Neural Networks

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

412

L.B. Almeida C.J. Wellekens (Eds.)

Neural Networks

EURASIP Workshop 1990
Sesimbra, Portugal, February 15–17, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Editors

Luis B. Almeida
Instituto de Engenharia de Sistemas e Computadores INESC
Rua Alves Redol, 9-2, Apartado 10105
P-1017 Lisboa Codex, Portugal

Christian J. Wellekens
Philips Research Laboratory Brussels
Avenue Van Becelaere 2, Box 8
B-1170 Brussels, Belgium

CR Subject Classification (1987): C.1.3, C.3–5, F.1.1, F.2.2, I.2.6–8, L4.0, I.5.4, I.m

ISBN 3-540-52255-7 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-52255-7 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

Preface

This book contains both the full and the invited contributions to the 1990 EURASIP Workshop on Neural Networks, held in Sesimbra, Portugal, February 15-17, 1990. Though sponsored by a European organization (the European Association for Signal Processing, EURASIP), no restrictions were placed on the origin of the participants in this workshop. Instead, the selection of the full contributions was performed by an international Technical Committee. The quality demands that were imposed are reflected in the acceptance ratio, which was only about 40%.

The field of the contributions has not been restricted to an overspecialized topic: one main characteristic of the connectionist community is its multidisciplinarity. Psychologists may identify the essential features of the world to be learned and propose original learning schemes, biologists can describe architectures that have not been studied previously by computer scientists, and engineers may perform simulations and implementations of connectionist architectures, while the help of mathematicians is most welcome to formalize these nonlinear models suggested by nature. Authors of this book belong to all these disciplines.

The two invited papers, by George Cybenko and by Eric Baum, deserve a special mention. They deal with two different aspects of a subject which we consider very important for the consolidation of the field: the formal study of the capabilities of neural networks. George Cybenko introduces the definition of a formal measure of problem complexity which is relevant to neural networks and discusses some of its properties. Eric Baum studies the relationships between training set size, network size and generalization capability. We can only hope that these will form the embryo of a body of theory that will allow neural network problems to be approached with an engineering methodology, instead of the present trial-and-error manner.

Besides the published papers, about 20 posters were displayed at the workshop; they allowed their authors to present their current research effort but they were not included in this publication due to their inherent incompleteness.

It is our pleasure to thank all the authors who coped with the strictly imposed deadlines for supplying their camera ready manuscripts. Their cooperation and good will were rewarded by allowing Springer-Verlag to have the proceedings available for the participants at the workshop.

The original Technical Committee, which had 8 members, was greatly enlarged to allow a more complete review of the submitted papers. We wish to thank all its members. The free cooperation of reviewers is an essential part of research.

We also wish to acknowledge the cooperation given by Joaquim Rodrigues and Fernando Silva. A very special acknowledgment goes to Ilda Gonçalves, who provided invaluable help in every aspect of the preparation of this volume and of the workshop itself, always with a smile. Acknowledgments to INESC, Philips Research Laboratory Brussels and Bell Communications Research are also due, for providing most of the resources needed for this kind of organization.

Lisbon, November 1989 — Luis B. Almeida

Morristown, November 1989 — Christian J. Wellekens

Conference Chairman

Luis B. Almeida, INESC, Portugal

Technical Chairman

Christian J. Wellekens, Philips Research Laboratory, Belgium
and Bellcore, NJ, USA

Technical Committee

Emile Aarts, Philips Natuurkundig Laboratorium, The Netherlands
Bernard Angéniol, Thompson CSF-DSE, France
Hervé Bourlard, Philips Research Laboratory, Belgium
John Bridle, RSRE/SRU, United Kingdom
David Burr, Bellcore, NJ, USA
Gérard Chollet, ENST, France
Renato de Mori, Mac Gill University, Canada
Pierre A. Devijver, ENSTBr, France
Gérard Dreyfus, ESPCI, France
Frank Fallside, University of Cambridge, United Kingdom
Françoise Fogelman, Université de Paris V, France
Michael A. Franzini, Carnegie Mellon University, PA, USA
Stan Gielen, Katholieke Universiteit Nijmegen, The Netherlands
Jean Gobert, LEP, France
Martin Hasler, EPFL, Switzerland
Jean-Paul Haton, CRIN, France
Jeanny Hérault, Institut National Polytechnique de Grenoble, France
Geoffrey Hinton, University of Toronto, Canada
Ron E. Howard, AT&T Bell Labs, NJ, USA
Larry D. Jackel, AT&T Bell Labs, NJ, USA
Jörg Kindermann, GMD, FRG
Yann LeCun, AT&T Bell Labs, NJ, USA
Hong C. Leung, MIT, MA, USA
Jean-Sylvain Liénard, LIMSI, France
Carver Mead, California Inst. of Technology, CA, USA
Michael Mozer, University of Colorado, CO, USA
Heinz Mühlenbein, GMD, FRG
Traian Muntean, Université de Grenoble, France

