Boltzmann Learning of Parameters in Bayes Visual Reconstruction*

Lars Kai Hansen
CONNECT, Electronics Institute, build. 349
Technical University of Denmark,
DK-2800 Lyngby, Denmark

Abstract

The Boltzmann Machine learning rule is applied to the parameter estimation problem for Bayesian visual reconstruction models. The learning rule can be used for models with *hidden* units. As an illustration we apply the learning rule to the socalled *Weak String Model*.

1 Introduction

Reconstruction of imagery that has been degraded through a non-ideal imaging system is a fundamental aim of early vision processing. Typical degradations encompas defocusing (blur) and noise. The *Bayesian* approach is a very successful device for visual reconstruction [7, 13, 19], a particular attraction is that it leads to algorithms that map well onto networks of simple processing elements ie. *artificial neural networks*. With the advent of low-cost massively parallel hardware [14], this virtue may end up being decisive for (near) future real life applications. However, while the Bayesian approach allows formulation of collective models that solve complex early vision tasks, general and flexible tools for parameter estimation are lacking. Direct maximum-likelihood estimation is hampered by the difficulty of obtaining analytical expressions for derivatives of normalization constants [1]. Besag has introduced two methods based on approximate maximum-likelihood estimation, the *coding* method [1], and the *pseudo-likelihood* method [2, 15]. Both methods pose difficulties if the image model includes non-observable attributes ie. *hidden* units.

In this contribution we discuss the use of the Boltzmann Machine learning rule for parameter estimation. The learning rule may be applied to general situations with hidden units without complication. In the present formulation the method is based on *supervised learning*, ie. a learning database comprising both input and output examples, *unsupervised* learning (learning from input only) will be devoted a separate presentation [9].

*Proceedings of the 1. Danish Conference on Pattern Recognition and Image Analysis, Dept. of Computer science, University of Copenhagen, (Ed. S.I. Olsen), 1992.

Geman and Geman [7] introduced Metropolis sampling from Gibbs distributions as a simulation tool for visual reconstruction and showed that a *Simulated Annealing* strategy could improve the quality of the sampling process see also [4]. The sampling process implements a stochastic neural network with symmetric connections. Hinton and Sejnowski [12] studied supervised learning in such networks and introduced the term *Boltzmann Machines* to emphasize the relation to statistical physics. Invoking the *Mean Field* approximation, Peterson and Anderson derived a deterministic self-consistent set of equations for the time-averages of the stochastic variables. Using these averages in the learning rule, they obtained substancial improvements in speed and performance [17].

In the next section the Bayesian approach to reconstruction is outlined, as model example we study the socalled *Weak String* model. This model was recently used to benchmark deterministic versus stochastic optimization schemes [3]. Section three discuss the Boltzmann Machine learning rule and section four contains experiments and concluding remarks.

2 Bayesian Visual Reconstruction

The Bayes approach to visual processing has long tradition, see eg. [13, 7] or [5] for a recent introduction. The basic idea is to consider both the source (un-degraded) signal and the degradation as stochastic processes. The Bayes formula can then be used to obtain the distribution $P(x|y)$ of the reconstructed signal (x), conditioned on the observed degraded signal (y):

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{1}$$

According to the standard interpretation the conditional distribution is the product of the distribution of the degradation process: $P(y|x) \equiv P(x \rightarrow y)$, and the *prior* distribution of the reconstructed signal $P(x)$. $P(x|y)$ of equation (1) is referred to as the *posterior* distribution. A useful estimate of the recontructed signal is given by the location of the mode of the posterior distribution, the socalled *Maximum A Posteori* estimate.

2.1 The Weak String Model

The *prior* distribution reflects our general insight on the source, for example expressing that the image represents light reflected from extended 3D structures etc. As an example we will study the socalled *Weak String* model see eg. Blake [3].

The Weak String is a simple model for reconstruction of piece-wise continous curves (1D signals) from noisy observations. In the lattice version the reconstructed curve is described by the unit values V_j. The prior information formalizes the expectation that neighbor values should be close, except when they are disconnected by the rare occasion of an active edge-unit, e_j. The *prior* probability distribution, for a string of N units, reads:

$$P[V,e] = Z_1^{-1} \exp\left(-\lambda \sum_{j=1}^{N-1} (1-e_j)(V_j - V_{j+1})^2 - \mu \sum_{j=1}^{N-1} e_j\right) \tag{2}$$

This is a *Gibbs* distribution[1]. Z_1 is a normalization constant. Note that the second term in the exponent acts as a *chemical potential* by which we can control the number of active edge-units (ie. $e_j = +1$).

