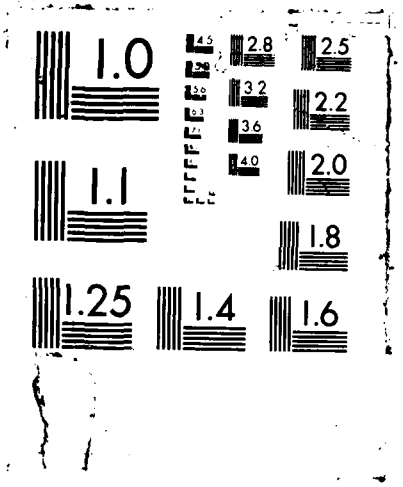
AD-A189 266 EXPERIMENTS WITH THE BACK PROPAGATION ALGORITHM: A SYSTEMATIC LOOK AT A S..(U) ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) M D BEDWORTH ET AL. 1/1

UNCLASSIFIED 29 JUN 87 RSRE-MEMO-4049 DRIC-BR-103551 F/G 12/9 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

UNLIMITED

RSRE
MEMORANDUM No. 4049

ROYAL SIGNALS & RADAR ESTABLISHMENT

EXPERIMENTS WITH THE BACK PROPAGATION ALGORITHM: A SYSTEMATIC LOOK AT A SMALL PROBLEM

Authors: M D Bedworth and J S Bridle

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.

RSRE MEMORANDUM No. 4049

UNLIMITED

ROYAL SIGNALS AND RADAR ESTABLISHMENT
Memorandum 4049

Experiments with the Back-Propagation Algorithm: A Systematic Look at a Small Problem [1]

Mark D. Bedworth
John S. Bridle
Royal Signals and Radar Establishment,
St Andrews Road, Great Malvern, England

June 29, 1987

Abstract

The multi-layer perceptron is a type of feed forward neural network for which there exists a learning algorithm based on error back-propagation. Experimental results are presented of the use of this error back-propagation algorithm on a carefully selected, simple problem. The effects of varying the structure of the network, the number of hidden units, the size of the training set and the initial weight values have been investigated. A number of ways of analysing solutions to such a simple problem are demonstrated. Explanations of the observed behaviour are offered which may provide insights applicable to a range of problems.

[1]This work was carried out while the authors were attached to the National Electronics Research Initiative in Pattern Recognition at RSRE.

1 Introduction

The error back-propagation learning algorithm (EBP, see [1] for a full explanation) for the multi-layer perceptron (MLP) has generated considerable interest in a number of fields including speech and image processing. Problems in these areas are often inherently large, stochastic and continuous in nature. In contrast, EBP has been demonstrated mainly on small, deterministic, discontinuous problems. Only limited practical information is available about the use of EBP on the kind of real world problem for which it is likely to be most useful [2,3,4,5]. A small, stochastic, discontinuous problem was constructed, which was intended to have some properties of realistic problems whilst being small enough to analyse in detail. The problem is a generalisation of the exclusive-OR problem and is a simple member of a class including hidden Markov models and Markov random fields, which are important in automatic speech recognition and machine vision. An MLP was trained to discriminate between patterns generated by two different Markov sources. The problem was chosen because it is stochastic rather than deterministic, it is possible to find optimal solutions with other techniques, and the complexity of the problem can be altered by changing the parameters of the generators.

2 Markov Source Discrimination

Each pattern is a short binary first-order Markov chain; i.e. a sequence of N zeros and ones in which the probability of a one in any position depends on the symbol at the previous position.

Such patterns are generated by simple Markov sources. We limited our investigation to symmetrical Markov sources, which are completely specified by a probability, p, of any bit being different from the previous bit.

$$p \triangleq P(x_i \neq x_{i-1}). \tag{1}$$

The probability of any particular sequence of symbols, $x_1 \ldots x_N$, being produced by a source with known transition probability, p, is:

$$P(x_1 \ldots x_N \mid p) = \prod_{i=2}^{N} T_i, \tag{2}$$

$$T_i \triangleq \begin{cases} p & \text{if } x_i \neq x_{i-1} \\ (1-p) & \text{if } x_i = x_{i-1} \end{cases}, \tag{3}$$

$$P(x_1 \ldots x_N \mid p) = p^T (1-p)^{N-1-T}. \tag{4}$$

Figure 1 illustrates the kind of Markov source used to generate the data used in the experiments.

