CHAPTER 20

Dynamic Cue Combination in Distributional Population Code Networks

Rama Natarajan and Richard S. Zemel

INTRODUCTION

In this chapter, we investigate how hierarchical networks of neural populations might code stable representations of transient stimulus variables along with the uncertainty about them, while allowing consistent semantics of inputs and outputs at every level of the neural hierarchy. One effective paradigm to study this question is the integration of information from multiple sensory sources. Significant behavioral and neurophysiological evidence substantiates the notion that when separate sensory modalities receive correlated inputs about an external stimulus, the information is often combined to produce a coherent, unitary percept of the world (Stein & Meredith, 1993). This ability of neural systems to combine information from multiple sources affords considerable, ecologically relevant advantages. For instance, it has been shown to attenuate the effects of input variability owing to noise, thereby reducing the overall uncertainty in the final estimate of the stimulus variable.

Sensory cue integration and sensorimotor combinations involve many important and challenging statistical issues; for example, the different sources of information are not equally reliable, and their reliability can change with changing stimulus conditions and task demands. Bayesian probability theory has provided a normative framework for predicting how sensory systems might integrate information optimally, to make perceptual inferences about the environment from noisy sensory data. These predictions have been quantitatively tested using psychophysical experiments that examine what computations are performed when reliability of the cues changes over time. Results from widely ranging studies of human cue integration are consistent in that subjects behave in a manner that takes the uncertainty in sensory inputs into account, to form statistically optimal estimates of stimulus variables (see Knill & Pouget, 2004, for a review). Several different computational strategies have been proposed and validated; some examples are linear weighted averaging (Hillis, Watt, Landy, & Banks, 2004; Jacobs, 1999; Knill & Saunders, 2003; van Beers, Sittig, & van der Gon, 1999), extensions thereof (Landy, Maloney, Johnston, & Young, 1995), multiplicative interactions, and fully probabilistic linear and nonlinear Bayesian inference (Knill, 2003).

Regardless of the exact inference strategy, the common theme in all these studies is that sensory uncertainty determines the optimal weighting scheme for combining sensory inputs for perceptual judgments. Similarly, studies also demonstrate that sensory and motor uncertainties determine how sensory signals should be transformed and used to plan actions and guide behavior. An example of a commonly used paradigm involves manipulating continuous visual feedback from the hand to control pointing movements. Subjects are able to compensate for experimentally induced changes

in sensory (such as position and velocity) and motor uncertainties and optimally adjust the degree to which they rely on the feedback to make corrections (Körding & Wolpert, 2004; Saunders & Knill, 2003, 2004).

What is less clear is how neural systems might represent information about the various cues, to implement these putative computations in a manner sensitive to the uncertainties about the cues. More generally, because of the noise and ambiguities inherent in sensory inputs, many perceptual tasks are better viewed as statistical-inference problems. One normative hypothesis is that the brain represents and computes with probability distributions over all the different stimuli that are consistent with the inputs, as opposed to unitary estimates of stimulus variables (Zemel, Dayan, & Pouget, 1998).

In this chapter, we consider how a network consisting of several neural populations can represent and combine information from multiple cues in a statistically optimal manner, in order to carry out proper probabilistic inference. We then take this aim a step further, considering time-varying inputs. If the neural system can maintain, at each stage of local computation, full distributions over all possible stimulus values, it can allow considerable flexibility in accommodating changing cue uncertainties. This representational scheme thus pushes the envelope on current psychophysical studies of cue combination, yielding a computational network of neural populations where the responses approximate proper inference even when cues' values and uncertainties vary rapidly.

Influential ideas abound regarding how neural systems might encode (Jazayeri & Movshon, 2006; Rao, Olshausen, & Lewicki, 2002) and interpret (decode) probability distributions over stimulus variables (Deneve, Latham, & Pouget, 1999; Paradiso, 1988; Seung & Sompolinsky, 1993; Snippe & Koenderink, 1992), and how various computations, including uncertainty-sensitive, Bayesian optimal statistical processing, might be performed through feedforward and recurrent connections between neural populations (Deneve, Latham, & Pouget, 2001; Pouget, Zhang, Deneve, & Latham, 1998). However, theoretical investigations have for the most part neglected the *temporal* dimension of coding; often, the stimuli are treated as being discrete rather than evolving along full trajectories (Brunel & Nadal, 1998; Van Rullen & Thorpe, 2001). Otherwise it is assumed that encoded stimulus variables do not vary quickly with time, and therefore population spike counts as opposed to actual timing of spikes suffice.

In this chapter we consider how a neural system could properly combine cues even as their reliabilities change dynamically and continuously in time. The rest of the chapter is organized as follows: The next section describes a framework for characterizing how uncertain information may be represented in population codes, and how the representation may be optimized to facilitate optimal decoding. We then motivate and present a coding scheme within this framework that applies to time-varying stimulus variables. Next, we describe a hierarchical network setup that utilizes this coding approach, and we demonstrate an application to a dynamic cue-combination task, a novel adaptation of a sensorimotor task (Körding & Wolpert, 2004). Specifically, we consider how a neural population might recursively integrate dynamic inputs from multiple sensory modalities with learned prior information, to determine appropriate motor commands that control behavior. Then, we present results of simulated experiments with this network. Finally, we discuss implications of this scheme and suggest some future directions.

APPROXIMATING PROBABILISTIC INFERENCE IN POPULATION-CODE NETWORKS

To illustrate the cue-combination process that we address in this chapter, let us consider a simple target-localization task that involves pointing a finger at a visual target. The computational problem entails extracting a task-relevant physical attribute of the environment from multisensory data. In this case, we compute the amount of displacement s^c of the finger from the current location to the target position. Several useful cues (i.e., functions of the sensory

input; see definition in Chapter 5) are available in this context. In our example, we employ a visual cue s^v signaling the visual location of the target, and a proprioceptive cue s^p conveying information about the position of the finger.

We begin by assuming that the basic representations in the network are population codes; that is, information about sensory and motor variables is represented in the joint spiking activity of large populations of neurons (Barlow, 1953; Georgopoulos, Schwartz, & Kettner, 1983). It is impossible to infer the exact location of the target or the finger position from the neural activity due to uncertainty that arises from noise in sensation and the sensorimotor transmission process, as well as potential ambiguities in the stimulus such as low-contrast visual information about target location. Hence, the population codes are treated as encoding whole probability density functions (PDFs) over the underlying variables rather than just a single best estimate. The goal is to provide a rich representation of uncertainty in the aspects of the stimuli that are represented (see Pouget, Dayan, & Zemel, 2003, for a review).

We formulate a neural implementation that respects or aims to closely approximate proper combination of probabilistic information. According to the probabilistic population-coding hypothesis, *explicit* activities $\boldsymbol{\rho} = \{\rho_i\}$ of multiple neurons $i = \{1, 2, 3, \ldots, n\}$ represent a probability distribution $p(s|\boldsymbol{\rho})$ over an *implicit* underlying variable s (such as the position of the target), using some encoding scheme; decoding then describes how the distribution can be recovered from the population activity (Anderson, 1994; Rao et al., 2002; Zemel et al., 1998). One proposal for a neural implementation of cue combination using the distributional coding scheme entailed specifying a simple two-layer feedforward network (Zemel & Dayan, 1997). In the context of the target-localization task we described earlier, the first layer consists of separate populations of visual and proprioceptive neurons that encode probability distributions $p(s^v|\boldsymbol{\rho}^v)$ and $p(s^p|\boldsymbol{\rho}^p)$ over visual and proprioceptive positions, in their neural activities $\boldsymbol{\rho}^v$ and $\boldsymbol{\rho}^p$, respectively. Representing a complete distribution in this manner may be essential for correctly combining information from the different sensory sources, if the statistical relationship between the inputs (visual and proprioceptive) and outputs (displacement s^c) is to be preserved.

Cue combination can then be cast as the problem of inferring the distribution over s^c based on the inputs $\boldsymbol{\rho}^v$ and $\boldsymbol{\rho}^p$, that is, $p(s^c|\boldsymbol{\rho}^v, \boldsymbol{\rho}^p)$. Given that the inputs specify probability distributions $p(s^v|\boldsymbol{\rho}^v)$ and $p(s^p|\boldsymbol{\rho}^p)$ according to some encoding scheme, the Bayes-optimal method of performing inference in this approach is specified as follows (Zemel & Dayan, 1997):

$$p(s^c|\boldsymbol{\rho}^v, \boldsymbol{\rho}^p) = \int_{s^v, s^p} p(s^v|\boldsymbol{\rho}^v) p(s^p|\boldsymbol{\rho}^p) \times p(s^c|s^v, s^p) ds^v ds^p \quad (20.1)$$

$$\propto p(s^c) \int_{s^v, s^p} p(s^v|\boldsymbol{\rho}^v) p(s^p|\boldsymbol{\rho}^p) \times p(s^v, s^p|s^c) ds^v ds^p, \quad (20.2)$$

where $p(s^c)$ is the prior distribution over s^c. We will assume that the information in the different inputs is independent and that there is a probabilistic generative model that specifies how the distribution $p(s^v, s^p|s^c)$ over possible visual target and proprioceptive finger locations is produced, for any given displacement s^c. Eq. 20.2 establishes the ideal observer, or the standard by which inferences about the distribution over s^c should be judged.

Computations in the network are guided by this probabilistic formulation. The feedforward connections determine how $\boldsymbol{\rho}^v$ and $\boldsymbol{\rho}^p$ are combined to produce $\boldsymbol{\rho}^c$, such that $\boldsymbol{\rho}^c$ defines a distribution that approximates Eq. 20.2 as closely as possible. Figure 20.1 illustrates the generative and recognition operations, showing the activities, the distributions that they represent, and the various probabilistic relationships.

