ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 161-165

Incremental Increased Complexity Training

Jacques Ludik, jludik@cs.sun.ac.za
Ian Cloete ian@cs.sun.ac.za
Computer Science Department, University of Stellenbosch, Stellenbosch 7600, SOUTH AFRICA

Abstract

The training strategy used in connectionist learning has not received much attention. This paper discusses a new strategy for backpropagation learning and shows that it produces 52% faster convergence compared to conventional training using a fixed set for a complex *addition* experiment.

1 Introduction

The technique of back propagation (BP) for supervised learning to produce simulated neural networks is both established and well-known [Rumelhart *et al*, 1986]. It is also well-known that standard back propagation has defects concerning its ability to produce an effective solution in weight space within a feasible time. To address this problem the typical approach taken was to investigate alternative methods for selecting initial parameter values, or for adjusting parameter values during BP training, e.g. [Silva & Almeida, 1990]. This paper suggests that the training strategy, that is the method for presentation of examples to the network during learning, be investigated as an alternative method to produce an effective solution within a feasible time. This paper proposes a new training schedule called *incremental increased complexity training* (*iict*) and illustrates experimentally that it speeds up learning by 52% compared to the conventional training strategy of using a *fixed* set (*fst*) of training patterns.

2 Incremental Increased Complexity Training

Recently [Cloete & Ludik, 1993] proposed the *increased complexity training* (*ict*) strategy in which a fixed training set is partitioned into a sequence of disjoint subsets, each of which is "more complex" than the previous. Training occurs on the first subset until a termination criterion is met, then on the first two subsets which are merged in random order, etc. *ict* was experimentally compared with *cst* [Cottrell 91] using both a Simple Recurrent Network

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 161-165

and Temporal Autoassociation Network showing a vast improvement in learning time [Ludik & Cloete, 1993]. Ranking the training data into disjoint subsets according to some complexity measure is dependent on the task to be accomplished and may still cause rather large training subsets to be constructed, which again may bring the defects of BP learning into play. In this paper we therefore extend this strategy by experimenting with various increments in training subset size, while maintaining the complexity ranked order of *ict* subsets. The suggested *iict* strategy is discussed in the context of the *addition* task.

In the *addition* task [Cottrell 91] the aim is to learn to sequentially add two base four numbers. This Simple Recurrent Network [Elman 90] has five inputs, 16 hidden and 16 state units, and six output units. Each base four number is represented as a two-digit binary number. The network input therefore consists of one column of digits at each time step, as well as a end-of-input bit indicating when the last column of digits is input. The six output units represent the sum (two units, i.e. two bits) of the one column of digits and four possible actions (four units). Actions are to write the sum, to remember or output the carry, shift to the next column of digits, and indicate if done. Each single input pattern may generate a varying sequence of outputs. We call such an entire addition sequence a temporal pattern. The input and target patterns for each time step is called a single pattern.

This task is very complex, because the network must learn a sequence of inputs, and for each input a different sequence of outputs may be required. Furthermore, it was reported [Cottrell 91] that conventional training on a fixed set fails to converge, and it requires a large number of weight updates.

Temporal patterns were constructed for 1- to 3-column addition. Each fixed training set consisted of 320 temporal patterns of varying lengths. Training subsets of the fixed set were ranked in complexity order according to the number of columns of digits to be added, i.e. three sets of examples were constructed for examples of 1-, 2- and 3-column addition. All network initial conditions for every experiment were indentical. Weights were updated after every single pattern. For successful training a Root Mean Square (RMS) error value of less that 0.15 for the fixed training set and a success ratio greater than 97% on an independent test set were required.

For the initial experiment with *ict* the optimum RMS termination criteria for each subset was $0.65 - 0.45 - 0.15$. These RMS values were obtained out of combinations tried from 0.4 to 0.7 for the first subset, 0.35 to 0.6 for the second, and 0.25 to 0.45 for the third. *ict* was compared with *cst* [Cottrell 91] in which random training subsets are constructed as follows: The first subset consists of 1% of the training set, the next of 2% combined with the first, the next of 4% combined with the previous set, etc. For *cst* training subsets of the same size as the complexity ranked subsets were randomly constructed from the fixed set. Figure 1 compares the *ict*, *cst* and conventional *fixed* training strategies. For both *ict* and *cst* the method of determining the required RMS termination criteria per subset was unsatisfactory since it required experimentation in stead of being performed algorithmically.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 161-165

