Spiking allows neurons to estimate their causal effect

Benjamin James Lansdell[1,+] and Konrad Paul Kording[1]

[1]*Department of Bioengineering, University of Pennsylvania, PA, USA*
[+]lansdell@seas.upenn.edu

Keywords: causal inference, reinforcement learning, reward-modulated learning, plasticity, noise correlations

Summary

Neural plasticity can be seen as ultimately aiming at the maximization of reward. However, the world is complicated and nonlinear and so are neurons' firing properties. A neuron learning to make changes that lead to the maximization of reward is an estimation problem: would there be more reward if the neural activity had been different? Statistically, this is a causal inference problem. Here we show how the spiking discontinuity of neurons can be a tool to estimate the causal influence of a neuron's activity on reward. We show how it can be used to derive a novel learning rule that can operate in the presence of non-linearities and the confounding influence of other neurons. We establish a link between simple learning rules and an existing causal inference method from econometrics, yielding proofs of both the correctness of the approach as well as its asymptotic behavior.

Introduction

Many learning problems in neuroscience can be cast as optimization, in which a neural network's output is modified to increase performance [38]. One way of implementing this is through hedonistic synapses, in which each synapse changes itself so that the expected reward increases [41]. Such optimization requires the local estimation of the influence of a change of the synaptic weight on the expected reward of the organism. This estimation is made by each synapse randomly perturbing its output and observing how these random changes correlate with reward outcomes. While this is in principle a solution to the optimization problem, it is quite inefficient [36].

Optimization through gradient descent, on the other hand, has been shown to be efficient in artificial neural networks [26]. This relies on the propagation of derivatives through the network in a backwards direction. However, backpropagation as a method of learning in the brain suffers from two problems: first, it is hard to argue that the brain has the necessary hardware to propagate gradients (though this issue may be surmountable [29, 15]); and second, cortical networks often have low firing rates in which the stochastic and discontinuous nature of spiking output cannot be neglected [42]. In such a case the derivative is simply undefined. This raises the question if there are ways of locally evaluating the effect of activity on reward without having to propagate derivatives.

The theory of reinforcement learning provides methods that do not require backpropagated gradient signals [45]. In reinforcement learning an agent acts on the world through a policy, which defines what to do as a function of what the system believes about its environment. By exploring many actions, the system can learn to optimize its own influence on the reward. Hedonistic synapses are a special case of reinforcement learning. If we think of neurons as reinforcement learners, all they need is to have an input, an output and access to a globally distributed reward signal. Such a system can naturally deal with spiking systems.

Reinforcement learning methods could provide a framework to understand learning in some neural circuits, as there are a large number of neuromodulators which may represent reward or expected reward. Examples include dopaminergic neurons from the substantia nigra to the ventral striatum representing a reward prediction error [50, 39], and climbing fibre inputs in the cerebellum representing an error signal for adaptive motor control [30]. In essence, these reinforcement learning mechanisms locally add noise, measure the correlation of this noise with ultimate (potentially long term) reward and use the correlation to update weights. These ideas have extensively been used to model learning in brains [6, 11, 10, 28, 33, 52, 41]. Such mechanisms for implementing reinforcement learning-type algorithms may thus be compatible with what we know about brains.

There are two factors that cast doubt on the use of reinforcement learning-type algorithms in neural circuits. First, even for a fixed stimulus, noise is correlated across neurons [4, 8, 22, 5, 53]. Thus if the noise a neuron uses for learning is correlated with other neurons then we can not know which neuron's changes in output is responsible for changes in reward. In such a case, the synchronizing presynaptic activity acts as a so-called confounder, a variable that affects both the outcome and the internal activities. Estimating actual causal influences is hard in the presence of confounding. Second, it requires biophysical mechanisms to distinguish perturbative noise from input signals in presynaptic activity, and in general it is unclear how a neuron could do so (though see [10] for one example in zebrafinches). Thus it is preferable if a neuron does not have to inject noise. These factors make optimization through reinforcement learning a challenge in many neural circuits.

The field of causal inference deals with questions of how to estimate causal effects in the presence of confounding [34, 2] and may thus hold clues on how to optimize in the presence of correlated presynaptic activity. After all, reinforcement learning relies on estimating the effect of an agent's activity on a reward signal, and thus it relies on estimating a causal effect [51, 34, 16]. The causal inference field has described a lot of approaches that allow estimating causality without having to randomly perturb the relevant variables. We may look to that field to gain insights into how to estimate actual causal influences of neurons onto reward.

The gold-standard approach to causal inference is through intervention. When some spiking is randomly perturbed by independent noise then the correlation of reward with those random perturbations reveals causality. Most neural learning rules proposed to date perform this type of causal inference. Perturbation-based learning rules using external input noise have been proposed in numerous settings [6, 10, 11, 28, 33], while other approaches rely on intrinsically generated noisy perturbations to infer causality [52, 41]. These methods can be thought of as equivalent to randomized controlled trials in medicine [31] and A/B tests in computer science [23]. When feasible, interventions provide an unbiased way to identify causal effects.

In many cases interventions are not possible, either because they are difficult, expensive or unethical to perform. A central aim of causal inference is identifying when causal effects can be measured. A popular approach derives from focusing on a graph that contains all relevant variables and their relationship to one another [34]. This way of thinking provides some methods to identify causality in the presence of (observed) confounding. These methods rely on the assumptions about the relevant variables and relationships being accurate. Ultimately in large, complicated systems with many, potentially unknown, confounding factors this may be unrealistic. Economists thus typically focus on causal inference methods that rely on cases

where observational data can be thought of as approximating data obtained from interventions, so-called quasi-experimental methods [2]. A neuron embedded in a large and complicated system (the brain) may benefit from quasi-experimental methods to estimate its causal effect.

Here we frame the optimization problem as one of quasi-experimental causal learning and use a commonly employed causal inference technique, regression discontinuity design (RDD), to estimate causal effects. We apply this idea to learning in neural networks, and show that focusing on the underlying causality problem reveals a new class of learning rules that is robust to noise correlations between neurons.

Results

Regression discontinuity design (RDD) [1, 21] is a popular causal inference technique in the field of economics. RDD is applicable in cases where a binary treatment of interest, H, is determined by thresholding an input variable Z, called a forcing or running variable. We would like to estimate the effect of the treatment on an output variable, R. Under such circumstances, RDD allows us to estimate the causal effect in these cases without intervention.

Thresholds are common in many settings, making RDD a widely applicable method. There are many policies in economics, education and health-care that threshold on, for example, income, exam score, and blood pressure. We will explain the method with an example from education.

Suppose we wish to know the effect of the National Certificate of Merit on student outcomes [47]. Students who perform above a fixed threshold in an aptitude test receive a certificate of merit (Figure 1A). We can use this threshold to estimate the causal effect of the certificate on the chance of receiving a scholarship. A naive estimate of the effect can be made by comparing the students who receive the certificate to those who do not, which we will term the observed dependence (OD):

$$\beta^{OD} := \mathbb{E}(R|H = 1) - \mathbb{E}(R|H = 0).$$

But of course there will be differences between the two groups, e.g. stronger students will tend have received the certificate. Effects based on student skills and the certificate will be superimposed, confounding the estimate.

A more meaningful estimate comes from focusing on marginal cases. If we compare the students that are right below the threshold and those that are right above the threshold then they will effectively have the same exam performance. And, since exam performance is noisy, the statistical difference between marginally sub- and super- threshold students will be negligible. Therefore the difference in outcome between the two populations of students will be attributable *only* to the fact one group received the certificate and the other did not, providing a measure of causal effect (Figure 1A). If χ is the threshold exam score, then RDD computes

$$\beta^{RD} := \lim_{z \to \chi^+} \mathbb{E}(R|Z = z) - \lim_{z \to \chi^-} \mathbb{E}(R|Z = z).$$

This estimates the causal effect of treatment without requiring the injection of noise. RDD uses local regression near the threshold to obtain statistical power while avoiding confounding. In this way RDD can estimate causal effects without intervention.

