NEURAL LEARNING ALGORITHMS:
Some Empirical Trials

Richard Forsyth,
Dept. of Psychology
University of Nottingham
Nottingham NG7 2RD, UK.

[currently at:
Dept. Mathematical Sciences
University of the West of England
Bristol BS16 1QY, UK.]

Abstract: The recent revival of Connectionism has led to an upsurge of interest in trainable pattern associators and pattern classifiers of many types. However, one training method currently dominates the field -- the back propagation algorithm. This method is crowding out other neural learning algorithms and other inductive techniques. The present paper reports some empirical trials comparing seven different neural learning algorithms (including two versions of back propagation) on four test problems. Though limited in scope the present study does shed light on the performance of a variety of learning techniques, compared under relatively uniform conditions. The results cast some doubt on the status of back propagation as an 'industrial strength' learning algorithm. It appears to scale up rather poorly; and on two pattern recognition tasks it gave a higher error rate than a commonly used statistical technique. These results suggest that the neurocomputing community as a whole may be in danger of becoming fixated at a local optimum, just like some of its algorithms.

Keywords: Machine Learning, Neural Computing, Back Propagation, Genetic Algorithms, Simulated Annealing, Distributed Memory.

1. Introduction

The recent resurgence of interest in Neural Computing, also known as Connectionism, threatens to distort the field of Machine Learning by overemphasizing connectionist learning at the expense of other kinds of inductive system, and one neural learning algorithm in particular -- the method of back propagation (Werbos, 1974; Parker, 1982; Rumelhart & McClelland, 1986). This has now become almost a 'connectionist cliché' (Forsyth, 1990), yet it suffers from a number of known problems including poor scalability (Tesauro, 1987) and 'network paralysis' (Wasserman, 1989). Moreover, several reports describe alternative neural computing models which outperform back propagation in speed, accuracy or both (Lippman, 1987).

For example, Shepanski (1988) compared standard numerical optimization techniques against a back-propagation network with 2 hidden layers on a problem involving the reconstruction of a noisy input signal. The numerical method proved 117 times faster and 39 times more accurate.

The present study is an empirical investigation of the status of back propagation as a learning algorithm. Seven different neural learning programs were written by the author and applied to four different test problems. This format is conceptually simple; nevertheless, it is hoped that it serves a useful purpose by comparing several alternative methods on a relatively 'level playing field'.

2. Seven Neural Learning Algorithms

Six algorithms were chosen for testing -- one in two variants -- giving seven distinct methods. All but the last are weight-adjustment methods.

1.	RANDNET	Pure Monte Carlo Search
2.	GREEDNET	'Greedy' or best-first search
3.	ANNA	Simulated Annealing (Metropolis et al., 1953)
4.	BP/V	'Plain Vanilla' Back Propagation
5.	BP+M	Back Propagation with Momentum
6.	GENE	Genetic Search (Forsyth, 1981; Ackley, 1987)
7.	CMAC	Albus's CMAC Memory Model (Albus, 1981)

3. The Four Test Problems

The first two test problems were mainly intended to test speed and accuracy of learning. The second two problems require the systems to act as pattern recognizers and were mainly intended to test the ability to generalize.

1. EXOR : Exclusive OR
2. PARITY4 : 4-Bit Parity
3. ZOOBASE : Animal Classification
4. CHOX : Image Identification

EXOR is the ubiquitous Exclusive-Or problem, identified by Minsky & Papert (1969) as the simplest Boolean function that a one-layer Perceptron (Rosenblatt, 1962) cannot solve. It is included for its historical importance, and as a link between the present work and earlier studies. The second test problem, 4-bit parity, extends the EXOR problem from two input lines to four. The task here for the nets is to learn to respond with 1 when an even number of inputs are on and zero when an odd number are on.

The ZOOBASE data-set contains details of 101 animal species, each described in terms of 17 features, such as the number of legs it has or whether it gives milk to its young. There are 7 output lines, all of which are off (0) except one -- indicating the

zoological class of the species concerned. To test the systems' capacity to generalize from seen to unseen cases, the data was randomly split into two subsets -- a training set of 57 cases and a test set containing the remaining 44 examples.

The CHOX data-set requires the nets to perform a simplified industrial inspection task. This data comes from a larger database of images previously described by Shepherd (1983). CHOX contains 80 records in all, each with nine measurements obtained from silhouette photographs of different chocolates. These shape descriptors describe the geometric features of the chocolates as seen by a vision system with a resolution of 96x96 pixels. The 80 cases were randomly divided into two subsets, 44 training examples and 36 test cases. There were eight output lines of which all were zero except one, indicating the chocolate type that produced the image. The nets had to learn to assign each image to its correct class.

