INNC 90 PARIS

Volume 2

INTERNATIONAL NEURAL NETWORK CONFERENCE

JULY 9-13, 1990
PALAIS DES CONGRES - PARIS - FRANCE

UNDER THE PATRONAGE OF THE COMMISSION OF THE EUROPEAN COMMUNITIES

THE INTERNATIONAL NEURAL NETWORK SOCIETY (INNS), THE IEEE NEURAL NETWORK COUNCIL COOPERATING SOCIETIES

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

KLUWER ACADEMIC PUBLISHERS
DORDRECHT / BOSTON / LONDON

THE ORGANISATION

The conference INNC-90-PARIS held at the Palais des Congrès from July 9 - 13, 1990 is divided into four main parts with 24 sessions:

- Applications (Image, Speech and Signal Processing, Robotics and Control, Optimization, Classification, Decision).

- Implementation (Neurobiological models, Cognitive science, Electronic and Optical neurocomputers, Implementations on Parallel Hardwares, Languages and Development environment).

- Theory (Supervised and Unsupervised Learning, Associative Memories, Architectures, Analysis of Network Dynamics, Statistics).

- Commercial (European and Government programs, Commercial Products, Company Strategy).

The trade show to display the latest innovations on neural networks will have approximately 40 exhibitors.

On July 9th, nine tutorials will be given by the world's leading specialists in the field.

453 papers were received from 35 countries (half from Europe):

. 221 papers were accepted for oral presentation.

. 164 papers were accepted for poster presentation.

Three-fourths of the papers were from scientists, one-fourth from industrialists.

We wish to thank for their help and their confidence: the INNS, the IEEE, the sponsors, the chairs, the speakers, the exhibitors, the reviewers, the participants and particularly the THOMSON, Bernard ANGENIOL and Bernard WIDROW who made this first large European Conference on Neural Networks supported by the INNS and the IEEE possible.

Nina THELLIER
INNC-90-PARIS Conference manager

INNC-90-PARIS CONFERENCE COMMITTEE

CONFERENCE CO-CHAIRS:

Bernard WIDROW: Professor at Stanford University, President INNS
IOL: THOMSON-CSF, Chairman of Pygmalion, Esprit Project on Neural Networks

PROGRAM CHAIR:

Teuvo KOHONEN: Professor at the Helsinki University of Technology

PROGRAM COMMITTEE:

Igor ALEKSANDER: Professor at the Imperial College London
Shun-Ichi AMARI: Professor at the University of Tokyo
PER: Nobel Prize laureate, Professor at Brown University, co-founder of Nestor Inc.
Rolf ECKMILLER: Professor at the University of Dusseldorf
Françoise FOGELMAN: Professor at the University of PARIS XI
Stephen GROSSBERG: Professor at Boston University
Federico FAGGIN: President and founder of SYNAPTICS
Philip TRELEAVEN: Professor at the University College London
Fernando ALDANA: Vice-president Universidad Politechnica de Madrid

CONFERENCE MANAGEMENT:

Nina THELLIER, NTC: 19, rue de la Tour, 75116 PARIS

ISBN 0-7923-0831-X
IEEE Catalog Number 90-TH-0318-6

Published by Kluwer Academic Publishers,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

Kluwer Academic Publishers incorporates
the publishing programmes of
D. Reidel, Martinus Nijhoff, Dr W. Junk and MTP Press.

Sold and distributed in the U.S.A. and Canada
by Kluwer Academic Publishers,
101 Philip Drive, Norwell, MA 02061, U.S.A.

In all other countries, sold and distributed
by Kluwer Academic Publishers Group,
P.O. Box 322, 3300 AH Dordrecht, The Netherlands.

