

Bayesian Computation in Recurrent Neural Circuits

Rajesh P. N. Rao

Department of Computer Science and Engineering

University of Washington

Seattle, WA 98195

E-mail: *rao@cs.washington.edu*

Appeared in: *Neural Computation* **16**(1): 1-38, 2004

Abstract

A large number of human psychophysical results have been successfully explained in recent years using Bayesian models. However, the neural implementation of such models remains largely unclear. In this paper, we show that a network architecture commonly used to model the cerebral cortex can implement Bayesian inference for an arbitrary hidden Markov model. We illustrate the approach using an orientation discrimination task and a visual motion detection task. In the case of orientation discrimination, we show that the model network can infer the posterior distribution over orientations and correctly estimate stimulus orientation in the presence of significant noise. In the case of motion detection, we show that the resulting model network exhibits direction selectivity and correctly computes the posterior probabilities over motion direction and position. When used to solve the well-known random dots motion discrimination task, the model generates responses that mimic the activities of evidence-accumulating neurons in cortical areas LIP and FEF. The framework introduced in the paper posits a new interpretation of cortical activities in terms of log posterior probabilities of stimuli occurring in the natural world.

1 Introduction

Bayesian models of perception have proved extremely useful in explaining results from a number of human psychophysical experiments (see, for example, (Knill & Richards, 1996; Rao, Olshausen, & Lewicki, 2002)). These psychophysical tasks range from inferring 3D structure from 2D images and judging depth from multiple cues to perception of motion and color ((Bloj, Kersten, & Hurlbert, 1999; Weiss, Simoncelli, & Adelson, 2002) and chapters by Mamassian *et al.* and Jacobs in (Rao et al., 2002)). The strength of the Bayesian approach lies in its ability to quantitatively model the interaction between prior knowledge and sensory evidence. Bayes rule prescribes how prior probabilities and stimulus likelihoods should be combined, allowing the responses of subjects during psychophysical tasks to be interpreted in terms of the resulting posterior distributions.

Additional support for Bayesian models comes from recent neurophysiological and psychophysical studies on visual decision-making. Carpenter and colleagues have shown that the reaction time distribution of human subjects making eye movements to one of two targets is well-explained by a model computing the log likelihood ratio of one target over the other (Carpenter & Williams, 1995). In another study, the saccadic response time distribution of monkeys could be predicted from the time taken by neural activity in area FEF to reach a fixed threshold (Hanes & Schall, 1996), suggesting that these neurons are involved in accumulating evidence (interpreted as log likelihoods) over time. Similar activity has also been reported in the primate area LIP (Shadlen & Newsome, 2001). A mathematical model based on log likelihood ratios was found to be consistent with the observed neurophysiological results (Gold & Shadlen, 2001).

Given the growing evidence for Bayesian models in perception and decision-making, an important open question is how such models could be implemented within the recurrent networks of the primate brain. Previous models of probabilistic inference in neural circuits have been based on the concept of neural tuning curves or basis functions and have typically focused on the estimation of static quantities such as stimulus position or orientation (Anderson & Van Essen, 1994; Zemel, Dayan, & Pouget, 1998; Deneve, Latham, & Pouget, 1999; Pouget, Dayan, & Zemel, 2000; Eliasmith & Anderson, 2003). Other models have relied on mean-field approximations or various forms of Gibbs sampling for perceptual inference (Hinton & Sejnowski, 1986; Dayan, Hinton,

Neal, & Zemel, 1995; Dayan & Hinton, 1996; Hinton & Ghahramani, 1997; Rao & Ballard, 1997; Rao, 1999; Rao & Ballard, 1999; Hinton & Brown, 2002).

In this paper, we describe a new model of Bayesian computation in a recurrent neural circuit. We specify how the feedforward and recurrent connections may be selected to perform Bayesian inference for arbitrary hidden Markov models (Sections 2-4). The approach is illustrated using two tasks: discriminating the orientation of a noisy visual stimulus and detecting the direction of motion of moving stimuli. In the case of orientation discrimination, we show that the model network can infer the posterior distribution over orientations and correctly estimate stimulus orientation in the presence of considerable noise (Section 5.1). In the case of motion detection, we show that the resulting model network exhibits direction selectivity and correctly computes the posterior probabilities over motion direction and position (Section 5.2). We then introduce a decision-making framework based on computing log posterior ratios from the outputs of the motion detection network. We demonstrate that for the well-known random dots motion discrimination task (Shadlen & Newsome, 2001), the decision model produces responses that are qualitatively similar to the responses of “evidence accumulating” neurons in primate brain areas LIP and FEF (Section 5.3). Some predictions of the proposed framework are explored in Section 6. We conclude by discussing several possibilities for neural encoding of log probabilities and suggest a probabilistic framework for on-line learning of synaptic parameters (Sections 7 and 8).

2 Modeling Cortical Networks

We begin by considering a commonly used neural architecture for modeling cortical response properties, namely, a recurrent network with firing-rate dynamics (see, for example, (Dayan & Abbott, 2001)). Let \mathbf{I} denote the vector of input firing rates to the network and let \mathbf{v} represent the output firing rates of the recurrently connected neurons in the network. Let \mathbf{W} represent the feedforward synaptic weight matrix and \mathbf{M} the recurrent weight matrix. The following equation describes the dynamics of the network:

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{W}\mathbf{I} + \mathbf{M}\mathbf{v} \quad (1)$$

where τ is a time constant. The equation can be written in a discrete form as follows:

$$v_i(t+1) = v_i(t) + \epsilon(-v_i(t) + \mathbf{w}_i \mathbf{I}(t) + \sum_j M_{ij} v_j(t)) \quad (2)$$

where ϵ is the integration rate, v_i is the i th component of the vector \mathbf{v} , \mathbf{w}_i is the i th row of the matrix \mathbf{W} , and M_{ij} is the element of \mathbf{M} in the i th row and j th column. The above equation can be rewritten as:

$$v_i(t+1) = \epsilon \mathbf{w}_i \mathbf{I}(t) + \sum_j m_{ij} v_j(t) \quad (3)$$

where $m_{ij} = \epsilon M_{ij}$ for $i \neq j$ and $m_{ii} = 1 + \epsilon(M_{ii} - 1)$. How can Bayesian computation be achieved using the dynamics given by Equation 3? We approach this problem by considering Bayesian inference in a simple hidden Markov model.

3 Bayesian Computation in a Cortical Network

Let $\theta_1, \dots, \theta_N$ be the hidden states of a hidden Markov model. Let $\theta(t)$ represent the hidden state at time t with transition probabilities denoted by $P(\theta(t) = \theta_i | \theta(t-1) = \theta_j)$ for $i, j = 1 \dots N$. Let $\mathbf{I}(t)$ be the observable output governed by the probabilities $P(\mathbf{I}(t) | \theta(t))$. Then, the posterior probability of being in state θ_i at time t is given by Bayes rule:

$$P(\theta(t) = \theta_i | \mathbf{I}(t), \dots, \mathbf{I}(1)) = k P(\mathbf{I}(t) | \theta(t) = \theta_i) P(\theta(t) = \theta_i | \mathbf{I}(t-1), \dots, \mathbf{I}(1)) \quad (4)$$

where k is a normalization constant. The equation can be made recursive with respect to the posterior probabilities as follows:

$$P(\theta_i^t | \mathbf{I}(t), \dots, \mathbf{I}(1)) = k P(\mathbf{I}(t) | \theta_i^t) \sum_j P(\theta_i^t | \theta_j^{t-1}) P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1)) \quad (5)$$

where we have written “ $\theta(t) = \theta_i$ ” as θ_i^t for notational convenience. This equation can be written in the log domain as:

$$\begin{aligned} \log P(\theta_i^t | \mathbf{I}(t), \dots, \mathbf{I}(1)) &= \log P(\mathbf{I}(t) | \theta_i^t) + \log k + \\ &\quad \log \left[\sum_j P(\theta_i^t | \theta_j^{t-1}) P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1)) \right] \end{aligned} \quad (6)$$

A recurrent network governed by Equation 3 can implement Equation 6 if:

$$v_i(t+1) = \log P(\theta_i^t | \mathbf{I}(t), \dots, \mathbf{I}(1)) \quad (7)$$

$$\epsilon \mathbf{w}_i \mathbf{I}(t) = \log P(\mathbf{I}(t) | \theta_i^t) \quad (8)$$

$$\sum_j m_{ij} v_j(t) = \log \left[\sum_j P(\theta_i^t | \theta_j^{t-1}) P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1)) \right] \quad (9)$$

with the normalization term $\log k$ being added after each update. In other words, the network activities are updated according to:

$$v_i(t+1) = \epsilon \mathbf{w}_i \mathbf{I}(t) + \sum_j m_{ij} v_j(t) - \log \sum_j e^{u_j(t)} \quad (10)$$

where $u_j(t) = \epsilon \mathbf{w}_i \mathbf{I}(t) + \sum_j m_{ij} v_j(t)$. The negative log term, which implements divisive normalization in the log domain, can be interpreted as a form of global recurrent inhibition.¹

In Equation 10, the log likelihoods can be computed using linear filters $\mathbf{F}(\theta_i)$ ($= \epsilon \mathbf{w}_i$). For example, $\mathbf{F}(\theta_i)$ could represent a spatially localized Gaussian or an oriented Gabor filter when the inputs $\mathbf{I}(t)$ are images. Note that as formulated above, the updating of probabilities is directly tied to the time constant of neurons in the network as specified by the parameter τ in Equation 1 (or ϵ in Equation 2). However, it may be possible to achieve longer integration time constants by combining slow synapses (for example, NMDA synapses) with relatively strong recurrent excitation (see, for example, (Seung, 1996; Wang, 2001)).

A particularly challenging problem is to pick recurrent weights m_{ij} such that Equation 9 holds true (the alternative of learning such weights is addressed in Section 8.1). For Equation 9 to hold true, we need to approximate a *log-sum* with a *sum-of-logs*. We tackled this problem by generating a set of random probabilities $x_j(t)$ for $t = 1, \dots, T$ and finding a set of weights m_{ij} that satisfy:

$$\sum_j m_{ij} \log x_j(t) \approx \log \left[\sum_j P(\theta_i^t | \theta_j^{t-1}) x_j(t) \right] \quad (11)$$

for all i and t (as an aside, note that a similar approximation can be used to compute a set of recurrent inhibitory weights for implementing the negative log term in Equation 10). We examine this sum-of-logs approximation in more detail in the next section.

¹If normalization is omitted, the network outputs can be interpreted as representing log joint probabilities i.e. $v_i(t+1) = \log P(\theta_i^t, \mathbf{I}(t), \dots, \mathbf{I}(1))$. However, we have found that the lack of normalization makes such a network prone to instabilities.

4 Approximating Transition Probabilities using Recurrent Weights

A set of recurrent weights m_{ij} can be obtained for any given set of transition probabilities $P(\theta_i^t | \theta_j^{t-1})$ by using the standard pseudoinverse method. Let A represent the matrix of recurrent weights i.e. $A_{ij} = m_{ij}$, and let L represent a matrix of log probabilities, the j th element in the t th column representing $\log x_j(t)$ for $t = 1, \dots, T$. Let B represent the matrix of log sums i.e. $B_{it} = \log \left[\sum_j P(\theta_i^t | \theta_j^{t-1}) x_j(t) \right]$. To minimize the squared error in Equation 11 with respect to the recurrent weights m_{ij} ($= A_{ij}$), we need to solve the equation:

$$AL = B \quad (12)$$

Multiplying both sides by the pseudoinverse of L , we obtain the following expression for the recurrent weight matrix:

$$A = BL^T(LL^T)^{-1} \quad (13)$$

We first tested this approximation as a function of the transition probabilities $P(\theta_i^t | \theta_j^{t-1})$ for a fixed set of uniformly random probabilities $x_j(t)$. The matrix of transition probabilities was chosen to be a sparse random matrix. The degree of sparseness was varied by choosing different values for the density of the matrix, defined as the fraction of nonzero elements. Approximation accuracy was measured in terms of the absolute value of the error between the left and right sides of Equation 11, averaged over both time and neurons. As shown in Figure 1, for any fixed degree of sparseness in the transition probability matrix, the average error decreases as a function of the number of neurons in the recurrent network. Furthermore, the overall errors also systematically decrease as the density of the random transition probability matrix is increased (different curves in the graph). The approximation failed only in one case, when the density was 0.1 and the number of neurons was 25, due to the occurrence of a $\log 0$ term in B (see discussion below).