This formulation requires specifying a form for the decoder that determines how the distribution is derived from the activities, and then optimizing the network synaptic connections to approximate the ideal observer under the specified encoding and decoding schemes. One possible form of decoding is a kernel density

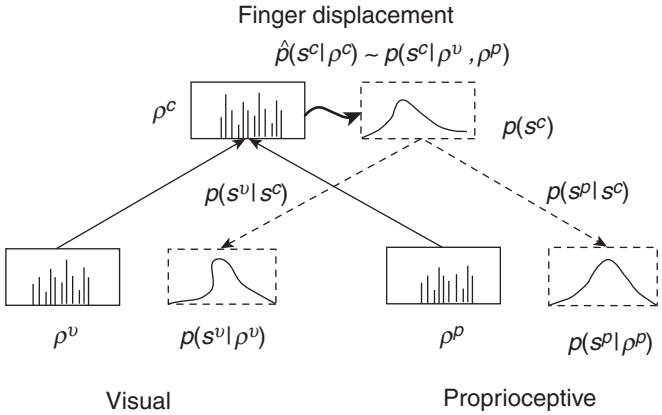

Figure 20.1 An example of a network formulated to combine population code representations of PDFs. The elements shown in solid lines depict explicit network components, including the population responses ($\boldsymbol{\rho}^v$, $\boldsymbol{\rho}^p$, $\boldsymbol{\rho}^c$). The elements shown in dashed lines are the implicit components, i.e., the probabilistic information represented by the explicit network components. The probabilistic combination provides the target for the network population codes and weights; the decoded distribution $\hat{p}(s|\boldsymbol{\rho})$ should match the posterior distribution $\hat{p}(s|\boldsymbol{\rho})$ obtained by Bayesian inference given the input PDFs $p(s^v|\boldsymbol{\rho}^v)$ and $p(s^p|\boldsymbol{\rho}^p)$.

estimate, in which the PDF is a sum of PDFs proposed by each neuron in the population weighted by its activity (Anderson, 1994); a second is a more optimal but complicated maximum-likelihood scheme (Zemel et al., 1998). An alternative decoding approach decodes the PDF as a log-linear combination of the PDFs proposed by each neuron (Hinton, 1999):

$$\hat{p}(s|\boldsymbol{\rho}) \propto \prod_i \hat{p}_i(s|\boldsymbol{\rho}_i)^{\boldsymbol{\rho}_i} = \exp\left(\sum_i \boldsymbol{\rho}_i \log \hat{p}_i(s|\boldsymbol{\rho}_i)\right). \quad (20.3)$$

We have applied this *product-of-experts* formulation (Huys, Zemel, Natarajan, & Dayan, 2007; Natarajan, Huys, Dayan, & Zemel; 2008) and shown that this scheme has the attractive feature of allowing each neuron to be considered independently of the others, while permitting a wide variety of PDFs. Note in particular that this product representation can produce much more peaked distributions than the kernel density estimate, which involves a linear combination of PDFs.

Given this decoder, computation in neural circuitry can be optimized such that applying this simple decoder to the network outputs will approximate the target distribution (Eq. 20.2) as faithfully as possible. An important point is that we are not proposing that neural circuits actually perform decoding at each level. Rather, the motivation for using this simple decoder is to explore whether a coding scheme can be found that will approximate proper inference, when the information about the relevant variable(s) at each level can be easily accessed, as in a log-linear decoding scheme.

This framework, along with this product-of-experts decoder, bears an interesting relationship to other recent proposals, notably the probabilistic population codes of Pouget and colleagues (Ma, Beck, Latham, & Pouget, 2006). Adopting the same decoder, they show that under some limited assumptions about the individual neuron's PDFs ($\hat{p}_i(s|\boldsymbol{\rho}_i)$), several natural computations can be carried out by linear operations on activities in the input populations.

DYNAMIC PROBABILISTIC POPULATION CODES

Time-Varying Continuous Inputs

The formulation described earlier readily applies to static cue-integration problems, in which cue

weights are assumed to remain constant over time. In this section, we consider a more realistic situation in which the input forms a continuous stream and the underlying variable of interest follows some trajectory. Hence, information from each cue can accrue over time in ways that affect how the different cues are combined dynamically.

To consider continuous, dynamic inputs, we switch the description of the neural activity from rate based to spike trains. We also adopt a different version of the decoder, which ignores correlations in the inputs and treats each spike independently, linearly integrating the information conveyed by them. The log-linear spike decoder (Hinton & Brown, 2000) fits these desiderata, and it can be seen as a simple extension of the product-of-experts decoder described in the previous section to spiking inputs, where the effect of past spikes decays exponentially over time. Here again, the motivation for this decoder is that the information necessary to perform proper probabilistic inference is available in a form that biological neurons might reasonably be expected to access.

Deriving an Optimal Time-Varying Target Distribution Consider a dynamic stimulus variable, say the position of a visual target, that evolves over time forming a *trajectory* defined by $\vec{s}_T = \{s_1, \cdots, s_t, \cdots, s_T\}$, where s_t is the position at time t. Thus, $\vec{s}_T$ defines the trajectory through which this object moves in the world. Note that we employ a discrete time representation. A population of neurons $i = \{1, 2, \ldots, N\}$ that responds selectively to the position generates a population spike sequence $\vec{\xi}_T = \{\xi_1, \cdots, \xi_t, \cdots, \xi_T\}$, where ξ_t is a binary spike vector at time t, such that ξ_t^i is 1 when neuron i spikes at time t.[1]

We assume that the spikes $\vec{\xi}_T$ are generated by an encoding model $P(\vec{\xi}_T|\vec{s}_T)$ that specifies the probability of a particular population spike sequence being evoked by the trajectory $\vec{s}_T$. Since neural spiking is stochastic and not capable of representing the stimulus value exactly, we consider the population response to implicitly define a distribution over likely stimulus trajectories. Then, the posterior distribution $p(s_T|\vec{\xi}_T)$ over stimulus positions can be estimated as a Bayesian inverse of the respective encoding model. Thus, we adopt the filtering goal of estimating the posterior distribution over trajectories implied by the input spikes, rather than the prediction one of making an estimate about the future course of the trajectory.

Spike-Generation Model We adopt a standard (tuning curve plus noise) spike-generation model for the set of all the J spikes $\xi_\varsigma \equiv \{\xi_{t_j}^i\}_{j=1}^J$ at times $0 < \{t_j\}_{j=1}^J \leq T$ evoked by the trajectory $\vec{s}_T$, where ς is a collection of all spike times. Under this model, the population response is governed by neurons whose expected responses are defined by partially overlapping Gaussian tuning functions, and whose stochastic observed responses are modeled as Poisson variables. The spikes are probabilistically related to the stimulus by means of a tuning function defined for each neuron i in the population as follows:

$$f_i(s_{t_j}) = r_{\max} \exp\left(-\frac{(s_{t_j} - \theta_i)^2}{2\sigma^2}\right), \qquad (20.4)$$

where s_{t_j} is the value of the stimulus variable at time t_j, θ_i is the preferred stimulus value of neuron i, $r_{\max}$ is the maximum input firing rate and σ is the tuning width. By this simple Gaussian definition, each neuron fires maximally at its preferred value θ_i and the activity drops off monotonically according to σ, as the stimulus s_{t_j} drifts away from θ_i.

The actual observed spikes are generated as inhomogeneous and instantaneous Poisson processes governed by the tuning functions (Barbieri et al., 2004; Brown, Frank, Tang, Quirk, & Wilson, 1998):

$$P(\vec{\xi}_\varsigma|\vec{s}_T) \propto \left(\prod_j f_{i(j)}(s_{t_j})\right) \exp\left(-\sum_i \sum_t f_i(s_t)\right). \qquad (20.5)$$

This spiking model entails some assumptions about the tuning properties and response of the input neurons. In our abstraction, the spikes

are conditionally independent events given the stimulus and the current response is independent of past activity. The tuning properties of the input population are constrained such that each neuron has the same maximum firing rate and tuning width. Furthermore, the tuning functions are assumed to span the stimulus state-space evenly and densely such that $\sum_i \sum_t f_i(s_t)$ is approximately a constant. This leads to:

$$P(\vec{\xi}_\varsigma|\vec{s}_T) \propto \prod_j f_{i(j)}(s_{t_j}). \tag{20.6}$$

Substituting for the tuning function from Eq. 20.4, the input spike generation model is then:

$$P(\vec{\xi}_\varsigma|\vec{s}_T) \propto r_{\max} \times \exp\left(-\frac{\left(\vec{s}_\varsigma - \vec{\theta}(T)\right)^T \left(\vec{s}_\varsigma - \vec{\theta}(T)\right)}{2\sigma^2}\right). \tag{20.7}$$

where $\vec{s}_\varsigma$ is the vector of all the stimulus positions ordered by spike time, and $\vec{\theta}(T)$ is the corresponding vector of preferred stimulus values of the spiking neurons.

Ideal Observer In Huys et al. (2007) we investigated the computational consequences of probabilistically encoding such stimulus trajectories in the spiking activity of populations of neurons. We summarize the basic setup and findings here. We first specify a prior over trajectories. The trajectories are defined over continuous time, and they are assumed to be drawn from a Gaussian process (GP). This means that for any finite collection of times $\Im = \{t_i\}$, the stimulus values $s_\Im$ are drawn from a multivariate Gaussian distribution with mean m and covariance matrix C:

$$p(\varphi_{(0,T)}) \sim N(\boldsymbol{m}, C) \quad C_{t_i t_j} = c \exp\left(-\alpha \left\| t_i - t_j \right\|^\zeta\right), \tag{20.8}$$

where C is the matrix of $C_{t_i t_j}$ at the discretized times. Different values of ζ determine the different classes of prior distributions over trajectories, and c parameterizes the overall scale of the process. The value of α scales the temporal extent of the interactions in the stimulus.

Under the assumptions described above, the posterior distribution for an observed population spike train can be derived as a Gaussian distribution with a mean $\mu(T)$ that is a weighted sum of the preferred positions of neurons that fired, and a variance $\nu^2(T)$ that depends only on C and the tuning width σ^2:

$$\mu(T) = \boldsymbol{k}(\vec{\xi}_\varsigma, T) \cdot \vec{\theta}(T) \tag{20.9}$$

$$\nu^2(T) = C_{TT} - \boldsymbol{k}(\vec{\xi}_\varsigma, T) \cdot C_{\varsigma T}. \tag{20.10}$$

Here $\vec{\theta}(T)$ is the vector of preferred stimulus values of the spiking neurons, ordered by spike time, ς is a collection of all spike times, $C_{TT} = c$ is the static stimulus variance at the observation time, and $C_{\varsigma T}$ is a vector of the cross-covariance between the spike times and observation time T. Recall that C depends only on the times of spikes, not on their identities. The posterior variance is similar to a Kalman filter and depends only on when data are observed, not what data. The weight on each spike depends strongly on the time at which the spike occurred:

$$\boldsymbol{k}(\vec{\xi}_\varsigma, T) = C_{T\varsigma}(C_{\varsigma\varsigma} + \boldsymbol{I}\sigma^2)^{-1}. \tag{20.11}$$

Here, $C_{\varsigma\varsigma}$ is the covariance matrix of the stimulus at all the spike times. A spike that occurred in the distant past will be given small weight.

Even under such a constrained and analytically tractable formulation of the encoding model (Eq. 20.5), the population spike trains tend to be sparse, so decoding trajectory information as a Bayesian inverse of the encoding model turns out to be ill posed. Prior information about what trajectories are likely plays a critical role. For the simple GP prior, the form of the dynamics is controlled by the various parameters, and particularly by the parameter ζ. If $\zeta = 1$, stimuli undergo Markovian dynamics, and the posterior distribution can be written in a recursive form that depends on only current spikes and a single population vector summarizing information in past spikes. But when $\zeta = 2$, stimulus correlations extend further back in time, producing stimuli with *smooth*

autocorrelations. This regime is the most ecologically relevant, as natural trajectories tend to vary smoothly over time. In this case, however, a simple recursive decoding is not possible, as instead decoding couples together the spikes, which means that the entire population spiking history is required to properly interpret a new spike.

Thus, our analyses show that neural correlations induced by naturalistic, smooth stimuli lead to a decoding problem that can only be resolved by access to information that is nonlocal both in time and across neurons. This makes trajectory inference computationally very hard. In a network of neurons, the downstream neural population that obtains these spikes as inputs faces a complicated inference problem, because computational and behavioral constraints do not permit retaining the full spiking history.

