Online Learning and Inference in Spiking Populations

Rama Natarajan
Dept. of Comp. Sci.
Univ. of Toronto

Quentin J. M. Huys
Gatsby CNU
UCL

Peter Dayan
Gatsby CNU
UCL

Richard S. Zemel
Dept. of Comp. Sci.
Univ. of Toronto

Abstract

We propose a reinforcement based framework for learning in recurrently connected populations of spiking neurons. Learning makes use of a reward signal, which conveys information about the quality of probabilistic inference based on the population spikes, and yet requires predominantly local information to specify synaptic plasticity. We apply this framework to the canonical example of probabilistic inference, namely the Bayesian combination of prior and likelihood about an input, but in the richest case of rapidly changing stimuli sparsely sampled by input spikes and re-represented in a plastic spiking population. We develop the ideal observer, which here involves inference in a Gaussian process, in a form that bears directly on the spiking network and compare their relative responses.

The timing of individual spikes is known to play a critical role in a wide variety of neural systems, including electric fish [6]; bats [8] and barn owls [3]. Spike timing appears particularly important when the relevant information in the animal's environment changese quickly (around $30 - 300$ms), on the same order of magnitude as typical interspike intervals [2]. These experimental results have prompted numerous proposals for how networks of neurons can collectively produce various characteristic forms of spiking behavior [5], as well as theoretical studies proving that neurons that convey information by individual spike times have computational advantages over neurons with sigmoidal activation functions (cite: Maass).

However, there are fewer proposals as to how the strengths of synaptic connections within and between spiking populations can be learned to carry out difficult information processing tasks, particularly those that change on the fast timescales that makes timing important. Learning is difficult in spiking populations because information is directly represented on only a temporally sparse, and spatially interacting 'trellis'; thus some of the most interesting examples of learning either operate with acausal inference (*eg* [9]; [7]) or punctate rather than population codes [4].

Here we consider how a particular form of spiking network can learn to perform one of the most fundamental aspects of Bayesian inference, namely combining likelihood and prior information associated with a dynamically evolving input stimulus. As in [10], we consider how a population of neurons can learn to implement a recurrent computation that remaps input spikes associated with the current value of the stimulus into output spikes that represent the stimulus' trajectory in a tractable, population-coded manner. This repre-

sentation implicitly fills in the sparse trellis, using temporally-extended decoding kernels. As a significant extension to that paper, we consider this in a fully spiking context, using an algorithm suggested in the field of reinforcement learning [1] to learn feedforward and recurrent synaptic weights to optimize the fidelity of coding. A so-called direct-policy reinforcement learning algorithm is appropriate here since (a) it is designed to operate with sampled processes such as spikes, and (b) the decision for a neuron in the population to spike at one time has an impact on what is represented at future times (via the temporally extended kernel associated with the assumed decoder for the population). Feedforward and recurrent weights thus learn to embody correctly the prior, the likelihood *and* the decoder.

In another key departure from our earlier work, we derive the recurrent population model using an approach similar to maximum entropy or additive random field models popular in other domains, such as language modeling, in which a large library of features exist, and learning involves deciding which features should be invoked. In this view, the feature associated with a neuron signals a particular relationship between the input and the underlying variable, and learning decides how to instantiate population spike trains so that the set of induced features convey appropriate information about the relevant variables over time.

In section 1, we describe the basic formulation of the maximum entropy model in the case of probabilistic inference about a changing stimulus. In section 2, we show how the Gaussian process ideal observer from [10] can be adapted to provide constraints on the output spikes in the recurrent population. Section 3 develops the online learning rule. Finally, in section 4 we apply the plastic population to various cases of inference in trajectories with weaker and stronger priors.

1 MODEL FORMULATION

Our new approach is formalized as follows. The inputs to the population over time $\mathbf{R}(T) = R(1), R(2), ..., R(T)$ arise in relation to the state X of underlying variable(s). The state is also a trajectory: $\mathbf{X}(T) = X(1), X(2), ..., X(T)$. Note that each R is a vector, with one entry for every neuron that provides input into the population. We generally assume that each entry is binary, representing whether the respective neuron spiked within the time interval t.

