Probabilistic Synapses

Laurence Aitchison\(^1\), Alex Pouget\(^{1,2,3}\) and Peter E. Latham\(^1\)

\(^1\)Gatsby Computational Neuroscience Unit, UCL, London, UK
\(^2\)University of Geneva, Geneva, Switzerland
\(^3\)University of Rochester, Rochester, US

Abstract

Learning, especially rapid learning, is critical for survival. However, learning is hard: a large number of synaptic weights must be set based on noisy, often ambiguous, sensory information. In such a high-noise regime, keeping track of probability distributions over weights — not just point estimates — is the optimal strategy. Here we hypothesize that synapses take that optimal strategy: they do not store just the mean weight; they also store their degree of uncertainty — in essence, they put error bars on the weights. They then use that uncertainty to adjust their learning rates, with higher uncertainty resulting in higher learning rates. We also make a second, independent, hypothesis: synapses communicate their uncertainty by linking it to variability, with more uncertainty leading to more variability. More concretely, the value of a synaptic weight at a given time is a sample from its probability distribution. These two hypotheses cast synaptic plasticity as a problem of Bayesian inference, and thus provide a normative view of learning. They are consistent with known learning rules, offer an explanation for the large variability in the size of post-synaptic potentials, and make several falsifiable experimental predictions.

To survive, animals must accurately estimate the state of the external world. This estimation problem is plagued by uncertainty: not only is information often extremely limited (e.g., because it is dark) or ambiguous (e.g., a rustle in the bushes could be the wind, or a it could be a predator), but sensory receptors, and indeed all neural circuits, are noisy. Historically, models of neural computation ignored this uncertainty, and relied instead on the idea that the nervous system represents a single point estimate \([1]\). However, this does not seem to be what animals do — not only does ignoring uncertainty lead to suboptimal decisions, it is inconsistent with a large body of experimental work \([2, 3]\). Thus, the current view is that in many, if not most, cases, animals keep track of uncertainty, and use it to guide their decisions \([3]\).
Accurately estimating the state of the world is just one problem faced by animals. They also need to learn, and in particular they need to leverage their past experience. It is believed that learning primarily involves changing synaptic weights. But estimating the correct weight, like estimating the state of the world, is plagued by uncertainty: not only is the information available to synapses often extremely limited (in many cases just pre and post synaptic activity), but that information is extremely noisy. Historically, models of synaptic plasticity ignored this uncertainty, and relied instead on the idea that synapses make a single point estimate of their weight [3]. However, uncertainty is important for optimal learning — just as it is important for optimal inference of the state of the world.

Motivated by this observation, we propose two hypotheses. The first, Bayesian Plasticity, states that during learning, synapses do indeed take uncertainty into account. Under this hypothesis, synapses do not just try to find a point estimate of their weights, as is done in almost all learning rules in neuroscience; instead, they learn a probability distribution over their weights. This allows synapses to adjust their learning rates on the fly: when uncertainty is high, learning rates are turned up, and when uncertainty is low, learning rates are turned down. These adjustments allow synapses to learn faster, so there is likely to be considerable evolutionary pressure for such a mechanism.

Bayesian Plasticity is a hypothesis about what synapses compute. It does not, however, tell us how synapses should set their weights. For that we need a second hypothesis. Here we propose that weights are sampled from the probability distribution describing the synapses’s degree of uncertainty. Under this hypothesis, which we refer to as Synaptic Sampling, trial to trial variability gives us a direct readout of uncertainty: the larger the trial to trial variability in a synaptic strength, the larger the uncertainty. Combined, these hypotheses make several strong experimental predictions. One is consistent with re-analysis of existing experimental data; the others, which are feasible in the not so distant future, could falsify the model.

We begin our analysis with a derivation of learning rules under the assumption that synapses keep track of their uncertainty (Bayesian Plasticity). That gives us a set of rules for updating not just the mean weight (as all standard learning rules do), but also the uncertainty. We then add to our framework a method for choosing the PSP variability (Synaptic Sampling). Finally, we discuss the experimental implications of our two hypotheses.

We begin with a simplified model of synaptic integration. Neurons in vivo receive a constant barrage of spikes, and each incoming spike produces a PSP — a small change in the postsynaptic neuron’s membrane potential. Very approximately, PSPs combine linearly, allowing us to write the membrane potential relative to rest as

$$V(t) = \sum_i w_i(t)x_i(t) + \eta V(t) \quad (1)$$

where $x_i(t)$ is the synaptic input from neuron $i$, $w_i(t)$ is the corresponding PSP amplitude,
and $\eta V(t)$ is the membrane potential noise. For simplicity we work in discrete time, so $t = 0, 1, 2, ..., \text{ and time steps are on the order of the membrane time constant, around 10 ms}$ [4]. The synaptic inputs, $x_i(t)$, represent the number of incoming spikes in a time step. For most of our analysis, $x_i(t)$ is either 0 (no spike) or 1 (spike), with the probability of a spike chosen to correspond to typical firing rates observed in cortex. To take into account variability in PSP amplitudes, $w_i(t)$ varies from time step to time step. See Methods, Sec. M1.3 for additional details.

We are interested in how synapses learn a set of target weights, denoted $w_{\text{tar},i}(t)$, and a target membrane potential, $V_{\text{tar}}(t)$; the two are related via

$$V_{\text{tar}}(t) = \sum_i w_{\text{tar},i}(t)x_i(t). \tag{2}$$

These target weights have different meanings in different contexts, but broadly, they are the weights that allow the neuron to perform its particular task as effectively as possible. For instance, in a cerebellar Purkinje cell, the target weights might allow the cell to best predict the occurrence of an airpuff; in motor cortex, the target weights might allow the cell to contribute to the best possible skilled movement (e.g., a golf swing that gives a hole-in-one); and in visual cortex, the target weights might enable the cell to pick out the most interesting visual feature in its input. Note that the target weights are unlikely to be fixed, as the statistics of the external world are not fixed (e.g., the stimuli predicting an airpuff can change), nor is the organism (e.g., as you get stronger you will need to adapt your golf swing). Thus, we expect the target weights to change over time, something we include in our analysis (see Methods, Sec. M1.2).

To learn the target weights, synapses get information from the presynaptic input, back-propagating action potentials, and, for supervised and reinforcement learning, an explicit feedback signal, denoted $f$. The simplest feedback signal, which corresponds to the typical supervised learning set-up [5, 6], is $f(\delta) = \delta$, where $\delta$ is the prediction error corrupted by additive noise, $\eta_\delta$,

$$\delta(t) = V_{\text{tar}}(t) - V(t) + \eta_\delta(t). \tag{3}$$

We refer to this as continuous feedback, because $f$ is a continuous function of $\delta$. However, our framework is flexible enough to cover many other supervised and reinforcement learning feedback signals, including discontinuous ones, and even unsupervised learning, for which there is no feedback signal. In particular, we consider three scenarios. The first corresponds to cerebellar learning, in which a Purkinje cell receives a complex spike if its output is too high, thus triggering long term depression [7]. To mimic the all-or-nothing nature of a complex spike [8], we use a binary feedback signal: $f(\delta) = \text{sign}(\delta - \theta)$. For this feedback signal, $f$ is 1 if the noisy error signal, $\delta$ is above a threshold, $\theta$, and $f$ is $-1$ if it is below that
threshold. The second scenario corresponds to reinforcement learning, in which the feedback, now representing the reward, reports the magnitude of the noisy error signal, but not its direction, \( f(\delta) = -|\delta| \). The third corresponds to unsupervised learning, in which there is no feedback signal. Instead, synapses adjust their weights using a Hebbian-like learning rule to find the most interesting (in this case, non-Gaussian) direction in the inputs. See Methods, Sec. M1.5, for additional details.

For the continuous feedback signal, \( f = \delta \), there is a well known rule for finding the optimal weights: the delta rule [5, 9], which changes the mean PSP amplitude, \( m \), according to

\[
\Delta m = \alpha x \delta.
\]

(4)

(We focus on the mean weight because the actual weight, \( w \), varies considerably from one time step to the next due to stochastic vesicle release [10].) This is the product of a learning rate, \( \alpha \) (red), a presynaptic term, \( x \) (green) and a postsynaptic term \( \delta \) (blue). Importantly, the learning rate, \( \alpha \), is the same for all synapses, so all synapses whose presynaptic cells are active (i.e., for which \( x = 1 \)) change by the same amount (the red arrow labeled “delta rule” in Fig. 1).

In the absence of any other information about the history of inputs, the delta rule is perfectly reasonable. However, suppose that, based on previous information, synapse 1 is relatively certain about its weight, whereas synapse 2 is uncertain (error bars in Fig. 1). In that case, new information should have a larger impact on synapse 2 than synapse 1, so synapse 2 should update its weight more (red arrow labeled “optimal” in Fig. 1). Thus, the delta rule does not exploit information about uncertainty, even when it is available, making it suboptimal. To do better, synapses need to compute their uncertainty (essentially, provide error bars), and exploit that information when updating the weights. In essence, synapses must solve an inference problem, in which the goal is to infer the probability distribution over the target weights given available data. So instead of keeping track of point estimates and updating those when spikes arrive, as in the delta rule, synapses keep track of probability distributions over their weights, and update the whole distribution when spikes arrive. That updating process is illustrated in Fig. 2.

We refer to learning in which synapses keep track of probability distributions as Bayesian Plasticity, so named because the update rules are derived using Bayes’ theorem. Synapses do not, of course, have the resources to keep track of arbitrary probability distributions. We therefore assume that each synapse uses an approximate form for its probability distribution, a log normal, chosen because it does not allow weights to change from excitatory to inhibitory (see Methods, Sec. M2). Using this approximate distribution, synapses only have to keep track of the mean and variance, denoted \( m \) and \( s^2 \), respectively. As we show in Supplementary Information, Secs. S1.2 and S1.3, in the case of supervised learning with continuous
feedback, $f = \delta$, the update rules for the mean, $m_i$ and variance, $s_i^2$, are approximately,

$$\Delta m_i \approx \alpha_i x_i \delta - \frac{1}{\tau} (m_i - m_{prior})$$  \hspace{1cm} (5a)

$$\Delta s_i^2 \approx -\alpha_i x_i^2 s_i^2 - \frac{2}{\tau} (s_i^2 - s_{prior}^2)$$  \hspace{1cm} (5b)

where $\alpha_i$ is the learning rate, which now varies across synapses (see Eq. (6) below), and $\tau$, $m_{prior}$, and $s_{prior}^2$ are fixed parameters. To move to the fully general case, including reinforcement and unsupervised learning, we simply replace the postsynaptic terms, $\delta$ in the update for the mean, and $s_i^2$ in the update for the variance by something slightly more complicated (see Supplementary Information, Eq. (S.21)).

The update rule for the mean weight, Eq. (5a), is very similar to the delta rule, in that it is composed of a learning rate (red), a presynaptic term (green) and a postsynaptic term (blue). However, there are two important differences. First, as we show in Supplementary Information, Sec. S1.2, the learning rate, $\alpha_i$, is proportional to each synapse’s uncertainty, as measured by $s_i^2$,

$$\alpha_i = \frac{s_i^2}{s_\delta^2}$$  \hspace{1cm} (6)

where $s_\delta^2$ represents the average variability in $\delta$, and hence in the feedback signal (see Supplementary Information, Eq. (S.10), for the definition of $s_\delta^2$). Thus, when a synapse is more
Figure 2: Updating the distribution over weights using Bayes theorem. At time $t$, synapse $i$’s current probability distribution over the target weight, $w_{\text{tar},i}$, is given by $P(w_{\text{tar},i}(t)|\text{Data up to } t-1)$ (red curve). The neuron receives a small amount of new information via the likelihood, $P(\text{Data at } t|w_{\text{tar},i}(t))$ (green curve). This leads to a new distribution, $P(w_{\text{tar},i}(t)|\text{Data up to } t)$ (blue curve).

uncertain about its target weight, new information causes a larger change in the mean weight — exactly what we expected, given Fig. 1. In contrast, as the feedback signal gets noisier, and thus less informative, the learning rate falls. Second, there is a decay term (grey), which causes the mean to decay back to its prior value. This accounts for the fact that the underlying target weight, $w_{\text{tar},i}$, changes over time (as mentioned above), so information from the recent past is more relevant than information from the distant past.

Although the update rule for the uncertainty, $s_{i}^{2}$ (Eq. (5b)), does not have a counterpart in classical learning rules, it does have a natural interpretation. The first term in Eq. (5b) reduces uncertainty (note the negative sign) whenever the presynaptic cell is active ($x_{i} = 1$), and thus whenever the synapse updates its estimate of the weight. The second term has the opposite effect: it increases uncertainty. That term arises because random drift reduces knowledge about the target weights.

