PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Synaptic balancing: A biologically plausible local learning rule that provably increases neural network noise robustness without sacrificing task performance

Christopher H. Stock[1], **Sarah E. Harvey**[2]*, **Samuel A. Ocko**[2], **Surya Ganguli**[2,3]

1 Neuroscience Graduate Program, Stanford University School of Medicine, Stanford, California, United States of America, **2** Department of Applied Physics, Stanford University, Stanford, California, United States of America, **3** Stanford Institute for Human-Centered Artificial Intelligence, Stanford University, Stanford, California, United States of America

* harveys@stanford.edu

Citation: Stock CH, Harvey SE, Ocko SA, Ganguli S (2022) Synaptic balancing: A biologically plausible local learning rule that provably increases neural network noise robustness without sacrificing task performance. PLoS Comput Biol 18(9): e1010418. https://doi.org/10.1371/journal.pcbi.1010418

Editor: Blake A. Richards, McGill University, CANADA

Received: December 8, 2020

Accepted: July 20, 2022

Published: September 19, 2022

Data Availability Statement: Code for reproducing the results of this paper may be found on GitHub (https://github.com/chris-stock/synaptic-balancing).

Funding: The authors received no specific funding for this work.

Competing interests: The autors have declared that no competing interest exist.

Abstract

We introduce a novel, biologically plausible local learning rule that provably increases the robustness of neural dynamics to noise in nonlinear recurrent neural networks with homogeneous nonlinearities. Our learning rule achieves higher noise robustness without sacrificing performance on the task and without requiring any knowledge of the particular task. The plasticity dynamics—an integrable dynamical system operating on the weights of the network—maintains a multiplicity of conserved quantities, most notably the network's entire temporal map of input to output trajectories. The outcome of our learning rule is a synaptic balancing between the incoming and outgoing synapses of every neuron. This synaptic balancing rule is consistent with many known aspects of experimentally observed heterosynaptic plasticity, and moreover makes new experimentally testable predictions relating plasticity at the incoming and outgoing synapses of individual neurons. Overall, this work provides a novel, practical local learning rule that exactly preserves overall network function and, in doing so, provides new conceptual bridges between the disparate worlds of the neurobiology of heterosynaptic plasticity, the engineering of regularized noise-robust networks, and the mathematics of integrable Lax dynamical systems.

Author summary

The precise pattern of synaptic connections in a neural network entirely determines the network's function. Learning rules are programs for modifying synapses to allow the network to attain some functional goal. By carefully tweaking synaptic connectivity, neural networks are able learn new tasks, form new memories, and stabilize neural activity. Due to the complexity of network structure, there are many different synaptic weight patterns that can in principle perform the same computation. These different solutions, however, may be more or less desirable in other other ways, such as their vulnerability to unreliable components in the network. In this paper, we present a synaptic learning rule which

locates robust solutions among many synaptic weight configurations that equivalently perform the same task. Our biologically plausible rule relies on information which is entirely locally available to each neuron and possesses a number of desirable mathematical properties. Our analysis reveals that networks are most robust when the aggregate incoming and outgoing synapses are balanced in every neuron. Intriguingly, our learning rule makes predictions which resemble widely experimentally observed patterns of compensatory heterosynaptic plasticity in cortical dendrites.

Introduction

As any neural circuit computes, it is subject to additional fluctuations either within the circuit or from other brain regions [1–4], and these fluctuations can impair performance [5–8]. The fundamental puzzle we address is what kind of plasticity rule can make the dynamics of a neural circuit more robust to such fluctuations in a manner that: (1) works for any task; (2) is completely agnostic to the learning rule used to solve a task; and (3) does not impair network performance after a task is learned. Our main contribution is the discovery of a plasticity rule that provably accomplishes all of three objectives for any nonlinear recurrent neural network with a homogeneous nonlinearity, such as the commonly studied rectified linear function. Furthermore, our plasticity rule is biologically plausible and computable using only information that is locally available at each synapse and its adjacent neurons. Finally, at a mathematical level our plasticity rule connects to the theory of integrable dynamical systems [9–11] and heat diffusion, while experimentally, its features are similar to observed aspects of heterosynaptic plasticity [12–16].

