

Hebbian learning in linear-nonlinear networks with tuning curves leads to near-optimal, multi-alternative decision making

Tyler McMillen¹, Pat Simen² and Sam Behseta¹

¹ Department of Mathematics, California State University at Fullerton, Fullerton, CA 92834

²Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544

E-mail: tmcmillen@fullerton.edu, psimen@math.princeton.edu, sbehseta@fullerton.edu

September 25, 2009

Abstract

Optimal performance and physically plausible mechanisms for achieving it have been completely characterized for a general class of two-alternative decision making tasks, and data suggest that humans can implement the optimal procedure. A greater number of alternatives complicates the analysis, but here too, analytical approximations to optimality that are physically and psychologically plausible have been analyzed. These analyses leave open questions that have begun to be addressed: 1) How are near-optimal model parameterizations learned from experience?; 2) What if a continuum of decision alternatives exists?; 3) How can neurons' broad tuning curves be incorporated into an optimal-performance theory that assumes idealized, zero-width tuning curves? We present a possible answer to all of these questions in the form of an extremely simple, reward-modulated Hebbian learning rule for weight updates in a neural network that learns to approximate the multiple sequential probability ratio test.

KEYWORDS: synaptic weight learning, leaky accumulator, drift-diffusion model, neural network, multi-hypothesis sequential test, sequential ratio test.

1 Introduction

Tuning curves are ubiquitous in neural responses to stimuli (Butts and Goldman, 2006). The relationship between tuning curve shape and decision making performance has intrigued researchers for several years (see, e.g. Pouget et al. (1999)). Naively, one may suppose that task participants may improve their performance by sharpening the tuning curves of the neurons involved. However, wider tuning curves are in some cases more efficient in conveying information, and the most informative tuning curve shape depends strongly on the covariance of the noise (Zhang and Sejnowski, 1999; Seriés et al., 2004). Moreover, in several tasks a subject may improve his performance without significantly altering the shapes of the tuning curves in the neurons involved. For instance, in an angle discrimination task, monkeys are able to learn to discriminate between finer angles over time, while the tuning curves are altered very little (Ghose et al., 2002; Law and Gold, 2008). This suggests that improvements in performance take place in a learning process downstream from the receptor neurons.

In this paper we explore the ways in which a subject may improve performance in decision tasks, given tuning curve shapes in receptor neurons. We do not consider the alteration of receptor units' tuning curves, but rather how the information in tuning curves can be utilized more efficiently over the course of many trials. In particular, we are interested in how well a subject can do in

timed tasks where there is a speed-accuracy tradeoff. Generally, in such tasks, given more time one may improve one’s accuracy, but at the expense of making fewer decisions per allotted time. The optimal strategy is the one that results in the most correct decisions per unit time, i.e. the one that maximizes the *reward rate*. For a task in which the time to respond is fixed by the experimenter, this amounts to a strategy that chooses the most likely hypothesis. For tasks in which the subject is free to decide at any time, the subject must set his or her thresholds for decisions. These two types of experimental protocols are referred to as the *interrogation* and *free response protocols*, respectively.

Here we are mainly interested in the free response protocol. In the case where the subject is free to respond at any time, and the next trial begins a time D after a response, called the response-stimulus interval (RSI), the reward rate (RR) is defined in terms of the error rate (ER) and mean reaction time (MRT):

$$\text{RR} = \frac{1 - \text{ER}}{D + \text{MRT}}. \quad (1)$$

The optimal threshold for a response depends on the RSI D . This may be seen by noting that if D is zero, then the best strategy is to decide as quickly as possible, since even if the ER is what would be achieved by random guessing, since the MRT is small, the RR will be large. Conversely, if D is very large, one should take one’s time to ensure a small probability of error, since there will be few chances for trials. Between these two extremes of low threshold (small MRT) and high threshold (large MRT) is a range of optimal thresholds. Thus, given any decision mechanism, there will be an optimal threshold for each D . The best mechanism, in terms of optimizing RR over the whole range of possible D ’s, is the one that minimizes the MRT for a given ER. In this paper we are mostly concerned with the latter, that is, elaborating on what is the optimal test and how it can be achieved in a neural network. This question is related to the former question of where one sets one’s threshold, but we leave this for future work. In the Discussion section we will discuss the ways this can be examined.

Our canonical example is the motion direction task. In this task, a subject observes a collection of moving dots on a screen. A certain proportion of them are moving in one of N directions, while the rest are moving randomly. The observer must then determine the direction in which the coordinated dots are moving. In a typical experiment the animal indicates the direction of movement by moving its eyes in the perceived direction. (See Churchland et al. (2008); Niwa and Ditterich (2008); Law and Gold (2008) for recent results in such tasks.) The difficulty of this task depends essentially on three factors: (1) the proportion of dots that are moving coherently (signal-to-noise ratio), (2) the number of possible directions of coordinated movement, and (3) the distance between these alternatives.

In the following section we describe a network for decision making for the above task. It consists of three layers: a signal layer, a layer of leaky, competing accumulators (LCA’s), and a decision layer. The main result of this paper is a learning algorithm for the weights between the layer of accumulators and the decision layer. McMillen and Behseta (2009) showed that overlapping of signals corresponding to different alternatives can be an advantage if the resulting output of the accumulators is multiplied by a matrix that encodes the possible alternatives. This matrix multiplication amounts to tuning the weights from the accumulator layer to the decision layer in order that they mimic the shapes of the possible alternatives. In §3 we describe a simple Hebbian learning rule that performs remarkably well at learning the optimal weights as described in McMillen and Behseta (2009). In this section we explain why the algorithm works. In §4 we present numerical simulations of the blocks of trials involving four alternatives. We see how performance improves over the course of trials, and how the optimal weights are learned for a variety of different shapes. Finally, we conclude in §5 with a few remarks and a discussion of how the algorithm we propose can be adapted to situations in which the delay between trials is changing.