The recurrently-connected population receives afferent input from R, and conveys information about $X(t)$ to downstream neurons. Consider $\mathbf{S}(T) = S(1), S(2), ..., S(T)$ as the responses of the population within each time interval. We treat $S(t)$ as a binary representation of spikes within the population, just as $R(t)$ is the binary vector for incoming spikes.

As in a standard maximum entropy, or recognition modeling view, the inputs up to and including time T are viewed as data that the network can use to represent $X(T)$, rather than information that must be explained. In a departure from conventional maximum entropy treatments, we formulate the objective in a recursive online manner, so that the network makes use of its own spikes $\mathbf{S}(T-1)$ up until time T in addition to its inputs $\mathbf{R}(T)$:

$$P(X(T)|\mathbf{S}(T-1); \mathbf{R}(T)) = \sum_{S(T)} P(X(T), S(T)|\mathbf{R}(T), \mathbf{S}(T-1)) \quad (1)$$

$$= \sum_{S(T)} P(X(T)|\mathbf{S}(T)) P(S(T)|\mathbf{R}(T), \mathbf{S}(T-1)) \quad (2)$$

where the spike vector $S(T)$ is a stochastic random vector. The constraint that the population spikes convey all the information about $X(T)$ present in its input motivates the conditional independence of $X(T)$ and $\mathbf{R}(T)$ given $\mathbf{S}(T)$ assumed in Equation 2.

The second term on the right-hand side of this equation represents the stochastic spiking

model, which stipulates how the population spikes arise, while the first term represents information contained in these spikes about $X(T)$. For both terms we describe a common, recursive, and online mechanism for inference. The first term involves a form of spike response model (cite: Gerstner). The probability that a unit j in the population will spike depends on its internal state variable, or "membrane potential" $b_j(T)$, which in turn depends on the spike times of its pre-synaptic units:

$$b_j(T) = \sum_i \sum_{\tau=1}^{T} R_i(\tau) w_{ij} \psi(t-\tau) + \sum_k \sum_{\tau=2}^{T} S_k(\tau) u_{kj} \psi(t-\tau) \tag{3}$$

Here, $\psi()$ is a spike response function (SRF), which models the unweighted post-synaptic potential (PSP) of a single spike impinging on the neuron. This is typically a decaying function of the time interval, so that more distant spikes have diminished effects. Weights $\mathbf{W} = \{w_{ij}\}$ and $\mathbf{U} = \{u_{kj}\}$ describe the synaptic efficacies that modulate the heights of the PSPs. Rather than explicitly model a balanced excitatory-inhibitory structure (Destexhe, van Vreeswijk), we treat the spikes generated by the population as stochastic samples based on these membrane potentials:

$$P(S(T)|\mathbf{S}(T-1);\mathbf{R}(T)) = \prod_j P(S_j(T)|\mathbf{S}(T-1);\mathbf{R}(T)) \tag{4}$$

$$P(S_j(T)=1|\mathbf{S}(T-1);\mathbf{R}(T)) = \sigma(b_j(T) = (1+\exp(-b_j(T)))^{-1} \tag{5}$$

The population spikes are assumed *conditionally* independent: these spikes will be highly *marginally* correlated, but the variability will be independent given earlier population spikes and past and current input spikes.

Note that if the SRF is a decaying exponential, $\psi(a) = \exp(-\alpha a)$, where α controls the integration time constant of the neuron, then the potential has a simple recursive form

$$b_j(T) = \sum_i R_i(T) w_{ij} + \sum_k S_k(T-1) u_{kj} + \exp(-\alpha) b_j(T-1) \tag{6}$$

which leads to a form of a stochastic leaky integrate and fire neuron, with $\exp(-\alpha)$ controlling the leak. The temporal integration has an important effect in this model, allowing it to go beyond a simple first-order Markov model, in that the current distribution over population activities may depend on activities several time-steps into the past.

