Foreword

July 1997 saw the start of a six month international research programme entitled *Neural Networks and Machine Learning*, hosted by the Isaac Newton Institute for Mathematical Sciences in Cambridge. During the programme many of the world's leading researchers in the field visited the Institute for periods ranging from a few weeks up to six months, and numerous younger scientists also benefited from a wide range of conferences and tutorials.

Amongst the many successful workshops which ran during the six month Newton Institute programme, the one week workshop on the theme of *On-line Learning in Neural Networks*, organized by David Saad, was particularly notable. He succeeded in assembling an impressive list of speakers whose talks spanned essentially all of the major research issues in on-line learning. The workshop took place from 17 to 21 November, with the Newton Institute's purpose-designed building providing a superb setting.

This book resulted directly from the workshop, and comprises invited chapters written by each of the workshop speakers. It represents the first book to focus exclusively on the important topic of on-line learning in neural networks. On-line algorithms, in which training patterns are treated sequentially, and model parameters are updated after each presentation, have traditionally played a central role in many neural network models. Indeed, on-line gradient descent formed the basis of the first effective technique for training multi-layered networks through error back-progagation. It remains of great practical significance for training large networks using data sets comprising several million examples, such as those routinely used for optical character recognition.

During the early years of the development of back-propagation, many heuristics were proposed to improve its performance, particularly in terms of its convergence speed. Often these were lacking in theoretical foundation and so their generality and applicability were not always clear. More recently there have been many complementary attempts to provide a theoretical analysis of on-line learning, leading to a deeper understanding of the algorithms, and to the proposal of more theoretically motivated variants. Such analyses have come from a variety of complementary viewpoints, most of which are represented in the chapters of this book.

In drawing this material together, this book will prove invaluable both to researchers interested in on-line learning techniques, and to students wishing to broaden their knowledge of neural networks and machine learning.

Christopher M. Bishop
August 1998

Introduction

David Saad

Neural Computing Research Group, Aston University
Birmingham B4 7ET, UK.
saadd@aston.ac.uk

Artificial neural networks (ANN) is a field of research aimed at using complex systems, made of simple identical non-linear parallel elements, for performing different types of tasks; for review see (Hertz et al 1990),(Bishop 1995) and (Ripley 1996). During the years neural networks have been successfully applied to perform regression, classification, control and prediction tasks in a variety of scenarios and architectures. The most popular and useful of ANN architectures is that of layered feed-forward neural networks, in which the non-linear elements (neurons) are arranged in successive layers, and the information flows unidirectionally; this is in contrast to the other main generic architecture of recurrent networks where feed-back connections are also permitted. Layered networks with an arbitrary number of hidden units have been shown to be universal approximators (Cybenko 1989; Hornik et al 1989) for continuous maps and can therefore be used to implement any function defined in these terms.

Learning in layered neural networks refers to the modification of internal network parameters, so as to bring the map implemented by the network as close as possible to a desired map. Learning may be viewed as an optimization of the parameter set with respect to a set of training examples instancing the underlying rule. Two main training paradigms have emerged: batch learning, in which optimization is carried out with respect to the entire training set simultaneously, and on-line learning, where network parameters are updated after the presentation of each training example (which may be sampled with or without repetition). Although batch learning is probably faster for small and medium training sets and networks, it seems to be more prone to local minima and is very inefficient in the case of training large networks and for large training sets. On-line learning is also the more natural approach for learning non-stationary tasks, whereas batch learning would require retraining on continuously changing data sets.

On-line learning of continuous functions, mostly via gradient based methods on a differentiable error measure is one of the most powerful and commonly used approaches to training large layered networks in general, e.g., (LeCun et al 1989), and for nonstationary tasks in particular; it is also arguably the most efficient technique in these cases. However, on-line training suffers from several drawbacks:

- The main difficulty with on-line training is the sensitivity of most training methods to the choice of training parameters. This dependence may not only slow down training, but may also have bearing on its ability to converge successfully to a desired stable fixed point.
- Most advanced optimization methods (e.g., conjugate gradient, variable metric, simulated annealing etc) rely on a fixed error surface whereas on-line learning produces an inherently stochastic error surface.
- The Bayesian approach provides an efficient way of training and has been applied quite naturally and successfully within the framework of batch learning. Extensions to the on-line case, where explicit information on past examples is not stored, have been limited so far.