A neuron is confronted with a similar inference problem. How can it estimate the effect of its activity on reward, when its activity may be correlated with other neurons? The RDD method aligns well with a defining feature of neuron physiology: a neuron receives input over a given time period, and if that input places the neuron's voltage above a threshold, then the neuron spikes (Figure 1A). Thus we propose that a neuron can use this behavior to estimate the effect of its spiking on an observed reward signal.

More specifically: neurons spike when their maximal drive Z_i exceeds a threshold, in analogy to the score in the schooling example, and then can receive feedback or a reward signal R through neuromodulator signals. Then the comparison in reward between time periods when a neuron almost reaches its firing threshold to moments when it just reaches its threshold allows an RDD estimate of its own causal effect (Figure 1B,C). Thus, rather than using randomized perturbations from an additional noise source, a neuron can take advantage of the interaction of its threshold with its drive.

To implement RDD a neuron can estimate a piece-wise linear model of the reward function at time periods when its inputs place it close to threshold:

$$R = \gamma_i + \beta_i H_i + [\alpha_{ri} H_i + \alpha_{li}(1 - H_i)](Z_i - \mu). \tag{1}$$

Here H_i is neuron i's spiking indicator function, γ_i, α_{li} and α_{ri} are the slopes that correct biases that would otherwise occur from having a finite bandwidth, Z_i is the maximum neural drive to the neuron over a short time period, and β_i represents the causal effect of neuron i's spiking over a fixed time window of period T. The neural drive used here is the leaky, integrated input to the neuron, that obeys the same dynamics as the membrane potential except without a reset mechanism. By tracking the maximum drive attained over a short time period, marginally super-threshold inputs can be distinguished from well-above-threshold inputs, as required to apply RDD. Proposed physiological implementations of this model are described in the discussion.

To demonstrate that a neuron can use RDD to estimate causal effects here we analyze a simple two neuron network obeying leaky integrate-and-fire (LIF) dynamics. The neurons receive an input signal x with added noise, correlated with coefficient c. Each neuron weighs the noisy input by w_i. The correlation in input noise induces a correlation in the output spike trains of the two neurons [43], thereby introducing confounding. The neural output determines a non-convex reward signal R. Most aspects of causal inference can be investigated in a simple model such as this [35], thus demonstrating that a neuron can estimate a causal effect with RDD in this simple case is an important first step to understanding how it can do so in a larger network.

Applying the RDD estimator shows that a neuron can estimate its causal effect (Figure 2A,B). To show that it removes confounding, we implement a simplified RDD estimator that considers only average difference in reward above and below threshold within a window of size p. When p is large this correspond to the biased observed dependence estimator, while small p values approximate the RDD estimator and result in an unbiased estimate (Figure 2A). Instead a locally linear RDD model, (1), can be used. This model is more robust to confounding (Figure 2B), allowing larger p values to be used. Thus the linear correction that is the basis of many RDD implementations [21] allows neurons to readily estimate their causal effect.

To investigate the robustness of the RDD estimator, we systematically vary the weights, w_i, of the network. RDD works better when activity is fluctuation-driven and at a lower firing rate (Figure 2C). Thus RDD is most applicable in irregular but synchronous activity regimes [7]. Over this range of network weights RDD is less biased than the observed dependence (Figure 2D). The causal effect can be used to estimate $\frac{\partial R}{\partial w_i}$ (Figure 2E,F), and thus the RDD estimator may be used in a learning rule to update weights so as to maximize expected reward (see Methods).

To demonstrate how a neuron can learn β through RDD, we derive an online learning rule from the linear model. The rule takes the form:

$$\Delta \mathbf{u}_i = \begin{cases} -\eta[\mathbf{u}_i^T \mathbf{a}_i - R]\mathbf{a}_i, & \theta \leq Z_i < \theta + p \text{ (just spikes);} \\ -\eta[\mathbf{u}_i^T \mathbf{a}_i + R]\mathbf{a}_i, & \theta - p < Z_i < \theta \text{ (almost spikes),} \end{cases}$$

where $\mathbf{u}_i$ are the parameters of the linear model required to estimate β_i, η is a learning rate, and $\mathbf{a}_i$ are drive-dependent terms (see Methods).

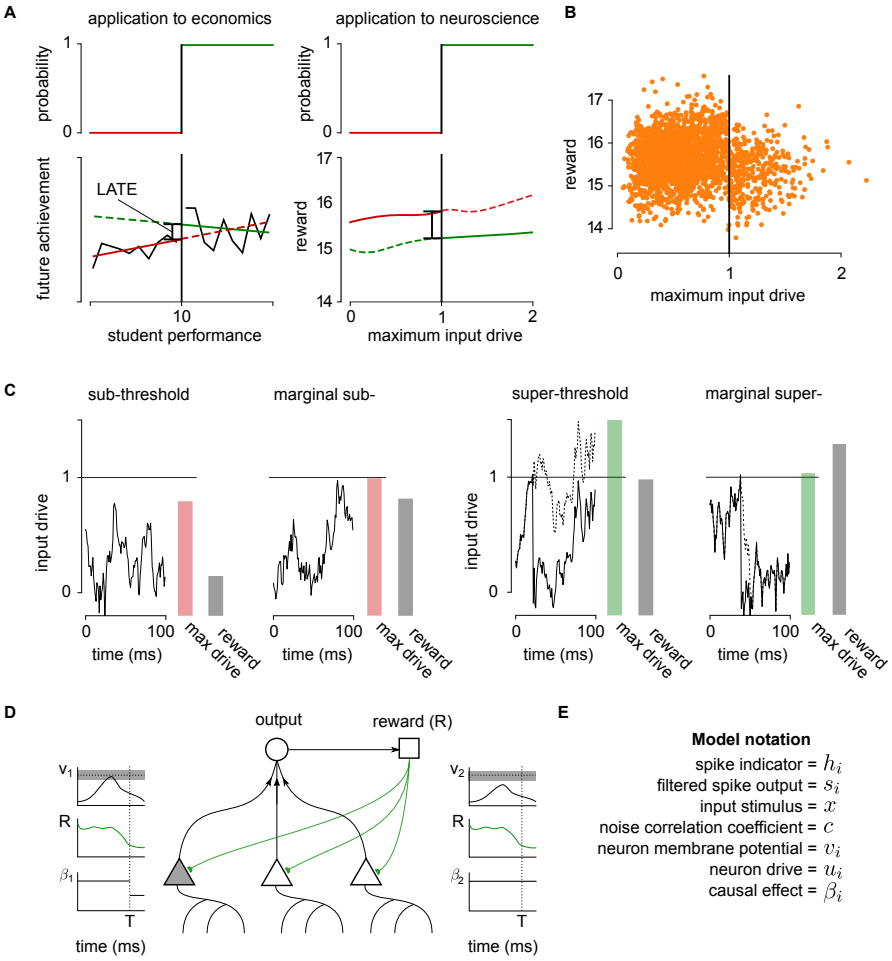

Figure 1: **Applications of regression discontinuity design.** (A) (left) In education, the effect of receiving a certificate of merit on chance of receiving a scholarship can be obtained by focusing on students at the threshold. The discontinuity at the threshold is a meaningful estimate of the local average treatment effect (LATE), or causal effect. Figure based on [47]. (right) In neuroscience, the effect of a spike on a reward function can be determined by considering cases when the neuron is driven to be just above or just below threshold. (B) The maximum drive versus the reward shows a discontinuity at the spiking threshold, which represents the causal effect. (C) This is judged by looking at the neural drive to the neuron over a short time period. Marginal sub- and super-threshold cases can be distinguished by considering the maximum drive throughout this period. (D) Schematic showing how RDD operates in network of neurons. Each neuron contributes to output, and observes a resulting reward signal. Learning takes place at end of windows of length T. Only neurons whose input drive brought it close to, or just above, threshold (gray bar in voltage traces; compare neuron 1 to 2) update their estimate of β. (E) Model notation.