Printed on acid-free paper

Printed in the Netherlands

BENCHMARKING

- Chair: Philip TRELEAVEN

* ORAL PRESENTATION

SUPERVISED LEARNING

- Chairs: Yann LE CUN and Luis BORGES de ALMEIDA

* ORAL PRESENTATION

BFGS Optimization for Faster and Automated Supervised Learning

Roberto Battiti and Francesco Masulli
Dipartimento di Fisica, Universita' di Genova
Via Dodecaneso 33, 16146 Genova, Italy
Computer address: battiti%genova.infn.it@iboinfn.bitnet

Abstract

Standard back-propagation learning (BP) is known to have slow convergence properties. Furthermore no general prescription is given for selecting the appropriate learning rate, so success is dependent on a trial and error process. In this work a well known optimization technique (the *Broyden-Fletcher-Goldfarb-Shanno* memoryless quasi-Newton method) is employed to speed up convergence and to select parameters. The strict locality requirement is relaxed but parallelism of computation is maintained, allowing efficient use of concurrent computation. While requiring only limited changes to BP, this method yields a speed-up factor of 100 – 500 for the medium-size networks considered.

Comparisons are done with a version of BP employing learning rate adaptation. This last method is in itself interesting, since it converges in a number of iterations close to that of optimized BP, with no need for parameter optimization.

1 Introduction

Back-propagation learning [8] is a gradient-descent method: in a given iteration the *search direction* is given by the negative gradient of the energy, while the step along this direction is given by a constant (*learning rate*) chosen by the user.

Now, it is well known from the optimization literature that pure gradient-descent methods tend to be very inefficient [3]. A case in which this happens is when "the search space contains long ravines that are characterized by sharp curvature across the ravine and a gently sloping floor" [8].

A recent overview of other heuristics employed to accelerate back-propagation (like the *momentum term*) has been presented in [5]. Unfortunately up to now there are no good general prescriptions for selecting the parameters defining the optimization strategy. It is usually left to the user to find a good or optimal combination of these parameters that leads to avoidance of local minima and fast convergence times. This process of *meta-optimization* (optimization of the optimization method) leads in general to a sizeable waste of computational resources.

The focus of this work has been on transferring some *meta-optimization* techniques to the learning algorithm itself. Since this involves measuring optimization performance and correcting some parameters while the optimization algorithm is running, some *global* information is required. *Parallelism* of computation is nonetheless maintained, resulting in efficiency close to 100% when the algorithm runs on a parallel computer.

In all cases the "standard" back-propagation algorithm is used to find the value of the negative gradient for a given configuration. The differences are in the definition of the *search direction* and/or in the selection of a *step size* along the selected direction.

In the "bold driver" method the search direction remains equal to the negative gradient but the step size is adapted during the computation. In the BFGS method both the search direction and the step size are changed in a very efficient way.

In both cases the network is updated only after the entire set of patterns to be learned has been presented to it.

2 The "Bold Driver" Method

This strategy has been suggested independently in [10,6,1]. The proposed heuristic is to start with a given learning rate and to monitor the value of the energy function $E(\mathbf{w}_n)$ after each learning cycle. If E decreases, the learning rate is increased by a factor ρ. Vice versa if E increases, this is taken as an indication that the step made was too long, the learning rate is decreased by a factor σ, the last change is cancelled and a new trial is done. The process of reduction is repeated until a step that decreases the energy value is found.

Heuristically, ρ has to be close to unity in order to avoid frequent "accidents", because the computation done in the last back-propagation step is wasted in these cases. We choose $\rho = 1.1$ and $\sigma = 0.5$.

The performance of this apparently "quick and dirty" method is close to and usually better than that obtainable by optimizing a learning rate that is to remain fixed during the procedure.

