Implementing belief propagation in neural circuits

Aaron P. Shon and Rajesh P. N. Rao
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350

Abstract

There is growing evidence that neural circuits may employ statistical algorithms for inference and learning. Many such algorithms can be derived from independence diagrams (*graphical models*) showing causal relationships between random variables. A general algorithm for inference in graphical models is *belief propagation*, where nodes in a graphical model determine values for random variables by combining observed values with messages passed between neighboring nodes. We propose that small groups of synaptic connections between neurons in cortex correspond to causal dependencies in an underlying graphical model. Our results suggest a new probabilistic framework for computation in the neocortex.

Key words: graphical models, cerebral cortex, integrate-and-fire model, Bayesian computation, recurrent networks

1 Introduction

Graphical models [1] have recently gained favor in the statistical machine learning community. These models represent causal dependencies between random variables using a graph structure. Nodes in the graph stand for random

Email address: {aaron,rao}@cs.washington.edu (Aaron P. Shon and Rajesh P. N. Rao).

variables, while edges between nodes represent conditional dependencies between variables (see, e.g., Fig. 1(a)). These models encompass as special cases a rich variety of statistical frameworks, including the well-known Kalman filter [2] and Rauch-Tung-Striebel smoother [3] for continuous state spaces with Gaussian density and linear dynamics, hidden Markov models (HMMs) [4] for discrete state spaces described using a Markov chain, and Markov random fields (MRFs) [5]. Inference in graphical models can be custom-tailored to a specific graphical model (e.g., the forward-backward algorithm [6] for HMMs), but recent work has demonstrated that inference can be computed for many classes of graphical model using a single algorithm. This algorithm is called belief propagation [1], and although designed to work only in acyclic graphs, it can give excellent performance even in some cases of loopy graph structure [7]. Evidence is emerging [8] that recurrent neural circuits may implement a type of Bayesian computation, in which a "bottom-up" likelihood is combined with a "top-down" prior to yield a posterior distribution over states of a probabilistic model. We argue that recurrently connected networks of biological neurons can implement a form of belief propagation. In a direct analog to graphical models, we propose that synaptic connections between neurons can encode the same causual dependencies as the edges in graphical models. We further demonstrate how the central equations of belief propagation can be implemented using integrate-and-fire models of biological neural networks.

2 Framework for probabilistic inference

2.1 Graphical models and belief propagation

The belief that a random variable X_i in a graphical model takes on value x_i is denoted by $b_i(x_i)$. The sum-product algorithm for belief propagation [1] uses the product of the local evidence observed for X_i, denoted $\phi_i(x_i)$, and messages m_{ji}, received from other nodes, to determine what value X_i should take on.

Messages to node i come from nodes j in a neighborhood $\mathcal{N}(i)$. The belief over values x_i of X_i is normalized by a constant k_1 to yield a distribution:

$$b_i(x_i) = k_1 \phi_i(x_i) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i) \tag{1}$$

We interpret the log of b_i as the rate at which neuron i should fire, or equivalently the log probability that the temporal or spatial feature detected by that neuron is present in the environment. Normalizing constant k_1 could be computed by, e.g., a group of inhibitory interneurons receiving input from all the pyramidal cells in a cortical column.

The message passed from node j to signal that node i should assume value x_i is computed by marginalizing (integrating out) variable X_j. Each value x_j that X_j could assume is weighted by a function $\psi_{ji}(x_j, x_i)$, multiplied by the product of messages coming into node j, and again multiplied by the evidence $\phi_j(x_j)$ coming into node j. Another constant k_2 can be used to normalize over different values x_j. The final form of message $m_{ji}(x_i)$ is:

$$m_{ji}(x_i) = k_2 \sum_{x_j} \phi_j(x_j) \psi_{ji}(x_j, x_i) \prod_{o \in \mathcal{N}(j) - \{i\}} m_{oj}(x_j) \tag{2}$$

2.2 Neural correlates of algorithmic computations

Based on neurophysiological findings that suggest neural spike rates perform computations in the log domain [9,10], some researchers have proposed that neuronal spike rates transmit log probabilities. Similarly, we advocate the hypothesis that neurons transmit log beliefs via their firing rates r_{isi}. The meaning of a neuron's firing or not firing depends on the neurons to which it connects; neurons in sensory cortices, for instance, might use firing rates to encode the belief that a particular feature is present. For example, in visual cortex, the random variable X_i could encode the belief that an edge of a particular orientation is present in neuron i's receptive field. Thus, the firing rate-based tuning curve of neuron i represents the neuron's "belief" that its receptive field contains an edge at a certain preferred orientation.