We also tested the accuracy of Equation 11 as a function of the conditional probabilities $x_j(t) = P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1))$ for a fixed set of uniformly random transition probabilities $P(\theta_i^t | \theta_j^{t-1})$. The matrix of conditional probabilities was chosen to be a sparse random matrix (rows denoting indices of neurons and columns representing time steps). The degree of sparseness was varied as before by choosing different values

for the density of the matrix. It is clear that the approximation fails if any of the $x_j(t)$ equals exactly 0 since the corresponding entry $\log x_j(t)$ in L will be undefined. We therefore added a small arbitrary positive value to all entries of the random matrix before renormalizing it. The smallest value in this normalized matrix may be regarded as the smallest probability that can be represented by neurons in the network. This ensures that $x_j(t)$ is never zero while allowing us to test the stability of the approximation for small values for $x_j(t)$. We define the density of the random matrix to be the fraction of entries that are different from the smallest value obtained after renormalization.

As shown in Figure 2, for any fixed degree of sparseness of the conditional probability matrix, the average error decreases as the number of neurons is increased. The error curves also show a gradual decrease as the density of the random conditional probability matrix is increased (different curves in the graph). The approximation failed in two cases: when the number of neurons was 75 or less and the density was 0.1, and when the number of neurons was 25 and the density was 0.25. The approximation error was consistently below 0.1 for all densities above 0.5 and number of neurons greater than 100.

In summary, the approximation of transition probabilities using recurrent weights based on approximating a log sum with a sum of logs was found to be remarkably robust to a range of choices for both the transition probabilities and the conditional probabilities $P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1))$. The only cases where the approximation failed were when two conditions both held true: (a) the number of neurons used was very small (25 to 75) and (b) a majority of the transition probabilities are equal to zero or a majority of the conditional probabilities are near zero. The first of these conditions is not a major concern given the typical number of neurons in cortical circuits. For the second condition, we only need assume that the conditional probabilities can take on values between a small positive number ϵ and 1, where $\log \epsilon$ can be related to the neuron's background firing rate. This seems a reasonable assumption given that cortical neurons in vivo typically have a non-zero background firing rate. Finally, it should be noted that the approximation above is based on assuming a simple linear recurrent model of a cortical network. The addition of a more realistic non-linear recurrent function in Equation 1 can be expected to yield more flexible and accurate ways of approximating the log sum in Equation 9.

5 Results

5.1 Example 1: Estimating Orientation from Noisy Images

To illustrate the approach, we first present results from a simple visual discrimination task involving inference of stimulus orientation from a sequence of noisy images containing an oriented stimulus. The underlying hidden Markov model uses a set of discrete states $\theta_1, \theta_2, \dots, \theta_M$ that sample the space of possible orientations of an elongated bar in the external world. During each trial, the state is assumed to be fixed i.e.

$$P(\theta_i^t | \theta_j^{t-1}) = 1 \quad \text{if } i = j \quad (14)$$

$$= 0 \quad \text{otherwise} \quad (15)$$

Finally, the likelihoods are given by the product of the image vector $\mathbf{I}(t)$ with a set of oriented filters $\mathbf{F}(\theta_i)$ (see Figure 3A) corresponding to the orientations represented by the states $\theta_1, \theta_2, \dots, \theta_M$:

$$\log P(\mathbf{I}(t) | \theta_i^t) = \epsilon \mathbf{w}_i \mathbf{I}(t) = \mathbf{F}(\theta_i) \mathbf{I}(t) \quad (16)$$

For each trial, the sequence of input images was generated by adding zero-mean Gaussian white noise of fixed variance to an image containing a bright oriented bar against a dark background (Figure 3B).

The goal of the network was to compute the posterior probabilities of each orientation given an input sequence of noisy images containing a single oriented bar. Clearly, the likelihood values, given by Equation 16, can vary significantly over time (see Figure 4, middle row), the amount of variation being determined by the variance of the additive Gaussian noise. The recurrent activity of the network however allows a stable estimate of the underlying stimulus orientation to be computed across time (Figure 4, bottom row). This estimate can be obtained using the maximum a posteriori (MAP) method: the estimated orientation at any time is given by the preferred orientation of the neuron with the maximum activity i.e. the orientation with the maximum posterior probability. We tested the MAP estimation ability of the network as a function of noise variance for a stimulus set containing 36 orientations (0 to 180 in 10 degree steps). For these experiments, the network was run for a fixed number of time steps (250 time steps) and the MAP estimate was computed from the final set of activities. As shown in Figure 5, the classification rate of the network using the MAP method is

high (above 80%) for zero-mean Gaussian noise of standard deviations less than 1 and remains above chance (1/36) for standard deviations up to 4 (i.e. noise variances up to 16).

5.2 Example 2: Detecting Visual Motion

To illustrate more clearly the dynamical properties of the model, we examine the application of the proposed approach to the problem of visual motion detection. A prominent property of visual cortical cells in areas such as V1 and MT is selectivity to the direction of visual motion. In this section, we interpret the activity of such cells as representing the log posterior probability of stimulus motion in a particular direction at a spatial location, given a series of input images.

For simplicity, we focus on the case of 1-D motion in an image consisting of X pixels with two possible motion directions: leftward (L) or rightward (R). Let the state θ_{ij} represent motion at a spatial location i in direction $j \in \{L, R\}$. Consider a network of N neurons, each representing a particular state θ_{ij} (Figure 6A). The feedforward weights are Gaussians i.e. $\mathbf{F}(\theta_{iR}) = \mathbf{F}(\theta_{iL}) = \mathbf{F}(\theta_i) = \text{Gaussian centered at location } i$ with a standard deviation σ . Figure 6B depicts the feedforward weights for a network of 30 neurons, 15 encoding leftward and 15 encoding rightward motion.

To detect motion, the transition probabilities $P(\theta_{ij}|\theta_{kl})$ must be selected to reflect both the direction of motion and speed of the moving stimulus. For the present study, the transition probabilities for rightward motion from the state θ_{kR} (i.e. $P(\theta_{iR}|\theta_{kR})$) were set according to a Gaussian centered at location $k + x$, where x is a parameter determined by stimulus speed. The transition probabilities for leftward motion from the state θ_{kL} were likewise set to Gaussian values centered at $k - x$. The transition probabilities from states near the two boundaries ($i = 1$ and $i = X$) were chosen to be uniformly random values. Figure 6C shows the matrix of transition probabilities. Given these transition probabilities, the recurrent weights m_{ij} can be computed using Equation 13 (see Figure 6D). Figure 7 shows that for the given set of transition probabilities, the log-sum in Equation 11 can indeed be approximated quite accurately by a sum-of-logs using the weights m_{ij} .

5.2.1 Bayesian Estimation of Visual Motion

Figure 8 shows the output of the network in the middle of a sequence of input images depicting a bar moving either leftward or rightward. As shown in the figure, for a leftward moving bar at a particular location i , the highest network output is for the neuron representing location i and direction L , while for a rightward moving bar, the neuron representing location i and direction R has the highest output. The output firing rates were computed from the log posteriors v_i using a simple linear encoding model: $f_i = [c \cdot v_i + m]^+$ where c is a positive constant ($= 12$ for this plot), m is the maximum firing rate of the neuron ($= 100$ in this example), and $+$ denotes rectification (see Section 7.1 for more details). Note that even though the log likelihoods are the same for leftward and rightward moving inputs, the asymmetric recurrent weights (which represent the transition probabilities) allow the network to distinguish between leftward and rightward moving stimuli. The plot of posteriors in Figure 8 shows that the network correctly computes posterior probabilities close to 1 for the states θ_{iL} and θ_{iR} for leftward and rightward motion respectively at location i .

5.2.2 Encoding Multiple Stimuli

An important question that needs to be addressed is whether the network can encode uncertainties associated with multiple stimuli occurring in the same image. For example, can the motion detection network represent two or more bars moving simultaneously in the image? We begin by noting that the definition of “multiple stimuli” is closely related to the notion of “state.” Recall that our underlying probabilistic model (the hidden Markov model) assumes that the observed world can be in only one of M different states. Thus, any input containing “multiple stimuli” (such as two moving bars) will be interpreted in terms of probabilities for each of the M states (a single moving bar at each location).

We tested the ability of the motion detection network to encode multiple stimuli by examining its responses to an image sequence containing two simultaneously moving bars under two conditions: (a) both bars moving in the same direction and (b) the bars moving in opposite directions. The results are shown in Figures 9 and 10 respectively. As seen in these figures, the network is able to represent the two simultaneously moving stimuli under both conditions. Note the two peaks in activity representing the

location and direction of motion of the two moving bars. As expected from the underlying probabilistic model, the network computes a posterior probability close to 0.5 for each of the two peaks, although the exact values fluctuate depending on the locations of the two bars. The network thus interprets the two simultaneously moving stimuli as providing evidence for two of the states it can represent, each state representing motion at a particular location and direction. Such a representation could in turn be fed to a higher level network, whose states could represent frequently occurring combinations of lower level states, in a manner reminiscent of the hierarchical organization of sensory and motor cortical areas (Felleman & Van Essen, 1991). We discuss possible hierarchical extensions of the model in Section 8.2.

5.2.3 Encoding Multiple Velocities

The network described above is designed to estimate the posterior probability of a stimulus moving at a fixed velocity as given by the transition probability matrix (Figure 6C). How does the network respond to velocities other than the ones encoded explicitly in the transition probability matrix?

We investigated the question of velocity encoding by using two separate networks based on two transition probability matrices. The recurrent connections in the first network encoded a speed of 1 pixel/time step (Figure 11A) while the those in the second network encoded a speed of 4 pixels/time step (Figure 11C). Each network was exposed to a range of velocities, ranging from 0.5-5 pixels/time step in increments of 0.5. The resulting responses of a neuron located in the middle of each network are shown in Figure 11B and 11D respectively. The bars depict the peak response of the neuron during stimulus motion at a particular velocity in the neuron's preferred and non-preferred (null) direction. The response is plotted as a posterior probability for convenience (see Equation 7). As evident from the figure, the neurons exhibit a relatively broad tuning curve centered on their respective preferred velocities (1 and 4 pixels/time step respectively).

These results suggest that the model can encode multiple velocities using a tuning curve even though the recurrent connections are optimized for a single velocity. In other words, the posterior distribution over velocities can be represented by a fixed number of networks optimized for a fixed set of preferred velocities, in much the same way as a fixed number of neurons can sample the continuous space of orientation, position, or

any other continuous state variable. In addition, as discussed in Section 8.1, synaptic learning rules could be used to allow the networks to encode the most frequently occurring stimulus velocities, thereby allowing the state space of velocities to be tiled in an adaptive manner.

5.3 Example 3: Bayesian Decision-Making in a Random Dots Task

To establish a connection to behavioral data, we consider the well-known random dots motion discrimination task (see, for example, (Shadlen & Newsome, 2001)). The stimulus consists of an image sequence showing a group of moving dots, a fixed fraction of which are randomly selected at each frame and moved in a fixed direction (for example, either left or right). The rest of the dots are moved in random directions. The fraction of dots moving in the same direction is called the *coherence* of the stimulus. Figure 12A depicts the stimulus for two different levels of coherence. The task is to decide the direction of motion of the coherently moving dots for a given input sequence. A wealth of data exists on the psychophysical performance of humans and monkeys as well as the neural responses in brain areas such as MT and LIP in monkeys performing the task (see (Shadlen & Newsome, 2001) and references therein). Our goal is to explore the extent to which the proposed model can explain the existing data for this task and make testable predictions.

The motion detection network in the previous section can be used to decide the direction of coherent motion by computing the posterior probabilities for leftward and rightward motion, given the input images. These probabilities can be computed by marginalizing the posterior distribution computed by the neurons for leftward (L) and rightward (R) motion over all spatial positions i :

$$P(L|\mathbf{I}(t), \dots, \mathbf{I}(1)) = \sum_i P(\theta_{iL}|\mathbf{I}(t), \dots, \mathbf{I}(1)) \quad (17)$$

$$P(R|\mathbf{I}(t), \dots, \mathbf{I}(1)) = \sum_i P(\theta_{iR}|\mathbf{I}(t), \dots, \mathbf{I}(1)) \quad (18)$$

Note that the log of these marginal probabilities can also be computed directly from the log posteriors (i.e. outputs of the neurons) using the log sum approximation method described in Sections 3 and 4. The log posterior ratio $r(t)$ of leftward over rightward motion can be used to decide the direction of motion:

$$r(t) = \log P(L|\mathbf{I}(t), \dots, \mathbf{I}(1)) - \log P(R|\mathbf{I}(t), \dots, \mathbf{I}(1)) \quad (19)$$

$$= \log \frac{P(L|\mathbf{I}(t), \dots, \mathbf{I}(1))}{P(R|\mathbf{I}(t), \dots, \mathbf{I}(1))} \quad (20)$$

If $r(t) > 0$, the evidence seen so far favors leftward motion and vice versa for $r(t) < 0$. The instantaneous ratio $r(t)$ is susceptible to rapid fluctuations due to the noisy stimulus. We therefore use the following decision variable $d_L(t)$ to track the running average of the log posterior ratio of L over R:

$$d_L(t + 1) = d_L(t) + \alpha(r(t) - d_L(t)) \quad (21)$$

and likewise for $d_R(t)$ (the parameter α is between 0 and 1). We assume that the decision variables are computed by a separate set of “decision neurons” that receive inputs from the motion detection network. These neurons are once again leaky-integrator neurons as described by Equation 21, with the driving inputs $r(t)$ being determined by inhibition between the summed inputs from the two chains in the motion detection network (as in Equation 19). The output of the model is “L” if $d_L(t) > c$ and “R” if $d_R(t) > c$, where c is a “confidence threshold” that depends on task constraints (for example, accuracy versus speed requirements) (Reddi & Carpenter, 2000).

