A Neural Net for Reconstruction of Multiple Curves with a Visual Grammar

Eric Mjolsness, Anand Rangarajan, and Charles Garrett
Department of Computer Science
Yale University
New Haven, CT 06520-2158

Abstract

We derive a neural net for reconstructing a set of curves from ungrouped dot locations. The network performs Bayesian inference on a *visual grammer*, which serves as probabilistic model of the image formation process, by means of quadratic matching objective function.

1 Introduction

We derive a neural net for reconstructing a set of curves from ungrouped dot locations. The network performs Bayesian inference on a visual grammer, which serves as probabilistic model of the image formation process, by means of quadratic matching objective function analogous to those used in neural nets for inexact graph matching [3, 1].

The steps involved in the derivation are: (1) formulate a stochastic grammar (we choose relatively simple grammars involving multiple hierarchical objects); (2) derive its probability distribution on images, along with the partition function which is a configuration space integral over both discrete and continuous variables; (3) change variables by exploiting the structure of the original grammar; (4) use Mean Field Theory (MFT) to derive an objective function whose optimization permits the approximation of averages under the distribution; (5) introduce optimizing neural net dynamics, possibly after transforming the objective function to decrease the size of the network. Recent advances in MFT methods have greatly enhanced the practicality of the resulting networks.

2 Multiple Curves

2.1 Curve Grammar

This problem involves perceptual organization: one must extract multiple curves from a random-dot pattern. A probabilistic grammar for curve grouping is shown in Table 1.

Level 0 of the grammar generates a set of curves with certainty. Level 1 of the grammar augments the curve set and Level 2 extends each curve. The first term in e_2 measures the slope of the curve between two adjacent points on the curve. The second term corresponds to the expected distance between two dots lying on the curve and the third term allows for different curvatures. Level 3 scrambles the dots using a permutation matrix which results in the observed dots. The image generation probabilities can be obtained from the grammar. The probability distribution associated with a particular rule Γ^r is

$$\Pr(\text{new terms}, \{\text{new parameters}\}|\text{old terms}, \text{old parameters}) = e^{-\beta E_r}/Z_r \tag{1}$$

where $\beta \to 1$. Such probabilities can be repeatedly combined to yield a final probability distribution for the entire grammar.

One model for the prior probability $\Pr(P)$ is to feign ignorance of the permutation and use the maximum entropy distribution on P, namely a uniform distribution. This model seems artificial because there is no

0-7803-0164-1/91/0000-0615$01.00©1991 IEEE

Table 1: Probabilistic Grammar for curve grouping

make set of curves	Γ^0 :	root	$\rightarrow$	curveset(0)
		E_0	$=$	0 No alternatives $\Rightarrow$ certainty.
extend curve set	Γ^1 :	curveset(c)	$\rightarrow$	$\begin{cases} \{\text{curveset}(c+1), \text{curve}(c+1, s=0, \mathbf{x}, \theta)\} & \text{if } \omega_c = 1; \\ \text{nothing} & \text{if } \omega_c = 0. \end{cases}$
		$E_1(\mathbf{x})$	$=$	$\mu\omega_c + \frac{1}{2\sigma_0^2}\lvert\mathbf{x}\rvert^2$
extend curve by one dot	Γ^2 :	curve($c, s, \mathbf{x}, \theta$)	$\rightarrow$	$\begin{cases} \{\text{curve}(c, s+1, \mathbf{x}', \theta'), \text{dot}(c, s, \mathbf{x}, \theta)\} & \text{if } \omega_{cs} = 1; \\ \text{nothing} & \text{if } \omega_{cs} = 0. \end{cases}$
		$E_2(\mathbf{x}, \theta, \mathbf{x}', \theta')$	$=$	$\nu\omega_{cs} + e_2(\mathbf{x} - \mathbf{x}', \theta, \theta - \theta')$, where
		$e_2(\mathbf{\Delta x}, \theta, \Delta\theta)$	$\equiv$	$\frac{1}{2\sigma_\theta^2}\left(\arctan\left(\frac{\Delta x_2}{\Delta x_1}\right) - \theta\right)^2 + \frac{1}{2\sigma_r^2}\left(\lvert\mathbf{\Delta x}\rvert^2 - l^2\right)^2 + \frac{1}{2\sigma_{\text{bend}}^2}(\Delta\theta)^2$
scramble all dots	Γ^3 :	$\{\text{dot}(c, s, \mathbf{x}_{cs})\}$	$\rightarrow$	$\{\text{imagedot}(\mathbf{x}_i = \sum_{cs} P_{cs,i}\mathbf{x}_{cs})\}$
		$E_3(\{\mathbf{x}_i\})$	$=$	$-\log \sum_{\left\{\begin{array}{l} P \vert \sum_i P_{cs,i} \leq 1 \\ \text{and} \sum_{cs} P_{cs,i} = 1 \end{array}\right\}} \Pr(P) \prod_i \delta(\mathbf{x}_i - \sum_{cs} P_{cs,i}\mathbf{x}_{cs})$