Simulations (Fig. 3) show that the mean weight tracks the target weight very effectively (compare the red and blue lines, which correspond to the mean of the inferred distribution and the target weight, respectively). Just as importantly, the synapse’s estimate of its uncertainty tracks the difference between its estimate and the actual target (the blue line should be inside the 95% confidence intervals 95% of the time; in practice, we have: supervised continuous, 95%; supervised binary, 94%; reinforcement, 89%; unsupervised, 87%).

The critical aspect of the learning rules in Eq. (5) is that the learning rate — the change in mean PSP amplitude, $m_i$, per spike — increases as the synapse’s uncertainty, $s_{i}^{2}$, increases.
Figure 3: Bayesian learning rules track the true weight and estimate uncertainty. The blue line is the true weight, the red line represents the median of the inferred distribution, and the red area represents 95% confidence intervals. The total time course is 5 times the characteristic time over which the target weights change (see Methods, Sec. M1.2). A. Supervised learning, continuous feedback ($f = \delta$). B. Supervised learning, binary feedback ($f = \Theta (\delta - \theta)$). C. Reinforcement learning ($f = -|\delta|$). D. Unsupervised learning (no feedback).
Figure 4: Bayesian learning rules have a lower mean squared error (MSE) than classical learning rules. The red line is the mean squared error for the classical learning rule, relative to our Bayesian learning rule (the blue line at 1). The Bayesian learning rule does not have a tunable learning rate parameter, so the Bayesian mean squared error is the same for all learning rates. A. Supervised learning, continuous feedback ($f = \delta$). B. Supervised learning, binary feedback ($f = \Theta (\delta - \theta)$). C. Reinforcement learning ($f = -|\delta|$). D. Unsupervised learning (no feedback). See Methods, Sec. M4, for further details.

This is a general feature of our learning rules, and not specific to any one of them. Consequently, independent of the learning scenario, we expect performance to be better than for classical learning rules, which do not take uncertainty into account. To check whether this is true, we computed the mean squared error between the actual and target membrane potential, $V$ and $V_{\text{tar}}$, for classical learning rules, and plotted them relative to our learning rules. The results are shown in Fig. 4. In this figure, the red line gives the mean squared error for the classical learning rules relative to the error for our optimal rules. Note that the Bayesian learning rules do not have an externally imposed learning rate parameter, so their mean squared error is a single value that does not vary with learning rate. Even if the learning rates for the classical learning rules are chosen optimally, performance is worse than it is for the probabilistic learning rules, and if they are chosen sub-optimally, performance can be much worse.

Fig. 4 indicates that there is a clear advantage to using uncertainty to adjust learning
rates. But does the brain actually take this strategy? Addressing that question will require a new generation of plasticity experiments: at present, in typical plasticity experiments only changes in weights are measured; to test our hypothesis, it will be necessary to measure changes in learning rates, and at the same time determine how those changes are related to the synapse’s uncertainty. This presents two challenges. First, measuring changes in learning rates is difficult, as weights must be monitored over long periods of time and under natural conditions, preferably in vivo. However, with the advent of increasingly sophisticated experimental techniques, such experiments should be feasible in the not so distant future. Second, we cannot measure the synapse’s uncertainty directly. It is, therefore, necessary to find a proxy. Below we discuss two possible approaches.

The first approach is indirect: use neural activity measured over long periods in vivo to estimate the uncertainty a synapse should have; then, armed with that estimate, test the prediction that the learning rate increases with uncertainty. To estimate the uncertainty a synapses should have, we take advantage of a general feature of essentially all learning rules: synapses get information only when the presynaptic neuron spikes. Consequently, the synapse’s uncertainty should fall as the presynaptic firing rate increases. In fact, under mild assumptions, we can derive a very specific relationship: the relative change in weight under a plasticity protocol, ∆m_i/m_i, should scale as 1/√ν_i where ν_i is the firing rate of the neuron presynaptic to synapse i,

\[ \frac{\Delta m_i}{m_i} \propto \frac{1}{\sqrt{\nu_i}} \]  

(7)

a relationship that holds in our simulations (Fig. 5; see also Supplementary Information, Sec. S4). In essence, firing rate is a proxy for uncertainty, with higher firing rate indicating lower uncertainty and vice versa. This prediction can be tested by observing neurons in vivo, estimating their firing rates, then performing long term potentiation or depression experiments to determine the relative change in synaptic strength, ∆m_i/m_i.

The second approach involves the introduction of a new hypothesis, which is that PSP variability provides a proxy for uncertainty. That we might expect a relationship between variability and uncertainty is based on the following normative reasoning (see Methods, Sec. M5, for an extended discussion): the uncertainty associated with a particular computation should depend on the uncertainty in the weights; thus, to make optimal decisions, the brain needs to know that degree of uncertainty; one way to communicate it is via variability in PSP amplitude. This leads to the Synaptic Sampling hypothesis, which states that the variance in PSP amplitude is equal to the variance of the inferred posterior distribution over the target weight, s^2_i,

\[ \text{PSP variance} = s^2_i. \]  

(8)
Figure 5: Simulations confirming that the normalized learning rate ($\alpha_i/m_i$, which is proportional to $\Delta m_i/m_i$) is inversely related to the square root of the firing rate. As predicted, the best fit line on a log-log plot has a slope close to $-1/2$. A. Supervised learning, continuous feedback ($f = \delta$). B. Supervised learning, binary feedback ($f = \Theta(\delta - \theta)$). C. Reinforcement learning ($f = -|\delta|$). D. Unsupervised learning (no feedback).
This is analogous to setting the PSP mean to the mean of the distribution over the target weight, \( m_i \). We call this the Synaptic Sampling hypothesis because the synapses “sample” PSP amplitudes from their inferred distribution over weights.

Bayesian Plasticity combined with Synaptic Sampling tells us that synapses with higher variability (and hence higher uncertainty) should have higher learning rates. More quantitatively, Bayesian Plasticity tells us that the relative change in PSP amplitude, \( \Delta m_i/m_i \), is proportional to the synapse’s uncertainty, (Eqs. (5a) and (6)) and Synaptic Sampling relates uncertainty to variability (Eq. (8)); consequently,

\[
\frac{\Delta m_i}{m_i} \propto \frac{\text{PSP variance}}{\text{PSP mean}} \equiv \text{Normalized Variability},
\]

where we have defined the normalized variability to be the ratio of PSP variance to its mean. We verify that this relationship holds in simulation in Fig. 6.

Equation (9) implies that when the PSP variance is high, learning is fast. Testing that experimentally is straightforward, if technically difficult: simply monitor the PSP mean and variance for long periods in vivo, and compare normalized variability to changes in the mean. The in vivo requirement is important: our analysis assumes a constant barrage of presynaptic spikes, whereas in many in vitro preparations the vast majority of cells are silent (see Supplementary Information, Sec. S4).

In addition to the experiment proposed above, there is a slightly more indirect test of Bayesian plasticity and Synaptic Sampling. Combining Eq. (7) and (9), we see that the normalized variability and firing rate obey the relationship,

\[
\frac{1}{\sqrt{r_i}} \propto \text{Normalized Variability}.
\]

This is intuitively sensible: as discussed previously, higher presynaptic firing rates means the synapse is more certain, and Synaptic Sampling states that higher certainty should reduce the observed variability.

This relationship can be tested by estimating presynaptic firing rates in vivo, and comparing them to the normalized variability measured using paired recordings. Such data can be extracted from experiments by Ko and colleagues [11]. In those experiments, calcium signals in mouse visual cortex were recorded in vivo under a variety of stimulation conditions, which provided an estimate of firing rate; subsequently, whole cell recordings of pairs of identified neurons were made in vitro, and the mean and variance of the PSPs were measured. In Fig. 7A we plot the normalized variability versus the firing rate on a log-log scale; on this scale, our theory predicts a slope of \(-1/2\) (red line). The normalized variability does indeed decrease as the firing rate increases (blue line), \( p < 0.003 \), and the slope is not significantly different from \(-1/2\) \( (p = 0.56) \). This pattern is broadly matched by simulated data (Fig. 7B)
Figure 6: Simulations confirming that that the normalized learning rate \( \frac{\alpha_i}{m_i} \), which is proportional to \( \Delta \frac{m_i}{m_i} \), is proportional to the normalized variability \( \frac{s_i^2}{m_i} \). The red line is the best fitting straight-line that passes through the origin. A. Supervised learning, continuous feedback \( (f = \delta) \). B. Supervised learning, binary feedback \( (f = \Theta(\delta - \theta)) \). C. Reinforcement learning \( (f = -|\delta|) \). D. Unsupervised learning (no feedback).
Figure 7: Normalized variability (the ratio of the PSP variance to the mean) as a diagnostic of our theory. **A.** Normalized variability falls as firing rate increases. The red line, which has a slope of $-1/2$, is our prediction (the intercept, for which we do not have a prediction, was chosen to give the best fit to the data). The blue line is fitted by linear regression, and the grey region represents 2 standard errors. Its slope, -0.62, is statistically significantly different from 0 ($p < 0.003$) and not significantly different from $-1/2$ ($p = 0.57$). Firing rate was measured by taking the average signal from a spike deconvolution algorithm [12]. Units are arbitrary because the scale factor relating the average signal from the deconvolution algorithm and the firing rate is not exactly one [13]. Data from layer 2/3 of mouse visual cortex [11]. **B.** Simulated normalized variability versus firing rate; supervised learning with continuous feedback ($f = \delta$).

It seems unlikely that this pattern emerged spuriously, as that would require a confound that simultaneously influenced two very different types of measurement, calcium measurements of the pre-synaptic firing rate and patch-clamp measurement of the PSPs. The most obvious confound actually predicts a positive slope: if more calcium indicator is present in the presynaptic cell, then we might expect measured firing rates to be higher, and vesicle release probabilities to be lower (as the indicator buffers calcium involved in vesicle release). Lower probabilities imply higher variability, so we would expect higher measured firing rates to be associated with higher variability — the opposite of our prediction.

In summary, based primarily on theoretical considerations of optimality we proposed that synapses do not just keep track of point estimates of their weights, as they do in classical learning rules; instead, they compute approximate probability distributions over their weights. They then use those distributions to set learning rates: the wider the distribution (that is, the more the uncertainty in the target weight) the higher the learning rate. This allows different synapses to have different learning rates, and leads to learning rules that
allow synapses to exploit all locally available information, and so learn as rapidly as possible — much more rapidly than classical learning rules, which do not keep track of uncertainty (Fig. 4). The critical difference between our learning rules and classical ones is that the learning rates themselves undergo plasticity; the rules for updating the mean weight are very similar to classical learning rules. Thus, our framework is consistent with the vast majority of work on synaptic plasticity [14, 15, 16, 17, 18, 19].

The hypotheses that synapses keep track of uncertainty, which we refer to as the Bayesian Plasticity hypothesis, makes the general prediction that learning rates, not just synaptic strengths, are a function of pre and postsynaptic activity — something that should be testable with the next generation of plasticity experiments. In particular, it makes a specific prediction about learning rates in vivo: learning rates should vary across synapses, being higher for synapses with lower presynaptic firing rates.

We also make a second, independent, hypothesis, Synaptic Sampling. This hypothesis states that the variability in PSP size associated with a particular synapse matches the uncertainty in the strength of that synapse. This allows synapses to communicate their uncertainty to surrounding circuitry — information that is critical if the brain is to monitor the accuracy of its own computations. The same principle has been applied to neural activity, where it is known as the neural sampling hypothesis [20, 21, 22, 23] (except that here variability in neural activity matches uncertainty about the state of the external world). The neural sampling hypothesis meshes well with synaptic sampling: uncertainty in the weights increases uncertainty in the current estimate of the state of the world, and likewise, variability in the weights increase variability in current neural activity (see Methods, Sec. M5). However, while there is some experimental evidence for the neural sampling hypothesis [22, 24, 23], it has not been firmly established. Whether other proposals for encoding probability distribution with neural activity, such as probabilistic population codes [3, 25], can be combined with Synaptic Sampling is an open question.

By combining our two hypotheses, we were able to make additional predictions. These predictions focused on what we call the normalized variability — the ratio of the variance in PSP size to the mean. First, we predicted that plasticity should increase with normalized variability, which remains to be tested. Second, we predicted that normalized variability should decrease with presynaptic firing rate. We reanalysed data from [11] to show that this is indeed the case (Fig. 7).

