ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 15-20

TRACKING GLOBAL MINIMA USING A RANGE EXPANSION ALGORITHM

D Gorse and A Shepherd

Department of Computer Science, University College London

J G Taylor

Department of Mathematics, King's College London

Abstract. Conventional methods of supervised learning are inevitably faced with the problem of local minima; evidence is presented that networks trained using conjugate gradient algorithms may be particularly susceptible to being trapped in a sub-optimal solution. A new technique is described which manipulates the range of the function to be learned in such a way that the network is able to remain in or near a global minimum at each step of training. Progressive expansion of the range toward the original function definition enables the network to move across the error-weight surface in a controlled way and thus to reach a superior solution. The method has applicability to a wide range of supervised learning problems, and is not dependent on the details of any specific training algorithm.

1. Introduction

The problems to which neural computing techniques are most frequently applied involve the supervised learning of an input-output mapping, defined implicitly by a set of P input patterns together with their desired outputs. Such tasks can be formulated as error-minimisation problems, where the error function is usually given by

$$E = \frac{1}{2P}\sum_{p=1}^{P} E_p = \frac{1}{2PN}\sum_{p=1}^{P}\sum_{i=1}^{N}(t_{i,p} - z_{i,p})^2$$

where $t_{i,p}$ and $z_{i,p}$ are the desired and actual values of the ith output unit for pattern p, for a network with N output units. E is a function of all the parameters (weights and thresholds) of the network.

The minimisation of a multidimensional function such as E above is a well-studied problem in numerical analysis, for which many techniques exist. Steepest descent has long been regarded as an inefficient technique by numerical analysts, yet it is still the method most often used in neural network applications (implemented by the well-known method of error backpropagation). However there has recently been a growing interest in the use of neural implementations of conjugate gradient algorithms as an alternative to gradient descent, as discussed for example in [1]. These are generally much faster and can produce impressively low error levels - although as

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 15-20

will be seen later may be particularly likely to be trapped in local minima. Conjugate gradient algorithms gain their advantage from the use of second-order information about the shape of the error-weight surface. Explicit computation of the Hessian matrix may be avoided if the computation of successive search directions is done using a line minimisation algorithm; in the simulations to be presented here the Brent algorithm (a combination of parabolic interpolation and golden section search) was used. Further details of conjugate gradient techniques and line minimisation algorithms may be found for example in [2].

2. Local minima in the XOR problem

In order to investigate the performance of conjugate gradient techiques in comparison to more commonly used training algorithms the familiar example of the XOR problem was chosen. The experiments used the minimal (2,2,1) architecture with two inputs x_1, x_2, two hidden units with outputs y_1, y_2 and weights $\underline{w}_1$, $\underline{w}_2$, and a single output layer unit with classification output z and weights $\underline{a}$. The target output d_p was 0 for input patterns (0,0) and (1,1), 1 for patterns (0,1) and (1,0).

XOR was chosen for several reasons. Firstly, the problem has been well studied by other workers, with results available for many variations on the standard learning algorithms. Secondly, it is a 'small' problem, so it is feasible to run a large number of trials starting from different sets of initial weights. Finally - and most importantly in this context - XOR is the only classification problem we are aware of for which an analytic solution for the local minima has been obtained. Lisboa and Perantonis [3] give a complete solution to this problem for several networks including the (2,2,1) architecture used here. It is important to know where the local minima lie in order to know for certain that a given net is indeed trapped, as opposed to simply having encountered a section of the error-weight surface which appears flat to within the precision of the host machine.

Three learning algorithms were used: on-line backpropagation (BP), batched BP, conjugate gradients (CG) with Brent line minimisation and Polak-Ribiere update. Over 100 trials, 97% of the on-line BP nets succeeded in solving the problem, compared to a figure of 83% for batched BP, and only 55% for CG. However the final error levels for the successful CG nets were very much lower: an average of $E \in O(10^{-11})$, as opposed to $O(10^{-4})$ for both forms of BP. The results for on-line BP are unsurprising, and similar to those reported by many other workers for this architecture. On-line BP appears significantly more successful than batch mode; it is likely that the departure from strict gradient descent introduced by single-pattern updates serves as a useful source of noise that helps avoid local minima [4]. Is it possible to retain the precision of CG techniques, but modify the algorithm in such a way as to improve robustness in the face of local minima? We have experimented with various ideas: the addition of annealing techniques, forms of 'on-line' CG using overlapping pattern sets, hybrid BP/CG algorithms [5], [6]. However the most successful modification to the CG algorithm, which will be described below, has much wider implications since it deals with the problem of local minima in a very general context, independently of the details of the training algorithm.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 15-20

