A CLASSICAL ALGORITHM FOR AVOIDING LOCAL MINIMA

D Gorse and A Shepherd
Department of Computer Science
University College, Gower Street, London WC1E 6BT, UK

J G Taylor
Department of Mathematics
King's College, Strand, London WC2R 2LS, UK

Conventional methods of supervised learning are inevitably faced with the problem of local minima; evidence is presented that conjugate gradient and quasi-Newton techniques are particularly susceptible to being trapped in sub-optimal solutions. A new classical technique is presented which by the use of a homotopy on the range of the target outputs allows supervised learning methods to find a global minimum of the error function in almost every case.

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

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

where $d_{i,p}$ 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; this parameter list can be written as a multidimensional vector $\underline{w}$. The problem is to change $\underline{w}$ so as to avoid those solutions of the minimisation condition $\partial E/\partial \underline{w} = 0$ which do not correspond to the lowest value of E, the local minima of the error-weight surface. The most commonly used supervised training technique, error backpropagation (BP) (equivalent to gradient descent with a fixed step length) is well known to have difficulties with local minima, especially for non linearly separable problems [1]. What is less well known is that the neural implementations of more efficient classical minimisation algorithms, such as conjugate gradients (CG) or the quasi-Newton method (QN), are even more likely to be trapped in sub-optimal solutions. Table 1 shows the percentage success in reaching a global minimum for 100 (2-2-1) networks learning to solve the XOR problem.

XOR	% reaching global minimum	
method	sigmoid in final layer	linear in final layer
on-line BP	85	95
batched BP	75	96
CG	51	80
QN	34	66

Table 1

XOR is a useful benchmark because it is a non linearly separable problem with known local minima [2], but one which can be solved by a small network with only 9 adaptive weights. Linear outputs in the second layer (as opposed to sigmoidal squashing for both computational layers) improve the percentage success, but there is a clear trend toward worsened performance for the more sophisticated algorithms. Simple on-line BP (without momentum) performs best; this may be due to the method's stochastic features, as discussed in [3].

Trapping in local minima can also be observed for continuous function learning problems. McInerney et al [4] have discovered (by exhaustive search of the error-weight surface) local minima in a (1-2-1) network (with a linear output node) learning the sine function. This problem was also investigated, using the same training set as in [4], and the results are summarised in Table 2.

sine	% reaching global minimum
batched BP	100
CG	96
QN	87

Table 2

These results do not show as high a probability of trapping in local minima as in the XOR example, but there is still a significant correlation between the probability of failure and the convergence speed of the method; the quasi-Newton method, with a 13% failure rate, would probably not be a good choice unless multiple restarts were acceptable.

It is commonly believed - though we do not know of any 'no-go theorem' to this effect - that the only techniques guaranteed to converge to a global minimum with a probability approaching 1 are stochastic in character, with methods based on simulated annealing [5] and, currently, genetic algorithms [6] being among the most popular. However these techniques can be very slow and must be applied carefully in order to ensure a good solution. Is there a way to retain the fast convergence of techniques like conjugate gradients and the quasi-Newton method whilst improving the robustness of these algorithms in the face of local minima? We will present here a new and purely classical method which is guaranteed to succeed in avoiding local minima in almost all cases.

Expanded range approximation (ERA)

The basic idea underpinning this new algorithm is that of a homotopy on the range of the target values d_p (for simplicity we consider just one output node). This range is modified by compressing these values down to their mean value $\langle d \rangle = \frac{1}{P}\sum_{p=1}^{P} d_p$ and then progressively expanding these compressed targets back toward their original values (hence the epithet 'expanded range approximation', or ERA, we have coined for this approach). We define a modified training set

$$S(\lambda) = \{\underline{x}_p,\ d_p(\lambda)\} = \{\underline{x}_p,\ \langle d \rangle + \lambda(d_p - \langle d \rangle)\}$$

where the $d_p(\lambda)$ are the new, compressed, targets. The problem defined by $S(0)$ is easy for the network to solve (the corresponding error-weight surface can be shown to have only a global minimum); $S(1)$ is the original problem with training set $\{\underline{x}_p,\ d_p\}$. The homotopy parameter λ interpolates between these extremes. A λ-parametrised error function can be defined during training on each of the sets $S(\lambda)$ by