The first term in Equation 2 represents the decoding aspect of our model. In this paper we adopt an interpretation of spikes that relies on a basis set for representing $X(T)$:

$$P(X(T)|\mathbf{S}(T)) = \frac{1}{Z} \exp\left[\sum_j \sum_{\tau=0}^{T} S_j(\tau) \phi_j(X) \psi(t-\tau)\right] \tag{7}$$

where associated with each population neuron j, is a spatial kernel $\phi_j(X)$.

Note that, as above, if the SRF $\psi(a) = \exp(-\alpha a)$ involves exponential decay, then we can express $d(T) = \log P(X(T)|\mathbf{S}(T))$ recursively:

$$d(T) = \sum_j S_j(T) \phi_j(X) + \exp(-\alpha) d(T-1)$$

Decoding the information contained in the population spikes $\mathbf{S}(T)$ may thus be readily carried out by downstream neurons. One potential difficulty if the exact posterior is required is the normalization term, or partition function Z. However, in general this computation will not be necessary. A more likely operation of comparing the relative log-probabilities assigned to two values of X can be computed as follows:

$$\log \frac{P(x_1(T)|\mathbf{S}(T))}{P(x_2(T)|\mathbf{S}(T))} = \sum_j S_j(T)[\phi_j(x_2) - \phi_j(x_1)] + \exp(-\alpha)[d_1(T-1) - d_2(T-1)] \tag{8}$$

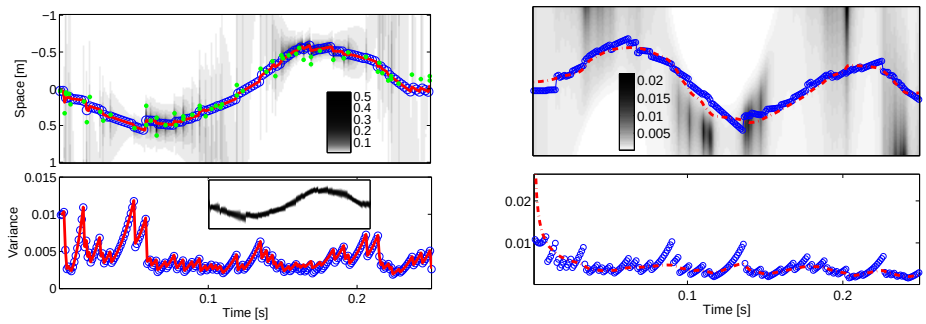

Figure 1: Comparison of GP and MaxEnt distributions. **Top row**: an inferred $\mathbf{S}$ is shown in grayscale in the background. Mean of the exact inference with the GP is shown in blue. Mean of the MaxEnt distribution represented by $\mathbf{S}$ is shown in red (dash-dotted). **Bottom row**: Variance of the GP distribution (blue) and the MaxEnt distribution (red). The columns show results for different temporal convolution kernels $\psi(t) = \exp(-t/\tau)$: **left** $\tau = 2$ ms **right** $\tau = 30$ ms. For illustration, the **inset** in the bottom right panel shows the GP distribution and the green dots in the left top panel show the input spikes $\mathbf{R}$.

2 IDEAL OBSERVATION

In [10], we explored a continuous Gaussian Process (GP) model as a form of ideal observer to understand the statistical constraints inherent in the problem. Briefly, under this model, (continuous-time) trajectories $X(t)$ are generated according to a Gaussian process prior, with a covariance function $\mathcal{C}(t - t') \propto \exp(-\|t - t'\|_x)$ specifying the degree to which $X(t')$ and $X(t)$ are related, in terms of the covariance of their joint Gaussian distribution. Different covariance functions generate different characteristic trajectories (e.g. smooth or rough), while the mean trajectory $\mathbf{n}$ can capture more global tendencies such as a drift.

