nr. of processors	execution time	speedup	efficiency
1	3.50	1.00	
2	2.10	1.70	0.84
4	1.25	2.80	0.70
8	0.90	3.90	0.49

Table 1: Characteristics for parallel implementation of recurrent backpropagation; time in units of 10^3 seconds

nr. of processors	execution time	speedup	efficiency
1	2.10	1.00	
2	1.40	1.52	0.76
4	0.95	2.20	0.55
8	0.80	2.62	0.33

Table 2: Same as Table 1 for feedforward backpropagation

5 Acknowledgements

We wish to thank P. Braun and Dr. F. Hergert-Mückusch for helpful discussions. One of us (H. B.) appreciates the use of the *ipsc/2* multiprocessor system of the Institut für Informatik, TU München.

References

[1] Almeida, L.B., *A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment*, in: Proceedings of the IEEE First InternationalConference of Neural Networks, San Diego, California, Vol. II (1987) 609-618.

[2] Pineda, F.J., *Generalization of Backpropagation to Recurrent and Higher Order Neural Networks*, in: D.Z. Anderson (ed.), Neural Information Processing Systems, Am. Inst. Phys., NY (1988) 602.

[3] Rumelhart, D.E., Hinton, G.E., Williams, R.J., *Learning Internal Representations by Error Propagation*, in: Rumelhart, D.E., McClelland, J.L., Parallel Distributed Processing, Vol. I, MIT Press (1986) 318-362.

[4] Ramacher, U., Schürmann, B., *Unified Description of Neural Algorithms for Time-Independent Pattern Recognition*, in: U. Ramacher, U. Rückert (ed.), VLSI Design of Neural Networks, Kluwer Academic Publishers (1990) 255-270.

[5] Pearlmutter, B.A., *Learning state space trajectories in recurrent neural networks*, Neural Computation, 1 (1989) 263-269.

[6] Provided by S. Knerr from E.S.P.C.I., Paris (1990).

[7] Hollatz, J., Schürmann, B., *The "Detailed Balance" Net: A Stable Asymmetric Artificial Neural System for Unsupervised Learning*, Proceedings of the IEEE International Conference on Neural Networks, San Diego, (1990) 453-459.

Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)
Elsevier Science Publishers B.V. (North-Holland), 1991

THE NORMALIZED BACKPROPAGATION AND SOME EXPERIMENTS ON SPEECH RECOGNITION

E.Monte,J.Arcusa, J.B.Mariño, E.Lleida.

Dept. de Teoria del Senyal i Comunicacions. UPC.
Apartat 30002.
08080 Barcelona.SPAIN

ABSTRACT

In the paper we present the theoretical development of the normalized backpropagation, and we compare it with other algorithms that have been presented in the literature.[1]

The algorithm that we propose is based on the idea of normalizing the adaptation step in the gradient search by the variance of the input. This algorithm is simple and gives good results in comparison with other algorithms that accelerate the learning and has the additional advantage that the parameters are calculated by the algorithm, so the user does not have to make several trials in order to trim the adaptation step and the momentum until the best combination is found.

The task which we have designed in order to compare the algorithms is the recognition of digits in the Catalan language, with a data base of 1000 items, spoken by 10 speakers. The algorithms that we have compared with the normalized back propagation are: D.E.Rumelhart and J.L. McClelland [1], Franzini [2], Suddhard [3], Fahlman [4], Monte [5].

1. INTRODUCTION

It is well known that the Back propagation algorithm suffers of a series of drawbacks, the most important one, being the computational burden of training a network. Some papers have appeared that try to accelerate the convergence rate of the algorithm [2], [4],[6],[5] and others have presented ideas that can help to improve the performance of the networks [3]. In this paper we introduce an algorithm that yields a convergence rate nearly as good as the one the best with other methods, with the advantage that the user does not have to tune the adaptation step of the gradient search or the momentum. In order to compare the performance of the algorithms, instead of using toy problems (i.e: exclusive or, coders, etc.) we have tried a more real problem such as the speech recognition task. The comparison is done with the speech recognition task instead of being done with other tasks more simpler because in at least one of the algorithms [2], we have found that the scaling of the network from a toy problem to a more complex problem yields very different performances.

This work was supported by the PRONTIC grant nº 105/88

2. DESCRIPTION OF THE NORMALIZED BACKPROPAGATION ALGORITHM.

The type of adaptation of the units in a neural network, is very similar to the adaptation laws that are used for the LMS algorithm used for adaptive filtering [6],[7], in the following paragraph we will show how to adapt the idea of a Newton search to the context of neural nets on the hypothesis that the data at the input of each unit has a gausian statistic and there is incorrelation between the elements of the input vector.

It is known in the area of adaptive filtering that a filter of the RLS kind does a search by a Newton Method, and the iteration is of the following kind:

/1/ $$w(n+1)=w(n)+R(n)^{-1}X(n)*e(n)$$

Where:
- w: are the weights.
- R: is the autocorrelation matrix of the input.
- X: is the input vector
- e: is the error.

