Statistical Mechanics of Reward-Modulated Learning in Decision-Making Networks

Kentaro Katahira
katahira@mns.k.u-tokyo.ac.jp
Japan Science Technology Agency, ERATO, Okanoya Emotional Information Project, 351-0198 Saitama, Japan; Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-5861 Japan; and RIKEN Brain Science Institute, 351-0198 Saitama, Japan

Kazuo Okanoya
okanoya@brain.riken.jp
Japan Science Technology Agency, ERATO, Okanoya Emotional Information Project, 2-1 Hirosawa, Wako, 351-0198 Japan; RIKEN Brain Science Institute, 351-0198 Saitama, Japan; and Graduate School of Arts and Sciences, University of Tokyo, 153-8902 Tokyo, Japan

Masato Okada
okada@k.u-tokyo.ac.jp
Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-5861 Japan; Japan Science Technology Agency, ERATO, Okanoya Emotional Information Project, 351-0198 Saitama, Japan; and RIKEN Brain Science Institute, 351-0198 Saitama, Japan

The neural substrates of decision making have been intensively studied using experimental and computational approaches. Alternative-choice tasks accompanying reinforcement have often been employed in investigations into decision making. Choice behavior has been empirically found in many experiments to follow Herrnstein’s matching law. A number of theoretical studies have been done on explaining the mechanisms responsible for matching behavior. Various learning rules have been proved in these studies to achieve matching behavior as a steady state of learning processes. The models in the studies have consisted of a few parameters. However, a large number of neurons and synapses are expected to participate in decision making in the brain. We investigated learning behavior in simple but large-scale decision-making networks. We considered the covariance learning rule, which has been demonstrated to achieve matching behavior as a steady state (Loewenstein & Seung, 2006). We analyzed model behavior in a thermodynamic limit where the number of plastic synapses went to infinity. By means of techniques of the statistical mechanics, we can derive deterministic differential equations in this limit for the order parameters, which allow an exact calculation...
of the evolution of choice behavior. As a result, we found that matching behavior cannot be a steady state of learning when the fluctuations in input from individual sensory neurons are so large that they affect the net input to value-encoding neurons. This situation naturally arises when the synaptic strength is sufficiently strong and the excitatory input and the inhibitory input to the value-encoding neurons are balanced. The deviation from matching behavior is caused by increasing variance in the input potential due to the diffusion of synaptic efficacies. This effect causes an undermatching phenomenon, which has been often observed in behavioral experiments.

1 Introduction

Decision making is a high-level cognitive process that chooses one action from two or more alternatives based on the expectation of a future outcome predicted from given information and past experience. Since the outcomes of decision making can be measured as the behavior of subjects, it has been traditionally studied by using behavioral experiments. However, its neural substrates have been extensively investigated in the past few decades in neurophysiological and imaging studies (Sugrue, Corrado, & Newsome, 2005; Gold & Shadlen, 2007; Rushworth & Behrens, 2008). Computational approaches have been used to bridge the gap between low-level cellular mechanisms and high-level cognitive functions (for a review, see Wang, 2008).

Decision making has often been studied in experiments in which a subject repeatedly chooses actions and rewards are given depending on the action. The choice behavior of subjects in such experiments is known to obey Herrnstein’s matching law (Herrnstein, Rachlin, & Laibson, 1997). This law states that the proportional allocation of choices matches the relative reinforcement obtained from those choices. The neural correlates of matching behavior have been investigated (Sugrue, Corrado, & Newsome, 2004; Corrado, Sugrue, Seung, & Newsome, 2005) and the computational models that explain them have been interesting (Loewenstein & Seung, 2006; Soltani & Wang, 2006; Sakai & Fukai, 2008a, 2008b; Loewenstein, 2008).

Models with a small number of parameters have mainly been analyzed in previous studies (Loewenstein & Seung, 2006; Sakai & Fukai, 2008a, 2008b; Loewenstein, 2008) with a few exceptions in simulation studies (Seung, 2003; Soltani & Wang, 2006). However, a large number of neurons and synapses should participate in decision making in a real brain. In this study, we analyzed the learning process in a limit where the number of plastic synapses was infinite by means of a statistical mechanics approach (Kinzel & Rujan, 1990; Saad, 1998; Reents & Urbanczik, 1998). We took into consideration two specific learning rules that belonged to a class of the covariance rule proposed by Loewenstein and Seung (2006): (1) the
reward-modulated Hebb and (2) the delta rules. For both rules, we found qualitative differences between a network consisting of a large number of plastic synapses and small networks with few synapses. The goal of this study is not to construct a realistic model of biophysical neural circuits that can be directly compared to physiological and behavioral data. Rather, we intend to provide qualitative results that would provide various insights into what would happen if many synaptic weights simultaneously changed according to rewards and actions.

In this study, we considered the limit of a large number of synapses with two scalings: (1) a large input fluctuation case, where the individual fluctuations in input of sensory neurons can affect fluctuations in output neurons, and (2) a small input fluctuation case, where individual fluctuations in input neurons are averaged out in output potential and individual fluctuations cannot affect the stochasticity of model output. In the latter, the model can be reduced to a case with a single neuron per stimulus, which Loewenstein (2008) analyzed. To investigate the effect of individual fluctuation in sensory neurons, we mainly concentrated on the former case. Even if the fluctuations of sensory neurons are small, the effect of fluctuations can appear when the number of neurons is finite and the finite size effect is not negligible. Thus, analyzing cases with large fluctuations in input would provide important insights into learning behavior in decision-making networks. In addition, under some assumptions, our model with the large input fluctuation is mathematically equivalent to the more realistic balanced excitation and inhibition model with “strong synapses” in which each single synapse out of $N$ synapses has a contribution of $O(1/\sqrt{N})$, rather than $O(1/N)$, which implies a large mean of order $\sqrt{N}$ and finite variance of order 1. This interpretation is discussed in section 7.3 and illustrated with a recurrent spiking neuron network model in appendix E.

Previous studies have taken into consideration learning rules that achieve matching behavior as an equilibrium point in learning (i.e., the rules make the average update vanish at a point where the network demonstrates matching behavior; Herrnstein & Vaughn, 1980; Loewenstein & Seung, 2006; Sakai & Fukai, 2008a; Loewenstein, 2008). However, we found that matching behavior is not necessarily a steady state in our large-scale decision-making networks for the case of large fluctuations in input. This is caused by the diffusion of synaptic weights. In this letter, the term diffusion refers to a phenomenon where the distribution over the population of synaptic weights broadens due to individual fluctuations in presynaptic neurons. This diffusion increases the variance in the potential of output units since the broader synaptic weight distributions are, the more they amplify fluctuations in individual inputs. This makes the choice behavior of the network more random and moves the probabilities of choosing alternatives to equal probabilities than that predicted by the matching law. Previous studies have shown that covariance rules lead to matching behavior, and thus synaptic weights converge to finite values even without any constraints.
on the weight updates (Loewenstein & Seung, 2006; Loewenstein, 2008). In the case with large fluctuations in input with large-scale networks, however, the weights increase without limits due to the diffusion effect if weight constraints are absent. Nevertheless, the choice probability of a network with diverging weights asymptotically approaches matching behavior. If we impose a weight-normalization constraint, the diffusion effect becomes more evident than in cases without normalization. In addition, the greater the learning rate, the larger the diffusion effect. We also found that the delta rule was more susceptible to the diffusion effect than the reward-modulated Hebb rule. Thus, the deviation from matching behavior is relatively large for the delta rule. In the variants of the covariance-learning rules we took into account, the differences between actual rewards and the expected values of the rewards were used to drive learning. We explored what effect mistuning the subtraction baseline had on the learning behavior. Mistuning the baseline causes deviation from the matching behavior, as has been reported (Loewenstein, 2008). We found that the effect of mistuning was asymmetric in the direction of mistuning. Furthermore, the reward-modulated Hebb rule is more robust against mistuning than the delta rule.

Our results suggest that when we discuss the learning processes in a decision-making network, it may be insufficient to consider only a steady state for individual weight updates, and we should therefore consider the dynamics of the weight distribution and the network architecture. To the best of our knowledge, this study is the first to analyze learning processes in large-scale decision-making networks by means of a statistical mechanical approach. The short version of this letter with a different model formulation is in Katahira, Okanoya, and Okada (2010).

2 Formulation of Matching Law

First, let us formulate the matching law. Let $a = 1, \ldots, n_a$ denote the indices of multiple alternatives. Here, we consider stochastic choice behavior, where at each time step, an agent chooses alternative $a$ with probability $p_a$, without any dependence on previous actions or rewards. We denote the reward as $r$. For simplicity, we restrict $r$ to a binary: $r = 0$ represents the absence of a reward, and $r = 1$ means that a reward is given. The expected return, $\langle r | a \rangle$, refers to the average reward per choice $a$, and the income, $I_a$, refers to the total amount of reward resulting from the choice, $a$. We summarize the notations as:

- $p_a$: Probability of choosing $a$.
- $\langle r | a \rangle$: Expected return from choice $a$ (the average reward given when action $a$ is chosen).
- $I_a$: Income (total amount of reward) from choice $a$.
- $I_a / (\sum_{a'} I_a)$: Fractional income from choice $a$. For a large number of trials, this should equal $\langle r | a \rangle p_a$. 
Figure 1: Decision-making network architecture. The network consists of two groups, each encoding two alternatives. Each group consists of $N$ sensory-input neurons denoted by $x_{ai} (i = 1, \ldots, N$ and $a = A, B)$ and output units whose potential and final output correspond to $u^a$ and $y^a$.

- $\langle r \rangle = \sum_{a} \langle r|a \rangle p_a$: Average reward per trial over possible choice behavior.

The matching law states that the proportional allocation of choices matches the fraction of income obtained from those choices. The proportional allocation of choices converges to $p_a$ for a large number of trials. For a large number of trials, the fraction of income from an alternative $a$ is expressed as

$$\frac{\langle r|a \rangle p_a}{\sum_{a'} \langle r|a' \rangle p_{a'}} = \frac{\langle r|a \rangle p_a}{\langle r \rangle}$$

The matching law states that the above quantity equals $p_a$ for all alternatives. To make this hold, it should satisfy

$$\langle r|1 \rangle = \langle r|2 \rangle = \cdots = \langle r|n_a \rangle = \langle r \rangle.$$  (2.1)

Note that $\langle r|a \rangle$ is the average reward given the current choice, and this is a function of the past choice. Equation 2.1 is a necessary and sufficient condition for the matching law (except for the case in which some alternative is never chosen), and we will often use this identity.