In machine learning, the idea that it is advantageous to keep track of the distribution over weights has a long history [26, 27, 28]. The first suggestion that such a scheme might be useful in a neuroscience context, however, was relatively recent [3], and the first theoretical study was even more recent [29]. The latter study bore some resemblance to ours, in that weights were sampled from a distribution. However, there was an important difference: the
distribution had to be fixed, and could be determined only after the animal had seen all data. Because this is unrealistic, an online algorithm was developed in which, as in our scheme, weights were updated on each time step. However, for this algorithm to agree with sampling from a fixed distribution, changes in synaptic strength per time step had to be very small (on the order of $10^{-4}$). Thus, unlike in our scheme, there was almost no spike-to-spike variability in PSP size. So, although this was an important step toward a probabilistic treatment of synaptic plasticity, the algorithm was unable to deal with the realistic situation in which the distribution over synaptic weights is changing continuously as the animal receives new information, and it doesn’t produce the variability in PSP size seen in vivo.

If the Bayesian Plasticity hypothesis is correct, synapses would have to keep track of, and store, two variables: the mean and variance of the log of the synaptic weight (or, equivalently, the mean weight and the learning rate). The complexity of synapses [30, 31, 32], and their ability to use interesting, non-trivial learning rules (e.g. synaptic tagging, in which activity at a synapses “tags” it for future long term changes in strength [33, 34, 35], and metaplasticity, in which the learning rate can be modified by synaptic activity without changing the synaptic strength [36, 37, 38]), suggests that representing uncertainty — or learning rate — is quite possible. It will be nontrivial, but important, to work out how.

Our framework has several implications, both for the interpretation of neurophysiological data and for future work. First, under the Synaptic Sampling hypothesis, PSPs are necessarily noisy. Consequently, noise in synapses (e.g., synaptic failures) is a feature, not a bug. We thus provide a normative theory for one of the major mysteries in synaptic physiology: why neurotransmitter release is probabilistic. Second, our approach allows us to derive local, biologically plausible learning rules, no matter what information is available at the synapse, and no matter what the statistics of the synaptic input. Thus, our approach provides the flexibility necessary to connect theoretical approaches based on optimality to complex biological reality.

In neuroscience, Bayes theorem is typically used to analyze high level inference problems, such as decision-making under uncertainty. Here we have demonstrated that Bayes theorem, being the optimal way to solve any inference problem, big or small, could be implemented in perhaps the smallest computationally relevant elements in the brain: the synapse.

Acknowledgements

LA and PEL were supported by the Gatsby Charitable Foundation; AP was supported by a grant from the Simons Collaboration for the Global Brain and a grant from the Swiss National Foundation, #31003A_165831.
Methods

Here we provide a complete description of our model (Sec. M1, which includes a table containing a list of all parameters), sketch the derivation of the learning rules (Sec. M2; the full derivation is given in Supplementary Information), discuss the advantages of our local approach to learning (Sec. M3), provide details of the simulations (Sec. M4), give a normative explanation for the Synaptic Sampling hypothesis (Sec. M5), and, finally, provide additional details of the statistical test used for Fig. 7A (Sec. M6).

M1 Complete description of our model

In the main text we specified how the membrane potential depends on the weights and incoming spikes (Eq. (1)) and how the target membrane potential depends on the target weights (Eq. (2)), and we defined the prediction error (Eq. (3)). Here we describe how the weights, $w_i$, the target weights, $w_{tar,i}$, and the spikes, $x_i$, are generated. We also provide a summary of how the feedback signal, $f$, depends on the prediction error, $\delta$, and we provide details of the unsupervised learning model.

M1.1 Synaptic weights

To take variability in PSP amplitudes into account, we use

$$w_i = m_i + \sqrt{k_i m_i} \eta_{w_i}, \tag{11}$$

where $\eta_{w_i}$ is zero mean, unit variance noise. Under the Synaptic Sampling hypothesis, the variability is equal to the uncertainty, so $k_i = s_i^2/m_i$. However, when comparing classical and Bayesian learning rules (Figs. 3 and 4), we set $k_i = k$ for all synapses. This was necessary to make a fair comparison, as there is no way to compute uncertainty for classical learning rules. The value of $k$ came from measured data [39]: we plotted $s_i^2$ vs $m_i$ and fit a straight line that passed through the origin; $k$ is the slope of that line; this resulted in $k = 0.0877$.

When plotting learning rate versus firing rate (Fig. 5), we also used $k_i = k$, primarily for convenience. However, in Figs. 6 and 7, which explicitly involved the Synaptic Sampling hypothesis, we used $k_i = s_i^2/m_i$.

M1.2 The target weights

The target weights are the weights that in some sense optimize the performance of the animal. We do not expect these weights to remain constant over time, for two reasons. First, both the general state of the world and the organism change over time, thus changing the target weights. Second, we take a local, single neuron view to learning, and define the
target weights on a particular neuron to be the optimal weights given the weights on all the other neurons in the network. Consequently, as the weights on surrounding neurons change due to learning, the target weights on our neuron will also change. While these changes may be quite systematic, to a single synapse deep in the brain they are likely to appear random.

In our model we assume that the log of the target weights follow an Ornstein-Uhlenbeck process. Specifically, we define

$$\lambda_{\text{tar},i} = \log |w_{\text{tar},i}|$$ (12)

(note the absolute value sign, which allows the weights to be either positive or negative), and let $$\lambda_{\text{tar},i}$$, the log weight, evolve according to

$$\Delta \lambda_{\text{tar},i}(t+1) = -\frac{1}{\tau} (\lambda_{\text{tar},i}(t) - \mu_{\text{prior}}) + \sqrt{\frac{2\sigma^2_{\text{prior}}}{\tau}} \eta_{\text{tar},i}$$ (13)

where $$\tau$$ is the characteristic time scale over which the weights change. Note that $$\tau$$ is measured in time steps; to convert to time it needs to be multiplied by $$\Delta t$$, the size of the time step. Under this noise process, the mean value of $$\lambda_{\text{tar},i}$$, denoted $$\mu_i$$, and the variance, denoted $$\sigma_i^2$$, evolve according to

$$\mu_i(t+1) = \left(1 - \frac{1}{\tau}\right) \mu_i(t) + \frac{\mu_{\text{prior}}}{\tau}.$$ (14a)

$$\sigma_i^2(t+1) = \left(1 - \frac{1}{\tau}\right)^2 \sigma_i^2(t) + \frac{2\sigma^2_{\text{prior}}}{\tau}.$$ (14b)

We chose this particular noise process for three reasons. First, $$w_{\text{tar},i}$$ is equal to either $$+e^{\lambda_{\text{tar},i}}$$ (for excitatory weights) or $$-e^{\lambda_{\text{tar},i}}$$ (for inhibitory weights), and thus cannot change sign as $$\lambda_{\text{tar},i}$$ changes with learning. Consequently, excitatory weights cannot become inhibitory, and vice versa, so Dale’s law is preserved. Second, spine sizes obey this stochastic process [40], and while synaptic weights are not spine sizes, they are correlated [41]. Third, this noise process gives a log-normal stationary distribution of weights, as is observed experimentally [39].

The parameters of these dynamics, $$\mu_{\text{prior}}$$ and $$\sigma^2_{\text{prior}}$$, were set to the mean and variance of measured log-weights using data from Ref. [39]. We used a time step, $$\Delta t$$, of 10 ms, within the range of measured membrane time-constants (e.g. [4]), and set $$\tau$$ to $$10^5$$ (corresponding to 1,000 seconds, or around 15 minutes) for both types of supervised learning, and $$10^6$$ (corresponding to 10,000 seconds, or around 2 1/2 hours) for reinforcement and unsupervised learning. These values of $$\tau$$ were chosen so that uncertainty roughly matched observed variability; see Sec. S5.
M1.3  The synaptic inputs, $x_i(t)$, with feedback

For models with a feedback signal, on each time step $x_i$ is drawn from a Bernoulli distribution representing the number of spikes (0 or 1) from the presynaptic cell,

$$P(x_i) = (\nu_i \Delta t)^{x_i} (1 - \nu_i \Delta t)^{1-x_i}. \quad (15)$$

The firing rates, $\nu_i$, are drawn from a log-normal distribution chosen to match observed firing rates. We choose a distribution that is intermediate between the relatively narrow ranges found by some [42], and the extremely broad ranges found by others [43]: we use a log-normal distribution, with median at 1 Hz, and with 95% of firing rates being between 0.1 Hz and 10 Hz,

$$\log \nu_i \sim \mathcal{N} \left( 0, \left( \frac{\log 10}{2} \right)^2 \right). \quad (16)$$

M1.4  Feedback signals for supervised and reinforcement learning

The feedback signal is different for every type of learning (these are mentioned in the text, and are repeated here for completeness).

For supervised learning with continuous feedback, the feedback signal is simply $\delta$,

$$f(\delta) = \delta. \quad (17)$$

For supervised learning with binary feedback, the feedback signal is 1 if $\delta$ is above $\theta$, and $-1$ if it is below $\theta$,

$$f(\delta) = \text{sign}(\delta - \theta). \quad (18)$$

Binary feedback is intended to model Purkinje cells, which receive a complex-spike feedback signal relatively rarely (around once per second; corresponding to once in 100 time steps). To match that rate, $\theta$ should be set high enough that $\delta$ is above $\theta$ relatively rarely. While this is possible (and we have run these simulations), this makes comparison between the Bayesian and classical rules difficult: it is not sufficient simply to fix $\theta$, as this may give rise to different values of $P(f = 1)$ in Bayesian and classical learning. While it may be possible to resolve these difficulties, for the purposes of fair comparison we use $\theta = 0$. Because the distribution over $\delta$ is symmetric around 0, this implies that $P(f = 1)$ remains at 1/2 throughout the simulations for both classical and Bayesian learning.

For reinforcement learning, the feedback signal, representing the reward, is simply minus the magnitude of $\delta$,

$$f(\delta) = -|\delta|. \quad (19)$$
M1.5 Models without a feedback signal

For unsupervised learning, there is no feedback signal. Instead, information for setting the weights comes from structure in the synaptic inputs, $\mathbf{x}$, which is generated by a very different process from supervised and reinforcement learning (for which there was no structure in the input). Specifically, we assume that the cell’s input is Gaussian in every direction except one, $\mathbf{w}_{\text{tar}}$, in which the input is Laplacian. The cell’s goal is to find that one interesting direction (as was done in [44]).

Formally, $\mathbf{x}$ is generated by,

$$P(\mathbf{x}|\mathbf{w}_{\text{tar}}, V_{\text{tar}}) \propto \mathcal{N}(\mathbf{x}; \mathbf{0}, \Lambda) \delta (V_{\text{tar}} - \mathbf{w}_{\text{tar}}^T \mathbf{x}).$$ (20)

where the target membrane potential, $V_{\text{tar}}$, is Laplacian distributed,

$$P(V_{\text{tar}}) = \frac{e^{-|V_{\text{tar}}|/b}}{2b}.$$ (21)

We let

$$b^2 = \frac{\mathbf{w}_{\text{tar}}^T \Lambda \mathbf{w}_{\text{tar}}}{2},$$ (22)

so that moments of $\mathbf{x}$ are the same whether we draw from the full distribution (Eq. (20)) or just from the Gaussian, $\mathcal{N}(\mathbf{x}; \mathbf{0}, \Lambda)$ — as is easy to show by direct calculation. While our theory does not require it, in simulations, we use whitened input, i.e., a diagonal input covariance, to match, for instance, the whitened input from retina to V1,

$$\Lambda_{ij} = \delta_{ij} \nu_i \Delta t.$$ (23)

The diagonal elements are chosen to match the variance expected from a Poisson process.

