

Encyclopedia of Artificial Intelligence

Juan Ramón Rabuñal Dopico
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

Alejandro Pazos Sierra
University of A Coruña, Spain

Information Science
REFERENCE

INFORMATION SCI

Hershey • New York

Director of Editorial Content: Kristin Klinger
Managing Development Editor: Kristin Roth
Development Editorial Assistant: Julia Mosemann, Rebecca Beistline
Senior Managing Editor: Jennifer Neidig
Managing Editor: Jamie Snavelly
Assistant Managing Editor: Carole Coulson
Typesetter: Jennifer Neidig, Amanda Appicello, Cindy Consonery
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: <http://www.igi-global.com/reference>

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: <http://www.eurospanbookstore.com>

Copyright © 2009 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher.

Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Encyclopedia of artificial intelligence / Juan Ramon Rabunal Dopico, Julian Dorado de la Calle, and Alejandro Pazos Sierra, editors.
p. cm.

Includes bibliographical references and index.

Summary: "This book is a comprehensive and in-depth reference to the most recent developments in the field covering theoretical developments, techniques, technologies, among others"--Provided by publisher.

ISBN 978-1-59904-849-9 (hardcover) -- ISBN 978-1-59904-850-5 (ebook)

I. Artificial intelligence--Encyclopedias. I. Rabunal, Juan Ramon, 1973- II. Dorado, Julian, 1970- III. Pazos Sierra, Alejandro.

Q334.2.E63 2008

006.303--dc22

2008027245

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this encyclopedia set is new, previously-unpublished material. The views expressed in this encyclopedia set are those of the authors, but not necessarily of the publisher.

If a library purchased a print copy of this publication, please go to <http://www.igi-global.com/agreement> for information on activating the library's complimentary electronic access to this publication.

Stochastic Approximation Monte Carlo for MLP Learning

Faming Liang

Texas A&M University, USA

INTRODUCTION

Over the past several decades, multilayer perceptrons (MLPs) have achieved increased popularity among scientists, engineers, and other professionals as tools for knowledge representation. Unfortunately, there is no a universal architecture which is suitable for all problems. Even with the correct architecture, frustrating problems of connection weights training still remain due to the rugged nature of the energy landscape of MLPs. The energy function often refers to the sum-of-square error function for conventional MLPs and the negative log-posterior density function for Bayesian MLPs.

This article presents a Monte Carlo method that can be used for MLP learning. The main focus is on how to apply the method to train connection weights for MLPs. How to apply the method to choose the optimal architecture and to make predictions for future values will also be discussed, but within the Bayesian framework.

BACKGROUND

As known by many researchers, the energy landscape of an MLP is often rugged. The gradient-based training algorithms, such as back-propagation (Rumelhart et al., 1986), conjugate gradient, Newton's method, and the BFGS algorithm (Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970), tend to converge to a local minimum near the starting point, rendering the training data learned insufficiently. To reduce the chance of converging to local minima, a number of variants of these algorithms have been proposed based on the idea of perturbation (von Lehmen et al., 1988, Tang et al., 2003 and references therein). In practice, the effects of these perturbations are usually limited, which only delay the learning process converging to local minima a reasonable number of iterations (Ingman & Merlis, 1991).

To avoid the local-trap problem, simulated annealing (SA) (Kirkpatrick et al., 1983) has been employed by some authors to train neural networks. Amato et al. (1991) and Owen & Abunawass (1993) show that for complex learning tasks, SA has a better chance to converge to a global minimum than have the gradient-based algorithms. Geman & Geman (1984) show that the global minimum can be reached by SA with probability 1 if the temperature decreases at a logarithmic rate of $O(1/\log t)$, where t denotes the number of iterations. In practice, however, no one can afford to have such a slow cooling schedule. Most frequently, people use a linearly or geometrically decreasing cooling schedule, which can no longer guarantee the global energy minimum to be reached (Holley, et al., 1989).