actual uniform-probability scrambling mechanism in natural image-generation processes. But, it can be shown that a broad class of more detailed models $\Pr(P)$ are equivalent to the uniform-distribution model.

In our notation, $c = 1, .., C$ is the curve index, $s = 1, .., S_c$ is the dot index along a curve where S_c is the number of dots in curve c. The dot locations and orientations generated by the grammar are $\{\mathbf{x}_{cs}\}$ and $\{\theta_{cs}\}$ and the image dot locations and orientations are $\{\mathbf{x}_i\}$ and $\{\theta_i\}$. δ^K and δ^D correspond to the Kronecker and Dirac delta functions respectively. The overall joint probability is:

$$\Pr(\{\mathbf{x}_i\}, \{\theta_i\}, \{\mathbf{x}_{cs}\}, \{\theta_{cs}\}, \{P_{cs,i}\}, C, \{S_c, c = 1, .., C\}|N) = (1 - q_1)\prod_{c=1}^{C}\Bigg[q_1(1 - q_2)q_2^{S_c}$$
$$\exp\left\{-\beta\left[E_1(\mathbf{x}_{c1}) + \sum_{s=1}^{S_c - 1} E_2(\mathbf{x}_{c(s+1)} - \mathbf{x}_{cs}, \theta_{cs}, \theta_{c(s+1)} - \theta_{cs})\right]\right\}/Z_1 Z_2^{S_c}$$
$$\prod_{s=1}^{S_c}\left\{\delta^D(\mathbf{x}_{cs}, \sum_{i=1}^{N} P_{cs,i}\mathbf{x}_i)\delta^D(\theta_{cs}, \sum_{i=1}^{N} P_{cs,i}\theta_i)\delta^K(N, \sum_{c=1}^{C} S_c)\right\}\Bigg] \tag{2}$$

where $q_1 = \exp(\mu)$, $q_2 = \exp(\nu)$, C is the number of curves, $P_{cs,i}$ is the permutation matrix introduced in Level 3 of the grammar and N is the number of perceived image dots. The distribution function contains the free parameter β corresponding to the inverse of the temperature. β is identified with a deterministic annealing process. The expected values are computed from the distribution at $\beta = 1$. Also, the expected values serve as an approximation to the most probable values under the distribution.

The properties of a permutation matrix P can be exploited to simplify the above expression. Using the commutation property and the properties of a Dirac delta function, we can easily integrate out $\{\mathbf{x}_{cs}\}, \{\theta_{cs}\}$ and simplify the resulting expression to get

$$\Pr(\mathbf{x}_i, \theta_i, P_{cs,i}, C, (S_c, c = 1, .., C)|N) = (1 - q_1)\left[\frac{q_1(1 - q_2)}{Z_1}\right]^C \left[\frac{q_2}{Z_2}\right]^N \exp(-\beta E(\{P\}, \{\mathbf{x}_i\}, \{\theta_i\})) \tag{3}$$

where

$$E(\{P\}, \{\mathbf{x}_i\}, \{\theta_i\}) = \sum_{i=1}^{N}\left(\sum_{c=1}^{C} P_{c1,i}\right) E_1(\mathbf{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sum_{c=1}^{C}\sum_{s=1}^{S_c - 1} P_{cs,i}P_{c(s+1),j}\right) E_2(\mathbf{x}_j - \mathbf{x}_i, \theta_i, \theta_j - \theta_i) \tag{4}$$

Our goal is the solution to an inference problem. We seek to group the observed features (dots) corresponding to points on curve(s). Thus far, we have obtained a joint distribution on the dots and the model. Using Bayes theorem, we can switch to the distribution of the model conditioned on the data giving us

