TATION PAGE

Form Approved
OMB No 0704-0188

2

ɔ to average ' hour per response, including the time for reviewing instructions, searching existing data sources
ewing the collection of information. Send comments regarding this burden estimate or any other aspect of this
den, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson
ce of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

DATE

3. REPORT TYPE AND DATES COVERED
FINAL 30 SEP 89 - 29 SEP 91

4. TITLE AND SUBTITLE
"ANALYSIS & DESIGN OF NEURAL NETWORKS" (U)

5. FUNDING NUMBERS
61102F
7013/DARPA

6. AUTHOR(S)
Drs. George Cybenko & P. R. Kumar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Illinois
Center for Super Computing/Research Development
506 S. Wright Street
Urbana IL 61801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-89-0536

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release;
Distribution Unlimited

12b. DISTRIBUTION CODE
UL

13. ABSTRACT *(Maximum 200 words)*

The training problem for feedforward neural networks is nonlinear parameter estimation that can be solved by a variety of optimization techniques. Much of the literature on neural networks has focused on variants of gradient descent. The training of neural networks using such techniques is known to be a slow process with more sophisticated techniques not always performing significantly better. It is shown that feedforward neural networks can have ill-conditioned Hessians and that this ill-conditioning can be quite common. The analysis and experimental results lead to the conclusion that many network training problems are ill-conditioned and may not be solved more efficiently by higher order optimization methods. The analysis are for completely connected layered networks, they extend to networks with sparse connectivity as well. The results suggest that neural networks can have considerable redundancy in parameterizing the function space in a neighborhood of a local minimum, independently of whether or not the solution has a small residual.

14. SUBJECT TERMS

15. NUMBER OF PAGES
4

16. PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	**18 SECURITY CLASSIFICATION OF THIS PAGE**	**19 SECURITY CLASSIFICATION OF ABSTRACT**	**20 LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Final Report for AFOSR Grant No. 89-0536
Analysis and Design of Neural Networks

George Cybenko and P.R. Kumar, Investigators
University of Illinois at Urbana

January 1992

1 Introduction

Our investigations into artificial neural networks began with work on the universality of simple classes of neural networks. While that work has been continued and generalized by many researchers in the US and abroad our focus shifted to questions of feasibility in both the theoretical and computational senses. The theoretical questions deal with how much data is needed in order for neural networks and other types of approximating classes to accurately model a specific system with high confidence. The computational questions deal with the efficiency of finding a good approximator from the class of feedforward neural networks.

Our work on the first question has been positively answered. That is, within the so-called PAC learning framework, we have demonstrated that a large class of learning problems can be solved using an amount of data that is not prohibitive. Moreover, the class of such learning systems is also quite large and there exist universal, although not efficient or practical, learning algorithms.

Our work on the second question has resulted in negative results. Namely, we discovered intrinsic properties of feedforward neural networks that imply slow learning rates for conventional learning algorithms. These results coincide with many practitioners' experiences that generic approaches to training neural networks can be prohibitively expensive. It is important to add that our results do not preclude the possibility of improving training efficiency by using preprocessing techniques that might be specialized for a particular problem domain.

This combination of results has led us to explore methods that satisfy desirable theoretical learning properties and are efficiently trainable. In particular we are now investigating the use of large, dense memories in novel, adaptive ways to store, retrieve and interpolate training data. Such approaches appear promising because new memory technologies are imminent and will likely be very compact and inexpensive.

2 Learning, Smooth Simultaneous Estimation, and Generalization

The primary reason for using neural networks is that they can *generalize* based on "training data." This "generalization" pertains to the ability to perform well on *novel* inputs *not* seen in the training data, and is to be contrasted with the ability to fit the training data itself. We are studying issues related to this ability of neural networks to "generalize" based on limited data. Our research derives from the "Probably Approximately Correct" (PAC) paradigm of computational learning.

We have extended the PAC framework to encompass more practical considerations. Our results also expose connections between learning and estimation theory, and they suggest superior training schemes for neural networks.

Results in PAC learning are typically bounds on the number of examples required to "learn" a function when the examples are drawn from an arbitrary distribution. That is, bounds are obtained for the number of randomly drawn training samples, labeled by a "target" function, that are required to guarantee with high confidence that a network trained on the examples will perform well on future samples.