Other stochastic algorithms that have been used in MLP training include the genetic algorithm (Goldberg, 1989) and Markov chain Monte Carlo (MCMC). Although the genetic algorithm works well for some problems, see, e.g., van Rooij et al. (1996), there is no theory to support its convergence to global minima. MCMC algorithms are mainly used for Bayesian MLPs (MacKay, 1992a, Neal, 1996, Muller & Insua, 1998, de Freitas et al., 2000, Liang, 2003, 2005a, 2005b), which will be discussed later.

MAIN FOCUS OF THE CHAPTER

This article presents how the stochastic approximation Monte Carlo (SAMC) (Liang et al., 2007) algorithm can be used for MLP learning, including training, prediction and architecture selection.

A Brief Review for the SAMC Algorithm

Suppose that we are working with the Boltzmann distribution,

$$p(x) = \frac{1}{Z} e^{-U(x)/\tau}, \quad x \in \Omega, \quad (1)$$

where Z is the normalizing constant, $U(x)$ is the energy function, τ is the temperature, and Ω is the sample space. Without loss of generality, we assume that Ω is compact. For MLPs, x denotes the vector of connection weights, and Ω can be restricted to a hyper-rectangle $[-B_\Omega, B_\Omega]^{\dim(\Omega)}$, where B_Ω is a large number such that Ω includes at least a global minimum of $U(x)$. Furthermore, we assume that the sample space can be partitioned according to the energy function into m disjoint subregions: $E_1 = \{x: U(x) \leq u_1\}$, $E_2 = \{x: u_1 < U(x) \leq u_2\}$, ..., $E_{m-1} = \{x: u_{m-2} < U(x) \leq u_{m-1}\}$, and $E_m = \{x: U(x) > u_{m-1}\}$, where u_1, \dots, u_{m-1} are pre-specified real numbers. SAMC seeks to draw samples from each subregion with a pre-specified frequency. If this goal can be achieved, then the local-trap problem can be avoided successfully. Let x_{t+1} denote a sample simulated from the distribution

$$p_{\theta_t}(x) \propto \sum_{i=1}^m \frac{\Psi(x)}{e^{\theta_i}} I(x \in E_i) \quad (2)$$

using the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953, Hastings, 1970), where $\Psi(x) = e^{-U(x)/\tau}$ and $\theta_t = (\theta_{t1}, \dots, \theta_{tm})$ is an m -vector in a space Θ . For simplicity, we assume that Θ is compact, e.g., $\Theta = [-B_\Theta, B_\Theta]^{\dim(\Theta)}$ with B_Θ being a large number. Since adding to or subtracting from θ_t a constant will not change $p_{\theta_t}(x)$, θ_t can be kept in the compact set in simulations by adjusting with an additive constant. Let the proposal distribution, $q(x, y)$, of the MH moves satisfy the minorisation condition (Mengersen & Tweedie, 1996), i.e.,

$$\sup_{\theta \in \Omega} \sup_{x, y \in \Omega} \frac{p_\theta(y)}{q(x, y)} < \infty \quad (3)$$

Since Ω is compact, a sufficient design for the minorisation condition is to choose $q(x, y)$ as a global proposal distribution. A proposal distribution is said global if $q(x, y) > 0$ for all $x, y \in \Omega$. For MLPs, $q(x, y)$ can be chosen as a random walk Gaussian proposal, $y \sim N(x, \sigma^2 I)$, where I is an identity matrix and σ^2 is calibrated such that the MH moves have a desired acceptance rate. As discussed later, restricting the proposal distribution to be global ensures the convergence of the annealing SAMC algorithm to the global energy minima.

Let $\{\gamma_t\}$ be a positive non-decreasing sequence satisfying the conditions:

- i. $\sum_{t=0}^{\infty} \gamma_t = \infty$,
- ii. $\sum_{t=0}^{\infty} \gamma_t^\delta < \infty$

for some $\delta \in (1, 2)$. For example, one can set

$$\gamma_t = \left(\frac{t_0}{\max(t_0, t)} \right)^\eta \quad (4)$$

for some values of $t_0 > 1$ and

$$\eta \in \left(\frac{1}{2}, 1 \right)$$

A large value of t_0 will allow the sampler to reach all subregions very quickly, even in the presence of multiple local minima. Let $\pi = (\pi_1, \dots, \pi_m)$ be an m -vector with $0 < \pi_i < 1$ and

$$\sum_{i=1}^m \pi_i = 1$$

which defines a desired sampling frequency distribution on the subregions. With the above notations, an iteration of SAMC can be described as follows.