The algorithms based on Newton search are known to be much faster than than algorithms based on steepest descend. There is one case in which the Newton algorithm is similar to a the LMS, that is; when the input vector has gaussian statistics, then the autocorrelacion matrix is diagonal and the elements are the variance of the input signal. In this case instead of the product of the inverse of the autocorrelation matrix, we only have to divide each element of the input vector by the variance of the input vector. Then the gradient search that we are doing with is shown in /2/ can be expressed in a different way.

/2/ $$w(n+1) = w(n) + \eta \nabla E(n)$$

where:
- η: is the adaptation step.
- $\nabla E(n)$: is the gradient of the cost function.

In the normalized Back Propagation we propose that instead of using a fixed adaptation step for all the training to use an adaptive adaptation step based on the variance of the input of the unit. Thus if the input of the unit is X then the adaptation step would be:

/3/ $$\eta = 1/\sigma^2(n)$$

Where: $\sigma^2(n)$: is the variance of the input data.

/4/ $$\sigma^2(n) = \beta \sum_{i=1}^{N} x(i) + (1-\beta) * \sigma^2(n-1)$$

Where:
- β: is an exponential window.
- x(i): is the "i" element of the input of a given unit.

If we use this idea then the formula for updating the weights is of the form:

/5/ $$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n)/\sigma^2(n),$$

This modification is quite straight forward on the traditional BackPropagation algorithm [1]. Nevertheless one still has to do some thing about the momentum. We know that the momentum normally filters the instantaneous estimation of the gradient, so we know that the momentum has to have a value less than one. We have tried several strategies to find the best way of estimating a good value of the momentum, and the one that has given us the best result has been making the value of the momentum equal to the square root of the adaptation step. Empirically we have found that this choice gives results similar to the best choice of the adaptation step and momentum with the algorithm [1].

3. DESCRIPTION OF THE EXPERIMENT

The task that we have decided in order to compare the different algorithms that we have studied is the classification of speech signal. The experiment consisted of the classification of isolated words. The data based was composed of 1000 items. There were 10 speakers, who repeated ten times each word. The corpus was the 10 digits, spoken in Catalan language. The signal was sampled at a rate of 8kHz and afterwards the LPC coefficients were calculated. Once we had the signal represented by means of the LPC coefficients we did a prealignment so that all the items had the same number of frames, afterwards we did a KL transform on the data in order to reduce the information [8].

The experiments where the recognition of isolated words, dependent of the speaker and independent of the speaker. The speaker independent experiment was done by training 9 speakers and doing the recognition with the one that was left out, then the test speaker was rotated so that the experiment was carried with all the speakers. This experiment was done with all the algorithms that have been compared in this paper.

4. BRIEF DESCRIPTION OF THE ALGORITHMS.

In this paper we will compare several algorithms, the one presented by Franzini [2], is based on changing the adaptation rate as a function of the variation of the gradient between two consecutive iterations, the algorithm presented by Suddhard [3] uses as hints during the training the number of the speaker whose utterance is presented at the moment, the algorithm that we present in this paper uses a initial estimation of the variance which is $\sigma^2=2$. Nevertheless we have discovered that the algorithm is quite independent of the initial value of σ. The memory factor β was taken near to zero: 0.1

5. RESULTS OF THE EXPERIMENTS.

In the following figure we present the results of comparing the algorithms, it can be seen that the normalized backpropagation although does not yield the same error rate that the Franzini algorithm, gives a result that is better than the others. It has to be emphizied that the results were obtained after several trials with different values for the adaptation step and the momentum, and the best curves were selected. The normalized back propagation has the advantage that all this trials are not necessary and after some training gives good error rates, the advantage of this algorithm is that it gives good results without the need of time consuming trials.

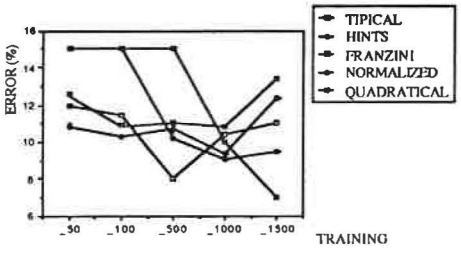

Figure 1: Recognition rate.Speaker independent recognition.

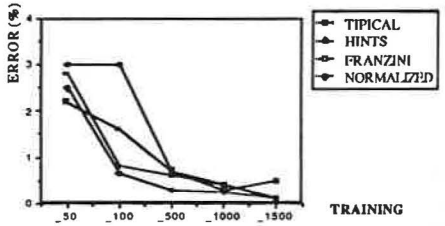

Figure 2: Recognition rate. Speaker dependent recognition.

There are two special things about the error curves, one is that all have a concave form, that is due to the fact that the recognition is done with utterances that were not used in the training, so when there is an excessive training the network does not generalize well. The other is that the error rates are not as good as error rates that have been obtained with other methods, like template matching or Hidden Markov Models, this is due to the fact that the networks that we have used are small, when more nodes are used in the hidden unit one can obtain better results. The results obtained of speaker dependent recognition are very similar for all the algorithms and are presented in figure 2.