Note that this form allows $x_i$ to be positive or negative. To some extent, this could be remedied by adding an offset to $x_i$, but considerable work will be needed to write down biologically realistic models for $P(\mathbf{x}|\mathbf{w}_{\text{tar}}, V_{\text{tar}})$ in which Bayesian inference can be performed.
### M1.6 Parameter settings

| Parameter       | Value     | Basis                                           |
|-----------------|-----------|-------------------------------------------------|
| $\mu_{\text{prior}}$ | -0.669    | Matched to data from [39] (Sec M1.2)            |
| $\sigma^2_{\text{prior}}$ | 0.863     | Matched to data from [39] (Sec M1.2)            |
| $n$ (sup., unsup.) | 1000      | Offers a good trade-off between biological realism and computational tractability |
| $n$ (reinforcement) | 100       | Uses a reduced number of synapses for reinforcement learning because of the increased difficulty of the learning problem |
| $\tau$ (supervised) | $10^5$    | Supplementary Information, Sec. S5; corresponds to 1,000 s |
| $\tau$ (unsup., rein.) | $10^6$    | Supplementary Information, Sec. S5; corresponds to 10,000 s |
| $\Delta t$      | 10 ms     | Typical membrane time constant [4]              |
| $\gamma_V$      | 1 mV      | Small value because once the effects of stochastic vesicle release are excluded, membrane potential variability is thought to be small [46, 47] |
| $\gamma_\delta$ | 1 mV      | This is difficult to determine, so we use a small nominal value for computational tractability |
| $k$             | 0.0877    | Matched to data from [39] (Sec M1.1)            |
| $\theta$        | 0         | Sec. M1.4.                                       |

### M2 Inference when there is a feedback signal

Here we outline how a synapse can infer a distribution over the log of its target weight, $\lambda_{\text{tar},i}$, using all past data. We focus on supervised and reinforcement learning, for which there is a feedback signal, as it is relatively straightforward; we analyze unsupervised learning, for which there is no feedback signal, in Supplementary Information (see in particular Sec. S2).

As our model is in a well-understood class, hidden Markov models (HMMs), this inference process is straightforward: we use the standard, two-step procedure for inference in HMMs. In the first step the synapse incorporates new data using Bayes theorem. The data in one time step, denoted $d_i$, includes the presynaptic input, $x_i$, the feedback signal, $f$, the cell’s membrane potential, $V$, and the actual PSP amplitude, $w_i$,

$$d_i(t) \equiv (x_i(t), f(t), V(t), w_i(t)),$$

(24)
and we use $D_i(t)$ to denote all past data,
\[ D_i(t) \equiv (d_i(t), d_i(t-1), \ldots). \] (25)

Using this notation, we have
\[ P(\lambda_{\text{tar},i}|D_i) = P(\lambda_{\text{tar},i}|d_i, D_i(t-1)) \propto P(d_i|\lambda_{\text{tar},i}) P(\lambda_{\text{tar},i}|D_i(t-1)). \] (26)

To reduce clutter, here and in what follows all quantities without an explicitly specified time index are evaluated at time step $t$; so, for instance, $w_{\text{tar},i} \equiv w_{\text{tar},i}(t)$ and $D_i \equiv D_i(t)$.

In the second step, the synapse takes into account random changes in the target weight,
\[ P(\lambda_{\text{tar},i}(t+1)|D_i) = \int d\lambda_{\text{tar},i} P(\lambda_{\text{tar},i}(t+1)|\lambda_{\text{tar},i}) P(\lambda_{\text{tar},i}|D_i). \] (27)

Combining both steps takes us from the distribution at time $t$, $P(\lambda_{\text{tar},i}|D_i(t-1))$, to the distribution at the time $t+1$, $P(\lambda_{\text{tar},i}(t+1)|D_i(t))$.

Equations (26) and (27) tell us how to make exact updates to the distribution over the target weight. However, the exact distribution is too complex for a synapse to work with, let alone store. To simplify the problem faced by the synapse, we specify a family of approximate distributions: a Gaussian in the log-domain, with mean $\mu_i$ and variance $\sigma_i^2$,
\[ P(\lambda_{\text{tar},i}|D(t-1)) = \mathcal{N}(\lambda_{\text{tar},i}; \mu_i, \sigma_i^2). \] (28)

The corresponding mean, $m_i$, and variance, $s_i^2$, of the distribution over $w_{\text{tar},i}$ are
\[ m_i \equiv E[w_{\text{tar},i}|D(t-1)] = e^{\mu_i + \sigma_i^2/2}, \] (29a)
\[ s_i^2 \equiv \text{Var}[w_{\text{tar},i}|D(t-1)] = \left(e^{\sigma_i^2} - 1\right) m_i^2 \approx \sigma_i^2 m_i^2, \] (29b)
the latter valid in the limit $\sigma_i^2 \ll 1$. This is, in fact, a good approximation: on average, $s_i^2/m_i^2 \approx 0.076$ (Supplementary Information, Eq. (S.70c)); combining this with Eq. (29b) gives, again on average, $\sigma_i \approx 0.073$. We thus use it throughout most of our analysis.

This approximate distribution has two advantages. First, log-normal distributions always give positive values, leading to learning rules that cannot, for instance, take an excitatory synapse and turn it inhibitory. Second, if the synapse is not given any data, then the dynamics (Equation (13)) imply that the distribution over $\lambda_{\text{tar},i}$ approaches a Gaussian at long times — exactly our approximating distribution.

As we will see below, the likelihood, $P(d_i|\lambda_{\text{tar},i})$ is typically not Gaussian in $\lambda_{\text{tar},i}$; consequently, even if $P(\lambda_{\text{tar},i}|D_i(t-1))$ is Gaussian, $P(\lambda_{\text{tar},i}|D_i)$ will not be (see Eq. (26)). A natural way to remedy this is Assumed Density Filtering (ADF) [48]. Formally, this requires
Figure 8: A graph describing the dependencies in our simulations. The target weight, $w_{\text{tar},i}(t)$ evolves independently of all other variables, under the exponentiated Ornstein-Uhlenbeck process described in Eq. (13). The data, $d_i(t)$, which includes the feedback signal, $f(t)$, the presynaptic input, $x_i(t)$, the postsynaptic activity $V(t)$ (see Eq. (24)), and the PSP amplitude, $w_i(t)$, depends on both the target weight, $w_{\text{tar},i}(t)$, and on past inferences, $m_i(t)$ and $s_i^2(t)$. In particular, the feedback signal, $f(t)$, depends on the target weight, and the PSP amplitude, $w_i(t)$, depends on the mean estimate of the target weight, $m_i(t)$ (see Eq. (11)). Finally, the mean and uncertainty at time $t$, $m_i(t)$ and $s_i(t)$, depend on the mean and uncertainty at the previous time step, $m_i(t-1)$ and $s_i^2(t-1)$, and also on past data, $d_i(t-1)$, through the learning rules, Eq. (5).

us to find the log-normal distribution with the smallest KL-divergence; this can be achieved by matching moments,

$$\mu_i(t + 1) = E [\lambda_{\text{tar},i}(t + 1)|D_i]$$  \hspace{1cm} (30a)

$$\sigma_i^2(t + 1) = \text{Var} [\lambda_{\text{tar},i}(t + 1)|D_i].$$  \hspace{1cm} (30b)

The central difficulty is computing moments of the inferred distribution, which will require further approximations beyond the assumed density filter. This is dealt with in more depth in Supplementary Information, Secs. S1.2 and S2; see in particular Eq. (S.14).

To summarise our model for a single synapse, we can write down a dependency graph describing how each variable is generated (see Fig. 8). This is a graphical model – a compact method for describing dependencies among random variables. This graphical model has the extremely unusual feature that the results of inference at one time step influence the data at subsequent time steps.

M3 Problems with inference at the cellular level

Our strategy of performing Bayesian inference at the level of the synapse is actually quite unusual (and is potentially the most important theoretical advance in the paper). The more
typical approach is to perform some type of inference at the level of the whole cell (i.e., infer all the weights jointly). We chose our approach because it is unlikely that synapses can communicate much information to each other. The lack of communication is not a problem if we consider each synapse as performing an inference problem, conditioned on the data available to it. However, it is a problem if inference is performed at the cellular level. To illustrate this in the simplest possible context, we consider a cell with two synapses. Synapses are trying to infer their target weights based on the data, \( d_1 \) and \( d_2 \), available at synapse 1 and 2, respectively. Without communication, the best each synapse can do is to compute its target weight, based on its data, \( P(w_{\text{tar},1}|d_1) \) and \( P(w_{\text{tar},2}|d_2) \). However, if we try to infer both weights at the cellular level, then even making the strong approximation that the distribution over each target weight is independent,

\[
P(w_{\text{tar},1}, w_{\text{tar},2}|d_1, d_2) \approx P(w_{\text{tar},1}|d_1, d_2) P(w_{\text{tar},2}|d_1, d_2),
\]

we cannot prevent each synapse from “seeing” all the data (except in the unlikely event that \( d_1 \) really gives no information about \( w_{\text{tar},2} \) and vice-versa).

It may seem highly suboptimal for each synapse to perform inference independently, as synapses have to throw away information (for instance, \( w_{\text{tar},1} \) must average over its prior uncertainty in \( d_2 \), and, likewise, \( w_{\text{tar},2} \) must average over its prior uncertainty in \( d_1 \)). However, from a biological point of view it is quite natural. Nonetheless, this is an unusual approach, and considerable further work is necessary to understand its theoretical properties.

M4 Details of simulations

We performed two sets of simulations, the first, for Bayesian Plasticity, with \( k_i \) fixed at \( k = 0.0877 \) (Figs. 3-5), and the second, for Synaptic Sampling, with \( k_i = s_i^2/m_i \) (Figs. 6 and 7) (see Sec. M1.1).

To reduce the variability in the MSE (mean squared error) estimates, for both Bayesian and classical learning rules we ran all simulations using the same inputs, \( x_i \), and target weights, \( w_{\text{tar},i} \). We repeated the protocol 24 times, with different inputs and target weights. Using the same inputs and target weights reduced the variability in MSE measurements between learning rates below what might be expected based on the 2 s.e.m. error bars in Fig. 4.

To avoid error bars on the MSE that were larger than the mean (something that makes little sense, as the MSE is non-negative), we computed means and standard deviations in the log-MSE domain, which does not have a zero lower-bound, and then mapped back to the linear domain.
Figure 9: A schematic diagram of a stick-person jumping over a puddle. The probability of landing in the puddle, $P_{\text{wet}}$, depends not only on the mean estimate, but also on the uncertainty.

M5 Synaptic Sampling

Here we provide an expanded normative argument for Synaptic Sampling. The argument starts with the observation that to select the correct action, knowing the uncertainty in task relevant quantities is critical [49]. For instance, to decide whether you can jump over a puddle without getting your feet wet, it is important to have not only an estimate of mean landing location, but also the uncertainty in that estimate (Fig. 9). Uncertainty about the landing location comes from two sources, uncertainty about the current state of the world and uncertainty about the target weights (i.e. the weights that would give the best estimate of landing location). To see how the brain might compute uncertainty in landing location, we consider a simplified scenario in which we use $x_{\text{tar} \rightarrow}$ to denote the best possible spike-based representation of the true state of the external world. The neuron’s estimate of landing location is a function of the neuron’s output, $V$, so the optimal estimate of landing location is given by the target output,

$$V_{\text{tar}} = w_{\text{tar} \rightarrow} \cdot x_{\text{tar} \rightarrow} + \text{noise}, \quad (32)$$

where the noise represents the small amount of uncertainty about landing location that remains when $w_{\text{tar} \rightarrow}$ and $x_{\text{tar} \rightarrow}$ are known precisely. Note that the assumption that the synapse combines $w_{\text{tar} \rightarrow}$ and $x_{\text{tar} \rightarrow}$ via a dot product is for simplicity only; the cell could use any nonlinear relationship and our arguments would hold.

Of course, the brain knows neither the target weights, $w_{\text{tar} \rightarrow}$, nor the true state of the external world, $x_{\text{tar} \rightarrow}$. The brain could compute a “best guess” of $x_{\text{tar} \rightarrow}$, and the neuron could use a “best guess” of $w_{\text{tar} \rightarrow}$, resulting in

$$V_{\text{best guess}} = w_{\text{best guess} \rightarrow} \cdot x_{\text{best guess} \rightarrow} + \text{noise}. \quad (33)$$

However, this scheme is unable to give an estimate of uncertainty — so offers little guidance as to whether or not you should jump over the puddle.
To get an estimate of uncertainty, it is necessary to account for uncertainty both in the state of the world, \( x_{\text{tar}} \), and in the relationship between the state of the world and jump distance, parameterised by \( w_{\text{tar}} \). As information about \( x_{\text{tar}} \) comes from sensory data, and information about \( w_{\text{tar}} \) comes from training data (e.g., from past jumps), we can represent our (probabilistic) knowledge about these quantities as two distributions, \( P(x_{\text{tar}}|\text{Sensory Data}) \) and \( P(w_{\text{tar}}|\text{Training Data}) \). To combine these distributions into a distribution over \( V_{\text{tar}} \), we need to integrate over all possible settings of \( x_{\text{tar}} \) and \( w_{\text{tar}} \),

\[
P(V_{\text{tar}}|\text{Sensory Data}, \text{Training Data}) = \int dw_{\text{tar}} dx_{\text{tar}} P(V_{\text{tar}}|x_{\text{tar}}, w_{\text{tar}}) P(x_{\text{tar}}|\text{Sensory Data}) P(w_{\text{tar}}|\text{Training Data}).
\]

It is difficult for neurons to compute this distribution directly (as that would involve a complicated high-dimensional integral). However, by combining neural and synaptic sampling, it is possible for neural circuits to evaluate the integral via sampling; that is, by drawing samples, \( V \), from the distribution,

\[
V \sim P(V_{\text{tar}}|\text{Sensory Data}, \text{Training Data}).
\]

To do that, we simply need to set neural activity, \( x \), to a pattern that represents a plausible state of the world,

\[
x \sim P(x_{\text{tar}}|\text{Sensory Data}),
\]

(this is known as the neural sampling hypothesis [20, 21, 22]), and set the synaptic weights, \( w \), to values that represent a plausible setting for the value of the target weights (this is our hypothesis, Synaptic Sampling),

\[
w \sim P(w_{\text{tar}}|\text{Training Data}).
\]

A sample of landing location is given by combining the sampled inputs and the sampled weights, which could be done by a single neuron,

\[
V = w \cdot x + \text{noise}.
\]

Thus, simply by drawing repeated samples, a single neuron can estimate uncertainty about \( V \), and thus about landing location.

