

Lecture Notes in Statistics

Edited by P. Bickel, P. Diggle, S. Fienberg, K. Krickeberg,
I. Olkin, N. Wermuth, S. Zeger

118

Springer Science+Business Media, LLC

Radford M. Neal

Bayesian Learning for Neural Networks



Springer

Radford M. Neal
Department of Statistics and
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 1A4

ISBN 978-0-387-94724-2 ISBN 978-1-4612-0745-0 (eBook)
DOI 10.1007/978-1-4612-0745-0

CIP data available.
Printed on acid-free paper.

© 1996 Springer Science+Business Media New York
Originally published by Springer-Verlag New York, Inc. in 1996
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher Springer Science+Business Media, LLC, except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Camera ready copy provided by the author.

9 8 7 6 5 4 3

Preface

This book, a revision of my Ph.D. thesis,¹ explores the Bayesian approach to learning flexible statistical models based on what are known as “neural networks”. These models are now commonly used for many applications, but understanding why they (sometimes) work well, and how they can best be employed is still a matter for research. My aim in the work reported here is two-fold — to show that a Bayesian approach to learning these models can yield theoretical insights, and to show also that it can be useful in practice. The strategy for dealing with complexity that I advocate here for neural network models can also be applied to other complex Bayesian models, as can the computational methods that I employ.

In Chapter 1, I introduce the Bayesian framework for learning, the neural network models that will be examined, and the Markov chain Monte Carlo methods on which the implementation is based. This presentation presupposes only that the reader possesses a basic statistical background.

Chapter 1 also introduces the major themes of this book, which involve two fundamental characteristics of Bayesian learning. First, Bayesian learning starts with a prior probability distribution for model parameters, which is supposed to capture our beliefs about the problem derived from background knowledge. Second, Bayesian predictions are not based on a single estimate for the model parameters, but rather are found by integrating the

¹*Bayesian Learning for Neural Networks*, Department of Computer Science, University of Toronto, 1995.

model's predictions with respect to the posterior parameter distribution that we obtain when we update the prior to take account of the data. For neural network models, both these aspects present difficulties — the prior over network parameters has no obvious relation to any prior knowledge we are likely to have, and integration over the posterior distribution is computationally very demanding.

I address the first of these problems in Chapter 2, by defining classes of prior distributions for network parameters that reach sensible limits as the size of the network goes to infinity. In this limit, the properties of these priors can be elucidated. Some priors converge to Gaussian processes, in which functions computed by the network may be smooth, Brownian, or fractionally Brownian. Other priors converge to non-Gaussian stable processes. Interesting effects are obtained by combining priors of both sorts in networks with more than one hidden layer. This work shows that within the Bayesian framework there is no theoretical need to limit the complexity of neural network models. Indeed, limiting complexity is likely to conflict with our prior beliefs, and can therefore be justified only to the extent that it is necessary for computational reasons.

The computational problem of integrating over the posterior distribution is addressed in Chapter 3, using Markov chain Monte Carlo methods. I demonstrate that the hybrid Monte Carlo algorithm, originally developed for applications in quantum chromodynamics, is superior to the methods based on simple random walks that are widely used in statistical applications at present. The hybrid Monte Carlo method makes the use of complex Bayesian network models possible in practice, though the computation time required can still be substantial.

In Chapter 4, I use a hybrid Monte Carlo implementation to test the performance of Bayesian neural network models on several synthetic and real data sets. Good results are obtained on small data sets when large networks are used in conjunction with priors designed to reach limits as network size increases, confirming that with Bayesian learning one need not restrict the complexity of the network based on the size of the data set. A Bayesian approach is also found to be effective in automatically determining the relevance of inputs.

Finally, in Chapter 5, I draw some conclusions from this work, and briefly discuss related work by myself and others since the completion of the original thesis.

Readers interested in pursuing research in this area may obtain free software implementing the methods, as described in Appendix B. One should note, however, that this software is not intended for use in routine data analysis. The software is also designed only for use on Unix systems.