The network we propose has several advantages. For one, it is extremely simple. A summary of the complete model is given in Table 1 on page 8. This simplicity makes the model tractable for analysis, and it also suggests an extremely simple physical substrate that could plausibly be implemented in the brain (a substrate composed of the same electric circuit components used to model the individual neural membrane). The model also incorporates decision making criteria: a decision is made once one of the units in the decision layer crosses a threshold. Thus we examine how it performs in the free response protocol. It is also trivial to adapt the model to the interrogation protocol, since one need simply stop the test after some time and select the alternative corresponding to the unit with the largest value in the decision layer. The network is also easily adaptable. If the signal vectors change, the network can quickly learn the optimal weights to encode these new alternatives.

Furman and Wang (2008) investigated one generalization of such networks in the context of a more detailed but less analytically tractable model, and suggested that a continuous decision was difficult for reduced models of the type investigated here. However, we have shown that the solution to the continuum problem is straightforward: simply adding more units as done by Furman and Wang (2008) is sufficient for achieving decision making in the continuum context. What is more however, when decision making requires a discrete set of responses, nearly optimal performance can be analytically assessed for the LCA model through a direct mapping onto the multiple sequential probability ratio test (MSPRT), the only known, computationally tractable N-alternative hypothesis testing procedure with fixed thresholds that approximates optimal performance (that is, it approximately minimizes RT for a given accuracy, with approximation error decreasing to 0 as accuracy approaches 100%). The model of Bogacz and Gurney (2007) also leverages the same analytical tractability to implement the MSPRT in the case of Kronecker delta tuning curves.

Beck et al. (2008) investigated the same problem from the perspective of optimal integration of information using a scheme that requires spike statistics and the way that neurons integrate spikes to meet certain restrictions. Given their primary focus on information integration, Beck et al. (2008) do not address the issue of when to make the decision, and therefore cannot address how reward rates can be maximized by their approach. In fact, the approximately optimal MSPRT itself was introduced because the optimal, Bayesian decision procedure requires dynamically changing decision boundaries within trials, with boundary trajectories that are unique to each particular problem. The advantage of the model proposed here is that with nothing more than a classic, nearly linear firing-rate model that can be implemented with an economy of physical components, we can implement an approximately optimal decision making procedure and thereby give a complete, decision-theoretic account of decision making by these networks.

Furthermore, the dynamics of decision making in our model are essentially the same as in the aforementioned models. On each trial, an initial, broadly spread Gaussian signal is sharpened over the course of a trial by a process of competitive integration. The peak of the sharpened signal can then be compared to a threshold for response initiation, and threshold adaptation can then be used to adjust speed-accuracy tradeoffs by the model. Such a process requires no violation of the assumptions made in linear systems theory, and is therefore highly analytically tractable.

2 The leaky competing accumulator model for decision making

We propose a three layer neural model. The first layer acts simply as a sensory amplifier; the next layer integrates the information from the first layer, but also exhibits competitive dynamics that gradually build a commitment to one course of action over the alternatives; the last layer triggers a discrete motor response when commitment to one response is sufficiently strong. For convenience, we refer to these three layers, respectively, as the MT, LIP and SC layers. These labels reflect the fact that our model exhibits known properties of neurons in the monkey middle temporal area

(MT), the lateral intraparietal (LIP) and the superior colliculus (SC) in decision making tasks requiring eye movement responses (Beck et al., 2008). The architecture of this circuitry is expected to apply without major modification to decision making tasks involving other stimulus and response modalities, however. Nevertheless, tasks requiring eye movements to target locations in response to visual motion signals are the most commonly studied form of this task in monkeys (and increasingly humans).

We suppose that MT neurons have tuning curves that are preferentially sensitive to a single, given direction of visual motion, and that another layer is stimulated by the activity in this input layer. By virtue of their excitatory connections to LIP, model MT units' tuning curves and their feedforward connections to LIP in turn define tuning curves for LIP units. Questions of major importance in computational neuroscience are: Through what sort of learning process do these tuning curves arise? Can we define an optimal connection scheme that maximizes some function, such as the rate of reward earned by the model? And is it really necessary that we abandon classic neural network assumptions when modeling multi-alternative decision making, as recent models have done? These classic assumptions are that the brain circuits in question are approximately linear systems (at least over a limited range of inputs), and that they employ simple learning schemes (such as Hebbian learning, or error-updating rules such as the Widrow-Hoff, Rescorla-Wagner or delta rule). Recent work (e.g. McMillen and Holmes (2006); Bogacz and Gurney (2007)) that avoids discussion of tuning curves shows that these assumptions allow simple neural network models to map precisely onto abstract, optimal hypothesis testing procedures. We now demonstrate that a model consistent with these assumptions does remarkably well at approaching optimal (reward maximizing) performance in decision making tasks with multiple alternatives. The scheme we propose is similar to the models of Beck et al. (2008); Furman and Wang (2008), although we give a different interpretation of the evolution of accumulator activity in LIP that are consistent with the assumptions just described. The model's layers are represented mathematically by S , x , and z . Figure 1 shows a diagram of the model.

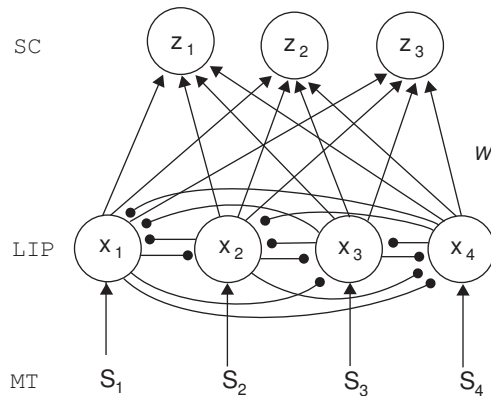


Figure 1: Neural network model with 4 accumulators and 3 alternatives. The weight matrix W denotes the weights of the connections between the x_i 's and z_j 's. Arrows represent excitatory connections and circles represent inhibitory connections.

