

Neuronal integration of dynamic sources: Bayesian learning and Bayesian inference

Hava T. Siegelmann and Lars E. Holzman

*Department of Computer Science, University of Massachusetts at Amherst,
Amherst, Massachusetts 01003, USA*

(Received 16 June 2010; accepted 31 August 2010; published online 28 September 2010)

One of the brain's most basic functions is integrating sensory data from diverse sources. This ability causes us to question whether the neural system is computationally capable of intelligently integrating data, not only when sources have known, fixed relative dependencies but also when it must determine such relative weightings based on dynamic conditions, and then use these learned weightings to accurately infer information about the world. We suggest that the brain is, in fact, fully capable of computing this parallel task in a single network and describe a neural inspired circuit with this property. Our implementation suggests the possibility that evidence learning requires a more complex organization of the network than was previously assumed, where neurons have different specialties, whose emergence brings the desired adaptivity seen in human online inference. © 2010 American Institute of Physics. [doi:10.1063/1.3491237]

Our senses work in parallel passing multimodal data of the same fact or object to the brain. A fundamental question in the field of computational neuroscience is how the brain accommodates sensory data from different sources to form one holistic picture. Cue integration experiments, in which subjects experience apparently synchronized cross modal stimuli, but where one source is displaced from its counterpart, can reveal how the brain handles parallel inputs. The results suggest that the human brain computes a weighted average over the different cues, fitting neatly with Bayes' probability theory, where each piece of information is weighted by the amount, on which the brain relies on the channeling sense. Studies now suggest that this relative reliability must also take into account changing conditions, such as varying lighting for vision, environmental noise levels for audio, etc. Using fixed reliabilities, as was previously modeled, skews decision making and contradicts both optimality and human studies. A question arises as to whether the previous Bayes-based theory still holds for cue integration given the need of adapting the levels of reliabilities. We propose it does and introduce a neural network architecture that can both constantly learn the reliabilities of sensory data and use them in the integration of cues. While the brain does not necessarily use the method that we are proposing for our architecture and there is currently no way to test exactly how the brain makes these computations, our work provides a proof of concept of the brain's ability to handle learning and inference in parallel and within one network.

I. INTRODUCTION

People may perform Bayesian-like inference in the way they combine data, such as in scene and object recognition, sensorimotor learning, utilizing common sense, and making decisions.¹⁻⁴ *Bayesian inference* is reasoning about values of variables, which takes into consideration existing probability

distributions and new evidence. Optimally, this process combines different sources, weighting their relative reliability, into the most plausible conclusion in terms of posterior probability of an unseen variable. *Bayesian learning*, as opposed to Bayesian inference, is a technique derived from Bayesian statistics that allows the optimal estimation of probability distributions including both prior distributions of the variables and conditional distributions relating them.

Neural models were proposed, demonstrating Bayesian inference given fixed conditional probabilities.⁵⁻⁹ Yet, Bayesian inference alone cannot explain perception unless the relative reliabilities are updated over time.¹⁰⁻¹³ Two recent solutions were suggested. One introduces an approximation of an inference rule via reinforcement learning without specifying neural correlates;¹⁴ the other one¹⁵ suggests a framework of neurons with spikes that behave in a Poisson-like way to enable learning. We provide another approach, where a demonstrated novel analog neural circuit calculates posterior probabilities and at the same time and in a different time scale learns the relative reliabilities to fit the reality of changing conditions. The approach is innovative in that it introduces neurons of different roles and specialties: Those that carry distributions as in the traditional approach and those used for explicit evidence based learning. While existing artificial neural network models are far more simplified than the brain, they do share certain inherent characteristics standard computational models do not possess. Thus, our work can be interpreted as a proof of concept demonstrating that brain-like architectures are able to both execute optimal inference of parallel stimuli—doing so adaptively and to suggest that optimal evidence learning may require a more complex, less homogeneous neural organization than previously assumed.

II. THE NEURON MODEL

The model neuron used in our suggested Bayesian circuit is based on sigma-pi operations, inspired by the obser-

vation that neurons and systems have local nonlinearities (e.g., in dendrites) that cause them to approximate low-order polynomials.^{16–19} The input synapses to each neuron j are clustered to dendritic regions $d_j \in D_j$.^{20,21} One abstract weight, $w_j^{I(d_j)}$, is defined per dendritic region, where $I(d_j)$ is the set of presynaptic neurons and j is the postsynaptic neuron. Modification of the weight represents long-term potentiation/depression that may occur through plasticity of the entire synaptic region (e.g., tuning the nonlinearity with respect to the spatial locality of individual synapses) and/or plasticity of single synapses (e.g., the addition or removal of receptors).

In our simplified model, neuron j computes local multiplication in the individual dendritic regions. The soma sums the activity from the dendritic regions weighted by $w_j^{I(d_j)}$ and the output activity of the neuron is piecewise linear with respect to the somatic activity,

$$a_j = L\left(\sum_{d_j \in D_j} w_j^{I(d_j)} \prod_{i \in I(d_j)} a_i\right), \tag{1}$$

where

$$L(x) = \begin{cases} 0, & x < 0 \\ x, & 0 < x < 1 \\ 1, & x > 1. \end{cases} \tag{2}$$

A normalization via division will be used. This can be implemented by changing the neural update equation to

$$a_j = L\left(\sum_{d_j \in D_j} w_j^{I(d_j)} e^{\sum_{i \in I(d_j)} \log a_i}\right) \tag{3}$$

or instead by assuming that a normalization occurs naturally within the neuronal circuit, where the firing rates keep the relative ratios rather than the exact values.

