Boosting and Other Machine Learning Algorithms

Harris Drucker
AT&T Bell Laboratories
Holmdel, NJ 07733
drucker@monmouth.edu

Corinna Cortes
L.D. Jackel
Yann LeCun
Vladimir Vapnik

Abstract

In an optical character recognition problem, we compare (as a function of training set size) the performance of three neural network based ensemble methods (two versions of boosting and a committee of neural networks trained independently) to that of a single network. In boosting, the number of patterns actually used for training is a subset of all potential training patterns. Based on either a fixed computational cost or training set size criterion, some version of boosting is best. We also compare (for a fixed training set size) boosting to the following algorithms: optimal margin classifiers, tangent distance, local learning, k-nearest neighbor, and a large weight sharing network with the boosting algorithm showing the best performance.

1. INTRODUCTION

Ensemble of learning machines (ofter called committee machines) may be justified if they show improved performance over single machines. Committee machines have been investigated by Perrone and Cooper (1992) Srihani (1992), Suen (1992), Benediksson and Swain (1992), and Hansen and Salamon (1990). In many of these cases, the committee members are neural networks trained on the same data, but initialized with different weights. Boosting is different in that each learning machine in an ensemble is trained sequentially on patterns that have been filtered by the previously trained members of the ensemble. We show detailed experimental evidence that boosting is superior to the other techniques analyzed here but because of space restrictions we can only summarize the algorithms. Details of the original boosting algorithm are given in Schapire (1990), applications to neural networks are given in Drucker, Schapire, and Simard (1993a,b) and a much more detailed version of this paper is to be published (Bottou, et. al., 1994) although this latter reference does not have the results for non-neural based algorithms presented here.

We have to carefully distinguish between the size of the training set and the actual number of patterns trained on. Assume we have a training set of size N which may consist of what we may call (but hard to define) typical patterns, hard patterns, easy patterns, mislabeled patterns, and outliers (not all mutually exclusive). By judiciously picking a subset of size N_c to actually train on, we may get better generalization performance than if we had trained on all N patterns. Of course one has to have some procedure to pick this subset but if we have some "smart" procedure we obtain two advantages: generalization is better and time to train is less. Of course, there is some computational cost in examining patterns and then discarding them, but that cost is typically much less than using them in training which requires a forward propagation, backward propagation, and weight updates for each pattern for each epoch. To distinguish between N and N_c we shall call the former the training set size and the latter the computational cost. When we do not explicitly train on all N patterns, the computational cost is a fair method to compare algorithms.

2. PROCEDURE

A database of 120,000 handwritten and segmented digits from the National Institute of Standards and Technology (NIST) was used for evaluation: two thousand of which were used for testing and randomly chosen subsets of the remaining 118,000 used for training. The pixel arrays were subsampled to give a 10 by 10 size input space so the algorithms could run in a reasonable time. For each of the four algorithms, each of the three architectures, and a particular training set size, the training and test performance were evaluated ten times (with different initialization of weights and different sets of training patterns) to obtain a mean for that training set size and

ML Conference Rutgers U. 1994

N training patterns, not just the N_c trained on. The error rate on the patterns discarded is small and dominates the overall error rate as the number of training patterns increases. In the limit, approximately 80% of the N patterns are easy patterns.

The parallel machines and the single machine use all the patterns in the training set to train on. Therefore the ratio of computational cost to training set size is 1 for these algorithms. The modified boosting algorithm is more expensive: the computational cost is larger than the training set size.

Figure 4 shows the training error rates for all four algorithms. As expected, for the parallel and single machine, training error rate increases while for the original boosting algorithm, it decreases. For the modified algorithm, the slope is positive until it attains either a negative or horizontal slope.

We show (Figure 5) a plot of test performance against training set size for the four algorithms. Generally, each algorithm has a range where it is superior. For small training set size a single machine is better while for large training set size, both the modified boosting and the original boosting algorithm give similar results (with differences statistically insignificant at the largest training set size). However, the modified boosting algorithm is better over a broader range. For the 100-10-10 network, we ask which algorithm is better if one has a given time to train. This is shown as a plot of test error rate versus computational cost (Figure 6). This clearly shows that the original boosting algorithm is better. This is not unexpected considering Figure 3 which shows that a smaller and smaller part of the training set contributes to the computational cost as the training set size increases.

A single layer fully-connected 100-10 network (Figures 7-8) has the same general error curve characteristics as the 100-10-10 network. Once again, both boosting algorithms asymptote to the same value although the original boosting algorithm is better (in terms of computational cost) over a broader range. Another item of interest is that the parallel and single machine have the same training and test performance for large training set size. This is not surprising in light of the fact that single layer networks have one minimum and all the machines in the parallel set of machines eventually obtain the same decision boundary. For this network, a single machine is as good as parallel machine. A plot of training error vs training set size for the 100-10 network would have show the same general characteristics as that of 100-10-10 network: negative slope for the original boosting algorithm, an initial positive slope and then turning flat or negative for the modified algorithm, and positive slopes for the remaining algorithms.

A final network (Figures 9-10) consists of a weight sharing network (LeCun 1990). The general idea is to reduce the number of free parameters by sharing weights. The network has 757 neurons, 5400 connections and 1014 free weights. This weight-shared network give generally better boosting performance than the other networks, especially the boosting networks.