3. Tracking global minima by range expansion

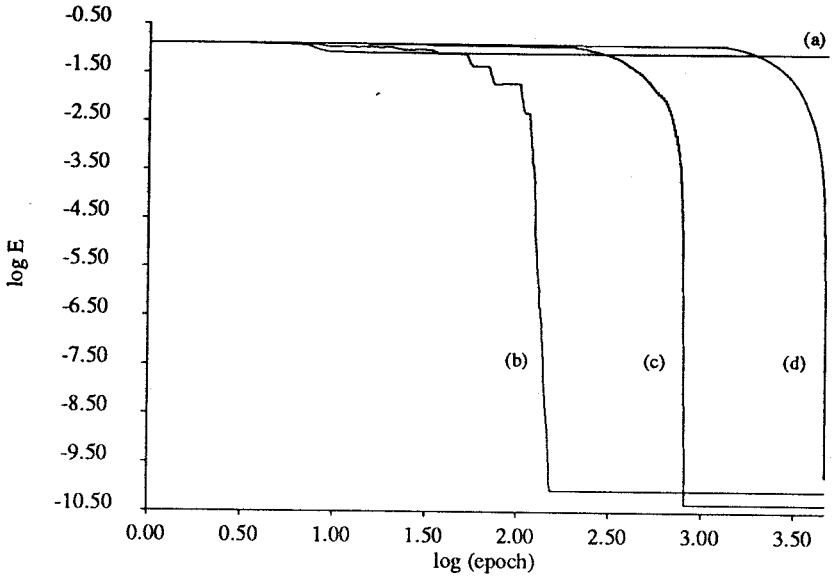

Fig. 1. Training curves for decreasing step size ε: (a) ε = 1.0, (b) ε = 0.2, (c) ε = 0.02, (d) ε = 0.002.

Consider a modified version of the XOR problem, in which the target output for pattern p is defined by $d_p(\eta) = 0.5 + \eta(d_p - 0.5)$ where the d_p are the binary target values of the original problem and $\eta \in [0,1]$. It was observed that with η small the net was much less likely to be trapped in a local minimum. This leads to the idea of starting with a small η, solving the problem within this compressed range, taking a larger η, training again to convergence...until $\eta = 1.0$ (original XOR problem solved). Simulations were carried out of a 7-stage process with $\eta_0 = \frac{1}{127}$, $\eta_k = (2^{k+1} - 1)\eta_0$, k=1..6. In this experiment 100% of trials converged to an average error of $O(10^{-10})$, compared with 55% for the original single-stage CG training process.

The XOR problem has relatively few local minima [3]. More generally it would be advisible to expand the function range in uniform steps of ε: $\eta(n) = n\varepsilon$, $n = 1..\frac{1}{\varepsilon}$ Simulations were also carried out for the XOR problem with ε = 1.0 (original single-step problem), ε = 0.2 (5 steps), ε = 0.02 (50 steps), ε = 0.002 (500 steps). Typical results (using the same set of initial weights for each ε) are shown in Figure 1. All three versions of the range expansion algorithm ((b)-(d)) succeed in solving the problem, but the original single-step CG algorithm (a) is clearly trapped in a local minimum.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 15-20

3.1 Predicting the weight changes

It can be shown that for sufficiently small step sizes there is a global minimum (and a unique solution for the 'next step' weights) at each step of training. Progressive expansion of the range η allows this global minimum to be tracked up to $\eta = 1$ (solution of original problem). We assume the problem has been solved for a range expansion η, and consider the effect of expanding the range by a further small quantity ε. The error at expansion $\eta+\varepsilon$ is

$$E(\eta+\varepsilon) = \frac{1}{2P}\sum_{p=1}^{P} [d_p(\eta+\varepsilon) - z_p(\eta+\varepsilon)]^2 \qquad (1)$$

where the desired output for pattern p at this point is given by

$$d_p(\eta+\varepsilon) = d + (\eta+\varepsilon)(d_p - d) = d_p(\eta) + \varepsilon(d_p - d) \qquad (2)$$

In this expression the d_p are the $\eta = 1$ (uncompressed) targets, and d is the mean of these fully expanded target values. The network weights at the start of training when $\eta = 0$ must be adjusted so that the output of the net (we assume here that there is just one output node, though the method clearly generalises to mappings to multidimensional target patterns) is equal to d for all input patterns in the training set. This is most easily achieved by setting all output unit weights except the bias to zero (the initial hidden weights can then be randomly chosen) and adjusting the bias a_0 so that $z_p(a_0) = d$.