The key idea behind our learning rule is to exploit a many-to-one mapping between patterns of synaptic strength and the task, or the temporal input-output map implemented by a recurrent neural network. Indeed, modern neural network models of animal behavior typically possess a large number of tunable parameters—far more than necessary to perform a given simple behavior. In this overparameterized regime, it has been observed across multiple behaviors and organisms that many distinct model configurations are able to generate equivalent levels of task performance [17, 18]. This observation has spurred theoretical and numerical investigations into specific means by which a task may constrain network connectivity and function [19–23]. However, the complex, nonlinear nature of many classes of network models in neuroscience—notably recurrent neural networks (RNNs)—has made it difficult in many cases to obtain a precise theoretical characterization of the space of synaptic patterns that all map to the same task [24].

Besides a theoretical interest in formally characterizing equivalence classes of synaptic weights that solve the same task, we are also motivated by a complementary biological question: how might neural systems actually implement local plasticity rules which take advantage of network overparameterization to maintain desirable functional properties, including task performance, in the presence of internal and external sources of variation? Studies of homeostatic plasticity in cortical circuits have shed light on synaptic mechanisms thought to maintain stable computation, including synaptic scaling, heterosynaptic plasticity, and other compensatory processes [13, 25–28]. However, a fundamental neuroscientific question remains: can such local homeostatic or compensatory plasticity rules operate in such a way so as not to impair task performance, while still accruing some other additional benefit?

In this work, we provide insights into both the nature of overparameterization in recurrent neural networks and how local plasticity rules might exploit this overparameterization to

specifically improve network robustness without impairing task performance. First we precisely characterize the equivalence classes of synaptic weights that all solve the same task in nonlinear recurrent networks with homogenous nonlinearities. Using this characterization, we derive a simple associated plasticity rule that locally modifies synaptic weights while remaining within these equivalence classes. Intriguingly, we find our theoretically derived plasticity rule resembles experimentally observed heterosynaptic plasticity rules.

Outline of this paper

The structure of this paper is as follows. First, we introduce the nonlinear recurrent network model and a transformation of network weights that exactly preserves task performance. Second, we formalize a notion of noise robustness in recurrent networks and show that this quantity is convex with respect to the coordinates of the task-preserving transformation. We then derive a dynamical transformation of the network, called synaptic balancing, which maximizes robustness while exactly maintaining task performance, and which is implementable entirely by local update rules. We next show that synaptic balancing is exponentially stable in any recurrent network which does not contain irreducible feedforward structure. Subsequently, we introduce several generalizations of our model and show that synaptic balancing is an instance of a broader class of integrable dynamical systems on synaptic weight matrices known as Lax dynamical systems. These dynamical systems lead to isospectral flows on matrices that preserve all eigenvalues of the matrix.

Turning to the behavior of synaptic balancing alongside task-relevant learning dynamics, we then prove that a broad class of regularized networks naturally approaches the equilibrium of our rule throrightugh training. We conversely show empirically that our rule is able to improve the task performance of trained networks in previously unseen noisy regimes.

Finally, we address the role of synaptic balancing as a candidate local plasticity rule for maintaining stable network computation in neural circuits. We present exact and approximate solutions to the trajectory of synaptic balancing, deriving a formal connection between synaptic balancing and heat diffusion in a network. In closing, we draw a connection between synaptic balancing and experimentally observed patterns of heterosynaptic plasticity in cortical synapses.

Results

A task-preserving transformation defines a manifold of equivalent recurrent networks

In this section, we first describe a class of nonlinear recurrent neural networks that has been extensively studied in diverse contexts [6, 7, 19, 23, 24, 29–31]. We then describe a natural symmetry acting on the weight space of these neural networks that *exactly preserves* the entire temporal mapping of input to output trajectories. Since the input-output mapping determines the task a neural network performs, the action of this symmetry enables us to traverse the manifold of weight configurations that preserve the task performed by any neural network.