If the neurons generating the input spikes are independent inhomogeneous Poisson processes with mean rates given by Gaussian tuning functions in $X(t)$, with dense, uniform coverage of the whole interval in which $X(t)$ lives (tuning centers $\mathbf{m} = (m_1, \ldots, m_n)$), then the true, Bayesian, posterior distribution of $X(T)$ given the whole collection of input spikes $\mathbf{R}(T)$ is Gaussian with mean $\hat{\mu}(T)$ and variance $\hat{\nu}(T)^2$, where

$$\hat{\mu}(T) = \mathbf{k}^T(\boldsymbol{\theta} - \mathbf{n}) \qquad \hat{\nu}(T)^2 = \mathcal{C}_{TT} - \mathbf{k}^T\mathcal{C}_{lT} \quad \text{where} \quad \mathbf{k} = \mathcal{C}_{Tl}(\mathcal{C}_{ll} + \mathbf{I}\sigma^2)^{-1} \tag{9}$$

is a kernel function that depends on the times of all the spikes in the population, with θ_{t_i} being the preferred tuning value of the neuron spiking at time t_i and $\mathcal{C}_{t_k t_l}$ being the covariance function $\mathcal{C}(t_k - t_l)$.

Here we make use of the optimality of the GP inference to generate constraints on the maximum entropy decoding. Let us try to find a set of spikes $\mathbf{S}(T)$ that when decoded according to the log-linear model, matches the distribution inferred by the GP as well as possible. Let us further require that these spikes be functions of the input spikes $\mathbf{R}(T)$ and the previous $\mathbf{S}$ only. We thus rewrite the GP distribution in terms of $\mathbf{S}(T)$:

$$p(X(T)|\mathbf{R}(T)) \propto \exp\left(-(X(T) - \hat{\mu}(T))^2/2\hat{\nu}(T)^2\right) \tag{10}$$

$$\propto \exp\left(-\sum_j \left[\int_t S_j(T - t)\psi(t)\right](X(T) - m_j)^2/2\sigma^2\right) \tag{11}$$

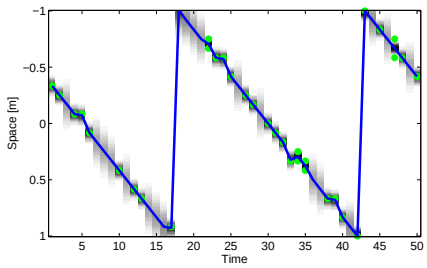

Figure 2: GP performance on canonical walk. The drift (canonical walk) was included in the GP as a mean $\mathbf{n}$. The shaded gray indicates the posterior distribution. The green dots are the spikes and the blue line is the posterior mean.

since a sum of square terms is another square itself, all provided these constraints are met:

$$\hat{\mu}(T) = \frac{\sum_j \zeta_j(T) m_j}{\sum_k \zeta_k(T)} \qquad \hat{\nu}(T)^2 = \frac{\sigma^2}{\sum_j \zeta_j(T)} \qquad \zeta_j(T) = \int_t S_j(T-t)\psi(t) \qquad (12)$$

In the the same way that the kernel functions $\mathbf{k}$ specify effective, prior-dependent, *encoding* firing rates for the inputs in the population, the kernel functions $\int_t S_i(T-t)\psi(t)$ specify effective *decoding* firing rates for the recurrent population. Equation 11 is just the same as Equation 7.

By solving Equations 12, we can examine the nature of the match between the temporal structures of encoding and decoding, and also understand the constraint under which the recoding in the network must operate. As there are more variables $S_i(t)$ than equations once the problems has been temporally discretized, one should hope that there will be at least one highly faithful population representation of the posterior. Ideally, we would furthermore find a solution to Equations 12 in terms of binary $S_i(t)$. However, finding such discrete solutions is a notoriously hard computational problem, so we typically search in a continous (but positive) space, via a map $S_i(t) = \sigma(b_i(t))$. It is no accident that this can be seen as the mean field version of the network studied above. Indeed, in future work, we would like to search for *recursively-specifiable* solutions, for which, at least approximately, an optimizing $b_j(t)$ will be found that satisfies an equation exactly like Equation 3.