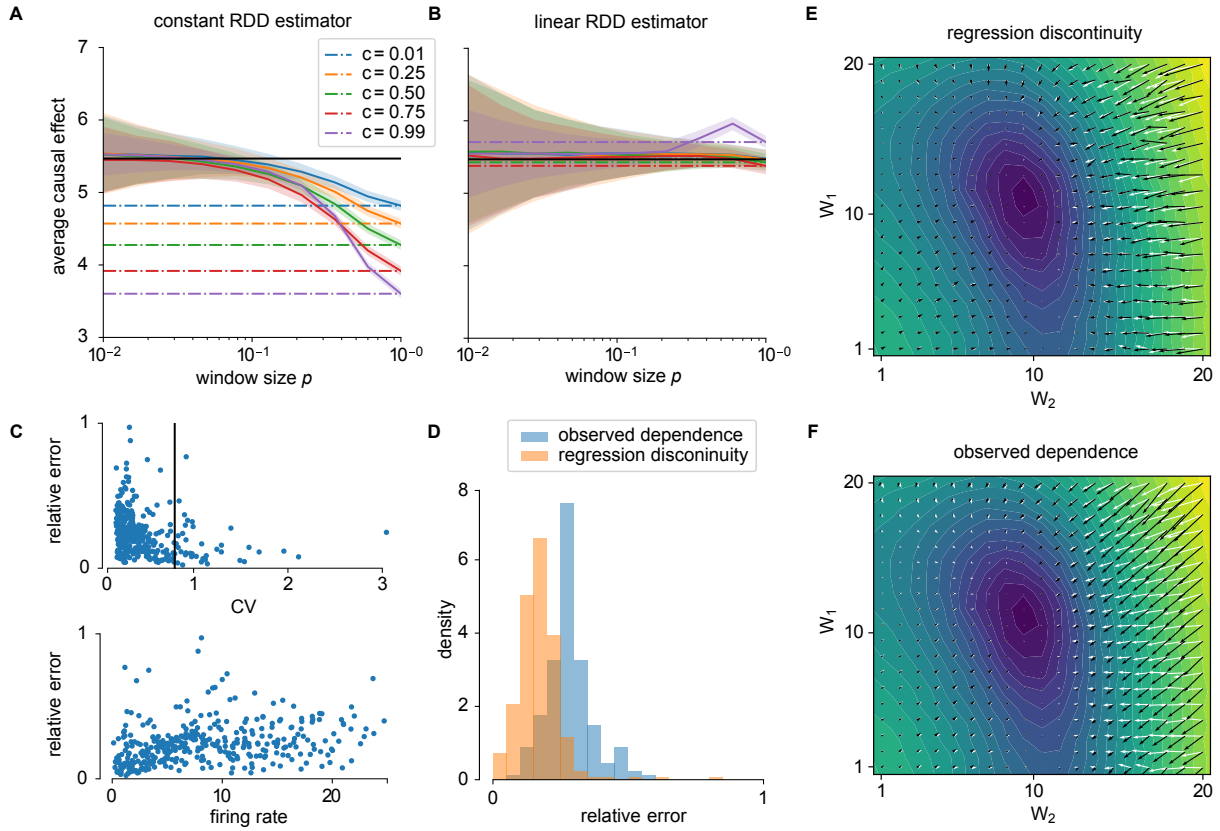

Figure 2: **Estimating reward gradient with RDD in two-neuron network.** (A) Estimates of causal effect (black line) using a constant RDD model (difference in mean reward when neuron is within a window p of threshold) reveals confounding for high p values and highly correlated activity. $p = 1$ represents the observed dependence, revealing the extent of confounding (dashed lines). (B) The linear RDD model is unbiased over larger window sizes and more highly correlated activity (high c). (C) Relative error in estimates of causal effect over a range of weights ($1 \leq w_i \leq 20$) show lower error with higher coefficient of variability (CV; top panel), and lower error with lower firing rate (bottom panel). (D) Over this range of weights, RDD estimates are less biased than just the naive observed dependence. (E,F) Approximation to the reward gradient overlaid on the expected reward landscape. The white vector field corresponds to the true gradient field, the black field correspond to the RDD (E) and OD (F) estimates. The observed dependence is biased by correlations between neuron 1 and 2 – changes in reward caused by neuron 1 are also attributed to neuron 2.

When applied to the toy network, the online learning rule (Figure 3A) estimates β over the course of seconds (Figure 3B). When the estimated β is then used to update weights to maximize expected reward in an unconfounded network (uncorrelated – $c = 0.01$), RDD-based learning exhibits higher variance than learning using the observed dependence. RDD-based learning exhibits trajectories that are initially meander while the estimate of β settles down (Figure 3C). When a confounded network (correlated – $c = 0.5$) is used RDD exhibits similar performance, while learning based on the observed dependence sometimes fails to converge due to the bias in gradient estimate. In this case RDD also converges faster than learning based on observed dependence (Figure 3D,E). Thus the RDD based learning rule allows a network to be trained on the basis of confounded inputs.

Discussion

Here we have shown that neurons can estimate their causal effect using the method known as regression discontinuity in econometrics. We have found that spiking can be an advantage, allowing neurons to quantify their effect. We have also shown that a neuron can readily estimate the gradients of a cost after its weights and that these estimates are unbiased.

There are multiple caveats for the use of RDD. The first is that the rule is only applicable in cases where the effect of a single spike is relevant. Depending on the way a network is constructed, the importance of each neuron may decrease as the size of the network is increased. As the influence of a neuron vanishes, it becomes hard to estimate this influence. While this general scaling behavior is shared with other algorithms (e.g. backpropagation with finite bit depth), it is more crucial for RDD where there will be some noise in the evaluation of the outcome.

A second caveat is that the RDD rule does not solve the temporal credit assignment problem. It requires us to know which output is associated with which kind of activity. There are multiple approaches that can help solve this kind of problem, including the actor-critic methods [45].

A third caveat is that, as implemented here, the rule learns the effect of a neuron's activity on a reward signal for a fixed input. Thus the rule is applicable in cases where a fixed output is required. This includes learning stereotyped motor actions, or learning to mimic a parent's birdsong [10, 11]. Applying the learning rule to networks with varying inputs, as in many supervised learning tasks, would require extensions of the method. One possible extension that may address both of these caveats would be, rather than directly estimate the effect of a neuron's output on a reward function, to use the method to learn weights on feedback signals so as to approximate the causal effect – that is, to use the RDD rule to "learn how to learn." This approach has been shown to work in artificial neural networks, suggesting RDD-based learning may provide a biologically plausible and scale-able approach to learning [9, 32, 15].

This paper introduces the RDD to neuronal learning and artificial neural networks. It illustrates the difference in behavior of RDD and observed-dependence learning in the presence of confounding. While it is of a speculative nature, at least in cases where reward signals are observed, it does provide a biologically plausible account of neural learning. It produces specific predictions about neural plasticity and addresses a number of issues with other learning mechanisms.

