be competitive to global optimization.

IV. FINAL REMARKS

ALTHOUGH genetic algorithms are almost the only well known GO method applied to neural networks several alternatives to simple gradient-based optimization and training of neural systems exist: many global optimization methods, initialization methods and ensemble methods. To determine which of these methods will prove to be the most useful requires large-scale empirical comparisons. Only a few global optimization methods have already been tried in the context of neural networks, usually with very good results. Unfortunately no systematic comparison of these methods is available. In cases published so far networks trained with GO methods performed very well, sometimes finding solutions of much better quality, but for many datasets gradient-based solutions are satisfactory.

Genetic algorithms are certainly very interesting but disproportionately much effort has been put into their applications to neural systems. There is no empirical evidence to support the idea that they lead to the best solution of the learning problem or to optimal neural architectures. Other global minimization techniques, ensemble methods and initialization methods applied to neural systems are also worth of investigating, especially in view of the computational complexity of genetic algorithms. Several such methods have been proposed in this paper.

Acknowledgement: I would like to thank the Polish Committee for Scientific Research, grant no. 8T11F 014 14 for partial support.

References

[1] R. Horst and P.M. Pardalos (eds), *Handbook of global optimization.* Kluwer, Dodrecht 1995; R. Horst and H. Tuy, *Global optimization*. Springer Verlag, 1990; J.D. Pinter, *Global optimization in action*. Kluwer, Dodrecht 1996.

[2] Y. Shang and B.W. Wah, *Global optimization for neural network training*. IEEE Computer, 29: 45-54, 1996.

[3] S.E. Fahlman and C. Lebiere, The Cascade-Correlation learning architecture. In *Advances in Neural Information Processing Systems*, vol. 2, Morgan Kaufmann, pp. 524-532, 1990.

[4] Z. Michalewicz, *Genetic algorithms+data structures=evolution programs*. 3rd ed, Springer, Berlin, 1996.

[5] S. Kirkpatrick, C.D. Gellat Jr. and M.P. Vecchi, Optimization by simulated annealing. *Science*, 220: 671-680, 1983.

[6] W. Duch, R. Adamczak and N. Jankowski, Initialization and optimization of multilayered perceptrons. In *Third Conference on Neural Networks and Their Applications*, pp.

99-104, Kule, Poland, Oct. 1997; Initialization of adaptive parameters in density networks, ibid, pp. 105-110.

[7] W. Duch and R. Adamczak, Statistical methods for construction of neural networks. *International Congress on Neural Information Processing*, Kitakyushu, Japan, Oct. 1998, pp. 629-642.

[8] J. Schmidhuber and S. Hochreiter, Guessing can outperform many long time lag algorithms. *Technical Note* IDSIA-19-96, 1996.

[9] F-M. Dittes, *Optimization of rugged landscapes: a new general purpose Monte Carlo approach*. *Physical Review Letters*, 76: 4651-4655, 1996.

[10] L. Ingberg, Adaptive simulated annealing (ASA): Lessons learned. *J. Control and Cybernetics*, 25: 33-54, 1996.

[11] L. Herault, Rescaled simmulated annealing. In *World Congress of Computational Intelligence*, pp. 1239-1244, IJCNN'98 Proceedings, Anchorage, Alaska, May 1998.

[12] K.P. Unnikrishnan, K.P. Venugopal, Alopex: a correlation-based learning algorithm for feeedforward and recurrent neural networks. *Neural Computations*, 6: 469-490, 1994.

[13] R. Battiti and G. Tecchiolli, Training neural nets with the reactive tabu search. *Transactions on Neural Networks*, 6: 1185-1200, 1995.

[14] H.V. Gupta, K. Hsu and S. Sorooshian, Superior training of artificial neural networks using weight-space partitioning. In *Proc. of International Conference on Neural Networks*, pp. 1919-1923, Houston, USA, 1997.

[15] B. Orsier, A new global optimization method: "Rolling-Stone Scheme" and its applications to supervised learning of multi-layered perceptrons. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks*, Vol.2, pp. 395-398, North-Holland, Amsterdam 1992.

[16] J. Kennedy and R.C. Eberhart, Particle swarm optimization. In *Proc. of the IEEE International Conf. on Neural Networks*, Piscataway, New Jersey, 1995, Vol. 4, pp. 1942-1948

[17] L. Piela, J. Kostrowicki and H.A. Szeraga, The multiple-minima problem in comformational analysis of molecules. Deformation of the potential energy hypersurface by the diffusion equation method. *Journal of Physical Chemistry*, 93: 3339-3346, 1989.

[18] E. Hansen, *Global optimization using interval analysis*. Dekker, New York 1992.

[19] W. Duch, R. Adamczak and K. Grąbczewski, Extraction of logical rules from backpropagation networks. *Neural Processing Letters* 7: 1-9, 1998.

[20] M.P. Perrone and L.N. Cooper, When neural networks disagree: ensemble methods for hybrid neural networks. In: *Artificial neural networks for speech and vision*, ed. J. Mammone, London, Chapman and Hall 1993, pp. 126-142

[21] L. Breiman, Bagging predictors. *Machine Learning* 26 (1996) 123-140

[22] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decission trees: bagging, boosting and randomization. *Machine Learning* 28 (1998) 1-22