Of the many people who have contributed to this work, I would like first of all to thank my thesis advisor, Geoffrey Hinton. His enthusiasm for understanding learning, his openness to new ideas, and his ability to provide insightful criticism have made working with him a joy. I am also fortunate to have been part of the research group he has led, and of the wider AI group at the University of Toronto. I would particularly like to thank fellow students Richard Mann, Carl Rasmussen, and Chris Williams for their helpful comments on this work and its precursors. My thanks also go to the present and former members of my Ph.D. committee, Mike Evans, Scott Graham, Rudy Mathon, Demetri Terzopoulos, and Rob Tibshirani.

I am especially pleased to thank David MacKay, whose work on Bayesian learning and its application to neural network models has been an inspiration to me. He has also contributed much to this work through many conversations and e-mail exchanges, which have ranged from the philosophy of Bayesian inference to detailed comments on presentation. I have benefited from discussions with other researchers as well, in particular, Wray Buntine, Brian Ripley, Hans Henrik Thodberg, and David Wolpert.

This work was funded by the Natural Sciences and Engineering Research Council of Canada and by the Information Technology Research Centre. For part of my studies, I was supported by an Ontario Government Scholarship.

Contents

Preface	iii
1 Introduction	1
1.1 Bayesian and frequentist views of learning	3
1.1.1 Models and likelihood	3
1.1.2 Bayesian learning and prediction	4
1.1.3 Hierarchical models	6
1.1.4 Learning complex models	7
1.2 Bayesian neural networks	10
1.2.1 Multilayer perceptron networks	10
1.2.2 Selecting a network model and prior	14
1.2.3 Automatic Relevance Determination (ARD) models .	15
1.2.4 An illustration of Bayesian learning for a neural net .	17
1.2.5 Implementations based on Gaussian approximations .	19
1.3 Markov chain Monte Carlo methods	22
1.3.1 Monte Carlo integration using Markov chains	23
1.3.2 Gibbs sampling	25
1.3.3 The Metropolis algorithm	26
1.4 Outline of the remainder of the book	28

2	Priors for Infinite Networks	29
2.1	Priors converging to Gaussian processes	31
2.1.1	Limits for Gaussian and other priors with finite variance	32
2.1.2	Priors that lead to smooth and Brownian functions	34
2.1.3	Covariance functions of Gaussian priors	37
2.1.4	Fractional Brownian priors	39
2.1.5	Networks with more than one input	40
2.2	Priors converging to non-Gaussian stable processes	43
2.2.1	Limits for priors with infinite variance	43
2.2.2	Properties of non-Gaussian stable priors	45
2.3	Priors for nets with more than one hidden layer	48
2.4	Hierarchical models	51
3	Monte Carlo Implementation	55
3.1	The hybrid Monte Carlo algorithm	56
3.1.1	Formulating the problem in terms of energy	57
3.1.2	The stochastic dynamics method	58
3.1.3	Hybrid Monte Carlo	60
3.2	An implementation of Bayesian neural network learning	63
3.2.1	Gibbs sampling for hyperparameters	66
3.2.2	Hybrid Monte Carlo for network parameters	68
3.2.3	Verifying correctness	73
3.3	A demonstration of the hybrid Monte Carlo implementation	74
3.3.1	The robot arm problem	75
3.3.2	Sampling using the hybrid Monte Carlo method	76
3.3.3	Making predictions	84
3.3.4	Computation time required	87
3.4	Comparison of hybrid Monte Carlo with other methods	88
3.5	Variants of hybrid Monte Carlo	91
3.5.1	Computation of trajectories using partial gradients	91
3.5.2	The windowed hybrid Monte Carlo algorithm	95
3.5.3	Hybrid Monte Carlo with persistent momentum	97
4	Evaluation of Neural Network Models	99
4.1	Network architectures, priors, and training procedures	100
4.2	Tests of the behaviour of large networks	102
4.2.1	Theoretical expectations concerning large networks	103