Optimizing a Network to Approximate Time-Varying Target Distributions

We now turn to the network version of the dynamic population-code combination problem, that is, the continuous analog to Figure 20.1. In Natarajan et al. (2008), we considered the distribution produced by the ideal observer in the previous section as the optimal target distribution for the network. The decoder for the population spikes was a spike-based version of the product of experts, in which each spike is considered independently of the others in the spike sequence, and spikes from different neurons can be combined without taking correlations into account. We then formulated a recurrent network, the connection strengths of which can be optimized so that this simple decoder approximates the dynamic target distribution. The details of the recurrent network and the decoding model are provided next.

Nonlinear Recurrent Network Approximating Optimal Inference Let $j = \{1, 2, \ldots, M\}$ be a population of recurrently connected output neurons in a network, receiving inputs $\vec{\xi}_T$ from the population $i = \{1, 2, \ldots, N\}$. We ascribe simple dynamics to the recurrent population, similar to the general spiking-neuron model of Kistler, Gerstner, and van Hemmen (1997). The output population spikes are specified by $\vec{\boldsymbol{\rho}}_T$, where $\rho_t^j = 1$ if neuron j spikes at time t. Like the inputs, the output spikes also convey information about $\vec{s}_T$.

The strength of the synaptic connection from an input neuron i to an output neuron j is characterized by the weight W_{ij}, and lateral connections between neurons j and k in the output population by U_{jk}. This scheme is illustrated in Figure 20.2. In a discrete time representation, the internal state (analogous to the membrane potential) of a neuron j at time T can be characterized by a continuous variable h_T^j:

$$h_T^j = \sum_{\tau=0}^{T-1} \sum_{i=1}^{N} \xi_{T-\tau}^i W_{ij} \eta(\tau) + \sum_{\tau=0}^{T-2} \sum_{k=1}^{M} \rho_{T-\tau-1}^k U_{kj} \eta(\tau). \tag{20.12}$$

The term $\eta(\tau)$ specifies the effect of a past spike on the current membrane potential; this typically has a form where temporally distant spikes have diminishing effects compared to more recent ones. Note that the feedforward and lateral weights can be negative. Thus, h_T^j is a signed variable where large negative values make spikes unlikely; large positive values make them likely.

For each neuron j in the population, its response is governed by a stochastic binary spiking rule:

$$P(\rho_T^j = 1) = \sigma(h_T^j). \tag{20.13}$$

The spikes are generated independently among the population. This implies that the probability of any population binary (spike/no-spike) vector ρ_T at time T is:

$$P(\vec{\boldsymbol{\rho}}_T | \vec{\xi}_T) = \prod_j \sigma(h_T^j)^{\rho_T^j} (1 - \sigma(h_T^j))^{(1-\rho_T^j)}, \tag{20.14}$$

where the sigmoid function is defined by $\sigma(h_T^j) = \frac{1}{1+\exp(-h_T^j)}$.

Note that the dynamics can be simplified considerably if the influence of past spikes is defined as:

$$\eta(\tau) = \exp(-\beta\tau), \tag{20.15}$$

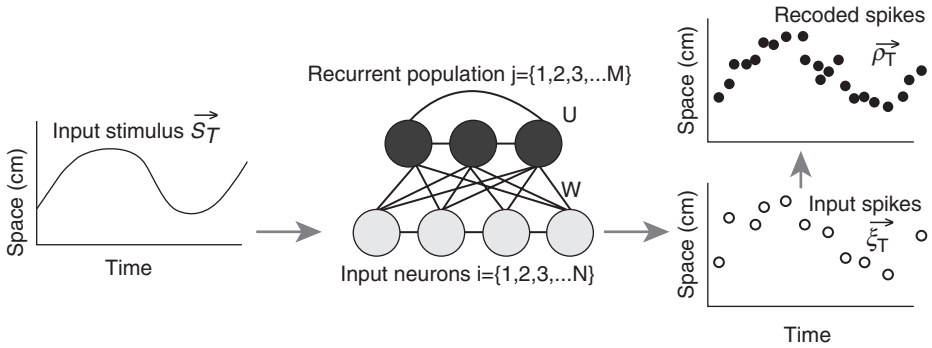

Figure 20.2 Schematic description of our network. Sensory *input* neurons $i = 1, \ldots, N$ generate population spikes in response to a stimulus trajectory. Downstream *output* population $j = 1, \ldots, M$ receives the input via feed-forward synapses **W**; recurrently processes the input using learned lateral synaptic weights **U** to recode the input into a representation that is independently decodeable by downstream neurons. Note the convention for this and subsequent figures: the neurons are arranged based on their preferred stimulus values, and a circle indicates that the neuron at that position spiked at the particular time.

where β is the *temporal decay constant*. Using an exponentially decaying function to characterize the postsynaptic potential of a neuron allows us to express the dynamics of the output population in a recursive formulation:

$$h_T^j = \sum_i \xi_T^i W_{ij} + \sum_k \rho_{T-1}^k U_{kj} + \eta(1) h_{T-1}^j. \tag{20.16}$$

Then the probability of a population spike vector is a function solely of the membrane potential and population outputs at the previous time step, and the current input spikes:

$$P(\boldsymbol{\rho}_T | \vec{\xi}_T) = P(\boldsymbol{\rho}_T | h_{T-1}; \boldsymbol{\rho}_{T-1}; \xi_T). \tag{20.17}$$

Figure 20.2 illustrates our network formulation.

An Independent Probabilistic Decoder The decoding model specifies the probabilities of various stimulus trajectories based on the spike sequence. Rather than considering the complicated space of possible trajectories given an entire spike train, we focus on an instantaneous version of the problem. In mathematical terms, our aim is to decode at time T a distribution over the stimulus position at that time, s_T, given all the spike observations up to that particular time, $\vec{\boldsymbol{\rho}}_T = \{\boldsymbol{\rho}_1, \cdots, \boldsymbol{\rho}_T\}$; that is, we want to infer $\hat{p}(s_T | \vec{\xi}_T)$ to approximate closely the true posterior distribution $p(s_T | \vec{\xi}_T)$. In this scheme, the total effect of the spike-train $E(s, T, \vec{\boldsymbol{\rho}}_T)$ is specified as a linear combination of the individual spikes:

$$E(s, T, \vec{\boldsymbol{\rho}}_T) = \sum_{j=1}^{M} \sum_{\tau=0}^{T-1} \kappa_{\text{std}}(j, s, \tau) \rho_{T-\tau}^j, \tag{20.18}$$

where $\kappa_{\text{std}}(j, s, \tau)$ is the spatiotemporal decoding kernel defined as being uniform and separable; uniform in that the shape of the kernel is the same for the entire population, and at all times, and separable as $\kappa_{\text{std}}(j, s, \tau) = \phi_j(s)\psi(\tau)$, where $\phi_j(s)$ is the spatial component and $\psi(\tau)$ is the temporal component. Then the spatiotemporal contribution of the spikes (Eq. 20.18) can be rewritten as follows:

$$E(s, T, \vec{\boldsymbol{\rho}}_T) = \sum_j \left[\sum_{\tau=0}^{T-1} \psi(\tau) \rho_{T-\tau}^j \right] \phi_j(s). \tag{20.19}$$

The spatial dimension of the kernel is parameterized as

$$\phi_j(s) = \frac{|s - s_j|^2}{\omega}, \tag{20.20}$$

where ω is the projective width of the neurons and s_j is the preferred stimulus value of neuron j.

Thus, $\phi_j(s)$ defines the influence of the output of a particular neuron j in the population $j = \{1 \ldots M\}$, on the spatial interpretation of the underlying variable.

The temporal dimension of the kernel is specified as an exponential decay (analogous to the postsynaptic potential of the neuron, η in Eq. 20.16): $\psi(\tau) = \exp(-\gamma\tau)$. The parameter γ, which we refer to as the *temporal integration constant*, specifies the timescale for synaptic integration in the readout. It effectively controls what effect the spiking of a neuron at one time has on the interpretation at future times. Hence, the kernels are very simple, specified using just three parameters: the preferred stimulus value s_j and spatial projective width ω for neuron j, and the temporal integration constant γ. The approximating probability distribution over stimulus trajectories $\hat{p}(s_T|\vec{\boldsymbol{\rho}}_T)$ is interpreted from the spiking activity as:

$$\hat{p}(s_T|\vec{\boldsymbol{\rho}}_T) \propto \exp(-E(s, T, \vec{\boldsymbol{\rho}}_T)). \qquad (20.21)$$

Eqs. 20.18 and 20.21 define our decoding hypothesis. Under this model, a downstream neuron simply has to add the responses of all the neurons that impinge on it; linear functions of these afferent spikes establish the information available in the population inputs. The information is thus readily accessible: if the convolution of spikes with the spatiotemporal kernels $\kappa_{\text{std}}(j, s, \tau)$ can be considered as defining the postsynaptic potentials of neurons, then information about the relative probabilities of stimulus values can be thought of as being accessed by simply adding (and exponentiating) the neuronal excitatory postsynaptic potential (EPSPs). Figure 20.3 illustrates the decoding method.

This choice of decoder is crucial, especially for ecologically relevant, smoothly varying trajectories. If we allow a complex decoder, for example, the Bayesian decoder $p(s_T|\vec{\xi}_T)$ described earlier, then the network needs access to all the input spikes (Huys et al., 2007). To prevent infinite regress, the decoder applied to the output spikes should access the type of information that biological neurons might reasonably be expected to extract. In our case, this is particularly a decoder that does not require access to all spikes at all times. We chose an independent decoder that integrates the information provided by spikes recursively, which provides a more plausible analog to how real neurons may integrate information.

Independent decoding of the input spikes $\vec{\xi}_T$ is far from optimal, especially for the smooth case. This decoder cannot by itself access the information in the input spikes that is due to the stimulus autocorrelations. Thus, this choice of output decoder shifts the onus of providing the information about the stimulus correlations from the decoder onto the network. In essence, the task of the network is to produce spikes that represent the information contained in the stimulus autocorrelations to make up for the fact that the decoder of its output spikes $\vec{\boldsymbol{\rho}}_T$ will, by itself, neglect any such information in the inputs $\vec{\xi}_T$.

In Natarajan et al. (2008), the overall aim of the network was to form output spikes such that the posterior obtained by decoding these spikes according to Eqs. 20.18 and 20.21 would faithfully approximate optimal decoding of input spikes $p(s_T|\vec{\xi}_T)$. The network encoded spatiotemporal stimulus regularities in the lateral connections of the output population. The network was trained to minimize the KL divergence between the decoded distribution $\hat{p}(s_T|\vec{\rho}_T)$ and the analytically-derived optimal distribution $p(s_T|\vec{\xi}_T)$. Results showed that the network obviated the need to maintain a spiking history by making the temporal prior information explicit in the spiking activity of the population. The recurrent structure thus kept the information contained in the past spikes online. It handled noisy and sparse input appropriately by re-expressing the information contained in the input into a set of spikes that were independently decodeable.