Our argument appears to assume that the brain uses the output of a single neuron to make predictions. This is not too implausible — the cerebellum does contain a large number of Purkinje cells [50] that are believed to use supervised learning to, among other things, make predictions (though perhaps not about landing location). However, it is certainly possible that such a computation is performed by a large multi-layer network. As long as that network is effectively feedforward, we can still, by the logic described above, estimate its uncertainty by combining synaptic sampling with the sampling hypothesis.
M6  Firing rate data

To obtain the $p$-value for Fig. 7A, we performed standard linear regression: we regressed log(variance/mean) against log(firing rate) and log(mean); the former to test our prediction and the latter to eliminate the PSP amplitude as a possible confound. To estimate the firing rate, we took the mean of a FOOPSI-based firing rate estimate [12] computed by the authors of [11]. This estimate is proportional to the true firing rate, with a constant of proportionality that differs from one [13]; because our predicted relationship was linear on a log-log plot, the constant of proportionality plays no role. Using this approach, the best fit line was statistically significantly different from zero ($p < 0.003$), and its slope, $-0.62$ was not significantly different from our prediction, $-1/2$ ($p = 0.57$).

However, there are multiple ways to estimate the firing rate from Calcium traces, and it is not clear a-priori which is most sensible. Thus, we also tried estimating the firing rate using the number of times the FOOPSI signal was above a threshold of 0.01 (we checked that this was a sensible threshold by plotting histograms of the FOOPSI signal). This approach also gave a significant slope ($p < 0.008$, and the best fit-line, which had a slope of $-1.05$, was not significantly different from our prediction of $-1/2$ ($p = 0.16$).

References

[1] T. Poggio, “A theory of how the brain might work,” in Cold Spring Harbor Symposia on Quantitative Biology, vol. 55, Cold Spring Harbor Laboratory Press, 1990.

[2] D. C. Knill and W. Richards, Perception as Bayesian Inference. Cambridge University Press, 1996.

[3] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, “Probabilistic brains: knowns and unknowns,” Nature Neuroscience, vol. 16, no. 9, pp. 1170–1178, 2013.

[4] S. J. Tripathy, S. D. Burton, M. Geramita, R. C. Gerkin, and N. N. Urban, “Brain-wide analysis of electrophysiological diversity yields novel categorization of mammalian neuron types,” Journal of Neurophysiology, vol. 113, no. 10, pp. 3474–3489, 2015.

[5] B. Widrow and M. E. Hoff, “Adaptive switching circuits.,” 1960.

[6] J. S. Albus, “A theory of cerebellar function,” Mathematical Biosciences, vol. 10, no. 1, pp. 25–61, 1971.

[7] M. Ito, M. Sakurai, and P. Tongroach, “Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells,” Journal of Physiology, vol. 324, no. 1, pp. 113–134, 1982.
[8] J. Eccles, R. Llinas, and K. Sasaki, “The excitatory synaptic action of climbing fibres on the purkinje cells of the cerebellum,” Journal of Physiology, vol. 182, no. 2, pp. 268–296, 1966.

[9] P. Dayan and L. F. Abbott, Theoretical Neuroscience. Cambridge, MA: MIT Press, 2001.

[10] T. Branco and K. Staras, “The probability of neurotransmitter release: variability and feedback control at single synapses,” Nature Reviews Neuroscience, vol. 10, no. 5, pp. 373–383, 2009.

[11] H. Ko, L. Cossell, C. Baragli, J. Antolik, C. Clopath, S. B. Hofer, and T. D. Mrsic-Flogel, “The emergence of functional microcircuits in visual cortex,” Nature, vol. 496, no. 7443, pp. 96–100, 2013.

[12] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, “Fast nonnegative deconvolution for spike train inference from population calcium imaging,” Journal of Neurophysiology, vol. 104, no. 6, pp. 3691–3704, 2010.

[13] A. M. Packer, L. E. Russell, H. W. P. Dalgleish, and M. Häusser, “Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo.,” Nature Methods, vol. 12, no. 2, pp. 140–146, 2015.

[14] G.-Q. Bi and M.-M. Poo, “Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type,” Journal of Neuroscience, vol. 18, no. 24, pp. 10464–10472, 1998.

[15] L. F. Abbott and S. B. Nelson, “Synaptic plasticity: taming the beast.,” Nature Neuroscience, vol. 3 Suppl, pp. 1178–1183, 2000.

[16] G. G. Turrigiano and S. B. Nelson, “Homeostatic plasticity in the developing nervous system.,” Nature Reviews Neuroscience, vol. 5, no. 2, pp. 97–107, 2004.

[17] J.-P. Pfister and W. Gerstner, “Triplets of spikes in a model of spike timing-dependent plasticity.,” Journal of Neuroscience, vol. 26, no. 38, pp. 9673–9682, 2006.

[18] R. Ponte Costa, R. C. Froemke, P. J. Sjöström, and M. C. W. van Rossum, “Unified pre- and postsynaptic long-term plasticity enables reliable and flexible learning.,” eLife, vol. 4, 2015.

[19] L. Ziegler, F. Zenke, D. B. Kastner, and W. Gerstner, “Synaptic consolidation: from synapses to behavioral modeling.,” Journal of Neuroscience, vol. 35, no. 3, pp. 1319–1334, 2015.
[20] P. O. Hoyer and A. Hyvarinen, “Interpreting neural response variability as Monte Carlo sampling of the posterior,” Advances in Neural Information Processing Systems, pp. 293–300, 2003.

[21] J. Fiser, P. Berkes, G. Orbán, and M. Lengyel, “Statistically optimal perception and learning: from behavior to neural representations,” Trends in Cognitive Sciences, vol. 14, no. 3, pp. 119–130, 2010.

[22] P. Berkes, J. Fiser, G. Orbán, and M. Lengyel, “Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment,” Science, vol. 331, no. 6013, pp. 83–87, 2011.

[23] G. Orbán, P. Berkes, J. Fiser, and M. Lengyel, “Neural variability and sampling-based probabilistic representations in the visual cortex,” Neuron, vol. 92, no. 2, pp. 530–543, 2016.

[24] R. M. Haefner, P. Berkes, and J. Fiser, “Perceptual decision-making as probabilistic inference by neural sampling,” Neuron, vol. 90, no. 3, pp. 649–660, 2016.

[25] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, “Bayesian inference with probabilistic population codes,” Nature Neuroscience, vol. 9, no. 11, pp. 1432–1438, 2006.

[26] W. L. Buntine and A. S. Weigend, “Bayesian backpropagation,” Complex systems, vol. 5, no. 6, pp. 603–643, 1991.

[27] D. J. MacKay, “A practical Bayesian framework for backpropagation networks,” Neural Computation, vol. 4, no. 3, pp. 448–472, 1992.

[28] C. Blundell, J. Cornebise, K. Kavukcuoglu, and W. Dean, “Weight uncertainty in neural networks,” arXiv:1505.05424 [stat.ML], 2015.

[29] D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass, “Network plasticity as bayesian inference,” PLoS computational biology, vol. 11, no. 11, p. e1004485, 2015.

[30] H. Kasai, N. Takahashi, and H. Tokumaru, “Distinct initial snare configurations underlying the diversity of exocytosis,” Physiological Reviews, vol. 92, pp. 1915–1964, 2012.

[31] T. C. Südhof, “The presynaptic active zone,” Neuron, vol. 75, pp. 11–25, 2012.

[32] K. Michel, J. A. Müller, A.-M. Oprisoreanu, and S. Schoch, “The presynaptic active zone: A dynamic scaffold that regulates synaptic efficacy,” Experimental Cell Research, vol. 335, pp. 157–164, 2015.
[33] U. Frey and R. G. Morris, “Synaptic tagging and long-term potentiation.,” Nature, vol. 385, no. 6616, pp. 533–536, 1997.

[34] R. L. Redondo and R. G. M. Morris, “Making memories last: the synaptic tagging and capture hypothesis.,” Nature Reviews Neuroscience, vol. 12, no. 1, pp. 17–30, 2011.

[35] T. Rogerson, D. J. Cai, A. Frank, Y. Sano, J. Shobe, M. F. Lopez-Aranda, and A. J. Silva, “Synaptic tagging during memory allocation.,” Nature Reviews Neuroscience, vol. 15, no. 3, pp. 157–169, 2014.

[36] W. C. Abraham and M. F. Bear, “Metaplasticity: the plasticity of synaptic plasticity.,” Trends in Neurosciences, no. 4, pp. 126–130, 1996.

[37] W. C. Abraham, “Metaplasticity: tuning synapses and networks for plasticity.,” Nature Reviews Neuroscience, vol. 9, p. 387, May 2008.

[38] S. R. Hulme, O. D. Jones, C. R. Raymond, P. Sah, and W. C. Abraham, “Mechanisms of heterosynaptic metaplasticity.,” Philosophical transactions of the Royal Society of London. Series B, Biological sciences, vol. 369, no. 1633, p. 20130148, 2014.

[39] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii, “Highly nonrandom features of synaptic connectivity in local cortical circuits.,” PLoS Biology, vol. 3, no. 3, p. e68, 2005.

[40] Y. Loewenstein, A. Kuras, and S. Rumpel, “Multiplicative Dynamics Underlie the Emergence of the Log-Normal Distribution of Spine Sizes in the Neocortex In Vivo,” Journal of Neuroscience, vol. 31, pp. 9481–9488, June 2011.

[41] M. Matsuzaki, N. Honkura, G. C. Ellis-Davies, and H. Kasai, “Structural basis of long-term potentiation in single dendritic spines,” Nature, vol. 429, no. 6993, pp. 761–766, 2004.

[42] D. H. O’Connor, S. P. Peron, D. Huber, and K. Svoboda, “Neural activity in barrel cortex underlying vibrissa-based object localization in mice,” Neuron, vol. 67, no. 6, pp. 1048–1061, 2010.

[43] K. Mizuseki and G. Buzsáki, “Preconfigured, Skewed Distribution of Firing Rates in the Hippocampus and Entorhinal Cortex,” Cell Reports, vol. 4, no. 5, pp. 1010–1021, 2013.

[44] N. Intrator and L. N. Cooper, “Objective Function Formulation of the BCM Theory of Visual Cortical Plasticity: Statistical Connections, Stability Conditions,” Neural Networks, vol. 5, pp. 3–17, 1992.
[45] T. Binzegger, R. Douglas, and K. Martin, “A quantitative map of the circuit of cat primary visual cortex.,” J. Neurosci., vol. 24, pp. 8441–8453, 2004.

[46] H. L. Bryant and J. P. Segundo, “Spike initiation by transmembrane current: a white-noise analysis.,” Journal of Physiology, vol. 260, no. 2, pp. 279–314, 1976.

[47] Z. F. Mainen and T. J. Sejnowski, “Reliability of spike timing in neocortical neurons.,” Science, vol. 268, no. 5216, pp. 1503–1506, 1995.

[48] T. P. Minka, A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology, 2001.

[49] M. O. Ernst and M. S. Banks, “Humans integrate visual and haptic information in a statistically optimal fashion,” Nature, vol. 415, no. 6870, pp. 429–433, 2002.