The two examples shown in Figure 1 show that it is indeed possible, despite local optima, to find a set of $S(t)$ that faithfully represent the posterior. However, the degree to which this is possible depends on the match between the kernels $\psi(t)$ and $\phi(X)$ and the GP. For example, in the right column we show that the representation deteriorates if $\psi(t)$ is much slower than the typical timescale of $X(t)$.

3 ONLINE LEARNING AND INFERENCE

This Gaussian Process formulation acts as an ideal observer for the overall aim of conveying the proper posterior distribution over $X(T)$ contained in $\mathbf{R}(T)$. It thereby formalizes the inference task that our recurrently conected population of spiking neurons interpreted as in section 1 has to learn to solve. In the network, inference involves producing and interpreting population spikes; based on the model defined above, inference is direct.

We formulate the learning objective of the system based on a measure of the quality of the posterior distribution over $X(T)$ conveyed by the population spikes. If the true value of $X(T)$ is known, then this objective is simply to maximize $L = \sum_t \log P(X^*(t)|\mathbf{R}(T), \mathbf{S}(T-1))$. In [10], we considered an objective of this sort, but

in a mean-field formulation of the network in which we had access to something closer to the membrane potentials in the population ($b_j(t)$) rather than just the spikes. Here, we turn to an alternative form of learning, namely reinforcement learning, in which spikes are seen as the actions of independent agents (*viz* the neurons), and the value of these actions are evaluated collectively in terms of the quality of the present and future population representation which they produce. The population spiking probabilities $P(S(t)|\mathbf{R}(T), \mathbf{S}(T-1))$ (see Equation 5) can be thought of as defining a stochastic policy that maps the observations $\mathbf{R}(T), \mathbf{S}(T-1)$ into the probability distributions over the output units. The policy is directly parameterized by the weights $\mathbf{W}$ and $\mathbf{U}$, as well as α, all of which we summarize as Ω.

The *instantaneous* reward signal at time T, assuming access to the true stimulus state $X^*(T)$ is:

$$v(\Omega, t) = \log P(X^*(T)|\mathbf{R}(T), \mathbf{S}(T-1)) \tag{13}$$

Given an objective function based on *long run* average reward $J(T) = 1/T \sum_{t=1}^{T} v(\Omega, t)$, the gradient with respect to Ω of the expected reward is not computable in closed form since the output spikes are stochastically sampled. We resort to an approximate gradient approach (called a direct policy method) in which the gradient is estimated via simulation and the policy is improved by adjusting the parameters in the gradient direction [1]. The gradient with respect to Ω is:

$$\frac{\partial J(T)}{\partial \Omega} = \frac{(T-1)}{T}\frac{\partial J(T-1)}{\partial \Omega} + \frac{1}{T} v(T) z(T) \tag{14}$$

where the eligibility trace $z(T)$ that relates the output spikes $\mathbf{S}(T)$ with the weights $\mathbf{W}$ and $\mathbf{U}$ is defined as

$$z(T) = \beta z(T-1) + \frac{1}{p(T)}\frac{\partial p(T)}{\partial \Omega}, \quad \frac{1}{p(T)}\frac{\partial p(T)}{\partial \Omega} = \sum_j \frac{\partial b_j(T)}{\partial \Omega}(S_j(T) - \sigma(b_j(T)))$$

The gradient for each of the parameters in Ω takes the form:

$$v(T) R_i(T) [S_j(T) - \sigma(b_j(T))] \tag{15}$$

Learning therefore depends on correlations between the reward signal and pre-synaptic spikes. The magnitude of the gradient is also modulated by $S_j(T) - \sigma(b_j(T))$, which is maximized when the actions (whether or not to spike) are unpredictable. Not unsurprisingly, learning stops when the distribution of responses of the features under the model match their empirical distribution, as in standard random field and maxent models.