First, RDD-based learning does not require independent noise. It is sufficient that something, in fact anything that is presynaptic, produce variability. As such, RDD approaches do not require the noise source to be directly measured. This allows to rule to be applied in a wider range of neural circuits or artificial systems.

Second, RDD-based learning removes confounding due to noise correlations. Noise correlations are known to be significant in many sensory processing areas [8]. While noise correlations' role in sensory encoding has been well studied [4, 8, 22, 5, 53], their role in learning has been less studied. This work suggests that

understanding learning as a causal inference problem can provide insight into the role of noise correlations in learning.

Finally, in previous work, spiking is typically seen as a disadvantage, and systems thus aim to remove spiking discontinuities through smoothing responses [19, 18, 27]. The RDD rule, on the other hand, exploits the spiking discontinuity. Moreover, the rule can operate in environments with non-differentiable or discontinuous reward functions. In many real-world cases, gradient descent would be useless: even if the brain could implement it, the outside world does not provide gradients (but see [48]). Our approach may thus be useful even in scenarios, such as reinforcement learning, where spiking is not necessary. Spiking may, in this sense, allow a natural way of understanding a neuron's causal influence in a complex world.

The proposed learning rule is novel and has a number of differences from well-studied learning rules such as STDP. First, it requires the separation of estimates of weight updates and the effect of the neuron's activity on reward, in a similar fashion to what are called synthetic gradient methods in machine learning [9]. Physiologically, this may correspond to the separate integration of two learning signals. Such separation is possible in apical and basal dendrites in cortical neurons [15, 24]. Second, the learning rule is applied even when the neuron does not spike, in contrast to STDP. The rule thus requires tracking not just when the neuron spikes but also when it is close to threshold, regardless of spiking. It must therefore involve a form of sub-threshold dependent plasticity [12, 13, 44]. Further, the rule requires that spiking switch the sign of plasticity. This is compatible with the interaction of modulatory influences of neuromodulators and neuronal firing [40, 3]. However, the role of these sort of interactions has not been unexplored. The rule suggests roles for theoretically unexplored forms of plasticity. Plasticity rules beyond STDP have recently started to be placed in novel theoretical frameworks (e.g. [37]).

The most important aspect of RDD-based learning is the explicit focus on causality. A causal model is one that can describe the effects of an agent's actions on an environment. Learning through the reinforcement of an agent's actions relies, even if implicitly, on a causal understanding of the environment [14, 25]. Here, by explicitly casting learning as a problem of causal inference we have developed a novel learning rule for spiking neural networks. We present the first model to propose a neuron does causal inference. We believe that focusing on causality is essential when thinking about the brain or, in fact, any system that interacts with the real world.

Methods

Neuron, noise and reward model

We consider the activity of a network of n neurons whose activity is described by their spike times

$$h_i(t) = \sum \delta(t - t_s^i).$$

Here $n = 2$. Synaptic dynamics $\mathbf{s} \in \mathbb{R}^n$ are given by

$$\tau_s \dot{s}_i = -s_i + h_i(t), \tag{2}$$

for synaptic time scale τ_s. An instantaneous reward is given by $R(\mathbf{s}) \in \mathbb{R}$. In order to have a more smooth reward signal, R is a function of $\mathbf{s}$ rather than $\mathbf{h}$. The reward function used here has the form of a Rosenbrock function:

$$R(s_1, s_2) = (a - s_1)^2 + b(s_2 - s_1^2)^2.$$

The neurons obey leaky integrate-and-fire (LIF) dynamics

$$\dot{v}_i = -g_L v_i + w_i \eta_i, \tag{3}$$

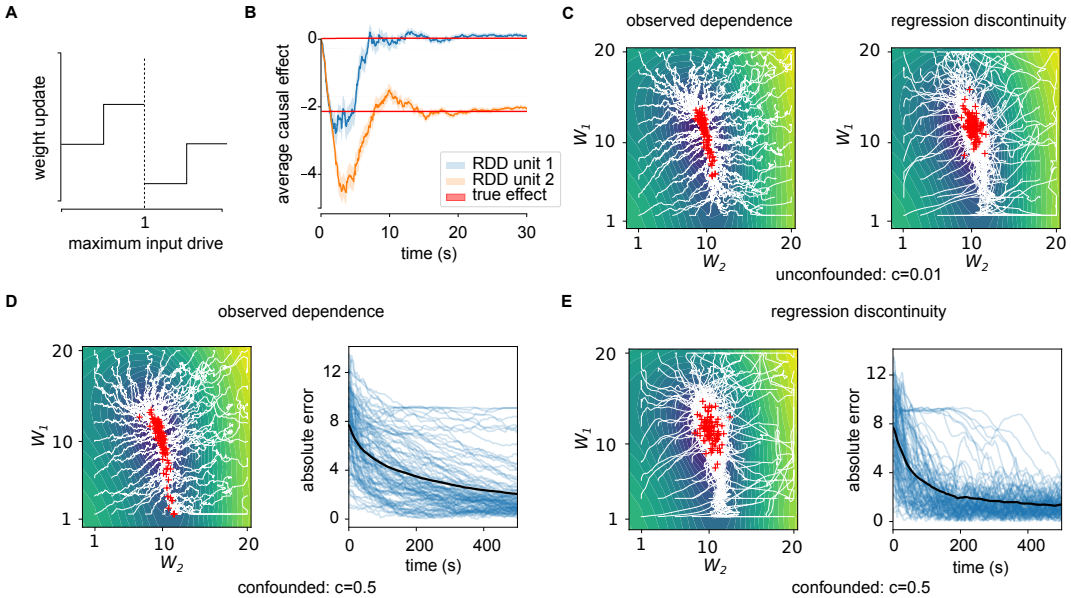

Figure 3: **Applying the RDD learning rule.** (A) Sign of RDD learning rule updates are based on whether neuron is driven marginally below or above threshold. (B) Applying rule to estimate β for two sample neurons shows convergence within 10s (red curves). Error bars represent standard error of the mean. (C) Convergence of observed dependence (left) and RDD (right) learning rule to unconfounded network ($c = 0.01$). Observed dependence converges more directly to bottom of valley, while RDD trajectories have higher variance. (D,E) Convergence of observed dependence (D) and RDD (E) learning rule to confounded network ($c = 0.5$). Right panels: error as a function of time for individual traces (blue curves) and mean (black curve). With confounding learning based on observed dependence converges slowly or not at all, whereas RDD succeeds.

where integrate and fire means simply:

$$v_i(t^+) = v_r, \quad \text{when } v_i(t) = \theta.$$

Noisy input η_i is comprised of a common DC current, x, and noise term, $\xi(t)$, plus an individual noise term, $\xi_i(t)$:

$$\eta_i(t) = x + \sigma_i \left[\sqrt{1-c}\xi_i(t) + \sqrt{c}\xi(t)\right].$$

The noise processes are independent white noise: $\mathbb{E}(\xi_i(t)\xi_j(t')) = \sigma^2\delta_{ij}\delta(t-t')$. This parameterization is chosen so that the inputs $\eta_{1,2}$ have correlation coefficient c. Simulations are performed with a step size of $\Delta t = 1$ms. Here the reset potential was set to $v_r = 0$. Borrowing notation from Xie and Seung 2004 [52], the firing rate of a noisy integrate and fire neuron is

$$\mu_i = \left[\frac{1}{g_L}\int_0^\infty \frac{1}{u}\left(\exp\left(-u^2 + 2y_i^{th}u\right) - \exp\left(-u^2 + 2y_i^r u\right)\right) du\right]^{-1},$$

where $y_i^{th} = (\theta - w_i x)/\sigma_i$ and $y_i^r = -w_i x/\sigma_i$, $\sigma_i = \sigma w_i$ is the input noise standard deviation.