These shortcomings of current on-line training methods and the quest for more insight into the training process itself motivate the analytical study of these methods presented in this book. This collection is based on presentations given during the workshop on 'On-line learning in neural networks' as part of the Newton Institute program on Neural Networks and Machine Learning in November 1997.

The second chapter of the book opens with a thorough overview of traditional on-line training methods starting from the early days of neural networks These include Rosenblatt's perceptron, Widrow's Adaline, the K-means algorithm, LVQ2, quasi-Newton methods, Kalman algoritms and more. A unified framework encompasing most of these methods which can be analyzed using the tools of stochastic approximation, is presented and utilized to obtain convergence criteria under rather weak conditions.

Chapter 3 provides a different point of view for describing the parameter training dynamics based on the master equation, which monitors the evolution of their probability distribution. This chapter examines two different scenarios: In the first case, one derives exact dynamical equations for a general architecture when the learning rule is based on using only the sign of the error gradient. The analysis is carried out by monitoring the evolution of all surviving moments of the parameter probability distribution, providing an exact solution of the moments evolution. In the second case, one employs a perturbation approach based on monitoring the evolution of leading moments of the parameter probability distribution in the asymptotic regime. This is carried out for both constant and decaying learning rate and enables one to obtain the typical generalization error decay and convergence criteria for different polynomial annealing schedules which become exact asymptotically.

A statistical based description of on-line training techniques, with emphasis on more advanced training methods, is presented in chapter 4. A rigorous comparison between the asymptotic performance of batch and on-line training methods is carried out for both variable and fixed learning rates, showing that

on-line learning is as effective as batch learning asymptotically. The chapter also introduces a practical modification of an established method (Barkai et al 1996) for learning rate adaptation and its analysis. The new method is based on gradient flow information and can be applied to learning continuous functions and distributions even in the absence of an explicit loss function. The method is first analyzed and then successfully applied in the subsequent chapter to handle the real-world problems of blind source separation and learning in non-stationary environments, demonstrating the method's potential.

One of the main difficulties with using on-line learning methods for practical applications is sensitivity to the choice of training parameters such as teh learning rate. These parameters often have to be varied continuously to ensure optimal performance. Chapter 6 offers a practical method for varying the parameters continuously and automatically and examines the performance of the suggested algorithm on some benchmark problems.

Statistical mechanics offers an alternative description of on-line learning which enables us to examine all stages of the training process. This description, which may formally be derived from the master equation description of the stochastic training process (Mace and Coolen 1998), is based on monitoring the evolution of a set of macroscopic variables, sometimes termed order parameters, which are sufficient to capture the main features of the training process. This framework usually relies on a teacher-student scenario, where the model (student) parameters are modified in response to examples generated by the underlying rule (teacher) simulated by a parallel network which generates the training examples. The first in a series of chapters which make use of statistical mechanics techniques focuses on the analytical derivation of globally optimal learning parameters and learning rules for two layer architectures, known as soft committee machines (Biehl and Schwarze 1995; Saad and Solla 1995), these are two layer networks with unit hidden to output weights. Variational methods are applied to the order parameter dynamics in order to determine optimal learning rate schedules under different learning scenarios. Locally optimal methods are shown to be inadequate for complicated network architectures.

Similar techniques are employed in chapter 8 for studying the effect of noise on locally optimal training methods in tree committee machines with a general number of hidden nodes. This architecture, of two layer networks of binary elements with no overlapping receptive fields and unit hidden to output weights, realizes a discrete mapping in contrast to the continuous one realized by the soft committee machine considered in the previous chapter. The asymptotic properties of the optimal training rule and the robustness of the process to multiplicative output noise are studied within the statistical mechanics framework.