SAMC Algorithm

- a. Generate $x_{t+1} \sim K_{\theta_t}(x_t, \cdot)$ with a single MH step:
 1. Generate y according to the proposal distribution $q(x_t, y)$.
 2. Calculate the ratio

$$r = e^{\theta_{J(x_t)} - \theta_{J(y)}} \frac{\Psi(y) q(y, x_t)}{\Psi(x_t) q(x_t, y)},$$

where $J(x)$ denote the index of the subregion that the sample x belongs to.

- 3. Accept the proposal with probability $\min(1, r)$. If it is accepted, set $x_{t+1} = y$; otherwise, set $x_{t+1} = x_t$.

- b. Set $\theta^* = \theta_t + \gamma_t (e_{t+1} - \pi)$, where γ_t is called the gain factor, $e_{t+1} = (e_{t+1,1}, \dots, e_{t+1,m})$, and $e_{t+1,i} = 1$ if $x_{t+1} \in E_i$ and 0 otherwise.
- c. If $\theta^* \in \Theta$, set $\theta_{t+1} = \theta^*$; otherwise, set $\theta_{t+1} = \theta^* + c^*$, where c^* is a constant vector and is chosen such that $\theta^* + c^* \in \Theta$. The existence of c^* is obvious, since B_Θ has been set to a large number and it is reasonable to assume that $\max_{i=1}^m \theta_i^* - \min_{i=1}^m \theta_i^* \ll B_\Theta$ holds at each iteration.

A remarkable feature of SAMC is its self-adjusting mechanism. If a proposal is rejected, the weight of the subregion that the current sample belongs to will be adjusted to a larger value, and thus a proposal of jumping out from the current subregion will be less likely rejected in the next iteration. This mechanism effectively prevents the system from getting trapped in local minima. This is very important for MLP training as its energy landscape is often rugged.

SAMC falls into the category of stochastic approximation algorithms (Robbins & Monro, 1951, Andrieu et al., 2005 and references therein). The convergence of SAMC can be extended from a theorem presented in Liang et al. (2007). Under mild conditions and as $t \rightarrow \infty$,

$$\theta_{ii} \rightarrow \begin{cases} C + \log \left(\int_{E_i} \psi(x) dx \right) - \log(\pi_i + \zeta), & E_i \neq \emptyset, \\ -\infty, & E_i = \emptyset, \end{cases} \quad (5)$$

where

$$\zeta = \sum_{j \in \{i: E_i = \emptyset\}} \pi_j / (m - m_0)$$

and $m_0 = \#\{i : E_i = \emptyset\}$ is the number of empty subregions, and C is an arbitrary constant. A subregion E_i is said to be empty if

$$\int_{E_i} \psi(x) dx = 0.$$

In SAMC, the sample space partition can be made blindly by simply specifying some values u_1, \dots, u_{m-1} . This may result in some empty subregions. The constant C can be determined by imposing a constraint on θ_t , say,

$$\sum_{i=1}^m e^{\theta_i}$$

is equal to a known number. In addition, Liang (2007) shows that θ_t can converge in the form L_2 at a rate of $O(1/t)$. Let $\pi_{ii} = P(x_t \in E_i)$ be the probability of sampling from the subregion E_i at iteration t . Equation implies that as $t \rightarrow \infty$, π_{ii} will converge to $\pi_i + \zeta$ if $E_i \neq \emptyset$ and 0 otherwise. This further implies that as the number of iterations goes to infinity, SAMC can approximately draw samples from each of the subregions with a pre-specified probability. With an appropriate specification of π , sampling can be biased to the low energy regions to increase the chance of finding the global minimum.

Annealing SAMC for MLP Learning

In theory, SAMC is able to find the global energy minima if the run is long enough. However, due to the broadness of the sample space, the process may be slow even when sampling is biased to low energy subregions. To accelerate the search process, one can iteratively shrink the sample space in simulations. As argued below, this modification preserves the theoretical property of SAMC when a global proposal distribution is used.