The three layer model is as follows. Upon presentation of a stimulus, MT neurons present a vector of signals to accumulators in the LIP layer. The signals presented to the LIP layer are referred to as S_i , and represent the total weighted sum of signals to the i th accumulator. A given stimulus will result in a given signal, so that the the signal vector to the LIP layer may be represented as a vector indexed by μ :

$$S^\mu = (S_1^\mu, S_2^\mu, \dots, S_n^\mu),$$

the task is to determine which of N possible signal vectors this represents. Notice that the number of vectors can be different from the number of signals, i.e. in general $n > N$.

We will generally take the S^μ signals to be Gaussian, as we interpret these in terms of MT tuning curves, as in McMillen and Behseta (2009),

$$S_i^\mu = a \exp \left[-\frac{(i - dir_\mu)^2}{2\phi^2} \right], \quad i = 1, \dots, n. \quad (2)$$

Here dir_μ is the peak of the signal, a is the height of the peak, ϕ is the width of the curve, and S_0 is the size of the background signal. Notice that if $\phi = 0$, then

$$S_i^\mu = a \delta_{i, dir_\mu},$$

where $\delta_{i,j}$ is the Kronecker delta, so that the signal is concentrated in the channel dir_μ . But, if $\phi > 0$, the signal will have a spread around the peak. For tuning curves associated with the dots motion task, tuning curves have been measured to have a width of about 40° (Law and Gold, 2008). The situation is illustrated in Fig. 2. Angles far apart have very little overlap in the signals, but when the angles are close the overlap is significant. For the two alternative case in which dots travel either left or right, the signals have very little overlap. Signals for alternatives corresponding to closer directions have more overlap.

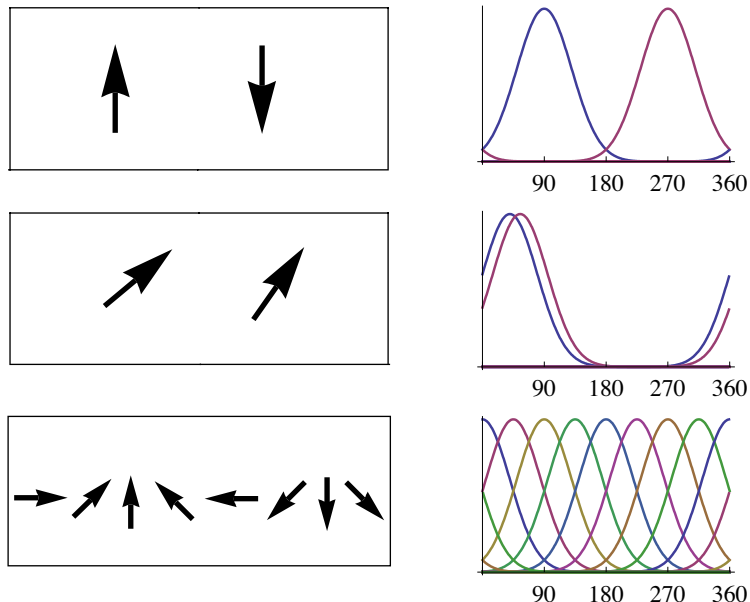


Figure 2: Possible directions of coordinated movement (left panels) and corresponding signal vectors (right panel).

We model the LIP layer as a set of n leaky competing accumulators. The linearized model for their evolution is a stochastic differential equation (Usher and McClelland, 2001; Bogacz et al., 2006; McMillen and Holmes, 2006):

$$dx_i = \left(-kx_i - m \sum_{j \neq i} x_j + S_i \right) dt + c dW_i, \quad i = 1, \dots, n, \quad (3)$$

where k is the decay rate, m is the mutual inhibition, and W_i is a Wiener process (white noise) representing the noise in the signal and from other sources. The signal-to-noise ratio is the ratio

of the magnitude of the largest signal to the variance of the noise, i.e. a/c . We can thus model changes in the direction coherence by changing this ratio. The effect of decay and inhibition is to concentrate the values of the accumulators onto the signal vectors. Thus, moderate values of w and k tend to increase the accuracy. Best results are achieved when decay and inhibition are balanced, i.e. $w = k$ (McMillen and Holmes, 2006). For simplicity, and to be concrete, throughout the rest of this paper we will present results for $k = w = 0.5$, $a = 2$ and $c = 1$. Results are qualitatively insensitive to these choices.

The output from accumulator j is fed into the i th unit of SC with weight w_{ij} . Thus, the SC units are given by

$$z_i = f(y_i), \quad y_i = \sum_{j=1}^n w_{ij} x_j. \quad (4)$$

That is, y_i is a weighted sum of the accumulators, and the values z_i of the SC units are obtained by passing these weighted sums through a function $f(x)$. We will usually take $f(x)$ to be a sigmoidal,

$$f(x) = \frac{1}{1 + e^{-\beta(x-b)}}.$$

A decision is made once one of the z_i 's saturates, or crosses a threshold. The threshold may be easily modified by altering b in the above sigmoidal. Since the sigmoidal is one-to-one, this equivalent to making a decision once one of the weighted sums y_i crosses a threshold. Thus, throughout we will suppose that a decision is made the first time $y_i \geq \theta$, where θ is a threshold.

The results in this paper are generally applicable, but to be precise we consider a motion direction task with 36 accumulators and interpret these as representing increments of 10° . If the direction $j \cdot 10^\circ$ is presented, the signal vector takes the shape S_i^μ as in (2), with $dir_\mu = j$. For concreteness we consider four possible directions of motion: $30^\circ, 60^\circ, 140^\circ, 220^\circ$. Thus, if say, the direction of coordinated movement is 60° , the signal vector has a peak at the sixth accumulator. The four possibilities are represented by the four possible signal vectors with peaks at accumulators 3, 6, 14 and 22. In this paper we only consider the case when all the possibilities are equally likely, in which case the appropriate initial condition for the accumulators is $x_i(0) = 0$.