Figures 12B and 12C show the responses of the two decision neurons over time for two different directions of motion and two levels of coherence. Besides correctly computing the direction of coherent motion in each case, the model also responds faster when the stimulus has higher coherence. This phenomenon can be appreciated more clearly in Figure 12D, which predicts progressively shorter reaction times for increasingly coherent stimuli (dotted arrows).

5.3.1 Comparison to Neurophysiological Data

The relationship between faster rates of evidence accumulation and shorter reaction times has received experimental support from a number of studies. Figure 12E shows the activity of a neuron in the frontal eye fields (FEF) for fast, medium, and slow responses to a visual target (Schall & Hanes, 1998; Schall & Thompson, 1999). Schall and collaborators have shown that the distribution of monkey response times can be reproduced using the time taken by neural activity in FEF to reach a fixed threshold (Hanes & Schall, 1996). A similar rise-to-threshold model by Carpenter and colleagues has received strong support in human psychophysical experiments that manipulate the

prior probabilities of targets (Carpenter & Williams, 1995) and the urgency of the task (Reddi & Carpenter, 2000) (see Section 6.2).

In the case of the random dots task, Shadlen and collaborators have shown that in primates, one of the cortical areas involved in making the decision regarding coherent motion direction is area LIP. The activities of many neurons in this area progressively increase during the motion viewing period, with faster rates of rise for more coherent stimuli (see Figure 12F) (Roitman & Shadlen, 2002). This behavior is similar to the responses of “decision neurons” in the model (Figures 12B–D), suggesting that the outputs of the recorded LIP neurons could be interpreted as representing the log posterior ratio of one task alternative over another (see (Carpenter & Williams, 1995; Gold & Shadlen, 2001) for related suggestions).

5.3.2 Performance of the Network

We also examined the performance of the network as a function of motion coherence for different values of the decision threshold. As shown in Figure 13, the results obtained show two trends that are qualitatively similar to human and monkey psychophysical performance on this task. First, for any fixed decision threshold T , the accuracy rate (% correct responses) increases from chance levels (50% correct) to progressively higher values (up to 100% correct for higher thresholds) when the motion coherence is increased from 0 to 1. These graphs are qualitatively similar to psychometric performance curves obtained in human and monkey experiments (see, for example, (Britten, Shadlen, Newsome, & Movshon, 1992)). We did not attempt to quantitatively fit the model curves to psychophysical data because the model currently operates only on 1D images. A second trend is observed when the decision threshold T is varied. As the threshold is increased, the accuracy also increases for each coherence value, revealing a set of curves that asymptote around 95-100% accuracy for large enough thresholds. However, as expected, larger thresholds also result in longer reaction times, suggesting a correlate in the model for the type of speed-accuracy tradeoffs typically seen in psychophysical discrimination and search experiments. The dependence of accuracy on decision threshold suggests a model for handling task urgency: faster decisions can be made by lowering the decision threshold, albeit at the expense of some accuracy (as illustrated in Figure 13). Predictions of such a model for incorporating urgency in the decision-making process are explored in Section 6.2.

6 Model Predictions

6.1 Effect of Multiple Stimuli in a Receptive Field

A first prediction of the model arises from the log probability interpretation of firing rates and the effects of normalization in a probabilistic system. In a network of neurons where each neuron encodes the probability of a particular state, the sum of the probabilities should sum to one. Thus, if an input contains multiple stimuli consistent with multiple states, the probabilities for each state would be appropriately reduced to reflect probability normalization. The current model predicts that one should see a concomitant logarithmic decrease in the firing rates as illustrated in the example below.

Consider the simplest case of two static stimuli in the receptive field, reflecting two different states encoded by two neurons (say, 1 and 2). Let v_1 and v_2 be the outputs (log posteriors) of neurons 1 and 2 respectively when their preferred stimulus is shown alone in the receptive field. In the ideal noiseless case, both log posteriors would be zero. Suppose both stimuli are now shown simultaneously. The model predicts that the new outputs v'_1 and v'_2 should obey the relation $e^{v'_1} + e^{v'_2} = 1$. If both states are equally likely, $e^{v'_1} = 0.5$ i.e. $v'_1 = -0.69$, a non-linear decrease from the initial value of zero. To see the impact on firing rates, consider the case where neuronal firing rate f is proportional to the log posterior, i.e., $f = [c \cdot v + m]^+$, where c is a positive constant and m is the maximum firing rate (see Section 7.1 for a discussion). Then, the new firing rate f'_1 for neuron 1 is related to the old firing rate $f_1 (= m)$ by $f'_1 = -0.69c + m = f_1 - 0.69c$. Note that this is different from a simple halving of the single stimulus firing rate, as might be predicted from a linear model. Thus, the model predicts that in the presence of multiple stimuli, the firing rate of a cell encoding for one of the stimuli will be non-linearly reduced in such a way that the sum of exponentials of the firing rates in the network approximates 1. Evidence for non-linear reductions in firing rates for multiple stimuli in the receptive field comes from some experiments in V1 (DeAngelis, Robson, Ohzawa, & Freeman, 1992) and other attention-related experiments in higher cortical areas (Reynolds, Chelazzi, & Desimone, 1999). The existing data is however insufficient to quantitatively validate or falsify the above prediction. More sophisticated experiments involving multi-unit recordings may be needed to shed light on this prediction derived from the log probability hypothesis of neural coding.

6.2 Effect of Changing Priors and Task Urgency

The strength of a Bayesian model lies in its ability to predict the effects of varying prior knowledge or various types of biases in a given task. We explored the consequences of varying two types of biases in the motion discrimination task: (a) changing the prior probability of a particular motion direction (L or R) and (b) lowering the decision threshold to favor speed instead of accuracy.

Changing the prior probability for a particular motion direction amounts to initializing the decision neuron activities d_L and d_R to the appropriate log prior ratios:

$$d_L(0) = \log P(L) - \log P(R) \quad (22)$$

$$d_R(0) = \log P(R) - \log P(L) \quad (23)$$

where $P(L)$ and $P(R)$ denote the prior probabilities of leftward and rightward motion respectively. Note that for the equiprobable case, these variables get initialized to 0 as in the simulations presented thus far. We examined the distribution of reaction times obtained as a result of assuming a slightly higher prior probability for L compared to R (0.51 versus 0.49). Figure 14A compares this distribution (shown on the right) to the distribution obtained for the unbiased (equiprobable) case (shown on the left). Both distributions shown are for trials where the direction of motion was L. The model predicts that the entire distribution for a particular task choice should shift leftwards (towards shorter reaction times) when the subject assumes a higher prior for that particular choice.

Similarly, when time is at a premium, a subject may opt to lower the decision threshold to reach decisions quickly, albeit at the risk of making some incorrect decisions. Such an urgency constraint was simulated in the model by lowering the decision threshold to half the original value, keeping the stimulus coherence constant (60% in this case). As shown in Figure 14B, the model again predicts a leftward shift of the entire reaction time histogram, corresponding to faster (but sometimes inaccurate) responses.

Although these predictions are yet to be tested in the context of the motion discrimination task (see (Mazurek, Ditterich, Palmer, & Shadlen, 2001) for some preliminary results), psychophysical experiments by Carpenter and colleagues lend some support to the model's predictions. In these experiments (Carpenter & Williams, 1995), a human subject was required to make an eye movement to a dim target that appeared randomly

to the left or right of a fixation point after a random delay of 0.5–1.5 seconds. The latency distributions of the eye movements were recorded as a function of different prior probabilities for one of the targets. As predicted by the model for the motion discrimination task (Figure 14A), the response latency histogram was observed to shift leftward for increasing prior probabilities.

In a second set of experiments, Reddi and Carpenter examined the effects of imposing urgency constraints on the eye movement task (Reddi & Carpenter, 2000). They asked subjects to respond as rapidly as possible and to worry less about the accuracy of their response. Once again, the latency histogram shifted leftwards, similar to the model simulation results in Figure 14B. Furthermore, in a remarkable follow-up experiment, Reddi and Carpenter showed that this leftward shift in the histogram due to urgency could be countered by a corresponding rightward shift due to lower prior probability for one of the targets, resulting in no overall shift in the latency histogram (Reddi & Carpenter, 2000). This provides strong evidence for a rise-to-threshold model of decision-making, where the threshold is chosen according to the constraints of the task at hand.

7 Discussion

In this section, we address the problem of how neural activity may encode log probability values, which are non-positive. We also discuss the benefits of probabilistic inference in the log domain for neural systems and conclude the section by summarizing related work in probabilistic neural models and probabilistic decision-making.

7.1 Neural Encoding of Log Probabilities

The model introduced above assumes that the outputs of neurons in a recurrent network reflect log posterior probabilities of a particular state, given a set of inputs. However, log probabilities take on non-positive values between $-\infty$ and 0, raising two important questions: (1) How are these non-positive values encoded using firing rates or a spike-based code? (2) How do these codes capture the precision and range of probability values that can be encoded? We address these issues below in the context of rate-based and spike-based encoding models.

A simple encoding model is to assume that firing rates are linearly related to the log posteriors encoded by the neurons. Thus, if v_i denotes the log posterior encoded by neuron i , then the firing rate of the neuron is given by:

$$f_i = [c \cdot v_i + m]^+ \quad (24)$$

where c is a positive gain factor, m is the maximum firing rate of the neuron, and $+$ denotes rectification ($[x]^+ = x$ if $x > 0$ and 0 otherwise). Note that the firing rate f_i attains the maximum value m when the log posterior $v_i = 0$ i.e. the posterior probability $P(\theta_i^t | \mathbf{I}(t), \dots, \mathbf{I}(1)) = 1$. Likewise, f_i attains its minimal value of 0 for posteriors below $e^{-m/c}$. Thus, the precision and range of probability values that can be encoded in the firing rate are governed by both m and c . Since e^{-x} quickly becomes negligible (e.g. $e^{-10} = 0.000045$), for typical values of m in the range 100-200 Hz, values for c in the range 10-20 would allow the useful range of probability values to be accurately encoded in the firing rate. These firing rates can be easily decoded when received as synaptic inputs using the inverse relationship: $v_i = (f_i - m)/c$.

A second straightforward but counterintuitive encoding model is to assume that $f_i = -c \cdot v_i$, up to some maximum value m .² In this model, a high firing-rate implies a low posterior probability i.e. an unexpected or novel stimulus. Such an encoding model allows neurons to be interpreted as novelty detectors. Thus, the suppression of neural responses in some cortical areas due to the addition of contextual information (e.g. surround effects in V1 (Knierim & Essen, 1992; Zipser, Lamme, & Schiller, 1996)) or due to increasing familiarity with a stimulus (e.g. response reduction in IT due to stimulus repetition (Miller, Li, & Desimone, 1991)) can be interpreted as increases in posterior probabilities of the state encoded by the recorded neurons. Such a model of response suppression is similar in spirit to predictive coding models (Srinivasan, Laughlin, & Dubs, 1982; Rao & Ballard, 1999) but differs from these earlier models in assuming that the responses represent log posterior probabilities rather than prediction errors.