Suppose that the subregions E_1, \dots, E_m have been arranged in ascending order by energy; that is, if $i < j$ then $U(x) < U(y)$ for any $x \in E_i$ and $y \in E_j$. Let $\kappa(u)$ denote the index of the subregion that a sample x with energy u belongs to. Let Ω_t denote the sample space at iteration t . Annealing SAMC, which will be abbreviated as ASAMC hereafter, starts with

$$\Omega_1 = \bigcup_{i=1}^m E_i,$$

and then iteratively sets

$$\Omega_t = \bigcup_{i=1}^{\kappa(U_{\min}^t + \Delta)} E_i \quad (6)$$

where U_{\min}^t is the minimum energy value obtained by iteration t , $\Delta > 0$ is a user specified parameter. The sample space Ω_t shrinks iteration by iteration. In this sense, the modified algorithm is called ASAMC.

Since the proposal distribution is global, the convergence property of SAMC still holds for ASAMC on the limiting space $\Omega_\infty = \lim_{t \rightarrow \infty} \Omega_t$, although Ω_∞ may contain some separated regions. The existence of Ω_∞ is true due to the monotonicity of the sequence $\Omega_1 \supseteq$

$\Omega_2 \supseteq \dots$. It follows from Scheffe's theorem (Scheffe, 1947) that as $t \rightarrow \infty$, x_t will converge in distribution to a random variable with density

$$p_\theta(x) \propto \sum_{i=1}^{K(u_{\min} + \Delta)} \frac{(\pi_i + \zeta) \Psi(x)}{\int_{E_i} \Psi(x) dx} I(x \in E_i), \quad (7)$$

where u_{\min} denotes the global minimum of the energy function $U(x)$. Again, as in SAMC, the convergence can be attained in the L_2 form at a rate of $O(1/t)$. If we let Δ go to zero, then the ASAMC samples will converge in distribution to the global minima of $U(x)$.

For an effective implementation of ASAMC, several issues need to be considered.

Sample space partitioning. Since within the same subregion, ASAMC is reduced to sampling from the unnormalized density $\Psi(x)$, we suggest that the maximum energy difference in each subregion should be bounded by a reasonable number, say, 2τ , to ensure that the local Metropolis-Hastings moves within the same subregion have a reasonable acceptance rate.

Choice of Δ . The performance of ASAMC depends on the value of Δ to some extent. If Δ is too large, ASAMC may take a long time to locate the global minimum due to the broadness of the sample space. If Δ is too small, ASAMC may also take a long time to locate the global minimum. In this case, the sample space may contain only a few separated regions, and the most proposed transitions will be rejected. In our experience, a value of Δ between 5 and 10 works well for most MLP problems.

Desired sampling distribution. The choice of π is not critical to the efficiency of ASAMC, as in which the sample space has been shrunk with iterations. On the contrary, in SAMC, π should be chosen carefully to bias sampling to low energy regions to improve ergodicity of the simulation.

Gain factor. To estimate the integrals

$$\int_{E_1} \Psi(x) dx, \dots, \int_{E_m} \Psi(x) dx$$

accurately, γ_t should be very close to 0 at the end of simulations. Otherwise, the resulting estimates may have a large variation. The decreasing speed of γ_t can be controlled by t_0 and η . In practice, we often fix $\eta = 1$ and vary the value of t_0 according to the complexity of the problem. The more complex the problem is, the larger value of t_0 one should choose.

Convergence diagnostic. A formal diagnostic for the convergence of ASAMC should base on multiple runs. A rough diagnostic for a single run can be done by comparing the observed sampling frequencies and the desired sampling frequencies of different subregions. If they match with each other very well, we may regard the run converged. Otherwise, one may re-run the algorithm with a larger number of iterations or a larger value of t_0 .