Two Markov sources which have different transition probabilities will generate characteristically different patterns. Given a pattern the discriminator has to "guess" which of

two known Markov sources was most likely to have generated it. The classic exclusive-OR problem is the limiting case of two bit patterns and transition probabilities of 0.0 and 1.0.

Experiments have been carried out with a number of different transition probabilities and pattern lengths. Results are presented for six-bit patterns generated by two equally likely Markov sources with transition probabilities of $p = 0.7$ and $p = 0.3$. In order to minimize the expected squared error, the output of the network should be $P(p = 0.7 \mid x_1 \ldots x_N)$.

Using Bayes' rule

$$P(p = 0.7 \mid x_1 \ldots x_N) = \frac{P(x_1 \ldots x_N \mid p = 0.7)P(p = 0.7)}{P(x_1 \ldots x_N)}. \tag{5}$$

But $P(p = 0.7) = \frac{1}{2}$, and $P(x_1 \ldots x_N) = \frac{1}{2}(P(x_1 \ldots x_N \mid p = 0.7) + P(x_1 \ldots x_N \mid p = 0.3))$, so

$$P(p = 0.7 \mid x_1 \ldots x_N) = \frac{P(x_1 \ldots x_N \mid p = 0.7)}{P(x_1 \ldots x_N \mid p = 0.7) + P(x_1 \ldots x_N \mid p = 0.3)} \tag{6}$$

$$P(p = 0.7 | x_1 \ldots x_N) = \frac{0.7^T 0.3^{N-1-T}}{0.7^T 0.3^{N-1-T} + 0.3^T 0.7^{N-1-T}}. \tag{7}$$

If this is calculated for each value of T we find that approximately 16% of the patterns will have been generated by the least likely of the two sources. Figure 2 shows ten examples of patterns generated by each of the two Markov sources. The output target is 0.0 for the "slow" Markov source ($p = 0.3$), and 1.0 for the "fast" Markov source ($p = 0.7$).

3 Possible MLP Solutions

One obvious strategy for discriminating between such patterns would be to count the number of transitions in each pattern and decide in favour of the "fast" Markov source when this number exceeds some calculable threshold. One can arrange for a MLP to do just this by placing an exclusive-OR network over each of the five pairs of adjacent elements in the input pattern. Such a network was created manually using solutions to the exclusive-OR problem found during earlier experiments (see figure 3). There were 15 hidden units in two hidden layers and 51 connections (including biases) in this network. (A similar solution without the second hidden layer, using five each of OR subnets and AND subnets had only 10 hidden units and 41 connections). As expected, the performance of the MLP in each of these cases was essentially equal to the 84% upper limit on performance as discussed in section 2.

Further work was carried out on simpler networks with a single hidden layer with various numbers of hidden units (see figure 4). There was no *a priori* knowledge used in the design of this network. The input to the network was treated as if it was the output of a set of virtual input units. Each layer was fully connected to the next layer, there were no connections between units in the same layer or between non-adjacent layers. The performance of the trained network was assessed by comparing the rounded output with the desired output. In general, for networks with more than one output this is equivalent to finding the member of the target set with the least Euclidean distance to the actual output of the MLP.

For each condition 20 training runs were made using different starting weights. The initial values of the weights were between -0.5 and +0.5 and the weights were updated after each presentation. After 10,000 presentations performance was assessed by testing on a set of 1000 input/output patterns. The worst, mean and best performance of each of the 20 runs were recorded. Unless otherwise stated there were 6 hidden units, 100 examples in the training set, the learning rate (η) was 0.05 and the momentum (α) was 0.95.

4 Analysis of Typical Solutions

Satisfactory performance was attained with a surprisingly small number of hidden units (5 or 6 being sufficient in most cases). Discrimination was achieved with the hidden units acting as feature detectors each reacting to some characteristic typical of one of the two Markov sources.

Figure 5 shows typical weight sets after learning. Each "T" shaped box shows the connections to and from the unit and all other units in the network as squares. The length of the side of each square indicates the magnitude of the weight; hollow squares represent positive weights and solid squares represent negative weights. The position of the box indicates one of the units involved in the connection, the postion within the box indicates the other unit. The direction of each connection should be clear from the context. The square at the extreme lower left corner of each box represents the bias on that particular unit [6]. The largest weights are around ± 5.0. Note that in stochastic problems of this kind there is no tendency for the magnitude of the weights to increase without limit.