3. OTHER MACHINE LEARNING TECHNIQUES

In comparing ensemble methods, we restricted the size of the input field to size 10x10 in order that, within the size of database, at some point the training and test performance would asymptote to the same value. In a comparison to other machine learning techniques we used a fixed training set size of 60,000, a 28x28 input field and a test set of size 10,000. The techniques were a large weight-sharing network (260,000 connections, 17,000 weights), boosting using this same network, k-nearest neighbor (best k = 3), optimal margin classifier (Boser, Guyon, and Vapnik, 1992), local learning (Bottou and Vapnik, 1992), tangent distance (Simard, LeCun, and Denker, 1993), a fully connected 784-n-10 (best n is 300) neural network, and linear classifier.

A detailed comparison of these techniques including learning time, recognition time, rejection rates, etc., can be found in Bottou, et.al. (1994). In the boosting algorithm, the database was artificially enlarged using deformations (Drucker, Schapire, and Simard, 1993a). The error rates in decreasing order were: linear classifier at 8.4%; k-nearest neighbor at 2.4%; fully connected network at 1.6%; weight sharing network, optimal margin classifier, tangent distance, local learning tied at 1.1%; and finally boosting at 0.7%.

4. CONCLUSIONS

For a given computational cost, the original form of boosting is best although both boosting algorithms achieve the same asymptotic performance. It may seem that using a subset of the training set for the actual training is "throwing away data", but by using the networks to select the data to be trained on, one can do better than training on all the data. Boosting does better than any method investigated here.

A potentially important observation made in this study is that the training error curve for the original boosting algorithm has negative slope until it asymptotes to some fixed value of training error. We make the following conjectures: (1) The capacity (in some sense which we cannot define yet) increases with increasing training set

size until it reaches an asymptotic value. (2) "Good" algorithms will have this same negative slope characteristic. We know that the difference of the test and training error rate should go to zero as we increase the size of the training set. Therefore, it is highly desirable that the training error curve has this negative slope so that the test error rate follows the training error rate to some small value. (3) Constructive algorithms such as that of cascade architectures (Littman and Ritter 1993), cascade-correlation (Fahlman and Lebiere 1990) and tiling (Mezard and Nadal 1989) that build networks incrementally and algorithms that use queries, filtering, hierarchical structures, or in general adapt the structure of the network or discard data intelligently have (or should have) this same negative slope.

We are attempting to prove or disprove these conjectures.

A final note: We have tried boosting on optimal margin and k-nearest neighbor classifiers. At the present time, we cannot explain why boosting does not work on these non-neural based classifiers.

References

Benediktsson, J.A. and Swain, P.H. (1992), "Consensus Theoretic Classification Methods", *IEEE Trans. Syst. Man Cybern.*, **22**, 4, 688-704.

Boser, B.E., Guyon, I., and Vapnik, V.N. (1992) "A Training Algorithm for Optimal Margin Classifiers", *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* **5**, 144-152.

Bottou, L., and Vapnik, V. (1992), *Neural Computation*, 4 6, 888-900.

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L.D., LeCun, Y., Muller, U.A., Sackinger, E., Simard, P., and Vapnik, V. (1994), "Comparison of Classifier Methods", A Case Study in Handwritten Digit Recognition", submitted to 12th International Conference on Pattern Recognition.

Drucker, H., Schapire R., and Simard, P. (1993a), "Improving Performance in Neural Networks Using a Boosting Algorithm", *Neural Information Processing Systems 5*, S.J. Hanson, J.D. Cowan, and C.L. Giles, (eds.), Morgan Kaufmann Publishers, 42-49.

Drucker, H., Schapire R., and Simard, P. (1993b), "Boosting Performance in Neural Networks", *International Journal of Pattern Recognition and Artificial Intelligence*, 7, 4, 705-720

Fahlman, S.E., and Lebiere, C. (1990), "The Cascade-Correlation Learning Architecture" *Advances in Neural Information Processing Systems 2*, E.S. Touretsky (ed), Morgan-Kaufmann Publishers, 524-532.

Hansen, L.K., and Salamon, P. (1990), "Neural Network Ensembles", *IEEE Trans. Patterns Analysis and Machine Intelligence*, 12, 10, 993-1001.

Le Cun, Y.; Boser. B., Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W; and Jackel, L.D. (1990), "Handwritten Digit Recognition with a Back-Propagation Network", *Neural Information Processing Systems 2*, D. Touretzsky (ed), Morgan Kaufmann, 396-404.

Littman, E. and Ritter, H. (1993), "Generalization Abilities of Cascade Network Architectures", *Advances in Neural Information Processing Systems 5*, S.J. Hanson, J.D. Cowan, C.L. Giles, (eds), Morgan Kaufmann Publishers, 188-195.

Mezard, M. and Nadal, J.-P. (1989) "Learning in Feedforward Layered Networks: The Tiling Algorithm" *Journal of Physics A*, **22**, 2191-2204.

Perrone, M. P. and Cooper, L.N. (1993), "When Networks Disagree: Ensemble Methods for Hybrid Neural Networks, to appear in *Neural Networks for Speech and Image Processing*, R.J. Mammone, ed., Chapman-Hall

Schapire, R. (1990) "The Strength of Weak Learnability", *Machine Learning* **5**, 2, 197-227.

Simard, P.Y., LeCun, Y., Denker, J. (1993), Efficient Pattern Recognition Using a New Transformation Distance", *Neural Information Processing Systems* **5** 50-58, Morgan Kaufmann.

Srihani, S. (1992) "High-Performance Reading Machines", *Proc. IEEE*, **80**, 7, 1120-1132

Suen, C.Y. et. al., (1992) "Computer Recognition of Unconstrained Handwritten Numerals", *Proc. IEEE*, **80**, 7, 1162-1180.