Erkki Oja, Lappeenranta Univ. of Technology, Finland
Guy A. Orban, Katholieke Universiteit Leuven, Belgium
Rolf Pfeifer, University of Zürich, Switzerland
Philippe Piret, Philips Research Laboratory, Belgium
Richard Prager, University of Cambridge, United Kingdom
Tomaso Poggio, MIT Cambridge, MA, USA
Eric Saund, Xerox Palo Alto Research Center, CA, USA
Sharad Singhal, Bellcore, NJ, USA
André Thayse, Philips Research Laboratory, Belgium
François Vallet, Thompson CSF, France
Paul van Dooren, Philips Research Laboratory, Belgium
Michel Verleysen, Université Catholique de Louvain, Belgium
Alex Waibel, Carnegie Mellon University, PA, USA
Michel Weinfeld, Ecole Polytechnique, France
David Zipser, UCSD, CA, USA

Table of Contents

PART I

INVITED PAPERS

When Are k-Nearest Neighbor and Back Propagation Accurate for Feasible Sized Sets of Examples?

Eric B. Baum
NEC Research Institute
4 Independence Way
Princeton, NJ 08540

Abstract: We first review in pedagogical fashion previous results which gave lower and upper bounds on the number of examples needed for training feedforward neural networks when valid generalization is desired. Experimental tests of generalization versus number of examples are then presented for random target networks and examples drawn from a uniform distribution. The experimental results are roughly consistent with the following heuristic: if a database of M examples is loaded onto a W weight net (for $M >> W$), one expects to make a fraction $\epsilon = \frac{W}{M}$ errors in classifying future examples drawn from the same distribution. This is consistent with our previous bounds, but if reliable strengthens them in that: (1) the bounds had large numerical constants and log factors, all of which are set equal one in the heuristic, (2) previous lower bounds on number of examples needed were valid only in a distribution independent context, whereas the experiments were conducted for a uniform distribution, and (3) the previous lower bound was valid for nets with one hidden layer only. These experiments also seem to indicate that networks with two hidden layers have Vapnik-Chervonenkis dimension roughly equal to their total number of weights.

We then consider the convergence of the k-nearest neighbor algorithm to a classifier making a fraction ϵ of errors when examples are drawn from the uniform distribution on S^n, the unit sphere in n dimensions, and classified according to a simple target function. We prove that if the target function is a single half space, then for k appropriately chosen ($k \sim \frac{n}{\epsilon^2} ln(\epsilon^{-1})$), k nearest neighbor yields an ϵ accurate classifier using a database of $M = O(\frac{n}{\epsilon^2} ln(\epsilon^{-1}))$ classified examples. However, when the target function is a union of two half spaces, k nearest neighbor requires a number of examples exponential in n to achieve high accuracy.

1 Introduction

When we learn in a natural environment, we are confronted with a rich and varied world. There is a nearly endless number of features we might observe. Almost all of these, however, will be irrelevant to any specific learning goal. Any particular concept we wish to learn may depend only on some simple, i.e. low parameter, function of some subset of the possible features. For example, one would like a learning algorithm which,

shown images of hand drawn numerals, could learn to read them correctly. Is it possible to just feed in to our learning algorithm the images, containing as many features as there are pixels, or must we preprocess and extract relevant features such as line ends? A key question then is whether it is possible to learn simple concepts in a high dimensional feature space using resources- time and information- bounded by some (hopefully low order) polynomial in n, the number of features.

The observation that various simple statistical pattern recognition algorithms seem to require a number of examples exponential in the dimension of the feature space has been dubbed "the curse of dimensionality" (see e.g. [Duda and Hart, 1973, p95]). Recently it has been possible in a quite general context to analyze how many examples are necessary for training a neural net, provided they can be successfully loaded. The answer does not seem catastrophic, and in fact bodes well for learning. Theorems have been proved that give upper and lower bounds that differ only by constant and log factors, on the number of examples necessary to achieve generalization. Very roughly speaking, these results indicate that if M random examples can be loaded onto a feedforward neural net with W weights and one output, one expects generalization so that about a fraction W/M of future test examples will be missclassified. In section 2 we will very briefly review these results and give some intuitive arguments as to why they hold. In section 3 we discuss some simple experiments which clarify the practical consequences of these results. The experiments indicate that the large constant factors appearing in the theorems are close to one in actual practice, but also seem to indicate that the degree of generalization actually achieved depends in some measure on the complexity of the target function. The key assumption in these theorems, of course, is that we are able to load the examples. Thus, while we have achieved some handle on how much information is necessary for learning, we have almost no understanding of when we can learn in a feasible amount of time.