Taking the log of equation 1, we make a correspondence to the neuroanatomy:

$$r_{\mathrm{isi}}^{post} \propto \log\left(b_i(x_i)\right) = \log k_1 + \log\phi_i(x_i) + \sum_{j\in\mathcal{N}(i)} \log m_{ji}(x_i) \tag{3}$$

The first term represents some background mean firing rate. The second term, $\log\phi_i(x_i)$, represents contributions to the postsynaptic firing rate from feedforward inputs arriving via other cortical or subcortical areas. The third term represents summed currents arriving from recurrent synaptic inputs (e.g., from other pyramidal cells in the same cortical column). Note that summation of currents is proportional to r_{isi} based on Stein's approximation [11] for sufficiently large currents I_e:

$$r_{\mathrm{isi}}^{post} \approx \left[\frac{E_L - V_{th}}{C_m R_m\left(V_{th} - V_{reset}\right)} + \frac{I_e}{C_m\left(V_{th} - V_{reset}\right)}\right] \tag{4}$$

where E_L is the leakage potential, V_{th} is the threshold voltage, V_{reset} is the resting potential, I_e is the current flowing into the cell, and C_m is membrane capacitance. Although Tal and Schwartz [12] showed that the current-firing rate relationship is logarithmic for very large currents and for certain membrane time constant values, we assume synaptic conductances and presynaptic firing rates fall in a range where the current-firing rate relationship is linear.

2.3 Neural message passing

We propose that expected values of the log current passed from one neuron to another (averaged over some time window) represent belief propagation messages passed using the sum-product algorithm. Our derivation begins as follows. Since neurons can only encode the states $x_i = 0$ or $x_i = 1$, messages must only integrate over those values:

$$m_{ji}(x_i) = k_2\phi_j(0)\psi_{ji}(0,x_i)\prod_{p\in\mathcal{N}(j)-\{i\}} m_{pj}(0) + k_2\phi_j(1)\psi_{ji}(1,x_i)\prod_{p\in\mathcal{N}(j)-\{i\}} m_{pj}(1) \tag{5}$$

Several implementations of this equation are possible, depending on how general a graphical model the cortex implements. In one, somewhat restricted, interpretation, the first term in the sum may be ignored (see Fig. 1(b) circuit i)). In this case, when neuron j does not fire (i.e. when $x_j = 0$), $\psi_{ji}(0, x_i) = 0$ regardless of the value of x_i. That is, when neuron j does not fire, it transmits a message value of 0, so only the second term of the sum contributes to the message. This interpretation could be justified by noting that when pyramidal cells do not fire, synaptic conductance for postsynaptic AMPA channels is effectively blocked. More flexible configurations are shown in Fig. 1(b) circuits ii) and iii), where inhibitory interneurons implement the term $m_{ji}(0, x_i)$.

Groups of neurons usually conceived as feature detectors in sensory cortices might thus be interpreted as performing probabilistic inference. Note that in circuits such as Fig. 1(b) ii) and iii), we must approximate a log of a sum using a sum of logs; this could be done via a least-squares weighting term in the peak synaptic conductances as in [8]. For ease of explication, the rest of this paper uses the specialized definition of $\psi_{ji}(0, x_i) = 0$ given above.

Note also that only $m_{ji}(x_i = 1)$ is transmitted, not $m_{ji}(x_i = 0)$ (though this value must be taken into account on the postsynaptic side). This makes intuitive sense, since in a graphical model with binary-valued nodes, we only need to report messages that a node X_i should take on the value 1; transmitting both the belief that the value 0 is correct and that the value 1 is correct would be redundant. Since the first term of the above sum disappears, and using the definition $\log b_j(x_j) \propto r_{\text{isi}}^{pre}$, we regroup the terms of the message:

$$m_{ji}(x_i) = \phi_j(x_j = 1)\psi_{ji}(1, x_i) \prod_{p \in \mathcal{N}(j) - \{i\}} m_{pj}(x_p = 1) \tag{6}$$

$$m_{ji}(x_i) = \psi_{ji}(1, x_i) b_j(1) \tag{7}$$

$$m_{ji}(x_i) = \psi_{ji}\,(1, x_i) \exp\left(r_{\text{isi}}^{pre}\right) \tag{8}$$

Each presynaptic spike injects an amount of current proportional to the synaptic conductance g_{ji}^{syn} from neuron j to neuron i. This implies $g_{ji}^{syn} \propto r_{\text{isi}}^{pre}$, i.e. $g_{ji}^{syn} \propto \log m_{ji}$. We define E_S as the reversal potential across an AMPA synapse and $\mathrm{E}\left(g_{ji}^{syn}\right)$ as the expected value of the conductance:

$$\log m_{ji}(x_i) = \mathrm{E}\left(g_{ji}^{syn}\right)(V - E_S) \tag{9}$$

Thus the expected value of the current passed into the cell is proportional to the log of the message. Assuming that the postsynaptic neuron is kept near the firing threshold, $V - E_S$ should remain nearly constant over time, making it irrelevant to the spike rate (and hence the value of the message). An intriguing alternative is that short-term synaptic plasticity [13] plays a role in regulating the value of this term, ensuring that the local membrane voltage V remains approximately constant even as the current into the cell changes as a result of changing conductance. Substituting equation 9 into equation 5 gives us the correct result: we obtain equation 3 above, showing how excitatory neurons could implement the sum-product algorithm for belief propagation.