ASAMC has been compared in Liang (2007) with simulated annealing, SAMC, and the BFGS algorithm on a number of examples, including the famous N-parity and two-spiral problems. The numerical results for the two-spiral problem are re-presented in Table 1 and

Table 1. Comparison of ASAMC, SAMC, SA and BFGS for the two-spiral problem. Notations: let z_i denote the minimum energy value obtained in the i th run. "Mean" = $\sum_{i=1}^{20} z_i / 20$, "SD" is the standard deviation of "mean", "Minimum" = $\min_{i=1}^{20} z_i$, "Maximum" = $\max_{i=1}^{20} z_i$, "Proportion" = $\#\{i : z_i \leq 0.2\}$, "Iteration" is the average number of iterations performed in each run, and "Time" is the average CPU time cost by each run.

Algorithm	Mean	SD	Minimum	Maximum	Proportion	Iteration(10^6)	Time
ASAMC	0.620	0.191	0.187	3.23	15	7.1	94m
SAMC	2.727	0.208	1.092	4.09	0	10.0	132m
SA-1	17.845	0.706	9.020	22.06	0	10.0	123m
SA-2	6.433	0.450	3.030	11.02	0	10.0	123m
BFGS	15.500	0.899	10.00	24.00	0	---	3s

Figure 1. Classification maps learned for the two-spiral problem by ASAMC with a MLP of 30 hidden units. The black and white points show the training data for the two different spirals, respectively. (a) Classification map learned in a run. (b) Classification map averaged over 20 run.

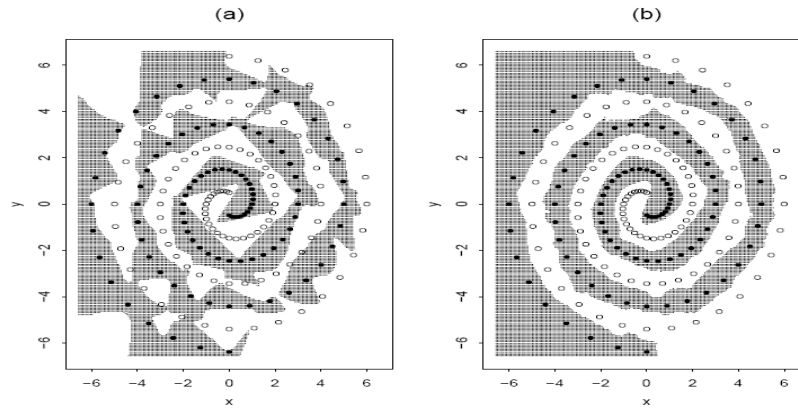


Figure 1. Refer to Liang (2007) for the settings of the respective algorithms. The results for the other examples are similar. In summary, ASAMC outperforms the other algorithms in both training and test errors. Like other stochastic algorithms, ASAMC requires longer training time than do the gradient-based algorithms. It provides, however, an efficient approach to train MLPs for which the energy landscape is rugged.

Bayesian MLP Learning

SAMC can also be used for training Bayesian MLPs. Let $\Psi(x)$ denote the posterior density of a MLP (up to a normalizing constant), and $\hat{g}_i = \lim_{t \rightarrow \infty} e^{\theta_i}$. Thus, the following density

$$\widehat{p}(x) \propto \sum_{i=1}^m \frac{\Psi(x)}{\hat{g}_i} I(x \in E_i) \tag{8}$$

can work as a trial density for sampling from $\Psi(x)$. As a trial density, it possesses two nice properties. First, the importance weight is bounded above by $\max_i \hat{g}_i$, assuming that \hat{g}_i has been normalized by an additional constraint, e.g.,

$$\sum_{i=1}^m \hat{g}_i$$

is a known constant. Second, sampling from $\widehat{p}(x)$ will lead to a random walk in the space of nonempty sub-

regions if we regard each subregion as a point. Hence, the whole sample space can be well explored.