Table 2: The Expression of Syntactical Constraints

Syntactical Constraints	Expressions
Total number of curves= C	$C = N - \sum_{i=1}^{N}\sum_{j=1}^{N} \text{next}_{ij}$
No-loop constraint	$\sum_{i=1}^{N} \text{next}_{ij}\text{mbr}_i = \text{mbr}_j - 1$
Start-element constraint	$\sum_{c=1}^{C} \text{start}_{cj} = 1 - \sum_{i=1}^{N} \text{next}_{ij}$
End-element constraint	$\sum_{j=1}^{N} \text{next}_{ij} \leq 1$
Presence-absence constraint	$\{\text{next}_{ij}\}, \{\text{start}_{ci}\} \in \{0,1\}$ and $\{\text{mbr}_i\} \in \{1,..,N\}$

$\Pr(\{P_{cs,i}\}, \{\theta_i\}|\{\mathbf{x}_i\}, N)$. The corresponding partition function is

$$Z = \sum_{\text{configs}} \Pr(\{\theta_i\}, \{P_{cs,i}\}, C, (S_c, c = 1, .., C)|\{\mathbf{x}_i\}, N) \tag{5}$$

The expected values can be computed (at $\beta = 1$) by taking appropriate derivatives of the partition function.

2.2 Change of variables

The energy function in Eq. 4 can be further simplified by a suitable change of variables. Consider the following transformations.

$$\text{next}_{ij} = \sum_{c=1}^{C}\sum_{s=1}^{S_c-1} P_{cs,i}P_{c(s+1),j}, \ \text{start}_{ci} = P_{c1,i}, \ \text{mbr}_i = \sum_{c=1}^{C}\sum_{s=1}^{S_c} sP_{cs,i} \tag{6}$$

The choice of the new variables is not arbitrary. $\{\text{next}_{ij}\}$ tracks the membership of the data elements i and j in the same curve with the constraint that j follows i as the ***next member*** in the chain. $\{\text{start}_{ci}\}$ turns on, i.e. $\text{start}_{ci} = 1$, if i is the starting element of curve c. $\{\text{mbr}_i\}$ reindexes the data element i in terms of its membership number (mbr) in a curve. The membership number of the starting element in any curve c is one and for the last element in the chain, it is S_c.

We now show that the $(\{\text{next}_{ij}\}, \{\text{start}_{cj}\}, \{\text{mbr}_i\})$ space is isomorphic to the $(P, C, \{S_c\})$ space.

The inverse transformations corresponding to Eq. 6 are listed below. The proof will be shown elsewhere.

$$P_{c(s+1),j} = \sum_{i=1}^{N} \text{next}_{ij}P_{cs,i}, \ P_{c1,j} = \text{start}_{cj}, \ C = \sum_{c=1}^{C}\sum_{j=1}^{N} \text{start}_{cj} \tag{7}$$

The $\{S_c\}$ variables can only be indirectly recovered from the new space. However, an explicit expression is not required in the derivation of the new probability distribution.

The constraints needed to adequately characterize the problem undergo a transformation as well. In their formulation in terms of the permutation matrix $\{P_{cs,i}\}$, the constraints remain ***general***. We require that $\{P_{cs,i}\}$ satisfy the properties of a permutation matrix and that the sum over all indices equals the number of observed dots N. As we move from the $(P, C, \{S_c\})$ space to the $(\{\text{next}_{ij}\}, \{\text{start}_{cj}\}, \{\text{mbr}_i\})$ space, the constraints become ***problem specific***. The new constraints are dictated by the grammar and are a natural consequence of the choice of the new variables. The new constraints are given in Table 2.

$$\Pr(\{\theta_i\}, \{\text{next}_{ij}\}, \{\text{start}_{ci}\}, \{\text{mbr}_i\}|\{\mathbf{x}_i\}, N) = \frac{1}{Z}\left[\exp\left(\left\{N - \sum_{i=1}^{N}\sum_{j=1}^{N} \text{next}_{ij}\right\}\log G - \beta E(\{\text{next}_{ij}\}, \{\mathbf{x}_i\}, \{\theta_i\})\right)\right]$$

where $G = \frac{q_1(1-q_2)}{(2\pi\sigma_0^2)}$ and

$$E(\{\text{next}_{ij}\}, \{\mathbf{x}_i\}, \{\theta_i\}) = \sum_{i=1}^{N}\left(1 - \sum_{j=1}^{N}\text{next}_{ji}\right)E_1(\mathbf{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N}\text{next}_{ij}E_2(\mathbf{x}_j - \mathbf{x}_i, \theta_i, \theta_j - \theta_i) \tag{8}$$