3 Model

3.1 Decision-Making Network. Let us consider a feedforward decision making network that consists of sensory-input neurons and output neurons that represent the subjective value of each alternative (we call the output neurons value-encoding neurons) (see Figure 1). This model is introduced
by simply extending Loewenstein and Seung’s model (2006; Loewenstein, 2008). We will consider only a case with two alternatives (each denoted as A and B), which has generally been studied in animal experiments. The network is divided into two groups (A and B), which participate in choosing each alternative. Sensory cues from both targets are given simultaneously via the N-neuron population, \( x^A = (x^A_1, \ldots, x^A_N) \) and \( x^B = (x^B_1, \ldots, x^B_N) \). With the synaptic efficacies (or weights) \( J^A = (J^A_1, \ldots, J^A_N) \) and \( J^B = (J^B_1, \ldots, J^B_N) \), the internal potentials of output units are given by

\[
  u^a = \sum_{i=1}^{N} J^a_i x^a_i, \quad a = A, B. \tag{3.1}
\]

The choice is made in such a way that alternative \( a \) is chosen if the potential of output unit \( u^a \) is higher than that of the other alternative. Although we do not model this comparison process explicitly, it can be carried out via a winner-take-all competition mediated by feedback inhibition, as has been commonly assumed in decision-making networks (Wang, 2002; Loewenstein & Seung, 2006). In this competition, the “winner” group gains a high firing rate while the “loser” enters a low firing state (Wang, 2002). Let \( y^A \) and \( y^B \) denote the final output of an output neuron after competition, and this is determined as

\[
  y^A = 1, \quad y^B = 0, \quad \text{if} \quad u^A \geq u^B, \\
  y^A = 0, \quad y^B = 1, \quad \text{if} \quad u^A < u^B.
\]

Each component of input vectors \( x^A \) and \( x^B \) independently obeys a gaussian distribution.\(^1\) We consider two types of scaling for the mean and variance of the gaussian distribution:

1. Case with large fluctuations in input—mean \( 1/N \) and variance \( 1/N \). These scalings for \( N \) ensure the output potential, \( u^a \), is in the finite order (\( \sim O(1) \)) and behaves stochastically. Due to the scaling of variance \( O(1/N) \), the fluctuations in input can cause fluctuations in the net input to the value-encoding neuron and make stochastic decision making possible.

2. Case with small fluctuations in input—mean \( 1/N \) and variance \( 1/N^2 \). Here, the fluctuations in output vanish in the limit \( N \to \infty \).

In the latter, the model can be reduced to a case with a single neuron per stimulus, which was analyzed by Loewenstein (2008) and that we explain later (see section 5). To investigate the effect of individual fluctuations in

---

1Theoretically, this can be replaced with other distributions with the same mean and the second moment.
sensory neurons, we mainly concentrate on the former case. In the former case, the ratio of the standard deviation to the mean activity of the sensory neurons diverges \((1/\sqrt{N})/(1/N) = \sqrt{N} \to \infty\). This may seem to be an unrealistic assumption. However, we used this scaling mainly for clarifying the effect of individual fluctuations in input neurons. A possible neural implementation of this scaling is discussed in section 7.3. If we set \(N = 1\), the model structure in either case is equivalent to that of Loewenstein and Seung’s (2006; Loewenstein, 2008).

### 3.2 Learning Rules

Let us consider learning rules that belong to the class of the covariance learning rule (Loewenstein & Seung, 2006). According to the covariance rule, the synaptic weight, \(J^a_i(t)\), is updated as

\[ J^a_i(t + 1) = J^a_i(t) + \Delta J^a_i(t) \]

with any one of

\[ \Delta J^a_i(t) = \eta (r_t - \bar{r}) N^a_i, \tag{3.2} \]

\[ \Delta J^a_i(t) = \eta r_t (N_i - \bar{N}_i^a), \tag{3.3} \]

\[ \Delta J^a_i(t) = \eta (r_t - \bar{r}) (N^a_i - \bar{N}^a_i), \tag{3.4} \]

where \(\eta\) is the learning rate, \(\bar{\cdot}\) denotes the expected value, and \(N^a_i\) is a measure of neural activity related to the synapse, \(J^a_i\) (Loewenstein & Seung, 2006). The expectation of these updates is proportional to covariance between the reward, \(r\), and neural activity \(N^a_i\). In this study, we focus on the first variant of the learning rule (see equation 3.2). Although our analysis can easily be applied to the following two cases, they would produce no quantitative differences. For the specific form of \(N^a_i\), we consider two cases: (1) the product of input neuron activity \(x^a_i\) and output activity \(y^a\) and (2) input neuron activity \(x^a_i\) itself. The first rule is called the reward-modulated Hebb rule, and the second is called the delta rule. Variants of the reward-modulated Hebb rule have recently been studied intensively (Soltani & Wang, 2006; Soltani et al., 2006; Fusi, Asaad, Miller, & Wang, 2007; Izhikevich, 2007; Florian, 2007; Farries & Fairhall, 2007; Legenstein, Pecevski, & Maass, 2008). The delta rule has been used as an example of the covariance rule (Loewenstein & Seung, 2006; Loewenstein, 2008) and has also been used for the learning rule in the model of perceptual learning (Law & Gold, 2009). The expected reward, \(\bar{r}\), can be estimated, for example, by calculating the running average as \(\bar{r} \leftarrow (1 - \gamma) r_t + \gamma \bar{r}\) with a constant, \(\gamma\) \((0 \ll \gamma < 1)\). It is difficult to accurately estimate the expected reward in a real brain. Thus, to include a case where the reward baseline has deviated from the true expected reward, we introduce a mistuning parameter, \(\beta\), as Loewenstein (2008) did. The subtraction baseline is now represented as \(\beta \bar{r}\). Here, \(\beta = 1\) corresponds to the true covariance learning case, \(\beta < 1\) is an underestimate of the reward, and \(\beta > 1\) is an overestimate of the reward.
Adding the mistuning parameter, $\beta$, we consider two learning rules in this study: the reward-modulated Hebb rule:

$$J_a^i(t+1) = J_a^i(t) + \eta (r_t - \beta \bar{r}) y^a_i,$$

and the delta rule:

$$J_a^i(t+1) = J_a^i(t) + \eta (r_t - \beta \bar{r}) x^a_i. \quad (3.5)$$

### 3.3 Reinforcement Schedule.

We adopted the competitive foraging task that Sakai and Fukai proposed (2008a) as the reinforcement schedule for training the model. This schedule combines two commonly used reinforcement schedules for behavioral experiments: variable interval (VI) and variable ratio (VR) schedules. Here, we assume the time is broken into discrete trials and the sequence of trials is labeled in discrete time steps. In the VI schedule task, a reward is assigned to all alternatives $a$ ($a = 1, 2, \ldots, n_a$) stochastically and independently, with a constant probability, $\lambda_a$. The reward remains until it is harvested by choosing the alternative. In the VR schedule task, a reward is assigned to each alternative independently with constant probabilities $\lambda_a$, but the reward is immediately withdrawn unless the subject chooses the alternative and harvests the reward. The competitive foraging task introduces a “withdrawal rate” denoted by $\mu_a$. With a probability of $\mu_a = 0$, $\forall a'$, the task corresponds to the VI schedule while $\mu_a = 1$, $\forall a'$ corresponds to the VR schedule. An intermediate region of $\mu_a$ provides mixtures of the VI and VR schedules. For pure VI and pure VR, the matching behavior coincides with the choice probability that gives the maximum reward for independent choice behaviors. However, for mixtures of VI and VR ($0 < \mu < 1$), the matching behavior deviates from the reward-maximizing choice probability. After this, we will consider only the case of two alternatives. The expected return from alternative $a$ is calculated as

$$\langle r | a \rangle = \frac{\lambda_a p_a}{1 - (1 - \lambda_a)(1 - \mu_a)(1 - p_a)}. \quad (3.7)$$

For details on the derivation, see Sakai and Fukai (2008a). The choice probability that gives matching behavior and reward-maximizing choice probability can also be found there. Figure 2 outlines the expected return and the expected income as a function of the choice probability, $p_A$, for an example of a competitive foraging task. We can see that the $p_A$ that yields matching behavior ($p_A^{\text{match}} = 0.6928$) deviates from those that yield maximizing behavior ($p_A^{\text{max}} = 0.5192$). In the results that follow, we have mainly used this reinforcement schedule setting.
4 Macroscopic Description of Model Behavior: The Case of Large, Fluctuating Input

This section describes model behavior in the thermodynamic limit ($N \to \infty$) by using order parameters. To do this, we first define the order parameters:

$$l_a = \frac{1}{\sqrt{N}} \| J^a \|, \quad \bar{J}_a = \frac{1}{N} \sum_{i=1}^{N} J_{ai}^a, \quad a = A, B. \quad (4.1)$$

For the case of large fluctuations in input, we find from simple calculation that $u^A$ and $u^B$ obey independent gaussian distributions whose means and variances are respectively given by $\bar{J}_a$ and $l_a^2$, ($a = A$ and $B$) (see Figure 3). From this, the probability that the network will choose alternative $A$ can be described as

$$p_A = \frac{1}{2} \text{erfc} \left\{ \frac{\bar{J}_A - \bar{J}_B}{\sqrt{2(l_A^2 + l_B^2)}} \right\}, \quad (4.2)$$

Figure 2: Expected return and expected reward per trial as function of choice probability, $p_A$, in competitive foraging task. Assignment rates are $\lambda_A = 0.4$ and $\lambda_B = 0.35$, and withdrawal rates are $\mu_A = 0.1$ and $\mu_B = 0.4$. For these parameter settings, the probability that corresponds to matching behavior is 0.6928 and one that maximizes expected reward is 0.5192 (they are indicated by vertical dotted lines).
Figure 3: The distribution of two outputs $u^A$ and $u^B$ is a two-dimensional gaussian whose mean and standard deviation are expressed by order parameters $\bar{J}_a$ and $l_a$, respectively, for the case of large fluctuations in input. The ellipse represents the contour of two standard deviations. Symbols are obtained by simulation with network ($N = 1000$). Open circles $\circ$ represent a plot where $u^A > u^B$; thus alternative A is chosen. Crosses $+$ represent opposite case.

where $\text{erfc}(\cdot)$ is the complementary error function, $\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$. The complementary error function is a monotonically decreasing function; thus, if $\bar{J}_A - \bar{J}_B > 0$, then $p_A > p_B$. This expression is in a closed form of the order parameter. Thus, if we can describe the evolution of these order parameters, we can completely describe how the behavior of the model changes as a consequence of learning. In the following, we often use an additional order parameter, the variance of weight, $\sigma_a^2$. This parameter is more convenient for gaining insights into the evolution of the weight than the weight norm, $l_a$. The order parameter set, $(\bar{J}_A, \bar{J}_B, \sigma_A^2, \sigma_B^2)$, gives equivalent information with the set, $(\bar{J}_A, \bar{J}_B, l_A, l_B)$, because of the relationship

$$\sigma_a^2 = l_a^2 - \bar{J}_a^2.$$

The diffusion of weight distributions is reflected by increases in $\sigma_a^2$—the differences between the growth of the second-order moment of weight distribution $l_a^2$ and that of the square of its mean $\bar{J}_a^2$.