This form of temporal recoding can be considered a natural extension of the line-attractor scheme of Pouget et al. (1998). Both methods formulate feed-forward and recurrent connections so that a simple decoding of the output can match optimal but complex decoding applied to the inputs. The model presented here extends this approach to spiking networks,

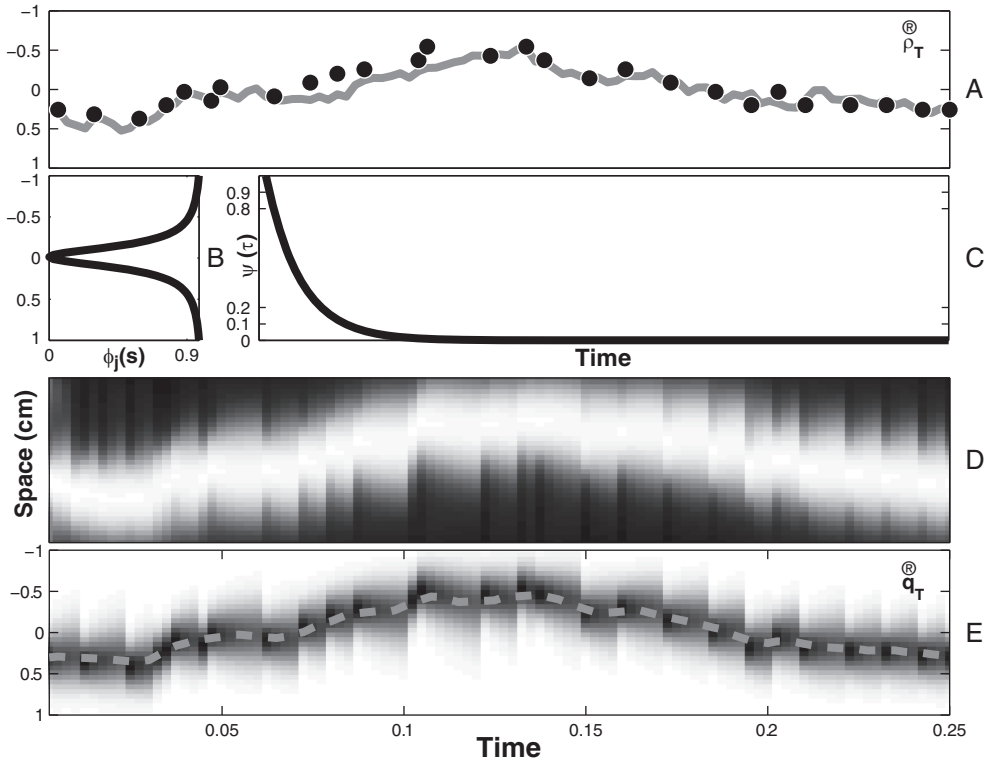

Figure 20.3 The independent decoding model. (A) Output spikes (solid black circles) of population $j = \{1, 2, \ldots M\}$ convey information about the stimulus trajectory (solid gray line). (B) and (C) An example of the form of spatial and temporal components of the decoding kernel. (D) Linear contribution of the population spikes through time. (E) Inferred posterior distribution (shaded in gray) over stimulus positions through time; posterior mean is in dashed gray line.

and more importantly, to dynamically varying inputs associated with a prior that must be learned.

Our method is also related to some other early proposals where the spiking activity of either a single neuron (Deneve, 2005) or a pair of neurons (Gold & Shadlen, 2001) is considered as reporting (logarithmic) probabilistic information about an underlying binary hypothesis. Another approach proposes that a population of neurons directly represents the (logarithmic) probability over the states of a hidden Markov model (Rao, 2004). Like Deneve (2005), we consider the transformation of input spikes to output spikes with a fixed assumed decoding scheme so that the dynamics of an underlying process is captured. Our decoding mechanism produces something like the predictive coding apparent in Deneve (2005), except that here, a neuron may be silent not only if it itself has recently spiked and thereby conveyed the appropriate information, but also if one of its population neighbors has recently spiked. This is explicitly captured by the recurrent interactions among the population.

APPLYING THE CODING SCHEME TO DYNAMIC CUE COMBINATION

In this section we consider how a neural system might employ the proposed coding scheme recursively, to dynamically weight and integrate information about a transient stimulus variable from different sensory modalities. Particularly, we evaluate the coding efficacy of

our approach when input to the model has the same characteristics as the output representation produced by it; that is, information can be decoded in an (approximately) optimal manner even when the decoder ignores correlations in the inputs and treats each spike independently, linearly integrating the information conveyed by them (Eq. 20.21). We examine whether and how the model can approximate proper inference in a challenging version of a cue-combination task, one in which the inputs and their reliabilities are continuously varying through time, and the information is communicated by spiking neural populations.

The advantage of independent interpretability of spikes is not confined to decoding. Neural computations underlying integration of information from different modalities (e.g., Ernst & Banks, 2002; Hillis, Ernst, Banks, & Landy, 2002) or sensorimotor combination (e.g., Körding & Wolpert, 2004) become straightforward, requiring only an addition in the log domain or multiplication of single-neuron (Chance, Abbott, & Reyes, 2002; Poirazi, Brannon, & Mel, 2003; Salinas & Abbott, 1996) or population (Deneve et al., 2001) activity.

In this section, we look beyond decoding as the canonical task; we explore how our dynamic coding scheme can be manipulated easily and integrated efficiently in a hierarchical manner to perform uncertainty-sensitive neural computations through time. Specifically, we consider how we can extend the framework presented earlier (see section on "Approximating Probabilisitic Inference in Population-Code Networks") to handle continuous, dynamic stimulus variables.

Dynamic Sensorimotor Task

We focus on a version of the sensorimotor integration task where Körding and Wolpert (2004) explored the consequence of manipulating the reliability of visual feedback during reaching. Figure 20.4 illustrates the experimental setup where a starting position and a target position are marked using a white and green circle, respectively, indicated at the bottom and top of a horizontal plane. Subjects were required to reach to the visual target from the starting position, within the virtual-reality environment that blocked the hand from view. A cursor was shown before the start of the movement; it was then hidden except for one point of visual feedback midway through the trial. At the onset of movement, unbeknownst to the subjects, the cursor was laterally displaced relative to the true

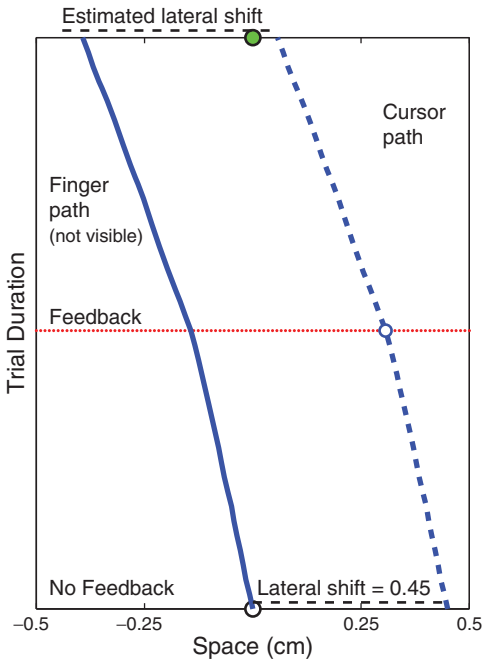

Figure 20.4 Experimental setup. 1-d position of the finger on the screen is represented along the x-axis, and the time-course of the trial along the y-axis. The unfilled circle at the bottom of the screen indicates the starting position of the cursor; the green filled circle at the top of screen shows the target position. The task is to reach the target as closely as possible. The invisible finger trajectory is indicated by a solid blue line; the true cursor position by a dashed blue line. On the trial shown here, the cursor is laterally displaced relative to the finger position by 0.45 cm. The mid-point of the trial is indicated by the horizontal black dashed line at the center of the figure; the blue unfilled circle along the cursor path indicates brief visual feedback. Based on this feedback, there is compensatory correction in the finger trajectory from the mid-point onwards. Further details of the setup are provided in the text.

finger position by an amount drawn randomly from a known prior distribution.

Midway through the movement, brief visual feedback about the cursor position was provided. On each trial, the reliability of the visual feedback was a random draw of positions from a Gaussian distribution, with position indicated using dots. Across trials, the feedback was varied across four conditions, σ_v^0 (a single precise white dot), σ_v^M, σ_v^L (small translucent dots distributed as a two-dimensional Gaussian with standard deviations of 1 and 2 cm, respectively), and σ_v^∞ (no feedback). The task was to land the cursor as close to the target as possible; feedback about the final cursor position was provided only in the high-reliability condition σ_v^0.

Subjects were trained for about 1000 trials in the task to ensure that they encountered many samples of the lateral shift drawn from the underlying prior distribution. Upon testing, subjects were found to generate compensatory corrections in their finger trajectories in the second half of the movement, in order to offset the imposed lateral displacement of the cursor. The authors assumed a Bayes-optimal strategy and inferred the priors used by subjects based on the measured shifts in final estimates of target position.

On test trials where no visual feedback was given, subjects appeared to learn the prior distribution over offsets: The estimated displacement was equivalent to the mean of the prior distribution. On trials in which feedback was provided, subjects combined prior knowledge of the distribution with sensory (visual) evidence in a manner that was appropriately sensitive to the degree of noise in the visual feedback. The authors accounted for their results within this framework as follows: Adjustments to ongoing motor commands were dependent on both the reliability of sensory inputs and on an estimation of prior probabilities of displacements in a manner consistent with Bayesian computation. Other studies (e.g., Tassinari, Hudson, & Landy, 2006) have employed related experimental paradigms without assuming any strategy, and they have quantitatively demonstrated that human performance does indeed result from a nearly optimal combination of sensory and prior information.

Dynamic Cue-Combination Network

The objective for our recurrent network model is to compute the distribution over cursor positions on the screen based on information about finger position and visual information about the cursor, each having varying noise levels.

We set up a hierarchical network much like that described earlier (see section, "Approximating Probabilisitic Inference in Population-Code Networks"), where two independent sensory populations (visual and proprioceptive) are connected by means of feedforward weights to a third population of downstream neurons (Fig. 20.5). The first layer re-represents its inputs into a form that is independently decodeable using our log-linear scheme. It is these spikes that serve as inputs to the downstream population, which in turn combines the information with knowledge of the prior distribution over the lateral shifts that is learned in the network synaptic connections. A posterior distribution is derived from the output spikes of the downstream population using the same decoding scheme. This makes the approach recursive in that the spikes at every level may be interpreted

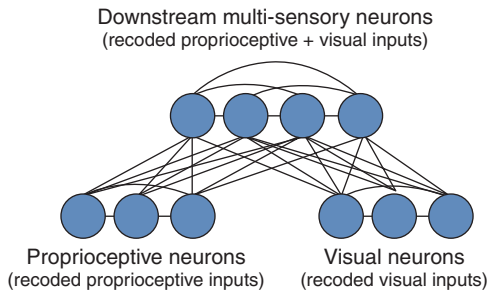

Figure 20.5 Dynamic combination network. Recurrently connected uni-sensory populations of proprioceptive and visual neurons recode their inputs into independently decodeable representations which then serve as inputs to a third recurrently connected downstream population that combines them. Feed-forward weights connect the two levels of the network. In this recursive scheme, inputs and outputs to the network have consistent semantics, i.e., they are independently interpretable using a fixed decoding scheme.

using the same decoding approach. The mean of the posterior distribution provides the final estimate of the cursor position on the screen. In our network model, this mean can be thought of as computing the "motor command" for compensation in the finger position. Details of our simulation, including the setup of each of these elements of recursive computation, are provided next.