To $O(\varepsilon)$ the output of the classifying node for pattern p is given by

$$z_p(\eta+\varepsilon) = z_p(\eta) + \varepsilon\, z'_p(\eta)\,\{\sum_{i=0}^{H} a'_i(\eta) y_{i,p}(\eta) + \sum_{i=1}^{H} a_i(\eta) y'_{i,p}(\eta) \sum_{j=0}^{H} w'_{ij}(\eta)\, x_{j,p}\} \qquad (3)$$

where

$$a'_i(\eta) \approx \frac{a_i(\eta+\varepsilon) - a_i(\eta)}{\varepsilon}, \quad w'_{ij}(\eta) \approx \frac{w_{ij}(\eta+\varepsilon) - w_{ij}(\eta)}{\varepsilon}$$

and the derivatives of the neuron outputs are in this case given by $z'_p = z_p(1 - z_p)$, $y'_{i,p} = y_{i,p}(1 - y_{i,p})$.

Using (2), (3) in (1) and differentiating w.r.t. a'_i, w'_{ij} allows $a_i(\eta+\varepsilon)$, $w_{ij}(\eta+\varepsilon)$ to be calculated using $y_{i,p}(\eta)$, $z_p(\eta)$ and their own previous values:

$$\frac{\partial E(\eta)}{\partial a'_i} = -\frac{\varepsilon}{P}\sum_{p=1}^{P} z'_p(\eta)\, y_{i,p}(\eta) \times [d_p(\eta) + \eta(d_p - d) - z_p(\eta) - ...] \equiv -\frac{\varepsilon}{P} f_i(\eta) \qquad (4)$$

$$\frac{\partial E(\eta)}{\partial w'_{ij}} = -\frac{\varepsilon}{P}\sum_{p=1}^{P} z'_p(\eta)\, a_i(\eta)\, y'_{i,p}(\eta)\, x_{j,p} \times [\ \cdots\ ] \equiv -\frac{\varepsilon}{P} f_{ij}(\eta) \qquad (5)$$

The quantity in the square brackets in (4), (5) is given by inserting (2), (3) into the expression $d_p(\eta+\varepsilon) - z_p(\eta+\varepsilon)$. Setting $f_i(\eta) = 0$, $f_{ij}(\eta) = 0$ gives 9 linear equations for the $a_i(\eta+\varepsilon)$, $w_{ij}(\eta+\varepsilon)$; it would be possible to solve these equations explicitly, but the process would be computationally expensive if there were many sets of weights to be

calculated and compared. For this reason we chose to assess the error in the above procedure by substituting successive pairs of weight values (obtained from simulation) into $f_i(\eta)$, $f_{ij}(\eta)$, defining an appropriate error function by

$$F(\eta) = \left[\frac{1}{9} \{ \sum_{i=0}^{H} f_i^2(\eta) + \sum_{i=1}^{H} \sum_{j=0}^{H} f_{ij}^2(\eta) \} \right]^{\frac{1}{2}}$$

F does not contain explicit factors of ε; a small value for the rms error F throughout training gives a reliable indication that the network is indeed tracking a global minimum. Figure 2 shows for the XOR problem how the value of F decreases with ε, for a single randomly chosen test net learning to solve the XOR problem, and that this effect persists throughout training. Figure 3 examines the behaviour of F with ε at the first step of training (it was not practicable to complete full training runs for the very small values of ε used in Figure 3). It can be seen that for small ε the system does appear to be behaving according to the analysis above, remaining in or near a global minimum throughout training. At higher orders of perturbation theory the error-weight surface begins to pick up more complex structures, so that more care is needed in training for larger step sizes ε. For example at $O(\varepsilon^2)$ there are linear equations for the w'_{ij} and for the second-order corrections a''_i, but the a'_i are now given by a *cubic* equation - there are now two minima available to the network, a global minimum and a local one. However the structure of the error-weight surface is *known*, which gives the possibility of avoiding the local minimum. Thus even if it is not practicable to use very small training steps this method has the potential of delivering more predictable results for difficult optimisation problems, and hopefully to significantly improve final error levels.