$$E(\lambda) = \frac{1}{P}\sum_{p=1}^{P} [\ \langle d \rangle + \lambda(d_p - \langle d \rangle) - z_p(\lambda)\ ]^2 \tag{2}$$

where the $z_p(\lambda)$ are the actual network outputs during this procedure. Setting $\lambda = 1$ gives $E(1) \equiv E$, the error function (1) in the case of a single output node. The ERA method involves first solving the problem $S(\lambda_1)$ for small λ_1, then the problem $S(\lambda_2)$ with $\lambda_2 > \lambda_1$, and so on up to the original problem $S(1)$. We have usually chosen to increase λ by uniform steps of η; an 'N-step ERA' method refers to the progressive solution of the N problems $S(\lambda_n = n\eta)$ for $n = 1..N=1/\eta$ ('1-step ERA' (η=1) is the conventional single step training technique).

As a first example, the ERA method was applied to the same 100 XOR networks (with sigmoidal output in the final layer) as in Table 1, using the CG algorithm. With 10-step ERA (η=0.1), the success rate improves dramatically from 51% when η=1 to 94%. Figure 1 shows a training curve for a particular set of XOR weights which led to a failure when η=1 (curve (a)), but succeeded with 10-step ERA (curve (b)). The error function plotted for the 10-step ERA case is the square root of E(1), the error with respect to the original, uncompressed targets. The errors $E(\lambda)$ always decrease during ERA steps, but the overall E can show local increases, as is evident from Figure 1.

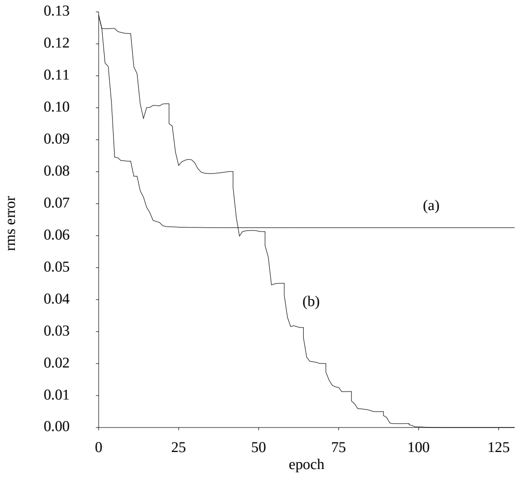

Figure 1

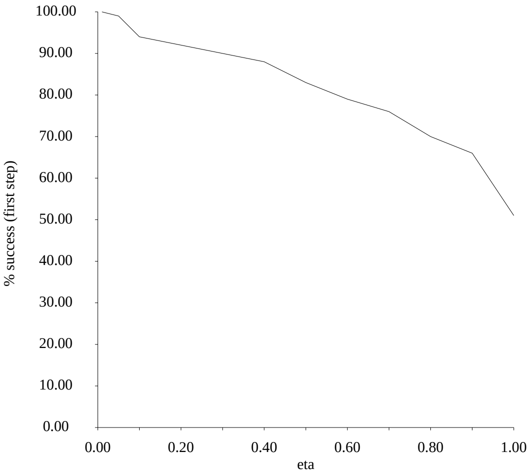

Figure 2

If the step size η is decreased, the percentage success improves still further: 100-step ERA (η=0.01) is 100% successful in solving the XOR problem. As a second example, 2-step ERA (η=0.5) was applied to the sine problem of Table 2, using the QN method. In this case - a continuous as opposed to binary problem, a linear as opposed to sigmoidal output in the final layer, a different training method - there was also a very significant improvement, from 87% success when η=1 to 100% for 2-step ERA.