Vapnik, V. (1982) *Estimation of Dependences Based on Empirical Data*, Springer-Verlag

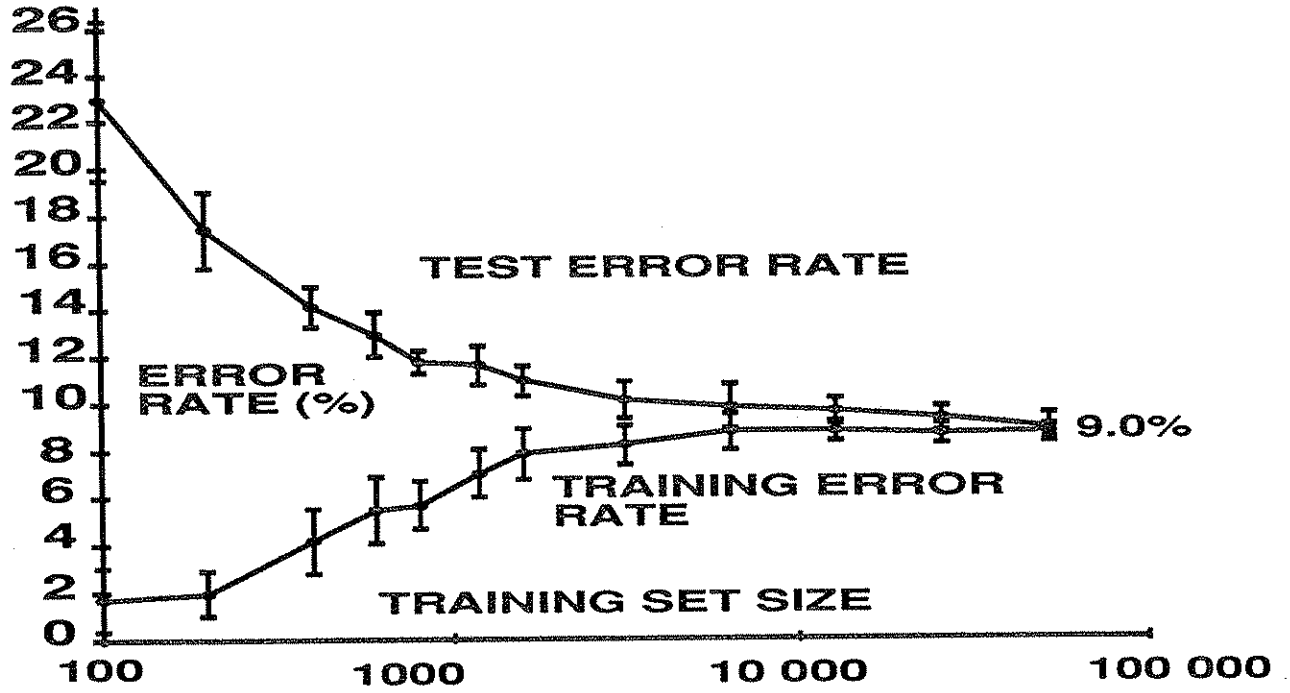

Figure 1. Training and test error rate for a single 100–10–10 network

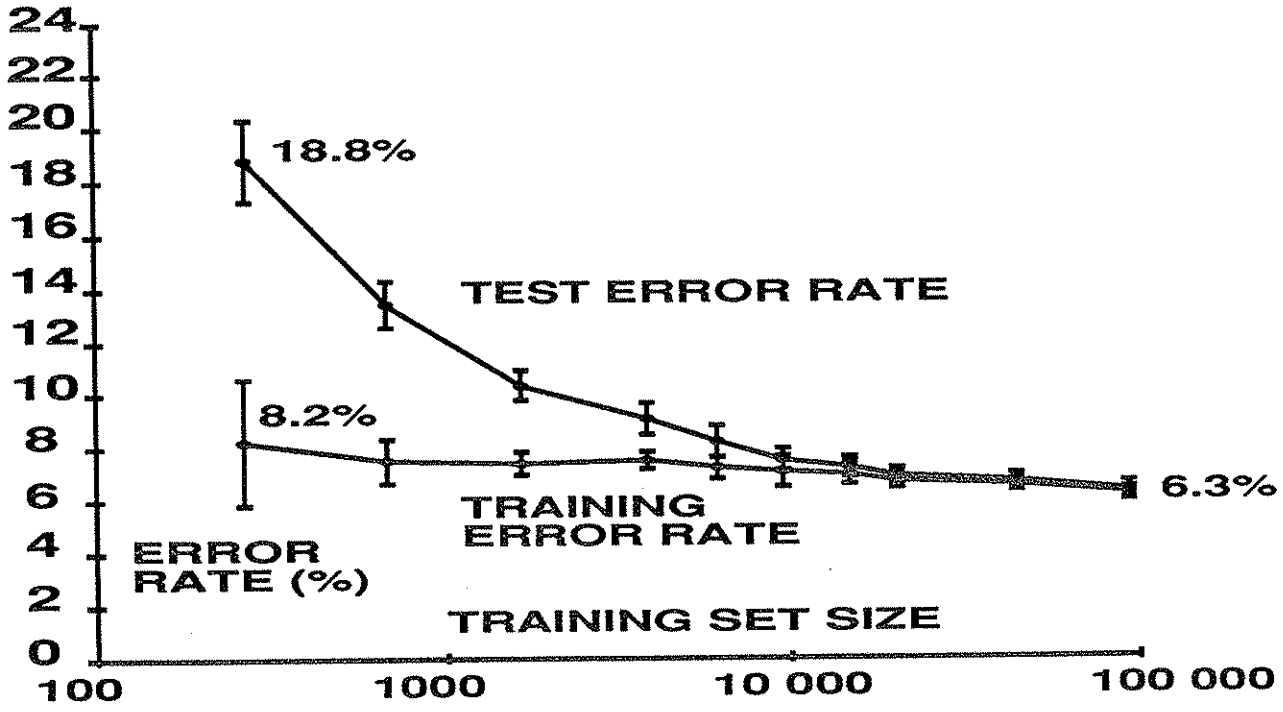

Figure 2. training and test error rate for a boosting network using three 100–10–10 networks.