McMillen and Behseta (2009) showed that the optimal weights w_{ij} in the above are achieved when the weights mimic the shape of the possible incoming signal vectors. That is to say, a threshold crossing test best approximates the optimal test when $w_{ij} = S_j^{\mu_i}$. The magnitude of the weights are not important in terms of optimality, as the magnitude may be incorporated into the thresholds. The performance of the threshold crossing tests is illustrated in Fig. 3. Here we consider a test with 36 accumulators and the four alternatives as described above. In Fig. 3 we plot the MRT for a fixed value of the ER (ER = 0.1). For each value of the spread we compute the threshold such that ER = 0.1, and find the corresponding MRT. Each panel demonstrates an important fact, as we elucidate below.

In the left panel of Fig. 3 we take the signal vectors to be as (2), and allow ϕ to vary. Thus, $\phi = 0$ corresponds to the case when the signal is concentrated in a single channel. Positive values of ϕ correspond to signals that are spread about a peak. In these computations, the weights are the optimal weights, i.e. $w_{ij} \propto S_j^{\mu_i}$. This panel thus shows the minimal MRT that can be achieved by a threshold crossing test for an ER of 0.1. We see that there is an advantage to a moderate spread in the signals if this information can be utilized by the decision mechanism. In fact, the optimal spread is near $\phi = 3$. It is interesting to note that this corresponds to a width in the shape of the signal vectors of about 30° , while the width of tuning curves in MT associated with the direction task as measured in Law and Gold (2008) are approximately 40° .

In the right panel we fix the spread in the signal vectors at $\phi = 4$, and compute MRT for various spreads in the weights. In order to get an idea of how the spread in the shape of the weights affects

performance when the signal shape is fixed, in these simulations we suppose that the weights also have a Gaussian shape:

$$w_{ij} = w_0 \exp \left[-\frac{(i-j)^2}{2\phi_W^2} \right], \quad j = 1, \dots, n,$$

where w_0 is a normalizing factor chosen so that $\sum_{j=1}^n w_{ij}^2 = 1$. The spread ϕ_W controls how the values of the accumulators are weighted before making a decision. In the case $\phi_W = 0$, we have $y_i = x_i$, so that the accumulator values are not weighted. When $\phi_W = \infty$, each y_i is the same, i.e. the sum of all accumulators. The right panel of Fig. 3 shows that MRT is minimized when $\phi_W = \phi$. That is, the optimal weights occur when the width of the weight shape is the same as that in the signal vector.

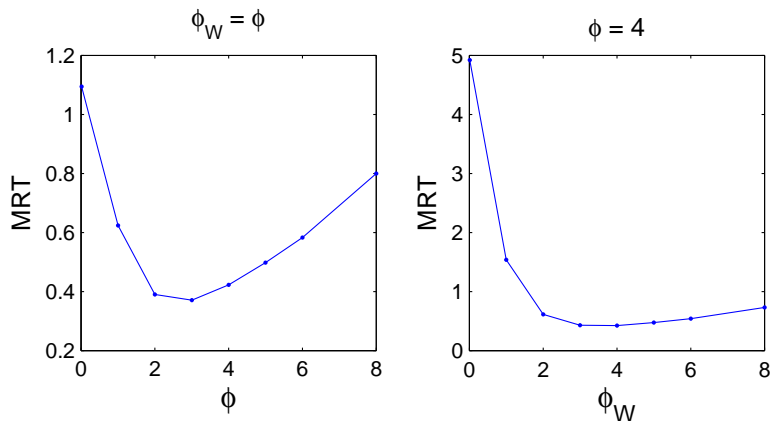


Figure 3: Effects of signal spread and weight shape. Left panel: MRT vs. spread in the signal vectors, where the weights have the same shape. Right panel: MRT vs. spread in shape of weights with signal vector fixed with $\phi = 4$. In all cases the threshold is such that $ER = 0.1$.

To reiterate, a moderate spread in the signals is a significant advantage, but only if the weights to SC can be tuned to take on the same shape as the possible signal vectors. In the following section we consider how the weights may be modified over the course of trials.

3 An algorithm for learning the LIP to SC weights

We propose a simple Hebbian weight learning algorithm for the weights w_{ij} . This algorithm is similar to the one employed by Law and Gold (2009) in a model of a two alternative task, but is simpler in many respects. The learning algorithm is a modification of a classical Widrow-Hoff rule. The theory of this update rule is described, e.g. in Hertz et al. (1991). In rules of this type, the connection strength being modified acts as a filter that tracks an input signal. At any point, its value is an exponentially decaying, weighted average of past input values. High frequency changes in this signal (representing noise) are filtered out by the algorithm, producing little change in the updated weight. In contrast, low frequency signal changes (representing, hopefully, the uncorrupted input signal) produce significant changes in the weight. If the signal is constant and noise is absent, the weight will converge exponentially on the value of the signal. If what is being tracked is a signal that depends on the product of activations in a sending unit and a receiving unit, then this rule is simply a Hebbian update rule with a decay term for forgetting old co-activation levels — a useful feature in a noisy neural system.

$S_i, i = 1, \dots, n$	given signals from MT
$dx_i = \left(-k x_i - w \sum_{j \neq i} x_j + S_i \right) dt + c dW_i$	accumulator evolution in LIP
$z_i = f(y_i), \quad y_i = \sum_{j=1}^n w_{ij} x_j$	decision units in SC
$w_{ij}^{\text{new}} = (1 - \alpha) w_{ij}^{\text{old}} + \alpha \Delta w_{ij}$	LIP to SC weight learning rule
$\Delta w_{ij} = r z_i x_j$	
$\sum_{j=1}^n w_{ij}^2 = w_0$	weight normalization (optional)

Table 1: Three layer model with weight learning rule.