It might seem that the ERA technique could become computationally expensive if very many small steps were required. In fact we suspect that it will be possible to make a short cut in many cases. We have so far observed that an ERA simulation which fails to find a global minimum of $E(\lambda_1)$ will not subsequently succeed as the homotopy parameter $\lambda \rightarrow 1$. Conversely, however, in our experience a simulation which succeeds at the first step never subsequently fails. This suggests that it is most important to get the first step right, and that subsequent range expansion does not need to be done so carefully. Figure 2 shows the percentage of successes at the first step (successful minimisations of $E(\lambda_1 = \eta)$ as a function of the size of η for the same 100 XOR networks used in the single step (conventional) and 10-step tests. The dependence appears to be roughly linear, with, in this case, 100% success at η=0.01.

All the initial simulations suggested a special role for η, the size of the first step. In order to try to get some further insight into the process, we looked at the trajectories in output space followed for the XOR problem by the $z_p(\lambda_1 = \eta)$, the first-step responses to the four patterns p = 00, 01, 10, 11. Since the initial weights are randomly chosen (from the interval [-1,1]) the trajectories in these experiments begin at some arbitrary point inside the hypercube $[0,1]^4$. The target for η=1 is the point (0,1,1,0); the targets for $\eta < 1$ lie on a line joining this point to (½,½,½,½). By taking pairs of these responses we were able to plot trajectories in the six 2-

dimensional $(z_{p_1}(\eta), z_{p_2}(\eta))$ subspaces during CG training. Figures 3a-d illustrate trajectories in $(z_{00}(\eta), z_{10}(\eta))$ space for η=1.0 (Figure 3a), η=0.3 (3b), η=0.2 (3c), η=0.1 (3d). A coordinate transformation

$$(x,\ y)\ =\ \frac{1}{\eta}(z_{00} - \tfrac{1}{2}(1-\eta),\ z_{10} - \tfrac{1}{2}(1-\eta))$$

is used in plotting the diagrams so that the scales are identical, and in each case the target is the top left hand corner. The midpoint on the y-axis represents a local minimum.