**4.1 Evolution Equation for Order Parameters without Weight Normalization.** Next, following the statistical mechanical analysis of online learning (Kinzel & Rujan, 1990; Saad, 1998; Reents & Urbanczik, 1998), we derive equations that describe the evolution of the order parameters. To do this, we first rewrite the learning rule in vector form:

$$J^t(t + 1) = J^t(t) + F_a x^d.$$

(4.4)
where for the reward-modulated Hebb rule, \( F_a = \eta (r_t - \beta \bar{r}_t) y^a \), and for the delta rule, \( F_a = \eta (r_t - \beta \bar{r}_t) \). Taking the square norm of each side of equation 4.4, we obtain

\[
l_a(t + 1)^2 = l_a(t)^2 + \frac{2}{N} F_a(t) u^a + \frac{1}{N} F_a(t)^2 + O \left( \frac{1}{N^2} \right).
\]

Summing up over all components on both sides of equation 4.4, we obtain

\[
\bar{J}_a(t + 1) = \bar{J}_a(t) + \frac{1}{N} F_a(t) \tilde{x}_a,
\]

where we have defined \( \tilde{x}_a = \sum_{i=1}^{N} x_i^a \). In both equations, the magnitude of each update is of order \( 1/N \). Therefore, to change the order parameters of order one, \( O(N) \) updates are needed. Within this short period that spans the \( O(N) \) update, the weight change in \( O(1/N) \) can be neglected, and the self-averaging property holds, that is, the amount of update within the \( N \) update converges to the ensemble average of each update multiplied by \( N \) because of the law of large numbers (for more rigorous discussion, see Reents & Urbanczik, 1998). By using this property and introducing continuous “time” scaled by \( N, \alpha = t/N \), the evolutions of the order parameters obey ordinary differential equations:

\[
\frac{dI_a^2}{d\alpha} = 2 \langle F_a u^a \rangle + \langle F_a^2 \rangle, \tag{4.5}
\]

\[
\frac{d\bar{J}_a}{d\alpha} = \langle F_a \tilde{x}_a \rangle, \tag{4.6}
\]

where \( \langle \cdot \rangle \) denotes the ensemble average over all possible inputs and arrivals of rewards. The specific form of the ensemble averages is obtained for reward-modulated Hebb rule as

\[
\langle F_a u^a \rangle = \eta \langle (r - \beta \langle r \rangle) y^a u^a \rangle
\]

\[
= \eta p_a \{ \langle r | a \rangle - \beta \langle r \rangle \} \langle u^a | a \rangle,
\]

\[
\langle F_a^2 \rangle = \eta^2 \langle (r - \beta \langle r \rangle)^2 (y^a)^2 \rangle
\]

\[
= \eta^2 p_a \{ (1 - 2\beta \langle r \rangle) \langle r | a \rangle + (\beta \langle r \rangle)^2 \},
\]

\[
\langle F_a \tilde{x}_a \rangle = \eta \langle (r - \beta \langle r \rangle) y^a \tilde{x}_a \rangle
\]

\[
= \eta p_a \{ \langle r | a \rangle - \beta \langle r \rangle \} \langle \tilde{x}_a | a \rangle.
\]
Reward-Modulated Learning in Decision-Making Networks

and for the delta rule,
\[ \langle F_a u^a \rangle = \eta \langle (r - \beta \langle r \rangle) u^a \rangle \]
\[ = \eta p_a \langle \langle r | a \rangle - \beta \langle r \rangle \rangle \langle u^a | a \rangle + \eta (1 - p_a) \langle \langle r | a' \rangle - \beta \langle r \rangle \rangle \langle u^a | a' \rangle \]
\[ = \eta \{ p_a \langle \langle r | a \rangle \rangle - \beta \langle r \rangle \} \langle u^a | a \rangle \]
\[ + \eta (1 - p_a) \{ \langle \langle r | a' \rangle \rangle - \beta \langle r \rangle \} \langle u^a | a' \rangle \]
\[ = \eta \{ p_a \langle \langle r | a \rangle \rangle - \beta \langle r \rangle \} \langle u^a | a \rangle \]
\[ + \eta (1 - p_a) \{ \langle \langle r | a' \rangle \rangle - \beta \langle r \rangle \} \langle u^a | a' \rangle \]

where we have used the identities
\[ \langle u_a \rangle = \langle u_a | a \rangle p_a + \langle u_a | a' \rangle (1 - p_a), \]
\[ \langle \tilde{x}_a \rangle = \langle \tilde{x}_a | a \rangle p_a + \langle \tilde{x}_a | a' \rangle (1 - p_a). \]

Also, more specifically,
\[ \langle u^a | a \rangle = \bar{J}_a + \frac{l_a^2}{p_a \sqrt{2\pi L^2}} \exp \left( -\frac{D_{l_a}^2}{2L^2} \right), \quad (4.7) \]
\[ \langle \tilde{x}_a | a \rangle = 1 + \frac{l_a}{p_a \sqrt{2\pi L^2}} \exp \left( -\frac{D_{l_a}^2}{2L^2} \right), \quad (4.8) \]

where we have defined \( L = \sqrt{l_A^2 + l_B^2} \) and \( D_{l} = \bar{J}_B - \bar{J}_A \). The details on the derivation are in appendix A. Equations 4.7 and 4.8 have intuitive forms. The first terms on the right-hand sides are the averages marginalized over choice (\( \langle u^a \rangle = \bar{J}_a, \langle \tilde{x}_a \rangle = 1 \)). The second terms derive from the condition that alternative \( a \) is chosen. The expectation of \( u^a \) should shift by a positive value provided that \( a \) is chosen; thus, the second term of equation 4.7 is always positive unless \( l_a = 0 \) (i.e., there are no fluctuations in \( u^a \)). The expected value of \( \tilde{x}_a \) shifts by an amount with the same sign as \( \bar{J}_a \); if \( \bar{J}_a < 0 \); the larger the \( \tilde{x}_a \), the smaller the likelihood is that alternative \( a \) will be chosen.

4.2 Evolution Equation for Order Parameters with Weight Normalization. Thus far, we have not considered any constraints on the weight update. As we shall see later, without constraints, the synaptic weights diverge. Here, we consider weight normalization in which the total length of the weight vector is kept constant. This type of constraint has been
partially supported by a physiological experiment (Royer & Paré, 2003). Also, it has been used in a model that explains perceptual decision making, and it can be closely fitted to the behavioral data (Law & Gold, 2009). However, we adopted this weight normalization because of analytical convenience rather than taking biological realism into account. Other weight constraints would produce no clear differences in the following results. Specifically, we constrained the norm of the weight as 
\[ \|J\|_2^2 = 2N, \]
where \( J = (J^A_1, \ldots, J^A_N, J^B_1, \ldots, J^B_N) \). This is equivalent to keeping \( I^A_A + I^B_B = 2 \). This is achieved by modifying the learning rule in the following way (Biehl, 1994):

\[
J^a(t + 1) = \frac{\sqrt{2N}(F^a + F^a x^a)}{\sqrt{\|J^A(t) + F^a x^A\|^2 + \|J^B(t) + F^B x^B\|^2}} \sqrt{1 + \frac{1}{N}\{F^A u^A + F^B u^B + \frac{1}{2}(F^2_A + F^2_B)\}},
\]

provided that \( \|J\|^2 = 2N \) holds at trial \( t \). Expanding the right-hand side and retaining the terms of order up to \( \frac{1}{N} \),

\[
J^a(t + 1) = J^a(t) + F^a x^a - \frac{1}{2N} \mathcal{F} J^a(t),
\]

where \( \mathcal{F} = F^A u^A + F^B u^B + \frac{1}{2}(F^2_A + F^2_B) \). With this approximated learning rule, we can obtain the differential equation similarly to equations 4.5 and 4.6:

\[
\frac{dI^2_a}{d\alpha} = 2\langle F^a u^a\rangle + \langle F^2_a\rangle - \langle \mathcal{F} \rangle I^2_a,
\]

\[
\frac{d\tilde{J}_a}{d\alpha} = \langle F^a x^a\rangle - \frac{1}{2} \langle \mathcal{F} \rangle \tilde{J}_a.
\]

With \( \langle \mathcal{F} \rangle = \langle F^A u^A\rangle + \langle F^B u^B\rangle + \frac{1}{2}(\langle F^2_A \rangle + \langle F^2_B \rangle) \), we can find that \( \frac{dI^2_a}{d\alpha} + \frac{d\tilde{J}_a}{d\alpha} \) becomes zero when \( I^A_A + I^B_B = 2 \); thus, the length of the weight is kept constant.

Scaling the weight with a common constant does not affect the choice probability. To demonstrate this, we calculate the order parameter after the scaling, \( J^a \rightarrow J^a/c \). Then order parameters change as \( I_a = I^a/c \) and \( \tilde{J}_a = \tilde{J}_a/c \), and thus constant \( c \) does not appear in the argument of the complementary error function in equation 6.2. However, as scaling affects the behavior of the order parameter, it thereby affects the evolution of the choice probability as is described in the following (see section 6.3).
5 Macroscopic Description of Model Behavior: The Case with Small Fluctuations in Input

Let us consider the macroscopic behavior of the case with small fluctuations in input, where input obeys the gaussian distribution of mean $1/N$ and variance $1/N^2$. The definition of the order parameters is the same as that in the case with large fluctuations. The fluctuations in net inputs $u^a$ vanish for this case in the large $N$ limit. In order for the network to make stochastic choices, we add pooling noise $\xi^a$ to the output unit. Here, we assume $\xi^a$ is a gaussian random variable whose mean is zero and variance is $\sigma_p^2$.

For the case with large fluctuations in input, adding pooling noise only additively increases the variance in the potential of output unit and does not affect learning behavior. From a simple calculation, we find that $u^A$ and $u^B$ obey independent gaussian distributions whose means and variances are, respectively, given by $\bar{J}_a$ and $l_a^2/N + \sigma_p^2$, $(a = A, B)$. From this, the probability that the network will choose alternative $A$ can be described as

$$p_A = \frac{1}{2} \text{erfc} \left\{ -\frac{\bar{J}_A - \bar{J}_B}{\sqrt{2} \left( (l_A^2 + l_B^2)/N + 2\sigma_p^2 \right)} \right\}, \quad \text{(5.1)}$$

from which we find that in the large $N$ limit, the contribution of variance in the weight distribution to choice probability vanishes. If we write the evolution equations for the order parameters retaining terms up to order $1/N$:

$$\frac{dl_a^2}{d\alpha} = 2\langle F_a u^a \rangle + \frac{2}{N} \langle F_a^2 \rangle, \quad \text{(5.2)}$$

$$\frac{d\bar{J}_a}{d\alpha} = \langle F_a \tilde{x}_a \rangle. \quad \text{(5.3)}$$

In terms of the variance in weight,

$$\frac{d\sigma_a^2}{d\alpha} = 2(\langle F_a u^a \rangle - \langle F_a \tilde{x}_a \rangle \bar{J}_a) + \frac{2}{N} \langle F_a^2 \rangle. \quad \text{(5.4)}$$

Since $\langle u^a|a' \rangle = \langle \tilde{x}_a|a' \rangle \bar{J}_a$ for large $N$, the first term of equation 5.4 vanishes. The second term also vanishes in the large $N$ limit. Thus, the learning dynamics is described by only equation 5.3. The model can be reduced to the network with $N = 1$ and each input unit obeys mean 1 and variance $1/N$. This reduced model corresponds to the one analyzed by Loewensetein (2008) where the small variance limit of input neurons was considered. Matching behavior is attained for this case as an equilibrium point for
covariance-based learning. In the following, we report only the results for the case with large fluctuations in input.