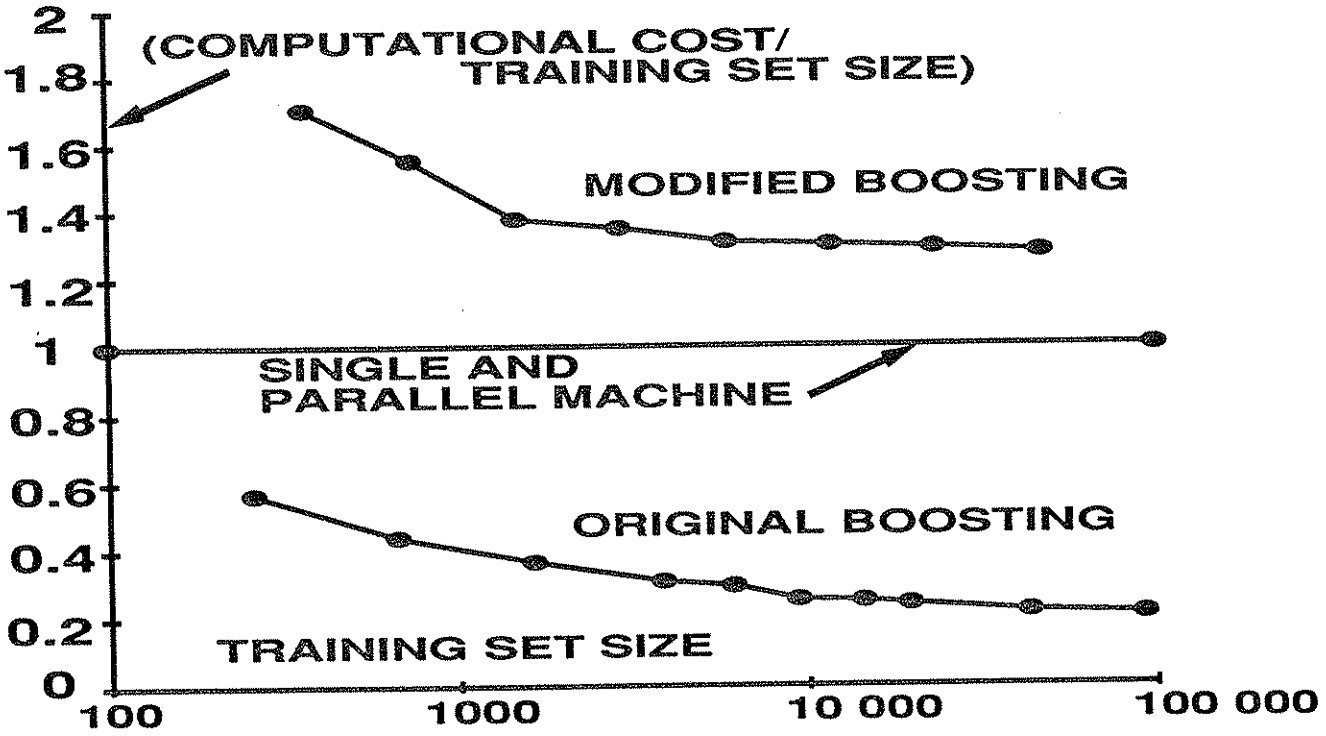

Figure 3. (Computational Cost)/(training set size) versus training set size.