Although there are differences in the detail of each solution, all of the solutions use the hidden units as "feature detectors" tuned to fast or slow variations and concentrating on different portions of the input pattern. Performance was close to the optimum in each case (between 81.6% and 82.6%).

Figure 6 shows the level of activation of the hidden units and the output unit for each of the 64 different input patterns; using the first weight set in figure 5. The patterns are ordered according to the number of transitions in the pattern (i.e. there are two patterns with no transitions, ten with one transition, twenty with two transitions, etc). The order of the hidden units has been chosen to clarify their function. In this solution the first and second hidden units respond to characteristics of patterns for the "slow" Markov source ($p = 0.3$). The third, fourth and fifth hidden units respond to characteristics of patterns for the "fast" Markov source ($p = 0.7$). Notice how detectors of similar features are most responsive to different portions of the input pattern. The sixth hidden unit is not used in this solution. Figure 7 is a graph showing the relationship between the output of the network and $P(p = 0.7 \mid T = n)$. Closest agreement occurs for transitions of three or more: this may be because there are more feature detectors for the "fast" Markov source than the "slow" Markov source.

Figure 8 shows a few input patterns which were automatically calculated to produce given outputs from the trained MLP. To generate each pattern a random input was presented

to the network (each element chosen at random from the interval 0/1) and an output was produced in the usual manner. The error at the output was propagated back to the input in the usual way; the input could then be modified in order to reduce the error at the output [7].

$$O_i = O_i + \gamma \frac{\partial E}{\partial O_i}. \tag{8}$$

This procedure was repeated a number of times (typically a few hundred) until the input pattern was approximately stable. These "fantasised" input patterns tend to be more extreme than those produced by the original two Markov sources. Note how the input patterns which generate an output of 0.5 are intermediate in two senses: firstly the level of activation of each element is less extreme and secondly the transition probability of the implied Markov source is between 0.3 and 0.7.

5 The Effect of Number of Examples in Training Set

Although there are only 64 different input patterns, none of them is unique to either of the two Markov sources. A finite set of input/output pairs cannot carry complete information about the generating Markov sources. The more examples are given, the better can the conditional probability function, $P(p = 0.7 \mid x_1 \ldots x_N)$, be estimated.

The network with 6 hidden units was trained on a variety of training sets having between 5 and 1000 patterns. Figure 9 is a graph of performance as a function of number of examples in the training set.

After an initial rise, the trained performance levels off as the size of the training set is increased. In this case a training set of 64 patterns (on average one of each different input) had a mean performance of 68.3%, 128 patterns 76.7% and 512 patterns 80.9%.

6 The Effect of Number of Hidden Units

A MLP was trained with up to 15 hidden units on suitable data, and performance on the test data was recorded. Figure 10 is a graph of trained performance as a function of the number of units in the hidden layer. The theoretical upper and lower bounds on performance are shown at 84% and 50%.

At first performance increases quite rapidly with additional hidden units. Having more than 6 hidden units makes little difference to the mean performance but the worst performance of the 20 runs improves until about 12 hidden units are present.

Note that a peak in best performance occurs when there are just four hidden units. This is a combined property of the number of hidden units and the size of the training set used. With only 100 training examples there is insufficient data for the network to extrapolate well

to the test data (see section 5). It is important for the network not to overspecialise on the training set as this will adversely affect the performance on the test data. With fewer hidden units there are insufficient degrees of freedom to overtrain on the training set (i.e. whilst average performance may not be as high as with many hidden units, if the performance is good on the training set it is likely to be good on the test set as well).

7 The Effect of the of the Initial Weights

Rumelhart et al [1] explained the need to start the weights off with small random values in order to break symmetry. The initial value of each weight was chosen at random from an interval centred on zero. Figure 11 is a graph of performance as a function of the size of the initial weights. For each run the number of training cycles was fixed at 10,000.

There is a very rapid rise in performance as the values of the initial weights move away from zero. For this problem the performance peaks when the size of the initial weights is approximately 1.0. Although mean performance then falls away gently, the worst of the 20 runs drops quickly back to 50%.