6 Results for the Case of Large Fluctuations in Input

6.1 Evolution of Choice Probability and Order Parameters. We first examined the accuracy of our macroscopic descriptions by comparing the theoretical predictions with computer simulations. We used a competitive foraging task with the same parameters as those in Figure 2. In the following numerical simulations, the probability of choosing alternative $A$, $p_A$, was calculated using equation 6.2 with numerically calculated values for the order parameters, $\bar{J}_a$, $\bar{l}_a$. The average reward $\bar{r}$ for the reward-baseline was computed using equation 3.7 with this $p_A$; we did not use the running average as stated in section 3.2 to avoid the need for tuning parameter $\gamma$ and to eliminate the effect of estimation error. Figure 4 plots the evolution of choice probability and order parameters in two learning rules without a weight normalization constraint. Simulations were done for $N = 1000$. The lines represent the results for theory, and the symbols plot the results for simulations. The results for theory agree well with those for the computer simulations, indicating the validity of our theory. We can see that the choice probability approaches a value that yields matching behavior ($p_A^{\text{match}} = 0.6928$), while the order parameters $\bar{J}_a$ and $\sigma_a$ continue to change without becoming saturated. The weight standard deviation, $\sigma_a$, always increases (the weight diffusion). The speed of diffusion is slightly faster for the delta rule than for the reward-modulated Hebb rule.

Figure 5 plots the results with weight normalization. Again, the results for theory agree well with those for computer simulations. For the reward-modulated Hebb rule, the choice probability saturates at a value below $p_A^{\text{match}}$. For the delta rule, the choice probability first approaches $p_A^{\text{match}}$ but without reaching $p_A^{\text{match}}$. It then returns to the uniform choice probability ($p_A = 0.5$) due to its larger diffusion effect than that of the reward-modulated Hebb rule. In what follows, we systematically examine the properties of our model. Since computer simulation with large $N$ is time-consuming, we report only the theoretical results of a macroscopic description.

6.2 Matching Behavior Is Not Necessarily a Steady State of Learning. From Figure 4, the choice probability seems to asymptotically approach matching behavior. However, matching behavior is not necessarily a steady state or a fixed point of learning in the case of large fluctuations in input. In Figure 6, the order parameters are initialized so that $p_A(0) = p_A^{\text{match}}$ and then equations 4.5 and 4.6 are numerically solved. We see that $p_A$ does not remain at $p_A^{\text{match}}$ but changes toward the uniform choice ($p_A = 0.5$) for both learning rules. We refer to this as the repulsive property of matching
Figure 4: Evolution of choice probability (left panels) and order parameters (right panels) for reward-modulated Hebb rules (A and B) and delta rule (C and D), without weight normalization. Parameters are $\beta = 1$ and $\eta = 0.1$, and task schedules are the same as those in Figure 2. Lines represent results of theory, and symbols plot the mean of 10 trials with computer simulation. Simulations were done for $N = 1000$. Error bars indicate standard deviation (s.d.). Error bars are almost invisible for choice probability since s.d. is very small. Initial conditions at $t = \alpha = 0$ were set at $f^a_i = 5.0$ for all $a, i$, and thus $\bar{J}^a = 5.0, \sigma_a = 0.0$, and $p_A = 0.5$.

behavior.\(^2\) Then, for the reward-modulated Hebb rule, $p_A$ evolves toward $p_A^{\text{match}}$, but does not do so for the delta rule. To understand the mechanism for this repulsive property of matching behavior, let us substitute $\beta = 1$ and the condition of the matching law, $\langle r | a \rangle = \langle r \rangle, \ a = A, B$ into equations 4.5 and 4.6, for the case without weight normalization. We then find that $\langle F_a^0 \rangle$ and $\langle F_a^2 \rangle$ are zero, but $\langle F_a^2 \rangle$ is nonzero and positive except for the noninteresting case where $r$ always takes the same value (0 or 1). Therefore, when $p_A = p_A^{\text{match}}$, the variance in the weight increases: $d\sigma_a^2 / d\alpha = d((f^2_i - \bar{J}^2_a))/d\alpha > 0$. This moves the choice probabilities toward even choice behavior, $p_A = 0.5$ (see equation 6.2). This is the reason that $p_A^{\text{match}}$ is repulsive. This result is

\(^2\)It should be noticed that the matching behavior is not a fixed point, as we shall explain later, although the term repulsive is often used for describing the property of a fixed point.
Figure 5: Same plots as in Figure 4 but with weight normalization. Initial conditions were set at $J_a = 1.0$ for all $a, i$ and thus $\overline{J}_a = 1.0$, $\sigma_a = 0.0$, and $p_A = 0.5$.

Figure 6: Perfect matching is not equilibrium point. We set the initial value of order parameters to derive perfect matching for the (A) no-normalization condition and (B) normalization condition. In both cases, choice probability that yields perfect matching is repulsive. Parameters were set to $\beta = 1$ and $\eta = 1$. For the no-normalization condition, initial conditions were first set at $\overline{J}_B = 1.0$, $\sigma_A = \sigma_B = 1.0$ and then $\overline{J}_A$ was determined so that $p_A = p_A^{\text{match}}$. In this case, $\overline{J}_A = 2.5545$. For the normalization condition, these values were rescaled so that normalization condition $\sqrt{\overline{l}_A^2 + \overline{l}_B^2} = 2$ was met.
in contrast with the \( N = 1 \) case where the average changes stop when \( p_A \) converges to \( p_A^{\text{match}} \). The results for \( N = 1 \) are given in appendix B (also see Loewenstein, 2008). With \( N = 1 \), the diffusion effect cannot appear by definition.

With weight normalization, \( \sqrt{2(l_A^2 + l_B^2)} \) in equation 4.2 is always two; thus, the only factor that determines choice probability is the difference between \( \bar{J}_A \) and \( \bar{J}_B \). Substituting \( \langle r|a \rangle = \langle r \rangle, \forall a \) into equation 4.12, only term \( \langle F^2_a \rangle \) remains, and we obtain

\[
\frac{d(\bar{J}_B - \bar{J}_A)}{da} = -\frac{1}{2}(\langle F^2_A \rangle + \langle F^2_B \rangle)(\bar{J}_B - \bar{J}_A).
\] (6.1)

Except for uninteresting cases where \( r \) is always 0 or 1, \( \langle F^2_A \rangle + \langle F^2_B \rangle > 0 \) holds; thus, the absolute difference, \( |\bar{J}_B - \bar{J}_A| \), always decreases. Here, again, the choice probability at \( p_A^{\text{match}} \) approaches unbiased choice behavior due to the diffusion effect. The reason that the diffusion effect works in weight normalization may be intuitively understood by decomposing the learning process into a weight update process and a weight normalization process. In the weight update process, the weight variance increases due to diffusion, as in the case with no normalization. In the subsequent normalization process, the increases in synaptic variance may be increased or reduced so that the normalization condition holds. Assume that the normalization constant is \( c \), that is, \( l_i^a \rightarrow l_i^a/c \). Then the total synaptic weight variance becomes \( \sigma_A^2 + \sigma_B^2 \rightarrow (\sigma_A^2 + \sigma_B^2)/c^2 \) after normalization. If \( c > 1 \), the increases in weight variance in the weight update process decrease. In this process, however, the weight mean differences are simultaneously reduced: \( \bar{J}_A - \bar{J}_B \rightarrow (\bar{J}_A - \bar{J}_B)/c \). The normalization constant, \( c \), is cancelled out in the argument of the complementary error function in equation 6.2, as stated in section 4.2. Thus, the same diffusion effects on choice probability remain after the weight normalization process.

Nevertheless, the choice probability of the reward-modulated Hebb rule without weight normalization asymptotically converges to \( p_A^{\text{match}} \). The reason for this can be explained as follows. First, we rewrite the choice probability as

\[
p_A = \frac{1}{2}\text{erfc} \left\{ -\frac{\bar{J}_A - \bar{J}_B}{\sqrt{2(\bar{J}_A^2 + \bar{J}_B^2 + \sigma_A^2 + \sigma_B^2)}} \right\}.
\] (6.2)

From this expression, we find that the larger the magnitude of \( \bar{J}_a \) is, the weaker the effect of increases in \( \sigma_a \). The diffusion term, \( \langle F^2_a \rangle \), which moves \( p_A \) away from \( p_A^{\text{match}} \), depends on \( p_A \) but not on the magnitude of \( \bar{J}_a \)’s. Thus, within the order parameter set satisfying \( p_A = p_A^{\text{match}} \), the larger the magnitudes of \( \bar{J}_a \)’s are, the weaker is the repulsive effect. If \( |\bar{J}_B - \bar{J}_A| \rightarrow \infty \)
Figure 7: Dependence of learning on initial values of order parameters. Reward-modulated Hebb rule (A) without normalization and (C) with normalization. Delta rule (B) without normalization and (D) with normalization. Parameters for model and task schedules are same as those in Figure 4.

while $\sigma_A$, $\sigma_B$ are finite, $p_A$ can stay at $p_A^{\text{match}}$. Because $|\bar{J}_B - \bar{J}_A|$ can increase faster than $\sigma_A$ and $\sigma_B$ in the reward-modulated Hebb rule without any weight constraints, the network approaches such situations. This is the reason that in Figure 6A, the $p_A$ returned to $p_A^{\text{match}}$ after it was repulsed from $p_A^{\text{match}}$. When weight normalization is imposed, the magnitude of $\bar{J}_a$'s is limited as $|\bar{J}_B - \bar{J}_A| < 2$. Thus, the diffusion effect prevents $p_A$ from approaching $p_A^{\text{match}}$. In the delta rule, the magnitude of $\bar{J}_a$'s cannot increase independent of $\sigma_a$'s. Thus, $p_A$ saturates before it reaches $p_A^{\text{match}}$, where the increase in $|\bar{J}_B - \bar{J}_A|$ and those in $\sigma_a$'s are balanced. Such balanced points depend on the initial balance of $J_a$'s and $p_a$'s, as we will see in the next section.

6.3 Initial Value Dependence of Learning Behavior. Here, we explain how the initial values of order parameters affect the time evolution of choice probability $p_A$. As we stated in section 4.2, choice probability is ambiguous in terms of scaling; there is freedom in that the scaling transformation of weight (say, $J^i_a \rightarrow J^i_a/c$) does not change the choice probability. Even if learning starts from the same $p_A$, because of this ambiguity the trajectory of $p_A$ can differ depending on the initial value. Figure 7 plots the evolution of choice probability from various different initializations of order parameters. In Figures 7A and 7B, all of the initial values of $p_A$ are 0.5, but the values of order parameters $\bar{J}_a$ and $\sigma_a$ differ. For the reward-modulated Hebb rule, the $p_A$ trajectories differ for different initializations, but all the trajectories approach the same point. In contrast, for the delta rule, the asymptotic
values clearly depend on the initial value of the order parameters. As the initial value of $\bar{J}_a$ is larger than that of $\sigma_a$, the asymptotic values tend to approach $p_A^{\text{match}}$. As we stated in the previous section, $p_A$ saturates where the increases in $|\bar{J}_B - \bar{J}_A|$ and in $\sigma_a$’s are balanced in the delta rule. The delta rule for the weight normalization condition converges to one value regardless of initialization or the reward-modulated Hebb rule (see Figures 7C and 7D). However, this is because the diffusion effect is dominant in the delta rule so that it equalizes choice probabilities. In summary, the input values of order parameters affect the asymptotic behavior of the model so that the larger the variances in initial weight distributions are, the closer the asymptotic behavior is to unbiased behavior only in the delta rule with no normalization. The results presented in this letter should be considered while keeping this in mind.