4 EXPERIMENTS

We present results from three experiments designed to explore the capacity of our model in a simple, controllable paradigm involving inference about a 1-D variable X in a case in which input spikes are noisy and sparse. We applied the learning model exactly as above, except that the recurrent self-connections u_{jj} are forced to be negative, to implement a crude form of refractory period where a neuron that spiked at one time step cannot spike at the next. The state space of X is defined by Z discrete states, for ease of simulation. For the temporal kernel $\psi(t) = \exp(-\alpha)$, the value of α is derived empirically from the data, defined based on the mean persistence of the value of X in the input sequences.

Our first experiment involves a simple canonical walk that follows a fixed, repetitive trajectory through time. The learned temporal decay is rapid, so that the decoded position $X(T)$ depends only on spikes in the current time bin. The canonical walk is generated according to: $X(t+1) = [t * \Delta + \epsilon(t)] \mod 2 - 1$, where $\Delta = 2/N$ and $\epsilon(t)$ is mean-zero Gaussian noise. Figure 3 shows that the recurrent weights have learned the repetitive pattern in the

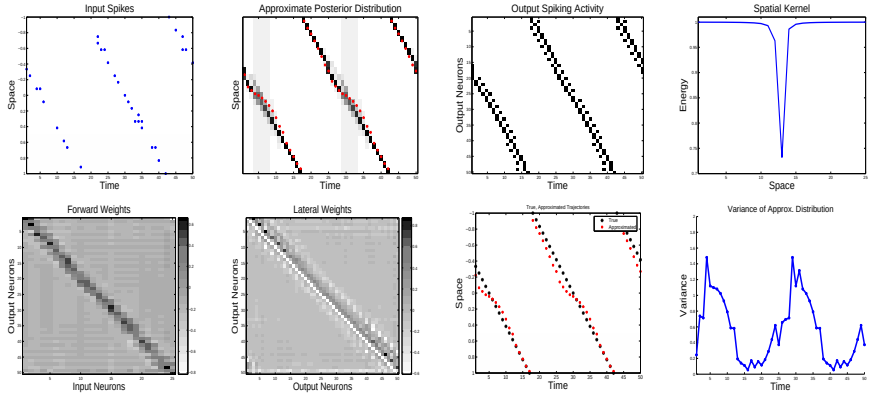

Figure 3: Experiment 1: Canonical Walk: N=25; M=50. Z=25.Simple canonical walk that follows a fixed, repetitive trajectory through time

trajectory. The network compensates for the lack of temporal integration at the output and the sparse input spikes that give rise to uncertainty, and faithfully reconstructs the position over time. In particular, it infers a good model of the directed component of the random walk.

The consequence of forced negative self-connections is directly reflected in the output spiking activity. Figure 3 shows that no neuron fires repetitively for any two timesteps. As such, neighboring neurons (defined as neurons with similar selectivity for position) cooperate to produce a good approximation. The network therefore cleans up the input and fills in for missing input spikes.

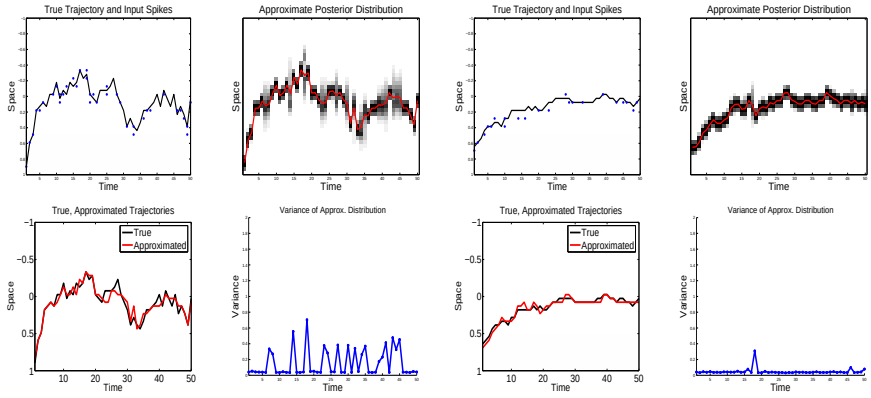

Figure 4: Experiment 2: Random Walk: N=25; M=30. Z=25. Top left: Rough walk and decoded distribution. Top right: Smooth walk and decoded distribution. Below: Mean and variances for the corresponding rough and smooth walks.