Comment: Many nonlinearities from real neurons are known, and we provide one such interpretation based on multiplications in the dendritic trees. While it is unknown whether our interpretation of the nonlinearities is more likely than other models, one can also consider the analog neurons described above as representing sets of probabilistic binary neurons. The value of each analog neuron corresponds, in this interpretation, to the fraction of the joint firing in the associated group: if they all fire, the value is 1; if none fires, they are represented by 0; and if some fire, they will be represented by the appropriate fraction. A similar distributed form of memory was described,²² where groups of neurons fire together and synchronize locally. The individual neurons in this case are probabilistic, and their firing probability is updated by Eq. (1). This also makes our work interface.¹⁵ It should be noted that the simplified sigmoid function of Eq. (2) is used here for simplicity of proofs only, yet the results will not be changed if it is substituted with any sigmoidal-like function.²³

III. THE INFERENCE

The literature on inference and learning in graphical models is by now vast and highly sophisticated (see, e.g., Refs. 24–26). As the purpose of this paper is to contribute to neuroscience rather than to graphical models, we chose to

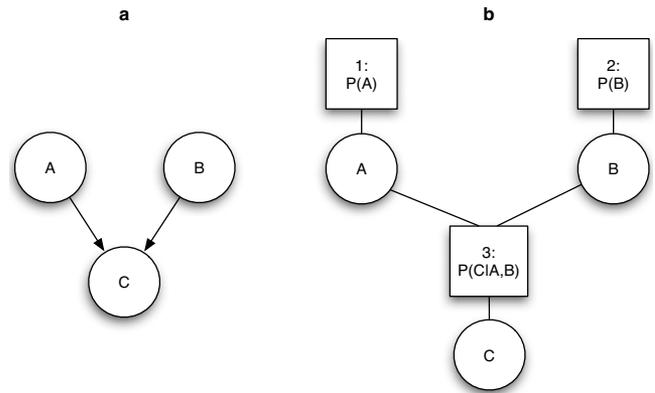


FIG. 1. (a) Bayesian network of three variables where C is dependent on both A and B. (b) The factor graph representation of the same network. The M nodes are A–C and the N nodes are 1–3.

simulate a simple propagation algorithm, which is sufficient to demonstrate our point. It is very likely that richer learning algorithms can also be implemented on neural inspired architectures.

A. Bayesian factor graphs

Bayesian inference is commonly performed computationally on an acyclic factor graph.²⁷ This is an undirected graph with two types of nodes. The M nodes represent the variables and the N nodes represent the “factors,” which include prior probabilities on the variables as well as relationships between the variables in the form of conditional probabilities (see Fig. 1). Bayesian inference is tractable in acyclical graphical models. Belief propagation is an efficient way to infer via local messages, and generalizations of which were suggested.²⁸ We focus on the “loopy belief propagation” approximation algorithm,²⁹ which we find easy to simulate in a neural model. The loopy propagation sends one message along each edge: a μ -message from a factor to an adjacent variable and a ν -message from a variable to an adjacent factor.

A message is a vector. The components of the μ -message from factor F to variable A sum ν -messages received at F from all other variables neighboring F , scaled by the probability of the assignment. The sum is taken over all possible assignments of F 's neighboring variables when A is set to the value a ,

$$\mu_{FA}(a) = \sum_{\vec{\xi} \in \mathcal{N}(F) \setminus A} \left[f(\vec{\xi}, a) \prod_{b \in \vec{\xi}} \nu_{BF}(b) \right]. \tag{4}$$

Here, $\mathcal{N}(F) \setminus A$ is the set of all variable nodes neighboring F except for A , $\vec{\xi} \in \mathcal{N}(F) \setminus A$ represents a particular assignment to those nodes, f is the value of the factor (a conditional probability) for a particular assignment of the neighboring variables, b is the assignment to one of the neighboring variables, and B is the variable that was assigned. The components of a ν -message from a variable A to a factor F are products of the $A=a$ components of μ -messages sent to variable A from all other neighboring factors. This can be expressed as

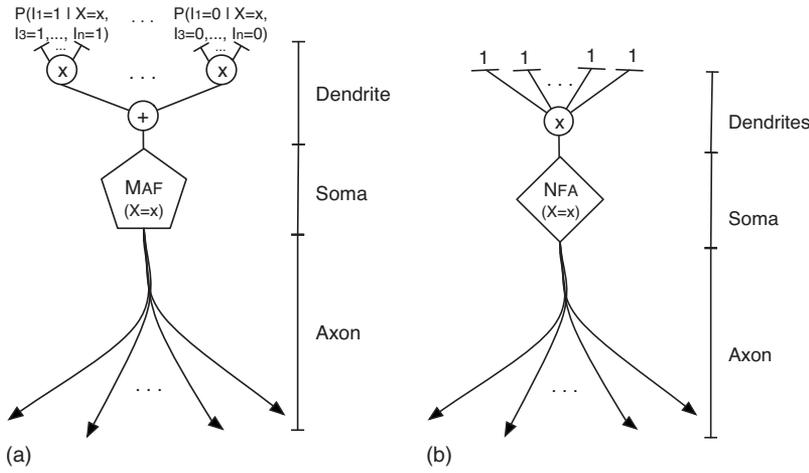


FIG. 2. (a) An M neuron. Neurons of this type calculate the μ -message from a factor F to a variable A . (b) An N neuron. Neurons of this type calculate the ν -message from a variable A to a factor F .