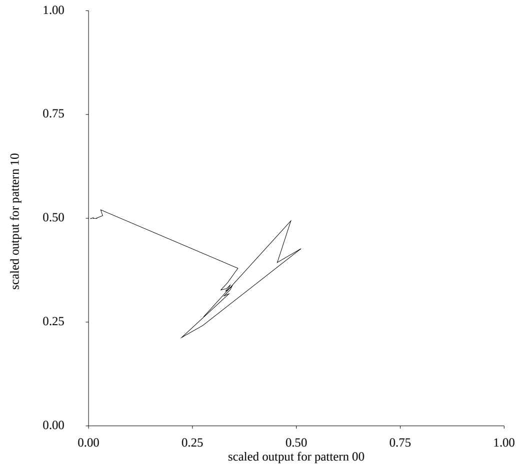

Figure 3a

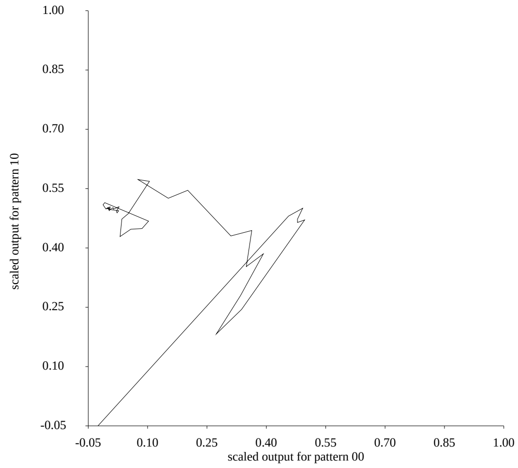

Figure 3b

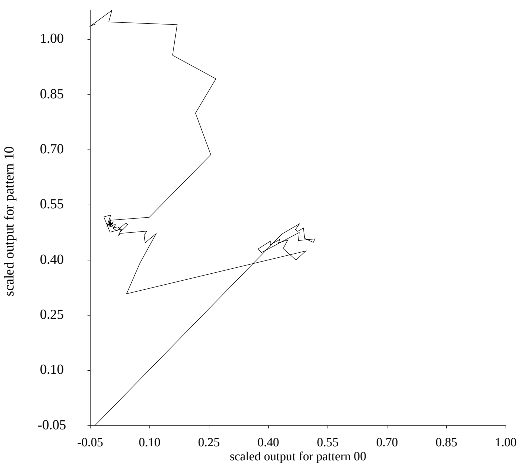

Figure 3c

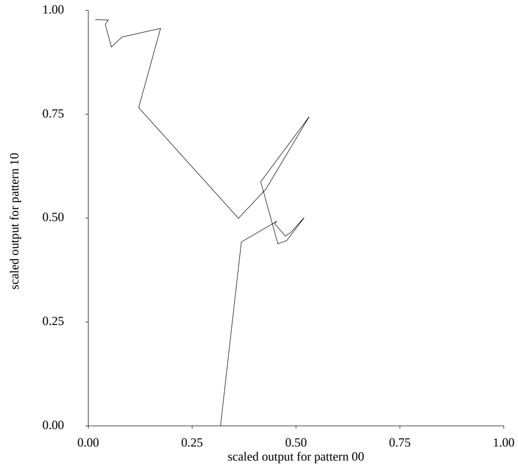

Figure 3d

Figure 3a shows a conventionally trained (η=1) network which fails to solve the XOR problem, becoming trapped in the local minimum corresponding to a final response to the four patterns of $\underline{z}$ = (0, ½, ½, 0). Figure 3b also shows a failure, for η=0.3, but notice that the trajectory appears to almost escape from the local minimum. Figure 3c shows a success at η=0.2, but the trajectory still spends a lot of time in the vicinity of the local minimum before escaping. Finally, Figure 3d, with η=0.1, shows a trajectory which entirely avoids the vicinity of the local minimum, heading more or less directly for the global minimum of $E(\lambda_1 = \eta)$. There appears to be a progressive change of behaviour as η is decreased; this progression is most marked for small values of η.

In order to further investigate this progressive change in the first-step behaviour we looked at the values of the 9 weights which were developed by typical examples of the (2-2-1) XOR networks after minimisation of the first-step error function $E(\eta)$. In Figures 4a,b the weights plotted for the 'test number' -1 are the original, randomly chosen weights $\in$ [-1,1], those for test number 0 represent the solution reached when η=0 (when the error-weight surface has only a global minimum), those for test numbers > 0 the weights $\underline{w}(\eta)$ developed during the minimisation of $E(\eta > 0)$ for progressively larger values of η (using a logarithmic scale). In Figure 4a, tests 1, 2, 3, 4 represent η-values 0.001, 0.01, 0.1, 1.0; in Figure 4b, tests 1, 2, 3 represent η-values 0.00001, 0.0001, 0.001.