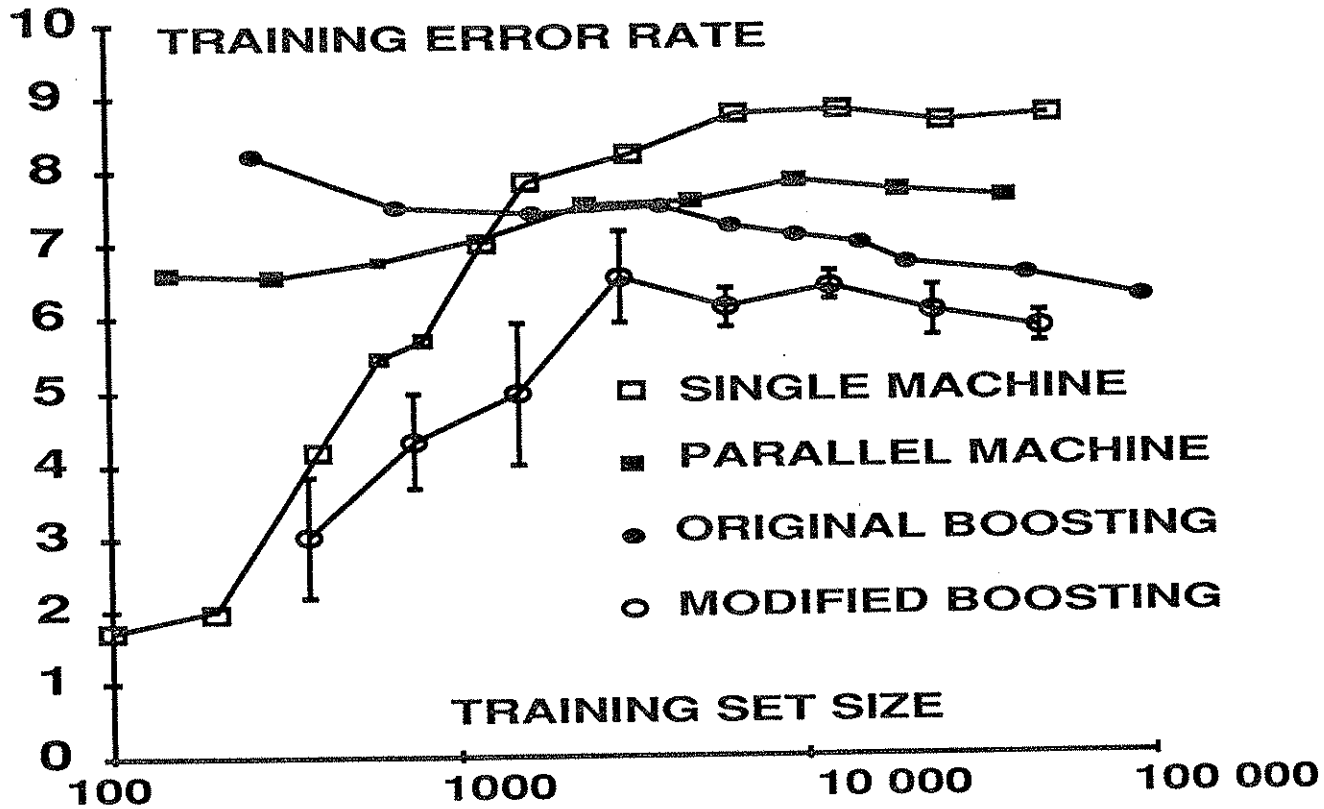

Figure 4. Training error rate versus training set size for four algorithms and a 100–10–10 network.

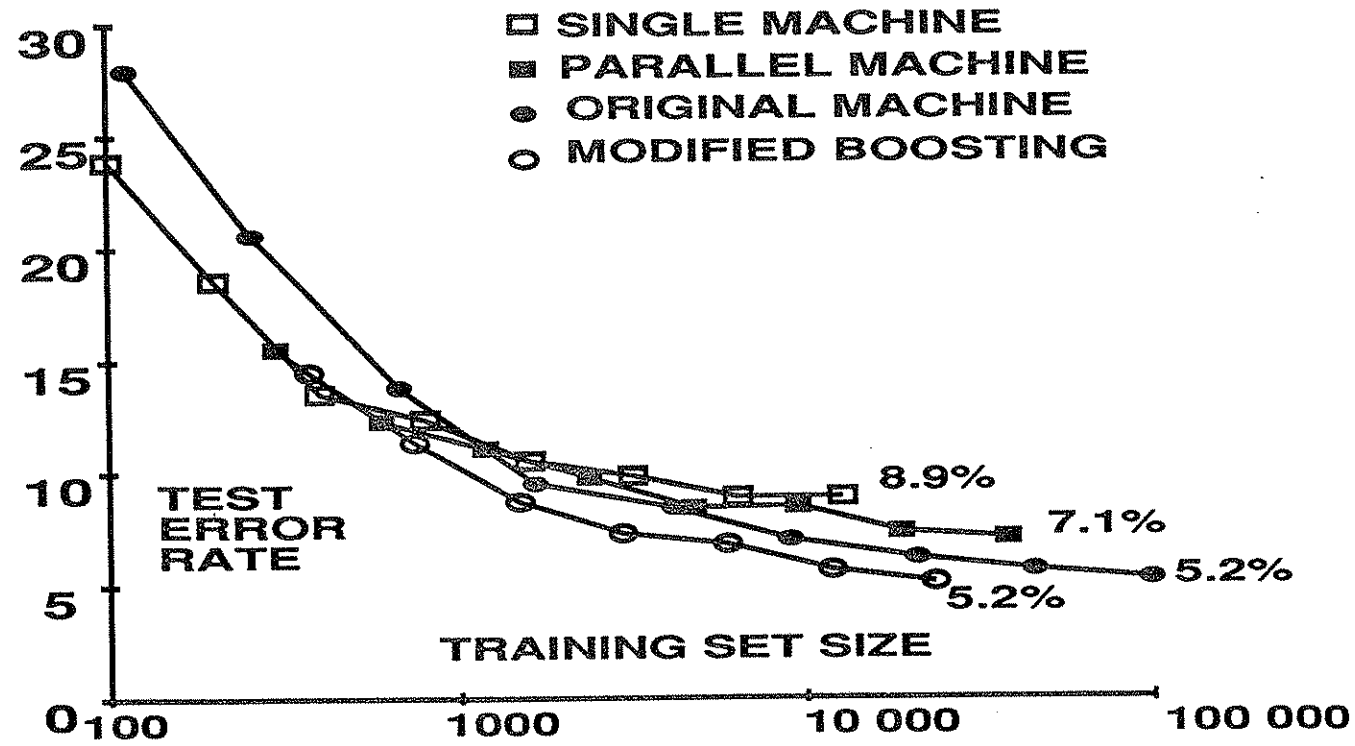

Figure 9. Test error rate versus training set size for four algorithms and a structured network.