After each trial the subject responds with a choice among alternatives, say i . At this time the weights to the output unit z_i corresponding to the choice made are updated, according to whether a reward is received or not. Then, if the choice corresponding to z_i is chosen, the weights are updated by the rule

$$w_{ij}^{\text{new}} = (1 - \alpha) w_{ij}^{\text{old}} + \alpha \Delta w_{ij}, \quad (5)$$

$$\Delta w_{ij} = r z_i x_j, \quad (6)$$

where r is the magnitude of the reward, and α is the learning rate. Notice that only the weights to the unit corresponding to the choice made are updated, and this is the sense in which the rule is Hebbian. In this discrete algorithm, we suppose that a reward is either earned or not so that r is either 1 or 0 depending on whether a correct decision is made. This rule can easily be modified to take into account a probability of receiving a reward for a correct response, but we do not consider such modifications here. In the free response protocol, z_i is always the same given a fixed threshold, but in the interrogation protocol, this value depends on the time elapsed in the trial. After each trial, the weights are normalized so that

$$\sum_{j=1}^n w_{ij}^2 = w_0. \quad (7)$$

We usually take $w_0 = 1$. The normalization (7) is thought to be a common feature of synaptic plasticity (Royer and Pare, 2003). Note that this normalization means that if an incorrect decision is made, then the weights are unchanged, since then $\Delta w_{ij} = 0$, and the normalization will cancel the multiplication by $1 - \alpha$. However, the normalization (7) is not a necessary feature of the model. Simulations with and without the normalization rule (7) behave similarly. Without the normalization the weight magnitudes will, of course, be different, but the shapes will be the same. Since the magnitude can be incorporated into the thresholds, overall there are no significant effects from the normalization. This is different from models such as in Law and Gold (2009), where such normalization is necessary for the stability of the algorithm. A summary of the model is in Table 1.

Thus, after each trial, if a correct decision is made the weights to the correct output unit are increased in proportion to the values of the accumulators \mathbf{x} . There is no need to estimate the probability of making a correct decision or an expected value of the reward, as e.g. in Law and

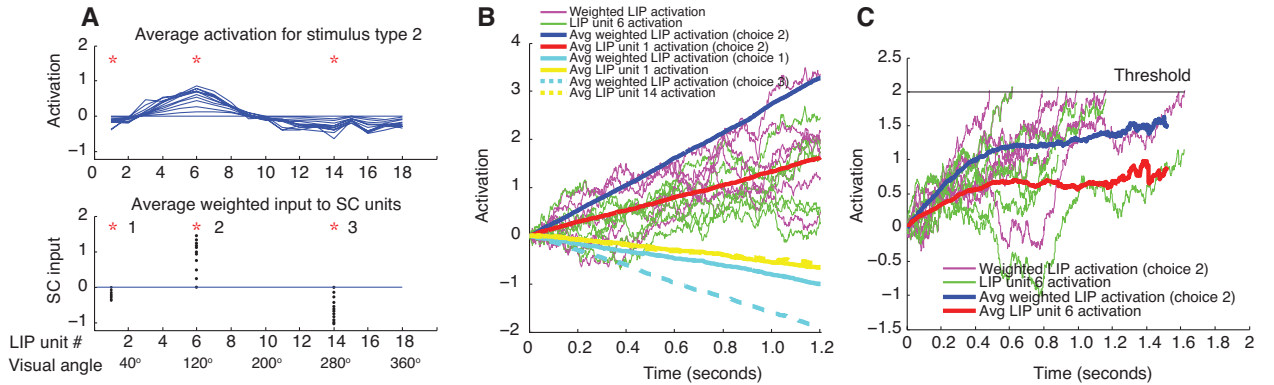


Figure 4: Panel A, top, shows LIP unit activations at several time points within a decision. Activations are averaged over many instances of stimulus type 2, which produces maximal activation in LIP unit 6 (visual angle 120°; here we arbitrarily quantized visual angle into 18 levels). Panel A, bottom, shows the weighted values of these activations feeding into each of 3 SC units. Panel B shows the average state of evidence accumulation for choice 2 (blue) and average LIP unit 6 activity (red) within fixed-viewing time trials, without thresholds applied to the evidence (the interrogation protocol). Here, cyan indicates the weighted evidence in favor of the other two responses, corresponding to stimuli with maximal activation at units 1 and 14 (10° and 200°, solid and dashed respectively). Yellow indicates activation in the units most sensitive to these stimuli. Individual timecourses of activity for 10 trials are shown for the weighted input to SC 2 (magenta) and for LIP 6 (green). These demonstrate that the weighted activation increases more rapidly, without an appreciable increase in noise. Panel C shows the average state of weighted evidence accumulation for choice 2 (blue) and average unit 6 activity (red) within free response trials (black line indicates threshold). The weighted sum produces a higher signal-to-noise ratio, and therefore better performance than evidence from unit 6 alone. Red and blue traces fall off over time because the average is based on fewer and fewer trials as time progresses (more and more decisions have already taken place by the end of the plot).

Gold (2009), as only the values of the units are used in the update rule. With this rule the weights track the shape of the vectors being passed from the LIP layer. The weights thus tend to oscillate around the means of the accumulator values, $\langle x_j(t) \rangle$.