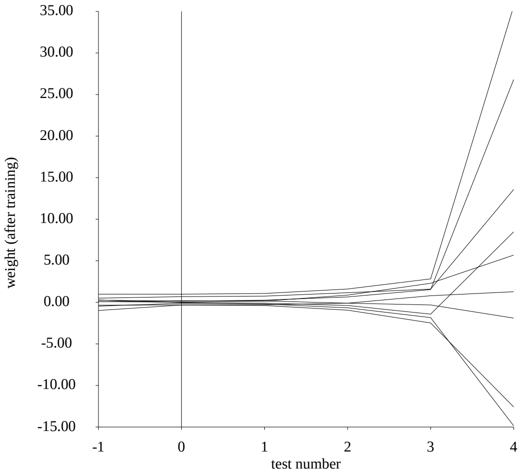

Figure 4a

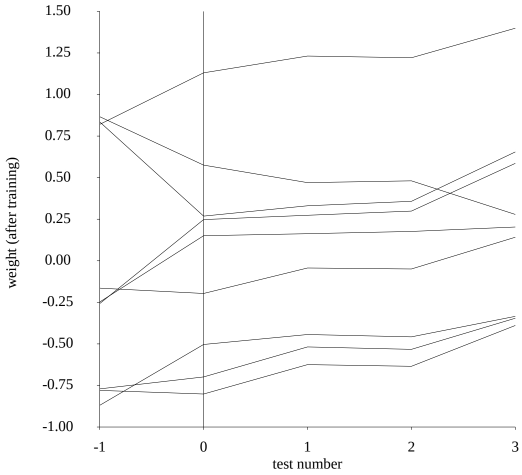

Figure 4b

Figure 4a shows almost linear relationships between the values of the 9 weights and $\log(\eta)$, but with a very striking change in behaviour (possibly representing a phase change of the learning system) at some critical value η_{crit} between 0.1 and 1.0. A somewhat closer investigation shows that in this case $0.1 < \eta_{crit} < 0.2$. We note that this system succeeds in finding the global minimum of E(0.1), but falls into a local minimum of E(0.2); the change in behaviour in $\underline{w}(\eta)$ evident in Figure 4a is clearly related to the switch from success to failure as η increases which is illustrated (for another XOR network with $0.2 < \eta_{crit} < 0.3$) in Figures 3a-d. One criticism that might be levelled at the results shown in Figure 4a is that the weights for small η look very similar to the starting weights - has the network been trained sufficiently in these cases for any meaningful conclusions to be drawn? By looking more closely at the behaviour for very small η, this criticism can be seen to be unfounded. Figure 4b, which uses a different set of starting weights, shows that in general there is a large difference between the starting weights and the η=0 solution, but thereafter a much smoother progression with increasing η.

Discussion

This paper has presented results which violate the widespread belief that the only way to avoid local minima in supervised learning problems with complex error-weight surfaces is to use computationally expensive stochastic procedures like simulated annealing or genetic algorithms. If the results presented here can be shown to be securely founded, and the ERA method shown to have wide applicability, there could be a significant changes in the way that supervised learning tasks are approached. We believe that it is possible to construct a rigorous mathematical proof that the ERA method will work in all but pathological (and rare) cases. The details of this proof are too lengthy to be presented here, but the general principles can be outlined. Initially we look at the first ERA step, for which the homotopy parameter $0 < \lambda \ll 1$. For such small λ, the error $E(\lambda)$ of (2) can be expanded as

$$E(\lambda) = E(0) + \lambda E_1 + O(\lambda^2)$$

where

$$E(0) = \frac{1}{P}\sum_{p=1}^{P} [<d> - z_p(0)]^2, \quad E_1 = \frac{1}{P}\sum_{p=1}^{P} (d_p - <d> - z'_p(0))z_p(0)$$

where the derivative $z'_p(0)$ depends on the particular learning law, but is assumed bounded. It is possible to show that $E(0)$ has only a global minimum. Then we can investigate the shape of the surface $E(\lambda)$ by looking at the effect of the additional term λE_1 for small λ. Outside some small neighbourhood N_0 of the global minimum P_0 of $E(0)$ the effect of this additional term can be made arbitrarily small by choosing λ small enough. In particular it can be shown that no local minima can exist outside N_0, as

$$\lambda \left|\frac{\partial E_1}{\partial \underline{w}}\right| < \left|\frac{\partial E(0)}{\partial \underline{w}}\right| \quad \rightarrow \quad \frac{\partial}{\partial \underline{w}}(E(0) + \lambda E_1) \neq 0$$

Within N_0 the global minimum of $E(0)$ will in general be shifted; there can be degenerate cases where a number of global minima could arise, but this set would be expected to be of measure zero. Continued expansion of the homotopy parameter $\lambda \rightarrow 1$ is handled in a similar way to the first step above. There may be obstructions to performing the homotopy up to λ=1, but we expect that such cases will be rare. Further investigations are in progress, and will be reported in the literature.

References

[1] M Gori and A Tesi, "On the problem of local minima in backpropagation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14, 76-86 (1992).

[2] E K Blum, "Approximation of Boolean functions by sigmoidal networks: Part I: XOR and other two-variable functions", *Neural Computation*, 1, 532-540 (1989).

[3] C Darken and J M Moody, "Towards faster stochastic gradient search", in: *Advances in Neural Information Systems 4*, Morgan Kaufmann, San Mateo, CA, 1009-1016 (1991).

[4] J M McInerney, K G Haines, S Biafore and R Hecht-Nielsen, "Error surfaces of multi-layer networks can have local minima", UCSD Tech. Rep. CS89-157, October 1989.

[5] L Ingber, "Very fast simulated re-annealing", *Mathl. Comput. Modelling*, 12, 967-973 (1989).

[6] J Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI (1975).