$$v_{AF}(a) = \prod_{j \in \mathcal{N}(A) \setminus F} \mu_{jA}(a), \tag{5}$$

where a is a particular assignment to variable A and $\mathcal{N}(A) \setminus F$ is the set of all factor nodes neighboring A except for F .

B. Neural implementation of inference

A main contribution of this paper is the implementation of the inference and weight learning steps within a particular neural inspired architecture. It was previously shown how to iteratively estimate priors in a spiking neural model,³⁰ but our work has the goal of combining learning with inference. Our architecture is somewhat tricky. The majority of the sigma-pi neurons in the architecture implement neither variables nor factors of the graphical model. Instead, these analog neurons represent messages in the propagation algorithm. With this nontrivial model, we gain the desired attribute of having the learning and inference processes coexist in the neural architecture as equal partners with equal importance, instead of learning being pushed to the implementation level serving the inference procedure. Thus, we will include two neural templates: type M neuron and type N neuron. Type M neuron, $M_{FA}(a)$, calculates the value of an entry in a μ -message sent from a factor to a variable (see Fig. 2). Each dendritic region corresponds to one assignment to the variables neighboring the factor F with the value of variable A set to a . An N neuron, $N_{AF}(a)$, calculates the value of an entry in a ν -message sent from a variable to a factor.

The posterior estimate of a variable, $p(A=a)$, is the product of the μ -messages sent to the variable node A . This is equivalent to multiplying any two messages that are passed along an edge connected to the variable node A , which can be easily done within a sigma-pi neuron,

$$P(A=a) = \frac{v_{AF}(a)\mu_{FA}(a)}{\sum_{b \in A} v_{AF}(b)\mu_{FA}(b)}. \tag{6}$$

Figures 3 and 4 exemplify the implemented Bayesian networks.

Inference from partial evidence is applied by the basic neural update. To enter evidence ε , the N and M neurons that include the observed variable A receive the entry of 1 for the value observed and 0 to all other entries,

$$N_{AF}(a) = M_{FA}(a) = \begin{cases} 1 & \text{if } (A=a) \in \varepsilon \\ 0 & \text{if } (A=a) \notin \varepsilon. \end{cases} \tag{7}$$

By applying the neural dynamics, the output activity of individual neurons, as described by Eq. (1), will converge to the appropriate values of the messages due to the one-to-one

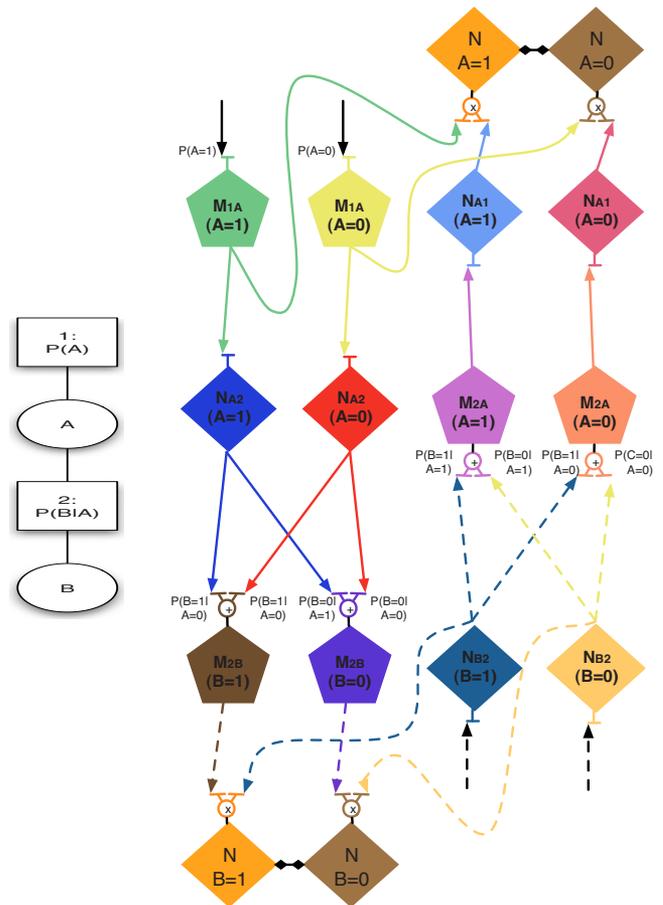


FIG. 3. (Color online) (Left) The factor graph of two variables in which B is conditioned on A. (Right) The neural implementation of these messages. Each neuron and its attached dendrite tree (shown as blunt ended edges) and axon tree (shown as directed edges) are represented in the same color. Un-labeled synapses have weight 1 and also unconnected edges carry the constant value 1.