We see several drawbacks to the standard PAC framework. First, the bounds tend to be overly conservative because the distribution of examples is allowed to be arbitrary. Second, the functions are restricted to being binary-valued; in essence, PAC learning applies only to sets. Lastly, it is assumed that the desired target function can be fitted perfectly with the type and size of neural network being employed. This is clearly unrealistic since it implies that zero generalization error is possible.

We have derived conditions for learning functions when some member of a known class of distributions generates the examples, thus extending the PAC framework. This allows one to incorporate prior knowledge into the learning process by specifying the class of underlying distributions. The incorporation of such prior knowledge allows us to obtain less conservative bounds on the size of the training set required for learning. We also allow the networks and targets to be real, vector, and even random functions of their inputs. So, our results apply to the full range of network configurations.

In practice, for a fixed architecture, no network may perfectly match the target function. We should therefore think of learning as the process of selecting the network with the least generalization error. Most neural network training schemes take the approach of minimizing the difference between the network's output and the value of the target function on the training data. That is, they seek a network that minimizes the empirical error on the *training data.*

We show that this is a special instance of a technique we call "smooth simultaneous estimation," i.e., the existence of a "smooth" estimator that produces accurate generalization error estimates for all the networks simultaneously. The smoothness requirement excludes overly complex estimators from consideration. This viewpoint shows that learning is possible if one can accurately estimate the generalization error from the training data for all the networks simultaneously, for then one may simply pick the network with the least estimated error. In many cases, an estimator that varies with the network architecture or draws on prior knowledge of the underlying distribution will produce good simultaneous error estimates, even though the empirical approach fails.

We have found a way to classify the instances in which smooth simultaneous estimation is possible. This characterization is particularly useful because it shows how to construct such an estimator, if it exists. This "canonical estimator" takes the form of a two-step procedure, as follows. First, one selects a finite subset of all the candidate networks based on a portion of the training data. This subset is chosen such that the output of any candidate network on the chosen portion of the training data can be closely approximated by the output of some network from the subset. Second, one estimates the error of each network in the subset empirically from the remainder of the training data. A training scheme based on the canonical estimator then selects the "trained" network by minimizing the error as estimated by the canonical estimator.

We feel that training networks with the canonical estimator will result in networks that perform better on novel inputs. In some simple cases where the empirical estimator works, we have determined bounds which show that the canonical estimator compares favorably with the empirical estimator, in terms of the amount of training data needed to achieve a given level of generalization performance. Additionally, if the canonical estimator does not work (i.e, does not simultaneously estimate), neither will the empirical error estimator.

The canonical estimator is also instrumental in showing how an ensemble of networks can learn many functions simultaneously from the same training data. Within the PAC framework, we have found that this "simultaneous learning" requires only a modest increase in the amount of training data.

We intend to apply the computational theory associated with PAC learning to our idea of learning from the canonical smooth simultaneous estimator. By doing so, we hope to address the trade-offs involved in efficiently training neural networks and achieving good generalization.

3 Ill-conditioning of Training Problems

The training problem for feedforward neural networks is a special case of parameter optimization. Loosely speaking, given a set of training data and an error criterion, training typically involves estimating network parameters that result in a local minimum for the objective function. Backpropagation is a method commonly used in the neural network community for solving such optimization problems. Backpropagation is a combination of efficient gradient computation coupled with a gradient descent method, often done stochastically or cyclically. We have explored and documented the intimate relationship between backpropagation and so-called automatic differentiation methods of more general applicability in scientific computing. Specifically, backpropagation is a special case of reverse mode derivative computation.

Many researchers have observed a slow convergence rate for such backpropagation methods when used with feedforward sigmoidal-type neural networks. A significant body of research projects have attempted to improve the convergence rates by modifying the basic gradient descent technique, using higher order, more sophisticated methods or preprocessing steps that are improvements over random starts. While many of these methods lead to better convergence rates, they are often not convincing because the better rates are either constant factor improvements or the methods themselves are heuristic and appear to be special cases.