If the initial weights are too small then movement in weight space will begin so slowly that a good set of weight values cannot be reached in the number of training cycles used in the experiments. If the initial weights are too large then the network will become locked into a suboptimal region of weight space from which it can never escape.

8 Conclusions

The experiments have provided us with some useful insights into using MLPs on larger problems. Some of the insights were useful in a local reimplementaion of NETtalk [3,5].

Further work is necessary in order to find how well the suggested explanations hold up on more complex problems. One possible stepping stone might be to have more sizeable patterns and discriminate between more than two generators. One obvious step towards applying similar techniques to automatic speech recognition would be to replace each Markov source with a hidden Markov model with multidimensional Gaussian output distributions.

References

[1] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation", ICS Report 8506 (1985) (Institute for Cognitive Science, University of California, San Diego).

[2] D. C. Plaut, S. J. Nowlan and G. E. Hinton, "Experiments on Learning by Back Propagation", CMU-CS-86-126 (1986) (Carnegie-Mellon University).

[3] T. J. Sejnowski and C. R. Rosenberg, "NETtalk: A Parallel Network that Learns to Read Aloud", Technical Report JHU/EECS-86/01 (1986) (Johns Hopkins University, Baltimore).

[4] J. L. Elman and D. Zipser, "Learning the Hidden Structure of Speech", ICS report 8701 (1987) (Institute for Cognitive Science, University of California, San Diego).

[5] N. A. McCulloch, M. D. Bedworth and J. S. Bridle, "NETspeak: A Multi-Layer Perceptron that can Read Aloud", RIPRREP/1000/4/87 (1987) (National Electronics Research Initiative in Pattern Recognition, RSRE, Malvern).

[6] G. E. Hinton, T. J. Sejnowski and D. H. Ackley, "Boltzmann Machines: Constraint Satisfaction Networks that Learn", CMU-CS-84-119 (1984) (Carnegie-Mellon University).

[7] J. S. Bridle, M. D. Bedworth and N. Dodd, "The MLP Inverted: Computation using Back Propagation", RSRE Memorandum 4050 (1987) (Royal Signals and Radar Establishment, Malvern).

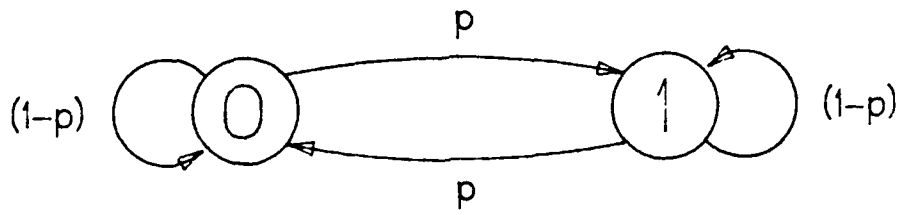


Figure 1: A binary, two state Markov source of the kind used in the experiments. The transition probability, p, of the state being different from the previous state completely specifies the model.

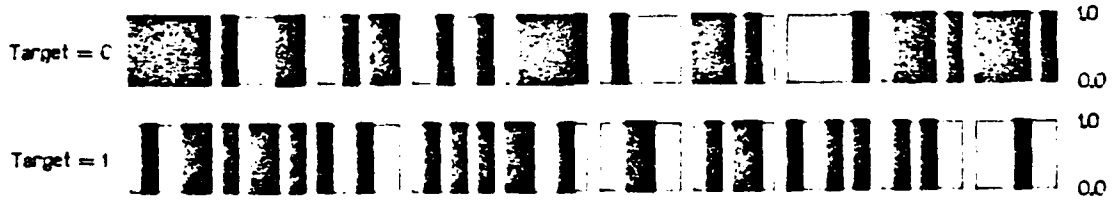


Figure 2: Examples of the six bit input patterns presented to the network. Each pattern was generated by one of the Markov sources ($p = 0.7$) or ($p = 0.3$). The target is 1 for the "fast" Markov source ($p = 0.7$) and 0 for the "slow" Markov source ($p = 0.3$).