The accumulator values on average take on the shape of the signal vector from the MT layer. This can be seen by writing (3) as

$$dx_i = \left(\lambda x_i - m \sum_{j=1}^n x_j + S_i \right) dt + c dW_i, \quad i = 1, \dots, n,$$

where $\lambda = m - k$ is the difference between inhibition and decay. Since the mean of the Ito integral $\int_0^t dW_i(\tau) d\tau$ vanishes (Gardiner, 2004), the mean value of the accumulators obey the ordinary differential equation

$$\frac{d}{dt} \langle x_i(t) \rangle = \lambda \langle x_i(t) \rangle - m \left\langle \sum_{j=1}^n x_j(t) \right\rangle + S_i.$$

Thus, the mean value of the i th accumulator is

$$\langle x_i(t) \rangle = S_i \frac{e^{\lambda t} - 1}{\lambda} - m \int_0^t e^{\lambda(t-\tau)} \left\langle \sum_{j=1}^n x_j(\tau) \right\rangle d\tau.$$

We calculate the mean of the sum of the accumulators as follows. (See also §4.4 in McMillen and Holmes (2006).) We sum the equations (3) to deduce that

$$d \sum_{i=1}^n x_i = \left(-(k + (n-1)m) \sum_{i=1}^n x_i + \sum_{i=1}^n S_i \right) dt + c \sum_{i=1}^n dW_i.$$

Since the sum of Wiener processes has mean zero, the mean of the sum is

$$\left\langle \sum_{j=1}^n x_j(t) \right\rangle = \sum_{j=1}^n S_j \frac{1 - e^{-\hat{\lambda} t}}{\hat{\lambda}},$$

where $\hat{\lambda} = k + (n-1)m$. Thus, we have

$$\langle x_i(t) \rangle = S_i \frac{e^{\lambda t} - 1}{\lambda} - \frac{m}{\hat{\lambda}} \sum_{j=1}^n S_j \left[\frac{e^{\lambda t} - 1}{\lambda} - \frac{e^{\lambda t} - e^{\hat{\lambda} t}}{\lambda + \hat{\lambda}} \right]. \quad (8)$$

Notice, now, that the mean of $x_i(t)$ has the form

$$\langle x_i(t) \rangle = S_i \cdot c_1(t) + c_2(t),$$

where c_1 and c_2 are functions of t , but they do not depend on i . Thus the mean of the vector of accumulators takes the shape of the vector of signals from MT.

The update rule (5-6) thus causes the weights to track values whose means take on the shape of the signal vectors. Therefore the weights tend, on average, to mimic the shape of the signal vector with oscillations about this shape that depend on the learning rate.

4 Results of simulations

Figure 5 shows results of simulations using the update rule (5 - 6) in the free response protocol. The weights are initially chosen randomly, with a peak added at w_{ij} . We see how the weights evolve over time, and how this affects the performance of the subject. The reward rate continually increases on average, and the ER continually decreases. The bottom panels show the weights to SC corresponding to $i = 14$, or to angle 140° . The weights for the other alternatives behave similarly. Simulations in which the weights are chosen differently show similar improvements in performance and similar matching of the weight profiles to the signal vector shapes. Cases in which the weights are all chosen randomly show a more dramatic improvement in RR since then the accuracy will initially be very low. Fig. 5 shows that even when the weight has a peak at the right position, a dramatic improvement occurs: for example, the RR more than doubles and the RT and ER both decrease over time.

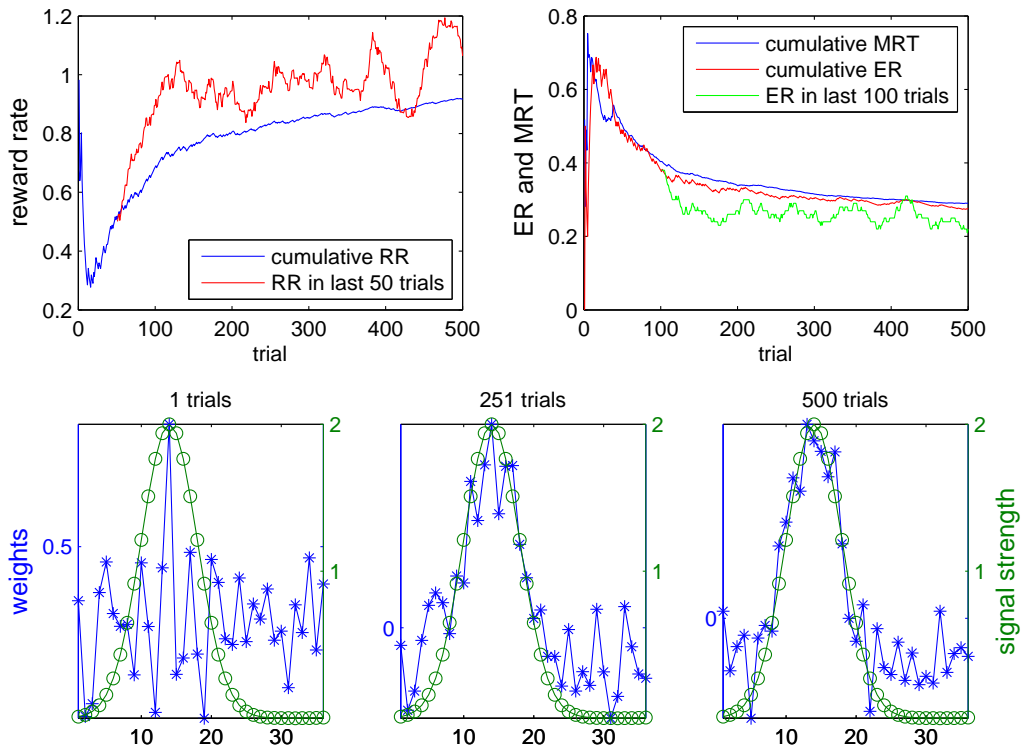


Figure 5: Effects of weight learning rule. The threshold is fixed at $\theta = 1$. There are four alternatives (3, 6, 14, 22), and the learning rate is $\alpha = .05$. In the bottom panel the signal strength is plotted on the right axis (circles), and the weights are shown on the left axis (stars). The RSI used in the calculation of RR is $D = 0.5$.