Our study of this situation has identified generic ill conditioning of Jacobian and Hessian matrices as a major stumbling block in training problems for such neural networks. Ill-conditioning of these matrices means that the surface of the objective function (optimization criterion) has steep, narrow valleys at many points in the parameter space. This arises from a local over-parameterization by the neural network model. It is important to separate this over-parameterization from the fact that the neural network model may not exactly model the phenomenon (that is, there is a nonzero residual).

Our identification of this ill-conditioning as a problem comes from careful analysis of feedforward network architecture structure and the properties of sigmoidal activation functions. We show that 5 different types of ill-conditioning can arise and our empirical studies show that at least 3 of these types arise in actual, sample problems. Our conclusions in this area of research are that generic methods for training feedforward neural networks are destined to be inefficient and will not likely scale well.

4 Future Work

The results we have obtained in this research effort have pointed us in the direction of using models and methods that satisfy the theoretical requirements of PAC learning systems yet are easily trainable. Loosely speaking, the PAC criterion can be meet by any method that does a reasonable job of reproducing the training data while achieving some sort of data compression (learning) asymptotically. The key then becomes the efficiency with which that compression is done. Backpropagation applied to feedforward networks attempts to do the compression by fitting a highly nonlinear model to the data. We believe that methods based on large memories and purely local properties are good candidates for efficient learning in a broad context. This same general sentiment is shared by researchers in applied statistics who are using adaptive spline techniques which gives us some confidence in this approach. We hope to continue this direction of research in the future. At present, we have a general conceptual framework and a preliminary implementation of a specific scheme for experimentation. That work will be presented at a forthcoming SPIE meeting in Orlando.

5 Educational Component

Four students have been involved in this research effort.

Ryan MacDonald completed his ECE Masters thesis in 1991. His work was the empirical foundation that lead to the research on ill-conditioning of Jacobian and Hessian systems. He was a recipient of the prestigious Luce Fellowship (the first in University of Illinois history) and spent a year in the Far East teaching. He is presently employed by an engineering software house in the Dallas area.

Kevin Buescher is an ECE Ph.D. student expected to complete his thesis work in 1992. He has worked on the PAC learning results which form the core of his thesis research.

Sirpa Saarinen is a CS Ph.D. student expected to complete her Ph.D. thesis work in 1992 also. Her work has involved the analytic studies of network Jacobian and Hessian matrices and she is presently carrying out the analysis and implementation of a memory based adaptive learning system.

Richard Burg is a CS Masters student who has implemented a shared-memory parallel algorithm method for k-d trees that will ultimately be used our future work. He is expected to complete his thesis work in late 1992.

Randall Bramley is a postdoctoral assistant who worked on this project. His Ph.D. work was on linear and nonlinear optimization methods and he contributed to the supported research dealing with ill-conditioning of training problems.

6 Relevant Publications

Buescher, K. L. and Kumar, P. R., "Learning and Smooth Estimation," document in preparation.

Buescher, K. L. and Kumar, P. R., "Estimating the Expected Loss of Many Hypotheses Simultaneously," submitted to the 1992 Conference on Information Sciences and Systems.

Buescher, K. L. and Kumar, P. R., "Simultaneous Estimation of the Error of Hypotheses," to appear, 1992 American Control Conference.

Buescher, K. L. and Kumar, P. R., "Simultaneous Learning of Concepts and Simultaneous Estimation of Probabilities," in *Computational Learning Theory: Proceedings of the Fourth Annual Workshop*, pp. 33–42, Morgan Kaufmann, San Mateo, CA, 1991.

Buescher, K. and Kumar, P. R., "Relating Simultaneous Learning and Simultaneous Estimation for Classes of Sets and Classes of Probabilities," in *Proceedings of the 25th Conference on Information Sciences and Systems*, pp. 108–113, The Johns Hopkins University, March 1991.

Saarinen, S., Bramley, R., and Cybenko, G., "Ill-conditioning in Neural Network Training Problems", to appear in SIAM Journal on Scientific and Statistical Computing, 1992.

Saarinen, S., Bramley, R., and Cybenko, G., "Neural Networks, Backpropagation and Automatic Differentiation", in *Automatic Differentiation*, edited by A. Greiwank and G. Corliss, SIAM Press, Philadelphia, PA 1991.