Finally,

$$Z = \int d\{\theta_i\} \sum_{\{\text{ Syntactical Constraints }\}} \exp\left[\left(N - \sum_{i=1}^{N}\sum_{j=1}^{N}\text{next}_{ij}\right)\log G - \beta E(\{\text{next}_{ij}\},\{\mathbf{x}_i\},\{\theta_i\})\right]$$

2.3 Derivation of the neural network

We are now in a position to obtain a neural network from the partition function. We seek an objective function that exploits redundancies and constraints inherent in the problem formulation. The process by which the objective function is obtained relies on recent advances in MFT, wherein a variety of approximations to the partition function are utilized. In this section, we proceed from the partition function to a neural net. The neural net enables fast, parallel computation of the expected values of the relevent variables of interest. In addition, the expected values are tracked through the variation of a control parameter, the temperature. As the temperature is lowered, the expected values can also be used as an approximation to the global maximum *aposteriori* (MAP) estimate.

Since the {start} variables do not appear in the energy function, they can be summed over in the partition function and as there are $C! = (N - \sum_{i=1}^{N}\sum_{j=1}^{N}\text{next}_{ij})!$ ways of assigning a curve number to a starting element, we get

$$\begin{aligned} Z \;=\; & \int d\{\theta_i\} \sum_{\{\omega_i\in\{0,1\}\}} \sum_{\left\{\substack{\text{next}_{ij}\in\{0,1\}| \\ \sum_{i=1}^{N}\text{next}_{ij}\leq 1}\right\}} \prod_{i=1}^{N}\delta^K\left(\sum_{j=1}^{N}\text{next}_{ij} - (1-\omega_i)\right) \\ & \sum_{\left\{\substack{\text{mbr}_i| \\ 1\leq\text{mbr}_i\leq N}\right\}} \prod_{j=1}^{N}\delta^K\left(\text{mbr}_j - 1 - \sum_{i=1}^{N}\text{next}_{ij}\text{mbr}_i\right) \\ & \exp\left[\log\left(N - \sum_{i=1}^{N}\sum_{j=1}^{N}\text{next}_{ij}\right)! + \left(N - \sum_{i=1}^{N}\sum_{j=1}^{N}\text{next}_{ij}\right)\log G - \beta E(\{\text{next}_{ij}\},\{\mathbf{x}_i\},\{\theta_i\})\right] \end{aligned} \tag{9}$$

The constraint $\sum_{j=1}^{N}\text{next}_{ij} \leq 1$ has been replaced by the constraint $\sum_{j=1}^{N}\text{next}_{ij} = 1 - \omega_i, \omega_i \in \{0,1\}$. We use the Gaussian form of the Kronecker delta functions, namely, $\delta^K(m,n) = \lim_{A\to\infty}\exp\left(-\frac{A}{2}(m-n)^2\right)$ and we introduce auxiliary variables through the transformation

$$y(e) = \int_R df\ \delta^D(e,f)y(f) = \int_R df \int_I dh\ e^{h(e-f)}y(f) \tag{10}$$

where R and I denotes integrals along the real and imaginary axis respectively. This is the well known trick in the statistical mechanics literature which enables us to convert a discrete sum on dependent variables into an exact integral on auxiliary variables ***and*** a discrete sum on ***independent*** variables. The transformation leads us naturally to the ***saddle-point*** approximation [2]. The energy function also contains the $\log((N-\sum_{ij}\text{next}_{ij})!)$ term which can be simplified by exploiting the Stirling's approximation for $M! = (M+\frac{1}{2})\log(M+\frac{1}{2})-(M+\frac{1}{2})$. This term is also transformed using auxiliary variables.