Network Preprocessing: Log-Linear Decodeable Unisensory Inputs We create input spikes for our network by applying the recurrent-network approach to Gaussian-Poisson spikes, as described earlier (see section on "Spike-Generation Model"), separately for visual and proprioceptive inputs. Neural populations $i_v = \{1, \cdots, N_v\}$ and $i_p = \{1, \cdots, N_p\}$ are defined such that the preferred stimulus positions of the neurons are evenly distributed in the stimulus space. The neurons are individually tuned to respond across a certain range of the stimulus space. The visual $\vec{\xi}^v$ and proprioceptive inputs $\vec{\xi}^p$ are determined by Gaussian tuning curves with Poisson variability (Eq. 20.4). In the simulations that follow, sparse inputs are generated by assigning a low value to the maximum input firing rate $r_{\max} = 0.144$ Hz and tuning width $\sigma = 0.1$ cm in Eq. 20.4.

Laterally connected visual and proprioceptive neural populations $j_v = \{1, 2, \cdots, M_v\}$ and $j_p = \{1, 2, \cdots, M_p\}$ have preferred stimulus positions distributed evenly across the stimulus space and receive input spikes $\vec{\xi}^v$ and $\vec{\xi}^p$, respectively, in response to the visually observed cursor position and the finger position. The lateral connections of the individual populations are indicated by U_v, U_p; the projective width used in our simulations is $\omega = 0.2$ (Eq. 20.20). Each population individually learns to recode its inputs into representations $\vec{\rho}^v$ and $\vec{\rho}^p$ that are independently decodeable.

For training the preprocessing systems, stimulus trajectories $\vec{s}^p$ and $\vec{s}^v$ are drawn from the Gaussian process prior distribution (Eq. 20.8). The parameter ζ of the covariance matrix C determines the smoothness of the trajectories. By changing the value of ζ we can consider different classes of GP priors, each defining the density over a set of trajectories with similar spatiotemporal dynamics. Since a hallmark of motor behavior is that movements are smooth, relatively straight trajectories (Flash & Hogan, 1985; Morasso, 1981), we choose parameters of the Gaussian process prior such that the resulting trajectories have smoothly varying dynamics. The particular values used in the simulation are specified in Figure 20.6.

This spike-generation model leads to a simple ideal observer to produce the training signal: the posterior distributions $p(s_T^p|\vec{\xi}_T^p)$ and $p(s_T^v|\vec{\xi}_T^v)$ for an observed population spike train, as described above (see section on "Ideal Observer"). Approximate posterior distributions $\hat{p}(s_T^p|\vec{\boldsymbol{\rho}}_T^p)$ and $\hat{p}(s_T^v|\vec{\boldsymbol{\rho}}_T^v)$ are decoded following Eqs. 20.18 and 20.21. Then each population $j_v = \{1, 2, \cdots, M_v\}$ and $j_p = \{1, 2, \cdots, M_p\}$ individually learns to recode its inputs by minimizing the divergences between the true and approximate posterior distributions over finger and cursor positions: $D_{KL}(p(s_T^p|\vec{\xi}_T^p)||\hat{p}(s_T^p|\vec{\xi}_T^p))$ and $D_{KL}(p(s_T^v|\vec{\xi}_T^v)||\hat{p}(s_T^v|\vec{\xi}_T^v))$, where $D_{KL}(p(s)||\hat{p}(s)) = \sum_s p(s)\log(p(s)/\hat{p}(s))$ is the Kullback-Leibler (KL) divergence, or relative entropy between the true $p(s)$ and model distribution $\hat{p}(s)$.

Note that beyond labeling the two unisensory populations as visual and proprioceptive, we have not detailed how the two modules differ from each other in the information they represent. Typically, in such a sensorimotor context, the proprioceptive neurons are simulated to encode information about finger position in joint-angle space, and the visual neurons would encode information in retinocentric space. However, neural mechanisms underlying sensorimotor transformations have been explored extensively, often formalized as coordinate transformations between body-centric and eye-centric representations (Baraduc, Guigon, & Burnod, 2001; Deneve et al., 2001; Pouget et al., 2003). Hence, we assume that the visual and proprioceptive inputs have already been converted into a common frame of reference by some mechanism (the basis-function approach of Pouget et al. can be readily implemented under our scheme), and we explore differences in the dynamics of the two unisensory networks.

Recall that in our setup, there is a steady stream of proprioceptive inputs from the sensed location of the finger, but visual information is very sparse since it is displayed only as a brief flash as in the original experiment. Therefore, in generating the stimulus trajectories $\vec{s}^p$ and $\vec{s}^v$, we consider slightly different dynamics in specifying the Gaussian-process prior. Recall (see section on "Ideal Observer") that different values of ζ determine the different classes of prior distributions over trajectories. We generated the finger trajectories using faster, smoother temporal dynamics by setting $\zeta = 2$ in Eq. 20.8; the cursor trajectories were generated by setting $\zeta = 1$. Accordingly, the proprioceptive and visual inputs in our network reflect the differences in the input dynamics (see the parameter values below Fig. 20.6). Consequently, the learned parameter β, which specifies the dynamics of the spiking activity (Eqs. 20.15–20.16), is observed to be different for the two submodules. We elaborate on this in the section on "Simulation Results."

Training the Network The recoded visual and proprioceptive inputs are then fed to the next stage of recurrent processing by neurons that combine the information. It is in these feedforward and lateral weights (as well as network dynamics) that we expect the learned prior distribution over lateral shifts to be embedded. Therefore, the trajectories for this second stage of processing are generated such that there is compensation for the imposed lateral shift at the midpoint of each trial. The visual stimulus (cursor s_t^v) indicates the position of the finger s_t^p at any time t. The cursor s_t^v is shifted laterally from the finger position s_t^p by a value δ_{shift} drawn from the known prior distribution $p_{\text{shift}} \propto N(\mu_{\text{shift}}, \sigma_{\text{shift}})$ that carries information about the probabilistic structure of variations in the task.

Halfway through the trial, at $t = T/2$, the visual stimulus is displayed only for that timestep, with varying degrees of uncertainty for different trials: σ_v^0 when the exact position of the cursor $s^v_{(t=T/2)}$ is shown with no variability,

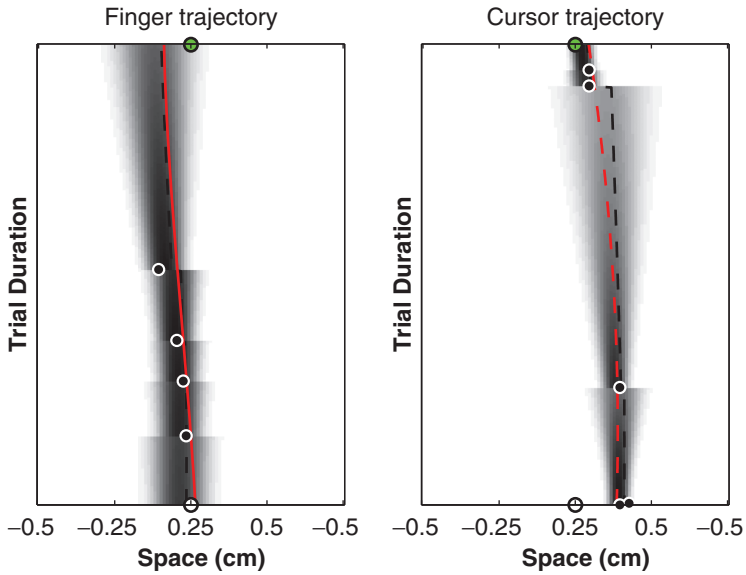

Figure 20.6 Generating visual and proprioceptive inputs with varying dynamics. The solid red line shows a sample finger trajectory generated from a GP prior with $\zeta = 2, \alpha = 0.05, c = 0.2$. The dashed red line is a sample cursor trajectory; $\zeta = 2, \alpha = 0.15, c = 0.5$. White circles shaded black indicate proprioceptive and visual input spikes in the left and right panels, respectively. The spatio-temporal distributions in gray-scale plot the approximate distributions derived from decoding proprioceptive (left) and visual (right) recoded outputs. The dashed white lines show the means of these posterior distributions.

and σ_v^M and σ_v^L when the cursor is displayed with medium and large variances. The idea is that based on this visual input, the subsequent finger positions change to compensate for the lateral shift. For this, we estimate the shift in the trial using Bayes rule, given the visual input and the known prior distribution. The posterior distribution over sensed lateral shifts in each trial is computed and the mean of this posterior is selected to be the estimate on that trial; the finger position is compensated by that amount.

The training signal for this stage is the true posterior distribution over cursor positions, derived following Eq. 20.2,

$$P(s_T^c|\vec{\boldsymbol{\rho}}_T^v, \vec{\boldsymbol{\rho}}_T^p) \propto P(\vec{s}_T^c) \int_{\vec{s}_T^v, \vec{s}_T^p} p(s_T^v|\vec{\xi}_T^v) p(s_T^p|\vec{\xi}_T^p) \times P(\vec{s}_T^v, \vec{s}_T^p|\vec{s}_T^c) d\vec{s}_T^v d\vec{s}_T^p. \quad (20.22)$$

Here, the combined stimulus prior $P(\vec{s}_T^v, \vec{s}_T^p|\vec{s}_T^c)$ can be seen as scaling the prior over displacements $P(\vec{s}_T^c)$. The network parameters are optimized to minimize the KL divergence between this distribution and the distribution derived from decoding its outputs $\vec{\boldsymbol{\rho}}^p$ according to Eq. 20.21. The proprioceptive inputs are received at a timescale that fits our general scheme well; the visual inputs are received just as a brief flash. The lateral connections in the top level integrate the inputs and maintain the belief state online.

SIMULATION RESULTS

In this section, we present results from simulating the network under various experimental conditions. The trained network from the previous section was tested on a set of 500 trials; in the following simulations, we assumed a prior distribution $p_{\text{shift}} \propto N(\mu_{\text{shift}}, \sigma_{\text{shift}})$, where $\mu_{\text{shift}} = 0.3$ and $\sigma_{\text{shift}} = 0.2$. We first analyze results on a single trial of the experiment followed by results summarized over the 500 test trials.

Figure 20.7 plots the estimated cursor distributions for three different stimulus conditions: σ_v^0, σ_v^M, and σ_v^L. In each vertical panel, the x-axis represents the stimulus space (i.e., finger/cursor position) and the y-axis represents the time steps in the course of the trial; each trial proceeds from the bottom up. The starting position of the finger movement is indicated by the black circle at the center of stimulus space (0 cm) and the target position by the circle shaded in green at the top of the panel. At the start of a trial, the cursor is shifted laterally by 0.45 cm. The true cursor position is indicated by the dashed line in red (σ_v^0), blue (σ_v^M), and white (σ_v^L). The estimated posterior distribution over cursor positions is plotted in grayscale for each step of the trial.