Recurrent network model. Consider a recurrent rate network of N neurons with an $N \times N$ synaptic weight matrix $\mathbf{J}$, such that neuron j is connected to neuron i with synaptic weight J_{ij}. The vector of neural activity $\mathbf{x} \in \mathbb{R}^N$ and readout vector $\mathbf{y}$ evolve under a time-varying external input $\mathbf{u}$ as

$$\tau \dot{\mathbf{x}} = \mathbf{f}_W(\mathbf{x}, \mathbf{u}) = -\mathbf{x} + \mathbf{J}\phi(\mathbf{x}) + \mathbf{W}^{\text{in}}\mathbf{u}, \tag{1}$$

$$\mathbf{y} = \mathbf{W}^{\text{out}}\mathbf{x}. \tag{2}$$

Here τ is a fast timescale of neural dynamics, ϕ is a typically nonlinear scalar function applied element-wise to $\mathbf{x}$, and the tuple $\mathcal{W} = (\mathbf{W}^{\text{in}}, \mathbf{J}, \mathbf{W}^{\text{out}})$ is the weight configuration which parameterizes the network.

We assume for simplicity that neural activity is initialized at the origin and runs until some time T. The dynamics of the full network, and in particular the output trajectory $\mathbf{y}(t)$, are a deterministic function of the input trajectory $\mathbf{u}(t)$ and the weight configuration $\mathcal{W}$. More formally, the input-output map F_{RNN} computed by the network is the map from input to output trajectories under a particular weight configuration:

$$F_{\text{RNN}}(\mathcal{W}) : \{\mathbf{u}(t)\}_{t=0}^{T} \mapsto \{\mathbf{y}(t)\}_{t=0}^{T} \tag{3}$$

In this work, we introduce a model of synaptic updates whose properties are well behaved when ϕ is a homogeneous function; that is, when $\phi(\alpha x) = \alpha\phi(x)$ for all $x \in \mathbb{R}$ and $\alpha \in \mathbb{R}^{+}$. Common activation functions which satisfy this condition include the linear ($\phi(x) = x$) and rectified linear ($\phi(x) = \max(0, x)$) units. In the main text of this paper we assume that ϕ is homogeneous, however, in S1 Appendix. 4 we give an extension to the more general class of positively homogeneous nonlinearities, which satisfy $\phi(sx) = s^{k}\phi(x)$, for any $s > 0$ and $k \in \mathbb{R}$.

Task-preserving transformation. We now describe a symmetry in weight space that exactly preserves the map (3). Indeed, the observation that many distinct weight configurations may yield quantitatively similar network behavior is widespread both in neuroscience, where simulations of the crustacean stomatogastric ganglion found a range of synaptic strengths producing a given network output [17], and in machine learning, in which non-convex cost functions possess many roughly equivalent local minima [19, 32]. In the class of networks we study, part of the degeneracy between weights and a given task originates from a symmetry in weight space that exactly preserves the deterministic input-output map computed by the network. Intuitively, this symmetry scales neurons' inputs while reciprocally scaling their outputs, such that the overall function computed by each neuron remains intact.

More precisely, consider a map π, parameterzied by $\mathbf{h} \in \mathbb{R}^{N}$, which takes as input a weight configuration $\mathcal{W} = (\mathbf{W}^{\text{in}}, \mathbf{J}, \mathbf{W}^{\text{out}})$, and produces as an output the weight configuration

$$\pi_{\mathbf{h}}(\mathcal{W}) = (e^{-\mathbf{H}}\mathbf{W}^{\text{in}}, e^{-\mathbf{H}}\mathbf{J}e^{\mathbf{H}}, \mathbf{W}^{\text{out}}e^{\mathbf{H}}). \tag{4}$$

Here $\mathbf{H}$ denotes the diagonal matrix diag{$\mathbf{h}$}.