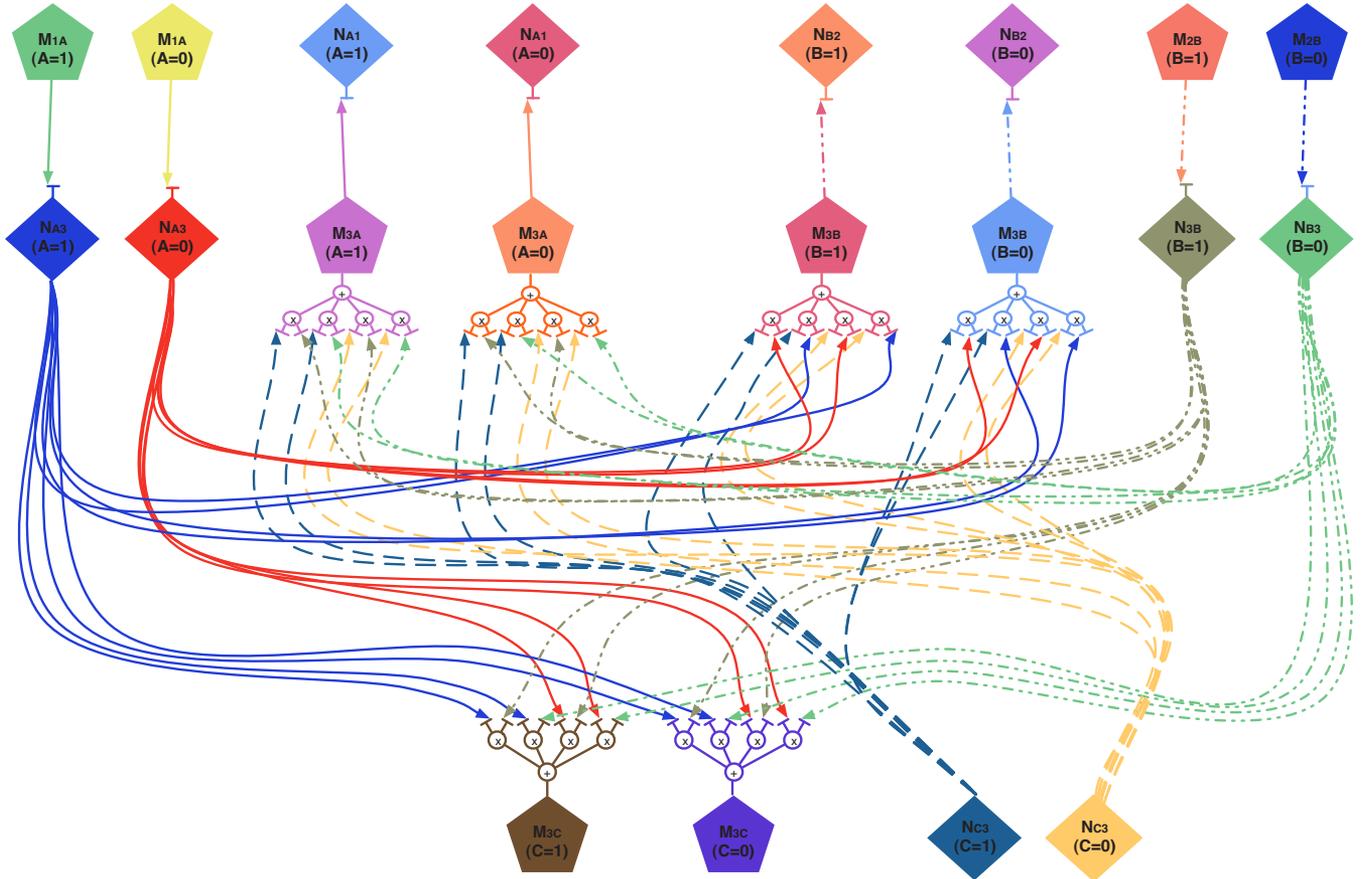


FIG. 4. (Color online) The neural implementation of the graph in Fig. 1(b). Synaptic trees fan out to consider evidence from multiple variables.

correspondence between the computation of the neurons and message-passing equations.

The update in our model is done when the neuron is ready. Such update can occur synchronously based on an external pacemaker or asynchronously when each neuron receives the input. Assuming that there is a latency of l milliseconds from the time a neuron receives its input until its output firing rate appears, and d is the diameter of the acyclic factor graph, the activity will then converge within $2dl$ to the right posterior estimates. While the architecture is cumbersome, it carries the benefit of having adaptable reliabilities.

IV. MODEL LEARNING IN SYNAPSES

A significant contribution of this work is to show how the circuit described above can learn online the distribution and mutual effects among the variables, and how the learning is done by adapting the weights of synaptic regions in this neural circuit using a “generalized Hebbian learning” rule.³¹ In generalized Hebbian learning, the weight of a synapse is updated from the current weight by a function F of the presynaptic and postsynaptic neural activities,

$$w_j^{I(d_j)}(t+1) = w_j^{I(d_j)}(t) + F[a_{i_1}(t), \dots, a_{i_n}(t), a_j(t)]. \quad (8)$$

The generalized Hebbian learning is local, can be done online with a single piece of evidence and thus is considered biologically plausible. Our particular implementation involves active steps of learning with neurons being allocated

and specialized with this role, differently than more traditional approaches.¹⁴ We propose that learning is important enough to brain function that it allocates dedicated neural hardware toward this end.