An interesting effect occurs when we consider modifying the peak conductance g_{ji}^{peak} by spike-timing dependent plasticity (STDP) [14] using an exponential window (Fig. 1(c)). Assume that the firing of a presynaptic neuron j is positively correlated with the firing of the postsynaptic neuron i (a similar argument will hold for negative correlations due to the symmetry of the STDP window). Further, we assume that STDP has synchronized the firing of the neurons (see e.g. [15]) such that neuron j fires over an interval right before neuron i begins firing. Thus, presynaptic spikes always occur over some interval $\left[t_{start}^{pre}, t_{start}^{post}\right]$, where t_{start}^{pre} and t_{start}^{post} denote the starting times for firing of the pre- and postsynaptic neurons, respectively. The length of this window will be inversely proportional to r_{isi}^{pre}. Fig. 1(d) shows how the amount of synaptic modification under STDP (in the case of such synchronized neurons) is logarithmic in r_{isi}^{pre}. A linear change in the belief b_i of the presynaptic neuron that the postsynaptic cell should fire thus effects a linear change in the message m_{ji} sent. That is, STDP appears to modify the function ψ_{ji} linearly in the belief of the presynaptic neuron.

Fig. 1 (e) demonstrates some preliminary results. We show a 15×15 image (black square on white), corrupted (top left) by additive Gaussian noise. Results are shown after running the sum-product algorithm on the noisy image (top right) and a recurrent network of integrate-and-fire neurons (bottom) with input currents ϕ proportional to the noisy input image, run for 200 msec of simulated time. Parameters of simulated neurons were: capacitance $C = 0.5$nF, resistance $R = 40\text{M}\Omega$, absolute refractory period $\tau = 5$msec, excitatory reversal potential $E_{AMPA} = 0$mV, inhibitory reversal potential $E_{GABA} = -80$mV, leakage potential $E_{leak} = -60$mV, threshold voltage $V_{th} = -40$mV, peak excitatory synaptic conductance $g_{exc}^{peak} = 0.0398$nS, and peak inhibitory synaptic conductance $g_{inh}^{peak} = 0.0011$nS. Output of the neural network is given by r_{isi}^{post} for each model neuron, averaged over the last 20 msec of simulation time (exponentiating the firing rate gives the estimated probability of each pixel's being on or off). The sum-product algorithm perfectly recovers the image; the neural network approximately recovers the image. Callout boxes show spike rasters over the last 20 msec for a few sample neurons.

3 Conclusion

We have described how biologically realistic neurons might implement belief propagation for probabilistic inference. Several issues remain for future work:

(1) Addressing reasons for the logarithmic current-firing rate relationship when neurons are in saturation.

(2) Determining whether recurrently connected topologies in biological networks give empirically reasonable results under loopy belief propagation.

(3) Exploring circuits where inhibitory synapses convey messages that the postsynaptic neuron shouldn't fire—how might $m_{ji}(x_i = 0)$ be conveyed?

(4) Identifying how low-level graphical models implemented by small numbers of neurons might lead to more complex, hierarchical graphical models in a wider view of cortical architecture. For instance, might cortical columns formed out of such low-level graphical models themselves comprise nodes in a higher-level graphical model (Fig. 1(d))?

Biological systems, living in complex, stochastic environments, have likely discovered similar approaches to machine learning algorithms for learning and inference. We expect that synapse-level interactions of the type described here interact with network-level effects to create powerful, flexible representations of graphical models in biological neural networks.

Acknowledgments

This work is being supported by the Sloan Foundation and NSF BITS grant no. 0130705.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, San Mateo, CA: Morgan Kaufman Publishers, 1988.

[2] R. E. Kalman, A new approach to linear filtering and prediction problems, Journal of Basic Engineering (1960) 35–45.

[3] H. E. Rauch, F. Tung, C. T. Striebel, Maximum likelihood estimates of linear dynamic systems, J. Amer. Inst. Aeronautics and Astronautics 3 (8) (1965) 1445–1450.

[4] L. E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, Ann. Math. Stat. 37 (1966) 1554–1563.

[5] J. W. Woods, Two-dimensional discrete Markovian fields, IEEE Trans. Inf. Thry. 18 (1972) 232–240.