It is worth emphasizing some basic properties of the transformation $\pi_{\mathbf{h}}$. The sign of synapses is preserved, and synapses which are initially zero remain so. Thus, the transformation does not modify the basic wiring diagram of the network—for example by creating or removing synapses, or changing excitatory to inhibitory synapses and vice versa. Rather, it scales the strength of existing synapses by a positive quantity. In addition $\pi_{\mathbf{h}}$ acts on the recurrent weight matrix $\mathbf{J}$ through a similarity transformation. Thus the entire eigenspectrum of the recurrent weight matrix is conserved under this map.

However, $\pi_{\mathbf{h}}$ satisfies an additional important condition: it exactly preserves the entire input-output output map of the network. For this reason, we refer to $\pi_{\mathbf{h}}$ as the task-preserving transformation. More formally, for any (finite) value of $\mathbf{h}$,

$$F_{\text{RNN}}(\mathcal{W}) = F_{\text{RNN}}(\pi_{\mathbf{h}}(\mathcal{W})). \tag{5}$$

This claim is summarized in Fig 1, which shows in panels a—f that while the dynamics of the hidden units for two networks related by the task-preserving transformation are different, the output trajectories agree. Eq (5) is proved in the following proposition.

Proposition 1 (Task-preserving transformation). *The transformation (4) exactly preserves the input-output relationship of the neural dynamics (1). Given two networks receiving the same*

time course of inputs $\mathbf{u}(t)$ *and with weight configurations* $\mathcal{W}^0$ *and* $\mathcal{W} = \pi_{\mathbf{h}}(\mathcal{W}^0)$ *respectively, then*:

1. *If* $\mathbf{x}^0(t)$ *is the time course of hidden unit neural activity under* $\mathcal{W}^0$*, then* $e^{-\mathbf{H}}\mathbf{x}^0(t)$ *is the time course of hidden unit neural activity under* $\mathcal{W}$.
2. *If* $\mathbf{y}^0(t)$ *is the time course of output neural activity under* $\mathcal{W}^0$, then $\mathbf{y}^0(t)$ *is also the time course of output unit neural activity under* $\mathcal{W}$.

The proof of proposition 1 is straightforward and can be found in S1 Appendix 1.

Given an initial weight configuration $\mathcal{W}^0$, we define the *task-preserving manifold* of $\mathcal{W}^0$ to be the manifold of weight configurations accessible from $\mathcal{W}^0$ by the task-preserving transformation, i.e., the orbit of $\mathcal{W}^0$ under the symmetry $\pi_{\mathbf{h}}$: $\{\pi_{\mathbf{h}}(\mathcal{W}^0) : \mathbf{h} \in \mathbb{R}^N\}$. The vector $\mathbf{h}$ provides a set of coordinates on this manifold. When referring to the task-preserving manifold we implicitly assume there exists some reference weight configuration $\mathcal{W}^0$ at which $\mathbf{h} = 0$.

A measure of robustness to noise in neural activity

We have shown that every weight configuration on the task-preserving manifold computes the same deterministic input-output map. However, biological systems are inherently noisy [1], and it has been found that cortical spike trains vary at the sub-millisecond level across trials with identical inputs [2–4]. Interestingly, even two networks which compute the same deterministic input-output map may exhibit differing responses to neural noise. In Fig 1, we give an example of two networks which perform identically in the deterministic setting but which respond differently when noise is injected to hidden units.