We define the input drive to the neuron as the leaky integrated input without a reset mechanism. That is, over each simulated window of length T:

$$\dot{u}_i = -g_L u_i + w_i \eta_i, \quad u_i(0) = v_i(0).$$

The RDD method operates when a neuron receives inputs that place it close to its spiking threshold – either nearly spiking or barely spiking – over a given time window. In order to identify these time periods, the method uses the maximum input drive to the neuron:

$$Z_i = \max_{0 \le t \le T} u_i(t).$$

The input drive is used here instead of membrane potential directly because it can distinguish between marginally super-threshold inputs and easily super-threshold inputs, whereas this information is lost in the voltage dynamics once a reset occurs. Here a time period of $T = 50$ms was used. Reward is administered at the end of this period: $R = R(\mathbf{s}_T)$.

Policy gradient methods in neural networks

The dynamics given by (3) generate an ergodic Markov process with a stationary distribution denoted ρ. We consider the problem of finding network parameters that maximize the expected reward with respect to ρ. In reinforcement learning, performing optimization directly on the expected reward leads to policy gradient methods [46]. These typically rely on either finite difference approximations or a likelihood-ratio decomposition [49]. Both approaches ultimately can be seen as performing stochastic gradient descent, updating parameters by approximating the expected reward gradient:

$$\nabla_{\mathbf{w}} \mathbb{E}_\rho R, \tag{4}$$

for neural network parameters $\mathbf{w}$.

Here capital letters are used to denote the random variables drawn from the stationary distribution, corresponding to their dynamic lower-case equivalent above. Density ρ represents a joint distribution over variables (Z, H, S, R), the maximum input drive, spiking indicator function, filtered spiking output, and reward variable, respectively. The spiking indicator function is defined as $H_i = \mathbb{I}(Z_i \ge \theta)$, for threshold θ.

We wish to evaluate (4). In general there is no reason to expect that taking a derivative of an expectation with respect to some parameters will have the same form as the corresponding derivative of a deterministic function. However in some cases this is true, for example when the parameters are separable from the distribution over which the expectation is taken (sometimes relying on what is called the reparameterization trick [36, 17]). Here we show that, even when the reparameterization trick is unavailable, if the system contains a Bernoulli (spiking indicator) variable then the expression for the reward gradient also matches a form we might expect from taking the gradient of a deterministic function.

The expected reward can be expressed as

$$\mathbb{E}(R) = \mathbb{E}(R|H_i = 1)P(H_i = 1) + \mathbb{E}(R|H_i = 0)P(H_i = 0), \tag{5}$$

for a neuron i. The key assumption we make is the following:

Assumption 1. The neural network parameters only affect the reward through their spiking activity, meaning that $P(R|H)$ is independent of parameters $\mathbf{w}$.

Note that, even if the joint conditional expectation $\mathbb{E}(R|H)$ is independent of $\mathbf{w}$, that doesn't mean that the marginal conditional expectation, $\mathbb{E}(R|H_i)$, is independent of $\mathbf{w}$. This is because it involves marginalization over unobserved neurons $H_{j\neq i}$, which may have some relation to H_i that is dependent on $\mathbf{w}$. That is,

$$\mathbb{E}(R|H_i = 1) = \sum_{j\neq i} \mathbb{E}(R|H_i = 1, H_j)P(H_j|H_i = 1; \mathbf{w}),$$

where the conditional probabilities may depend on $\mathbf{w}$. This complicates taking derivatives of the decomposition (5) with respect to $\mathbf{w}$.

Unconfounded network

However, if we assume that $H_i \perp\!\!\!\perp H_{j\neq i}$ then the reward gradient is simple to compute from (5). This is because now the parameter w_i only affects the probability of neuron i spiking:

$$\begin{aligned}
\frac{\partial}{\partial w_i}\mathbb{E}(R) &= \frac{\partial}{\partial w_i}\left(\mathbb{E}(R|H_i = 1)P(H_i = 1; w_i) + \mathbb{E}(R|H_i = 0)P(H_i = 0; w_i)\right) \\
\text{(Assumption 1)} \quad &= \frac{\partial P(H_i = 1; w_i)}{\partial w_i}\left(\mathbb{E}(R|H_i = 1) - \mathbb{E}(R|H_i = 0)\right) \\
&= \frac{\partial \mathbb{E}(H_i; w_i)}{\partial w_i}\left(\mathbb{E}(R|H_i = 1) - \mathbb{E}(R|H_i = 0)\right).
\end{aligned}$$

This resembles a type of finite difference estimate of the gradient we might use if the system were deterministic and H were differentiable:

$$\frac{\partial R}{\partial w} \approx \frac{\partial H}{\partial w}\frac{R(H = 1) - R(H = 0)}{\Delta H}.$$

Based on the independence assumption we call this the unconfounded case. In fact the same decomposition is utilized in a REINFORCE-based method derived by Seung 2003 [41]. Relations to other expected reward gradient estimators are described in the Supplementary Materials.

Confounded network

Generally it is not the case that $H_i \perp\!\!\!\perp H_{j\neq i}$, and then we must decompose the expected reward into:

$$\mathbb{E}(R) = \sum_{h_{j\neq i}} P(H_{j\neq i} = h_{j\neq i})(\mathbb{E}(R|H_i = 1, H_{j\neq i} = h_{j\neq i})P(H_i = 1|H_{j\neq i} = h_{j\neq i}) \\ + \mathbb{E}(R|H_i = 0, H_{j\neq i} = h_{j\neq i})P(H_i = 0|H_{j\neq i} = h_{j\neq i})).$$

This means the expected reward gradient is given by:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) = \sum_{h_{j\neq i}} P(H_{j\neq i} = h_{j\neq i})\frac{\partial P(H_i = 1|H_{j\neq i} = h_{j\neq i})}{\partial w_i}(\mathbb{E}(R|H_i = 1, H_{j\neq i} = h_{j\neq i}) - \mathbb{E}(R|H_i = 0, H_{j\neq i} = h_{j\neq i})) \\ = \mathbb{E}\left(\frac{\partial \mathbb{E}(H_i|H_{j\neq i})}{\partial w_i}\left(\mathbb{E}(R|H_i = 1, H_{j\neq i}) - \mathbb{E}(R|H_i = 0, H_{j\neq i})\right)\right),$$

again making use of Assumption 1. We additionally make the following approximation:

Assumption 2. The gradient term $\frac{\partial \mathbb{E}(H_i|H_{j\neq i})}{\partial w_i}$ is independent of $H_{j\neq i}$.

This means we can move it out of the expectation to give:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i}\mathbb{E}\left(\mathbb{E}(R|H_i = 1, H_{j\neq i}) - \mathbb{E}(R|H_i = 0, H_{j\neq i})\right). \tag{6}$$

We assume that how the neuron's activity responds to changes in synaptic weights, $\frac{\partial \mathbb{E}(H_i)}{\partial w_i}$, is known by the neuron. Thus it remains to estimate $\mathbb{E}(R|H_i = 1, H_{j\neq i}) - \mathbb{E}(R|H_i = 0, H_{j\neq i})$. It would seem this relies on a neuron observing other neurons' activity. Below we show how it can be estimated, however, using methods from causal inference.

The unbiased gradient estimator as a causal effect

We can identify the unbiased estimator (Equation (6)) as a causal effect estimator. To understand precisely what this means, here we describe a causal model.