[50] P. Dean, J. Porrill, C.-F. Ekerot, and H. Jörntell, “The cerebellar microcircuit as an adaptive filter: experimental and computational evidence,” Nature Reviews Neuroscience, vol. 11, pp. 30–43, Jan. 2010.
Supplementary Information

Here we give detailed derivations for our learning rules and predictions. In Sec. S1 we derive Bayesian learning rules for supervised and reinforcement learning, for which a feedback signal is present, including the simplified learning rules used in Eq. (5) of the main text; in Sec. S2 we derive Bayesian learning rules for unsupervised learning. We then discuss how to set $s^2$ (Sec. S3) and consider how to relate firing rates to uncertainty (Sec. S4). Finally, we provide a detailed description of how we set model parameters (Sec. S5).

S1 Learning rules with feedback

We begin by considering the standard classical learning rules that we use for comparison with our Bayesian learning rules; we then move on to the derivation of the Bayesian learning rules themselves.

S1.1 Classical learning rules

To make comparisons in Fig. 4, we need to specify classical learning rules for each type of learning. Each classical rule has a learning rate, $\alpha$, which is allowed to vary.

For supervised learning with continuous feedback and for supervised learning with binary feedback, we use the delta rule, (Eq. (4)) [1]. The delta rule is suitable for binary feedback because we set the threshold, $\theta$, to 0, so the proportion of positive and negative increments is the same.

For reinforcement learning, we use a standard policy gradient method [2],

$$\Delta w_i = -\alpha x_i (f - \langle f \rangle) (w_i - m_i).$$  \hspace{1cm} (S.1)

We compute the expected loss, $\langle f \rangle$ is over past trials, using an exponential moving average. To implement this moving average, on each timestep we updated $\langle f \rangle$ via,

$$\Delta \langle f \rangle = \alpha_{\text{reward}} (f - \langle f \rangle).$$  \hspace{1cm} (S.2)

A sweep across different settings of $\alpha_{\text{reward}}$ (Fig. S.1) indicated that a sensible value was $10^{-5}$. However, the precise value is not so critical, as the mean squared error was relatively flat over a broad range.

S1.2 Bayesian learning rules

Here we derive the update rules for the mean and variance, $\mu_i$ and $\sigma^2_i$. We begin with the difficult part: incorporating new data using Bayes theorem, Eq. (26). It is convenient to
Figure S.1: The mean squared error relative to the Bayesian learning rules (as in Fig. 4) for classical reinforcement learning rules with different settings of $\alpha_{\text{reward}}$ (on the $x$-axis) and with different settings for the learning rate, $\alpha$ (blue lines). The red line is set at a relative MSE of 1. While the relative MSE does not change much with $\alpha_{\text{reward}}$, it does seem that values are more reliable between $10^{-4}$ to $10^{-6}$, as we might expect given that the time constant in this simulation is $10^5$. We thus chose $\alpha_{\text{reward}} = 10^{-5}$ for Fig. 4. We do not see much change in the relative MSE as we change $\alpha_{\text{reward}}$, because the method asymptotically finds the correct weights even if $E[f]$ is not set correctly; setting $E[f]$ correctly merely minimises variance in the weight updates.
write the update rule as an integral over the prediction error, $\delta$,

$$ P(\lambda_{\text{tar},i}|D_i) = \int d\delta P(\lambda_{\text{tar},i}|\delta, d'_i, D_i(t-1)) P(\delta|f, d'_i) $$ \hfill (S.3)

where $d'_i$ is all the data except the feedback signal (see Methods, Eq. (24)),

$$ d'_i = (x_i, V, w_i) \tmspace{1em} (S.4) $$

We have not conditioned on $D_i(t-1)$ in the last term in Eq. (S.3) because $\delta$ is independent of past data, and, recall, quantities without an explicit time dependence should be evaluated at time $t$. This approach makes it considerably easier to generalise across feedback signals, as $P(\lambda_{\text{tar},i}|\delta, d'_i, D_i(t-1))$ is the same across all feedback signals; only $P(\delta|f, d'_i)$ differs.

We start by considering how to infer $\lambda_{\text{tar},i}$ from $\delta$ (i.e., how to compute the first term in the integral in Eq. (S.3)). As usual, we use Bayes theorem,

$$ P(\lambda_{\text{tar},i}|\delta, d'_i, D_i(t-1)) \propto P(\lambda_{\text{tar},i}|D_i) P(\delta|\lambda_{\text{tar},i}, d'_i) P(D_i|\lambda_{\text{tar},i}) P(\lambda_{\text{tar},i}|D_i(t-1)) \tmspace{1em} (S.5) $$

This is the analog of Eq. (26); the only difference is that $d_i$ in that equation has been replaced by $(\delta, d'_i)$, and we have performed a small amount of algebra. The second term, $P(d'_i|\lambda_{\text{tar},i})$, can be neglected as it is independent of $\lambda_{\text{tar},i}$ (without a feedback signal, $d'_i$ tells us nothing about the target weight). The last term, the prior, $P(\lambda_{\text{tar},i}|D_i(t-1))$, is given by the approximating Gaussian distribution from the previous time-step (Eq. (28)). We will obtain the constant of proportionality in Eq. (S.5) automatically, when we identify the distribution as a Gaussian.

To find an expression for the first term in Eq. (S.5), the likelihood, $P(\delta|\lambda_{\text{tar},i}, d'_i)$, we note that $\delta$ is the sum of a large number of independent terms, and so, via the central limit theorem, it is Gaussian. Its mean is given by

$$ E[\delta|\lambda_{\text{tar},i}, d'_i] = E[V_{\text{tar}} - V|\lambda_{\text{tar},i}, w_i] = \sum_j x_j E[w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i] $$ \hfill (S.6)

where the second expression follows from Eqs. (1) and (2). To evaluate the expectation, we note that for $j \neq i$, $E[w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i] = 0$, leaving only the ith term. Using also the fact that $w_{\text{tar},i} = \pm e^{\lambda_{\text{tar},i}}$ (positive if $w_{\text{tar},i}$ is an excitatory weight and negative if it is inhibitory), we have

$$ E[\delta|\lambda_{\text{tar},i}, d'_i] = x_i (\pm e^{\lambda_{\text{tar},i}} - w_i) \tmspace{1em} (S.7) $$

Next we compute the variance of $\delta$. If we assume that all the inputs, $x$, are known (we relax this assumption shortly), then

$$ \text{Var} [\delta|\lambda_{\text{tar},i}, V, w_i, x] = \gamma_\delta^2 + \text{Var} [V_{\text{tar}} - V|\lambda_{\text{tar},i}, w_i] \tmspace{1em} (S.8) $$

$$ = \gamma_\delta^2 + \gamma_V^2 + \sum_j \text{Var} [w_{\text{tar},j} - w_j|\lambda_{\text{tar},i}, w_i] x_j^2. $$
where again the second expression followed from Eqs. (1) and (2). Noting that the variance of $w_{\text{tar},j}$ is $s_j^2$ (Eq. (29b)), and that the noise variance in $w_j$ is $k_j m_j$ (Eq. (11)), this becomes,

$$\text{Var} \left[ \delta|\lambda_{\text{tar},i}, d'_i, x_i \right] = s^2_\delta - (s^2_i + k_j m_j) x^2_i \quad \text{(S.9)}$$

where

$$s^2_\delta \equiv \gamma^2_\delta + \sum_j (s^2_j + k_j m_j) x^2_j. \quad \text{(S.10)}$$

Because all the dependence on the $x_i$ is through $s^2_\delta$, we can relax the assumption that all the $x_i$ are known. Instead the synapse only needs to know $s^2_\delta$ for its distribution over $\delta$ to be Gaussian,

$$P(\delta|\lambda_{\text{tar},i}, d'_i, s^2_\delta) = \mathcal{N}(\delta; x_i (\pm e^{\lambda_{\text{tar},i}} - w_i), s^2_\delta - x^2_i (s^2_i + k_i m_i)) \quad \text{(S.11)}$$

Of course, the synapse cannot know $s^2_\delta$, as that involves a summation over all the inputs at every time step. Instead, we use an approximate value based on the average (see Sec. (S3)).

Because of the non-linearity, $e^{\lambda_{\text{tar},i}}$, this is a complicated function of $\lambda_{\text{tar},i}$. We can linearise the problematic term using statistical linearisation [3]. This involves finding the straight line that minimizes the expected squared error between the curve and a straight line,

$$0 = \frac{\partial}{\partial a} \mathbb{E} \left[ (\pm e^{\lambda_{\text{tar},i}} - (a(\lambda_{\text{tar},i} - \mu) + b))^2 \right] \quad \text{(S.12)}$$

$$0 = \frac{\partial}{\partial b} \mathbb{E} \left[ (\pm e^{\lambda_{\text{tar},i}} - (a(\lambda_{\text{tar},i} - \mu) + b))^2 \right], \quad \text{(S.13)}$$

where the expectation is taken under the prior ($P(\lambda_{\text{tar},i}|D(t-1))$). The solution is $a = b = m_i$ (note that $m_i$ is a signed quantity), which gives,

$$\pm e^{\lambda_{\text{tar},i}} \approx m_i \left(1 + \lambda_{\text{tar},i} - \mu_i\right). \quad \text{(S.14)}$$

Inserting Eq. (S.14) into Eq. (S.11), the likelihood becomes,

$$P(\delta|\lambda_{\text{tar},i}, d'_i, s^2_\delta) = \exp \left( -\frac{(\delta - x_i (m_i (\lambda_{\text{tar},i} - \mu_i) - (w_i - m_i)))^2}{2(s^2_\delta - (s^2_i + k_i m_i) x^2_i)} \right) \quad \text{(S.15)}$$

which is Gaussian in $\lambda_{\text{tar},i}$.

Examining Eq. (S.5) and noting, as discussed immediately after that equation, that the second term on the right hand side is independent of $\lambda_{\text{tar},i}$, we see that to compute the posterior we just need to multiply the likelihood, Eq. (S.15), by $P(\lambda_{\text{tar},i}|D_i(t-1))$. The latter distribution is also Gaussian in $\lambda_{\text{tar},i}$ (Methods, Eq. (28)); consequently, their
product is Gaussian. Straightforward, but somewhat tedious, algebra gives us their mean
and variance,

\[
E[\lambda_{\text{tar},i}|\delta, s_\delta^2, d_i', D_i(t-1)] = \mu_i + (\delta + x_i (w_i - m_i)) \frac{x_i m_i \sigma_i^2}{s_{\delta,i}^2}
\] (S.16a)

\[
\text{Var}[\lambda_{\text{tar},i}|\delta, s_\delta^2, d_i', D_i(t-1)] = \sigma_i^2 \left( 1 - \frac{\sigma_i^2 x_i^2 m_i^2}{s_{\delta,i}^2} \right)
\] (S.16b)

where

\[
s_{\delta,i}^2 \equiv s_\delta^2 - k_i m_i x_i^2 - (s_i^2 - m_i^2 \sigma_i^2) x_i^2 \approx s_\delta^2 - k_i m_i x_i^2.
\] (S.17)

The approximation is valid so long as \( \sigma_i^2 \ll 1 \) (see Methods, Eq. (29b)).

The next step is to substitute \( P(\lambda_{\text{tar},i}|\delta, d_i', D_i(t-1)) \) (which is, to reiterate, Gaussian, with mean and variance given by Eq. (S.16)) back into Eq. (S.3) and perform the integral over \( \delta \). Once we do that, we need to take into account changes to the optimal weight across time (Methods, Eq. (13)), and then bring the resulting distribution back into the log normal class (Methods, Eq. (28)), by computing the mean and variance of \( \lambda_{\text{tar},i} \). Fortunately, as is not hard to show, the above two steps commute: we can compute the mean and variance of \( \lambda_{\text{tar},i} \) first, and then take into account changes in the optimal weight across time. As is also straightforward to show, the mean and variance are given by

\[
E[\lambda_{\text{tar},i}|D_i] = \mu_i + (E[\delta|d_i] + x_i (w_i - m_i)) \frac{x_i m_i \sigma_i^2}{s_{\delta,i}^2}
\] (S.18a)

\[
\text{Var}[\lambda_{\text{tar},i}|D_i] = \sigma_i^2 - \frac{\text{Var}[\delta|d_i] \sigma_i^2 x_i^2 m_i^2}{s_{\delta,i}^2},
\] (S.18b)

where the expectation and variance are with respect to \( P(\delta|d_i) \), and, recall, \( d_i \) now includes

the feedback signal, \( f \) (see Eq. (24)).