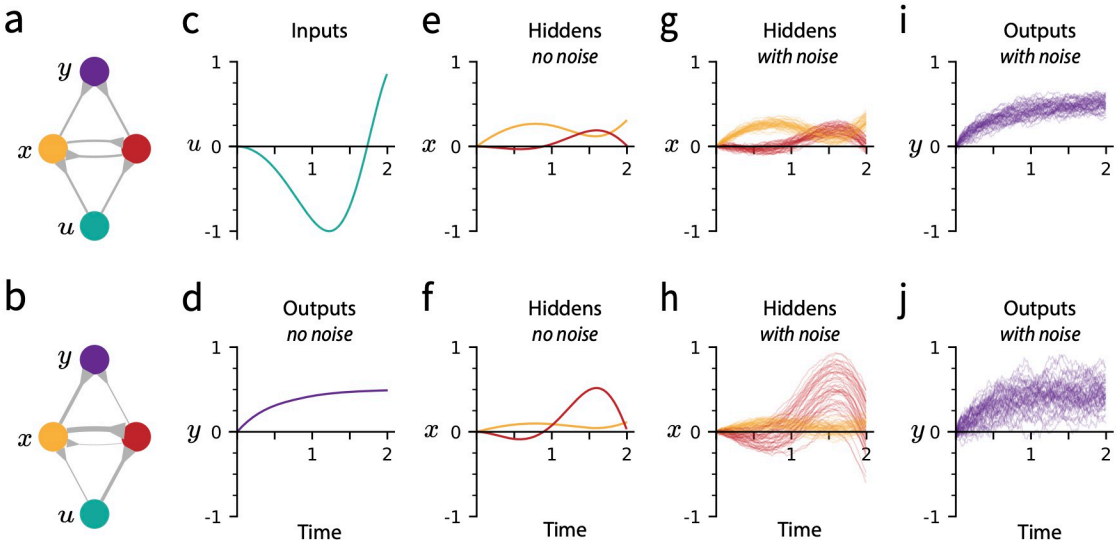

Fig 1. Two nonlinear recurrent networks consisting of input, hidden, and output neurons possess an identical input-output relationship, but behave differently in the presence of noise. (a-b) A single input u (teal) drives two recurrent neurons x_1 (yellow) and x_2 (red) which project to a single output y (purple). The connectivity patterns of the two networks are related by the task-preserving transformation (4). Line thickness denotes synaptic strength. **(c)** Input trajectory, fed to both networks. Horizontal axis in all panels is time. **(d)** Output trajectory, produced by both networks when run according to the deterministic dynamics (1). **(e-f)** Hidden unit neural activity under deterministic dynamics, for networks **a** and **b** respectively. Trajectories are identical up to a per-neuron scale factor, determined by the parameters of the task-preserving transformation. **(g-h)** Hidden neuron neural activity for networks **a** and **b** respectively when additive Gaussian noise is injected into the neural dynamics (1). 50 trials are shown. **(i-j)** Output neuron neural activity in the noisy case.

https://doi.org/10.1371/journal.pcbi.1010418.g001

In this section we introduce a quantitative notion of a network's robustness to noise, which we call sensitivity. This function describes the degree to which noise in neural activity may interfere with the underlying computation of the network. We connect our notion of sensitivity to the gain of neurons in the network and show that the sensitivity is well-behaved on the task-preserving manifold; in particular, that it possesses a convex geometry in the coordinates **h**.

Robustness of neural dynamics to random perturbations. A recurrent network whose dynamics are easily perturbed by small variations in neural activity is unlikely to perform a task robustly in the presence of neural noise. To capture this notion, we consider the magnitude of the response to a small, isotropic Gaussian perturbation $\delta \sim \mathcal{N}(0, \varepsilon^2 I)$ to the neural dynamics (1), averaged over the distribution of states $\mathcal{X}$ visited by the network during a task:

$$\frac{1}{\varepsilon^2}\langle \|\mathbf{f}(\mathbf{x}+\delta,\mathbf{u}) - \mathbf{f}(\mathbf{x},\mathbf{u})\|_2^2 \rangle_{\delta,\mathbf{x}},$$

where for simplicity we have dropped from **f** an explicit dependence on the weights $\mathcal{W}$. We define the sensitivity of a network to be the first-order approximation of this quantity,

$$S = \langle \| \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \|_F^2 \rangle_{\mathbf{x}\sim\mathcal{X}}, \tag{6}$$

where the approximation is made via Taylor expansion of **f** about **x**. Networks with lower sensitivity are less easily pushed away from their original trajectories and are therefore more likely to be robust to noise while performing tasks, as is illustrated in Fig 1(g)–1(j). In this paper, robustness refers to S^{-1}, the reciprocal of sensitivity.