6.4 Learning Rate Dependence of Learning Behavior. Next, we investigate how the learning rate, $\eta$, affects the learning performance. In the diffusion term, $\langle F_a^2 \rangle$, learning rate $\eta$ appears in the quadratic term of $\eta$. In contrast, only the first-order terms of $\eta$ appear in the other terms. Therefore, if $\eta$ is small, the repulsive effect from matching behavior due to the diffusion effect is expected to weaken. Figure 8 plots the dependence of the evolution of $p_A$ on $\eta$. As a whole, as $\eta$ is decreased, the asymptotic value, $p_A$, approaches matching behavior, but relaxation slows due to the diffusion of synaptic weights. As we previously discussed, the diffusion effect is more evident for the delta rule than for the reward-modulated Hebb rule, and for the case with the weight normalization condition than that with the nonnormalization condition. This tendency becomes evident as $\eta$ increases.

For the reward-modulated Hebb rule without normalization, networks approach matching behavior even for a very large learning rate ($\eta = 1000$). At the beginning of learning when $\bar{J}_a$ is of small magnitude, the diffusion term, $\langle F_a^2 \rangle$, has a large impact so that it greatly impedes learning for a large $\eta$ case. However, as the magnitude of the differences $\bar{J}_A - \bar{J}_B$ increases, this effect weakens and the dependence of $p_A$ on $\eta$ becomes quite small. Although there is still a deviation from perfect matching (see the inset of Figure 8A), the asymptotic value is almost unaffected in the reward-modulated Hebb rule. For the delta rule without normalization, the asymptotic values gradually depend on $\eta$. With normalization constraints, the reward-modulated Hebb rule also demonstrates graded dependence of asymptotic probability on $\eta$. These results reflect the fact that the greater learning rate $\eta$ is, the larger the diffusion effect.

6.5 Effect of Mistuning Reward Baseline. We have thus far considered the case where the baseline for the reward in the learning rules matches the expected reward, $\langle r \rangle$; thus, the learning rule belongs to the pure covariance rule of Loewenstein and Seung (2006; see equation 3.2). Here, we
Figure 8: Evolution of choice probability for various learning rates, $\eta$. The panels in the top row are for the nonweight normalization condition and those in the second row are for normalization condition. Columns at the left are for the reward-modulated Hebb rule and those at the right are for the delta rule. Parameters for model and task schedules are the same as those in Figure 4. Initial conditions were set at $\sigma_a = 0.0$, $(a = A, B)$, $\bar{J}_a = 10.0$ for the nonnormalization condition and $\bar{J}_a = 1.0$ for the normalization condition.

Figure 9 plots the dependence of asymptotic values $p_A(\infty)$ on the mistuning parameter, $\beta$. We obtained these by calculating the differential equation numerically up to $\alpha = 10^7$ for the case without normalization and $\alpha = 10^5$ for the case with normalization. We compared the mistuning effect on various task parameters, $\lambda_B = 0.32$, 0.35 (the same condition as with the previous results), and 0.40, while the other parameters were the same as those in Figure 2. We can see that as $\beta$ deviates from one, asymptotic behavior departs from matching behavior, but there is asymmetry in this effect: the deviation from matching is larger for $\beta > 1$ than for $\beta < 1$. For $\beta < 1$, the reward-modulated Hebb rule exhibits undermatching ($p_A < p_A^{\text{match}}$) and overmatching ($p_A > p_A^{\text{match}}$), depending on the task parameters. When $p_A^{\text{match}}$ is large (see Figures 9A and
Figure 9: Effect of mistuning reward baseline from actual expected reward $\langle r \rangle$. We compared two learning rules on various task parameters, $\lambda_B = 0.32, 0.35$ (same condition as with previous results), and 0.40, while other parameters were the same as those in Figure 2 ($\lambda_A = 0.4$, $\mu_A = 0.1$, and $\mu_B = 0.4$). For each task condition, matching choice probability, $p_A^{\text{match}}$, was 0.7524, 0.6928, and 0.6000. Panels are for (left) nonweight normalization condition and (right) weight normalization condition. Horizontal dotted lines indicate $p_A^{\text{match}}$ and even choice probability ($p_A = 0.5$). Learning rate is set at $\eta = 0.1$. Initial values were set at $\sigma_a = 0.0, \bar{J}_a = 1.0$ for all conditions.

9B), the learning rule demonstrates overmatching, and when $p_A^{\text{match}}$ is small (see Figures 9E and 9F), the rule demonstrates undermatching. For $\beta > 1$, undermatching is observed under all conditions.

The direction of bias is even flipped ($p_A < p_B$) for the delta rule. Let us consider the reason for this flip. The direction of bias is determined
by the sign of the differences between $\bar{J}_A$ and $\bar{J}_B$ (if $\bar{J}_A \geq \bar{J}_B$, then $p_A \geq p_B$; otherwise $p_B < p_A$). To make the flip occur, $\frac{d\bar{J}_A}{da} - \frac{d\bar{J}_B}{da} < 0$ should hold until it does occur. For the delta rule,

$$\frac{d\bar{J}_A}{da} - \frac{d\bar{J}_B}{da} = \eta(\langle r\mid A \rangle - \langle r\mid B \rangle)(p_A(\bar{x}_A\mid A) + p_B(\bar{x}_B\mid B) - 1).$$

In the situation in our experiment, $\langle r\mid A \rangle > \langle r\mid B \rangle$ for $p_A < p_A^{\text{match}}$ (see Figure 2). Thus, to make $\frac{d\bar{J}_A}{da} - \frac{d\bar{J}_B}{da}$ negative, it should satisfy $p_A(\bar{x}_A\mid A) + p_B(\bar{x}_B\mid B) < 1$. For this, $\bar{J}_A$ and/or $\bar{J}_B$ should be negative (see equation 4.8). This is interpreted intuitively as follows. For the delta rule, if $\bar{J}_A > 0$ holds, the larger the $\bar{x}_A$, the more likely the network is to choose the alternative, $A$. The delta rule utilizes this positive correlation. For this case, if $\bar{J}_A > 0$, $\bar{J}_B > 0$, and $\langle r\mid A \rangle > \langle r\mid B \rangle$ holds, the difference, $\bar{J}_A - \bar{J}_B$, increases on average and thus $p_A$ increases. However, if $\bar{J}_A$ and $\bar{J}_B$ are negative, a negative correlation arises: the larger the $\bar{x}_A$, the less likely the network is to choose the alternative, $A$. For this case, if $\langle r\mid A \rangle - \beta \langle r \rangle > 0$, $\bar{J}_A$ decreases on average. In addition, if $\langle r\mid A \rangle > \langle r\mid B \rangle$, the difference, $\bar{J}_A - \bar{J}_B$, decreases on average and $p_A$ decreases. In the case of $\beta > 1$, the expected returns, $\langle r\mid A \rangle$ and $\langle r\mid B \rangle$, are less than the baseline, $\beta \langle r \rangle$, around $p_A^{\text{match}}$; thus, the mean weights of both groups tend to be negative and $p_A$ cannot stay at the point near $p_A^{\text{match}}$.

### 6.6 Accelerating Learning Speed

The goal of this study was to provide the quantitative insights from a simple model rather than to develop a realistic model that could quantitatively fit an animal’s behavioral data. However, the relation between the model and an actual animal’s learning is worth discussing.

In our model, the update of synaptic efficacy for one trial is $O(1/N)$, and to make $O(1)$ change in the probability of choice $p_A$, $O(N)$ trials are needed. If there are many sensory neurons, this would be too slow compared to behavioral experiments done on an animal, which require only tens to hundreds of trials to convergence. For example, in the simulation described in Figure 5, learning converges after approximately 2000 time units, where each time unit is composed of $N$ trials. Since $N = 1000$ in the simulation, this means that it requires $2 \times 10^5$ trials. Here, let us consider how the learning speed can be accelerated so that it is comparable to an animal’s learning. If we use large learning rate $\eta$, that is, $O(N)$, to make each update $O(1)$, self-averaging properties do not hold, and our theory cannot be applied. Moreover, as we saw in Figure 8, increasing $\eta$ does not necessarily improve learning speed due to the diffusion effect. Here, we will explain a way that accelerates the learning speed within our original scaling, which holds self-averaging properties. To achieve this, we modify the input statistics so that the mean is $X_0/N$ and the variance is $1/N$, where $X_0$ is a constant of $O(1)$. Our original model corresponds to $X_0 = 1$. If we regard a positive departure
of individual sensory input from zero as a signal, the signal-to-noise ratio (SNR) is \((X_0/N)/(1/\sqrt{N}) = X_0/\sqrt{N}\). Thus, \(X_0\) is a parameter that controls the SNR of sensory input. We here report that the higher the SNR, the faster the learning.

For this case, the probability that the network will choose alternative \(A\) is written as

\[
p_A = \frac{1}{2} \text{erfc} \left\{ - \frac{X_0(\bar{J}_A - \bar{J}_B)}{\sqrt{2(l_A^2 + l_B^2)}} \right\}.
\]

(6.3)

The evolution equations for the order parameters do not change, except for the fact that the conditional averages are now

\[
\langle u^a | a \rangle = \bar{J}_a X_0 + \frac{l_a^2}{p_a \sqrt{2\pi L^2}} \exp \left( - \frac{X_0^2 D_j^2}{2L^2} \right),
\]

(6.4)

\[
\langle \tilde{x}_a | a \rangle = X_0 + \frac{\bar{J}_a}{p_a \sqrt{2\pi L^2}} \exp \left( - \frac{X_0^2 D_j^2}{2L^2} \right).
\]

(6.5)

Increasing \(X_0\) accelerates the learning speed due to two reasons. First, as we see from equation 6.3, large \(X_0\) amplifies the effect of the differences, \(\bar{J}_A - \bar{J}_B\). For example, if \(X_0 = 2\), only a half difference between \(\bar{J}_A\) and \(\bar{J}_B\) is needed to attain the same choice probability with our original model \((X_0 = 1)\). Second, signal terms like \(\langle F_a u^a \rangle\) and \(\langle F_a \tilde{x}_a \rangle\) are roughly proportional to \(X_0\) through \(\langle u^a | a \rangle\) and \(\langle \tilde{x}_a | a \rangle\) (see equations 6.4 and 6.5). Thus, the larger the \(X_0\) is, the faster the changes in order parameters. It should be noted that the value of \(X_0\) does not affect the diffusion term, \(\langle F^2_a \rangle\). Therefore, the strength of the diffusion effect is kept the same as that with the original model while the learning speed can be increased.

Figure 10 plots the results for \(X_0 = 20\) and various learning rates, \(\eta\). The other conditions are the same as those in Figure 8. Compared with Figure 8, we can see that the learning speeds were drastically improved, so that for large \(\eta\), \(p_A\) almost converged to near matching behavior within 0.1 time unit. This means that if \(N = 1000\), it takes only about 100 trials for learning. This speed is comparable to that in the behavioral experiments. We should note that for the high SNR case, the property where higher \(\eta\) and weight constraints lead to undermatching behavior due to the diffusion effect still holds (see Figure 10B).