At the start of the trial the estimated posterior appears to rely heavily on the prior distribution $p_{\text{shift}}(\Delta)$ for all three conditions; the estimated mean is centered around the mean of the Gaussian prior distribution at 0.3 cm. During the first half of the movement, the variance of the posterior distribution is seen to grow with time. The only inputs during this phase are the proprioceptive inputs from the unseen hand and the visual input at $t = 1$; the increasing variance of the posterior distribution indicates increasing unreliability of the finger position estimate.

When visual feedback is presented midway through the trial, the posterior mean shifts toward the cursor position (0.45 cm for the trials presented here) and the posterior variance shrinks. This increased certainty about the position of the cursor reflects augmentation of the sensory inputs with new visual information. From this point on, the posterior distribution appears to rely more on the visual feedback and less on the learned prior information. The variance in general is less during the second half of movement compared to the first. One clear difference between the decoded distributions for the three stimulus conditions is that the posterior variance increases as the visual reliability decreases.

Figure 20.8 plots the estimated mean of the posterior distribution in dashed red, blue, and white lines for each of the conditions discussed earlier. The finger trajectory on the trial is indicated by the solid line. The panels on the right of the plot indicate the amount of visual noise for each condition. For the high-reliability condition, σ_v^0, the final estimate of the target location (i.e., the mean of the estimated posterior

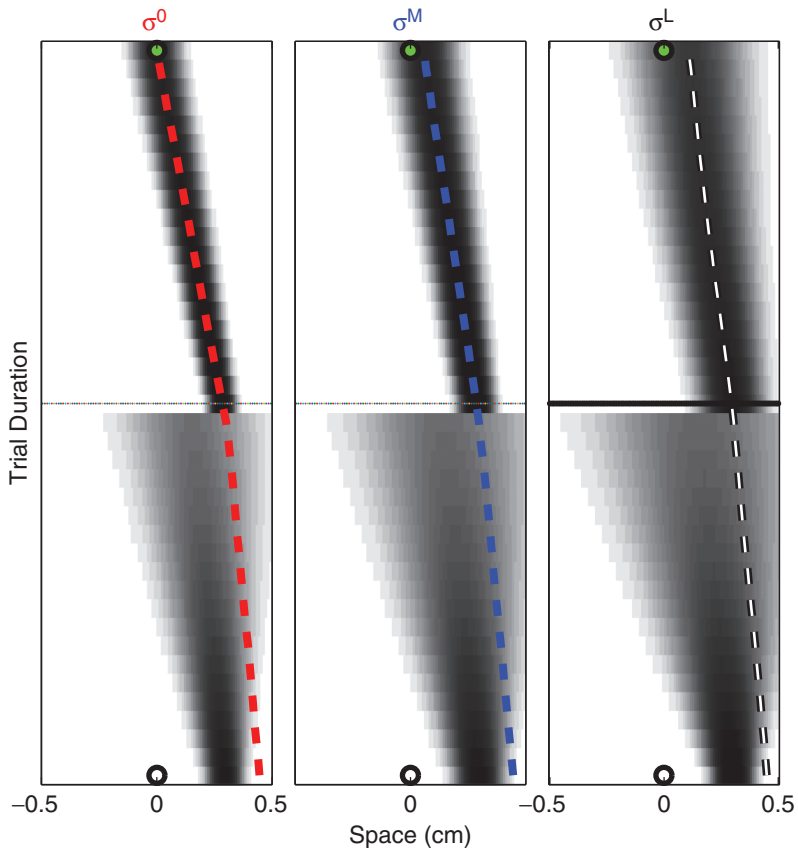

Figure 20.7 Simulation results. Estimated cursor distributions for three stimulus conditions. True lateral shift of visual cursor on all trials = 0.45 cm. Dashed lines in each panel indicate true cursor position: red—σ_v^0, blue—σ_v^M and white—σ_v^L. At each timestep, the estimated posterior distribution over cursor positions is plotted in gray-scale for all three stimulus conditions. At the start of the trial, the estimated posterior mean is centered around 0.3 cm, the mean of the Gaussian prior distribution over lateral shifts. Notice how for all three stimulus conditions, the posterior variance increases during the first half of the movement and shrinks at the mid-point (indicated by the horizontal dashed black line) when visual feedback is provided. At this point, there is also a sudden shift in posterior mean. These results are analyzed in detail in the text.

at the end of the trial) is right on target. For σ_v^M and σ_v^L, it is shifted away from the target location, toward the true displaced cursor position. The degree of this shift increases with increasing unreliability in visual feedback. These observations show that the sensory inputs were integrated with the learned prior in a manner that was appropriately sensitive to the degree of noise in the visual feedback: As the precision of the visual input decreases, the system relies more on prior knowledge of displacement statistics to adjust finger movement.

To quantify the system behavior, we calculate the average performance of the network over 500 trials. Figure 20.9 summarizes the final deviation of the estimated cursor position from the target, as a function of the true lateral shift on each trial. These values are averaged over the 500 trials for each condition. The results are qualitatively similar to those observed by Körding and Wolpert (2004): The slope of the plotted lines increases with visual uncertainty. For σ_v^0, the estimated cursor position lands on average at the true target position; with

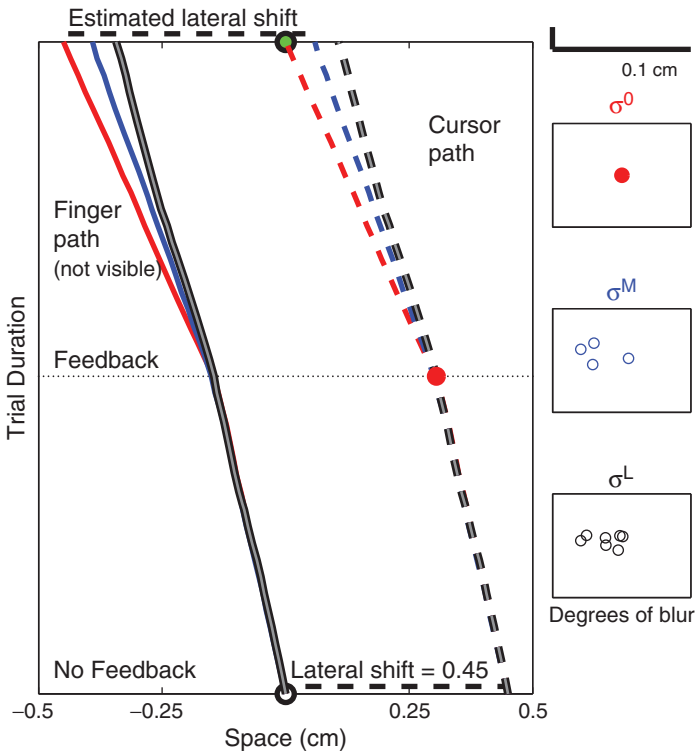

Figure 20.8 Estimated mean of the posterior distribution. Solid lines—finger trajectory; dashed lines—estimated posterior mean. Illustrations on the right indicate the degree of visual noise for each stimulus condition. For σ_v^0 (red line), the mean of the estimated posterior at the end of the trial is right on target. For σ_v^M and σ_v^L, (blue and black lines, respectively), it is shifted away from the target location, towards the true displaced cursor position. The degree of this shift increases with increasing unreliability in visual feedback. Results suggest that cue weighting is sensitive to uncertainty in the visual feedback.

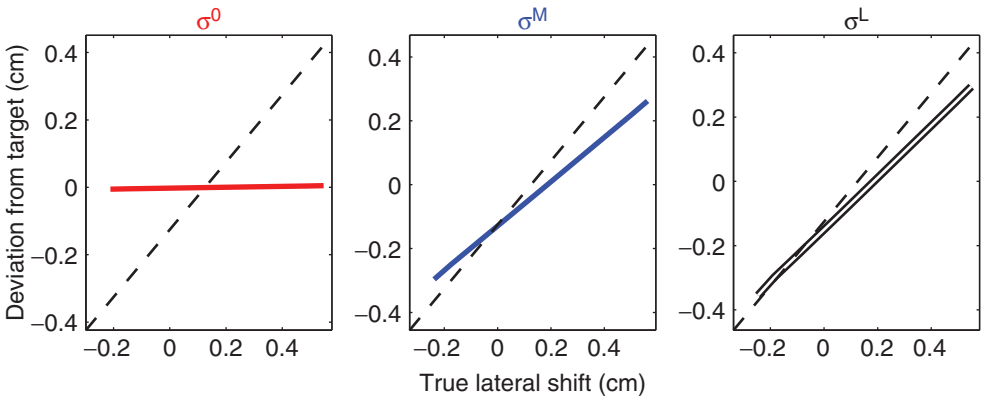

Figure 20.9 Summary results. Average network performance over 500 trials for each of the three conditions. The x-axis plots the true lateral shifts imposed on different runs of the simulations; the y-axis indicates the final deviation of the cursor from the target. The solid red, blue and black lines plot the final deviation of the estimated cursor position from the target for the three different stimulus conditions. For σ_v^0, this plot is almost a horizontal line, indicating that the estimated cursor position lands on average at the true target position. The slope of the blue and black lines increases with visual uncertainty. Results are qualitatively similar to those of Körding and Wolpert (2004).

increasing noisiness, the final cursor position is displaced. We conclude from these results that the sensory integration in our dynamic network was appropriately sensitive to the degree of noise in the visual feedback.

Network Preprocessing: Cue Differentiation

We can gain some insight into how the network achieves this behavior by examining its learned feedforward and lateral weights, temporal decay constant, and the decoding kernels. We first discuss the components of the preprocessing stage. Here, the learned feedforward connectivities for the visual and proprioceptive unisensory networks were qualitatively similar (not shown), displaying a strong local connectivity, with each input neuron having strong excitatory connections to neurons in the recurrent population having the same or very similar stimulus preference and inhibitory connections to their neighboring neurons.

Figure 20.10 plots the learned lateral connection strengths $\mathbf{U}_{\mathrm{v}}$ and $\mathbf{U}_{\mathrm{p}}$ for the recurrently connected visual and proprioceptive neurons. The plot is in the form of a matrix, where the rows $j_v = \{1, 2, \cdots, M\}$ and columns $k_v = \{1, 2, \cdots, M\}$ correspond to the output neurons spatially laid out in the stimulus space. Each cell of the weight matrix $\mathbf{U}$ is shaded according to the strength of the synapse $U_{j_v k_v}$. The connectivities learned by both networks are starkly different. We attribute this to the differences in temporal dynamics of the input stimulus (and therefore the dynamics of the neural inputs) to these networks.