In our neural implementation, the conditional or prior probabilities are stored in a distributed fashion in the weights of the synaptic regions, and updating the weights is done by

$$w_j^{I(d_j)}(t+1) = w_j^{I(d_j)}(t) + \gamma_t \text{ev}_i(\vec{Y} = \vec{y}) [\text{ev}(X = x, \vec{Y} = \vec{y}) - w_j^{I(d_j)}(t)], \quad (9)$$

where ev is short for evidence, X is a variable with value x , and \vec{Y} is a vector of variables with the values \vec{y} .

If γ is decreased by

$$\gamma_{t+1} = \frac{\gamma_t}{\gamma_t + 1} \quad (10)$$

and it is initialized by the prior probability of X , then $w_j^{I(d_j)}(t+1) = P_{t+1}(X=x | \vec{Y}=\vec{y})$. It is also possible to consider a constant learning rate that weights recent information more heavily, allowing the estimate to track changes in the world. Furthermore, if any value is not drawn from the distribution (i.e., is incorrectly assigned) and biases the estimator, then a constant learning rate causes the value to vanish over time, yielding an estimator that is asymptotically unbiased. A behavior consistent with a constant learning rate has also been

a constant learning rate has also been observed biologically in a noisy sensory integration task.³²

Next, we explain how the learning rule was chosen. Assume that at any time t , $ev_t(\cdot)$ is assigned either 1 or 0, depending on whether the variables referenced in this evidence co-occur as stimuli at time t . Assume also that weight updates occur only in the presence of the conditioned stimuli $\vec{Y}=\vec{y}$ and otherwise the weight remains unchanged. We can then prove that the update rule represents the Bayesian optimal estimate of the probability $P_t(X=x|\vec{Y}=\vec{y})$ as follows:

$$\begin{aligned} w_t &= w_{t-1} + \frac{1}{t}(ev_t - w_{t-1}) \\ &= \frac{ev_t}{t} + \frac{t-1}{t}w_{t-1} \\ &= \frac{ev_t}{t} + \frac{t-1}{t} \left[\frac{ev_{t-1}}{t-1} + \frac{t-2}{t-1}w_{t-2} \right] \\ &= \frac{ev_t + ev_{t-1}}{t} + \frac{(t-2)w_{t-2}}{t}. \end{aligned}$$

Introducing a term a , where $a < t$, we can express the cumulative weight as

$$w_{t+1} = \frac{\sum_{i=0}^a ev_{t+1-i}}{t} + \frac{(t-a)w_{t-a}}{t}. \quad (11)$$

Assuming we look at the entire cumulative history (i.e., $a=t$), we reduce to only the first term

$$w_{t+1} = \frac{\sum_{i=0}^t ev_i}{t}. \quad (12)$$

Recalling that the update only occurs in the presence of $\vec{Y}=\vec{y}$, this is equivalent to

$$w_{t+1} = \frac{\sum_{i=0}^t ev_i(X=x|\vec{Y}=\vec{y})}{\sum_{i=0}^t ev_i(\vec{Y}=\vec{y})}. \quad (13)$$

This is now the formulation of the expected value of $P_t(X=x|\vec{Y}=\vec{y})$ in the Dirichlet formulation of multinomial Bayesian priors,

$$P_t(X=x|\vec{Y}=\vec{y}) = \frac{\sum_{i=0}^t ev_i(X=x|\vec{Y}=\vec{y})}{\sum_{i=0}^t ev_i(\vec{Y}=\vec{y})}. \quad (14)$$

All that remains is to demonstrate that the update provided for the learning rate in Eq. (10) is equivalent to $1/(t+1)$ at time $t+1$. This can be shown by a simple algebra,

$$\gamma_{t+1} = \frac{1}{t+1} = \frac{\frac{1}{t}}{1 + \frac{1}{t}} = \frac{\gamma_t}{1 + \gamma_t}.$$

V. EXAMPLE WITH CUE INTEGRATION

Cue integration refers to the estimation of a hidden property based on disparate information from multiple senses. In Refs. 3 and 4, subjects were asked to locate a cued location

on the table in front of them and then point to it with the tip of their left index finger from underneath the table. The cues about the location were given both visually and through proprioception (the internal sense, in this case, of where the finger is). The subject's right hand is placed on the table but it is hidden by a mirror that reflects the location of the finger by a light dot. The motion of the subject's right arm is restricted to two dimensions, radial distance and azimuth, which is orthogonal to distance, using mechanically restricting apparatus. Three tasks were defined in this setup. In the first one, the subject is asked to point his left finger using only the data from a dot on the screen. In the second task, the subject is asked to point only by feeling where his hidden right index finger is. In the third task, the subject is asked to point after integrating both the visual and the proprioceptive cues. Feedback was provided after each trial.