The degradation process produce the measurements ($V_j \rightarrow d_j$). In the Weak String example we will for simplicity study the addition of zero-mean, unit variance, white Gaussian noise:

$$P[d|V] = Z_2^{-1} \exp\left(-1/2 \sum_{j=1}^{N} (V_j - d_j)^2\right) \tag{3}$$

Using Bayes formula (1), we can combine equations (2) and (3), to obtain the parametrized posterior distribution:

$$P[V,e|d] = Z^{-1} \exp\left(-1/2 \sum_{j=1}^{N} (V_j - d_j)^2 - \lambda \sum_{j=1}^{N-1} (1-e_j)(V_j - V_{j+1})^2 - \mu \sum_{j=1}^{N-1} e_j\right) \tag{4}$$

With the normalization constant given by:

$$Z = \int DV \int De \exp\left(-1/2 \sum_{j=1}^{N} (V_j - d_j)^2 - \lambda \sum_{j=1}^{N-1} (1-e_j)(V_j - V_{j+1})^2 - \mu \sum_{j=1}^{N-1} e_j\right) \tag{5}$$

This distribution is also a Gibbs distribution, corresponding to the usual Weak String cost-function [3]:

$$E[V,e,d] = 1/2 \sum_{j=1}^{N} (V_j - d_j)^2 + \lambda \sum_{j=1}^{N-1} (1-e_j)(V_j - V_{j+1})^2 + \mu \sum_{j=1}^{N-1} e_j \tag{6}$$

Note that the cost-function is linear in the two parameters (λ, μ).

2.2 Network design

By inspecting the posterior distribution we see that the two-component process (V_j, e_j) realizes a *Random Markov Field*. This property ensures that the neural network implementation only involves neighbor connectivity. Furthermore since the process is homogeneous along the string,

[1] A distribution of the form $P(x) = Z^{-1} \exp(-E(x)/T)$, where $E(x)$ is a cost-function, bounded from below, and T is a parameter

the network becomes highly *weight-shared* [16]. The few parameters of equation (6) controls all connections in the network, hence the learning process involves optimization over a low-dimensional parameter space. More complex Random Markov Field models have been designed for modelling textural features [6].

To enhance our sampling of the Random Markov variables we invoke a simulated annealing strategy as recommended by Geman and Geman [7]. The scheme is implemented by introducing a *temperature* T in the Gibbs distribution and design the sampler with a decreasing sequence of temperatures ending up at $T = 1$. The temperature dependent distribution reads:

$$P[V, e|d] = Z^{-1} \exp\left(-\frac{E[V, e, d]}{T}\right) \tag{7}$$

Direct sampling of this distribution, as investigated by Geman and Geman, is too slow for most applications and it is recommended to invoke a deterministic approximation scheme to obtain the necessary averages [3, 17]. The self-consistent Mean Field equations for the Weak String model read [9]:

$$V_j^{n+1} = \frac{\lambda[\left(1 - e_j^n\right) V_{j+1}^n + \left(1 - e_{j-1}^n\right) V_{j-1}^n] + d_j}{\lambda[\left(1 - e_j^n\right) + \left(1 - e_{j-1}^n\right)] + 1} \tag{8}$$

$$e_j^{n+1} = \tanh\left(\beta^n \left(\frac{\lambda}{2}\left(V_j^n - V_{j+1}^n\right)^2 - \mu\right)\right) \tag{9}$$

where $T^n = 1/\beta^n$ is the decreasing sequence of temperatures. These coupled equations define a recursive nearest neighbor connected neural network.