Figure 5 shows one block of 500 trials. In order to see how the weight update rule behaves on average, we carried out the same simulation for a number of blocks and then averaged the weights over each block, and then took the average over 150 blocks of trials. The values of such averaged weights are seen in Fig. 6. In this figure we show the averaged weights for different values of the threshold, as well as different values of the spread in the signals. We see that on average, the weight

profile shape is close to the signal shape. Also shown in these figures are the error rates and mean reaction times for these blocks of trials. Notice that in the lower left panel, the $ER = .58$ is not much smaller than would be achieved by random guessing (.75). In this case the threshold is very small, as is the corresponding MRT of .09. In this situation it will take the weights much longer to learn the shape of the signal vectors, since most of the time the decision will be incorrect. This is why the weights appear more erratic in this frame than in the others. However, even in this case, the average values of the weights take the same shape as the signal vector. Similar comments apply, *mutatis mutandis*, to the upper left panel.

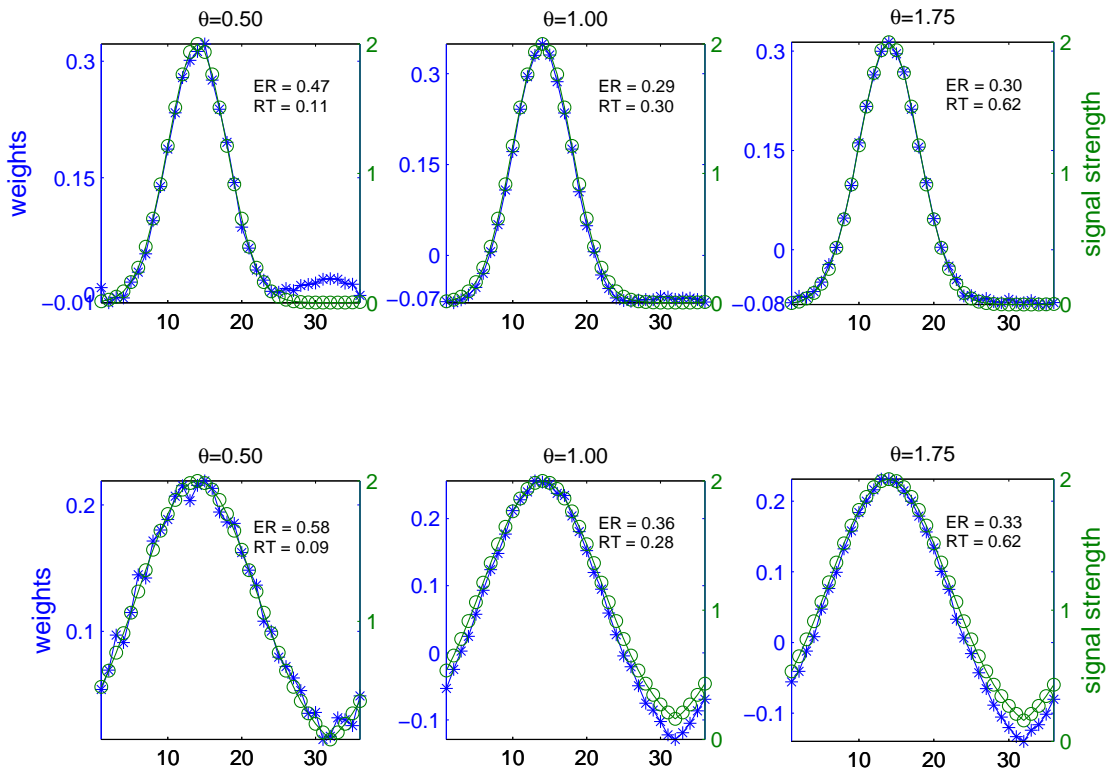


Figure 6: Averaged weights over 150 blocks of 500 trials. In the top row $\phi = 4$; in the bottom row $\phi = 8$.

We also performed simulations using different shapes for the signal vectors, in order to verify that the Gaussian shape achieved by the weight vectors are not simply an artifact of the Gaussian noise in the equation for the LIP units. This is seen in Fig. 7. Here we plot the averaged weights using cosine and $1 - |x|$ shapes for the signal vectors. As with the Gaussian shapes, we see that the weight vectors tend, on average, to the shape of the signal vectors.

Additionally, we performed a variety of simulations to examine the effects of the various parameters in the model. Generally, the model is insensitive to changes in the parameters a, c, k, w , in the sense that the weights tend on average toward the optimal weight shape mimicing the shape of the signal vectors. If the learning rate α is made smaller, the weights take longer to track to the shape of the signals, but there is less variation around these mean values.

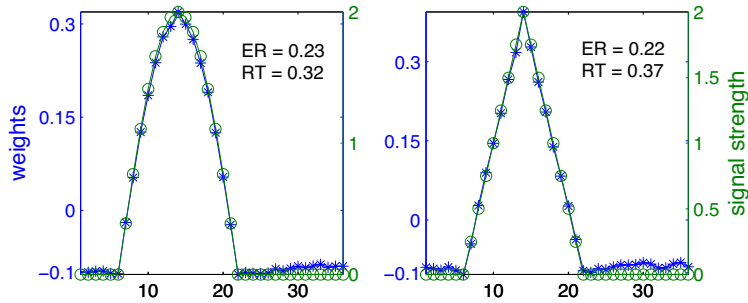


Figure 7: Averaged weights with cosine and $1 - |x|$ shapes.

5 Discussion

The simple rule (5-6) works remarkably well at learning the shapes of the signal vectors from MT to LIP. This leads to a dramatic improvement in performance, and occurs without any direct connection to the MT layer. The three layer model incorporates integration of information, a rule for making the decision, as well as a simple algorithm for learning to optimize reward rates by learning the shapes of the vectors of neural signals coming from an input layer. These components are the essential aspects of a complete decision-theoretic model.