7 Discussion

In this study, we analyze the reward-based learning procedure in simple, large-scale decision-making networks. If only a single plastic synapse is
Figure 10: Learning performance for high signal-to-noise ratio input case ($X_0 = 20$) for various learning rates, $\eta$. All conditions except for input statistics were the same as those in Figure 8.

taken into consideration, covariance learning rules always seem to make matching behavior a steady state of learning. However, we demonstrated that when the fluctuations in inputs from individual sensory neurons are so large that they influence the net input to value-encoding neurons, matching behavior is no longer a steady state. This is because the randomness in weight modifications affects the choice probability of the network, and the effect returns to the learning process as feedback. These results may offer suggestions for discussing learning behavior in large-scale neural circuits. Here, we discuss the implications of our findings.

7.1 Mechanism for Undermatching. Choices by animals in many experiments deviate slightly from matching behavior toward unbiased choice behavior, a phenomenon called undermatching (Baum, 1974; Sugrue et al., 2004; Corrado et al., 2005; Lau & Glimcher, 2005). There are several possible explanations for this phenomenon.

The learning rule employed by Soltani and Wang (2006) is equivalent to the state-less Q-learning in the literature on reinforcement learning (Sutton & Barto, 1998; Soltani et al., 2006). Q-learning distributes its choices according to the expected returns from individual options, and not according to income. Sakai and Fukai (2008a, 2008b) proved that Q-learning does not lead to matching behavior. Thus, Soltani-Wang’s model is intrinsically incapable of reproducing matching behavior. Soltani and Wang interpreted
that the departure from matching behavior due to limitations in the learning rule was a possible mechanism for undermatching. Indeed, in a specific reinforcement schedule, they found that their model could demonstrate undermatching that was comparable to that in experimental observations. However, we need to examine whether their learning rule can reproduce learning by animals in a broader range of tasks.

Loewenstein (2008) suggested that the mistuning of parameters in the covariance learning rule could cause undermatching, as we also confirmed in a large-scale network (see Figures 9A and 9B). If the plasticity mechanism, which is similar to the covariance rule, works in an actual brain, a plausible scenario is that deviation from true covariance will cause undermatching. However, it should be noted that in some task settings, deviation can cause overmatching rather than undermatching (see Figures 9E and 9F).

Our findings add one possible mechanism for undermatching: undermatching can be caused by the diffusion of synaptic efficacies, which leads to equalization in choice behavior. The effect of diffusion increases for high learning rates and with the weight normalization condition. It has been observed that learning is rapid in animal experiments so that they can adapt to dynamic environments (Sugrue et al., 2004). In addition, there may be normalization effects on synaptic efficacies (Royer & Paré, 2003). Thus, it is likely that the diffusion effect will appear as undermatching.

7.2 The Reward-Modulated Hebb Rule Is More Robust Against Parameter Changes Than the Delta Rule. We compared two learning rules that belong to the covariance learning rule: the reward-modulated Hebb and delta rules. The main differences in these rules is that the weights are updated in the former for only the neuron pool that wins the competition, but all the weights are updated in the latter. Thus, the total effect of diffusion is stronger for the delta rule than for the reward-modulated Hebb rule. As we have demonstrated, these differences cause quantitative differences between the two rules. First, with weight normalization, the delta rule always converges to unbiased choice behavior due to a strong diffusion effect, while the reward-modulated Hebb rule can reach near-matching behavior (see Figure 5). Second, without normalization, the learning rate does not affect the asymptotic behavior of the reward-modulated Hebb rule but it does affect the delta rule. For higher learning rates, the asymptotic choice probability is closer to unbiased choice in the delta rule (see Figure 8B). Third, the asymptotic choice probability depends on the initial value of order parameters in the delta rule but not in the reward-modulated Hebb rule (see Figure 7). In the delta rule, the larger the initial variance of weight is, the closer to unbiased choice the asymptotic probability. In the reward-modulated Hebb rule, the effect of initial weight variance is weakened as learning progresses, but in the delta rule, learning cannot diminish the effect of initial variance due to its diffusion effects that are larger than those in the reward-modulated Hebb rule.
Taken together, our results imply that the reward-modulated Hebb rule is appropriate and plausible for learning in decision making, at least compared to the delta rule. We should note that this conclusion was reached by taking into consideration the large-size limit of the decision-making network; if we consider only a steady state for a single plastic synapse, both learning rules seem to yield the same results.

7.3 Possible Neural Implementation of a Case with Large Fluctuations in Input. In the case with large fluctuations in input, the ratio of the standard deviation of input to its mean is \( (1/\sqrt{N})/(1/N) = \sqrt{N} \), and it thus diverges for large \( N \). It may be biologically unrealistic to assume that a single sensory neuron can achieve this ratio. The ratio of the variance of spike count to its mean has been observed to be larger than one but of order one (Dean, 1981; Softky & Koch, 1993). However, if we regard each input unit \( (x_i^a, i = 1, \ldots, N, a = A, B) \) to be an effective input to a pool of value-encoding neurons from an excitatory neuron and a pool of inhibitory interneurons, we can consider the neural substrates underlying our scaling. To demonstrate this, we first need to note that the large fluctuation case can be transformed into a mathematically equivalent form by rescaling \( J_i^a \rightarrow J_i^a/\sqrt{N}, \eta \rightarrow \eta/N \), and assuming individual inputs obey a gaussian distribution with mean \( 1/\sqrt{N} \) and standard deviation one. We call this the equivalent model. Then we need to seek a possible neural implementation of the scaling (for \( N \)) for the equivalent model. Let us consider a sensory neuron whose activity is denoted by \( x_i^E \) and an inhibitory neuron population whose input to a value-encoding neurons pool is \( h_I \). Here, we have omitted the index of action, \( a \). Assume that \( x_i^E \) obey a gaussian distribution with \( \mu_E \), and variance \( \sigma_E^2 \) (both are of order \( O(1) \); thus, the ratio of the standard deviation and mean is of order \( O(1) \)).

Let us consider the situation where the total excitatory inputs and an inhibitory input to the value-encoding neuron pool are roughly balanced. Denote the total input from excitatory input as \( h_E = \sum_i N J_i x_i^E \). Assume that the total synaptic input from an inhibitory neuron pool \( h_I \) obeys a gaussian distribution whose mean is slightly smaller than the mean of the total input from sensory neurons, \( \langle h_E \rangle = N\bar{J}\mu_E \sim O(\sqrt{N}) \) (where \( \bar{J} = \sum_i J_i/N \sim O(1/\sqrt{N}) \)). Then the mean of the effective input to the postsynaptic neuron is suppressed to nearly zero, while the variances are maintained in the same order—for example, if \( h_I \) obeys a gaussian distribution with mean \( \mu_I \) and variance \( \sigma^2_I \), then the effective inputs can be transformed to the model with \( x_i \sim \mathcal{N}(\mu_E/\sqrt{N}, 1) \), which have the same order as those in the equivalent model with pooling noise \( \xi = \mathcal{N}(0, \sigma^2_I) \) added to \( \mu^a \). As stated in section 5, adding pooling noise does not provide qualitative differences. Furthermore, the learning for the balanced excitatory and inhibitory model is converted to that of the original model. Assume that excitatory input \( x_i^E \) is used with a shift such that \( \Delta J_i^a \sim F_a(x_i^E - (1 - 1/\sqrt{N})\mu_E) \).
Subtracting a constant value from input activity $x^E_i$ does not make the learning rule deviate from the class of the covariance learning rule of Loewenstein and Seung (2006) as long as $\beta = 1$. This balanced excitatory and inhibitory model was simulated in Katahira et al. (2010). In this way, the properties of learning observed in the case with large fluctuations in input would be preserved in the inhibitory and excitatory balanced network. It seems unrealistic that the mean input from the inhibitory neuron pool can be as finely tuned as $\mu_I = (1 - \frac{1}{\sqrt{N}}) \langle h^E_e \rangle$. However, such a balance of excitatory-inhibitory input is achieved by a recurrent network consisting of an excitatory neuron pool and an inhibitory neuron pool without fine-tuning of the synaptic weights (van Vreeswijk & Sompolinsky, 1996, 1998). This scenario is illustrated in appendix E with simulations of recurrent spiking neuron networks. Also, such a balanced state can be established through the spike-timing-dependent plasticity (STDP) rule if the integral of the synaptic modification function is negative (Song, Miller, & Abbott, 2000). As we did not explicitly model the balancing mechanisms, future work might be to examine such possibilities.

### 7.4 Synaptic Weight Constraints

It has been common to impose various constraints on synaptic efficacies in many studies on models such that the efficacies are bound to a specific range and take only finite discrete values (Fusi, 2002; Soltani & Wang, 2006; Fusi, Drew, & Abbott, 2005). Such constraints are absent in our model, and we took into account only weight normalization. However, as long as efficient synaptic efficacies (perhaps via multiple synapses) take multiple values with these constraints on synaptic efficacies, the diffusion effect can appear and our qualitative results would still hold.

We found an asymmetry effect of deviation of the baseline from the expected reward. For the delta rule, the bias of choice probability is even flipped. However, this flip is due to the biologically implausible consequence that the sign of synaptic efficacies flips to negative for $\beta > 1$. If we restrict the weight to positive for delta rules, the choice probability would stop at unbiased choice behavior ($p_A = 0.5$). Even with such restrictions, the asymmetry property itself would hold.

### 7.5 Alternative Model Architectures

The network architecture of our model (see Figure 1) may be too simplistic in two respects. First, we assumed that the value-encoding output neurons received inputs from two distinct populations of sensory neurons. This might be the case when the sensory input for each alternative is completely different, for example, in position and in color, such as those in Sugrue et al.’s experiment (Sugrue et al., 2004). However, these output neurons may share inputs from sensory neurons in general cases. Second, the output consisted of only one unit...
for each action. More realistic models include large numbers of neurons in competing neuron pools that correspond to our output units (Wang, 2002).

To address the first issue, we analyzed an extreme case that was opposite to our original model: the model shared all the sensory input neurons. We refer to this model as the common input model. The results we obtained from the analysis are in appendix C. The behavior of a more realistic case in which portions of inputs are shared can be interpolated with the results from the original separate-input and the common-input models. We obtained qualitatively the same results for the common-input model as those with our original model for the reward-modulated Hebb rule. However, the delta rule did not work appropriately because $J_{i}^{A}$ and $J_{i}^{B}$ followed exactly the same update.

To address the second issue, we considered a multiple-output model in appendix D. In this case as well, the qualitative results for our two-output-unit case were expected to hold.

8 Conclusion

In this study, we have analyzed the properties of reward-based synaptic plasticity rules within the context of choice behaviors. To achieve this, we employed techniques from statistical mechanics. Although statistical mechanical analysis has been successively applied to analyze the dynamics of learning in neural networks, we applied it to reward-modulated learning in decision-making networks for the first time, to the best of our knowledge.