A causal model is a Bayesian network along with a mechanism to determine how the network will respond to intervention. This means a causal model is a directed acyclic graph (DAG) $\mathcal{G}$ over a set of random variables $\mathcal{X} = \{X_i\}_{i=1}^N$ and a probability distribution P that factorizes over $\mathcal{G}$ [34].

An intervention on a single variable is denoted $\text{do}(X_i = y)$. Intervening on a variable removes the edges to that variable from its parents, Pa_{X_i}, and forces the variable to take on a specific value: $P(x_i|\text{Pa}_{X_i} = \mathbf{x_i}) = \delta(x_i = y)$. Given the ability to intervene, the average treatment effect (ATE), or causal effect, between an outcome variable X_j and a binary variable X_i can be defined as:

$$ATE := \mathbb{E}(X_j|\text{do}(X_i = 1)) - \mathbb{E}(X_j|\text{do}(X_i = 0)).$$

We make use of the following result: if $\mathbf{S}_{ij} \subset \mathcal{X}$ is a set of variables that satisfy the *back-door criteria* with respect to $X_i \to X_j$, then it satisfies the following: (i) $\mathbf{S}_{ij}$ blocks all paths from X_i to X_j that go into S_i, and (ii) no variable in $\mathbf{S}_{ij}$ is a descendant of X_i. In this case the interventional expectation can be inferred from

$$\mathbb{E}(X_j|\text{do}(X_i = y)) = \mathbb{E}\left(\mathbb{E}(X_j|\mathbf{S}_{ij}, X_i = y)\right).$$

Given this framework, here we will define the causal effect of a neuron as the average causal effect of a neuron H_i spiking or not spiking on a reward signal, R:

$$\beta_i \coloneqq \mathbb{E}(R|\text{do}(H_i = 1)) - \mathbb{E}(R|\text{do}(H_i = 0)),$$

where H_i and R are evaluated over a short time window of length T.

We make the final assumption:

Assumption 3. Neurons $H_{j\neq i}$ satisfy the backdoor criterion with respect to $H_i \to R$.

Then it is the case that the reward gradient estimator, (6), in fact corresponds to:

$$\frac{\partial}{\partial w_i}\mathbb{E}(R) \approx \frac{\partial \mathbb{E}(H_i)}{\partial w_i}\beta_i. \tag{7}$$

Thus we have the result that, in a confounded, spiking network, gradient descent learning corresponds to causal learning.

Using regression discontinuity design

As described in the main text, to remove confounding, RDD considers only the marginal super- and sub-threshold periods of time to estimate (7). This works because the discontinuity in the neuron's response induces a detectable difference in outcome for only a negligible difference between sampled populations (sub- and super-threshold periods). The RDD method estimates [21]:

$$\beta_i^{RD} \coloneqq \lim_{x\to\theta^+} \mathbb{E}(R|Z_i = x) - \lim_{x\to\theta^-} \mathbb{E}(R|Z_i = x),$$

for maximum input drive obtained over a short time window, Z_i, and spiking threshold, θ; thus, $Z_i < \theta$ means neuron i does not spike and $Z_i \geq \theta$ means it does.

To estimate β_i^{RD}, a neuron can estimate a piece-wise linear model of the reward function:

$$R = \gamma_i + \beta_i H_i + [\alpha_{ri} H_i + \alpha_{li}(1 - H_i)](Z_i - \theta),$$

locally, when Z_i is within a small window p of threshold. Here γ_i, α_{li} and α_{ri} are nuisance parameters, and β_i is the causal effect of interest. This means we can estimate β_i^{RD} from

$$\beta_i \approx \mathbb{E}(R - \alpha_r(Z_i - \theta)|\theta \leq Z_i < \theta + p) - \mathbb{E}(R - \alpha_l(Z_i - \theta)|\theta - p < Z_i < \theta).$$

A neuron can learn an estimate of β_i^{RD} through a least squares minimization on the model parameters $\beta_i, \alpha_l, \alpha_r$. That is, if we let $\mathbf{u}_i = [\beta_i, \alpha_r, \alpha_l]^T$ and $\mathbf{a}_t = [1, h_{i,t}(z_{i,t} - \theta), (1 - h_{i,t})(z_{i,t} - \theta)]^T$, then the neuron solves:

$$\hat{\mathbf{u}}_i = \text{argmin}_u \sum_{t:(\theta-p<z_{i,t}<\theta+p)}^{T} \left[\mathbf{u}_i^T \mathbf{a}_t - (2h_{i,t} - 1)R_t\right]^2.$$

Performing stochastic gradient descent on this minimization problem gives the learning rule:

$$\Delta \mathbf{u}_i = \begin{cases} -\eta[\mathbf{u}_i^T \mathbf{a}_i - R_t]\mathbf{a}_i, & \theta \leq z_{i,t} < \theta + p \text{ (just spikes);} \\ -\eta[\mathbf{u}_i^T \mathbf{a}_i + R_t]\mathbf{a}_i, & \theta - p < z_{i,t} < \theta \text{ (almost spikes),} \end{cases}$$

for all time periods at which $z_{i,t}$ is within p of threshold θ.

Variance of RDD estimator

When estimated offline, the variance of the RDD estimator behaves as $\mathbb{V}(\beta) = \mathcal{O}\left(\frac{\sigma^2}{N}(\frac{1}{P_-} + \frac{1}{P_+})\right)$, where σ^2 is the noise variance in and $P_\pm$ are given by

$$P_- = \int \rho(z)\, dz, \quad P_+ = \int \rho(z)\, dz,$$

that is, the proportion of time bins producing maximum input drive just below/above threshold [20].

Implementation

`python` code used to run simulations and generates figures is available at: `https://github.com/benlansdell/rdd`.

Acknowledgements

The authors would like to thank Roozbeh Farhoodi, Ari Benjamin and David Rolnick for discussion and feedback.

Author contributions

K.P.K and B.J.L. devised the study, B.J.L. performed the analysis, and K.P.K and B.J.L. wrote the manuscript.

Supplementary material

A The causal effect as a finite difference operator

The estimator can also be considered as a type of finite-difference approximation to the reward gradient we would compute in the deterministic case. This relation is fleshed out here. Specifically, we show that

$$\beta_i = \mathbb{E}(R|\mathrm{do}(H_i = 1)) - \mathbb{E}(R|\mathrm{do}(H_i = 0)) \propto \mathbb{E}\left(\frac{\partial R}{\partial S_i}\right).$$

To show this we replace $\frac{\partial}{\partial S_i}$ with a type of finite difference operator:

$$D_i R(S_i, \mathbf{S}_{j\neq i}) \coloneqq \frac{1}{\Delta_s}\left(\mathbb{E}(R|S_i + \Delta_s, \mathbf{S}_{j\neq i}) - \mathbb{E}(R|S_i, \mathbf{S}_{j\neq i})\right).$$

Here $\mathbf{S}_{j\neq i} \subset \mathcal{X}$ is a set of nodes that satisfy the back-door criterion with respect to $H_i \to R$. When R is a deterministic, differentiable function of $\mathbf{S}$ and $\Delta_s \to 0$ this recovers the reward gradient $\frac{\partial R}{\partial S_i}$ and we recover gradient descent-based learning.

To consider the effect of a single spike, note that unit i spiking will cause a jump in S_i compared to not spiking (according to (2)). If we let Δ_s equal this jump then it can be shown that $\mathbb{E}(D_i R)$ is related to the causal effect.