References

1. J.A. Kinsella: Comparison and evaluation of variants of the conjugate gradient method for efficient learning in feed-forward neural networks with backward error propagation. Neural Networks, 3, 27-35 (1992).
2. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling: Numerical Recipes in Pascal. CUP (1990).
3. P.J.G. Lisboa and S.J. Perantonis: Complete solution of the local minima in the XOR problem. Network, 2, 119-124 (1991).
4. C. Darken and J. Moody: Towards faster stochastic gradient search. In: Advances in Neural Information Processing Systems, 4, Morgan Kauffman, San Mateo, California, 1009-1016 (1991).
5. A. Shepherd: Towards a hybrid conjugate gradient/backpropagation algorithm for training feed-forward neural networks. M.Sc Thesis, Department of Computer Science, University College London (1992).
6. D. Gorse and A. Shepherd. Adding stochastic search to conjugate gradient algorithms. Neural Network World, 2, 599-605 (1992).

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 15-20

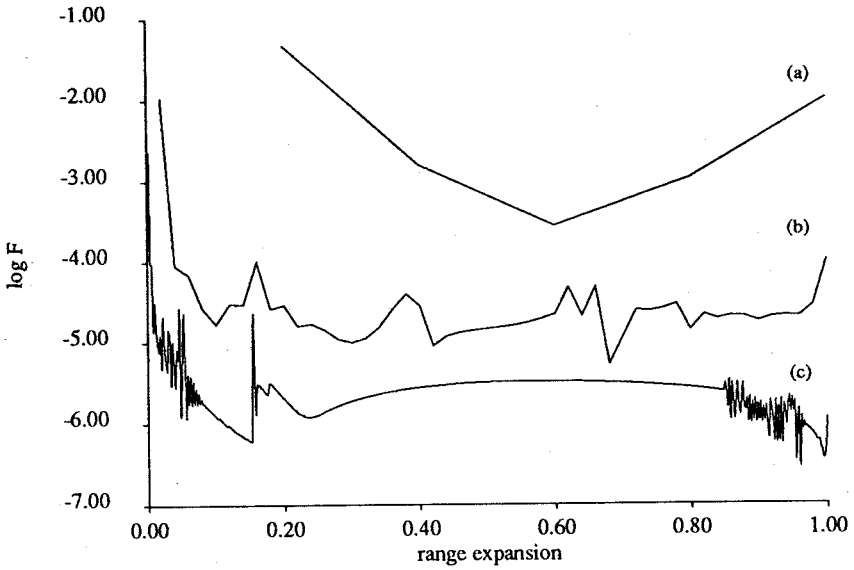

Fig. 2. Error F as a function of range expansion η, for decreasing step size ε: (a) $\varepsilon = 0.2$, (b) $\varepsilon = 0.02$, (c) $\varepsilon = 0.002$.

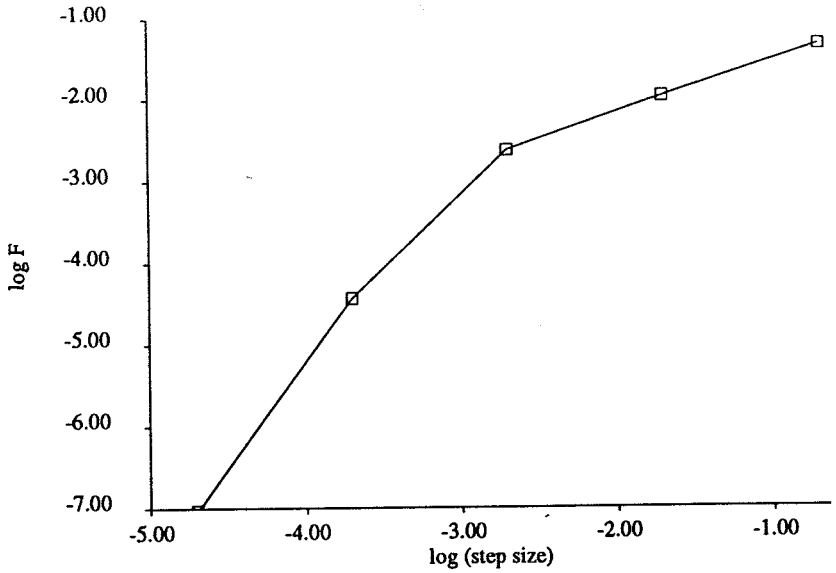

Fig. 3. Error F at the first step of training as a function of step size ε.