Our main contribution was finding the diffusion effect in reward-modulated learning in large-scale decision-making networks. This effect causes qualitative differences in learning consequences between a model with a few parameters and large-scale networks. First, it no longer makes matching behavior a steady state of the covariance learning rule. Second, it causes undermatching, a phenomenon observed in many behavioral experiments. Third, the reward-modulated Hebb rule is more appropriate for learning matching behavior than the delta rule when the effect exists.

To make the theoretical analysis easier, we did not incorporate several realistic constraints on synaptic learning, such as, a discrete state for synapses or bounds on weight values. Nevertheless, our study should provide some insights into reward-based learning in decision-making networks. In future studies, we intend to incorporate biologically realistic features, including network architectures and plasticity rules.

Appendix A: Calculation of Ensemble Averages

Here, we present the derivation of the ensemble averages that appeared in equations 4.7 and 4.8.
A.1 Derivation of \( \langle u^a|a \rangle \). By definition,

\[
p_A(u^A|A) = p_A \frac{\int_{-\infty}^{\infty} du^B p(u^B) \int_{u^B}^{\infty} u^A du^A p(u^A)}{\int_{-\infty}^{\infty} du^B p(u^B) \int_{u^B}^{\infty} u^A du^A p(u^A)}
\]

\[
= \int_{-\infty}^{\infty} du^B p(u^B) \int_{u^B}^{\infty} u^A du^A p(u^A)
\]

\[
= \int_{-\infty}^{\infty} du^B p(u^B) \left[ \frac{1}{\sqrt{2\pi l_A^2}} \int_{u^B}^{\infty} u^A du^A \exp \left( -\frac{(u^A - \bar{J}_A)^2}{2l_A^2} \right) \right].
\]

(A.1)

Replacing the variable as \( z = \frac{(u^A - \bar{J}_A)}{l_A} \), and hence \( u^A = l_A z + \bar{J}_A \), \( du^A = l_A dz \), we calculate the second integral as

\[
\frac{1}{\sqrt{2\pi l_A}} \int_{u^B}^{\infty} u^A du^A \exp \left( -\frac{(u^A - \bar{J}_A)^2}{2l_A^2} \right)
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{\frac{u^B - \bar{J}_A}{l_A}}^{\infty} dz l_A z \exp \left( -\frac{z^2}{2} \right) + \frac{\bar{J}_A}{\sqrt{2\pi}} \int_{\frac{u^B - \bar{J}_A}{l_A}}^{\infty} dz \exp \left( -\frac{z^2}{2} \right).
\]

(A.2)

The first term in equation A.2 is

\[
\frac{1}{\sqrt{2\pi}} \int_{\frac{u^B - \bar{J}_A}{l_A}}^{\infty} dz l_A z \exp \left( -\frac{z^2}{2} \right) = \frac{l_A}{\sqrt{2\pi}} \exp \left( -\frac{(u^B - \bar{J}_B)^2}{2l_B^2} \right).
\]

(A.3)

Multiplying this by \( \int_{-\infty}^{\infty} du^B p(u^B) \) leads to

\[
\frac{l_A}{\sqrt{2\pi (l_A^2 + l_B^2)}} \exp \left( -\frac{1}{2(l_A^2 + l_B^2)} \left( J_A - J_B \right)^2 \right).
\]

(A.4)

Multiplying the second term in equation A.2 by \( \int_{-\infty}^{\infty} du^B p(u^B) \) leads to

\[
\frac{\bar{J}_A}{2} \text{erfc} \left( \frac{-J_A - J_B}{\sqrt{2(l_A^2 + l_B^2)}} \right) = \bar{J}_A p_A.
\]

(A.5)
Taken together, we obtain

\[ p_A(u^A|A) = \frac{l_A^2}{\sqrt{2\pi(l_A^2 + l_B^2)}} \exp\left(-\frac{(\bar{J}_A - \bar{J}_B)^2}{2(l_A^2 + l_B^2)}\right) + p_A\bar{J}_A, \]  \hspace{1cm} (A.6)

from which we obtain equation 4.7.

**A.2 Deviation of \( \langle \tilde{x}_a | a \rangle \).** Since \( \tilde{x}^A \) is a summation of \( N \) random variables that independently obey a gaussian distribution with mean \( 1/N \) and variance \( 1/N \), it obeys a gaussian distribution whose mean is one and variance is one. Using Bayes’ theorem,

\[
p_A(\tilde{x}_A|A) = p_A \frac{\int_{-\infty}^{\infty} d\tilde{x}^A \tilde{x}^A p(\tilde{x}^A|a = A)}{\int_{-\infty}^{\infty} d\tilde{x}^A p(\tilde{x}^A|a = A)} = p_A \frac{\int_{-\infty}^{\infty} d\tilde{x}^A \tilde{x}^A p(a = A|\tilde{x}^A) p(\tilde{x}^A)/p(a = A)}{\int_{-\infty}^{\infty} d\tilde{x}^A p(a = A|\tilde{x}^A) p(\tilde{x}^A)/p(a = A)} = \int_{-\infty}^{\infty} d\tilde{x}^A \tilde{x}^A p(a = A|\tilde{x}^A) p(\tilde{x}^A). \]  \hspace{1cm} (A.7)

Since the variance of \( u^A \) given the value of \( \tilde{x}^A \) is \( l_A^2 - \bar{J}_A^2 \), \( p(a = A|\tilde{x}^A) \) is calculated as

\[
p(a = A|\tilde{x}^A) = \Pr[u^A > u^B|\tilde{x}^A] = \frac{1}{2} \text{erfc}\left(\frac{-\bar{J}_A \tilde{x}^A - \bar{J}_B}{\sqrt{2(l_A^2 - \bar{J}_A^2 + l_B^2)}}\right). \]  \hspace{1cm} (A.8)

Substituting this into equation A.7, we arrive at equation 4.8.

**Appendix B: Results for \( N = 1 \)**

To enable comparison with large \( N \), here, we report the results for the case of \( N = 1 \). This analysis is partly included in Loewenstein (2008).

For \( N = 1 \), the learning dynamics is described by the average trajectory approximation (Kempter, Gerstner, & van Hemmen, 1999). This analysis is based on the assumption that learning is slow, that is, the learning rate \( \eta \) is small; hence, the trajectory of learning is approximated with an average update. For the reward-modulated Hebb rule, the mean update for synaptic
**Figure 11:** Learning behavior for reward-modulated Hebb rule with \( N = 1 \). The learning rate is set to a small value (\( \eta = 0.0001 \)) so that the average trajectory approximation can be applied. Symbols and error bars correspond to average and standard deviation over 100 simulations. Lines indicate average trajectories obtained from equation B.1. In contrast, with large \( N \), \( p_A \) converges to perfect matching behavior, \( p_{\text{match}}^A \), and after a transient period, synaptic efficacies \( J^A \) and \( J^B \) are constant on average.

Efficacy \( J^a, a = \{A, B\} \) is

\[
\langle \Delta J^a \rangle = p_a \eta (\langle r | a \rangle - \langle r \rangle) \langle x^a_i | a \rangle, \tag{B.1}\]

and \( \langle x^a_i | a \rangle \) is obtained by using a similar calculation with equation 4.8:

\[
\langle x^a_i | a \rangle = 1 + \frac{J_a}{p_a \sqrt{2\pi (J_A^2 + J_B^2)}} \exp \left\{ -\frac{(J_A - J_B)^2}{2(J_A^2 + J_B^2)} \right\}. \tag{B.2}\]

Equation 4.2 for \( p_A \) still holds for \( N = 1 \) with \( l^a = |J^a| \). Figure 11 plots the results for the reward-modulated Hebb rule. In contrast to our results for large \( N \), network behavior converges to perfect matching behavior, and after convergence, the synaptic efficacies retain the same values on average. This result can be understood by substituting the condition of the matching law, \( \langle r | a \rangle = \langle r \rangle \), into equation B.1, which leads to making the average update vanish.

**Appendix C: Common Input Model**

In our original model, we assumed that the value-encoding neurons for different actions would receive inputs from separate sensory-neuron pools. As we discussed in section 7.5, the value-encoding neurons in a real brain may share inputs from some sensory neurons. Here, we consider an extreme
Figure 12: Architecture of the decision-making network with common sensory inputs. Output units $u^A$ and $u^B$ share the same sensory input pool denoted by $x_i$ ($i = 1, \ldots, N$). Except for this point, model is the same as that in Figure 1.

case that is opposite our original model: the model shares all the sensory input neurons denoted by $x = (x_1, \ldots, x_N)$ (see Figure 12). We call this model a common-input model.

Let us derive a macroscopic description of the behavior of the common-input model for the case with large fluctuations in input. The mean and variance of $u_a$ are $\bar{J}_a$ and $l_a^2$ ($a = A, B$) as in the separate-input model. However, $u^A$ and $u^B$ can have a correlation due to common inputs in this case. Defining an order parameter,

$$Q = \frac{1}{N} \sum_{i=1}^{N} J^A_i J^B_i, \quad (C.1)$$

we can confirm that the covariance between $u^A$ and $u^B$ is $Q$. From this fact, the probability of choosing alternative $A$ becomes

$$p_A = \frac{1}{2} \text{erfc} \left\{ \frac{-\bar{J}_A - \bar{J}_B}{\sqrt{2(l_A^2 + l_B^2 - 2Q)}} \right\}. \quad (C.2)$$

The correlation coefficient of $u^A, u^B$ is described as

$$R = \frac{Q}{l_A^B}. \quad (C.3)$$
We then calculate the differential equations for the evolution of the order parameters. For $\bar{J}_a$ and $l_a$, we can obtain them similarly to the input-unshared case as

$$\frac{dl_a^2}{d\alpha} = 2\langle F_a u_a \rangle + \langle F_a^2 \rangle,$$  \hspace{1cm} (C.4)

$$\frac{d\bar{J}_a}{d\alpha} = (F_a \bar{x}).$$ \hspace{1cm} (C.5)

To derive the evolution equation for $Q$, we take the product of both sides of

$$J_A(t + 1) = J_A(t) + F_A x,$$ \hspace{1cm} (C.6)

$$J_B(t + 1) = J_B(t) + F_B x.$$ \hspace{1cm} (C.7)

Then we obtain

$$Q(t + 1) = Q(t) + \frac{1}{N} \{ F_A(t) u^A + F_B(t) u^B + F_A(t) F_B(t) \} + O\left( \frac{1}{N^2} \right),$$

from which we obtain

$$\frac{dQ}{d\alpha} = \langle F_A u^B \rangle + \langle F_B u^A \rangle + \langle F_A F_B \rangle.$$ \hspace{1cm} (C.8)

For the delta rule, we notice from equation C.5 that $\frac{d\bar{J}_A}{d\alpha} = \frac{d\bar{J}_B}{d\alpha}$ since $F_A = F_B$. This reflects the fact that the updates for $J^A_i$ and $J^B_i$ are exactly the same for all $i$ in the delta rule. This means that the difference between $\bar{J}_A$ and $\bar{J}_B$ is preserved during learning. Thus, it is clear that the delta rule is not suitable for the common-input model. Therefore, we will report only the results for the reward-modulated Hebb rule below.