Let us first examine the weights $\mathbf{U}_{\mathrm{p}}$ (Fig. 20.10, left panel). Each neuron is seen to strongly excite its immediate neighbors and also to inhibit the ones further away. The strength of excitation is strongest for neurons tuned to the center of the stimulus space and decreases as the neuron's stimulus preference moves toward the periphery. This is likely a direct consequence of the spatial range of the stimuli the network was trained on. The negative weights allow one neuron's spikes to decrease the membrane potential of another. So the population activity not only indicates where the stimulus is at any given time (by spiking when the stimulus is at the neuron's preferred position) but also actively indicates where the stimulus may not be.

The temporal decay constant adapts to a large value, $\beta = 2.42$ (Eqs. 20.15–20.16). This allows the postsynaptic potential of neurons in the recoding population to decay slowly with time. This means that temporally distant spikes have very slowly diminishing effects on the current population activity. The decay constant reflects the temporal extent of the interactions in the stimulus, specified by α in Eq. 20.8. This same rate of decay is applied to γ in the temporal kernel to decode the spikes; it implies that the influence of a spike persists for a while, having a slowly decaying effect on the representation at future times.

Next, we examine the weights $\mathbf{U}_{\mathrm{v}}$ (Fig. 20.10, middle panel). Each neuron has strong positive connections to some (but not its immediate) neighbors, evident in the positive weights. However, the learned temporal decay constant $\beta = 1.85$ is much lower than that of the proprioceptive network.

Learned Network Components

Next, we analyze the components of the network. The feedforward weights from the visual and proprioceptive ensemble to the recurrently connected multisensory neurons display strong local connectivity. The rightmost panel in Figure 20.10 plots the lateral weights learned during the combination stage, in the same matrix format described earlier. The positive synaptic weight patterns are off diagonal, indicating that at every time step, each neuron is signaling several (again, not immediate) neighboring neurons to fire at subsequent time steps during recurrent processing. The learned temporal decay constant is large again, $\beta = 2.38$.

Since the lateral weights and the decay constant are learned together in an iterative manner with the decoding kernels fixed, they influence each other very closely. Combined with the input spikes to the population, they influence the membrane potential h_T (Eq. 20.16) of neurons in the recurrently connected population according to how the input population connects to the recurrent population via the feedforward

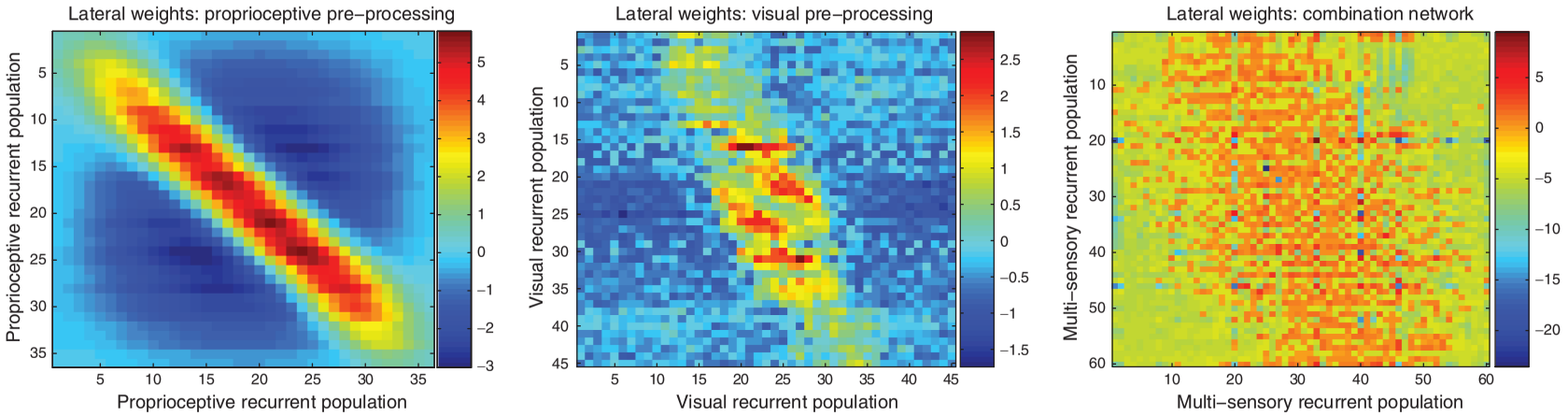

Figure 20.10 Understanding the lateral synaptic weights. Left panel: learned lateral weights $\mathbf{U}_p$, $\beta = 2.42$. Middle: lateral weights $\mathbf{U}_v$, $\beta = 1.85$. Right panel: lateral connections between the downstream multi-sensory neurons, $\beta = 2.38$. The connection patterns learned by the three populations are starkly different. These different network dynamics are a reflection of the differences in input stimulus dynamics; additionally, the multi-sensory neurons also learn to combine their two inputs. The color bars on the right indicate the strength of the lateral connections; warm colors correspond to positive weights and cold colors to negative weights. In the absence of input spikes, recurrent activity contributes to the persistent effect of previously observed spikes.

weights. In the absence of input spikes, recurrent activity contributes to the persistent effect of previously observed spikes. This triggers a cascaded pattern of activation among neurons of the recurrent population. This explains the increasing variance with time in the decoded distribution.

SUMMARY AND DISCUSSION

Our understanding of how the central nervous system might encode and interpret sensory input has significantly progressed in the last decades. Concomitantly, theoretical explorations have provided numerous hypotheses and results on how various ecologically relevant computations might be performed through feedforward and recurrent connections between neural populations (Deneve et al., 2001; Pouget et al., 1998). Our study is at the intersection of these two lines of work. We explore the characteristics of neural population codes desirable for uncertainty-sensitive, optimal downstream decoding of the represented information. We focus on time-varying inputs represented by a spiking network and attempt to understand dynamic population codes from a functional perspective, beyond the issue of stimulus representation.

Summary

In this chapter, we first reviewed our proposal for how uncertainty about time-varying sensory inputs can be represented in population spike times. In our scheme, recoding produces temporally independent spikes, and the decoding kernels convolve them in such a way that information originally encoded in spatiotemporal spike patterns can now be independently decoded to produce a smooth, inferred distribution. Our main contribution here was an extension of the coding scheme to develop a recursive, hierarchical formulation of the system.

The highlights of our proposal are as follows. We first specified a recursive application of our coding scheme; we demonstrated how the recoded information can be manipulated easily and integrated efficiently in two stages to causally reweight and integrate information about the stimulus variable from different modalities. One population encoded dynamic proprioceptive information while another represented visual information; both utilized the recoding scheme to efficiently represent a stimulus variable such as position. Then, a third population received these recoded spikes as inputs, combined them using the same representational scheme, and produced a spike-based representation of the posterior distribution over positions.

Second, in our formulation, the visual and proprioceptive subnetworks receive inputs of varying temporal dynamics. The proprioceptive neurons received inputs corresponding to smoothly varying finger trajectories; the visual neurons received sparser inputs corresponding to a visual stimulus (cursor position on screen) that was only briefly flashed midway through the experimental trial. The main computational advantage of using a fixed independent decoder such as ours is that rather than employing different decoding strategies for different stimulus dynamics, a neural population can recode the input representation into one that can be decoded independently regardless of stimulus dynamics. Our results demonstrated that the visual and proprioceptive subnetworks did in fact learn to reflect the different dynamics underlying their respective inputs. Additionally, our results showed that under this coding scheme, a downstream neural population (recurrently connected multisensory neurons) is able to appropriately compute with inputs from the different sources using the same fixed independent decoder.

The inputs to the visual and proprioceptive neurons were constrained to be sparse. This was meant to simulate those situations where the temporal dynamics of a stimulus are incompatible with (i.e., faster than) neural spiking dynamics. Such input sparsity makes a more challenging case for recoding; if spikes are dense relative to the movement in the stimulus (i.e., the likelihood term in Eq. 20.4 dominates either via very low noise σ or high firing rates $r_{\max}$), the contribution of the prior will be small and recoding may not be necessary. On the other hand, if spikes are sparse, the prior will be more important and approximations more costly; recoding will become important

due to the need to account for spatiotemporal stimulus dynamics and the desire to decode independently. Our choice of a sparse spiking regime implies that the complexity of the decoding task primarily derives from the smooth dynamics of the finger movements.

Third, our learning rule for the multisensory downstream population was derived by applying the independent decoder to the recoded unisensory inputs. The computational motivation for applying this rule recursively is that at each level of the hierarchy, additional information from other spiking populations is combined, such that the network is effectively consolidating the information in all of its inputs. This implies that the learning objective is to simply preserve the information in the combined inputs. This preservation is itself a form of computation, because it involves combining the distributions implied by the visual and proprioceptive unisensory inputs.

Finally, this model relates to experimental results from Körding and Wolpert (2004) using many simulations with smooth trajectories and varying visual noise levels. We demonstrated that our spiking cue-combination network can account for these data, yielding the same dependence of the final cursor position on the degree of noisiness in the visual feedback. Our implementation possesses some neural plausibility, since it employs a straight-forward nonlinear network requiring access to the type of information that biological neurons might reasonably be expected to extract.

Related Work

Several sensorimotor-transformation schemes have been proposed that exploit redundancy and broad tuning ubiquitous in the cortex to minimize the deleterious effects of noise (Baraduc et al., 2001; Salinas & Abbott, 1995) or to generate probabilistic population codes to represent Gaussian and more general forms of probability distributions to propagate uncertainty in neural processing (Knill & Pouget, 2004; Pouget et al., 1998; Pouget, Deneve, Ducom, & Latham, 1999).

One popular theory is based on a neural architecture that combines basis functions and attractor dynamics. The basis functions are used to recode sensory information into a flexible intermediate representation to facilitate the transformation into a motor command; attractor dynamics are used for optimal statistical inferences. As an example, Deneve et al. (2001) proposed a network architecture that uses gain-encoding to combine inputs from two sensory modalities (vision and audition) and perform optimal Bayesian inferences.

This model used basis functions to perform coordinate transformations of the inputs (from retinal to head-centric coordinates) and then decode the maximum-likelihood estimate of the encoded variable (object location) given both inputs. This scheme relaxes noisy population codes into smooth hills of activity over time (stable network states called attractors). As a consequence, the multimodal inputs are integrated with weights proportional to their reliability. Our model presented here can be viewed as extending this approach to spiking networks and, more important, to dynamically varying inputs associated with a prior that must be learned.

Similarly, Salinas and Abbott (1996) demonstrate the use of recurrent architectures in computing the basis functions themselves to produce a useful intermediate representation of visual target location for motor action. Their approach uses recurrently connected populations to produce multiplicative responses as an emergent property of the network, even when individual neurons can only simply sum their synaptic inputs linearly. This scheme is illustrated by modeling the combination of target-retinal-location and gaze-direction inputs in parietal cortex. Such a multiplicative parietal representation has been shown to be very versatile because downstream neurons can readily read out various task-specific linear combinations of inputs. In both these approaches, recurrent connectivity effectively suppresses input noise. Salinas and Abbott (1996) also suggest how their model can select between multiple inputs.

Our proposed decoding scheme also bears an interesting relationship to other recent proposals, notably the probabilistic population

codes of Ma and colleagues (2006). Adopting the same decoder, they show that under some limited assumptions about the individual neurons' probability density functions, several natural computations can be carried out by linear operations on activities in the input populations. This means that in these situations, simple neural circuits, namely linear ones, can approximate optimal inference, which makes the optimization of the circuit trivial. The formulation readily applies to static cue-integration problems, in which cue weights are assumed to remain constant over time.