4. Test Results

All seven methods were run five times on the EXOR problem, each time taking 2000 passes over the training examples. Table 1 shows the median error score (i.e. the 3rd-best of five runs) attained after 2000 passes over the training data by each method. This score is the squared deviation from the true value (0 or 1) of the system's output averaged over the four possible inputs.

[Table 1 -- Error Scores on EXOR.]

Method	Error-Score
CMAC	0.0000
BP+M	0.0004
GREEDNET	0.0008
GENE	0.0123
ANNA	0.0250
BP/V	0.1263
RANDNET	0.1414

The methods have been placed in order. The best score was obtained by CMAC, which always converged to a mean error score of less than 0.0001 within 250 passes. The main finding to emerge is the divergence between BP/V and BP+M, confirming that back propagation is very sensitive to correct choice of learning-rate and momentum parameters.

PARITY4 presents a more challenging test, and here the results start to show clear differences between the various methods. Table 2 lists the mean squared errors after 2000 passes (again the median of 5 runs) for all seven methods.

[Table 2 -- Error Scores on PARITY4.]

Method	Error Score
CMAC	0.0001
GENE	0.1516
ANNA	0.2119
RANDNET	0.2379

GREEDNET	0.2438
BP+M	0.2500
BP/V	0.2500

Here the superiority of the distributed memory method (CMAC) over the others is obvious. In fact, CMAC always converged to within 0.0025 of the correct answer within 350 passes on all 16 inputs. These results suggest that back propagation scales up rather poorly from a simple problem (EXOR) to a more complex version of the same task (PARITY4).

It is also worth noting that the back-propagation programs were the slowest of all on this problem. Table 3 gives the mean runtime for 2000 passes, rounded to the nearest second, taken by each method on a 386-based computer.

[Table 3 -- Average Runtimes.]

Method	Runtime	Ratio
RANDNET	661	1.000
GREEDNET	680	1.029
ANNA	688	1.041
GENE	779	1.179
CMAC	1511	2.286
BP/V	2261	3.421
BP+M	2271	3.436

(This comparison is actually unfair on CMAC, since it always converged within 350 passes. If time-to-convergence had been the criterion it would easily have been the fastest method.)

Such figures must be treated with caution. Nevertheless, they portray back propagation in a most unflattering light, and tend to support the contention that training a network by back propagation is a slow process.

To test generalization ability, the two methods that did best on the harder of the two logical problems (ANNA and GENE) as well as the better of the back-propagation methods (BP+M) were applied to two more realistic data-sets. In addition, to give a slightly broader perspective, a program was written implementing a well-established statistical classification method, the Nearest-Neighbour technique (Fix & Hodges, 1951) and applied to the same data. (Note: when being re-run on the training file, this program computes the distance of each case to all **other** exemplars, excluding the current case itself.)

For the comparisons reported below, success rate (percentage of cases correctly classified) is used as the primary performance measure. To force each system to give an unambiguous classification, the output line with the highest value was always picked as the system's decision.

Tables 4 and 5 show the results on the ZOOBASE and CHOX data after 200 passes over the training data.

[Table 4 -- Success Rates on ZOOBASE.]

Method	Training-Set%	Test-Set%
CMAC	100	70
BP+M	100	82
ANNA	60	55
GENE	54	39
Nearest-Neighbour	95	89
Default Strategy	46	34

(The default strategy was simply to pick the commonest category: it shows the success rate expected by chance.)

[Table 5 -- Success Rates on CHOX.]

Method	Training-Set%	Test-Set%
CMAC	100	86
BP+M	80	72
ANNA	39	22
GENE	32	28
Nearest-Neighbour	89	94
Default Strategy	18	17

BP+M converged to a solution with the ZOOBASE training data inside 200 passes and came reasonably near convergence with the CHOX data, while ANNA and GENE hardly got started.

CMAC and BP+M showed some signs of overfitting the training data, a finding also noted in Hart (1990). Indeed CMAC fitted the training data perfectly both times, and did show some ability to generalize; but it was not as accurate on unseen data as the nearest-neighbour method. (Once again BP+M was the slowest of the methods tested.)

5. Back Propagation: the Bubble-Sort of Connectionism?

This study is limited in aims and scope. Neverthless some trends may be discerned in the results, and some general remarks about the place of neural methods within the field of machine learning would appear to be justified.

Back propagation, it seems fair to say, is very good when it works well, but awful when it works badly. In other words, it is a brittle technique. It tends to scale up poorly, and demands considerable effort in tuning the learning parameters. Also it is very slow.

The genetic algorithm would appear to be robust but slow. That is to say, it was the best of the weight-space search methods on the most difficult problem (PARITY4) but took much longer to learn than back propagation on the easier problems (ZOOBASE and CHOX).