In the case of the neural dynamics (1), the Jacobian matrix takes the form,

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = -\mathbf{I} + \mathbf{J}\,\mathrm{diag}\{\phi'(\mathbf{x})\}. \tag{7}$$

This expression makes use of the gain, $\phi'(x_i)$, of neuron i. With respect to the distribution of neural activity $\mathcal{X}$, the gain has first and second moments

$$\begin{aligned} \mu_i &= \langle \phi'(x_i) \rangle_{x_i \sim \mathcal{X}_i} \\ \sigma_i^2 &= \langle \phi'(x_i)^2 \rangle_{x_i \sim \mathcal{X}_i} \end{aligned} \tag{8}$$

for each neuron i. We may rewrite the definition of sensitivity (6) using the Jacobian of the neural dynamics (7) in terms of the moments (8) to obtain

$$\begin{aligned} S &= \langle \| \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \|_F^2 \rangle_{\mathbf{x}\sim\mathcal{X}} \\ &= \langle \mathrm{Tr}[(-\mathbf{I} + \mathbf{J}\,\mathrm{diag}\{\phi'(\mathbf{x})\})(-\mathbf{I} + \mathbf{J}\,\mathrm{diag}\{\phi'(\mathbf{x})\})^T] \rangle_{\mathbf{x}\sim\mathcal{X}} \\ &= \mathrm{Tr}[\mathbf{I} - \mathbf{J}\langle \mathrm{diag}\{\phi'(\mathbf{x})\}\rangle_{\mathcal{X}} - \langle \mathrm{diag}\{\phi'(\mathbf{x})\}\rangle_{\mathcal{X}}\mathbf{J}^T + \mathbf{J}\langle \mathrm{diag}\{\phi'(\mathbf{x})\}^2\rangle_{\mathcal{X}}\mathbf{J}^T] \\ &= N - 2\,\mathrm{Tr}[\mathbf{J}\langle \mathrm{diag}\{\phi'(\mathbf{x})\}\rangle_{\mathcal{X}}] + \mathrm{Tr}[\mathbf{J}\langle \mathrm{diag}\{\phi'(\mathbf{x})\}^2\rangle_{\mathcal{X}}\mathbf{J}^T] \end{aligned}$$

$$S = \sum_{ij} \sigma_j^2 J_{ij}^2 - 2\sum_i \mu_i J_{ii} + N. \tag{9}$$

The sensitivity, then, can be succinctly written as a function of the recurrent weights and the moments of the neural gains. However, because the moments μ and σ^2 are defined as an average over the distribution of network states that depends—generally intractably—on the

weights themselves, sensitivity is not, as it may appear, a straightforward quadratic function of **J**. We next show that, perhaps surprisingly, sensitivity turns out to be well behaved as a function on the task-preserving manifold.

Sensitivity is a convex function on the task-preserving manifold. Because networks attaining weight configurations of lower sensitivity might see improved task performance in noisy environments (as in Fig 1), it is useful to analytically characterize how S varies as a function of network weights. We have just mentioned the complications of studying this question in general. Here, we find that S becomes tractable when we consider its variation on the task-preserving manifold, parameterized by the coordinates **h**. In particular, we show that S is convex with respect to **h**.

Proposition 2. *When considered as a function on the coordinates* **h** *of the task-preserving manifold, the sensitivity S is a convex function*

$$S = \sum_{ij} c_{ij}^0 e^{p(h_j - h_i)} + S_{\text{const}} \tag{10}$$

for some constants $\{c_{ij}^0\}_{i,j=1}^N$, p, *and* S_{const}, *where* $c_{ij}^0 \geq 0$ *for all i, j.*

Proof. Assume there is some original network $\mathcal{W}^0 = (\mathbf{W}^{\text{in},0}, \mathbf{J}^0, \mathbf{W}^{\text{out},0})$, whose distribution of neural activity is $\mathcal{X}^0$, and whose gains have moments $\{\mu_i^0\}_{i=1}^N$ and $\{(\sigma_i^0)^2\}_{i=1}^N$. Consider a transformed network $\mathcal{W} = \pi_{\mathbf{h}}(\mathcal{W}^0)$, with analogous quantities $\mathcal{X}$, $\{\mu_i\}_{i=1}^N$, and $\{\sigma_i^2\}_{i=1}^N$. From (4), the recurrent connectivity of the transformed network is $J_{ij} = J_{ij}^0 e^{h_j - h_i}$. The sensitivity of the transformed network in terms of the task-preserving transformation is obtained by plugging this J_{ij} into (9):

$$S = \sum_{ij} \sigma_j^2 (J_{ij}^0)^2 e^{2(h_j - h_i)} - 2\sum_i \mu_i J_{ii}^0 + N. \tag{11}$$