The specific form of the ensemble averages is obtained for reward-modulated Hebb rule as

$$\langle F_a \bar{x} \rangle = \eta p_a \{ \langle r|a \rangle - \beta \langle r \rangle \} \langle \bar{x}|a \rangle,$$

$$\langle F_a u^{a'} \rangle = \eta p_a \{ \langle r|a \rangle - \beta \langle r \rangle \} \langle u^{a'}|a \rangle$$

$$= \eta \{ \langle r|a \rangle - \beta \langle r \rangle \} (\bar{J}_{a'} - \langle u^{a'}|a' \rangle p_{a'}),$$

$$\langle F_A F_B \rangle = 0.$$
Also,

\[
\langle u^a|a \rangle = \bar{J}_a + \frac{(l_a^2 - Q)}{p_a \sqrt{2\pi L}} \exp \left( -\frac{D_j^2}{2L^2} \right),
\]

\[
\langle \tilde{x}_a|a \rangle = 1 + \frac{\bar{J}_a - \bar{J}_a'}{p_a \sqrt{2\pi L}} \exp \left( -\frac{D_j^2}{2L^2} \right),
\]

where now \( L = \sqrt{l_A^2 + l_B^2 - 2Q}. \)

The results for the reward modulated Hebb rule are given in Figure 13. Except for the fact that the input neuron pool is shared with two networks, all the conditions are the same as those in Figure 4. For both cases with weight normalization and without normalization, the probability of choosing \( A \) approaches matching behavior.

The correlation between \( u^A \), \( u^B \) obtained with equation C.3 decreases and asymptotically approaches a value that is close to zero. (For the case without normalization, this decay is slow compared to the case with normalization. See the inset of Figure 13, bottom left.) This is due to the fact that the updates for \( J^A \) and \( J^B \) are independent in the reward-modulated Hebb rule. Thus, the networks for action \( A \) and \( B \) tend to behave independently, and the qualitative property observed in the input-unsharing model holds; the variance in weight distribution increases due to diffusion processes (\( \sigma_A, \sigma_B \) in Figure 13, middle row). This can cause undermatching behavior.

In summary, when the output neurons for different actions share input from the sensory neuron pool, the performance of the delta rule is fatally affected and learning does not work. However, for the reward-modulated Hebb rule, the qualitative behaviors of learning remain basically unaffected.

**Appendix D: Multiple Output Model**

While the model considered in this letter has large numbers of input units, the output consists of only one unit for each action. More realistic models contain large numbers of neurons in competing neuron pools that correspond to our output units Wang (2002). Let us consider an extension of our model to a multiple-output case. The potential of output units is denoted by \( u^a_i (a = A, B, \ i = 1, \ldots, M) \) and computed as \( u^a_i = \sum_{j=1}^{N} J^a_{ij} x^a_j \). The network is assumed to choose action \( A \) when \( \sum_{i=1}^{M} u^A_i > \sum_{i=1}^{M} u^B_i \). If the initial weight values satisfy \( J^a_{ij} = J^b_{ik} \), this relation is preserved during learning in both learning rules treated in this study. Then, by reducing \( u^A = \sum_{i=1}^{M} u^A_i \), \( J^a_{ij} = J^b_{ij} \), the behavior of the model becomes equivalent to that of our original. On another extreme, if \( J^a_{ij} \) and \( J^a_{ik} (k \neq l) \) are distributed independently, individual output \( u^a_i \) fluctuates independently. The variance in summed output...
Figure 13: Sample behavior of the common-input model for reward-modulated Hebb rules with (left) no normalization and (right) normalization. From the top, the evolution of choice probability, order parameters, and directional cosine \( R = \frac{Q}{Q_l A_l B} \) corresponds to correlation coefficients of output units. Error bars indicate standard deviation (s.d.). Except for fact that the input neuron pool is common to \( u^A \) and \( u^B \), all conditions for simulation, including parameters and weight initialization, are the same as those in Figure 4. Initial conditions at \( t = \alpha = 0 \) were set as follows. Initial values of \( J_{ai} \) for all \( a, i \) in simulations were drawn from gaussian distribution with mean 5.0 and variance 1.0. Initial value for order parameters were fitted to initialization in simulations.

\[ \sum_{i=1}^{M} u_i^a = O(1/M) \] compared to individual output \( u_i^a \). Then the effect of diffusion can be weaker. However, as long as \( M \) is finite, the diffusion of weight distribution can affect performance. In addition, as learning continues, the correlation between \( u_i^a \)'s for different \( i \) increases since \( f_{ij}^a \) and \( f_{kj}^a (k \neq l) \) obey the same update rule. Thus, the properties of our analysis are expected to hold for the multiple-output-unit case.
Appendix E: Recurrent Spiking Neuron Network

To illustrate how our simple decision-making network can be related to more biologically realistic situations, we perform simulations with decision-making networks based on linear-nonlinear-Poisson (LNP) neuron models (Beck et al., 2008). The model consists of \( N_S \) sensory input afferents and recurrent networks that contain \( N_E \) pyramidal neurons and \( N_I \) inhibitory interneurons, both for two choices. Sensory neuron groups project to only the pyramidal neuron group of the same actions. Within the same choice group, the pyramidal neurons and the interneurons connect to each other in the all-to-all manner. For simplicity, we do not explicitly include connections between action groups.

The sensory input afferents are modeled by Poisson processes. At each discretized time step \( t \), the probability of a spike in neuron \( i \) for action \( a \) \((a = A, B)\) was determined by \( \lambda_a(t) \Delta t \), where \( \Delta t \) is the width of a discretized time bin. Here, we used \( \Delta t = 1 \) ms. We represented presence of targets by setting \( \lambda_a(t) = 40 \) Hz and the absence of targets by \( \lambda_a(t) = 5 \) Hz.

Neurons in the recurrent networks were represented by LNP neuron models. In the linear step, the membrane potential of pyramidal neuron \( i \) of choice group \( a \), denoted \( u_{a,i}(t) \), is described by

\[
- \frac{u_{a,i}(t + \Delta t)}{\Delta t} = \left(1 - \frac{u_{a,i}(t)}{\tau_E} \right) - \frac{1}{\tau_E} \left( \sum_j w_{EE} s_{E,j}^a(t) + \sum_j w_{EI} s_{I,j}^a(t) \right),
\]

where \( s_{E,j}^a \) and \( s_{I,j}^a \) are binary variables that represent the presence of a spike (=1) or the absence of a spike (=0) for the sensory input, the pyramidal neurons, and the interneurons, respectively. For the interneuron, the membrane potential \( u_{a,i}(t) \) evolves according to

\[
- \frac{u_{a,i}(t + \Delta t)}{\Delta t} = \left(1 - \frac{u_{a,i}(t)}{\tau_I} \right) + \frac{1}{\tau_I} \sum_j w_{IE} s_{E,j}^a(t).
\]

In the subsequent nonlinear step, the spiking probability is determined from

\[
p(s_{a,i}^m(t) = 1) = \frac{\nu_{\text{max}}}{1 + \exp \left( - \beta u_{a,i}^m(t) \right)} \Delta t,
\]

where \( m = I, E \). We set \( \nu_{\text{max}} = 100 \) Hz and \( \beta = 1 \). The time constants \( \tau_E, \tau_I \) were set to 0.5 s. These relatively long time constants represent a slow
synaptic current such as the NMDA receptor-mediated current, which has been assumed to underlie temporal accumulation processes in decision making (Wang, 2002; Beck et al., 2008). A decision was made when the population firing rate of one group of pyramidal neurons reached a fixed threshold (40 Hz). The population firing rates were calculated by using sliding window whose widths were five time bins and by averaging over pyramidal neurons in the same choice group. Our model did not include an attractor dynamics that realizes the winner-take-all competition.

The recurrent connections were set to \( w_{EE} = 100/\sqrt{N_E} \), \( w_{IE} = 50/\sqrt{N_E} \), and \( w_{EI} = 50/\sqrt{N_I} \). To examine the influence of the variance of the synaptic weight distribution, we generated \( J_{aij} \) as

\[
c_{ij} = [\tilde{c}_i + \sigma_c \xi_j]_+, \quad J_{aij} = w_{ES} c_{aij},
\]

(E.1)

where \( \xi_j \) was drawn from a unit gaussian distribution with zero mean and unit variance, and \( [a]_+ = \max(a, 0) \). \( \xi_j \) provides a weight variance that is common to postsynaptic neurons. We set \( w_{ES} = 50/\sqrt{N_S} \). The synaptic weight diffusion discussed in this letter is represented by \( \xi_j \), and thus \( \sigma^2 \xi > 0 \), since the diffusion is caused by fluctuation in presynaptic input.

An example of the model behavior when the synaptic weight parameters are \( \tilde{c}_A = 1.025 \), \( \tilde{c}_B = 0.975 \), and \( \sigma_c = 0.4 \) is shown in Figure 14A. After the target appeared, the population firing rate of pyramidal neurons in both groups gradually increased. In this case, the neurons in group A reached the threshold first; hence, the network’s choice was A. The population firing rates of inhibitory neuron groups tracked those of pyramidal neurons. Without inhibition from the inhibitory neurons, the firing of pyramidal neurons reaches the threshold too quickly under the current condition in which the input from individual sensory neurons was strong (of order \( 1/\sqrt{N_S} \)), so that their fluctuations can influence the output variance (data not shown). This mechanism, combined with the synaptic weight variance, can affect the choice probability. As shown in Figure 14B, the choice becomes more random as the weight variance increases. This feature is shared with our original feedforward network discussed in the main text.

The LNP network model differs from our original model on the point that the inputs are Poisson spike trains, whereas in our original model, the inputs are gaussian random variables. However, as the LNP network accumulates the Poisson spike trains over time through the long time constant, according to the central limit theorem, the accumulated inputs become closer to a gaussian random variable whose mean and variance are of order unity. In addition, recurrent inhibitory inputs suppress the output potential so that input fluctuation can affect the choice probability, as shown in Figure 14B. From these considerations, our original model, together with the balanced network interpretation (discussed in section 7.3), can approximate the behavior of more realistic decision-making networks.
Figure 14: Simulation results for the spiking decision-making network with recurrent connections where $N_S = N_E = N_I = 200$. (A) The population firing rates of the pyramidal neuron group (solid lines) and interneurons (dashed lines). The parameters are $\bar{c}_A = 1.025$, $\bar{c}_B = 0.975$, and $\sigma_c = 0.4$ (other parameters are shown in the main text). (B) The choice probability for various synaptic weight variances as a function of the difference between the mean of $c_A$ and $c_B$. Note that these means are in general different from $\bar{c}_A$ and $\bar{c}_B$ in equation E.1, due to the rectification operation (i.e., $\lceil \cdot \rceil_+$. The choice probability for each set of the synaptic weight parameters is computed from 1000 simulation trials. For each $\sigma_c$, the choice probabilities are fitted by a sigmoid function (shown by lines).

Acknowledgments

We thank Yutaka Sakai of Tamagawa University for helpful suggestions, especially on the balanced network discussed in section 7.3. We are also grateful to two anonymous reviewers for helpful comments. This work was partially supported by Grants-in-Aid numbers 20240020, 20650019, 18079003, 18079008, and 20300096, from the Ministry of Education, Culture, Sports, Science, and Technology of the Japanese Government.

References


Received December 12, 2009; accepted October 24, 2011.