The PAC learning model proposed by [Valiant,1984] provides a reasonable theoretical model in which to consider this question. We assume we are given examples drawn from some probability distribution D over some feature space, $\Re^n$ or $\{1,-1\}^n$, say, and classified according to some Boolean target function f. Thus examples consist of pairs $(\vec{x}, f(\vec{x}))$, where $\vec{x}$ is a feature vector drawn acording to some natural distribution D of examples and $f(\vec{x})$ is a classification that $\vec{x}$ is either a positive or negative example of the target concept. We are told that $f \in F$, where F is some simple class of Boolean functions. We ask when there is some learning algorithm A which can look at examples and produce, in time polynomial in n, ϵ^{-1}, and δ^{-1} a hypothesis g which will with probability $1-\delta$ correctly classify at least a fraction $1-\epsilon$ of future examples drawn from D. The acronym PAC stands for "Probably Almost Correct", i.e probably (with confidence $1-\delta$) the learning algorithm generates a classifier which is almost correct (i.e. makes a fraction smaller than ϵ of classification errors).

Thus for example the class F might consist of the class of half spaces: $F = \{f(x) : f(x) = \theta(w \cdot x - t), w \in \Re^n, t \in \Re\}$ where $\theta(x)$ is the Heaviside function, $\theta(y) = 1, y \geq 0; \theta(y) = 0, y < 0$. F therefor consists of the class of functions computable by a single linear threshold unit. Notice here that there is one simple feature which sums up the relevant information, namely $w \cdot x$. Other components of $\vec{x}$ are irrelevant. For this simple case, one can in fact give fast learning algorithms. Under reasonable assumptions about the distribution D, the Perceptron algorithm can be proven to learn very rapidly [Baum,

1989b]. With no assumptions on D at all, learning algorithms based on Karmarkar's algorithm can be proved to work rapidly [Blumer et al, 1987]. However, already when the class F consists of a union of two half spaces, $F = \{f : f(x) = 1$ if $w_1 \cdot x - t_1 > 0$ or $w_2 \cdot x - t_2 > 0$, else $f(x) = 0$ for some $w_1, w_2 \in \Re^n, t_1, t_2 \in \Re\}$, it is far from clear that an algorithm can be given which will learn in polynomial time. In this case there are two simple features which would suffice to render the problem trivial, but it is unclear whether there is any learning algorithm which can rapidly uncover them.[1]

If we could prove that it is not possible to learn rapidly the class F of unions of two half spaces, that would be evidence that automatic learning is impossible. My personal philosophy in answering this would be: humans are capable of learning in the natural world. Therefor, a proof within some model of learning that learning is not feasible is an indictment of the model. We should examine the model to see what constraints can be relaxed and made more realistic. One area in which the PAC learning model is generally too restrictive is in making no assumptions regarding the distribution D of examples. In Valiant's definition, and in most of the work of the Computational Learning Theory community, no assumption is made regarding D, and a class F of functions is called learnable only if a learning algorithm exists which works for every distribution D. Some elegant results are possible in this context, but we will argue first that this requirement is much too restrictive for natural learning, and second that available evidence indicates that natural distributions are frequently trivial, much more tractable even than uniform distributions. We will mostly work in this paper, with simple uniform distributions.

On the other hand, if we gave an algorithm that was able to learn a union of two half spaces from examples, but made detailed use of the assumption that the function to be learned was a union of two half spaces, it is problematic whether that would be helpful in a real world context. In practical situations, we desire to learn functions which may be simple, but are not drawn from any particular, explicitly parametrized class of functions. Many workers from the computational learning theory community hope to deal with this problem by giving algorithms for a specific class of functions which are robust against noise or distortion of the functions. Known positive results in this direction are however extremely limited.[2]

In this paper we will study the performance of an algorithm that intuitively might be expected to be effective in natural environments, learning relatively smooth functions, but which makes no evident explicit assumptions about the class F of functions to be

[1] In fact, the problem: "given a set of classified examples, are they consistent with classification by a union of two half spaces?" has been proved NP -complete [Blum and Rivest,1988]. Together with results of [Pitt and Valiant,1986] this implies that one can not PAC learn a union of two half spaces using as hypothesis function a union of two half spaces in polynomial time in the distribution independent sense (unless P=NP). It is of course an open question whether one can learn this class of functions using more general hypothesis functions. See e.g. [Baum, 1989a] and [Baum, 1989c] for more on this point.

