PREJUDICIAL SEARCH and BACKPROP

Jerome B. Soller
C.S. Department

Neil E. Cotter
E.E. Department

C.S. Department and E.E. Department

UUCS-91-009

Department of Computer Science
University of Utah
Salt Lake City, UT 84112, USA

May 22, 1991

Abstract

This paper introduces the combination of backward error propagation and prejudicial search. Prejudicial search is a method, which like simulated annealing, guarantees convergence to a global minimum as time approaches infinity. Unlike simulated annealing, its application is more flexible because it can be combined with other search methods. This method is applied to the exclusive-or problem. When prejudicial search is combined with backward error propagation, the resulting sum of mean squared error at each time step is between 10 and 70 percent (depending on the cooling schedule) of the normal backward error propagation error.

1 Introduction

Prejudicial search is a method for searching a cost space. Prejudicial search guarantees convergence to a global minimum as time approaches infinity[2]. Basically, it requires that the probability of choosing a random weight vector in a compact domain is greater than 1/k, where k is the number of iterations. The random weights are accepted if their global error is less than the error from the current weights. The remaining (k-1)/k of the probability at time k can be assigned to any search strategy. Although this method requires that the probability of the random search be at least 1/k, a higher rate of random sampling can be used. In the past[3], we have combined it with Barto and Sutton's ASE for the broomstick balancer problem[1]. Since this method guarantees convergence, we have combined with effective methods which don't guarantee convergence. Backward error propagation is the most commonly used algorithm of this type. For certain problems, backprop can lead to local minima or neuron saturation(high weight values leading to very small changes). Prejudicial search reduces the affect of certain randomly chosen starting weight vectors. The random search component finds good starting locations and the backward propagation does the fine tuning. The steps in the algorithm are described below:

1. Choose a domain for each of the weights and initialize the weights.

2. Flip a coin to see if the probability is less than the cooling rate. Usually, the cooling rate is 1/k, but for some problems a larger sampling rate improves the performance. If the probability is less than the cooling rate, perform step 3. Otherwise, step 4.

3. Choose a random weight vector in the space. Find the total error of this weight vector over all of the inputs. If this is less than the current weight vector's error, replace it. Otherwise, keep the current weight vector.

4. Perform the other search method. For the results presented below, perform backward error propagation.

5. Increment k and goto 2.

2 Results

We compared the performance of four different methods for the exclusive-or problem with two input units and two hidden units. The desired output is a 1 if the inputs are $0,1$ or $1,0$, and the desired output is a 0 otherwise. When we randomly choose weights, we present all four patterns in 1 iteration to determine the error. To find the weight changes in steps where backprop is used, we present one pattern every iteration.

backprop	*random search*	*Combined 1*	*Combined 2*
14.0	3.5	10.0	1.7

Note: The results shown above are scaled by 10^{-7}.

Table 1: Comparison Of Error Under The Curve For 4 Methods

1. Backward Error Propagation:
   Each of the random starting weights are chosen from the domain [-10,10]. The inputs are chosen randomly from the set of possibilities. The current weights are used to calculate the total error at every iteration.

2. Uniform Random Search:
   The domain is [-10,10]. The best weight vector is compared with a randomly chosen weight vector every iteration. If the new weight vector produces less error, it becomes the best weight vector.

3. Prejudicial Search combined with Backprop1:
   The cooling schedule was 1/k. The random weights were chosen from the same domain as above. Likewise, the current weights are used to calculate the error at every iteration.

4. Prejudicial Search combined with Backprop2:
   The cooling schedule was $1/\sqrt[10]{k}$.

The chart shows the error under the curve. This is calculated by summing the average mean squared error of the complete set of inputs each time step. For the exclusive-or problem, the set of inputs contains four vectors. For statistical reasons, we summed the error over 10 trials for each method. The least error was produced by the combination of prejudicial search and backprop with the $1/\sqrt[10]{k}$ cooling schedule. For this particular network and range of weights, random also performed extremely well. In general, the particular cooling schedule chosen depends on the problem and the range of the weights. For problems with very smooth cost surfaces, backward error propagation produces the best performance. Likewise, for problems with extremely rough cost surfaces, uniform random search would be the best method. Prejudicial search is most effective for problems between the extremes. This is particularly true for problems whose cost surface contains both rough and smooth regions.

3 Conclusion

Theoretically, combining prejudicial search with backward error propagation guarantees convergence. In practice, this hybrid method has superior performance for many problems sensitive to starting positions of the weights, neuron saturation, or local minima problems. This method performs best for problems whose surface is a combination of smooth and rough regions.

4 References

1. A.G. Barto, R.S. Sutton, and C.S. Anderson "Neuronlike adaptive elements that can solve difficult learning problems," IEEE Transactions Systems, Man, and Cybernetic, SMC-13:834-846, 1983.

2. Neil E. Cotter, Thierry Guillerm, Jerome B. Soller, and Peter Conwell "Prejudicial Searches And The Pole Balancer," *International Joint Conference on Neural Networks* Seattle, July 1991. (Dr. Cotter will present orally at Seattle)

3. Jerome B. Soller and Neil E. Cotter "Generality Vs. Speed Of Convergence In The Cart-Pole Balancer," University of Utah, Technical Report UUCS-91-008, April 1991.