3 The BFGS Memoryless Quasi-Newton Method

Shanno [9] reviews several variations of the conjugate gradient methods for function minimization and suggests one method that "substantially outperforms known conjugate gradient methods on a wide class of problems". In the "one-step Broyden-Fletcher-Goldfarb-Shanno memoryless quasi-Newton" method the search direction for the n'th iteration is defined as

$$\mathbf{d}_n = -\mathbf{g}_n + A_n \mathbf{p}_n + B_n \mathbf{y}_n \tag{1}$$

where $\mathbf{g}_n = \nabla E(\mathbf{w}_n)$ is the gradient, the coefficients A_n and B_n are defined as follows:

$$A_n = -\left(1 + \frac{\mathbf{y}_n \cdot \mathbf{y}_n}{\mathbf{p}_n \cdot \mathbf{y}_n}\right) \frac{\mathbf{p}_n \cdot \mathbf{g}_n}{\mathbf{p}_n \cdot \mathbf{y}_n} + \frac{\mathbf{y}_n \cdot \mathbf{g}_n}{\mathbf{p}_n \cdot \mathbf{y}_n} \quad , \quad B_n = \frac{\mathbf{p}_n \cdot \mathbf{g}_n}{\mathbf{p}_n \cdot \mathbf{y}_n} \tag{2}$$

and the vectors used are differences between subsequent weights and gradients: $\mathbf{p}_n = \mathbf{w}_n - \mathbf{w}_{n-1}$ and $\mathbf{y}_n = \mathbf{g}_n - \mathbf{g}_{n-1}$. Every N steps (N being the number of weights in the network) the search is restarted in the direction of the negative gradient .

Successive approximations $\mathbf{w}_n$ to the minimizer $\mathbf{w}^*$ of $E(\mathbf{w})$ are generated using a fast one-dimensional minimization along the search direction:

$$\epsilon_n = \min_{\epsilon} E(\mathbf{w}_n + \epsilon\ \mathbf{d}_n) \tag{3}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \epsilon_n\ \mathbf{d}_n \tag{4}$$

The one-dimensional minimization used in this work is based on quadratic interpolation and tuned to back-propagation where in a single step both the energy value and the negative gradient can be efficiently obtained. A small number of energy evaluations is sufficient [1].

This method has shown a consistent superior performance on different test problems. The results for two examples are presented in the following sections[1].

[1] A similar optimization approach, using Polak-Ribiere optimization, is presented in [4]. Now, a major difficulty with the Polak-Ribiere algorithm is that the search directions obtained are not necessarily *descent* directions, and numerical instability can result.

4 Test: the Dichotomy Problem

This problem consists in classifying a set of randomly generated patterns in two classes. It has been demonstrated in [2] that an arbitrary dichotomy for any set of N points in general position in d dimensions *can* be implemented with a network with one hidden layer containing $\lceil N/d \rceil$ neurons. In this test the pattern coordinates are random values belonging to the [0-1] interval.

A dichotomy problem is defined by the number of patterns generated. The dimension of the space and the number of inputs is two, the number of middle-layer units is $\lceil N/2 \rceil$ by the above criterion and one output unit is responsible for the classification.

Simulation runs have been made starting from small random weights, with maximum size r equal to 0.1. Correct performance is defined as coming within a margin of 0.1 of the correct answer.

Average results for different test runs (the random number seed is changed) are given in Fig. 1. Cases for correct solutions or local minima are shown separately. The speed-up factor increases for bigger networks (the largest value obtained is 116).

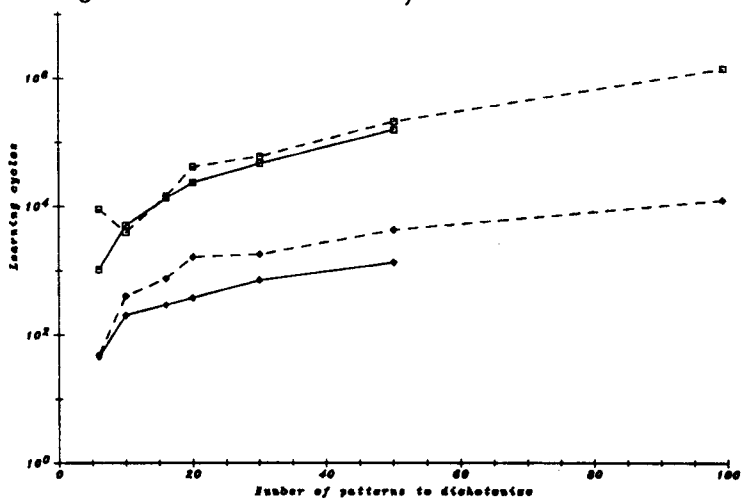

Figure 1: Performance comparison: "bold driver" method (□) versus BFGS optimization (◇). Continuous lines show the average number of cycles for convergence to correct solution, dashed lines for convergence to local minimum.