First, assuming the conditional independence of R from H_i given S_i and $\mathbf{S}_{j\neq i}$:

$$\begin{aligned}\beta_i &= \mathbb{E}\left(R|\mathrm{do}(H_i=1)\right) - \mathbb{E}\left(R|\mathrm{do}(H_i=0)\right)\\ &= \mathbb{E}\left(\mathbb{E}\left(R|\mathbf{S}_{j\neq i}, H_i=1\right) - \mathbb{E}\left(R|\mathbf{S}_{j\neq i}, H_i=0\right)\right)\\ &= \mathbb{E}\left(\mathbb{E}\left(\mathbb{E}\left(R|S_i,\mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i=1\right) - \mathbb{E}\left(\mathbb{E}\left(R|S_i,\mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i=0\right)\right).\end{aligned} \tag{8}$$

Now if we assume that on average H_i spiking induces a change of Δ_s in S_i within the same time period, compared with not spiking, then:

$$\rho(s_i|\mathbf{S}_{j\neq i}, H_i=1) \approx \rho(s_i-\Delta_s|\mathbf{S}_{j\neq i}, H_i=0). \tag{9}$$

This is reasonable because the linearity of the synaptic dynamics, (2), means that the difference in S_i between spiking and non-spiking windows is simply $\exp(-t_{si}/\tau_s)/\tau_s$, for spike time t_{si}. We approximate this term with its mean:

$$\begin{aligned}\Delta_s &= \mathbb{E}\left(\frac{1}{\tau_s}e^{-t_{si}/\tau_s}|\mathbf{S}_{j\neq i}, H_i=1\right)\\ &\approx \frac{1}{T}\left(1-e^{-T/\tau_s}\right),\end{aligned} \tag{10}$$

under the assumption that spike times occur uniformly throughout the length T window. These assumptions are supported numerically (Supplementary Figure 1).

Writing out the inner two expectations of (8) gives:

$$\begin{aligned}&\mathbb{E}\left(\mathbb{E}\left(R|S_i,\mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i=1\right) - \mathbb{E}\left(\mathbb{E}\left(R|S_i,\mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i=0\right)\\ &= \int_0^\infty \mathbb{E}(R|\mathbf{S}_{j\neq i}, S_i=s_i)\left[\rho(s_i|\mathbf{S}_{j\neq i}, H_i=1) - \rho(s_i|\mathbf{S}_{j\neq i}, H_i=0)\right]\, ds_i\\ \text{from (9)}\quad &= \int_0^\infty \mathbb{E}(R|S_i=s_i+\Delta_s,\mathbf{S}_{j\neq i})\rho(s_i|\mathbf{S}_{j\neq i}, H_i=0) - \mathbb{E}(R|S_i=s_i,\mathbf{S}_{j\neq i})\rho(s_i|\mathbf{S}_{j\neq i}, H_i=0)\, ds_i,\end{aligned}$$

after making the substitution $s_i \to s_i+\Delta_s$ in the first term. Writing this back in terms of expectations gives:

$$\begin{aligned}\beta &\approx \mathbb{E}\left(\mathbb{E}\left(\mathbb{E}\left(R|S_i+\Delta_s,\mathbf{S}_{j\neq i}\right) - \mathbb{E}\left(R|S_i,\mathbf{S}_{j\neq i}\right)|\mathbf{S}_{j\neq i}, H_i=0\right)\right)\\ &= \Delta_s\mathbb{E}(\mathbb{E}\left(D_iR(S_i,\mathbf{S}_{j\neq i})|\mathbf{S}_{j\neq i}, H_i=0\right))\\ &= \Delta_s\mathbb{E}(D_iR(S_i,\mathbf{S}_{j\neq i})|\mathrm{do}(H_i=0)).\end{aligned}$$

Thus estimating the causal effect is similar to taking a finite difference approximation of the reward gradient.

References

[1] Joshua Angrist and Jorn-Steffen Pischke. The Credibility Revolution in Empirical Economics: How Better Research Design is Taking the Con Out of Economics. *Journal of Economic Perspectives*, 24(2), 2010.

[2] Joshua D Angrist and Jorn-Steffen Pischke. *Mostly Harmless Econometrics : An Empiricist ' s Companion.* Princeton University Press, 2009.

[3] A Artola, S Brocher, and W Singer. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347(6288):69–72, 1990.

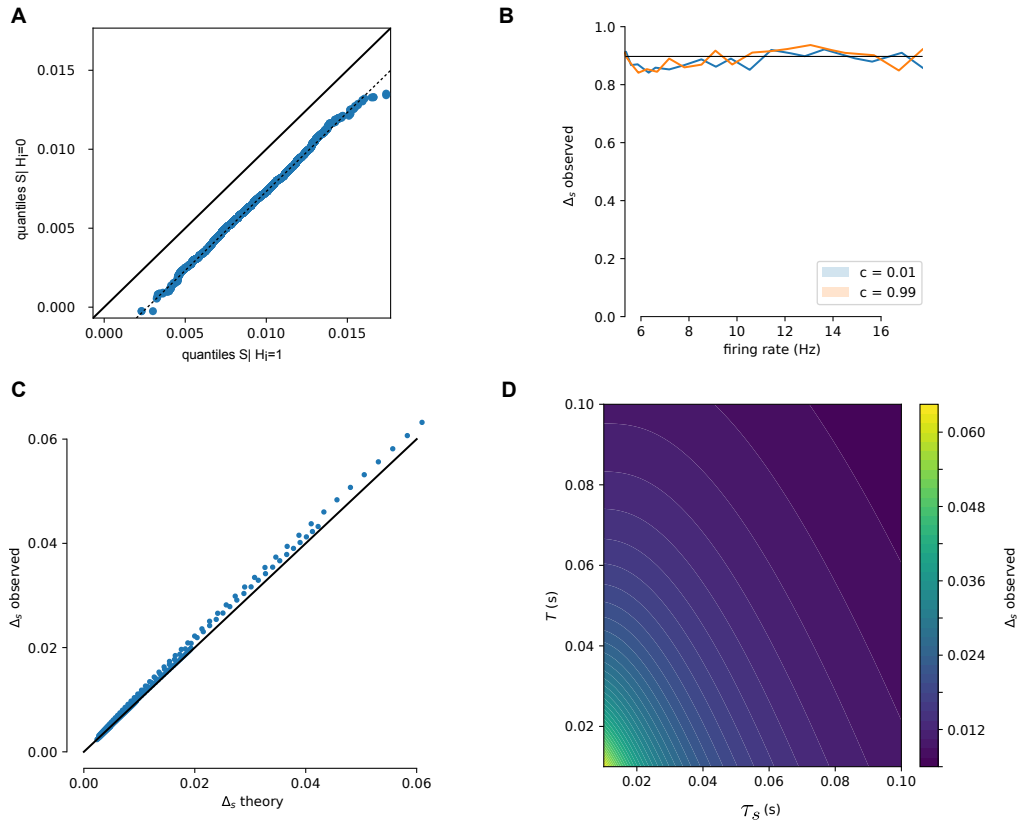

Figure 4: Supplementary Figure 1. **Relation between S_i and H_i over window T.** (A) Simulated spike trains are used to generate $S_i|H_i = 0$ and $S_i|H_i = 1$. QQ-plot shows that S_i following a spike is distributed as a translation of S_i in windows with no spike, as assumed in (9). (B) This offset, Δ_s, is independent of firing rate and is unaffected by correlated spike trains. (C) Over a range of values $(0.01 < T < 0.1, 0.01 < \tau_s < 0.1)$ the derived estimate of Δ_s (Equation (10)) is compared to simulated Δ_s. Proximity to the diagonal line (black curve) shows these match. (D) Δ_s as a function of window size T and synaptic time constant τ_s. Larger time windows and longer time constants lower the change in S_i due to a single spike.