3 Boltzmann Machine Learning

The objective of the Boltzmann Machine Learning rule is to minimize the Kullback information distance between a target distribution $P_{w^*}[V, e|d]$, (of which the database is a finite sample), and the distribution $P_w[V, e|d]$ sampled by the current network with parameters w:

$$D[P_{w^*}, P_w] = \int Ddp[d] \int DV \int DeP_{w^*}[V, e|d] \log \frac{P_{w^*}[V, e|d]}{P_w[V, e|d]} \tag{10}$$

$\int Dx$ is short for $\int \prod_j dx_j$. In this formulation we have added the option of specifying the distribution of *inputs* explicitly by $p[d]$,[2] [11] . The learning algorithm is derived by gradient descent minimization of the information distance. The recursive learning algorithm for a system with a cost-function linear in the parameters (ie. the logarithm of the corresponding Gibbs distribution is linear in the parameters) is:

[2] In case the model contain hidden units, we use the distance between the *marginal* distributions obtained by integrating over the hidden varables

$$\Delta w_\nu = \eta \left(\langle \Omega_\nu \left(V, e, d \right) \rangle_{clamped} - \langle \Omega_\nu \left(V, e, d \right) \rangle_{free} \right) \tag{11}$$

for a any cost-function linear in the parameters w_ν:

$$E(V, e, d) = \sum_\nu \Omega_\nu \left(V, e, d \right) w_\nu \tag{12}$$

Where $\Omega_\nu \left(V, e, d \right)$ is an expression in the stochastic variables. $\langle ... \rangle_{clamped}$ indicates that the average is performed with respect to the current Gibbs distribution, with *fixed* values for all the stochastic variables that are specified as *input or output* variables in an example of the database. Similarly $\langle ... \rangle_{free}$ is the average with only *input* variables fixed. Hence we can characterize the rhs. of equation (11) as the difference between the correlations in situations with and without the teacher specifying the correct output [10].

4 Experiments and concluding remarks

Within the Weak String model we can think of various learning situations. In this presentation we will consider the case where the supervisor specifies the output (V_j), while the edge-units (e_j) are hidden. We adapt the two-dimensional parameter set $w = (\lambda, \mu)$.

The experiment is a *teacher-student design*. The database of examples for supervised learning is created by adding noise to a piece-wise constant signal, hence forming the example input. The input is then presented to the teacher network, ie. a network with prescribed parameters $w = (\lambda^*, \mu^*)$. The output of the teacher network is included in the database as the example output.

In figure 1. we show a typical input-output example created with network with parameters $\lambda^* = 1.0$ and $\mu^* = 0.05$. These parameters are used for the teacher network in the learning process below.

The input was created as a rectangular signal degraded by additive white gaussian noise. The teacher experiences no difficulty in reconstruction the piece-wise continous signal.

In figure 2. we show learning trajectories for four different starting points of the stochastic recursive gradient descent process. An *on line* situation is simulated where the teacher produces a sequence of answers to degraded random rectangular signals. The gradient descent parameters where fixed at the values $\eta_\lambda = 15.5/N$ and $\eta_\mu = 0.03/N$, where $N = 100$ is the signal sample size.

In conclusion we have shown that the Boltzmann learning rule can be used for identification of parameters in Bayesian visual reconstruction models. The learning rule can be used in situation where incomplete or no supervision is present ie. for models with hidden units [9].

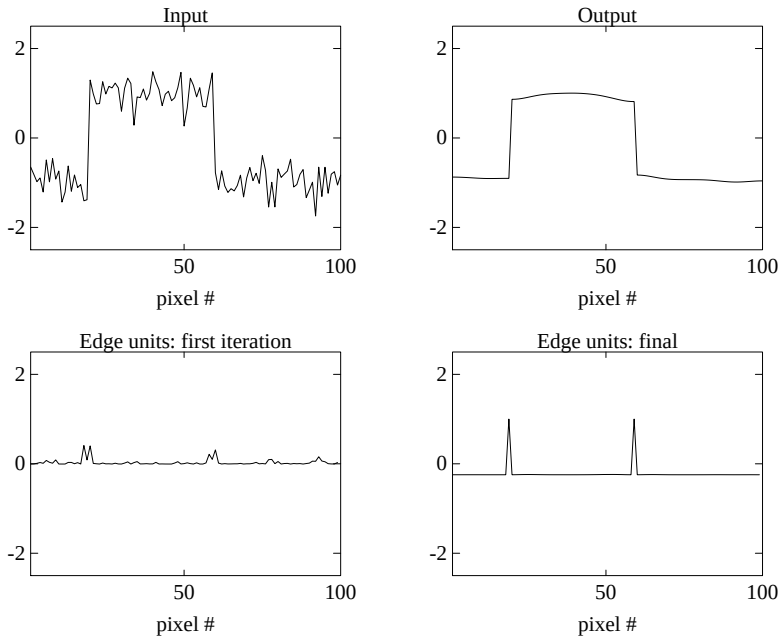

Figure 1: Teacher network performance. Input (upper left), output after 50 iterations (upper right), edge units after first iteration (lower left), and edge units after 50 iterations (lower right)