4.2.2	Tests of large networks on the robot arm problem . . .	104
4.3	Tests of Automatic Relevance Determination	113
4.3.1	Procedures for evaluating ARD models	114
4.3.2	Tests of ARD on the noisy LED display problem . . .	116
4.3.3	Tests of ARD on the robot arm problem	122
4.4	Tests of Bayesian models on real data sets	126
4.4.1	Methodology for comparing learning procedures . . .	126
4.4.2	Tests on the Boston housing data	127
4.4.3	Tests on the forensic glass data	136
5	Conclusions and Further Work	145
5.1	Priors for complex models	145
5.2	Hierarchical Models — ARD and beyond	147
5.3	Implementation using hybrid Monte Carlo	150
5.4	Evaluating performance on realistic problems	152
A	Details of the Implementation	153
A.1	Specifications	153
A.1.1	Network architecture	153
A.1.2	Data models	155
A.1.3	Prior distributions for parameters and hyperparameters	156
A.1.4	Scaling of priors	159
A.2	Conditional distributions for hyperparameters	159
A.2.1	Lowest-level conditional distributions	160
A.2.2	Higher-level conditional distributions	160
A.3	Calculation of derivatives	161
A.3.1	Derivatives of the log prior density	162
A.3.2	Log likelihood derivatives with respect to unit values .	162
A.3.3	Log likelihood derivatives with respect to parameters	163
A.4	Heuristic choice of stepsizes	164
A.5	Rejection sampling from the prior	166
B	Obtaining the software	169
	Bibliography	171
	Index	177

List of Figures

1.1	A multilayer perceptron network	11
1.2	An illustration of Bayesian inference for a neural network	18
2.1	Convergence of network priors to a Gaussian process	33
2.2	Functions drawn from Gaussian priors for networks of step-function units	35
2.3	Functions drawn from Gaussian priors for networks of tanh hidden units	37
2.4	Functions drawn from fractional Brownian priors	41
2.5	Behaviour of $D(x-s/2, x+s/2)$ for different sorts of functions	41
2.6	Functions of two inputs drawn from Gaussian priors	42
2.7	Functions drawn from Cauchy priors	46
2.8	Functions of two inputs drawn from non-Gaussian priors	47
2.9	Functions computed from networks with several hidden layers .	49
2.10	Functions drawn from a combined Gaussian and non-Gaussian prior	50
3.1	Sampling using the Langevin and hybrid Monte Carlo methods	63
3.2	Progress of hybrid Monte Carlo runs in the initial phase	78

3.3	Error in energy for trajectories computed with different step-sizes	80
3.4	Degree of correlation along a trajectory	80
3.5	Progress of hybrid Monte Carlo runs in the sampling phase . .	82
3.6	Autocorrelations for different trajectory lengths	83
3.7	Predictive distribution from Monte Carlo data	85
3.8	Average test error on the robot arm problem with different im- plementations	86
3.9	Progress of simple Metropolis and Langevin methods in the sampling phase	89
3.10	Error in energy for trajectories computed using partial gradients	94
3.11	Difference in free energy for windowed trajectories	94
4.1	Computational details for experiments on networks of varying size	107
4.2	Results on the robot arm problem with networks of varying size	109
4.3	Predictive distributions obtained using networks of varying size	110
4.4	Results of maximum likelihood learning with networks of vary- ing size	112
4.5	Digit patterns for the noisy LED display problem	116
4.6	Results on the noisy LED display problem	120
4.7	Relevant and irrelevant input weight magnitudes for the LED display problem	121
4.8	Input weight magnitudes for the robot arm problem with and without ARD	124
4.9	Descriptions of inputs for the Boston housing problem	128
4.10	Results of preliminary tests on the Boston housing data . . .	130
4.11	Cross-validation assessments on the Boston housing data . . .	134
4.12	Networks and priors tested on the forensic glass data	139
4.13	Results on the forensic glass data	140
4.14	Effect of vague priors in the forensic glass problem	142
5.1	A hierarchical network model capable of finding additive structure	149