$$<\text{mbr}_i>= v_i = g_{\text{mbr}}(u_i),\ <\text{next}_{ij}>= \chi_{ij} = g_{\text{next}}(W_{ij}),\ <\omega_i>= \zeta_i = g_\omega(\eta_i),\ <C+1/2>= \sigma = g_\tau(\tau) \tag{11}$$

with the angle brackets denoting the expected values of the variables under Z. All discrete variables become independent of each other in the partition function. Now, the discrete sums with respect to these independent variables can be carried out and Z can be written as:

$$\begin{aligned} Z \overset{def}{=} & \lim_{\frac{A}{\beta}\to\infty,\frac{B}{\beta}\to\infty} \int d\{\theta_i\}\int_R d\{\zeta_i\}\int_I d\{\eta_i\}\int_R d\{\chi_{ij}\}\int_I d\{W_{ij}\}\int_R d\{v_i\}\int_I d\{u_i\}\int_R d\tau\int_I d\sigma \\ & \exp(S(\{\zeta_i\},\{\eta_i\},\{\chi_{ij}\},\{W_{ij}\},\{v_i\},\{u_i\}) - \beta\hat{E}(\{\chi_{ij}\},\{\mathbf{x}_i\},\{\theta_i\},\{\zeta_i\},\{v_i\})) \end{aligned} \tag{12}$$

where S, the entropy is

$$\begin{aligned} S &= \sum_{i=1}^{N}(1+\exp(\eta_i)-\eta_i\zeta_i)+\sum_{j=1}^{N}\log\left(1+\sum_i \exp(W_{ij})\right)-\sum_i\sum_j W_{ij}\chi_{ij} \\ &+\sum_{i=1}^{N}\left\{\log\left(\frac{\exp(N+1)u_i-\exp(u_i)}{\exp(u_i)-1}\right)-u_iv_i\right\}+(\tau\log\tau-\tau)-\tau\sigma \end{aligned}$$

and the energy function is

$$\hat{E}(\{\chi_{ij}\},\{\mathbf{x}_i\},\{\theta_i\},\{\zeta_i\},\{v_i\}) = E(\{\chi_{ij}\},\{\mathbf{x}_i\},\{\theta_i\})+\frac{A}{2\beta}\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\chi_{ij}-1+\zeta_i\right)^2$$

$$+\frac{B}{2\beta}\sum_{j=1}^{N}\left(v_j-1-\sum_{i=1}^{N}\chi_{ij}v_i\right)^2-\frac{1}{\beta}\sigma\log\left(N+\frac{1}{2}-\sum_{i=1}^{N}\sum_{j=1}^{N}\chi_{ij}\right)-\frac{1}{\beta}(N-\sum_{i=1}^{N}\sum_{j=1}^{N}\chi_{ij})\log G \qquad (13)$$

The entropy term involves configuration space summations over (1) $\{\omega_i\}$, (2) $\{\text{next}_{ij}\}$, and (3) $\{\text{mbr}_i\}$. The first summation is an ordinary sum over the binary configurations of each $\omega_i, i = 1, .., N$ taken separately. The second summation imposes the constraint that the $\{\text{next}_{ij}\}$ variables sum up to one or zero, which leads to a very different partition function than one obtained without imposing this global constraint. Recent experiments and analysis have shown that the performance of the network is drastically improved [2]. The final summation imposes the constraint that the $\{\text{mbr}_i\}$ variables assume values in the set $\{1, .., N\}$.

As the temperature is reduced, the saddle-point approximation becomes increasingly accurate. When Z is approximated around its saddle points and $F \overset{def}{=} \hat{E} - \frac{1}{\beta}S$, we get