The second experiment explored the effect of temporal integration in encoding. Two random walks in 1-D space were generated with varying degrees of smoothness, one rough (s.dev.=0.4), the other smooth (sd=0.1) (see Figure 4). The learned decay constant was correspondingly fast for the rough walk; slow for the smooth one. Under a sparse input spike regime, the system produced spikes that lead to a posterior that matches the true walk

well. This is due both to the learned recurrent weights and the integration time constant α.

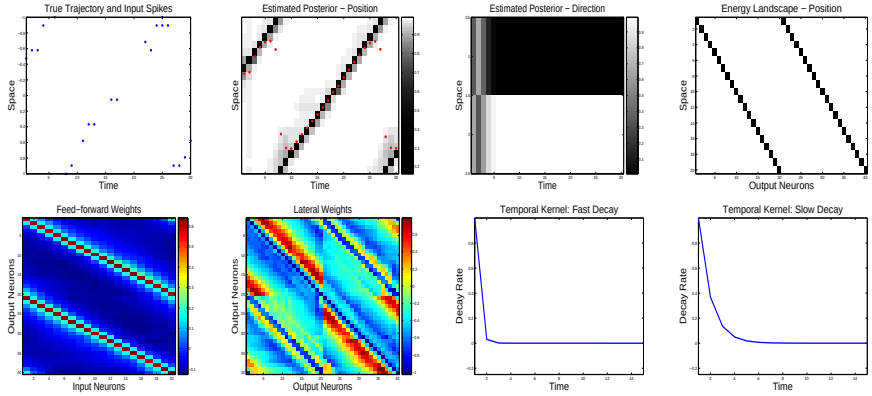

Figure 5: Experiment 3: Inferring velocity of the stimulus from the station position information. Top: Motion in Left direction. Below: Synaptic weights and temporal kernels with fast and slow decay for position and direction respectively.

In the third experiment, we show how a dynamic function of the stimulus (direction) can be inferred from encoding of static information (position). We focus on the simple case of rightward (R) or leftward (L) motion of a stimulus in 1-D space, with a constant speed. Each neuron in the output population encodes both position and direction of motion. In our approach, this implies particular 2-dimensional kernels ϕ_{ij} for each output neuron, where i indexes position and j, the direction. As a simple setup to allow easy interpretation of the synaptic weights learned by network, we have 2 subpopulations, each coding for motion in one specific direction. Figure 5 shows that the feedforward weights to 2 neurons coding the same position are the same, and only the lateral weights distinguish the direction for which they are coding.

5 DISCUSSION

In this paper we have shown approached the relationship between encoding and decoding in verious ways. In particular, we have shown a number of issues that arise in neural recoding of information and suggested neurally plausible solutions. In particular, we have drawn attention the structure of the code and pointed to a way in which a simple yet very powerful code could be maintained for probabilistic computations throughout the brain.

References

[1] Baxter and Bartlett, 2000. Proc IEEE Int Symp Circuits Syst, 3:271:274

[2] Bialek et al., 1991. Nature, 252:1854

[3] Carr and Konishi, 1988. PNAS. 85:8311-5.

[4] Deneve 2004. Nips 17.

[5] Gerstner and Kistler: Spiking Neuron Models. 2002. CUP

[6] Heiligenberg. Curr Opin Neurobiol. 1991 1:633-7

[7] Hinton and Brown 2000. NIPS 12:122

[8] Olsen and Suga. 1991. J Neurophysiol., 65:1275-96.

[9] Smith and Lewicki, 2005. Neural Comp., 17:19-45

[10] Zemel, Huys, Natarajan and Dayan, 2004. NIPS 17.