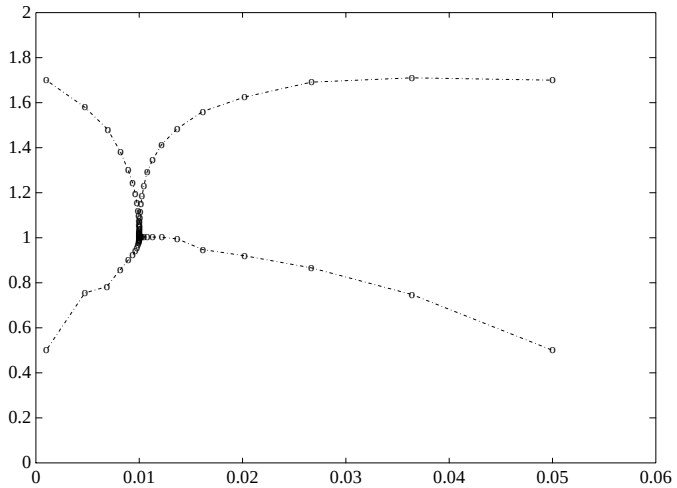

Figure 2: Learning trajectories for a teacher-student design. The teacher parameters are (μ^*, λ^*)= (0.01,1.0).

Acknowledgement

This research is supported by the Danish Research Councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center.

References

[1] J. Besag: *Spatial Interaction and the Statistical Analysis of Lattice Systems* J.Roy.Statist.Soc, **B36**, 192-236, (1974).

[2] J. Besag: *On the Statistical Analysis of Dirty Pictures* J.Roy.Statist.Soc, **B48**, 259-302, (1986).

[3] A. Blake: *Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-11** 2-12 (1989).

[4] P. Carnevali, L. Coletti and S. Paternello: *Image processing by simulated annealing* IBM Journal of Research and Development **29**, 569-579, (1985).

[5] J.B. Cole: *The Statistical Mechanics of Image Recovery and Pattern Recognition* Am.J.Phys. **59**, 839-42, (1991).

[6] G.R. Cross and A.K. Jain: *Markov Random Field Texture Models* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-5** 25-39 (1983).

[7] D. Geman and S. Geman: *Stochastic Relaxation, Gibbs distributions and the Bayesian restoration of images* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-6** 721-741 (1984).

[8] D. Geman, S. Geman, C. Graffigne and P. Dong: *Boundary Detection by Constrained Optimization* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-12** 609-628 (1990).

[9] L.K. Hansen, in preparation.

[10] J. Hertz: *A Gauge Theory in Computational Vision* Physica Scripta **39**, 161-167 (1989).

[11] J. Hertz, A. Krogh and R.G. Palmer: *Introduction to the Theory of Neural Computation* Addison Wesley, New York (1991).

[12] Hinton, G.E. and Sejnowski T.J.: *Learning and Relearning in Boltzmann Machines* In Parallel Distributed Processing, vol. 1, chap 7. Eds. Rumelhart D.E. et al. Cambridge: MIT Press.

[13] B.R. Hunt: *Bayesian Methods in Nonlinear Digital Image Restoration* IEEE Transactions on Computers **C-76**, 219-229, (1977).

[14] Intel Corporation: *80170NX, Electrically Trainable Analog Neural Network* Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051. (1991).

[15] S. Lakshmanan and H Derin: *Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-11** 799-813 (1989).

[16] Y. Le Cun, J.S. Denker, and S.A. Solla: *Optimal Brain Damage* in Neural Information Processing Systems, Denver 1989, ed. D. Touretzky, Morgan Kaufman, 598-605, (1989).

[17] C. Peterson and J.R. Anderson: *A Mean Field Learning Algorithm for Neural Networks* Complex Systems **1** 995-1019, (1987).

[18] T. Poggio, V. Torre and C. Koch. *Computational vision and regularization theory* Nature **317**, 314-319, (1985).

[19] M.W. Roth: *Survey of Neural Network Technology for Automatic Target Recognition* IEEE Transactions on Neural Networks **1**, 28-43, (1990).