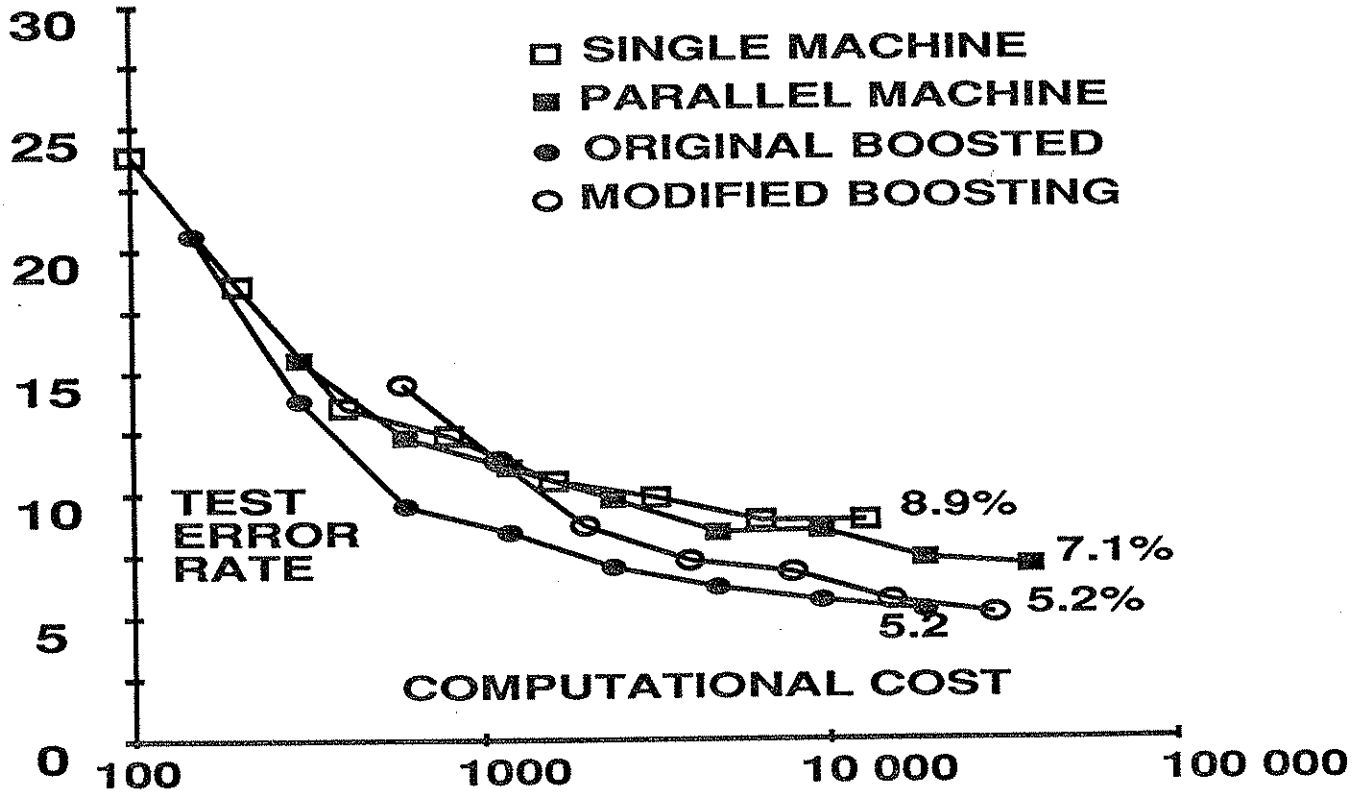

Figure 10. Computational cost versus training set size for four algorithms and a structured network.

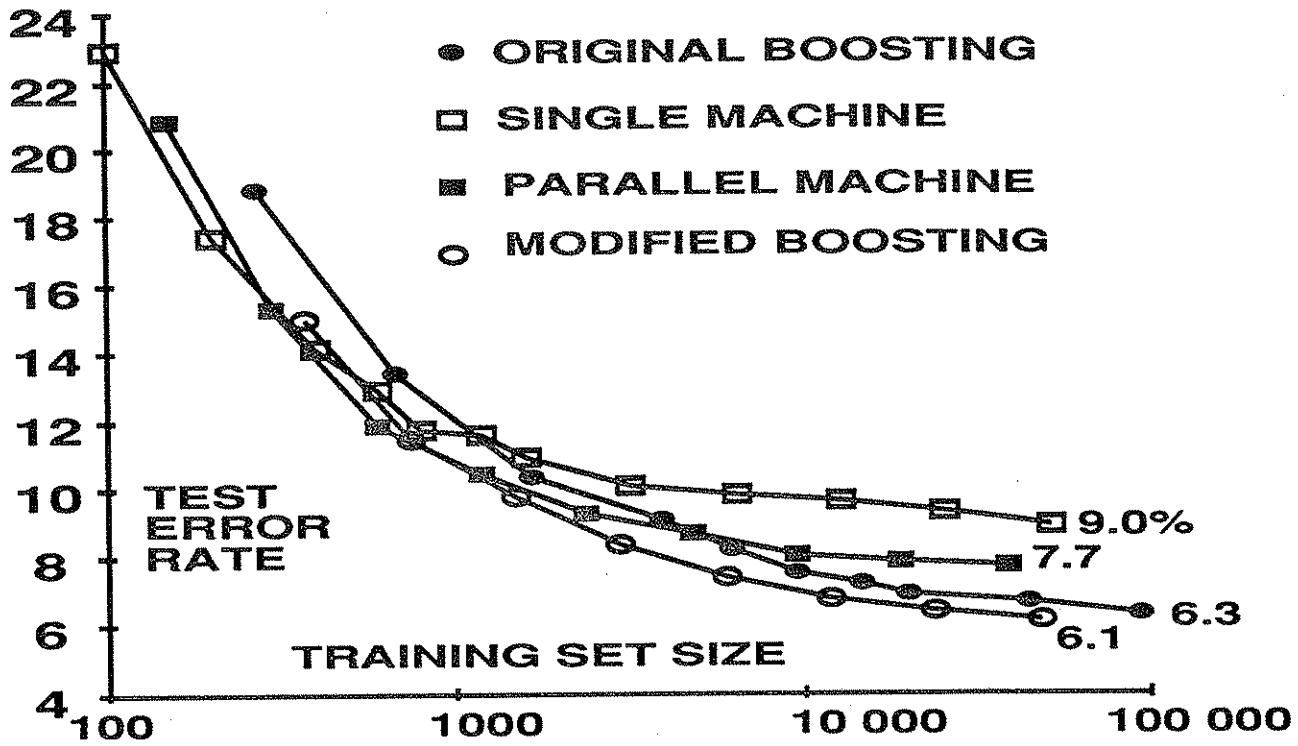

Figure 5. Test error rate versus training set size for four algorithms and a 100–10–10 network.