Suppose that important samples $(x_1, w_1), \dots, (x_n, w_n)$ have been drawn from using a MCMC sampler, where w_i denotes the importance weight of x_i . Let $f(z|x)$ denote the output of the MLP with input z . For a new input z_0 , the Bayesian point prediction is then

$$\widehat{f}(z_0) = \frac{\sum_{i=1}^n w_i f(z_0 | x_i)}{\sum_{i=1}^n w_i} \tag{9}$$

Evidence Evaluation for Bayesian MLPs

In addition to MLP learning, SAMC also provides a convenient way for evaluating evidence of Bayesian MLPs. As pointed out by MacKay (1992b), the Bayesian evidence can be used as a guideline of architecture selection for Bayesian MLPs. Let $f(D|x)$ denote the likelihood function of a given MLP model, and let $l(x)$ denote the prior density imposed on x . As before, we suppose that Ω has been restricted to a compact set. Define the function

$$\Psi(x, k) = \begin{cases} f(D|x)l(x), & k=1 \\ 1/|\Omega|, & k=0 \end{cases} \tag{10}$$

on the product space $\Omega \times \{0, 1\}$, where $|\Omega|$ denotes the hypervolume of the space Ω . Partition the product space as follows: $E_0 = \{(x, k) : k=0, x \in \Omega\}$, $E_1 = \{(x, k) : k=$

$1, U(x) \leq u_1\}, \dots, E_m = \{(x, k) : k = 1, U(x) > u_{m-1}\}$. If SAMC is run with this partition, the evidence of the MLP can then be estimated by

$$\widehat{EV} = \frac{\sum_{i=1}^m (\pi_i + \zeta) \widehat{g}_i}{(\pi_0 + \zeta) \widehat{g}_0} g_0, \quad (11)$$

where

$$g_0 = \int_{E_0} \Psi(x, 0) dx,$$

$$\widehat{g}_i = \lim_{t \rightarrow \infty} e^{\theta_t},$$

and $0 < \pi_0 < 1$. We note that $\Psi(x, 0)$ can be any non-negative function with g_0 being analytically available.

FUTURE TRENDS

In the future, we need to carry out a series of comparisons to assess the ability of SAMC in different aspects. For example, we need to compare SAMC with advanced MCMC samplers, such as parallel tempering (Geyer, 1991) and evolutionary Monte Carlo (Liang & Wong, 2001), to assess its ability in Bayesian prediction; and to compare SAMC with the Gaussian approximation method (MacKay, 1992b) to assess its ability in evidence evaluation.

CONCLUSION

This article proposes an innovative method for MLP training, prediction, and architecture selection. The strength of SAMC comes from its self-adjusting mechanism, which enables it to overcome the local-trap problems. Like simulated annealing and genetic algorithms, SAMC avoids the requirement for the gradient information of the objective function. Hence, it can be used as a general optimization, simulation, and integration tool in many other problems, such as combinatorial optimization, model selection, and statistical simulations.

REFERENCES

- Amato, S., Apolloni, B., Caporali, G., Madiesani, U., & Zanaboni, A. (1991). Simulated annealing approach in back-propagation. *Neurocomputing*, 3(5-6), 207-220.
- Andrieu, C., Moulines, E., & Priouret, P. (2005). Stability of Stochastic Approximation Under Verifiable Conditions. *SIAM J. Control and Optimization*, 44(1), 283-312.
- Broyden, C.G. (1970). The convergence of a class of double rank minimization algorithms. *J. Inst. Maths. Applns*, 6(3), 76-90.
- de Freitas, N., Niranjan, M., Gee, A.H., & Doucet, A. (2000). Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4), 955-993.
- Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer J.*, 13(3), 317-322.
- Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 721-741.
- Geyer, C.J., (1991). Markov chain Monte Carlo maximum likelihood, in *Computing Science and Statistics: Proceedings of the 23rd Symposium on the interface* (E.M. Keramigas, ed.), pp.156-163, Fairfax: Interface Foundation.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, & Machine learning*, Addison Wesley.
- Goldfarb, D. (1970). A family of variable metric methods derived by variational means. *Maths. Comp.*, 24(109), 23-26.
- Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chain and Their Applications. *Biometrika*, 57(1), 97-109.
- Holley, R.A., Kusuoka, S. & Stroock, D. (1989). Asymptotic of the spectral gap with applications to the theory of simulated annealing. *Journal of Functional Analysis*, 83(2), 333-347.
- Ingman, D. & Merlis, Y. (1991). Local minimization escape using thermodynamic properties of neural networks. *Neural Networks*, 4(3), 395-404.

- Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Liang, F. (2003). An effective Bayesian neural network classifier with a comparison study to support vector machine. *Neural Computation*, 15(8), 1959-1989.
- Liang, F. (2005a). Bayesian neural networks for nonlinear time series forecasting. *Statistics and Computing*, 15(1), 13-29.
- Liang, F. (2005b). Evidence evaluation for Bayesian neural networks using contour Monte Carlo. *Neural Computation*, 17(6), 1385-1410.
- Liang, F. (2007). Annealing stochastic approximation Monte Carlo algorithm for neural network training. *Machine Learning*, 68(3) 201-233.
- Liang, F., Liu, C. & Carroll, R.J. (2007). Stochastic Approximation in Monte Carlo Computation. *Journal of the American Statistical Association*, 102(477), 305-320.
- Liang, F. and Wong, W.H. (2001). Real parameter evolutionary Monte Carlo with applications in Bayesian mixture models. *Journal of the American Statistical Association*, 96(454), 653-666.
- MacKay, D.J.C. (1992a). A practical Bayesian framework for backprop networks. *Neural Computation*, 4(3), 448-472.
- MacKay, D.J.C. (1992b). The evidence framework applied to classification problems. *Neural Computation*, 4(5), 720-736.
- Mengersen, K.L. & Tweedie, R.L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1), 101-121.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6), 1087-1091.
- Muller, P. & Insua, D.R. (1998). Issues in Bayesian analysis of neural network models. *Neural Computation*, 10(3), 749-770.
- Neal, R.M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.
- Owen, C.B. & Abunawass, A.M. (1993). Applications of simulated annealing to the back-propagation model improves convergence, *SPIE Proceedings*, 1966, 269-276.
- Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3), 400-407.
- Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by back-propagating errors. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1* (D.E. Rumelhart and J.L. McClelland, ed.), pp.318-362, Cambridge, MA: MIT Press.
- Scheffe, H. (1947). A useful convergence theorem for probability distributions. *Ann. Math. Statist.*, 18(3), 434-438.
- Shanno, D.F. (1970). Conditioning of quasi-Newton methods for function minimization. *Maths. Comp.*, 24(111), 647-656.
- Tang, Z., Wang, X. Tamura, H., & Ishii, M. (2003). An algorithm of supervised learning for multilayer neural networks. *Neural Computation*, 15(5), 1125-1142.
- van Rooij, A.J.F., Jain, L.C., & Johnson, R.P. (1996). *Neural Network Training Using Genetic Algorithms*. Singapore: World Scientific.
- von Lehmen, A., Paek, E.G., Liao, P.F., Marrakchi, A., & Patel, J.S. (1988). Factors influencing learning by back-propagation. In *Proceedings of IEEE International Conference on Neural Networks*, pp.335-341, New York: IEEE Press.

KEY TERMS

Genetic Algorithm: A search heuristic used in computing to find true or approximate solutions to global optimization problems.

Markov Chain Monte Carlo (MCMC): A class of algorithms for sampling from probability distributions by simulating a Markov chain that has the desired distribution as its stationary distribution. The state of the Markov chain after a large number of steps is then used as a sample from the desired distribution.

Metropolis-Hastings Algorithm: A popular MCMC algorithm with the acceptance probability $\{1, [f(y)q(y,x)]/[f(x)q(x,y)]\}$ for a new state y given the current state x , where $f(\cdot)$ is the target distribution and $q(\cdot, \cdot)$ is the proposal distribution.

Model Evidence: The log-marginal likelihood of the data obtained by integrating out the parameters over the space of models. Its value expresses the preference shown by the data for different models.

Multiple Layer Perceptron (MLP): An important class of neural networks, which consists of a set of source nodes that constitute the input layer, one or more layers of computational nodes, and an output layer of computational nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis.

Simulated Annealing: A generic probabilistic meta-algorithm used to find true or approximate solutions to global optimization problems.

Stochastic Approximation Algorithm: A probabilistic meta-algorithm suggested by Robbins and Monro (1951) for solutions of regression equations.

S