To account for the random changes in weights between time steps we use Eq. (14),

\[
\mu_i(t+1) = \left( 1 - \frac{1}{\tau} \right) E[\lambda_{\text{tar},i}|D_i] + \frac{\mu_{\text{prior}}}{\tau}
\] (S.19a)

\[
\sigma_i^2(t+1) = \left( 1 - \frac{1}{\tau} \right)^2 \text{Var}[\lambda_{\text{tar},i}|D_i] + \frac{2\sigma_{\text{prior}}^2}{\tau}.
\] (S.19b)

Substituting Eq. (S.18) into Eq. (S.19), and using the fact that the updates to the mean and uncertainty are small on each time step,

\[
|E[\lambda_{\text{tar},i}|D_i] - \mu_i| \ll \mu_i \hspace{1cm} (S.20a)
\]

\[
|\text{Var}[\lambda_{\text{tar},i}|D_i] - \sigma_i^2| \ll \sigma_i^2, \hspace{1cm} (S.20b)
\]
and also using the fact that $\tau \gg 1$, we have

$$
\Delta \mu_i = \left( \frac{m_i \sigma_i^2}{s_{\delta,i}^2} \right) x_i \left( E[\delta | d_i] + x_i (w_i - m_i) \right) - \frac{1}{\tau} (\mu_i - \mu_{\text{prior}}), \tag{S.21a}
$$

$$
\Delta \sigma_i^2 = - \left( \frac{\sigma_i^4 m_i^2}{s_{\delta,i}^2} \right) x_i^2 \left( \frac{s_{\delta,i}^2 - \text{Var}[\delta | d_i]}{s_{\delta}^2} \right) - \frac{2}{\tau} (\sigma_i^2 - \sigma_{\text{prior}}^2). \tag{S.21b}
$$

Finally, to compute the mean and variance of $\delta$ conditioned on the data, $d_i$, we need to compute $P(\delta | d_i)$. We again use Bayes’ theorem,

$$
P(\delta | d_i) = P(\delta | f, x_i, w_i) \propto P(f | \delta) P(\delta | x_i, w_i) \tag{S.22}
$$

where the prior is given by multiplying the right hand side of Eq. (S.15) by $P(\lambda_{\text{tar},i} | D_i (t - 1))$ (which is Gaussian in $\lambda_{\text{tar},i}$; Methods, Eq. (28)), and integrating over $\lambda_{\text{tar},i}$; this leads to

$$
P(\delta | x_i, w_i) = N \left( \delta; -x_i (w_i - m_i), s_{\delta,i}^2 \right). \tag{S.23}
$$

The likelihood, $P(f | \delta)$, is specific to the feedback signal, and hence to the type of learning, as described below. For supervised learning with continuous feedback, the likelihood is a delta function,

$$
P(f | \delta) = \delta(\delta - f), \tag{S.24}
$$

so the posterior over $\delta$ (Eq. (S.22)) is a delta function located at $f$.

For supervised learning with binary feedback, the likelihood is a step function,

$$
P(f = 1 | \delta) = \Theta (\delta - \theta) \tag{S.25a}
$$

$$
P(f = -1 | \delta) = 1 - \Theta (\delta - \theta) \tag{S.25b}
$$

so the posterior over $\delta$ (Eq. (S.22)) is a truncated Gaussian, whose mean and variance can be computed in terms of the cumulative Normal function. We do not reproduce the expressions here, because they are not very illuminating.

For reinforcement learning, the likelihood is

$$
P(f | \delta) = \delta (f + |\delta|) \tag{S.26}
$$

so the posterior over $\delta$ (Eq. (S.22)) is a pair of delta-functions, with different weights, whose mean and variance are easy to compute. Again we do not reproduce those expressions because they are not very illuminating.
S1.3 Simplifying the learning rules

While we used the full equations in simulation (Eq. (S.21)), for illustrative purposes we presented simplified learning rules in the main text (Eq. (5)), valid for continuous feedback, \( f = \delta \). These simplifications involve rather severe approximations; we make them so that we can illustrate the essence of the learning rules in the simplest possible setting. We do not, though, use them in any of our simulations.

Using the expressions for \( m_i \) given in Eq. (29a), and assuming updates are small, we have, to first order in the updates,

\[
\Delta m_i = m_i \left( \Delta \mu_i + \frac{1}{2} \Delta \sigma_i^2 \right) \tag{S.27}
\]

Using the fact that \( \sigma_i^2 \) is small compared to \( m_i^2 \) (Methods, Eq. (29) and surrounding text), and assuming that the relative updates to the mean and uncertainty, \( \Delta \mu_i / \mu_i \) and \( \Delta \sigma_i^2 / \sigma_i^2 \), are about the same size, we may approximate this with the first term,

\[
\Delta m_i \approx m_i \Delta \mu_i. \tag{S.28}
\]

Using the approximate expression for \( s_i^2 \) given in Eq. (29b), and applying the same reasoning as above, we arrive at an approximate update rule for \( s_i^2 \),

\[
\Delta s_i^2 \approx m_i^2 \Delta \sigma_i^2. \tag{S.29}
\]

Inserting these approximate expressions for \( \Delta m_i \) and \( \Delta s_i^2 \) into Eq. (S.21), noting that for continuous feedback the mean of \( \delta \) is \( \delta \) and the variance is zero, again using the approximation \( s_i^2 \approx \sigma_i^2 m_i^2 \) (Eq. (29b)), and neglecting the term \( w_i - m_i \) in Eq. (S.21a), we have

\[
\Delta m_i \approx \left( s_i^2 \right) \frac{1}{s_{\delta,i}^2} x_i \delta - \frac{m_i}{\tau} \left( \mu_i - \mu_{\text{prior}} \right), \tag{S.30a}
\]

\[
\Delta \sigma_i^2 \approx - \left( s_i^2 \right) \frac{1}{s_{\delta,i}^2} x_i^2 s_i^2 - \frac{2m_i^2}{\tau} \left( \sigma_i^2 - \sigma_{\text{prior}}^2 \right). \tag{S.30b}
\]

To show that the decay term for the mean is approximately the form given in the main text (Eq. (5a)) we use Eq. (29a) to write

\[
m_i - m_{\text{prior}} = m_i \left( 1 - e^{-(\mu_i - \mu_{\text{prior}})} - \frac{1}{2} (\sigma_i^2 - \sigma_{\text{prior}}^2) \right). \tag{S.31}
\]

Taylor expanding and neglecting both \( \sigma_i^2 \) and \( \sigma_{\text{prior}}^2 \), we arrive at

\[
m_i - m_{\text{prior}} \approx m_i \left( \mu_i - \mu_{\text{prior}} \right). \tag{S.32}
\]

To show that the decay term for the variance is in approximately the form given in the main text (Eq. (5b)), we use our standard approximation for the variance,

\[
s_i^2 - s_{\text{prior}}^2 \approx \sigma_i^2 m_i^2 - \sigma_{\text{prior}}^2 m_{\text{prior}}^2. \tag{S.33}
\]

As \( E [m_i] = m_{\text{prior}} \), we replace \( m_{\text{prior}}^2 \) with \( m_i \) to give the required result.
S2 Bayesian learning rules without feedback

We begin by deriving classical learning rules, which will give some results and intuition that will prove useful for Bayesian learning.

S2.1 Classical learning rules

For unsupervised learning we use a maximum-likelihood learning rule. For maximum likelihood, there is no notion of separate target weights or membrane potential, so we let \( w_{\text{tar}} \rightarrow w \) and \( V_{\text{tar}} \rightarrow V \). We use the generative model defined in Methods, Sec. M1.5, wherein \( V \) is drawn from a Laplacian (Eq. (21)), and \( x \) depends on \( V \) through Eq. (20). The objective is to alter \( w \) so as to maximize the marginal likelihood, \( P(x|w) \), which is given by integrating out the latent variable, \( V \),

\[
P(x|w) = \int dV P(V) P(x|V, w). \tag{S.34}
\]

The un-normalized version of the distribution \( P(x|V, w) \) is given in Eq. (20). To perform the integral over \( V \) above we need the normalizer, which depends on \( V \),

\[
Z(V) = \int dV \frac{e^{-x^T \Lambda^{-1} x/2}}{\text{Det}(2\pi\Lambda)^{1/2}} \delta(V - w^T x) \tag{S.35}
\]

where \( \text{Det} \) denotes determinant. Using the Fourier transform representation of the delta-function, this becomes

\[
Z(V) = \int \frac{dq}{2\pi} e^{-qV} \int dx \frac{e^{-x^T \Lambda^{-1} x/2 + iqw^T x}}{\text{Det}(2\pi\Lambda)^{1/2}} \tag{S.36}
\]

The integrals over \( x \) and \( q \) are both Gaussian, and therefore straightforward, yielding

\[
Z(V) = e^{-V^2/2w^T \Lambda w} / (2\pi w^T \Lambda w)^{1/2}. \tag{S.37}
\]

The integral in Eq. (S.34) is now straightforward. Using Eq. (21) for \( P(V) \), we arrive at

\[
P(x|w) = \frac{e^{-x^T \Lambda^{-1} x/2}}{\text{Det}(2\pi\Lambda)^{1/2}} e^{-(w^T x)/b} (2\pi w^T \Lambda w)^{1/2} e^{(w^T x)^2/2w^T \Lambda w}. \tag{S.38}
\]

The gradient of the log-likelihood is, therefore, given by

\[
\frac{\partial \log P(x)}{\partial w} = \frac{\partial}{\partial w} \left[ -\frac{|w^T x|}{b} + \log w^T \Lambda w + \frac{(w^T x)^2}{2w^T \Lambda w} \right] \tag{S.39}
\]

\[
= -\text{sign}(w^T x) \frac{x}{b} + \frac{\Lambda w}{w^T \Lambda w} + \frac{w^T xx}{w^T \Lambda w} - \frac{(w^T x)^2 \Lambda w}{(w^T \Lambda w)^2}.
\]
Using $E \left[ (w^T x)^2 \right] = w^T E [xx] w = w^T \Lambda w$ (see Methods, Eqs. (20) and following text), we see that on average the second and fourth terms cancel. Taking that into account and, in a slight abuse of notation replacing $w^T x$ with $V$, we arrive at

$$\frac{\partial \log P(x)}{\partial w} \approx -\frac{\text{sign}(V) x}{b} + \frac{V x}{w^T \Lambda w}.$$  

(S.40)

As expected, this learning rule has a classic Hebbian form: increase the weight when $V$ is large, and decrease the weight when $V$ is small.

S2.2 Bayesian inference

For the Bayesian learning rule, we take exactly the same approach as previously (i.e. using Eq. (S.3)). Just as for the previous learning rules, all we need to do is compute the moments of the posterior distribution over $\delta$, and insert them into the learning rules (Eq. S.21). In unsupervised learning, the posterior over $\delta$ simplifies considerably, as we do not have a feedback signal, and we throw away information about $w_i, x_i$,

$$P(\delta|f, d_i) = P(\delta|d_i) \approx P(\delta|V).$$  

(S.41)

For unsupervised learning, it turns out to be easier to work in terms of $V_{\text{tar}}$ rather than $\delta$. As $V_{\text{tar}}$ and $\delta$ are related very simply (Eq. (3)) and $V$ is known, computing the moments of $\delta$ from the moments of $V_{\text{tar}}$, is trivial (we have neglected $\gamma_2^2$ for simplicity),

$$E[\delta|V] = E[V_{\text{tar}}|V] - V;$$

(S.42a)

$$\text{Var}[\delta|V] = \text{Var}[V_{\text{tar}}|V].$$

(S.42b)

To compute $P(V_{\text{tar}}|V)$, we use Bayes theorem,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}}) P(V|V_{\text{tar}})$$  

(S.43)

and introduce and integrate out other quantities that appear in the generative model,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}}) \int dx dw_{\text{tar}} P(V|x) P(x|V_{\text{tar}}, w_{\text{tar}}) P(w_{\text{tar}}).$$  

(S.44)

To compute $P(x|V_{\text{tar}}, w_{\text{tar}})$ we combine the $x$ dependence of $P(x|V_{\text{tar}}, w_{\text{tar}})$ (Eq. 20) with the normalizer (Eq. S.37), and noting that the normalizer can be rewritten as a Gaussian, which gives,

$$P(x|V_{\text{tar}}, w_{\text{tar}}) = \mathcal{N}(x; 0, \Lambda) \delta (V_{\text{tar}} - w_{\text{tar}}^T x) \mathcal{N}(V_{\text{tar}}; 0, w_{\text{tar}}^T \Lambda w_{\text{tar}})^{-1}.$$  

(S.45)