A spike-based encoding model can be obtained by interpreting the leaky integrator equation (Equation 1) in terms of the membrane potential of a neuron rather than its firing rate. Such an interpretation is consistent with the traditional RC circuit-based

²A different model based also on a negative log encoding scheme has been suggested by Barlow (1969, 1972).

model of membrane potential dynamics (see, for example, (Koch, 1999)). We assume that the membrane potential V_i^m is proportional to the log posterior v_i :

$$V_i^m = k \cdot v_i + T \quad (25)$$

where k is a constant gain factor and T is a constant nonnegative offset. The dynamics of the neuron’s membrane potential is then given by:

$$\frac{dV_i^m}{dt} = k \frac{dv_i}{dt} \quad (26)$$

where $\frac{dv_i}{dt}$ is computed as described in Sections 2 and 3. Thus, in this model, changes in the membrane potential due to synaptic inputs are proportional to changes in the log posterior being represented by the neuron. More interestingly, if T is regarded as analogous to the spiking threshold, one can define the probability of neuron i spiking at time $t + 1$ as:

$$P(\text{spike}_i(t+1)|V_i^m(t+1)) = e^{(V_i^m(t+1)-T)/k} = e^{v_i(t+1)} = P(\theta_i^t|\mathbf{I}(t), \dots, \mathbf{I}(1)). \quad (27)$$

Such a model for spiking can be viewed as a variant of the noisy threshold model, where the cell spikes if its potential exceeds a noisy threshold. As defined above, the spiking probability of a neuron becomes exactly equal to the posterior probability of the neuron’s preferred state θ_i given the inputs (see also (Anastasio, Patton, & Belkacem-Boussaid, 2000; Hoyer & Hyvärinen, 2003)). This provides a new interpretation of traditional neuronal spiking probabilities computed from a post stimulus time histogram (PSTH). An intriguing open question, given such an encoding model, is how these spike probabilities are decoded “on-line” from input spike trains by synapses and converted to the log domain to influence the membrane potential as in Equation 26. We hope to explore this question in future studies.

7.2 Benefits of Probabilistic Inference in the Log Domain

A valid question to ask is why a neural circuit should encode probabilities in the log domain in the first place. Indeed, it has been suggested that multiplicative interactions between inputs may occur in dendrites of cortical cells (Mel, 1993), which could perhaps allow Equation 5 to be directly implemented in a recurrent circuit (cf. (Bridle, 1990)). However, there are several reasons why representing probabilities in the log domain could be beneficial to a neural system:

- Neurons have a limited dynamic range. A logarithmic transformation allows the most useful range of probabilities to be represented by a small number of activity levels (see discussion above on neural encoding of log probabilities).
- Implementing probabilistic inference in the log domain allows the use of addition and subtraction instead of multiplication and division. The former operations are readily implemented through excitation and inhibition in neural circuits while the latter are typically much harder to implement neurally. As shown earlier in this paper, a standard leaky-integrator model suffices to implement probabilistic inference for a hidden Markov model if performed in the log domain.
- There is growing evidence that the brain uses quantities such as log likelihood ratios during decision-making (see Related Work below). These quantities can be readily computed if neural circuits are already representing probabilities in the log domain.

On the other hand, it is clear that representing probabilities in the log domain also makes certain operations such as addition or convolution harder to implement. Specific examples include computing the OR of two distributions or adding noise to/convolving a given distribution with noise. To implement such operations, neural circuits operating in the log domain would need to resort to approximations, such as the log-sum approximation for addition discussed in Section 4.

7.3 Neural Encoding of Priors

An important question for Bayesian models of brain function is how prior knowledge about the world can be represented and used during neural computation. In the model discussed in this paper, “priors” influence computation in two ways:

1. Long-term prior knowledge about the statistical regularities of the environment is stored within the recurrent and feedforward connections in the network. Such knowledge could be acquired through evolution, learning, or some combination of these two strategies. For example, the transition probabilities stored in the recurrent connections of the motion detection network capture prior knowledge about the expected trajectory of a moving stimulus at each location. This stored knowledge could be changed by adapting the recurrent weights if the statistics

of moving stimuli suddenly changes in the organism’s environment. Similarly, the likelihood function stored in the feedforward connections captures properties of the physical world and can be adapted according to the statistics of the inputs. The study of such stored “priors” constitutes a major research direction in Bayesian psychophysics (e.g. (Bloj et al., 1999; Weiss et al., 2002) and chapters in (Rao et al., 2002)).

2. At shorter time scales, recurrent activity provides “priors” for expected inputs based on the history of past inputs. These priors are then combined with the feedforward input likelihoods to produce the posterior probability distribution (cf. Equations 6 and 10). Note that the priors provided by the recurrent activity at each time step reflect the long-term prior knowledge about the environment (transition probabilities) stored in the recurrent connections, as discussed in 1 above.

7.4 Related Work

This paper makes contributions in two related areas: neural models of probabilistic inference and models of probabilistic decision-making. We review below previous work in these two areas with the aim of comparing and contrasting our approach with these previous approaches.

7.4.1 Neural Models of Probabilistic Inference

Perhaps the earliest neural model of probabilistic inference is the Boltzmann machine (Hinton & Sejnowski, 1983, 1986), a stochastic network of binary units. Inference in the network proceeds by randomly selecting a unit at each time step and setting its activity equal to 1 according to a probability given by a sigmoidal function of a weighted sum of its inputs. This update procedure, also known as Gibbs sampling, allows the network to converge to a set of activities characterized by the Boltzmann distribution (see, for example, (Dayan & Abbott, 2001) for details). The current model differs from the Boltzmann machine in both the underlying generative model as well as the inference mechanism. Unlike the Boltzmann machine, the present model utilizes a generative model with explicit dynamics in the state space and can therefore model probabilistic transitions between states. For inference, the model explicitly operates

on probabilities (in the log domain), thereby avoiding the need for sampling-based inference techniques which can be notoriously slow. These features also differentiate the proposed approach from other more recent probabilistic neural network models such as the Helmholtz machine (Dayan et al., 1995; Lewicki & Sejnowski, 1997; Hinton & Ghahramani, 1997).

Bridle first suggested that statistical inference for hidden Markov models could be implemented using a modification of the standard recurrent neural network (Bridle, 1990). In his model, known as an “Alpha-Net,” the computation of the likelihood for an input sequence is performed directly using an analog of Equation 5. Thus, Bridle’s network requires a multiplicative interaction of input likelihoods with the recurrent activity. As mentioned above, the model we have proposed performs inference in the log domain, thereby requiring only additive interactions, which are more easily implemented in neural circuitry. In addition, experimental data from human and monkey decision-making tasks have suggested an interpretation of neural activities in terms of log probabilities, supporting a model of inference in the log domain. The log domain implementation however necessitates the approximation of the transition probabilities using Equation 13, an approximation that is avoided if multiplicative interactions are allowed (as in Bridle’s approach).

A second set of probabilistic neural models stems from efforts to understand encoding and decoding of information from populations of neurons. One class of approaches uses basis functions to represent probability distributions within neuronal ensembles (Anderson & Van Essen, 1994; Anderson, 1995; Deneve & Pouget, 2001; Eliasmith & Anderson, 2003). In this approach, a distribution $P(\mathbf{x})$ over stimuli \mathbf{x} is represented using a linear combination of basis functions (kernels):

$$P(\mathbf{x}) = \sum_i r_i b_i(\mathbf{x}) \quad (28)$$

where r_i is the normalized response (firing rate) and b_i the implicit basis function associated with neuron i in the population. The basis function of each neuron is assumed to be linearly related to the tuning function of the neuron as measured in physiological experiments. The basis function approach is similar to the approach proposed in this paper in that the stimulus space is spanned by a limited number of neurons with preferred stimuli or state vectors. The two approaches differ in how probability distributions are decoded from neural responses, one using an additive method and the other

using a logarithmic transformation as discussed above.

A limitation of the basis function approach is that due to its additive nature, it cannot represent distributions that are sharper than the component distributions. A second class of models addresses this problem using a generative approach, where an encoding model is first assumed and a Bayesian decoding model is used to estimate the stimulus \mathbf{x} (or its distribution), given a set of responses r_i (Zhang, Ginzburg, McNaughton, & Sejnowski, 1998; Pouget, Zhang, Deneve, & Latham, 1998; Zemel et al., 1998; Zemel & Dayan, 1999; Wu, Chen, Niranjan, & Amari, 2003). For example, in the distributional population coding (DPC) method (Zemel et al., 1998; Zemel & Dayan, 1999), the responses are assumed to depend on general distributions $P(\mathbf{x})$ and a Bayesian method is used to decode a probability distribution over possible distributions over \mathbf{x} . The best estimate in this method is not a single value of \mathbf{x} but an entire distribution over \mathbf{x} , which is assumed to be represented by the neural population. The underlying goal of representing entire distributions within neural populations is common to both the DPC approach and the model being proposed in this paper. However, the approaches differ in how they achieve this goal: the DPC method assumes prespecified tuning functions for the neurons and a sophisticated, non-neural decoding operation whereas the method introduced in this paper directly instantiates a probabilistic generative model (in this case, a hidden Markov model) with a comparatively straightforward linear decoding operation as described above.

Several probabilistic models have also been suggested for solving specific problems in visual motion processing such as the aperture problem (Simoncelli, 1993; Koehler, Anton, & Burnod, 1999; Zemel & Dayan, 1999; Ascher & Grzywacz, 2000; Freeman, Haddon, & Pasztor, 2002; Weiss & Fleet, 2002; Weiss et al., 2002). These typically rely on a prespecified bank of spatiotemporal filters to generate a probability distribution over velocities, which is processed according to Bayesian principles. The model proposed in this paper differs from these previous approaches in making use of an explicit generative model for capturing the statistics of time-varying inputs (i.e. a HMM). Thus, the selectivity for direction and velocity is an emergent property of the network and not a consequence of tuned spatiotemporal filters chosen a priori.

Finally, Rao and Ballard have suggested a hierarchical model for neural computation based on Kalman filtering (or predictive coding) (Rao & Ballard, 1997, 1999; Rao, 1999). The Kalman filter model is based on a generative model similar to the one

used in the hidden Markov model as discussed in this paper, except that the states are assumed to be continuous-valued and both the observation and transition probability densities are assumed to be Gaussian. As a result, the inference problem becomes one of estimating the mean and covariance of the Gaussian distribution of states at each time step. Rao and Ballard showed how the mean could be computed in a recurrent network in which feedback from a higher layer to the input layer carries predictions of expected inputs and the feedforward connections convey the errors in prediction, which are used to correct the estimate of the mean state. The neural estimation of covariance was not addressed. In contrast to the Kalman filter model, the current model does not require propagation of predictive error signals and allows the full distribution of states to be estimated, rather than just the mean. In addition, the observation and transition distributions can be arbitrary and need not be Gaussian. The main drawback is the restriction to discrete states, which implies that distributions over continuous state variables are approximated using discrete samples. Fortunately, these discrete samples, which are the preferred states θ_i of neurons in the network, can be adapted by changing the feedforward and feedback weights (\mathbf{W} and \mathbf{M}) in response to inputs. Thus, the distribution over a continuous state space can be modeled to different degrees of accuracy in different parts of the state space by a finite number of neurons that adapt their preferred states to match the statistics of the inputs. Possible procedures for adapting the feedforward and feedback weights are discussed in Section 8.1.

7.4.2 Models of Bayesian Decision-Making

Recent experiments using psychophysics and electrophysiology have shed light on the probabilistic basis of visual decision-making in humans and monkeys. There is converging evidence that decisions are made based on the log of the likelihood ratio of one alternative over another (Carpenter & Williams, 1995; Gold & Shadlen, 2001). Carpenter and colleagues have suggested a mathematical model called “LATER” (Linear Approach to Threshold with Ergodic Rate) (Carpenter & Williams, 1995; Reddi & Carpenter, 2000) for explaining results from their visual decision-making task (described in Section 6.2). In their model, the evidence for one alternative over another is accumulated in the form of a log likelihood ratio at a constant rate for a given trial, resulting in a decision in favor of the first alternative if the accumulated variable exceeds a fixed threshold. This is similar to the decision-making process posited in the current paper,

except that we do not assume a linear rise to threshold. Evidence for the LATER model is presented in (Carpenter & Williams, 1995; Reddi & Carpenter, 2000). A related and more detailed model has been proposed by Gold and Shadlen to explain results from the random dots task (Gold & Shadlen, 2001). The Gold-Shadlen model is related to “accumulator models” commonly used in psychology (Ratcliff, Zandt, & McKoon, 1999; Luce, 1986; Usher & McClelland, 2001) and treats decision-making as a diffusion process that is biased by a log likelihood ratio. Rather than assuming that entire probability distributions are represented within cortical circuits as proposed in this paper, the Gold-Shadlen model assumes that the difference between two firing rates (e.g. from two different motion selective neurons in cortical area MT) can be interpreted as a log likelihood ratio of one task alternative over another (e.g., one motion direction over another). We believe that there are enormous advantages to be gained if the brain could represent entire probability distributions of stimuli because such a representation allows not just decision-making when faced with a discrete number of alternatives but also a wide variety of other useful tasks related to probabilistic reasoning, such as estimation of an unknown quantity and planning in uncertain environments.