[2] Also the theorem that one can only learn concept classes of finite V-C dimension [Blumer et al, 1987] places severe constraints on the malicious distortion any algorithm can tolerate in the distribution independent framework.

learned.[3] We will study the k-nearest neighbor algorithm in the PAC learning model, for simple uniform distributions. This algorithm is the following: one has a database of M classified examples. One hypothesizes the following classification for new examples: find the k nearest examples in the database, and guess that x is positive if more than half of these are positive examples, else guess x is a negative example. This algorithm has the following nice property. It can be proved that as M goes to infinity, k-nearest neighbor yields a classifier having error rate no worse than twice the Bayes risk. This result is independent both of the choice of D and of F (except for reasonable regularity conditions) [Cover and Hart,1967].

Unfortunately it is easy to see, that even if $n = 1$, that is we consider the case of only one feature, the rate of convergence of this algorithm can be arbitrarily slow [Cover, 1968]. Our interest here will be in making reasonable simplifying assumptions about D and F, and asking whether k-nearest neighbor will converge to give error rate less than ϵ in time bounded by a polynomial in n and ϵ^{-1}.

Our results are as follows. We take D to be the uniform distribution on S^n, the unit n-sphere. For F the class of half spaces, for k appropriately chosen ($k \sim n\epsilon^{-2}ln(\epsilon^{-1})$), k nearest neighbor converges to ϵ accuracy using $M = O(n\epsilon^{-2}ln(\epsilon^{-1}))$ examples. When F is the class of unions of two half spaces, however, the k- nearest neighbor algorithm will require a number of examples which is $\Omega(\epsilon^{-n})$ to converge to ϵ accuracy. This is traced to a certain breaking of symmetry. In a certain sense, which will be detailed, the k- nearest neighbor algorithm is locally, but not globally able to solve this problem. It is unclear whether some variant might be devised which will work, or whether one indeed must use the global information that the target function is a union of half spaces, or even if this global knowledge will suffice. More generally, it appears that k-nearest neighbor will not be able to learn any function (with polynomially many examples) unless a strong and unrealistic symmetry is respected.

Section 2 briefly reviews some results regarding the sample complexity of learning. Section 3 discusses some recent experimental results. These indicate some natural and optimistic extensions of our previous theorems to uniform distributions and multilayer nets. Sections 4 and 5 can be read independently of sections 2 and 3. Section 4 presents our positive results on convergence of k-nearest neighbor for a single half space. Section 5 discusses convergence for a union of half spaces and presents our negative conclusion that nearest neighbor will require an exponential number of examples except in trivial circumstances such as described in section 4. Section 6 is a brief discussion.

2 Sample Complexity of Learning

In [Baum and Haussler, 1989] the following result is demonstrated. Assume random examples are chosen from some probability distribution D on $\Re^n \times \{1, -1\}$.[4] Say one

[3] Also, in contrast to many learning algorithms previously considered, especially within the computational learning community, the nearest neighbor algorithm produces a hypothesis function which is not drawn from any evident fixed class of functions.

[4] This theorem thus holds in a more general context than PAC learning, which assumes a distribution D on $\Re^n$ and a deterministic classification by some target function.

attempts to load[5] these on a feedforward net of linear threshold units[6] with W weights and N units. If one can find a choice of weights such that at least a fraction $1-\epsilon/2$ of a set of m random training examples are correctly loaded, for m sufficiently large, then one has high confidence that the net will correctly classify all but a fraction ϵ of future examples. In fact, for $m \geq \frac{32W}{\epsilon} ln \frac{32N}{\epsilon}$ one has confidence at least $1-8e^{-1.5W}$; and for $m \geq \frac{64W}{\epsilon} ln \frac{64N}{\epsilon}$, one has confidence at least $1-8e^{-\epsilon m/32}$. Thus, assuming only that one's training set and testing set are drawn from the same distribution, one has assurance which is exponential in the size of the training set that, if we are able to successfully load the training set, we expect to achieve good generalization.

The intuition behind these results is straightforward. The point is that it is highly unlikely that a choice of the W weight values would exist which loads the training set unless the net also agreed with the underlying distribution. For any particular choice of weights, we have some hypothesis net. Say hypothesis net A had probability greater than ϵ of missclassifying a random example. Then the probability that A would correctly classify m examples is less than $(1-\epsilon)^m$, which is exponentially small as m grows.[7] Now although, since we use real valued weights, there are an uncountable number of possible choices of weights, it turns out that the number of functions implementable on any m examples by W weight, N node nets is bounded by $(Nem/W)^W$. This is a generalization of Cover's well known capacity result for simple Perceptrons [Cover, 1965] to nets with hidden layers [Baum, 1988]. Thus, for fixed W, the effective number of nets is bounded by a polynomial in m. Since the probability that any given net would load the training examples but not generalize well is exponentially small, and the effective number of nets is only polynomially large, it is clear that for sufficiently large sample size, the probability that any net would exist which loads the training sample but doesn't generalize is exponentially small.[8] This is the intuition behind the theorem.