To study the process of integration, conflicting data were given to the subjects in the third task and their predictions were analyzed. It has been shown that the estimates rely differentially upon each of the senses in the azimuth and distance dimensions with preference for visual feedback along the azimuth dimension and preference for proprioceptive feedback along the distance dimension. The predictions used the relative precisions of each sense and combined the estimates along each dimension.⁴

It was concluded that the integration of cues is done in a Bayesian manner. A neural model of Bayesian cue integration was described.³³ Assumptions were made in that model about uniform priors over the target locations and that the variances are explicitly encoded and fixed in the sensory representations. Our neural network model can combine the senses in a Bayesian way without knowing in advance the different reliabilities of the senses or their variances, by first learning these hidden parameters through interaction with the environment, with a similar rationale to Ref. 15.

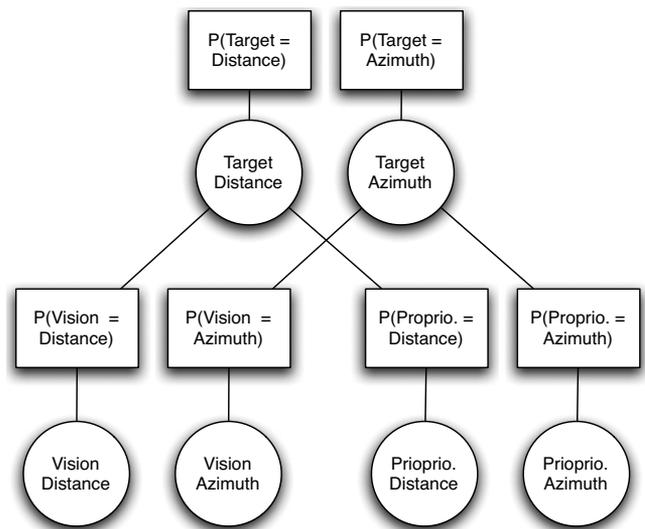


FIG. 5. A factor graph representation of the variables used to generate a Bayesian neural network for our experiments. The hidden variables are target distance and target azimuth, as well as the relative reliability of the first layer nodes.

TABLE I. Statistics about the size and complexity of the neural networks for cue integration. Many of the compartments have zero weights, which can be considered equivalent to having them pruned.

Network's properties	Cue integration
No. of M neurons	130
No. of N neurons	130
No. of P neurons	79
Average synaptic branches/neuron	4.67
Average nonzero branches/neuron	1.38
Average synapses/branch	1.46

A. Simulating cue integration

To simulate cue integration, we created a neural circuit of the type described in this paper. We used multinomial random variables to represent a discretized version of the space. For both vision and proprioception, separate variables represented the azimuth dimension (with values from -45° [-0.785 rad] to 45° [0.785 rad], with a step size of 7.5° [0.130 rad]) and the radial distance (with values from 0 to 6, with a step size of 0.5). We used similar variables for the hidden azimuth and distance of the target position, and the perceived locations were calculated with the factor graph model in Fig. 5. The network first learned the priors and the conditional probabilities from the evidence presented to it in a form of pairs along each dimension: (target location and proprioception) and (target location and vision). To prepare the evidence for one iteration of learning, a target location was chosen uniformly from the grid described by the variables. Gaussian noise was then applied to each estimate separately along each dimension. The variance of the noise depended on the sense being trained and was in accord with the experimental data. So, given a target location (T_{az}, T_{dist}) , a sensory location was produced as $(T_{az} + G(0, \sigma_{az, sense}), T_{dist} + G(0, \sigma_{dist, sense}))$ with $\sigma_{az, vision} < \sigma_{az, prop}$ and $\sigma_{dist, prop} < \sigma_{dist, vision}$.

We utilized 1000 training examples of each dimension (proprioception and vision). To analyze the robustness of the learning, the entire learning procedure was repeated five times and testing was performed on each learned network. In Table I, we report statistics about the network, showing both

TABLE II. The relative standard deviations for each of the experimental conditions.

Condition	$\sigma_{az, vision}$	$\sigma_{az, prop}$	$\sigma_{distance, vision}$	$\sigma_{distance, prop}$
A	1.0	2.0	2.0	1.0
B	1.0	1.5	1.5	1.0
C	1.0	1.75	1.25	1.0

how simple the topology is and how learning affects the structure. For testing, the network was presented with a set of points for each of the senses and it was allowed to iterate until producing the posterior distribution over the target location. All pairs of conflicting points from the grid with city-block distance greater than 2 were presented for testing.

Figure 6 demonstrates that the posterior estimates of the target locations are qualitatively similar to the results from the psychological experiments. The target estimates were compared to the points derived from Bayesian optimal estimates by taking the first moment of the product of the Gaussians,

$$\text{target}_{az} = \frac{\sigma_{az, vis}^2}{\sigma_{az, vis}^2 + \sigma_{az, pro}^2} \text{Vis}_{az} + \frac{\sigma_{az, pro}^2}{\sigma_{az, vis}^2 + \sigma_{az, pro}^2} \text{Pro}_{az}, \quad (15)$$

$$\text{target}_{dist} = \frac{\sigma_{dist, vis}^2}{\sigma_{dist, vis}^2 + \sigma_{dist, pro}^2} \text{Vis}_{dist} + \frac{\sigma_{dist, pro}^2}{\sigma_{dist, vis}^2 + \sigma_{dist, pro}^2} \text{Pro}_{dist}. \quad (16)$$