A number of models of decision-making, based typically on diffusion processes or races between variables, have been proposed in the psychological literature (Ratcliff et al., 1999; Luce, 1986; Usher & McClelland, 2001). Other recent models have focused on establishing a relation between the effects of rewards on decision-making and concepts from decision theory and game theory (Platt & Glimcher, 1999; Glimcher, 2002). These models, like the models of Carpenter, Shadlen, and colleagues, are mathematical models that are aimed mainly at explaining behavioral data but are formulated at a higher level of abstraction than the neural implementation level. The model presented in this paper seeks to bridge this gap by showing how rigorous mathematical models of decision-making could be implemented within recurrent neural circuits. A recent model proposed by Wang (2002) shares the same goal but approaches the problem from a different viewpoint. Starting with a biophysical circuit that implements an “attractor network,” Wang shows that the network exhibits many of the properties seen in cortical decision-making neurons. The inputs to Wang’s decision-making model are not based on a generative model of time-varying inputs but are assumed to be two abstract inputs modeled as two Poisson processes whose rates are determined by Gaussian distributions. The means of these distributions depend on the motion coherence

level linearly, such that the two means approach each other as the coherence is reduced. Thus, the effects of motion coherence on the inputs to the decision-making process are built-in rather than being computed from input images as in the present model. Consequently, Wang’s model, like the other decision-making models discussed above, does not directly address the issue of how probability distributions of stimuli may be represented and processed in neural circuits.

8 Conclusions and Future Work

We have shown that a recurrent network of leaky integrator neurons can approximate Bayesian inference for a hidden Markov model. Such networks have previously been used to model a variety of cortical response properties (Dayan & Abbott, 2001). Our results suggest a new interpretation of neural firing rates in such networks as representing log posterior probabilities. Such a hypothesis is consistent with the suggestions made by Carpenter, Shadlen, and others based on studies in visual decision-making. We illustrated the performance of the model using the examples of visual orientation discrimination and motion detection. In the case of motion detection, we showed how a model cortical network can exhibit direction selectivity and compute the log posterior probability of motion direction, given a sequence of input images. The model thus provides a probabilistic interpretation of direction selective responses in cortical areas V1 and MT. We also showed how the outputs of the motion detection network could be used to decide the direction of coherent motion in the well-known random dots task. The activity of the “decision neurons” in the model during this task resemble the activity of evidence-accumulating neurons in LIP and FEF, two cortical areas known to be involved in visual decision-making. The model correctly predicts reaction times as a function of stimulus coherence and produces reaction time distributions that are qualitatively similar to those obtained in eye movement experiments that manipulate prior probabilities of targets and task urgency.

Although the results described thus far are encouraging, several important questions remain: (a) How can the feedforward weights \mathbf{W} and recurrent weights \mathbf{M} be learned directly from input data, (b) How can the approach be generalized to graphical models that are more sophisticated than hidden Markov models, and (c) How can the approach be extended to incorporate the influence of rewards on Bayesian estimation, learning,

and decision-making? We conclude by discussing below some potential strategies for addressing these questions in future studies.

8.1 Learning Synaptic Weights

We intend to explore the question of synaptic learning using biologically-plausible approximations to the expectation-maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). For standard hidden Markov models, the E-step in the EM algorithm is realized by the “forward-backward” algorithm (Rabiner & Juang, 1986). The network described in this paper implements the “forward” part of this algorithm and can be regarded as an approximate E-step for the case of on-line estimation. One possible likelihood function³ for the on-line case is:

$$Q_t(\mathbf{W}, \mathbf{M}) = \sum_j P(\theta_j^t | \mathbf{I}(t), \dots, \mathbf{I}(1), \mathbf{W}, \mathbf{M}) \log P(\theta_j^t, \mathbf{I}(t) | \mathbf{I}(t-1), \dots, \mathbf{I}(1), \mathbf{W}, \mathbf{M})$$

This function can be re-written in terms of the outputs v_j of the model network and the summed inputs u_j from Section 3 for the current values of the weights \mathbf{W} and \mathbf{M} :

$$Q_t(\mathbf{W}, \mathbf{M}) = \sum_j e^{v_j(t+1)} u_j(t) \quad (29)$$

For synaptic learning, we can perform an on-line M-step to maximize Q_t through gradient ascent with respect to \mathbf{W} and \mathbf{M} (this actually implements a form of incremental or generalized EM algorithm (Neal & Hinton, 1998)):

$$\Delta \mathbf{w}_i(t) = \alpha \frac{\partial Q_t}{\partial \mathbf{w}_i} \quad (30)$$

$$= \alpha \epsilon g_i(t) e^{v_i(t+1)} \mathbf{I}(t)$$

$$\Delta m_{ik}(t) = \beta \frac{\partial Q_t}{\partial m_{ik}} \quad (31)$$

$$= \beta g_i(t) e^{v_i(t+1)} v_k(t)$$

where α and β are learning rates, and $g_i(t) = (1 + u_i(t) - Q_t(\mathbf{W}, \mathbf{M}))$ is a gain function. The above expressions were derived by substituting the right hand side of the equations for $v_i(t+1)$ and $u_i(t)$ respectively (see Section 3), and approximating the gradient by retaining only the first-order terms pertaining to time step t .

³This approximates the full on-line likelihood function $Q_t^F(\mathbf{W}, \mathbf{M}) = \sum_j P(\theta_j^t | \mathbf{I}(t), \dots, \mathbf{I}(1), \mathbf{W}, \mathbf{M}) \log P(\theta_j^t, \mathbf{I}(t), \mathbf{I}(t-1), \dots, \mathbf{I}(1) | \mathbf{W}, \mathbf{M})$.

The learning rules above are interesting variants of the well-known Hebb rule (Hebb, 1949): the synaptic weight change is proportional to the product of the input ($\mathbf{I}(t)$ or $v_k(t)$) with an exponential function of the output $v_i(t + 1)$ (rather than simply the output), thereby realizing a soft form of winner-take-all competition. Furthermore, the weight changes are modulated by the gain function $g_i(t)$, which is positive only when $(u_i(t) - Q_t(\mathbf{W}, \mathbf{M})) > -1$ i.e. when $P(\theta_i^t, \mathbf{I}(t) | \mathbf{I}(t - 1), \dots, \mathbf{I}(1), \mathbf{W}, \mathbf{M}) > (1/e)e^{Q_t(\mathbf{W}, \mathbf{M})}$. Thus, both learning rules remain anti-Hebbian and encourage decorrelation (Földiák, 1990) unless the net feedforward and recurrent input ($u_i(t)$) is high enough to make the joint probability of the state i and the current input exceed the threshold of $(1/e)e^{Q_t(\mathbf{W}, \mathbf{M})}$. In that case, the learning rules switch to a Hebbian regime, allowing the neuron to learn the appropriate weights to encode state i . An in-depth investigation of these and related learning rules is currently underway and will be discussed in a future article.

8.2 Generalization to Other Graphical Models

A second open question is how the suggested framework could be extended to allow neural implementation of hierarchical Bayesian inference in graphical models that are more general than hidden Markov models (cf. (Dayan et al., 1995; Dayan & Hinton, 1996; Hinton & Ghahramani, 1997; Rao & Ballard, 1997, 1999)). Such models would provide a better approximation to the multi-layered architecture of the cortex and the hierarchical connections that exist between cortical areas (Felleman & Van Essen, 1991). In future studies, we hope to investigate neural implementations of various types of graphical models by extending the method described in this paper for implementing Equation 5. In particular, Equation 5 can be regarded as a special case of the general sum-product rule for belief propagation in a Bayesian network (Jordan & Weiss, 2002). Thus, in addition to incorporating “belief messages” from the previous time step and the current input as in Equation 5, a more general rule for Bayesian inference would also incorporate messages from other hierarchical levels and potentially, from future time steps. Investigating such generalizations of Equation 5 and their neural implementation is an important goal of our ongoing research efforts.

8.3 Incorporating Rewards

A final question of interest is how rewards associated with various choices available in a task can be made to influence the activities of decision neurons and the decision-making behavior of the model. We expect ideas from reinforcement learning and Bayesian decision theory as well as recent neurophysiological results in the monkey (Platt & Glimcher, 1999) to be helpful in guiding our research in this direction.

Acknowledgments

This research was supported by a Sloan Research Fellowship, NSF grant no. 130705, and an ONR Young Investigator Award. I am grateful to Peter Dayan, Sophie Deneve, Aaron Hertzmann, John Palmer, Michael Rudd, Michael Shadlen, Eero Simoncelli, and Josh Tenenbaum for discussions on various ideas related to this work. I would also like to thank the two anonymous reviewers for their constructive comments and suggestions.

References

- Anastasio, T. J., Patton, P. E., & Belkacem-Boussaid, K. (2000). Using Bayes' rule to model multisensory enhancement in the superior colliculus. *Neural Computation*, *12*(5), 1165–1187.
- Anderson, C. H. (1995). Unifying perspectives on neuronal codes and processing. In *XIX International workshop on condensed matter theories*. Caracas, Venezuela.
- Anderson, C. H. & Van Essen, D. C. (1994). Neurobiological computational systems. In Zurada, J. M., Marks II, R. J., & Robinson, C. J. (Eds.), *Computational Intelligence: Imitating Life*, pp. 213–222. New York, NY: IEEE Press.
- Ascher, D. & Grzywacz, N. M. (2000). A Bayesian model for the measurement of visual velocity. *Vision Research*, *40*, 3427–3434.
- Barlow, H. B. (1969). Pattern recognition and the responses of sensory neurons. *Annals of the New York Academy of Sciences*, *156*, 872–881.

- Barlow, H. B. (1972). Single units and cognition: a neurone doctrine for perceptual psychology. *Perception, 1*, 371–394.
- Bloj, M. G., Kersten, D., & Hurlbert, A. C. (1999). Perception of three-dimensional shape influences colour perception through mutual illumination. *Nature, 402*, 877–879.
- Bridle, J. S. (1990). Alpha-Nets: A recurrent ‘neural’ network architecture with a hidden Markov model interpretation. *Speech Communication, 9*(1), 83–92.
- Britten, K. H., Shadlen, M. N., Newsome, W. T., & Movshon, J. A. (1992). The analysis of visual motion: a comparison of neuronal and psychophysical performance. *Journal of Neuroscience, 12*, 4745–4765.
- Carpenter, R. H. S. & Williams, M. L. L. (1995). Neural computation of log likelihood in control of saccadic eye movements. *Nature, 377*, 59–62.
- Dayan, P. & Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.
- Dayan, P. & Hinton, G. (1996). Varieties of Helmholtz machine. *Neural Networks, 9*(8), 1385–1403.
- Dayan, P., Hinton, G., Neal, R., & Zemel, R. (1995). The Helmholtz machine. *Neural Computation, 7*, 889–904.
- DeAngelis, G. C., Robson, J. G., Ohzawa, I., & Freeman, R. D. (1992). Organization of suppression in receptive fields of neurons in cat visual cortex. *J. Neurophysiol., 68*(1), 144–163.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B, 39*, 1–38.
- Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. *Nature Neuroscience, 2*(8), 740–745.
- Deneve, S. & Pouget, A. (2001). Bayesian estimation by interconnected neural networks (abstract no. 237.11). *Society for Neuroscience Abstracts, 27*.

- Eliasmith, C. & Anderson, C. H. (2003). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA: MIT Press.
- Felleman, D. J. & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*, 1–47.
- Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning. *Biol. Cybern.*, *64*, 165–170.
- Freeman, W. T., Haddon, J., & Pasztor, E. C. (2002). Learning motion analysis. In *Probabilistic Models of the Brain: Perception and Neural Function*, pp. 97–115. Cambridge, MA: MIT Press.
- Glimcher, P. (2002). Decisions, decisions, decisions: choosing a biological science of choice. *Neuron*, *36*(2), 323–332.
- Gold, J. I. & Shadlen, M. N. (2001). Neural computations that underlie decisions about sensory stimuli. *Trends in Cognitive Sciences*, *5*(1), 10–16.
- Hanes, D. P. & Schall, J. D. (1996). Neural control of voluntary movement initiation. *Science*, *274*, 427–430.
- Hebb, D. O. (1949). *The Organization of Behavior*. Wiley.
- Hinton, G. E. & Brown, A. D. (2002). Learning to use spike timing in a restricted Boltzmann machine. In *Probabilistic Models of the Brain: Perception and Neural Function*, pp. 285–296. Cambridge, MA: MIT Press.
- Hinton, G. E. & Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Phil. Trans. Roy. Soc. Lond. B*, *352*, 1177–1190.
- Hinton, G. & Sejnowski, T. (1983). Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 448–453. Washington 1983. IEEE, New York.
- Hinton, G. & Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. & McClelland, J. (Eds.), *Parallel Distributed Processing*, Vol. 1, chap. 7, pp. 282–317. MIT Press, Cambridge.