[6] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[7] Y. Weiss, W. T. Freeman, Correctness of belief propagation in Gaussian graphical models of arbitrary topology, Neural Computation 13 (10) (2001) 2173–2200.

[8] R. P. N. Rao, Bayesian computation in recurrent neural circuits, Neural Computation 16 (1).

[9] R. H. S. Carpenter, M. L. L. Williams, Neural computation of log likelihood in control of saccadic eye movements, Nature 377 (1995) 59–62.

[10] M. N. Shadlen, W. T. Newsome, Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey, J. Neurophysiol. 86 (4) (2001) 1916–1936.

[11] R. B. Stein, The frequency of nerve action potentials generated by applied currents, Proc. Royal Society B 167 (1967) 64–86.

[12] D. Tal, E. Schwartz, Computing with the leaky integrate-and-fire neuron: Logarithmic computation and multiplication, Neural Computation 9 (2) (1997) 305–318.

[13] H. Markram, M. Tsodyks, Redistribution of synaptic efficacy between neocortical pyramidal neurons, Nature 382 (1996) 807–810.

[14] H. Markram, J. Lübke, M. Frotscher, B. Sakmann, Regulation of synaptic efficacy by coindence of postsynaptic APs and EPSPs, Science 275 (1997) 213–215.

[15] T. Nowotny, V. P. Zhigulin, A. I. Selverston, H. D. I. Abarbanel, M. I. Rabinovich, Enhancement of synchronization in a hybrid neural circuit by spike-timing dependent plasticity, J. Neurosci 23 (30) (2003) 9776–9785.

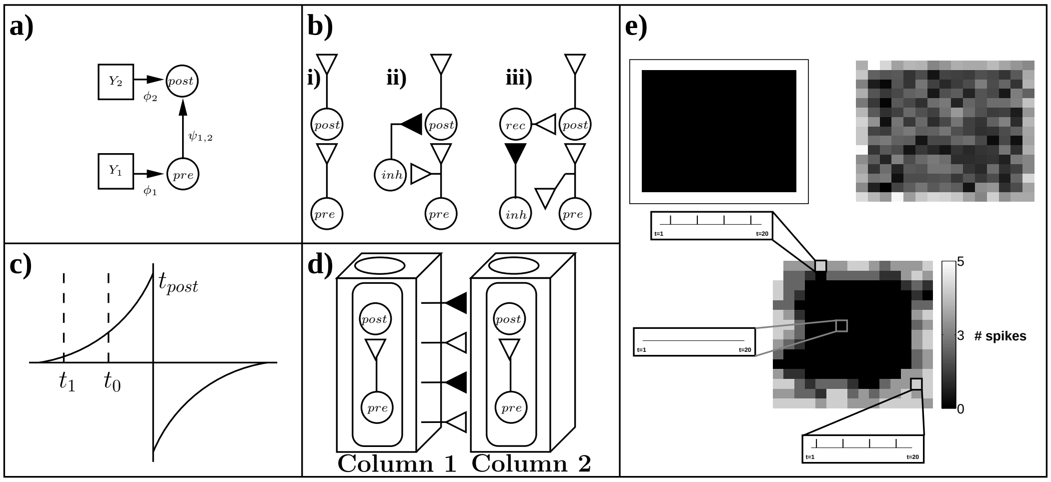

Fig. 1: Belief propagation in biologically plausible neural networks: (a) Sample graphical model. Circles represent (hidden) random variables X; squares represent observed random variables Y. (b) Three different neural circuits analogous to the graphical model. Empty connections represent excitatory synapses; filled connections represent inhibitory connections. Each pyramidal cell receives inputs from stellate cells on layer IV (not shown). In circuit i), the term $m_{ji}(x_j = 0)$ is not considered in the sum in equation 5. In circuit ii), this term is encoded in the activity of inhibitory neuron *inh*. In circuit iii), recurrent neuron *rec* provides additional excitation to *post* when *pre* does not fire. Other neural circuit configurations for message passing are possible; this remains a topic for further investigation. (c) An exponential STDP kernel influences ψ_{ji} in a linear manner. If neurons are synchronized such that neuron j reliably stops firing just before neuron i starts, and we double the presynaptic firing rate such that neuron j starts firing at time t_0 instead of t_1, we obtain a logarithmic change in peak conductance g_{ji}^{peak}. (d) Inference in continuous, vector-valued graphical models could be computed hierarchically by cortical columns, each containing many simple graphical models realized using simple neural circuits of the type shown in (b). (e) Example image (top left), noisy input image (top right), and image reconstructed by a 15 × 15 simulated network of recurrently-connected integrate-and-fire neurons (bottom); callouts show spike rasters from a few model neurons (see text for details).