The neural network was trained and tested five times for each of the three conditions to control for variance in parameter estimation. In each conditions A, B, and C (as in Table II), the relative variances were modified to test the ability of the network to learn and apply the appropriate conditional probabilities. The mean/standard deviation of the root-mean-square error between the network's estimates of the target and the optimal Bayesian estimate for conditions A, B, and C were, respectively, $1.49^\circ/0.023^\circ$, $1.94^\circ/0.037^\circ$, and $1.65^\circ/0.046^\circ$. The mean/standard deviation root-mean-

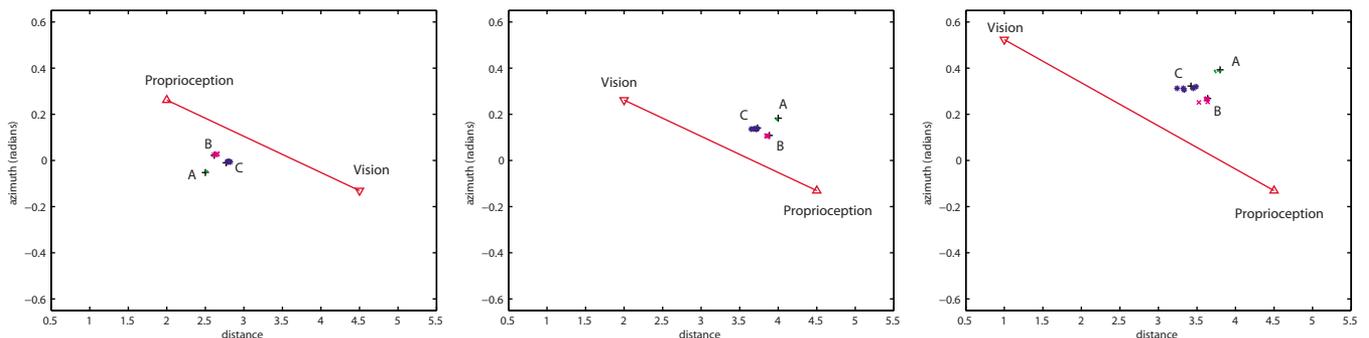


FIG. 6. (Color online) The results from cue integration neural network. Each box describes three experiments. The proprioception and vision provided conflicting information about distance (x axis) and azimuth (y axis). The information provided by proprioception and vision is indicated by the triangles, and a line is plotted between them. The data about these points were given with noise, having σ as described by conditions A–C from Table II. Black crosses in each figure indicate the optimal Bayesian estimates for each condition. Where not visible, the test points overlap exactly with the optimal point. In all neural experiments, the system managed to learn and then guess target locations very close to the Bayes optimal.

square errors for the distance were 0.066/0.0012, 0.0076/0.0025, and 0.0089/0.0034, respectively. Note that the error is significantly lower than the discretization of the variables, reflecting that multinomials can efficiently represent complex distributions. In fact, by observation, the majority of the error comes from variations produced by model learning over a relatively small number of examples. When the observed points are distant from each other in a particular dimension, there is increased variation due to the rarity of the noise model producing a point at that distance (despite the nonzero probability of doing so). In Fig. 6, the large amount of conflict in the distance dimensions increases the variability of the target estimate in that dimension.

VI. CONCLUSIONS

Much previous work within computational neuroscience has dealt with the problem of inference, or hidden state estimation, while in many cases prior assumptions are lacking, necessitating a learning phase that updates the priors based on data. We suggest that neural dynamics serves to perform the inference step, while Hebbian synaptic plasticity implements learning. The former process is surmised to operate on a fast time scale, as opposed to the latter process, which is slower.

Our work supports the hypothesis that in brainlike architectures, Bayesian priors may exist to provide default values of the probability distributions. The priors may be represented in a distributed fashion by the initial weights of the synaptic regions and also in the initial value of the learning rates. A large or fixed learning rate or a uniform distribution will describe noninformative data, which requires reliance on learning. A small learning rate or a highly peaked distribution will describe a genetic bias or expectation that is harder to overcome. It is possible that the brain includes different reliabilities in different areas for proper functioning.

Finally, we note that our approach deviates from the traditional one where learning is a mere emergent property of the neural mechanism that is developed toward applying inference. In the architecture introduced here, there are both neurons that represent prior and conditional probabilities (i.e., factors) in the traditional sense, but there are also neurons that represent messages which are key to learning the probabilities in a Bayesian fashion. This innovative view, accompanied by a nontrivial neural network design, indicates that perhaps to accomplish learning from evidence as is supported by experimental evidence, the simple architectures proposed before are not sufficient, and neurons of different specialties should exist to support optimal learning.

ACKNOWLEDGMENTS

This work was supported by ONR Grant No. N00014-09-1-0069 "Symbolic and Continuous Representation in Adaptive Memory with Efficient Inference."

¹K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature (London)* **427**, 244 (2004).

²T. L. Griffiths and J. B. Tenenbaum, "Optimal predictions in everyday cognition," *Psychol. Sci.* **17**(9), 767 (2006).