A great advantage of the simple model we have proposed is its flexibility. For instance, one can take a continuous limit in the update rule (5-6). The equation (5) may be rewritten as

$$\frac{w_{ij}^{\text{new}} - w_{ij}^{\text{old}}}{\alpha} = -w_{ij}^{\text{old}} + rz_i x_j.$$

Therefore, supposing that the weights w_{ij} are functions of time and that w_{ij}^{new} is the update to the weight w_{ij}^{old} after a time step of α , we may take the limit as this time step approaches zero to obtain the continuous version of the update rule:

$$\frac{dw_{ij}(t)}{dt} = -w_{ij}(t) + r(t)z_i(t)x_j(t). \quad (9)$$

Notice that if the threshold θ for the y_i 's is fixed, an overall increase in the weights w_{ij} has the effect of amplifying the accumulators. Thus, the threshold is crossed sooner, and the effective threshold is thus reduced. The continuous version of the update rule (9) can thus be modified to a scheme that not only updates the shapes of the weights, but the normalizing factor w_0 of the weights, and hence the overall strength of the weights.

Such an algorithm was used in the case of two-alternatives, with Kronecker delta tuning curves, in Simen and Cohen (2009). In the two-alternative context in which different responses are rewarded at random intervals, but with different expected delays to reward, the model selects the more frequently rewarded response more often (the ratio of responses of each type in fact matches the ratio of rewards earned for each response type, consistent with the ‘matching law’ of behavioral psychology (Herrnstein, 1997)). In a work in progress we examine a modified version of this approach that not only updates the thresholds but the weight shapes defining a tuning curve as well. Thus the continuous-time version of the algorithm already has a promising connection to well-known behavioral findings.

The model is amenable to modifications to account for changing conditions. Reward rates are optimized when the weight shapes are fixed at the shapes of the possible signal vectors. However, the rule (5-6) incorporates a variation around the means of these signals if the learning rate $\alpha > 0$.

In a sequence of trials, one may reduce α over time so that these variations continually decrease. In effect, one learns the incoming signals by first allowing α to be moderately large, and then reducing α as the weights get closer and closer to the means of the accumulators in the LIP layer.

In a future work we intend to explore the ways in which this model may be modified to account for situations in which various aspects such as the RSI, number of alternatives and signal-to-noise ratio are changing from trial to trial.

Throughout this paper we have taken the direction movement task as our canonical example. However, it is clearly applicable to a wider range of decision making problems. An interesting line of research is to explore how general such models are.

References

- Beck, J., Ma, W., Kiani, R., Hanks, T., Churchland, A., Roitman, J., Shadlen, M., Latham, P., and Pouget, A. (2008). Probabilistic population codes for bayesian decision making. *Neuron*, 60:1142–1152.
- Bogacz, R., Brown, E., Moehlis, J., Hu, P., Holmes, P., and Cohen, J. (2006). The physics of optimal decision making: A formal analysis of performance in two-alternative forced choice tasks. *Psych. Rev.*, 113(4):700–765.
- Bogacz, R. and Gurney, K. (2007). The basal ganglia and cortex implement optimal decision making between alternative actions. *Neural Comput.*, 19(2):442–477.
- Butts, D. A. and Goldman, M. S. (2006). Tuning curves, neuronal variability, and sensory coding. *PLoS Biol*, 4(4):e92.
- Churchland, A., Kiani, R., and Shadlen, M. (2008). Decision-making with multiple alternatives. *Nat. Neuroscience*, 11(6):693–702.
- Furman, M. and Wang, X. (2008). Similarity effect and optimal control of multiple-choice decision making. *Neuron*, 60:1153–1168.
- Gardiner, C. (2004). *Handbook of Stochastic Methods*. Berlin: Springer-Verlag, 3rd edition.
- Ghose, G., Yang, T., and Maunsell, J. (2002). Physiological correlates of perceptual learning in monkey v1 and v2. *J. Neurophysiol.*, 87:1867–1888.
- Herrnstein, R. (1997). *The Matching Law: Papers in Psychology and Economics*. Harvard University Press, Cambridge, MA.
- Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neural computation*. Santa Fe Institute Studies in the Sciences of Complexity. Lecture Notes, I. Addison-Wesley Publishing Company Advanced Book Program, Redwood City, CA.
- Law, C.-T. and Gold, J. (2008). Neural correlates of perceptual learning in a sensory-motor, but not a sensory cortical area. *Nat. Neurosci.*, 11:505–513.
- Law, C.-T. and Gold, J. (2009). Reinforcement learning can account for associative and perceptual learning on a visual-decision task. *Nat. Neurosci.*, 12(5):655–663.
- McMillen, T. and Behseta, S. (2009). On the effects of signal acuity in a multi-alternative model of decision making. *Neural Comput.*

- McMillen, T. and Holmes, P. (2006). The dynamics of choice among multiple alternatives. *J. Math. Psych.*, 50(1):30–57.
- Niwa, M. and Ditterich, J. (2008). Perceptual decisions between directions of visual motion. *J. Neurosci.*, 28(17):4435–4445.
- Pouget, A., Deneve, S., Ducom, J.-C., and Latham, P. (1999). Narrow vs. wide tuning curves: what’s best for a population code? *Neural Comput.*, 11:85–90.
- Royer, S. and Pare, D. (2003). Conservation of total synaptic weight through balanced synaptic depression and potentiation. *Nature*, 422:518–522.
- Seriés, P., Latham, P., and Pouget, A. (2004). Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nat. Neurosci.*, 7(10):1129–1135.
- Simen, P. and Cohen, J. (2009). Explicit melioration by a neural diffusion model. *Brain Res.*
- Usher, M. and McClelland, J. (2001). On the time course of perceptual choice: The leaky competing accumulator model. *Psych. Rev.*, 108:550–592.
- Zhang, K. and Sejnowski, T. (1999). Neural tuning: to sharpen or broaden? *Neural Comput.*, 11:75–84.