- Hoyer, P. O. & Hyvärinen, A. (2003). Interpreting neural response variability as Monte Carlo sampling of the posterior. In *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press. To appear.
- Jordan, M. I. & Weiss, Y. (2002). Graphical models: probabilistic inference. In *The Handbook of Brain Theory and Neural Networks, 2nd Edition*. Cambridge, MA: MIT Press.
- Knierim, J. & Essen, D. C. V. (1992). Neural responses to static texture patterns in area V1 of the alert macaque monkey. *Journal of Neurophysiology*, 67, 961–980.
- Knill, D. C. & Richards, W. (1996). *Perception as Bayesian Inference*. Cambridge, UK: Cambridge University Press.
- Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press.
- Koechlin, E., Anton, J. L., & Burnod, Y. (1999). Bayesian inference in populations of cortical neurons: a model of motion integration and segmentation in area MT. *Biological Cybernetics*, 80, 25–44.
- Lewicki, M. S. & Sejnowski, T. J. (1997). Bayesian unsupervised learning of higher order structure. In Mozer, M., Jordan, M., & Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press.
- Luce, R. D. (1986). *Response time: Their role in inferring elementary mental organization*. New York: Oxford University Press.
- Mazurek, M. E., Ditterich, J., Palmer, J., & Shadlen, M. N. (2001). Effect of prior probability on behavior and LIP responses in a motion discrimination task (abstract no. 58.11). *Society for Neuroscience Abstracts*, 27.
- Mel, B. W. (1993). Synaptic integration in an excitable dendritic tree. *J. Neurophysiology*, 70(3), 1086–1101.
- Miller, E. K., Li, L., & Desimone, R. (1991). A neural mechanism for working and recognition memory in inferior temporal cortex. *Science*, 254, 1377–1379.

- Neal, R. M. & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. I. (Ed.), *Learning in Graphical Models*, pp. 355–368. Norwell, MA: Kluwer Academic Publishers.
- Platt, M. L. & Glimcher, P. W. (1999). Neural correlates of decision variables in parietal cortex. *Nature*, *400*, 233–238.
- Pouget, A., Dayan, P., & Zemel, R. S. (2000). Information processing with population codes. *Nature Reviews Neuroscience*, *1*(2), 125–132.
- Pouget, A., Zhang, K., Deneve, S., & Latham, P. E. (1998). Statistically efficient estimation using population coding. *Neural Computation*, *10*(2), 373–401.
- Rabiner, L. & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, *3*, 4–16.
- Rao, R. P. N. (1999). An optimal estimation approach to visual perception and learning. *Vision Research*, *39*(11), 1963–1989.
- Rao, R. P. N. & Ballard, D. H. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, *9*(4), 721–763.
- Rao, R. P. N. & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive field effects. *Nature Neuroscience*, *2*(1), 79–87.
- Rao, R. P. N., Olshausen, B. A., & Lewicki, M. S. (2002). *Probabilistic Models of the Brain: Perception and Neural Function*. Cambridge, MA: MIT Press.
- Ratcliff, R., Zandt, T. V., & McKoon, G. (1999). Connectionist and diffusion models of reaction time. *Psychological Review*, *106*, 261–300.
- Reddi, B. A. & Carpenter, R. H. (2000). The influence of urgency on decision time. *Nature Neuroscience*, *3*(8), 827–830.
- Reynolds, J. H., Chelazzi, L., & Desimone, R. (1999). Competitive mechanisms subserve attention in macaque areas V2 and V4. *Journal of Neuroscience*, *19*, 1736–1753.

- Roitman, J. D. & Shadlen, M. N. (2002). Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of Neuroscience*, *22*, 9475–9489.
- Schall, J. D. & Hanes, D. P. (1998). Neural mechanisms of selection and control of visually guided eye movements. *Neural Networks*, *11*, 1241–1251.
- Schall, J. D. & Thompson, K. G. (1999). Neural selection and control of visually guided eye movements. *Annual Review of Neuroscience*, *22*, 241–259.
- Seung, H. S. (1996). How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, *93*, 13339–13344.
- Shadlen, M. N. & Newsome, W. T. (2001). Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. *Journal of Neurophysiology*, *86*(4), 1916–1936.
- Simoncelli, E. P. (1993). *Distributed representation and analysis of visual motion*. Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT.
- Srinivasan, M. V., Laughlin, S. B., & Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. *Proc. R. Soc. Lond. B*, *216*, 427–459.
- Usher, M. & McClelland, J. (2001). On the time course of perceptual choice: the leaky competing accumulator model. *Psychological Review*, *108*, 550–592.
- Wang, X.-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neuroscience*, *24*, 455–463.
- Wang, X.-J. (2002). Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, *36*(5), 955–968.
- Weiss, Y. & Fleet, D. J. (2002). Velocity likelihoods in biological and machine vision. In *Probabilistic Models of the Brain: Perception and Neural Function*, pp. 77–96. Cambridge, MA: MIT Press.
- Weiss, Y., Simoncelli, E. P., & Adelson, E. H. (2002). Motion illusions as optimal percepts. *Nature Neuroscience*, *5*(6), 598–604.

- Wu, S., Chen, D., Niranjan, M., & Amari, S.-I. (2003). Sequential Bayesian decoding with a population of neurons. *Neural Computation*, *15* (to appear).
- Zemel, R. S. & Dayan, P. (1999). Distributional population codes and multiple motion models. In *Advances in Neural Information Processing Systems 11*, pp. 174–180. Cambridge, MA: MIT Press.
- Zemel, R. S., Dayan, P., & Pouget, A. (1998). Probabilistic interpretation of population codes. *Neural Computation*, *10*(2), 403–430.
- Zhang, K., Ginzburg, I., McNaughton, B. L., & Sejnowski, T. J. (1998). Interpreting neuronal population activity by reconstruction: a unified framework with application to hippocampal place cells. *Journal of Neurophysiology*, *79*(2), 1017–1044.
- Zipser, K., Lamme, V. A. F., & Schiller, P. H. (1996). Contextual modulation in primary visual cortex. *Journal of Neuroscience*, *16*(22), 7376–7389.

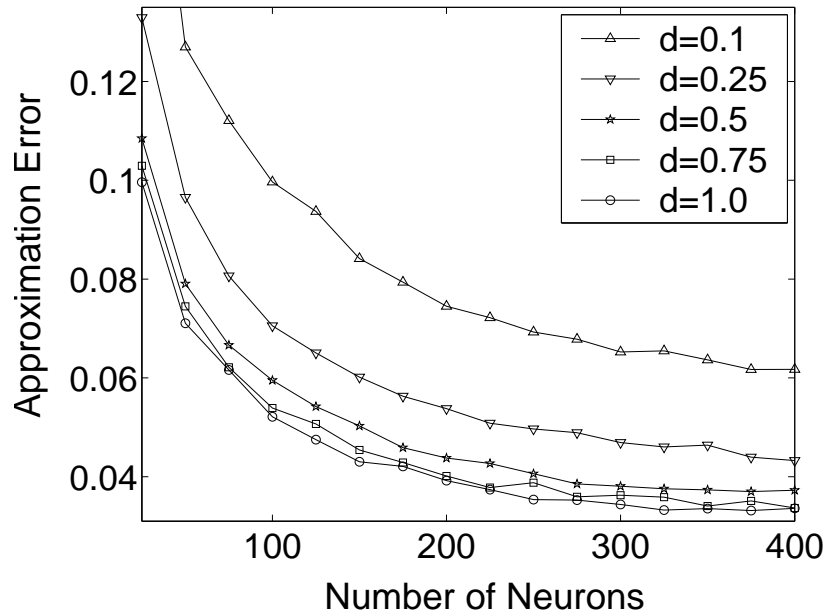


Figure 1: **Approximation Accuracy for Different Classes of Transition Probability Matrices.** Each data point represents the average approximation error from Equation 11 for a particular number of neurons in the recurrent network for a fixed set of conditional probabilities $x_j(t) = P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1))$. Each curve represents the errors for a particular density d of the random transition probability matrix (d is the fraction of nonzero entries in the matrix).

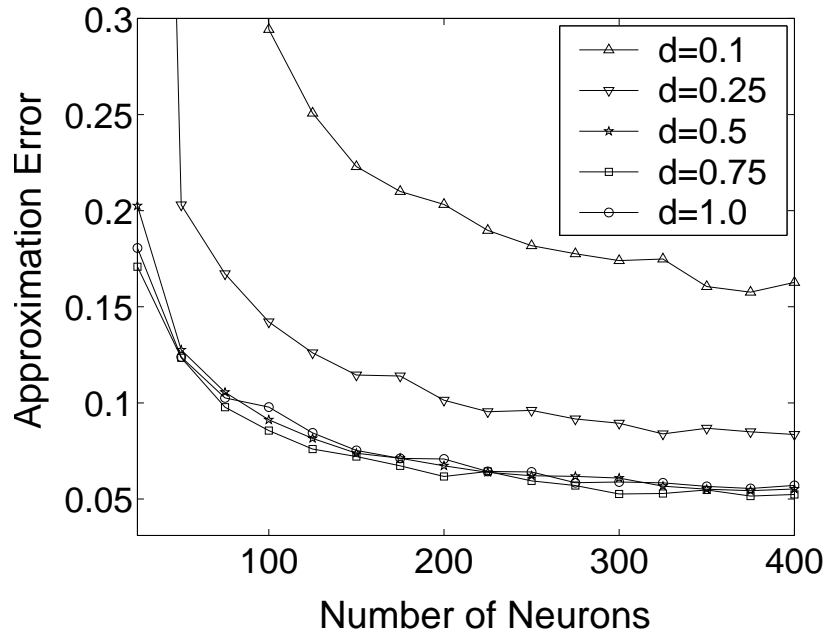


Figure 2: **Approximation Accuracy for Different Sets of Conditional Probabilities** $P(\theta_j^{t-1} | \mathbf{I}(t-1), \dots, \mathbf{I}(1)) (= x_j(t))$. The data points represent the average approximation error from Equation 11 as a function of the number of neurons in the network for a fixed set of transition probabilities. Each curve represents the errors for a particular density d of the random conditional probability matrix (see text for details).

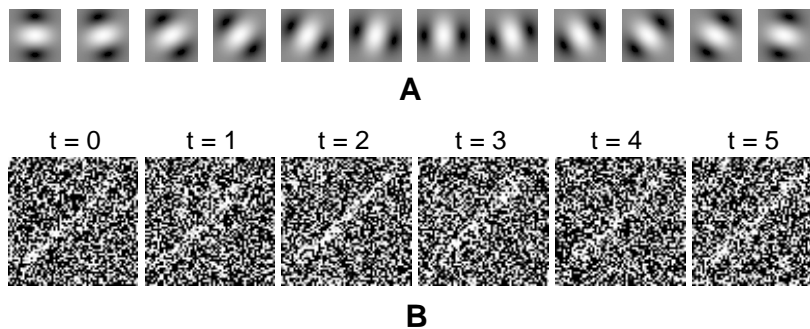


Figure 3: **Estimating Stimulus Orientation from Noisy Images.** (A) Twelve of the 36 oriented filters used in the recurrent network for estimating stimulus orientation. These filters represent the feedforward connections $F(\theta_i)$. Bright pixels denote positive (excitatory) values and dark pixels denote negative (inhibitory) values. (B) Example of a noisy input image sequence containing a bright oriented bar embedded in additive zero-mean Gaussian white noise with a standard deviation of 1.5.

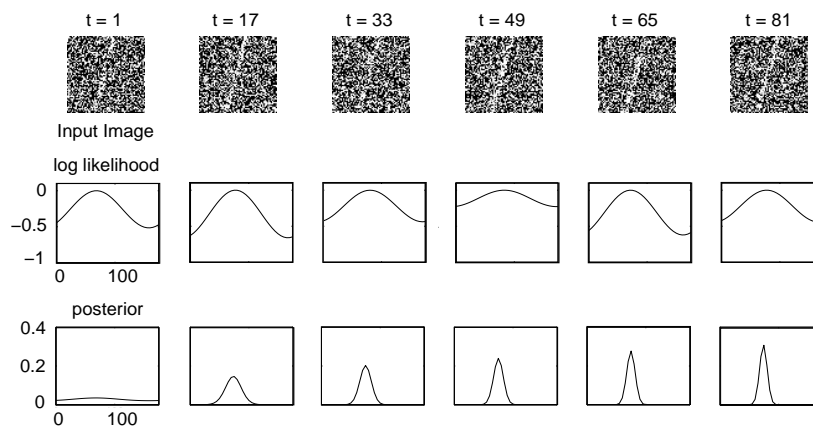


Figure 4: **Example of Noisy Orientation Estimation.** (Top Row) Example images at different time steps from a noisy input sequence containing an oriented bar. (Middle Row) Feedforward inputs representing log likelihood values for the inputs shown above. Note the wide variation in the peak of the activities. (Bottom Row) Output activities shown as posterior probabilities of the 36 orientations encoded by the neurons in the network. Note the stable peak in activity achieved through recurrent accumulation of evidence over time despite the wide variation in the input likelihoods.