Simulated annealing, it would appear, falls between back propagation and the genetic search. It fared slighty better on the easy problems than the genetic search but slightly worse on the hard one. It may well deserve more attention than it has received as a viable

compromise between speed and robustness. (And, like the genetic algorithm but unlike back propagation, simulated annealing can be used to optimize non-numeric knowledge structures, such as symbolic descriptions.)

CMAC was the only method that always fitted the training data perfectly; and it did show an ability to generalize to unseen cases comparable with that of back propagation. However the version used here was not as robust in the face of noise as the nearest-neighbour classification technique.

In view of the optimism displayed by some members of the neuro-computing fraternity, the fact that all the neural methods were beaten by a commonplace statistical technique can only be described as embarassing. Nearest-neighbour classification was included to serve roughly the same function as a placebo in a clinical trial. Yet it outperformed the more sophisticated algorithms.

To conclude: it seems appropriate to sound a warning against becoming fixated upon any single neural architecture. The real problem with back propagation is not that it sometimes fails to converge in a reasonable time. A more serious problem is that it has narrowed the vision of the entire neuro-computing community. It only works with nets of a certain type; consequently only nets of that type are seriously studied. Systems of a radically different nature, such as those of Albus (1981), Reilly et al. (1982) or Aleksander (1987), tend to receive relatively little attention. Non-connectionist inductive techniques receive even less.

Perhaps the neuro-computing community as a whole is in danger of becoming trapped at a local optimum, like some of its algorithms?

6. Acknowldgements

I would like to thank Dr Simon Jones of Loughborough University and Prof. Igor Aleksander of Imperial College for helpful comments on earlier drafts. (They should not, of course, be blamed for any faults that remain.) Thanks are also due to the I.E.E. for kindly allowing me to reprint this paper which was first presented at the **IEE Machine Learning Colloquium on 28th June 1990** in London. (A fuller version was later presented at the 'Neuro-Nîmes-90' Conference on Neural Networks in November 1990.)

7. References

Ackley, D.H. (1987) -- An empirical study of bit vector function optimization. In Davis, L. (ed.) Genetic algorithms & simulated annealing: Pitman, London.

Albus, J.S. (1981) -- Brains, behavior & robotics: McGraw-Hill, Peterboro, NH.

Aleksander, I. (1987) -- Adaptive vision systems and Boltzmann machines: a rapprochement: Pattern Recognition Letters, 6,

pp. 113-120.

Fix, E. & Hodges, J.L. (1951) -- Discriminatory Analysis, nonparametric classification: USAF School of Aviation Medicine, Report no. 4.

Forsyth, R.S. (1981) -- BEAGLE, a Darwinian approach to pattern recognition: Kybernetes, 10, pp 159-166.

Forsyth, R.S. (1990) -- The strange story of the Perceptron: AI Review, 4, pp 147-155.

Hart, A. (1990) -- Concept learning with a multi-layer Perceptron: in Mirzai, A.R. Artificial Intelligence: concepts and applications in engineering: Chapman & Hall, London.

Lippmann, R.P. (1987) -- An introduction to computing with neural nets: IEEE ASSP magazine, April 1987, pp 4-22.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953) -- Equations of state calculations by fast computing machines: J. Chemistry & Physics, 21, pp 1087-1091.

Minsky, M. & Papert, S. (1969) -- Perceptrons: an introduction to computational geometry: MIT Press, Cambridge, Mass.

Parker, D.B. (1982) -- Learning-logic: Invention report S81-64, file 1, Office of technology licensing, Stanford Univ.

Reilly, D.L., Cooper, L.N., Elbaum, C. (1982) -- A neural model of category learning: Biological Cybernetics, 45, pp 35-41.

Rosenblatt, F. (1962) -- Principles of neurodynamics: Spartan Books, N.Y.

Rumelhart, D.E. & McClelland, J.L. (1986) -- Parallel distributed processing, vol. 1: MIT Press, Cambridge, Mass.

Shepanski, J.F. (1988) -- Fast learning in artificial neural systems: Proc. 4th AI-West Conf., Long Beach, Tower Conf. Management Publishing, Glen Ellyn, Illinois.

Shepherd, B.A. (1983) -- An appraisal of a decision tree approach to image classification: Proc. 8th IJCAI, Karlsruhe, pp 473-475, Morgan Kaufmann publishers, Los Altos, CA.

Tesauro, G. (1987) -- Scaling relationships in back propagation learning: dependence on training set size: Complex Systems, 1, pp 367-372.

Wasserman, P.D. (1989) -- Neural computing: theory & practice: Van Nostrand Reinhold, New York.

Werbos, P.J. (1974) -- Beyond regression: new tools for prediction & analysis in the behavioral sciences: Masters thesis, Harvard Univ., Mass.