We consider a more realistic situation in which the input forms a continuous stream and the underlying variable of interest follows some trajectory. Hence, information from each cue can accrue over time in ways that affect how the different cues are combined through time. Our framework thus provides a computational characterization of the function of neural populations in a network, combining information from their input populations and performing computations in such a way that simple decoding closely approximates optimal inference.

Limitations of Our Approach

One limitation in the model is that the learning procedure is driven by supervision. The network assumes access to a global reward signal: knowledge of the optimal posterior based on the available information in the input spikes. The objective of learning in this setup was to assess whether any set of synaptic weights could be found that could perform meaningful computations dynamically, beyond the sole objective of input representation. If the aim extends to making the learning plausible in a biological context, then some feedback about the optimal distribution over stimulus values would be required. While information about the true underlying value can be plausibly derived, it is more challenging to envision how information about the optimal distribution could be obtained.

In our simulation of the Körding and Wolpert (2004) experiment, we are not focusing on how subjects actually learn, but rather how the neural populations can faithfully encode and transmit the information about the stimulus position via multiple cues. So, assuming that the population receptive and projective fields are defined, we only model how the feedforward and recurrent connections, and other network parameters adapt to ensure that the full distributional information is maintained online.

Predictions

Since we are always in the sparse spike limit in our simulations, the information per spike is of most relevance. If there were dense spiking, the population firing rate (DeCharms & Zador, 2000; Silberberg, Bethge, Markram, Pawelzik, & Tsodyks, 2004; Zhang & Sejnowski, 1999) might carry enough information to overwhelm any prior. The model makes two predictions about dynamically changing uncertainty: *(1)* During the first half of the trial, the variance of the time-varying posterior distribution over cursor positions rapidly increases with time. The variance decreases when visual feedback is received and then increases again much more slowly during the second half of movement. *(2)* The degree of uncertainty is consistent with the degree of noise in visual feedback.

Our model allows further predictions for extended versions of this task. For example, visual feedback could be provided more than once during the trial, and the model would predict the relationship between the estimated displacement and the particular feedback (position and uncertainty) on each trial. Our scheme is general enough to accommodate cases where the prior over cursor displacements is multimodal. Whereas standard dynamic models, such as Kalman filters, are unable to represent and update a multimodal posterior, our model can maintain and generate predictions in this case.

With regard to the population code, our study makes several predictions that may help initiate physiological investigations into representations of dynamic inputs. First, analysis of the visual and proprioceptive network dynamics suggests that the temporal dynamics of spiking reflect the temporal dynamics of the stimuli themselves. Second, lateral connections play a

critical role in encoding the temporal prior over stimulus dynamics, so that spikes during periods of limited input can be interpreted as making predictions about the underlying stimulus. In particular, during such periods, we expect to observe the way that the increased uncertainty (i.e., posterior variance) is coded. A future line of work involves examining whether the firing properties of the multisensory neurons in our network resemble those of parietal or motor cortical neurons. Since the firing patterns are by-products of the network dynamics, this investigation may be a useful step toward guiding and constraining our model formulations.

NOTE

1. Regarding notation: throughout this chapter we use a boldface variable to denote responses in a population of neurons; we use the arrow notation, e.g., $\vec{s}_T$, to denote a time-series vector, in this case of length T.

REFERENCES

Anderson, C. H. (1994). Basic elements of biological computational systems. *International Journal of Modern Physics C*, *5*, 135–137.

Baraduc, P., Guigon, E., & Burnod, Y. (2001). Recoding arm position to learn visuomotor transformations. *Cerebral Cortex*, *11*, 906–917.

Barbieri, R., Frank, L. M., Nguyen, D. P., Quirk, M. C., Solo, V., Wilson, M. A., & Brown, E. N. (2004). Dynamic analyses of information encoding in neural ensembles. *Neural Computation*, *16*, 277–307.

Barlow, H. B. (1953). Summation and inhibition in the frog's retina. *Journal of Physiology*, *137*, 69–88.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, *18*, 7411–7425.

Brunel, N., & Nadal, J-P. (1998). Mutual information, Fisher information, and population coding. *Neural Computation*, *10*, 1731–1757.

Chance, F. S., Abbott, L. F., & Reyes, A. D. (2002). Gain modulation from background synaptic input. *Neuron*, *35*, 773–782.

DeCharms, C., & Zador, A. (2000). Neural representation and the cortical code. *Annual Review of Neuroscience*, *23*, 613–647.

Deneve, S. (2005). Bayesian inference in spiking neurons. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. 17, pp. 353–360). Cambridge, MA: MIT Press.

Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: A neural implementation of ideal observers. *Nature Neuroscience*, *2*, 740–745.

Deneve, S., Latham, P. E., & Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, *4*, 826–831.

Ernst, M. O., & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, *415*, 429–433.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, *5*, 1688–1703.

Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1983). Neuronal population coding of movement direction. *Science*, *233*, 1416–1419.

Gold, J. I., & Shadlen, M. N. (2001). Neural computations that underlie decisions about sensory stimuli. *Trends in Cognitive Sciences*, *5*, 10–16.

Hillis, J. M., Ernst, M. O., Banks, M. S., & Landy, M. S. (2002). Combining sensory information: Mandatory fusion within, but not between, senses. *Science*, *298*, 1627–1630.

Hillis, J. M., Watt, S. J., Landy, M. S., & Banks, M. S. (2004). Slant from texture and disparity cues: Optional cue combination. *Vision Research*, *4*, 967–992.

Hinton, G. E. (1999). Products of experts. In *Ninth International Conference on Artificial Neural Networks, (ICANN 9)* (Vol. 1, pp. 1–6). New York, NY: IEEE.

Hinton, G. E., & Brown, A. D. (2000). Spiking Boltzmann machines. In S. A. Solla, T. K. Leen, & K-R. Müller (Eds.), *Advances in neural information processing systems (NIPS)* (Vol. 12, pp. 122–128). Cambridge, MA: MIT Press.

Huys, Q. J. M., Zemel, R., Natarajan, R., & Dayan, P. (2007). Fast population coding. *Neural Computation*, *19*, 404–441.

Jacobs, R. A. (1999). Optimal integration of texture and motion cues to depth. *Vision Research*, *39*, 3621–3629.

Jazayeri, M., & Movshon, J. A. (2006). Optimal representation of sensory information by neural populations. *Nature Neuroscience*, *9*, 690–696.

Kistler, W., Gerstner, W., & van Hemmen, J. L. (1997). Reduction of Hodgkin-Huxley equations to a threshold model. *Neural Computation*, *9*, 1069–1100.

Knill, D. C. (2003). Mixture models and the probabilistic structure of depth cues. *Vision Research*, *43*, 831–854.

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: The role of uncertainty in neural coding and computation. *Trends in Neurosciences*, *27*, 712–719.

Knill, D. C., & Saunders, J. A. (2003). Do humans optimally integrate stereo and texture information for judgments of surface slant? *Vision Research*, *43*, 2539–2558.

Körding, K., & Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, *427*, 244–247.

Landy, M. S., Maloney, L. T., Johnston, E. B., & Young, M. J. (1995). Measurement and modeling of depth cue combination: In defense of weak fusion. *Vision Research*, *35*, 389–412.

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, *9*, 1432–1438.

Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, *42*, 223–227.

Natarajan, R., Huys, Q. J., Dayan, P., & Zemel, R. S. (2008). Encoding and decoding spikes for dynamic stimuli. *Neural Computation*, *20*, 2325–2360.

Paradiso, M. A. (1988). A theory for the use of visual orientation information which exploits the columnar structure of striate cortex. *Biological Cybernetics*, *58*, 35–49.

Poirazi, P., Brannon, T., & Mel, B. W. (2003). Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell. *Neuron*, *37*, 977–987.

Pouget, A., Dayan, P., & Zemel, R. S. (2003). Inference and computation with population codes. *Annual Review of Neuroscience*, *26*, 318–410.

Pouget, A., Deneve, S., Ducom, J.-C., & Latham, P. E. (1999). Narrow vs wide tuning curves: What's best for a population code? *Neural Computation*, *11*, 85–90.

Pouget, A., Zhang, K., Deneve, S., & Latham, P. E. (1998). Statistically efficient estimation using population codes. *Neural Computation*, *10*, 373–401.

Rao, R. P. N. (2004). Bayesian computation in recurrent neural circuits. *Neural Computation*, *16*, 1–38.

Rao, R. P. N., Olshausen, B. A., & Lewicki, M. S. (Eds.). (2002). *Probabilistic models of the brain: Perception and neural function*. Cambridge, MA: MIT Press.

Salinas, E., & Abbott, L. F. (1995). Transfer of coded information from sensory to motor networks. *Journal of Neuroscience*, *15*, 6461–6474.

Salinas, E., & Abbott, L. F. (1996). A model of multiplicative neural responses in parietal cortex. *Proceedings of the National Academy of Sciences USA*, *93*, 11956–11961.

Saunders, J. A., & Knill, D. C. (2003). Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental Brain Research*, *152*, 341–352.

Saunders, J. A., & Knill, D. C. (2004). Visual feedback control of hand movements. *Journal of Neuroscience*, *24*, 3223–3234.

Seung, S. H., & Sompolinsky, H. (1993). Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences USA*, *90*, 10749–10753.

Silberberg, G., Bethge, M., Markram, H., Pawelzik, K., & Tsodyks, M. (2004). Dynamics of population rate codes in ensembles of neocortical neurons. *Journal of Neurophysiology*, *91*, 704–709.

Snippe, H. P., & Koenderink, J. J. (1992). Information in channel-coded systems: Correlated receivers. *Biological Cybernetics*, *67*, 183–190.

Stein, B. E., & Meredith, M. A. (Eds.). (1993). *The merging of the senses*. Cambridge, MA: MIT Press.

Tassinari, H., Hudson, T. E., & Landy, M. S. (2006). Combining priors and noisy visual cues in a rapid pointing task. *Journal of Neuroscience*, *26*, 10154–10163.

van Beers, R. J., Sittig, A. C., & Denier van der Gon, J. J. (1999). Integration of proprioceptive and visual position-information: An experimentally supported model. *Journal of Neurophysiology*, *81*, 1355–1364.

Van Rullen, R., & Thorpe, S. J. (2001). Rate coding versus temporal order coding: What the retinal

ganglion cells tell the visual cortex. *Neural Computation*, *13*, 1255–1283.

Zemel, R. S., & Dayan, P. (1997). Combining probabilistic population codes. *International Joint Conference on Artificial Intelligence*, *15*, 1114–1119. San Mateo, CA: Morgan Kaufmann.

Zemel, R. S., Dayan, P., & Pouget, A. (1998). Probabilistic interpretation of population codes. *Neural Computation*, *10*, 403–430.

Zhang, K., & Sejnowski, T. J. (1999). Neuronal tuning: To sharpen or to broaden. *Neural Computation*, *11*, 75–84.