We were also able to analyze in the same way learning procedures which start with W' weights, but in learning kill off many synapses arriving at a hypothesis net with W weights and N nodes. Our previous conclusion that good generalization could be expected for large enough sample size holds again, for only slightly larger sample size: $m \geq \frac{32W}{\epsilon} ln \frac{32NW'}{\epsilon}$. This bound differs from the previous bound only in the logarithmic

---

[5] We say an example is loaded if, when we present the example $\vec{x}$ to the input of the net, the output of the net is the correct classification (1 or 0). In the discussion, we have fixed the topology of the net and ask for some choice of weights such that a large fraction of our example set is loaded.

[6] These results have recently been generalized to feedforward nets of sigmoid functions [Haussler, 1989].

[7] For simplicity, in our intuitive explanation, we talk of loading the full set. Similarly the probability A will load a fraction $1-\epsilon/2$ of the examples is exponentially small. See [Baum and Haussler, 1989] for details.

[8] More precisely, the theorem is proved by a cross validation technique [Blumer et al., 1987]. One chooses a set of $2m$ examples and considers all the ways one can use a subset of size m as training set and its complement as testing set. The total number of functions implementable on the whole set is then bounded by $(Ne2m/W)^W$, but the probability is less than $(1-\epsilon)^m$ that any one such function fools us, i.e. loads the training set but fails on the testing set.

factor, so that very few extra examples are necessary. This follows as the number of ways of choosing W weights remaining from the W' initially present is bounded crudely by $(W')^W$, so that the total number of functions implementable on m points by such nets is bounded by $(W')^W(Nem/W)^W$. Because only W and not W' appears in the exponent, $WlnW'$ appears in the bound.

Notice that these results do not say anything about when it will be possible to load the examples or how to load them. The result is that *if the examples can be loaded* then good generalization is assured.

We have also given the following lower bound on the number of examples needed to assure generalization [Baum and Haussler, 1989]. Consider training a net which has n inputs completely connected to a hidden layer of k units which are then connected to the output unit. Any learning algorithm which uses fewer than (roughly) $m_L \sim W/\epsilon$ training examples[9] will be fooled by some distributions. That is to say, one can construct a distribution D such that there exists a choice of weights such that the net exactly agrees with the target classification, i.e. the error rate is zero. On the other hand, there will be a finite probability that the learning algorithm will find some other choice of weights, and will in fact output a hypothesis net which make a fraction greater than ϵ of errors. Thus one can not achieve high confidence of valid generalization.

The intuition behind this result is also clear. Using a net with W weights, thus roughly W parameters, to fit fewer than $O(W/\epsilon)$ examples, is overfitting and one can not guarantee generalization. More precisely, one can find a set S of kn input vectors (in fact any kn points in general position will do [Baum, 1988]) such that for any Boolean function on S, there is a choice of weights which implements it. Thus knowledge of the value of the target function on a subset of these kn points gives no knowledge whatever about its value on the other points, since all possible extensions can be realized by some choice of weights. Now one can specify a distribution D on the set S having the property that with some fixed probability a random set of m_L examples will not include any examples from some subset S_1 which has probability measure greater than 2ϵ [Ehrenfeucht et al.,1988] Thus the set of examples one sees simply does not contain sufficient information to specify an extension to the unseen input vectors which will achieve less than ϵ error rate.

Notice that both the upper and lower bounds on number of examples necessary for learning depend on the size of the net *trained*. (We call this the trainee net.) The complexity of the target function does not enter these bounds. Of course, if the target function is very complex, we will not be able to load the examples, in which case the theorems will not apply.

For practical applications, there are several problems with these results. One problem is that although the upper and lower bounds are reasonably tight in their scaling behavior, differing only by a logarithmic factor, the constant coefficients differ by a factor of a thousand. No serious effort has been made to address this, and no doubt with some work this could be substantially improved. Still, it appears that it would be difficult to rigorously prove results with tight constants (i.e. with the upper bound

[9] More precisely $m_L = \frac{2\lfloor k/2 \rfloor n-1}{32\epsilon}$ examples. Note $W = k(n+1)$ for the one hidden layer, one output net considered.