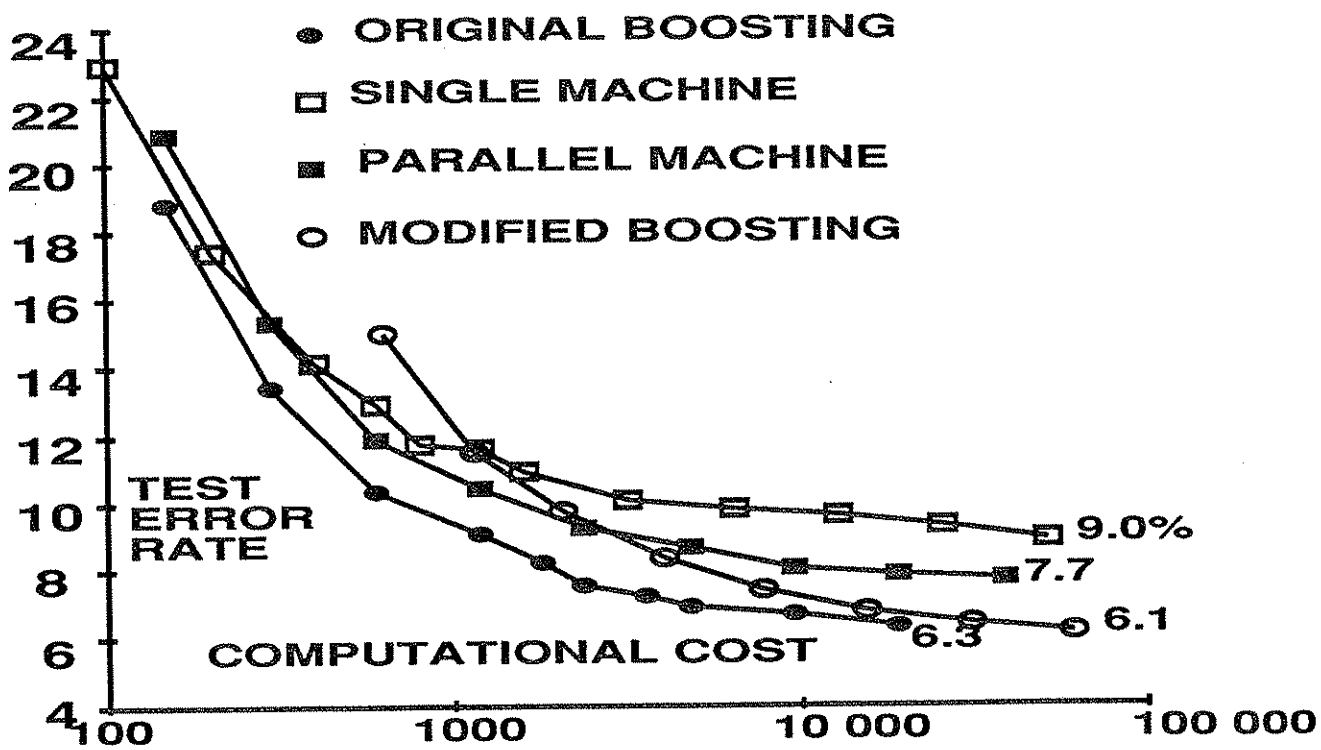

Figure 6. Test error rate versus computational cost for 100–10–10 network.