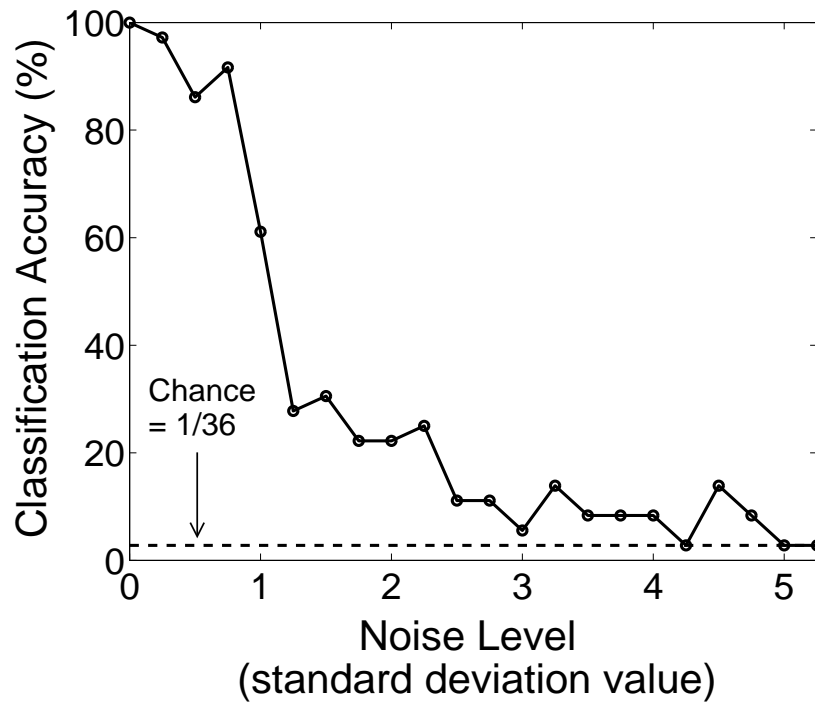


Figure 5: **Orientation Estimation Accuracy as a Function of Noise Level.** The data points show the classification rates obtained for a set of 36 oriented bar stimuli embedded in additive zero-mean Gaussian white noise with standard deviation values as shown on the x axis. The dotted line shows the chance classification rate ($1/36 \times 100$) obtained by the strategy of randomly guessing an orientation.

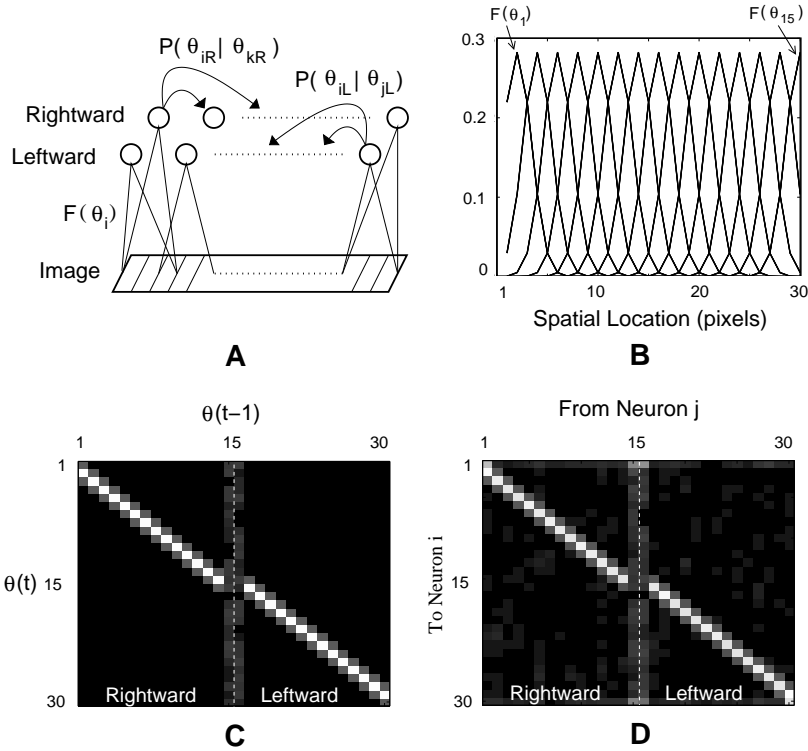


Figure 6: **Recurrent Network for Motion Detection.** (A) depicts a recurrent network of neurons, shown for clarity as two chains selective for leftward and rightward motion respectively. The feedforward synaptic weights for neuron i (in the leftward or rightward chain) are determined by $F(\theta_i)$. The recurrent weights reflect the transition probabilities $P(\theta_{iR}|\theta_{kR})$ and $P(\theta_{iL}|\theta_{jL})$. (B) Feedforward weights $F(\theta_i)$ for neurons $i = 1, \dots, 15$ (rightward chain). The feedforward weights for neurons $i = 15, \dots, 30$ (leftward chain) are identical. (C) Transition probabilities $P(\theta(t) = \theta_i | \theta(t-1) = \theta_j)$ for $i = 1, \dots, 30$. Probability values are proportional to pixel brightness. (D) Recurrent weights m_{ij} computed from the transition probabilities in (C) using Equation 13.

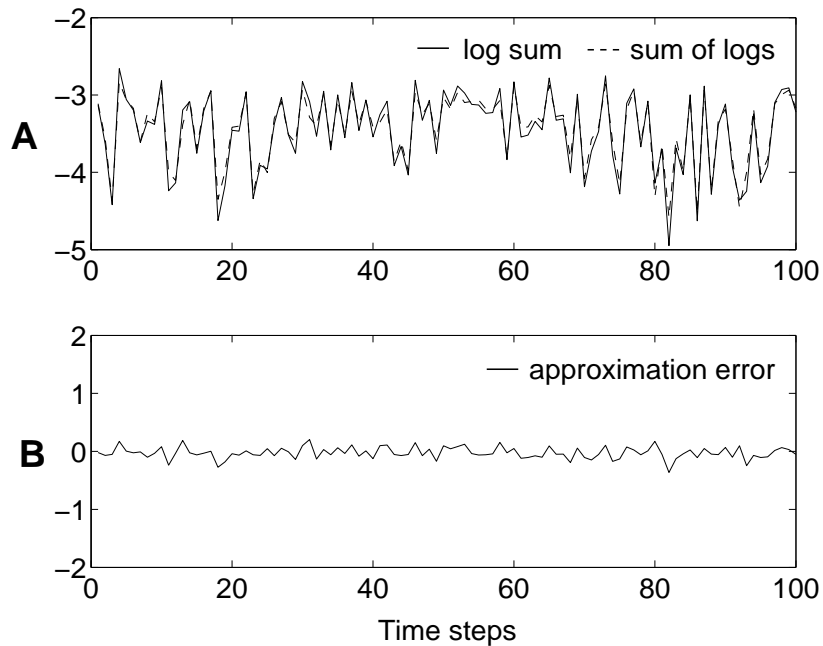


Figure 7: **Approximating Transition Probabilities with Recurrent Weights.** (A) compares the right and left sides of Equation 11 for random test data $x_j(t)$ (different from the training $x_j(t)$) using the transition probabilities and weights shown in Figures 6C and 6D respectively. The solid line represents the right side of Equation 11 while the dotted line represents the left side. Only the results for a single neuron ($i = 8$) are shown but similar results were obtained for the rest of the neurons. (B) shows the approximation error (difference between the two sides of Equation 11 over time).

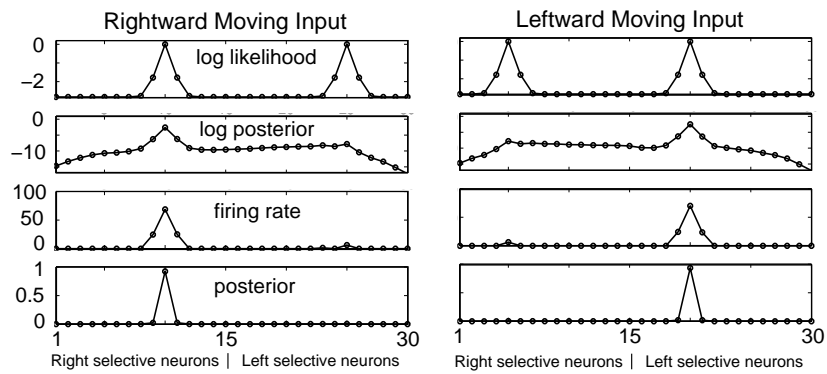


Figure 8: **Network Output for a Moving Stimulus.** (Left Panel) The four plots depict respectively the log likelihoods, log posteriors, neural firing rates, and posterior probabilities observed in the network for a rightward moving bar when it arrives at the central image location. Note that the log likelihoods are the same for the rightward and leftward selective neurons (the first 15 and last 15 neurons respectively, as dictated by the feedforward weights in Figure 6B) but the outputs of these neurons correctly reflect the direction of motion as a result of recurrent interactions. (Right Panel) The same four plots for a leftward moving bar as it reaches the central location.

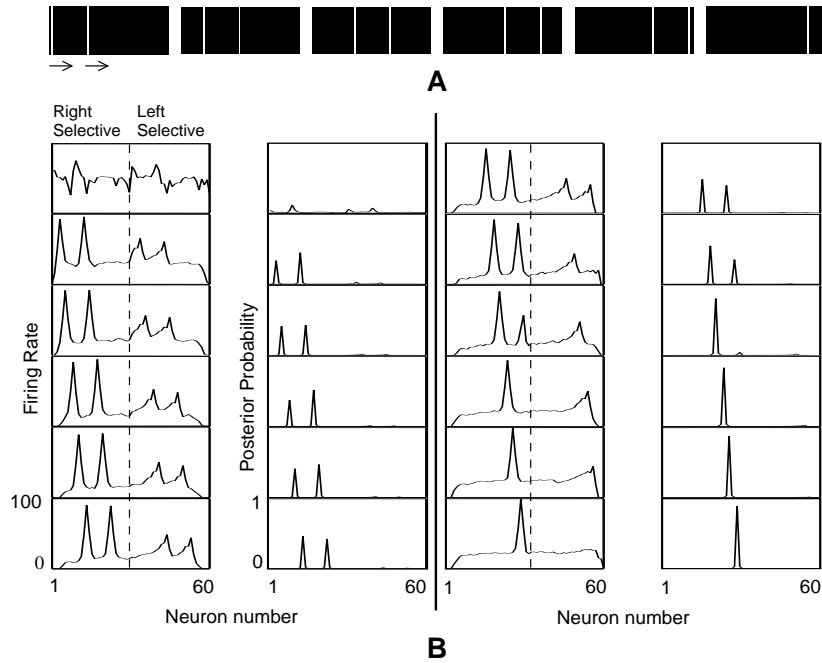


Figure 9: **Encoding Multiple Motions: Same Direction Motion.** (A) Six images from an input sequence depicting two bright bars, both moving rightward. (B) The panels from top-to-bottom on the left and continued on the right show the activities (left) and posterior probabilities (right) for the image sequence in (A) in a motion detection network of 60 neurons, the first 30 encoding rightward motion and the next 30 encoding leftward motion. Note the two large peaks in activity among the right selective neurons, indicating the presence of two rightward moving bars. The corresponding posterior probabilities remain close to 0.5 until only one moving bar remains in the image with a corresponding single peak of probability close to 1.

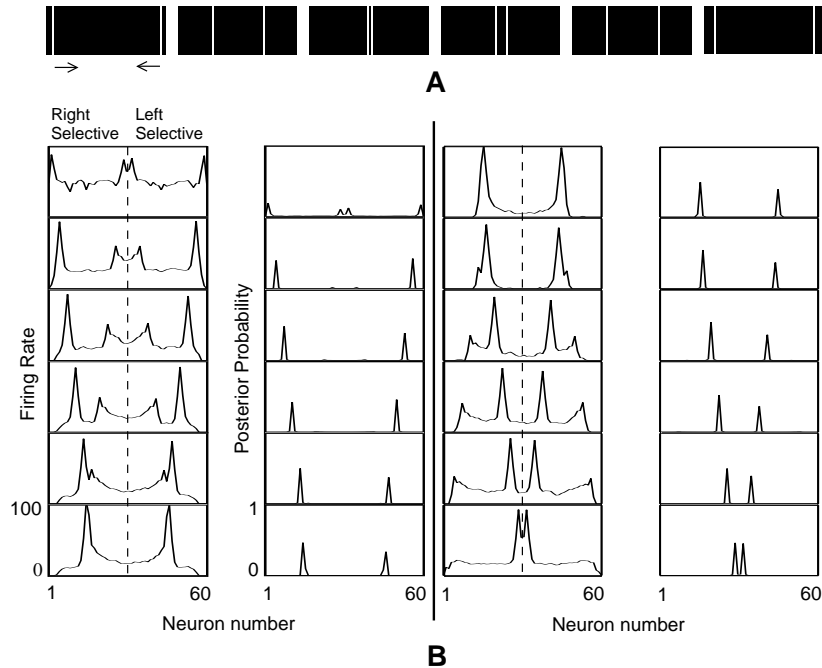


Figure 10: **Encoding Multiple Motions: Motion in Opposite Directions.** (A) Six images from an input sequence depicting two bright bars moving in opposite directions. (B) The panels show the activities in the motion detection network (left) and posterior probabilities (right) for the image sequence in (A). Note the two large peaks in activity moving in opposite directions, indicating the presence of both a leftward and a rightward moving bar. The corresponding posterior probabilities remain close to 0.5 throughout the sequence (after an initial transient).