Now we make an approximation; because $\Lambda$ is diagonal, $w_{\text{tar}}^T \Lambda w_{\text{tar}}$ is the sum of a large number of non-negative terms. If those terms were independent, $w_{\text{tar}}^T \Lambda w_{\text{tar}}$ would self-average: its
standard deviation would be much smaller than its mean. Because of $P(V_{\text{tar}}|w_{\text{tar}}, x)$, those terms are not quite independent. However, this term has minimal effect on the variance, so it still self averages. Thus, we can use,

$$w_{\text{tar}}^T \Lambda w_{\text{tar}} \approx \Delta t \sum_i \nu_j e^{2(\mu_j + \sigma_j^2)} \equiv v$$  \hspace{1cm} (S.46)

Substituting this into Eq. (S.43) and writing $\delta (V_{\text{tar}} - w_{\text{tar}}^T x)$ as $P(V_{\text{tar}}|w_{\text{tar}}, x)$, gives,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}}) \mathcal{N}(V_{\text{tar}}; 0, v)^{-1} Q(V, V_{\text{tar}})$$ \hspace{1cm} (S.47)

where

$$Q(V, V_{\text{tar}}) = \int dx d\delta P(V|x) P(V_{\text{tar}}|w_{\text{tar}}, x) P(w_{\text{tar}}) \mathcal{N}(x; 0, \Lambda).$$ \hspace{1cm} (S.48)

Integrating over $w_{\text{tar}}$, we get,

$$Q(V, V_{\text{tar}}) = \int dx P(V|x) P(V_{\text{tar}}|x) \mathcal{N}(x; 0, \Lambda).$$ \hspace{1cm} (S.49)

As $V$ is known and fixed, we only care about the $V_{\text{tar}}$ dependence, and so we can also write,

$$P(V_{\text{tar}}|V) \propto P(V_{\text{tar}}) \mathcal{N}(V_{\text{tar}}; 0, v)^{-1} Q(V_{\text{tar}}|V)$$ \hspace{1cm} (S.50)

where

$$Q(V_{\text{tar}}|V) \propto Q(V, V_{\text{tar}})$$ \hspace{1cm} (S.51)

Again, as $V$ is known, instead of computing the distribution over $Q(V_{\text{tar}}|V)$ directly, it is easier to compute $Q(\delta|V) = Q(\delta)$, then convert back. The distribution over $\delta$ is very simple,

$$Q(\delta) = \mathcal{N}(\delta; 0, s_\delta^2)$$ \hspace{1cm} (S.52)

(we compute a closely related quantity in Eq. (S.11)). Given the definition of $\delta$ (Eq. 3), the corresponding distribution over $V_{\text{tar}}$ is

$$Q(V_{\text{tar}}|V) = \mathcal{N}(V_{\text{tar}}; V, s_\delta^2).$$ \hspace{1cm} (S.53)

Thus, we can compute $P(V_{\text{tar}}|V)$ (Eq. (S.50)) by combining two Gaussian distributions, $Q(V_{\text{tar}}|V)$ and $\mathcal{N}(V_{\text{tar}}; 0, v)^{-1}$, with a Laplacian, $P(V_{\text{tar}})$. This gives rise to a mixture of two truncated Gaussian distributions, one for the rising, and one for the decaying part of the Laplacian. Thus, the mean and variance of $P(V_{\text{tar}}|V)$ can straightforwardly (if tediously) be computed – we do not reproduce these expressions here because they are not very enlightening. As described above (Eq. (S.42)), the mean and variance of $P(V_{\text{tar}}|V)$ trivially give the mean and variance of $P(\delta|V)$, which we can be inserted directly into the learning rules (Eq. S.21).
S3 Setting $s^2_δ$

Ideally, $s^2_δ$ (given in Eq. (S.10)) should be updated on every timestep. In reality, of course, this requires a non-local computation that the synapse is unable to perform. Therefore, for supervised and unsupervised learning, we approximate $s^2_δ$ using its average value,

$$E [s^2_δ] = \gamma^2_δ + \gamma^2_V + \sum_j (s^2_j + k_j m_j) \nu_j \Delta t.$$  \hspace{1cm} (S.54)

So long as the firing rates are stationary, this quantity changes slowly. Moreover, $s^2_δ$ is the same for the whole cell, so could be computed by molecular machinery in the cell (e.g. signalling cascades, tagging proteins, etc.)

For reinforcement learning, however, this approximation turns out to not be good enough. Instead we use a better approximation, and exploit the fact that $δ$ tells us, via Bayes' theorem, something about $s^2_δ$,

$$P (s^2_δ | δ) \propto P (δ | s^2_δ) P (s^2_δ).$$  \hspace{1cm} (S.55)

The likelihood, $P (δ | s^2_δ)$, is given by Eq. (S.15), but with all terms in the exponent (except $δ$) replaced by their means,

$$P (δ | s^2_δ) = N (δ; 0, s^2_δ).$$  \hspace{1cm} (S.56)

For analytic tractability, we set the prior, $P (s^2_δ)$, to the appropriate conjugate prior (an Inverse Gamma distribution),

$$P (s^2_δ) = \text{InverseGamma} (s^2_δ; \alpha, \beta) \propto s^{-2(\alpha+1)} e^{-\beta/s^2_δ}.$$  \hspace{1cm} (S.57)

To set $\alpha$ and $\beta$, we match the mean (Eq. (S.54)) and variance of $s^2_δ$, the latter given by

$$\text{Var} [s^2_δ] = \sum_j (s^2_j + k_j m_j) \nu_j \Delta t (1 - \nu_j \Delta t).$$  \hspace{1cm} (S.58)

The mean and variance of an Inverse Gamma distribution are given by,

$$E [s^2_δ] = \frac{\beta}{\alpha - 1}$$  \hspace{1cm} (S.59a)

$$\text{Var} [s^2_δ] = \frac{\beta^2}{(\alpha - 1)^2 (\alpha - 2)}.$$  \hspace{1cm} (S.59b)

Solving for $\alpha$ and $\beta$, we have

$$\alpha = \frac{E [s^2_δ]^2}{\text{Var} [s^2_δ]} + 2$$  \hspace{1cm} (S.60a)

$$\beta = E [s^2_δ] (\alpha - 1).$$  \hspace{1cm} (S.60b)
Substituting the prior and likelihood into Eq. (S.55) gives the posterior,
\[ P\left(s^2_\delta | \delta \right) \propto s_\delta^{-2(\alpha+1/2)+1} e^{-\left(\beta + \delta^2/2\right)/s^2_\delta}. \]  
(S.61)

Comparing to Eq. (S.57), we see that the posterior is another Inverse Gamma distribution (as expected, given that we use a conjugate prior). Finally, we use the posterior mean, as our estimate of \( s^2_\delta \delta \),
\[ E\left[s^2_\delta \delta | \delta \right] = \frac{2\beta + \delta^2}{2\alpha - 1} = \frac{2(\alpha - 1)E\left[s^2_\delta \right] + \delta^2}{2(\alpha - 1) + 1}. \]  
(S.62)

The mean value of \( s^2_\delta \delta \) conditioned on \( \delta \), \( E\left[s^2_\delta \delta | \delta \right] \), is, therefore, a weighted sum of \( E\left[s^2_\delta \right] \) and \( \delta^2 \). Because \( \alpha \) is large (both the mean and variance of \( s^2_\delta \) are proportional to \( n \), the number of synapses, and so both are large; consequently \( \alpha \) is also proportional to \( n \)), that quantity is weighted heavily toward \( E\left[s^2_\delta \right] \). However, the small contribution from \( \delta \) turns out to be important; without it, the mean squared error tends to be very large (data not shown).

**S4 The relationship between variability and firing rate**

We wish to find relationships between the mean and uncertainty, \( m_i \) and \( s^2_i \), and the firing rate, \( \nu_i \). To do so, we take the time average of Eq. (S.21b) in steady state (where \( \langle \Delta \sigma^2_i \rangle = 0 \)),
\[ 0 = \langle x_i \sigma^4_i m^2_i s^2_\delta - \text{Var} \left[ \delta d_i \right] \rangle - \frac{2}{\tau} \left( \sigma^2_{\text{prior}} - \langle \sigma^2_i \rangle \right) \]  
(S.63)

Here and in what follows the angle brackets indicate an average over times that are long enough to average over fluctuations but short compared to \( \tau \), the timescale over which the target weights change. For tractibility, we ignore correlations among the variables; consequently, Eq. (S.63) becomes
\[ 0 = \frac{\sigma^4_i m^2_i \nu_i \Delta t \chi_i}{s^2_\delta} + \frac{2\sigma^2_i}{\tau} - \frac{2\sigma^2_{\text{prior}}}{\tau} \]  
(S.64)

where we have replaced \( \langle x_i \rangle \) with \( \nu_i \Delta t \) and made the definition
\[ \chi_i \equiv \frac{s^2_\delta - \langle \text{Var} \left[ \delta d_i \right] \rangle}{s^2_\delta}. \]  
(S.65)

Solving for \( \sigma^2_i \), we have
\[ \sigma^2_i = \left( \frac{2m^2_i \nu_i \Delta t \chi_i \sigma^2_{\text{prior}}/s^2_\delta \tau + 1/\tau^2 \right)^{1/2} - 1/\tau \]  
(S.66)

In the limit that \( \tau \nu_i \Delta t \gg 1 \), the above expression simplifies considerably,
\[ \sigma^2_i \approx \frac{s_\delta/m_i}{\sqrt{\nu_i \Delta t}} \left( \frac{2\sigma^2_{\text{prior}}}{\tau \chi_i} \right)^{1/2}. \]  
(S.67)
Using the approximation $s^2_i \approx \sigma^2_i m^2_i$, valid so long as $\sigma^2_i \ll 1$ (see Methods, Eq. (29b) and following discussion), we arrive at

$$\frac{s^2_i}{m_i} \approx \frac{s_8}{\nu_i \Delta t} \left( \frac{2\sigma^2_{\text{prior}}}{\tau \chi_i} \right)^{1/2}. \quad (S.68)$$

Assuming the feedback signal typically removes a finite fraction of the prior variance concerning $\delta, \chi$, will be $O(1)$. Thus, because the relative learning rate, $\Delta m_i/m_i$, is proportional to $s^2_i/m_i$ (see main text, Eqs. (5b) and (6)), Eq. (S.68) corroborates our prediction about learning rates via Bayesian Plasticity (main text, Eq. (7)). However, note that the prediction regarding plasticity will not necessarily hold in experiments in which the network does not exhibit ongoing activity. That’s because without ongoing activity, only one input (the stimulated one, say input $i$) is active. In that case, $s^2_\delta \propto s^2_i$ (see Eq. (S.10), and note that the noise is small), and so the learning rate, $\alpha_i$, does not change with $s^2_i$ (Eq. (6)). In contrast, if there are many other inputs active, then there are many other contributions to $s^2_\delta$ (Eq. (S.10)), so $s^2_\delta$ changes little with $s^2_i$. Because in vitro preparations are typically quite, this prediction must be tested in vivo.

### S5 Setting model parameters

To find a sensible timescale for synaptic sampling (i.e., a timescale upon which the uncertainty is similar to the variability) we solve Eq. (S.68) for $\tau$,

$$\tau \sim \frac{2\sigma^2_{\text{prior}}}{\nu_i \Delta t \frac{s^2_i}{m_i} \frac{s^2_8}{s^2_8} \chi_i}. \quad (S.69)$$

where

$$\nu_i \Delta t \frac{s^2_i}{s^2_8} \sim 1/2n \quad (\text{see Eq. (S.10))} \quad (S.70a)$$

$$n = 1000 \quad \text{(Methods, Sec. M1.6)} \quad (S.70b)$$

$$\frac{s^2_i}{m_i} \sim 0.076 \quad \text{(Average value from [4])} \quad (S.70c)$$

$$\sigma^2_{\text{prior}} \sim 0.86 \quad \text{(from [4])}. \quad (S.70d)$$

We thus have

$$\tau \sim \frac{50,000}{\chi_i}. \quad (S.71)$$

For supervised learning, $\chi_i$ is relatively high. We thus use $\chi_i \sim 0.5$, and hence $\tau = 100,000$ timesteps, or 1,000 s. For unsupervised and reinforcement learning, we used $\tau = 1,000,000$ timesteps or 10,000 s to account for lower values of $\chi_i$. For reinforcement learning, the problem is so hard (i.e. $\chi_i$ is so small) that it was also necessary to use $n = 100$. 

13
References

[1] B. Widrow and M. E. Hoff, “Adaptive switching circuits.,” 1960.

[2] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” Machine Learning, vol. 8, no. 3-4, pp. 229–256, 1992.

[3] A. Gelb, Applied Optimal Estimation. MIT Press, Jan. 1974.

[4] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii, “Highly nonrandom features of synaptic connectivity in local cortical circuits,” PLoS Biology, vol. 3, no. 3, p. e68, 2005.