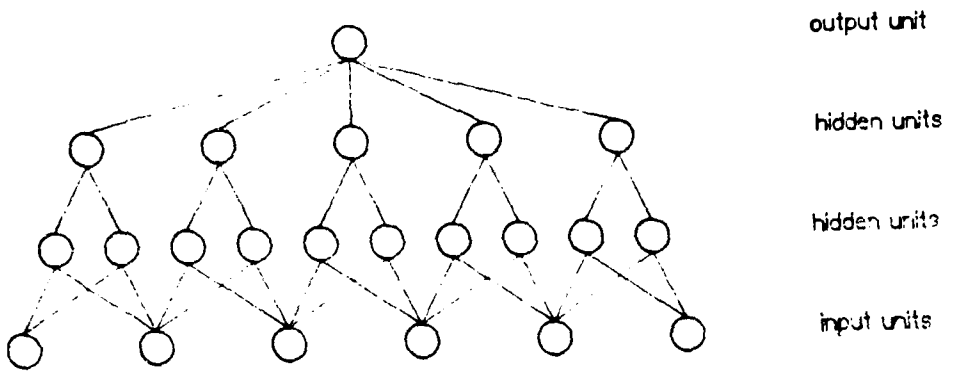


Figure 3: The structure of a hand crafted network which can solve the Markov source discrimination problem. Each exclusive-OR subnet detects a transition between a pair of elements in the input. The output unit essentially counts the number of transitions.

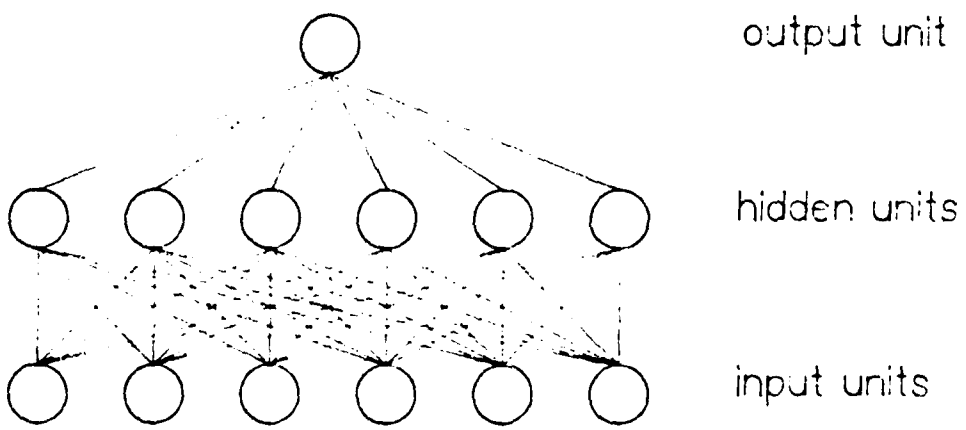


Figure 4: A fully connected, layered network of the sort used in the experiments. Six input units are connected to each of up to 15 hidden units in a single hidden layer. Each unit in the hidden layer is connected to the output unit.