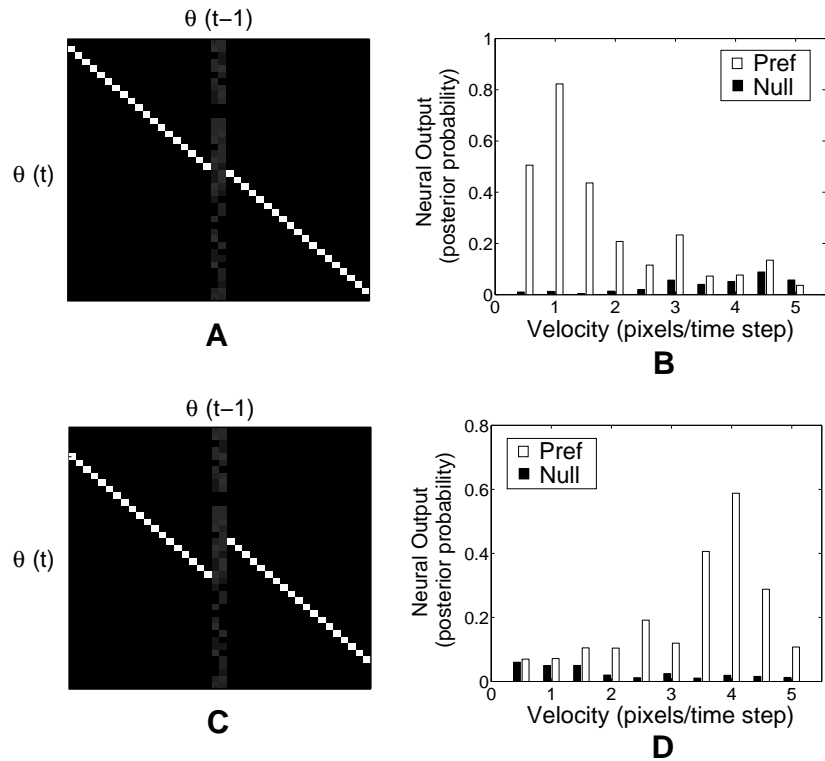


Figure 11: **Encoding Multiple Velocities.** (A) Transition probability matrix that encodes a stimulus moving at a speed of 1 pixel/time step in either direction (see also Figure 6C). (B) Output of a neuron located in the middle of the network as a function of stimulus velocity. The peak outputs of the neuron during stimulus motion (in both the neuron’s preferred (“Pref”) direction and the null direction) are shown as posterior probabilities. Note the relatively broad tuning curve indicating the neuron’s ability to encode multiple velocities. (C) Transition probability matrix that encodes a stimulus moving at a speed of 4 pixels/time step. (D) Peak outputs of the same neuron as in (B) (but with recurrent weights approximating the transition probabilities in (C)). Once again, the neuron exhibits a tuning curve (now centered around 4 pixels/time step) when tested on multiple velocities. Note the shift in velocity sensitivity due to the change in the transition probabilities. A small number of such networks can compute a discrete posterior distribution over the space of stimulus velocities.

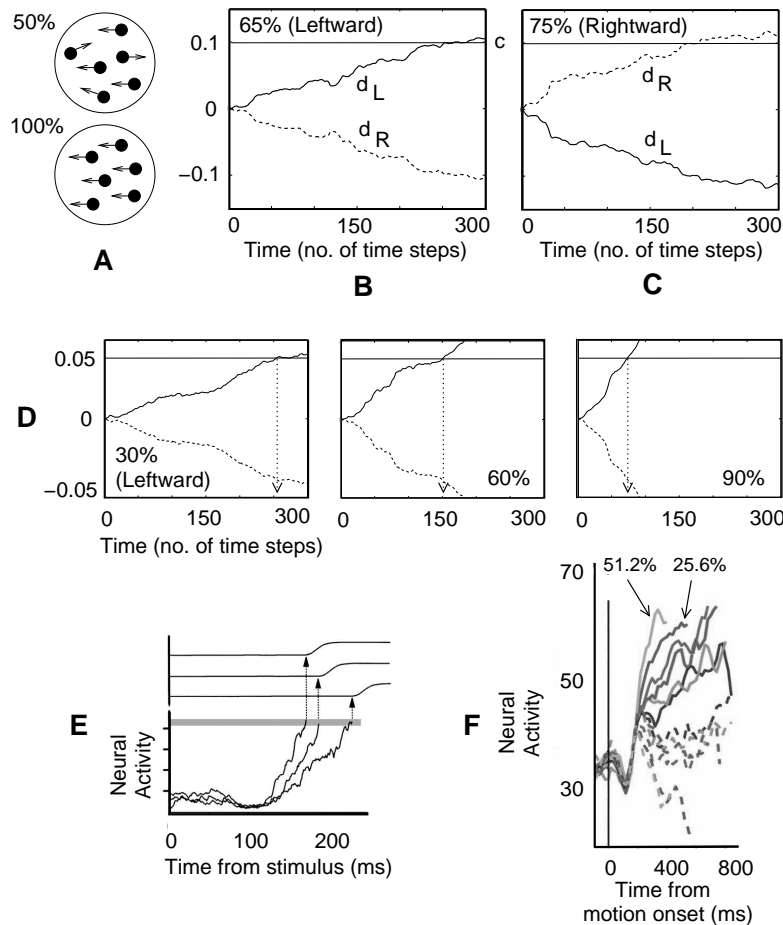


Figure 12: **Output of Decision Neurons in the Model.** (A) Depiction of the random dots task. Two different levels of motion coherence (50% and 100%) are shown. A 1-D version of this stimulus was used in the model simulations. (B) & (C) Outputs $d_L(t)$ and $d_R(t)$ of model decision neurons for two different directions of motion. The decision threshold is labeled “c.” (D) Outputs of decision neurons for three different levels of motion coherence. Note the increase in rate of evidence accumulation at higher coherencies. For a fixed decision threshold, the model predicts faster reaction times for higher coherencies (dotted arrows). (E) Activity of a neuron in area FEF for a monkey performing an eye movement task (from (Schall & Thompson, 1999) with permission). Faster reaction times were associated with a more rapid rise to a fixed threshold (see the three different neural activity profiles). The arrows point to the initiation of eye movements, which are depicted at the top. (F) Averaged firing rate over time of 54 neurons in area LIP during the random dots task, plotted as a function of motion coherence (from (Roitman & Shadlen, 2002) with permission). Solid and dashed curves represent trials in which the monkey judged motion direction toward and away from the receptive field of a given neuron, respectively. The slope of the response is affected by motion coherence (compare, for example, responses for 51.2% and 25.6%) in a manner similar to the model responses shown in (D).

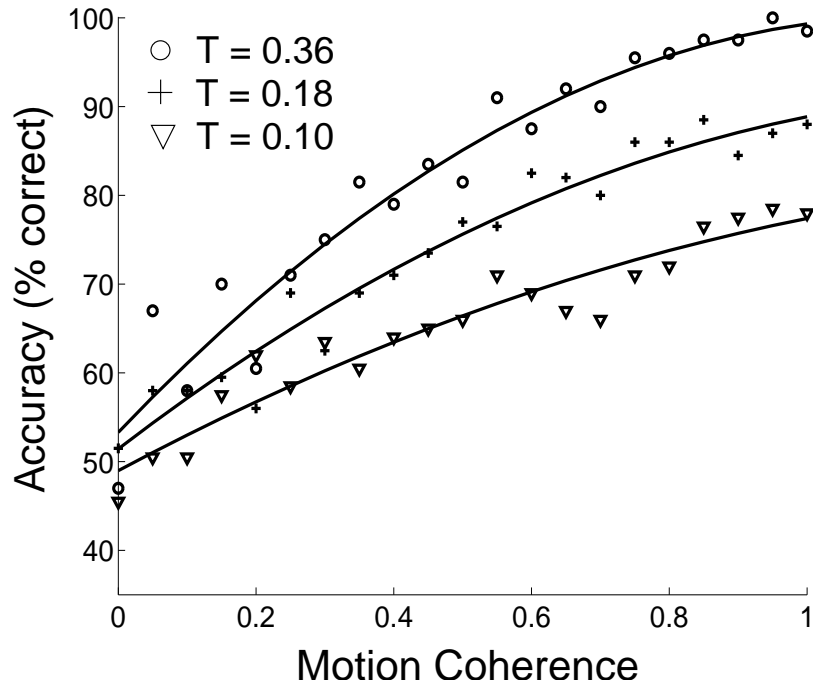


Figure 13: **Performance of the Model Network as a Function of Motion Coherence.** The three curves show the accuracy rate (% correct responses) of the motion network in discriminating between the two directions of motion in the random dots task for three different decision thresholds T . Two trends, which are also observed in human and monkey performance on this task, are the rapid increase in accuracy as a function of motion coherence (especially for large thresholds) and the increase in accuracy for each coherence value for increasing thresholds, with an asymptote close to 100% for large enough thresholds.

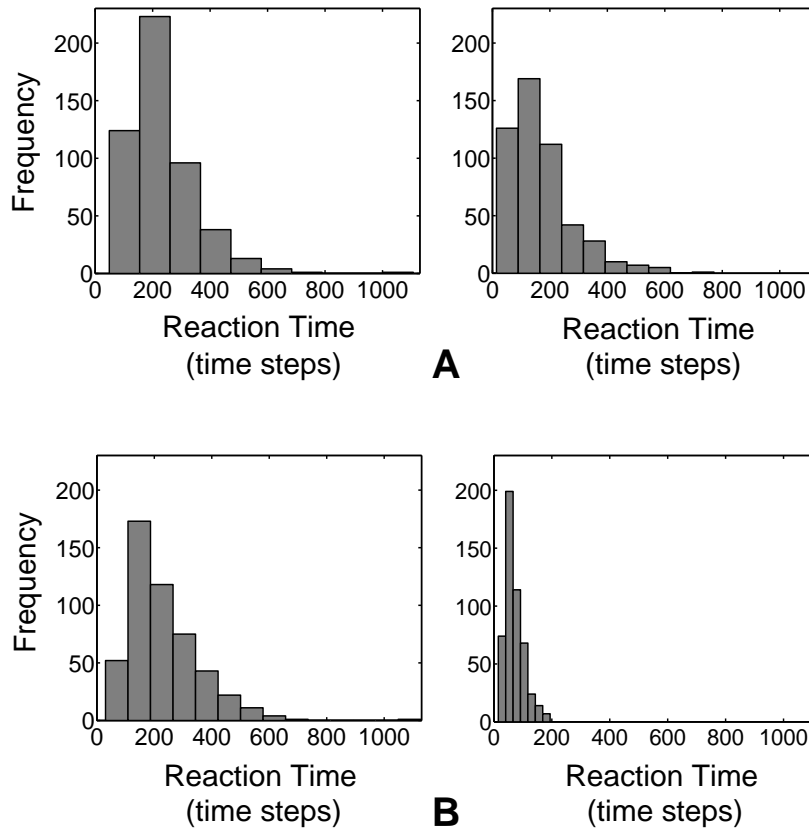


Figure 14: **Reaction Time Distributions as a Function of Prior Probability and Urgency.** (A) The left panel shows the distribution of reaction times for the motion discrimination task obtained from model simulations when leftward and rightward motion are equiprobable. The right panel shows the distribution for the case where leftward motion is more probable than rightward motion (0.51 versus 0.49). Note the leftward shift of the distribution towards shorter reaction times. Both distributions here were restricted to leftward motion trials only. (B) The right panel shows the result of halving the decision threshold, simulating an urgency constraint that favors speed over accuracy. Compared to the non-urgent case (left panel), the reaction time distribution again exhibits a leftward shift towards faster (but potentially inaccurate) responses.