5 Test: Predicting the Logistic Map

Back-propagation has been applied to temporal series prediction in [7], where the logistic map is used for some tests. The logistic map is a discrete-time, nonlinear dynamical system. The value of a variable x_n is mapped to a new one x_{n+1} according to the recurrence relation

$$x_{n+1} = r x_n (1 - x_n) \tag{5}$$

where $0 < x_n < 1$. When $r = 4$, given an initial value $0 < x_n < 1$, the generated sequence is ergodic and chaotic. Although simple, this map captures the essence of a whole class of real-world phenomena. In the biological context, the parameter r is related to the fertility, while the factor $(1 - x_n)$ is viewed as a natural limitation of resources.

A 3-layer feed-forward neural network as described in [7] is used in order to make the comparison. It consists of one input, five hidden and one output unit.

BD		BFGS		speedup (BD/BFGS)
cycles (st.dev.)	[generalization score]	cycles (st.dev.)	[generalization score]	
1310680 (1564720)	[0.0794]	**2638** (2242)	[0.0346]	496

Table 1: Results for logistic map prediction.

A sequence of 10 example pairs (x_n, x_{n+1}) generated by the logistic map is used as the training set. After convergence (when all the desired outputs are reproduced with a maximum error of 0.01) the network is tested for generalization by presenting to it a new sequence of $N = 500$ example pairs and calculating the root-mean-square prediction error ("generalization score").

The average convergence and generalization results for 10 different networks are shown in table 1. The speed-up factor is almost 500 [2], while the generalization score obtained with the BD and BFGS methods are comparable (actually a better score is obtained with BFGS). No significant difference has been observed between the representations obtained with the two teaching methods.

Acknowledgments

Part of this work was completed while one of the authors (R. Battiti) was at the California Institute of Technology and was supported by DOE grant DE-FG-03-85ER25009, the National Science Foundation with grant IST-8700064 and by IBM.

References

[1] R. Battiti, "Accelerated Back-propagation Learning: Two Optimization Methods," *Complex Systems*, in press.

[2] E. B. Baum, "On the Capabilities of Multilayer Perceptrons," *Journal of Complexity* **4**, 193–215 (1988).

[3] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization* (Academic Press,1981).

[4] A. H. Kramer, A. Sangiovanni-Vicentelli, "Efficient Parallel Learning Algorithms For Neural Networks," *Advances in Neural Information Processing Systems* Vol. **1**, 75–89 (Morgan Kaufmann, CA, 1988).

[5] R. A. Jacobs, "Increased Rates of Convergence Through Learning Rate Adaptation," *Neural Networks* **1**, 295–307 (1988).

[6] A. Lapedes and R. Farber, "A self-optimizing, nonsymmetrical neural net for content addressable memory and pattern recognition," *Physica* **22 D**, 247–259 (1986).

[7] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modeling," Los-Alamos Preprint LA-UR-87-1662.

[8] D. E. Rumelhart and J. L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, (MIT Press,1986).

[9] D. F. Shanno, "Conjugate gradient methods with inexact searches," *Mathematics of Operations Research* **3 - 3**, 244–256 (1978).

[10] T.P. Vogl, J.K. Mangis, A.K. Rigler, W. T. Zink and D. L. Alkon, "Accelerating the Convergence of the Back-Propagation Method," *Biological Cybernetics* **59**, 257-263 (1988).

[2] The actual computing time for BFGS is some minutes on a Sun workstation.