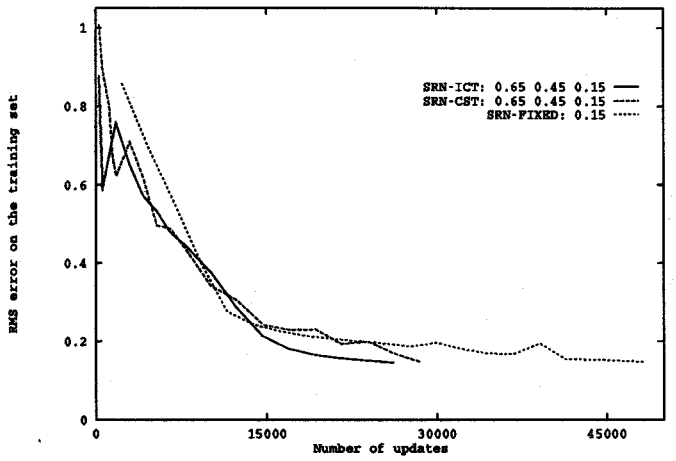

Figure 1: Performance of the SRN for *Addition*

When extending *ict* two issues must therefore be considered: the RMS termination criteria and the construction of training subsets. These issues are addressed by incrementally constructing subsequent training subsets and their termination criteria. For *incremental increased complexity training iict* subsets are constructed from the same complexity ranked subsets by incrementally merging a user-specified number of temporal patterns (the increment) to the previous subset, thus obtaining a sequence of training subsets in multiples of the increment. Examples presented to the network are therefore still complexity ordered, but subsets are constructed in "smaller steps". The problem of specifying an error termination criterion for each subset was addressed by linearly decreasing, the required RMS error value with each subset. The RMS error value obtained for the initial untrained network on the first subset is thus incrementally lowered until the user-specified RMS value is the required termination criterion for the final subset.

For the experiments increments of 1, 2, 4, 8, 16 and 32 temporal patterns on four different fixed sets numbered A to D were used. The distribution of examples of 1-, 2- and 3-column additions were 64:96:160 for set A, 32:96:192 for B, 16:64:240 for C, and 39:122:159 for D. An increment of 4 temporal patterns in all experiments produced the least number of weight updates, respectively 23 340 for set A, 24 856 for B, 27 851 for C, and 26 871 for D. The error curve for fixed set A is compared in Figure 2 for the various training strategies. *iict*, with increment 4, thus improves training time by 52% compared to *fst* (48 300), by 18% compared to *cst* (28 450), and by 11% compared to *ict* (26 140). An increment of 2 temporal patterns performed second best for sets A, B and C, while the 8 increment was second best for D. From these results we conjecture that the distribution of examples from each complexity class should not assign

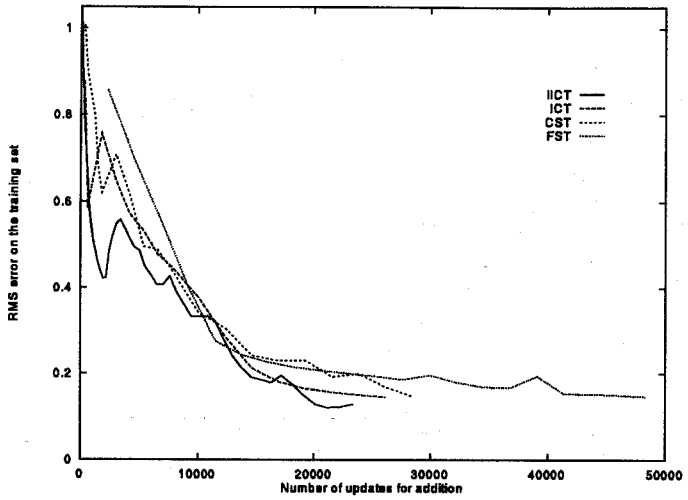

Figure 2: Performance of training schedules for *Addition*

too small importance to the less complex classes, as does set C. However, exactly how examples should be distributed must be further investigated. We simply note that the 4368 different temporal patterns for 1- to 3-column addition are dominated by additions of 3 columns, therefore if examples were chosen randomly according to this distribution the effect of increased complexity ordering would be nullified.

3 Conclusions

The typical approach taken by other researchers to address the defects of standard back propagation was to investigate alternative methods for selecting initial parameter values, or for adjusting parameter values during BP training. In this paper we suggested that the training strategy, that is the method for presentation of examples to the network during learning, is a viable alternative method to produce an effective solution within a feasible time. It was demonstrated that for all experiments *ict* and *iict* produced the best performance, where *iict* improved performance by 52% compared to conventional training on a fixed set.

References

[Cloete & Ludik, 1993] Cloete, I., and Ludik, J., "Increased Complexity Training", in *New Trends in Neural Computation*, eds. Mira, J., Cabestany, J., Prieto, A., Springer-Verlag, Berlin, pp.267-271..

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 161-165

[Cottrell 91] Cottrell, G.W., and Tsung, F.S., "Learning Simple Arithmetic Procedures", *High-Level Connectionist Models*, eds. J.A. Barnden, J.B. Pollack, Vol. 1, pp.305-321, 1991.

[Elman 90] Elman, J.L., "Finding Structure in Time", *Cognitive Science*, Vol. 14, pp. 179-211, 1990.

[Ludik & Cloete, 1993] Ludik, J., & Cloete, I., "Training Schedules for Improved Convergence", Int Joint Conf on Neural Networks, Nagoya, Japan, Oct 1993.

[Rumelhart *et al*, 1986] Rumelhart, D.E., Hinton G.E., & Williams R.J., "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing: Vol. 1, Foundations*, eds. Rumelhart, D.E., McClelland, J.L., Cambridge, MA: MIT Press, 1986.

[Silva & Almeida, 1990] Silva, F.M., Almeida, L.B., "Speeding up backpropagation", in *Advanced Neural Computers*, ed. Eckmiller, R., Elsevier Science Publishers, Amsterdam, 1990.