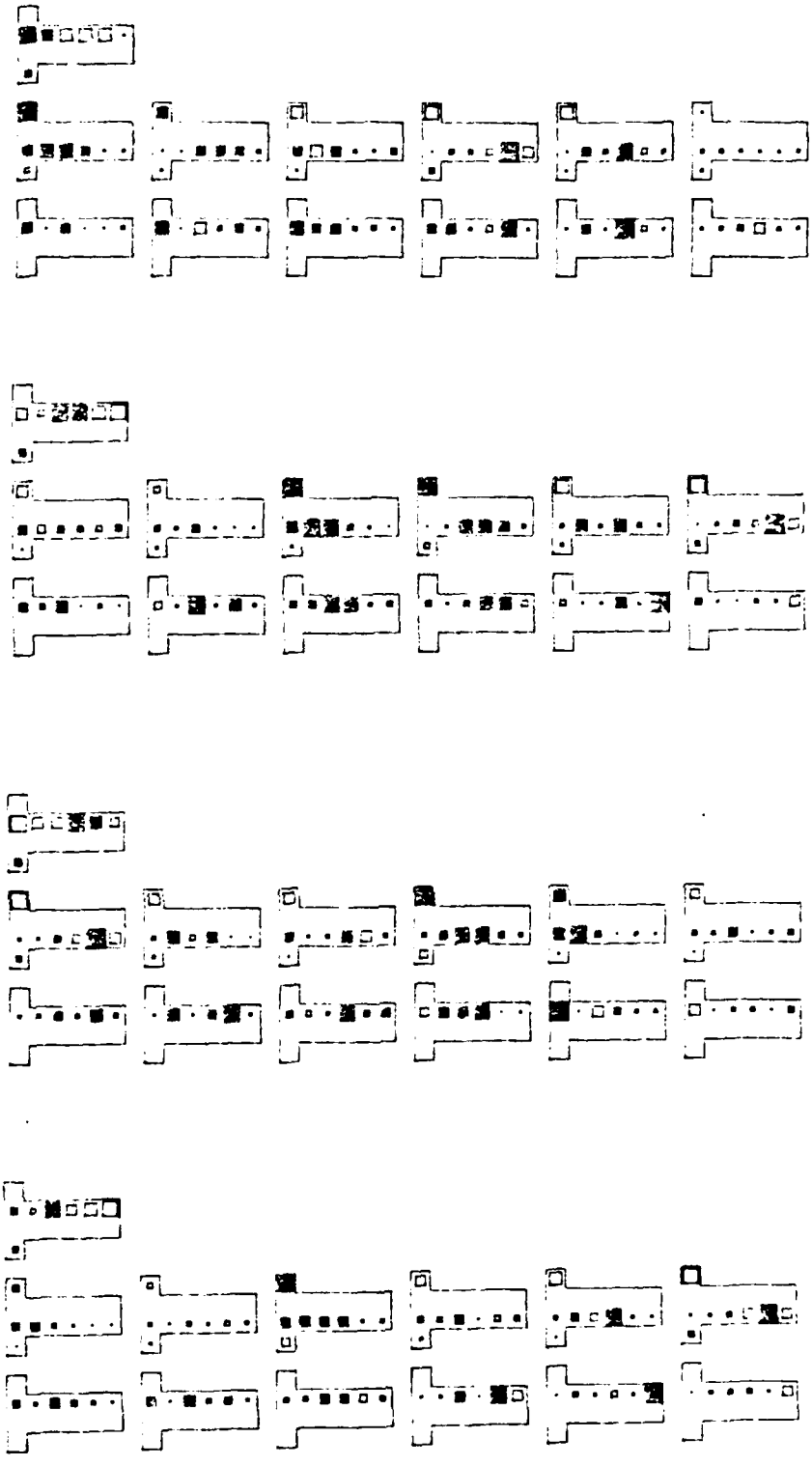


Figure 5: A selection of weight sets which found a solution to the Markov source discrimination problem. The value of each weight was found using the EBP algorithm after 10,000 presentations from a training set of 1000 patterns; $\eta = 0.05$ and $\alpha = 0.95$.

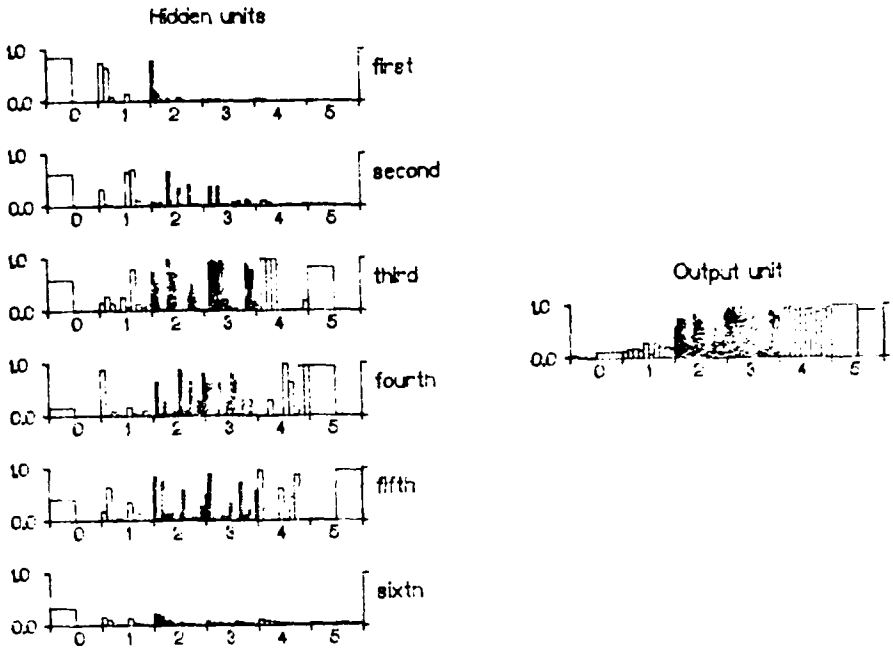


Figure 6: How each of the hidden units and the output unit respond to the 64 different input patterns.