6. CONCLUSIONS

In this paper we have presented a new algorithm and we have compared it with other existing accelerating algorithms. The results that have been obtained are quite encouraging, the normalized backpropagation behaves better than most and does not require several time consuming trials with the adaptation step and momentum, because it does an estimation of these parameters.

7. REFERENCES.

[1] D.E.Rumelhart, J.L. McClelland and the PDP group, "Parallel Distributed Processing: Explorations in Microstructrure of Cognition." Vol.1. Foundations". Eds. MIT Press, Cambridge Mass. 1986.

[2] M.Franzini, Kai-Fu-Lee, A.Waibel, "Connectionist Viterbi Training, A New Hybrid Method for Continuous Speech Recognition". Albuquerque ICASSP 90.

[3] S.C.Suddarth and Y.L.Kergosien, "Rule injection hints as a means of improving network performance and learning time" EURASIP Workshop 1990, Sesimbra,Portugal.

[4] S.E. Fahlman, "An Empirical Study of Learning Speed in Back Propagation Networks". Technical Report CMU-CS-88-162. June 88.

[5] E.Monte,E.Lleida, J.B.Mariño," New BackPropagation Algorithm Using Quadratical Potential Functions and an experiment on isolated word Recognition". EUROSPEECH 89. Paris1989.

[6] Cowan and Grand. "Adaptive Filters". Prientice Hall.

[7] M.Honig and D.Messerschmitt "Adaptive Filters, Structures, Algorithms and Applications". Kluwer Academic Publishers.84

[8] E.Lleida, C.Nadeu and J.B. Mariño, "Feature Selection Throug' Orthogonal Expansion In Isolated Word Recognition". Melecon-89.Lisbon.

Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)
Elsevier Science Publishers B.V. (North-Holland), 1991

An optimum weights initialization for improving scaling relationships in BP learning

Gian Paolo Drago
Istituto per i Circuiti Elettronici - CNR
via Opera Pia, 11 - 16145 Genova, Italy

Sandro Ridella
DIBE - Universita' di Genova
via Opera Pia, 11A - 16145 Genova, Italy

abstract- An algorithm for fast minimum search is proposed, which reaches very satisfying performances by making both the learning rate and the momentum term adaptive in an optimum way, and by executing controls and corrections both on the possible cost function increase and on eventual moves opposite to the direction of the negative of the gradient.
The global minimum search by restarting the algorithm is furthermore accelerated through an optimum criterion of initialization of the weights based on testing the neurons paralyzed state, that is when the magnitudes of both a neuron output and the error output are greater than a fixed threshold. Thanks to these improvements, we can obtain scaling relationships in learning more favourable than those previously obtained by Tesauro and Janssen.

1 Brief description of the AMBP algorithm

In this paper we will develop an Adaptive Momentum Backpropagation (AMBP) algorithm, whose improved optimization technique leads to scaling relationships in learning more favourable than Tesauro and Janssen's ones [7].

Our algorithm is based on three choices:

1) Developing an algorithm as fast as possible for local minimum search. This is based on the consideration that it is desirable to run a large number of *trials* with the weights initialized to different random values in each case. It is well known that it is convenient to restart a trial, with new random weights, whenever the network has failed to converge after a certain number of *epochs* [3]. The *epoch* is defined as a single presentation of each of the I/O patterns in the training set [6].

2) Providing an adaptive expression for the momentum term α. A satisfactory solution is derived from the Conjugate Gradient (CG) algorithm [4] [1], which does not require the computation of the Hessian matrix and guarantees the convergence in a finite number of steps for quadratic definite positive cost functions.

3) Avoiding *paralyzed states* during the random initialization of the weights. A neuron is considered to be in a paralyzed state when both the magnitude of its output and that of the network output error are greater than fixed thresholds at least for one training pattern. In this case the output of the network does not fit the target, but the error back-propagation cannot occur: this must be avoided during initialization.

A detailed description of our algorithm is presented elsewhere [2]; a short version is given below.

According to the first and the second choice, we developed an algorithm working by epoch. If $\mathbf{W}^k$ is the weights vector at the k_{th} iteration, and $\triangle\mathbf{W}^k = \mathbf{W}^{k+1} - \mathbf{W}^k$ is the variation of the weights vector at the k_{th} iteration, the *search move* $\triangle\mathbf{W}^{k+1}$ at the $(k+1)_{th}$ iteration is evaluated according to the Rumelhart's relationship:

$$\triangle\mathbf{W}^{k+1} = -\eta_{k+1}\mathbf{g}_{k+1} + \alpha_{k+1}\triangle\mathbf{W}^k \qquad (1)$$

where $\mathbf{g}_{k+1}$ is the gradient of the cost function with respect to the weights, α_{k+1} is the momentum term and η_{k+1} is the learning rate at the $(k+1)_{th}$ iteration.

It must be stressed that the usual BP [6] cannot be assimilated to a pure local minimum traditional search, since convergence is r etime obtained with the concurrence of one or more increments of the cost function. In this way system could escape a local minimum (see [6] at page 332): in the first place this *"up-hill"* moving is well controlled through Vogl's (1988) method [8] with satisfactory results.