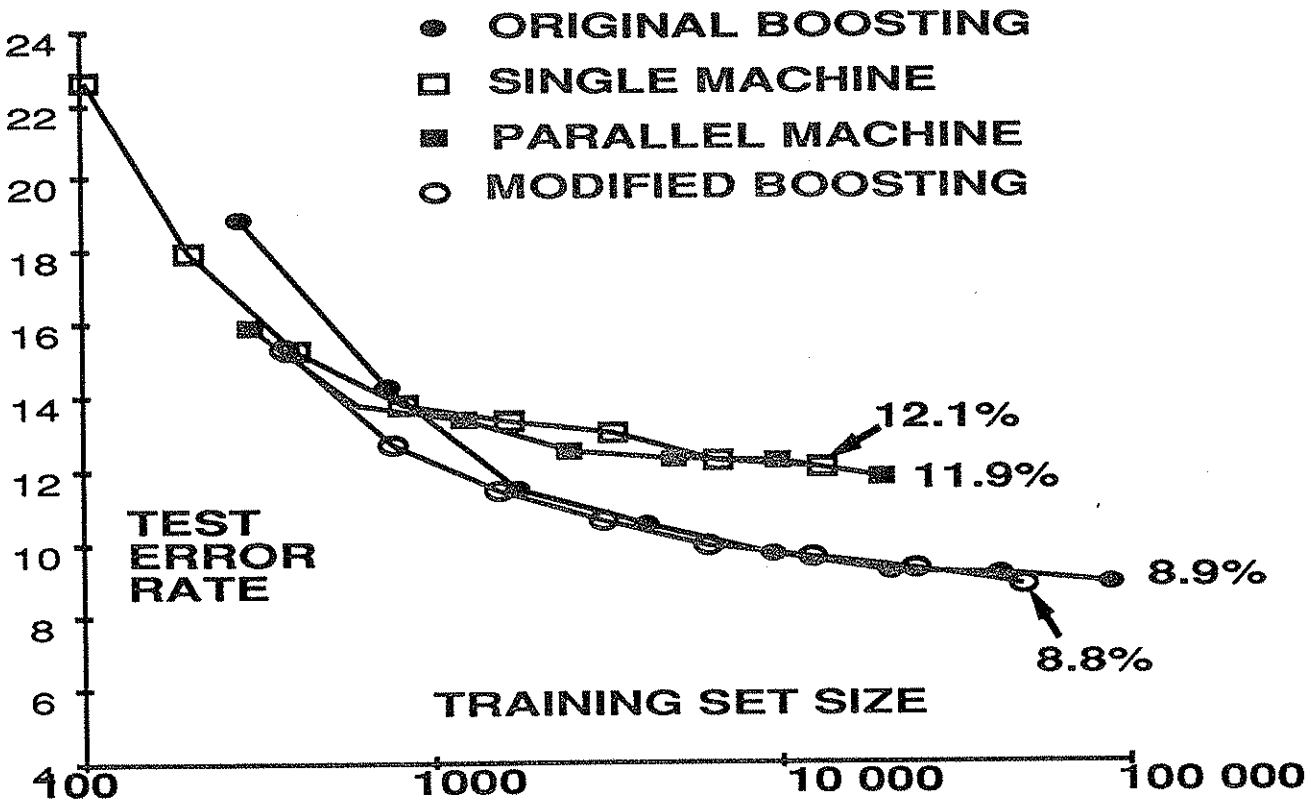

Figure 7. Test error rate versus training set size for four algorithms and a 100–10 network

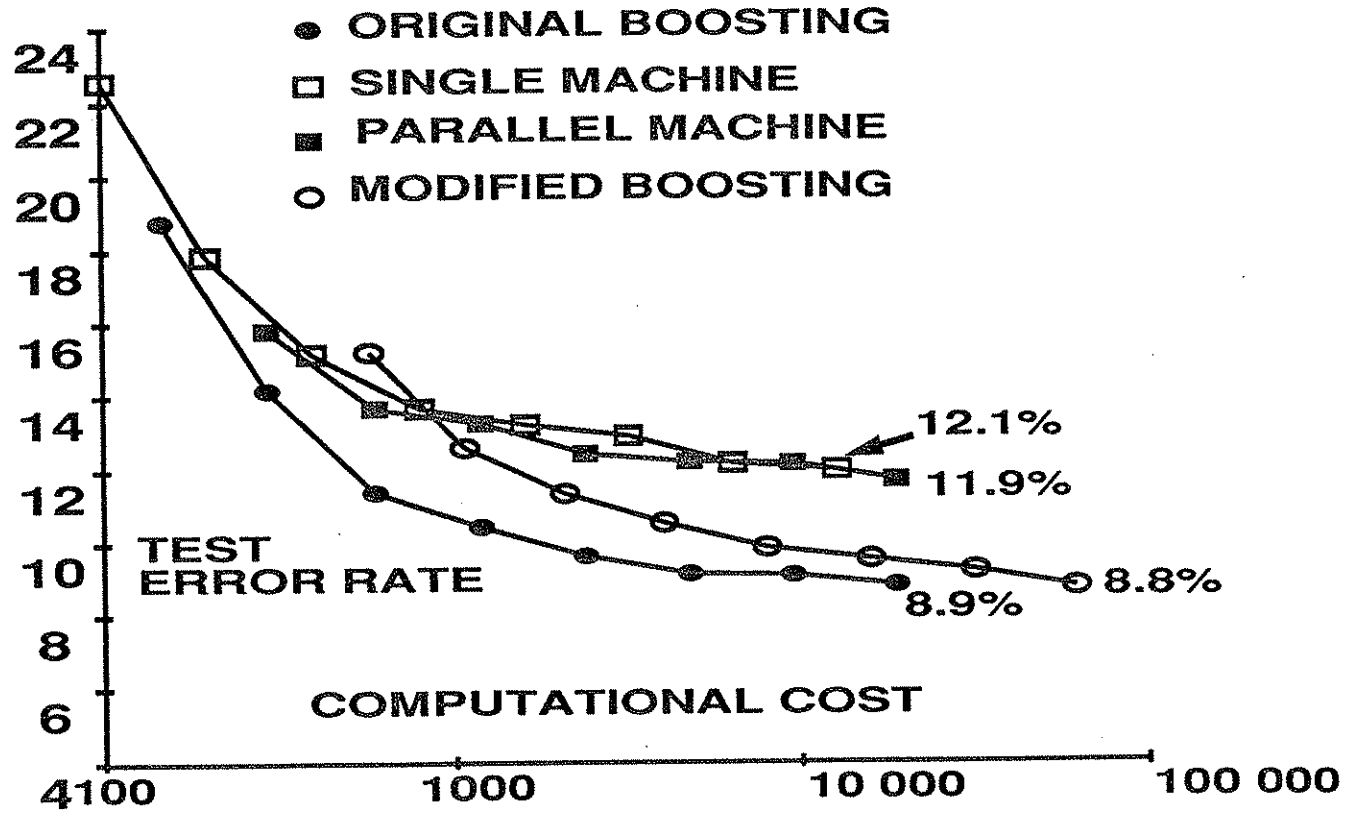

Figure 8. Computational cost versus training set size for four algorithms and a 100–10 network.