Next, in chapter 9, the statistical mechanics description is employed to examine the efficacy of several second order training methods aimed at speeding up training, for instance Newton's method, matrix momentum (Orr and Leen

1997) and natural gradient descent (Amari 1998). This study quantifies the advantage gained by using second order methods in general, and natural gradient descent in particular, in non-asymptotic regimes. A practical cheaper alternative to the latter, based on insights gained form information geometry, is presented in the subsequent chapter and analyzed using similar theoretical tools for various training scenarios, showing a significant improvement in training times.

Most chapters so far have concentrated on supervised learning. However, in chapter 11 the statistical mechanics framework is extended to the analysis of unsupervised learning scenarios and their dynamics. More specifically, this chapter examines the dynamics of on-line methods aimed at extracting prototypes and principle components from data. The authors consider on-line competitive learning (Winner Takes All and K-means) and Sanger's rule for on-line PCA. A similar set of equations to those used for supervised learning is constructed once the macroscopic variables have been identified, facillitating the study of their dependence on the choice of training parameters.

One of the main defficiencies of the current statistical mechanics framework is that training examples are presumed to be uncorrelated. This restriction exists in most analyses except in certain specific scenarios and limits considerably the usefulness of the theoretical analysis for practical cases where correlations typically emerge either due to the limited training data (which forces sampling with repitition) or due to correlations which exist within the data naturally.

Chapters 12, 13 and 14 tackle training scenarios where correlations within the data exist. In chapter 12, the effect of temporal correlations within the data is handled using the approaches of both stochastic approximation and statistical mechanics for small and large networks respectively. The small network analysis concentrates on a small learning rate expansion where the effect of correlations may be handled straightforwardly. Correlations in the large networks analysis are handled by assuming the distribution for the local fields to be Gaussian, rendering the analysis tractable. Special emphasis is given to the effect of correlations on plateaus in the evolution of the generalization error, which are often characteristic of on-line learning in complex non-linear systems.

The main difficulty of training with fixed example sets is the emerging correlations between parameter updates due to re-sampling, which generally give rise to non-Gaussian local field distributions. The method presented in chapter 13 extends the framework of (Saad and Solla 1995) in both linear and non-linear networks by projecting the evolving macroscopic parameters onto the most significant eigenspaces, obtaining an exact result in the linear case and an approximation in the general non-linear one. The performance of on-line methods is then compared to that of off-line methods in the case of biased and unbiased input distributions and for different types of noise. A different approach, presented in chapter 14, makes use of the dynamical

replica method for closing the equations of motion for a new set of order parameters. This enables one to monitor the evolving non-Gaussian distributions explicitly. The new order parameters include the old set, derived from the infinite training data analysis, as a subset in addition to a new continuous parameter which results from the emerging correlations between updates. The accuracy of results obtained by the method is demonstrated for simple training scenarios.

One method of speeding up training in both on-line and batch training scenarios is by learning with queries, in which case the input distibution is continuously modified to select the most informative examples. These modifications will depend on the current mapping realized by the system, and thus on the current set of parameters, and will improve the network's performance considerably. Chapter 15 deals with the estimation of decision boundaries from stochastic examples with and without queries, investigating the convergence rate in both cases and comparing them to the results obtained in batch learning. Results are also obtained for the fastest feasible convergence rates with and without queries.

An important extension of the Bayesian approach to on-line training is presented in chapter 16, based on approximating the evolving posterior by a multivariate Gaussian distribution. Updating the parameters of this distribution is carried out by on-line methods in response to the sequential presentation of training examples. This elegant and principled approach complements the Bayesian framework of batch learning and may hold a significant practical potential. The analysis shows a similar asymptotic behavior to that obtained by the somewhat less practical variational methods. This approach is investigated further in chapter 17 where it is employed for studying generic feed-forward architectures. The approximation used in the case of continuous weights is shown to have a similar computational complexity to that of Bayesian off-line methods while a different approach, based on a Hebbian approximation, was found to outperform several other on-line approaches, especially in the case of binary weights.