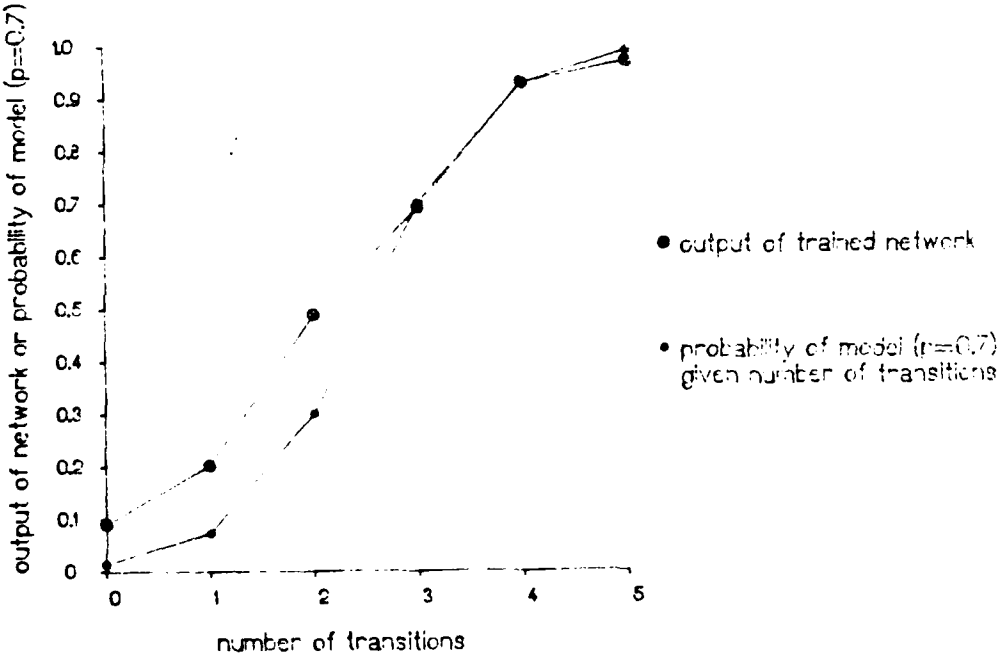


Figure 7: How well the output of the trained network fits the probability of the "fast" model given then number of transitions. Agreement is closest with pattern having 3 or more transitions.

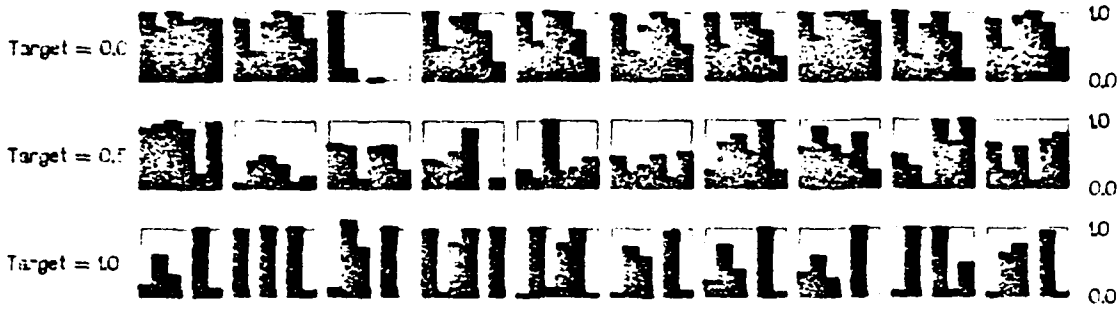


Figure 8: Input patterns which produce a desired output. The patterns were found using error back-propagation to modify a random input vector.