$$\begin{array}{lcccclcl} \frac{\partial F}{\partial \eta_i} & = 0 & \Rightarrow & \zeta_i & = & \frac{d}{d\eta_i}\log(1+\exp(\eta_i)) & = & \frac{1}{1+e^{-\eta_i}} & = & g_\omega(\eta_i) \\ \frac{\partial F}{\partial \zeta_i} & = 0 & \Rightarrow & \eta_i & = & -\beta\frac{\partial \hat{E}}{\partial \zeta_i} \\ \frac{\partial F}{\partial W_{ij}} & = 0 & \Rightarrow & \chi_{ij} & = & \frac{\partial}{\partial W_{ij}}\log(1+\exp(\sum_{i=1}^{N} W_{ij})) & = & \frac{\exp(W_{ij})}{1+\sum_{i=1}^{N}\exp(W_{ij})} & = & g_{\text{next}}(\{W_{ij}\}) \\ \frac{\partial F}{\partial \chi_{ij}} & = 0 & \Rightarrow & W_{ij} & = & -\beta\frac{\partial \hat{E}}{\partial \chi_{ij}} \\ \frac{\partial F}{\partial u_i} & = 0 & \Rightarrow & v_i & = & \frac{d}{du_i}\log\left(\frac{\exp((N+1)u_i)-\exp(u_i)}{\exp(u_i)-1}\right) & = & \begin{array}{c}\left(\frac{N}{1-\exp(-Nu_i)}\right) \\ -\left(\frac{1}{1-\exp(-u_i)}\right)\end{array} & = & g_{\text{mbr}}(u_i) \\ \frac{\partial F}{\partial v_i} & = 0 & \Rightarrow & u_i & = & -\beta\frac{\partial \hat{E}}{\partial v_i} \\ \frac{\partial F}{\partial \tau} & = 0 & \Rightarrow & \sigma & = & \frac{d}{d\tau}(\tau\log\tau-\tau) & = & \log\tau & = & g_\tau(\tau) \\ \frac{\partial F}{\partial \sigma} & = 0 & \Rightarrow & \tau & = & -\beta\frac{\partial \hat{E}}{\partial \sigma} \\ \frac{\partial F}{\partial \theta_i} & = 0 & \Rightarrow & \frac{\partial \hat{E}}{\partial \theta i} & = & 0 \end{array} \qquad (14)$$

The set of equations defines the fixed points of a dynamical system generated using gradient descent on the free energy F.

3 Experiments and Discussion

The algorithm was tested on a dot pattern corresponding to three curves. The pattern was hand drawn. The β parameter was increased from 0.01 to a final value of 2. This was necessary to force the expected values of the discrete variables to assume close to integer values. The result is shown in Figure 1. Instead of grouping the data into three curves, the network generated four curves and an isolated point. The parameter settings were:

$$\sigma_0 = 1,\ \sigma_\theta = 0.05,\ \sigma_r = 0.05,\ \sigma_{bend} = \pi/8,\ \text{and}\ l = 0.22$$

β was adjusted according to the schedule $2 - \beta_{new} = 0.9(2 - \beta_{old})$ and the parameters A and B were varied according to $A = B = \frac{2}{2-\beta}$ satisfying the constraint that they rise faster than β. It is instructive to note

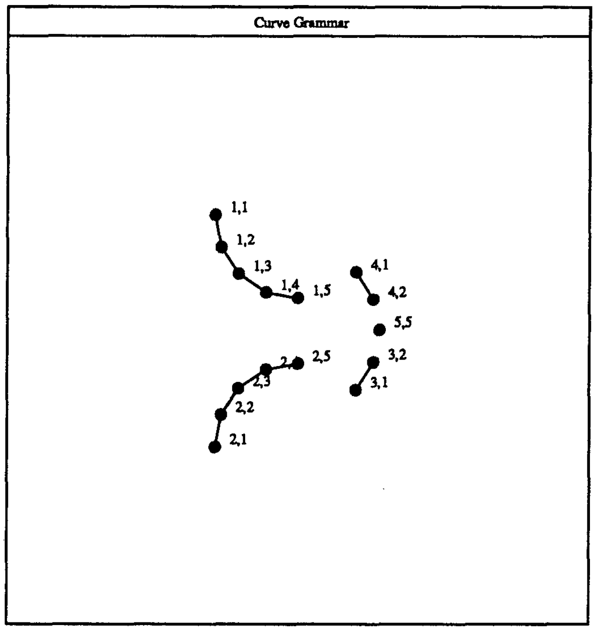

Figure 1: Multiple curve reconstruction

that the membership numbers along the broken curve increase in different directions. The network cannot correct itself once the different directions get entrenched.

4 Conclusions

We have suggested a visual grammar expressive enough to group random dots into curves. A neural network was derived that performs Bayesian inference on this visual grammar. Recent developments in MFT have been utilized in deriving the network from the partition function corresponding to the distribution of the dots conditioned on the data. The network yields the expected values of the variables which at sufficiently low temperatures are close to the integer values of the discrete random variables.

References

[1] Eric Mjolsness, Gene Gindi, and P. Anandan. Optimization in model matching and perceptual organization. *Neural Computation*, 1:218–229, 1989.

[2] Carsten Peterson and Bo Soderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22, 1989.

[3] Christoph von der Malsburg. Pattern recognition by labeled graph matching. *Neural Networks*, 1:141–148, 1988.