³R. J. van Beers, D. M. Wolpert, and P. Haggard, "When feeling is more

important than seeing in sensorimotor adaptation," *Curr. Biol.* **12**, 834 (2002).

⁴R. J. van Beers, A. C. Sittig, and J. J. Denier van der Gon, "Integration of proprioceptive and visual position-information: An experimentally supported model," *J. Neurophysiol.* **81**, 1355 (1999).

⁵R. P. N. Rao, "Hierarchical Bayesian inference in networks of spiking neurons," *Adv. Neural Inf. Process. Syst.* **17**, 1113 (2005).

⁶W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes," *Nat. Neurosci.* **9**, 1432 (2006).

⁷R. Zemel, P. Dayan, and A. Pouget, "Probabilistic interpretation of population codes," *Neural Comput.* **10**(2), 403 (1998).

⁸R. S. Zemel, J. M. Quentin, M. Huys, R. Natarajan, and P. Dayan, "Probabilistic computation in spiking populations," *Adv. Neural Inf. Process. Syst.* **17**, 1609 (2005).

⁹A. Steimer, W. Maass, and R. Douglas, "Belief-propagation in networks of spiking neurons," *Neural Comput.* **21**, 2502 (2009).

¹⁰A. Stocker and E. P. Simoncelli, "Sensory adaptation within a Bayesian framework for perception," *Adv. Neural Inf. Process. Syst.* **18**, 1291 (2006).

¹¹G. Hinton, "What kind of graphical model is the brain?," Proceedings of the International Joint Conference on Artificial Intelligence, Edinburgh, 2005.

¹²G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The wake-sleep algorithm for unsupervised neural networks," *Science* **268**, 1158 (1995).

¹³R. P. N. Rao, "Bayesian inference and attention in the visual cortex," *NeuroReport* **16**, 1843 (2005).

¹⁴B. Nessler, M. Pfeiffer, and W. Maass, "Hebbian learning of Bayes optimal decisions," *Adv. Neural Inf. Process. Syst.* **21**, 1169 (2009).

¹⁵J. M. Beck, R. Kiani, W. J. Ma, T. Hanks, A. K. Churchland, J. Roitman, M. N. Shadlen, P. E. Latham, and A. Pouget, "Probabilistic population codes for Bayesian decision making," *Neuron* **60**, 1142 (2008).

¹⁶C. Koch and I. Segev, "The role of single neurons in information processing," *Nat. Neurosci.* **3**, 1171 (2000).

¹⁷J. L. Pena and M. Konishi, "Auditory spatial receptive fields created by multiplication," *Science* **292**, 249 (2001).

¹⁸Y. Trotter and S. Celebrini, "Gaze direction controls response gain in primary visual-cortex neurons," *Nature (London)* **398**, 239 (1999).

¹⁹S. Treue and J. C. M. Trujillo, "Feature-based attention influences motion processing gain in macaque visual cortex," *Nature (London)* **399**, 575 (1999).

²⁰A. Polsky, B. W. Mel, and J. Schiller, "Computational subunits in thin dendrites of pyramidal cells," *Nat. Neurosci.* **7**, 621 (2004).

²¹P. Poirazi, T. Brannon, and B. W. Mel, "Pyramidal neuron as two-layer neural network," *Neuron* **37**, 989 (2003).

²²L. Valiant, "Memorization and association on a realistic neural model," *Neural Comput.* **17**, 527 (2005).

²³H. T. Siegelmann, *Neural Networks and Analog Computation: Beyond the Turing Limit* (Birkhäuser, Boston, 1998).

²⁴B. J. Frey and N. Jovic, "Comparison of algorithms for inference and learning in probabilistic graphical models," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1392 (2005).

²⁵M. J. Wainwright, "Estimating the wrong graphical model: Benefits in the computation limited setting," *J. Mach. Learn. Res.* **7**, 1829 (2006).

²⁶A. Globerson, Y. Weiss, D. Sontag, T. Meltzer, and T. Jaakkola, "Tightening lp relaxations for map using message passing," Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 2008.

²⁷F. V. Jensen, *Bayesian Networks and Decision Graphs* (Springer, New York, 2001).

²⁸W. T. Freeman, J. S. Yedidia, and Y. Weiss, "Understanding belief propagation and its generalizations," Proceedings of the International Joint Conference on Artificial Intelligence, Seattle, WA, 2001.

²⁹J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 2nd ed. (Morgan Kaufmann, San Francisco, 1988).

³⁰U. Ernst, D. Rotermund, and K. Pawelzik, "Efficient computation based on stochastic spikes," *Neural Comput.* **19**, 1313 (2007).

³¹T. H. Brown, E. W. Kairiss, and C. L. Keenan, "Hebbian synapses: Biophysical mechanisms and algorithms," *Annu. Rev. Neurosci.* **13**, 475 (1990).

³²R. J. Baddeley, H. A. Ingram, and R. C. Miall, "System identification applied to a visuomotor task: Near-optimal human performance in a noisy changing task," *J. Neurosci.* **27**(7), 3066 (2003).

³³S. Deneve and A. Pouget, "Bayesian multisensory integration and cross-modal spatial links," *J. Physiol. Paris* **98**(1-3), 249 (2004).