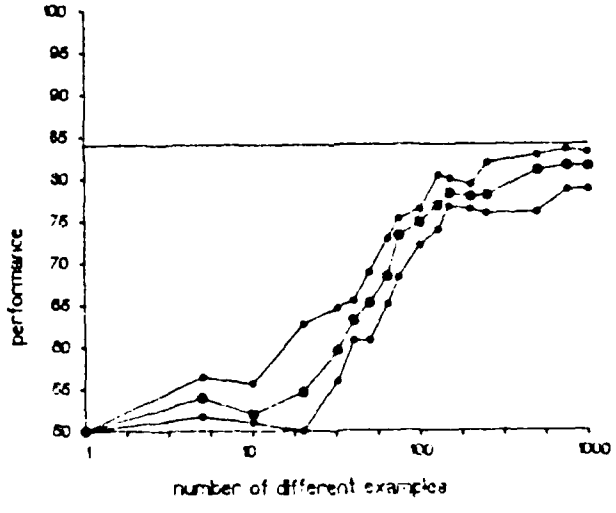


Figure 9: How the number of examples in the training set effects the performance of the trained network.

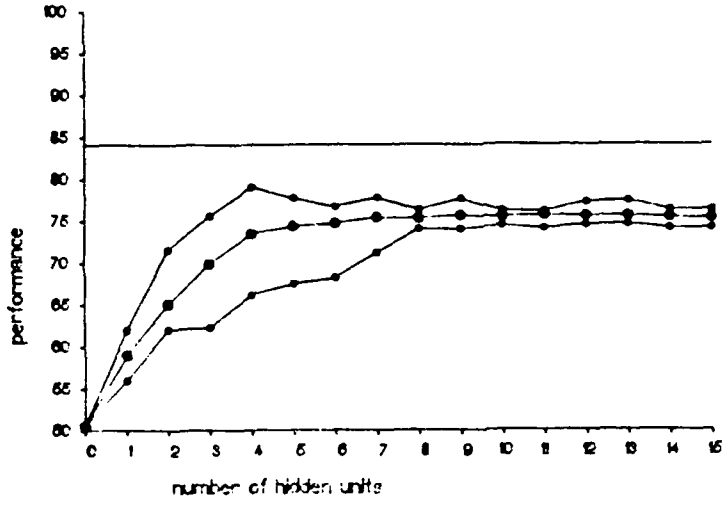


Figure 10: How the number of hidden units effects the performance of the network after training.

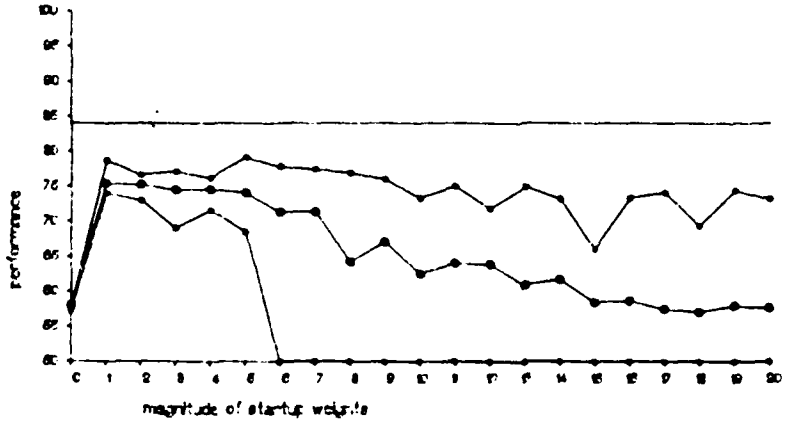


Figure 11: How the size of the initial weights effects training and performance.

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..... UNLIMITED .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification
	MEMO 4049		U/L

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location
7784000	RSRE, St Andrews Road, Malvern, Worcs. WR14 3PS

5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title
EXPERIMENTS WITH THE BACK PROPAGATION ALGORITHM: A SYSTEMATIC LOOK AT A SMALL PROBLEM.

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.
BEDWORTH, M.D.	BRIDLE, J.S.		1987.06	12

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

Descriptors (or keywords)

continue on separate piece of paper

Abstract A multi-layer perception is a type of feed forward neural network for which there exists a learning algorithm based on error back-propagation. Experimental results are presented of the use of this error back-propagation algorithm on a carefully selected, simple problem. The effects of varying the structure of the network, the number of hidden units, the size of the training set and the initial weight values have been investigated. A number of ways of analysing solutions to such a simple problem are demonstrated. Explanations of the observed behaviour are offered which may provide insights applicable to a range of problems.

END

DATE

FILMED

MARCH

1988

DTIC