This book is aimed at provideing a fairly comprehensive overview on recent developments in theoretical analysis of online learning methods in neural networks. The chapters were designed to contain sufficient detailed material to enable the non-specialist reader to follow most of it with minimal background reading.

References

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10, 251–276.

Biehl, M. and Schwarze , H. (1995). Learning by online gradient descent. *J. Phys. A*, 28, 643–656.

Bishop, C. M. (1995). *Neural networks for pattern recognition*, Oxford university Press, Oxford, UK.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function, *Math. Control Signals and Systems*, 2, 304–314.

Hertz, J., Krogh, A. and Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation*, Addison Wesley , Redwood City, CA.

Hornik, K. Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359–366.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1, 541–551.

Mace, C.W.H. and Coolen , A.C.C (1998a). Statistical mechanical analysis of the dynamics of learning in perceptrons. *Statistics and Computing*, 8, 55–88.

Orr, G.B. and Leen, T.K. (1997). Using Curvature Information for Fast Stochastic Search. *Advances in Neural Information Processing Systems 9*, edited by Mozer, Jordan and Petsche (Cambridge, MA: MIT Press) p 606–p 612.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, UK.

Saad, D. and Solla, S.A. (1995). Exact solution for online learning in multilayer neural networks. *Phys. Rev. Lett.*, 74, 4337–4340 and Online learning in soft committee machines. *Phys. Rev. E*, 52, 4225–4243.

Sompolinsky, H., Barkai, N., Seung, H.S. (1995), On-line learning of dichotomies: algorithms and learning curves. J-H. Oh, C. Kwon, S. Cho (eds.), *Neural Networks: The Statistical Mechanics Perspective*, 105–130 (Singapore: World Scientific) and Barkai, N., Seung, H.S. Sompolinsky, H. (1995). Local and global convergence of online learning. *Phys. Rev. Lett.*, 75, 1415–1418.

Online Learning and Stochastic Approximations

Léon Bottou

AT&T Labs–Research Red Bank, NJ07701, USA.
leonb@research.att.com

Abstract

The convergence of online learning algorithms is analyzed using the tools of the stochastic approximation theory, and proved under very weak conditions. A general framework for online learning algorithms is first presented. This framework encompasses the most common online learning algorithms in use today, as illustrated by several examples. The stochastic approximation theory then provides general results describing the convergence of all these learning algorithms at once.

1 Introduction

Almost all of the early work on *Learning Systems* focused on online algorithms (Hebb, 1949; Rosenblatt, 1957; Widrow and Hoff, 1960; Amari, 1967; Kohonen, 1982). In these early days, the algorithmic simplicity of online algorithms was a requirement. This is still the case when it comes to handling large, real-life training sets (LeCun et al., 1989; Müller, Gunzinger and Guggenbühl, 1995).

The early *Recursive Adaptive Algorithms* were introduced during the same years (Robbins and Monro, 1951) and very often by the same people (Widrow and Stearns, 1985). First developed in the engineering world, recursive adaptation algorithms have turned into a mathematical discipline, namely *Stochastic Approximations* (Kushner and Clark, 1978; Ljung and Söderström, 1983; Benveniste, Metivier and Priouret, 1990).

Although both domains have enjoyed the spotlights of scientific fashion at different times and for different reasons, they essentially describe the same elementary ideas. Many authors of course have stressed this less-than-fortuitous similarity between learning algorithms and recursive adaptation algorithms (Mendel and Fu, 1970; Tsypkin, 1971).

The present work builds upon this similarity. Online learning algorithms are analyzed using the stochastic approximation tools. Convergence is characterized under very weak conditions: the expected risk must be reasonably well behaved and the learning rates must decrease appropriately.

The main discussion describes a general framework for online learning algorithms, presents a number of examples, and analyzes their dynamical properties. Several comment sections illustrate how these ideas can be generalized and how they relate to other aspects of learning theory. In other words, the main discussion gives answers, while the comments raise questions. Casual readers may skip these comment sections.