[4] Cody Baker, Christopher Ebsch, Ilan Lampl, and Robert Rosenbaum. The correlated state in balanced neuronal networks. *bioRxiv*, 2018.

[5] Vikranth R. Bejjanki, Rava Azeredo da Silveira, Jonathan D. Cohen, and Nicholas B. Turk-Browne. Noise correlations in the human brain and their impact on pattern classification. *PLoS computational biology*, 13(8):e1005674, 2017.

[6] Guy Bouvier, Claudia Clopath, Célian Bimbard, Jean-Pierre Nadal, Nicolas Brunel, Vincent Hakim, and Boris Barbour. Cerebellar learning using perturbations. *bioRxiv*, page 053785, 2016.

[7] N Brunel. Dynamics of sparsely connected networls of excitatory and inhibitory neurons. *Computational Neuroscience*, 8:183–208, 2000.

[8] Marlene R. Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature Neuroscience*, 14(7):811–819, 2011.

[9] Wojciech Marian Czarnecki, Grzegorz Świrszcz, Max Jaderberg, Simon Osindero, Oriol Vinyals, and Koray Kavukcuoglu. Understanding Synthetic Gradients and Decoupled Neural Interfaces. *ArXiv e-prints*, 2017.

[10] Ila R Fiete, Michale S Fee, and H Sebastian Seung. Model of Birdsong Learning Based on Gradient Estimation by Dynamic Perturbation of Neural Conductances. *Journal of neurophysiology*, 98:2038–2057, 2007.

[11] Ila R Fiete and H Sebastian Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical Review Letters*, 97, 2006.

[12] Elodie Fino, Jean Michel Deniau, and Laurent Venance. Brief subthreshold events can act as Hebbian signals for long-term plasticity. *PLoS ONE*, 4(8), 2009.

[13] Elodie Fino and Laurent Venance. Spike-timing dependent plasticity in the striatum. *Frontiers in synaptic neuroscience*, 2(June):1–10, 2010.

[14] Samuel J Gershman. Reinforcement learning and causal models. In *Oxford Handbook of Causal Reasoning*, pages 1–32. Oxford university press, 2017.

[15] Jordan Guergiuev, Timothy P. Lillicrap, and Blake A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:1–37, 2017.

[16] York Hagmayer and Philip Fernbach. *Causality in Decision-Making*, volume 1. 2017.

[17] Nicolas Heess, Greg Wayne, David Silver, Timothy Lillicrap, Yuval Tassa, and Tom Erez. Learning Continuous Control Policies by Stochastic Value Gradients. *Advances in Neural Information Processing Systems*, 28:1–13, 2015.

[18] Dongsung Huh and Terrence J Sejnowski. Gradient Descent for Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[19] Eric Hunsberger and Chris Eliasmith. Spiking Deep Networks with LIF Neurons. *Advances in Neural Information Processing Systems*, 28:1–9, 2015.

[20] Gguido Imbens and Karthik Kalyanaraman. Optimal bandwidth choice for the regression discontinuity estimator. *Review of Economic Studies*, 79(3):933–959, 2012.

[21] Guido W Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2):615–635, 2008.

[22] Ingmar Kanitscheider, Ruben Coen-cagli, and Alexandre Pouget. Origin of information-limiting noise correlations. 2015.

[23] Ron Kohavi, Randal M. Rm Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, 2007:959–967, 2007.

[24] Konrad Kording and Peter Konig. Supervised and Unsupervised Learning with Two Sites of Synaptic Integration. *Journal of Computational Neuroscience*, 11:207–215, 2001.

[25] Mike E. Le Pelley, Oren Griffiths, and Tom Beesley. Associative Accounts of Causal Cognition. In *Oxford Handbook of Causal Reasoning*, volume 1, pages 1–27. Oxford university press, 2017.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521, 2015.

[27] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, 10:1–13, 2016.

[28] Robert Legenstein, Steven M Chase, Andrew B Schwartz, and Wolfgang Maas. A reward-modulated Hebbian learning rule can explain experimentally observed network reorganization in a brain control task. *Journal of Neuroscience*, 30(25):8400–8410, 2010.

[29] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random feedback weights support learning in deep neural networks. *Nature Communications*, 7:13276, 2016.

[30] David Marr. A theory of cerebellar cortex. *J. Physiol*, 202:437–470, 1969.

[31] Marcia L. Meldrum. A brief history of the randomized controlled trial: Oranges and Lemons to the Gold Standard. *Hematology/Oncology Clinics of North America*, 14(4):745–760, 2000.

[32] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning Unsupervised Learning Rules. *ArXiv e-prints*, 2018.

[33] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *eLife*, 6:1–24, 2017.

[34] Judea Pearl. *Causality: models, reasoning and inference*. Cambridge Univ Press, 2000.

[35] J Peters, D Janzing, and B Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

[36] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *Proceedings of the 31st International Conference on Machine Learning, PMLR*, 32(2):1278–1286, 2014.

[37] Pieter R Roelfsema and Anthony Holtmaat. Control of synaptic plasticity in deep cortical networks. pages 1–25, 2017.

[38] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

[39] Wolfram Schultz. Getting formal with dopamine and reward. *Neuron*, 36(2):241–263, 2002.

[40] Geun Hee Seol, Jokubas Ziburkus, Shiyong Huang, Lihua Song, In Tae Kim, Kogo Takamiya, Richard L Huganir, Hey-kyoung Lee, and Alfredo Kirkwood. Neuromodulators Control the Polarity of Spike-Timing-Dependent Synaptic Plasticity. *Neuron*, 55(6):919–929, 2007.

[41] Sebastian Seung. Learning in Spiking Neural Networks by Reinforcement of Stochastics Transmission. *Neuron*, 40:1063–1073, 2003.

[42] M. Shafi, Y. Zhou, J. Quintana, C. Chow, J. Fuster, and M. Bodner. Variability in neuronal activity in primate cortex during working memory tasks. *Neuroscience*, 146(3):1082–1108, 2007.

[43] Eric Shea-Brown, Krešimir Josić, Jaime De La Rocha, and Brent Doiron. Correlation and synchrony transfer in integrate-and-fire neurons: Basic properties and consequences for coding. *Physical Review Letters*, 100(10):1–4, 2008.

[44] Per Jesper Sjöström, Gina G Turrigiano, and Sacha B Nelson. Endocannabinoid-Dependent Neocortical Layer-5 LTD in the Absence of Postsynaptic Spiking. *Journal of Neurophysiology*, 92(6):3338–3343, 2004.

[45] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2017.

[46] Richard S. Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 1999.

[47] Donald L Thistlewaite, Donald T Campbell, Peter Aronow, Nicole Basta, Betz Halloran, Matias Cattaneo, Gonzalo Vazquez-Bare, Guido Imbens, Alessan-Dra Mattei, Fabrizia Mealli, Jasjeet Sekhon, Rocío Titiunik, Vivian Wong, and Coady Wing. Regression-discontinuity Analysis: An alternative to the ex-post facto experiment. *Journal of Educational Psychology*, 51:309–317, 1960.

[48] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *IEEE International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[49] Ronald Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:299–256, 1992.

[50] Roy a RA Wise. Dopamine, learning and motivation. *Nature reviews. Neuroscience*, 5(6):483–494, jun 2004.

[51] Jim Woodward. Interventionist theories of causation in psychological perspective. In *Causal learning: psychology, philosophy and computation*. Oxford university press, New York, 2007.

[52] Xiaohui Xie and H. Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69, 2004.

[53] Man Yi Yim, Ad Aertsen, and Arvind Kumar. Significance of Input Correlations in Striatal Function. *PLoS Comput Biol*, 7(11), 2011.