2 A Framework for Online Learning Systems

The starting point of a mathematical study of online learning must be a mathematical statement for our subjective understanding of what a learning system is. It is difficult to agree on such a statement, because we are learning systems ourselves and often resent this mathematical reduction of an essential personal experience.

This contribution borrows the framework introduced by the Russian school (Tsypkin, 1971; Vapnik, 1982). This formulation can be used for understanding a significant number of online learning algorithms, as demonstrated by the examples presented in section 3.

2.1 Expected Risk Function

In (Tsypkin, 1971; Tsypkin, 1973), the goal of a learning system consists of finding the minimum of a function $J(w)$ named the *expected risk function.* This function is decomposed as follows:

$$J(w) \triangleq \mathbf{E}_z\, Q(z,w) \triangleq \int Q(z,w)\, dP(z) \tag{2.1}$$

The minimization variable w is meant to represent the part of the learning system which must be adapted as a response to observing events z occurring in the real world. The *loss function* $Q(z,w)$ measures the performance of the learning system with parameter w under the circumstances described by event z. Common mathematical practice suggests to represent both w and z by elements of adequately chosen spaces $\mathcal{W}$ and $\mathcal{Z}$.

The occurrence of the events z is modeled as random independent observations drawn from an unknown probability distribution $dP(z)$ named the *grand truth distribution.* The risk function $J(w)$ is simply the expectation of the loss function $Q(z,w)$ for a fixed value of the parameter w. This risk function $J(w)$ is poorly defined because the grand truth distribution $dP(z)$ is unknown by hypothesis.

Consider for instance a neural network system for optical ZIP code recognition, as described in (LeCun et al., 1989). An observation z is a pair (x,y) composed of a ZIP code image x and its intended interpretation y. Parameters w are the adaptable weights of the neural network. The loss function

$Q(z, w)$ measures the economical cost (in hard currency units) of delivering a letter marked with ZIP code z given the answer produced by the network on image x. This cost is minimal when the network gives the right answer. Otherwise the loss function measures the higher cost of detecting the error and re-routing the letter.

Comments

Probabilities are used in this framework for representing the unknown truth underlying the occurrences of observable events. Using successive observations z_t, the learning system will uncover a part of this truth in the form of parameter values w_t that hopefully decrease the risk functional $J(w_t)$. This use of probabilities is very different from the Bayesian practice, where a probability distribution represents the increasing knowledge of the learning system. Both approaches however can be re-conciliated by defining the parameter space $\mathcal{W}$ as a another space of probability distributions. The analysis then must carefully handle two different probability distributions with very different meanings.

In this framework, every known fact about the real world should be removed from distribution $dP(z)$ by properly redefining the observation space $\mathcal{Z}$ and of the loss function $Q(z, w)$. Consider for instance that a known fraction of the ZIP code images are spoiled by the image capture system. An observation z can be factored as a triple (κ, x, y) composed of an envelope x, its intended ZIP code y, and a binary variable κ indicating whether the ZIP code image is spoiled. The loss function can be redefined as follows:

$$\begin{aligned} J(w) &= \int Q(z, w)\, dP(\kappa, x, y) \\ &= \int \left(\int Q(z, w)\, dP(\kappa|x, y) \right) dP(x, y) \end{aligned}$$

The inner integral in this decomposition is a new loss function $Q'(x, y, w)$ which measures the system performance on redefined observations (x, y). This new loss function accounts for the known deficiencies of the image capture system. This factorization technique reveals a new probability distribution $dP(x, y)$ which is no longer representative of this a priori knowledge.

This technique does not apply to knowledge involving the learning system itself. When we say for instance that an unknown function is smooth, we mean that it pays to bias the learning algorithm towards finding smoother functions. This statement does not describe a property of the grand truth distribution. Its meaning is attached to a particular learning system. It does not suggests a redefinition of the problem. It merely suggests a modification of the learning system, like the introduction of a regularization parameter.

2.2 Gradient Based Learning

The expected risk function (2.1) cannot be minimized directly because the grand truth distribution is unknown. It is however possible to compute an