For convexity, it is sufficient to show that the moments of the gain, μ_i and σ_i^2, are constant with respect to the task-preserving transformation; i.e., that $\mu_i = \mu_i^0$ and $\sigma_i^2 = (\sigma_i^0)^2$ for each neuron i. From Prop. 1, the transformed neural activity can be written in terms of the original rates as $x_i = e^{-h_i} x_i^0$ for all $t > 0$. By assumption, ϕ satisfies $\phi(\alpha x) = \alpha\phi(x)$, and therefore $\phi'(\alpha x) = \phi'(x)$, for all $x \in \mathbb{R}$ and $\alpha \in \mathbb{R}^+$. So $\phi'(x_i) = \phi'(e^{-h_i} x_i^0) = \phi'(x_i^0)$, and

$$\mu_i = \langle \phi'(x_i) \rangle = \langle \phi'(x_i^0) \rangle = \mu_i^0$$
$$\sigma_i^2 = \langle \phi'(x_i)^2 \rangle = \langle \phi'(x_i^0)^2 \rangle = (\sigma_i^0)^2$$

for each neuron i. Therefore the moments of the gains are constant with respect to **h**.

We conclude that (11) takes the form of (10), where the constants are $c_{ij}^0 = (\sigma_j^0)^2 |J_{ij}^0|^2 \geq 0$, $p = 2$, and $S_{\text{const}} = -2\sum_i \mu_i^0 J_{ii}^0 + N$. As (10) is a positive linear combination, with constant coefficients, of exponentiated linear functions of **h** (which are convex), then it is in turn a convex function of **h** [33, ch. 3].

Other notions of robustness. Our definition of sensitivity (6) is chosen to emphasize the aspect of recurrent networks which is typically most crucial to task performance; that is, the neural dynamics. However, there are other notions of robustness which are useful to mention. Here we briefly address how the task-preserving transformation (4), parameterized by **h**, interacts with several additional relevant notions of sensitivity.

First, consider the sensitivity $S^{\mathbf{u}\to\mathbf{y}}$ of the output trajectory $\{\mathbf{y}(t)\}$ to fluctuations in the input trajectory $\{\mathbf{u}(t)\}$. In our analysis, no choice of **h** can affect this quantity, precisely because the task-preserving transformation exactly preserves the input-output map. Put differently, this

notion of sensitivity is intrinsic to the function being computed, not to the manner in which the recurrent network computes it.

Second, consider the sensitivity $S^{\mathbf{x}\to\mathbf{y}}$ of the output trajectory $\{\mathbf{y}(t)\}$ to fluctuations in the time course of neural activity $\{\mathbf{x}(t)\}$ (Fig 1(i) and 1(j)), and respectively, the sensitivity $S^{\mathbf{u}\to\mathbf{x}}$ of neural activity $\{\mathbf{x}(t)\}$ to fluctuations in the input trajectory $\{\mathbf{u}(t)\}$. These sensitivities can be individually minimized by taking $\mathbf{h} \to -\infty$ to minimize the norm of $\mathbf{W}^{\text{out}}$ and, respectively, $\mathbf{h} \to \infty$ to minimize the norm of $\mathbf{W}^{\text{in}}$. The resulting tradeoff between $S^{\mathbf{x}\to\mathbf{y}}$ and $S^{\mathbf{u}\to\mathbf{x}}$ is separate from the problem of minimizing our choice of S, and it may be solved independently of the method we now present. In what follows, we will present a local learning rule that maximizes the robustness of the neural dynamics. We will further show that these dynamics will never result in the case of $\mathbf{h}$ or any of its elements becoming $\pm\infty$.

A local learning rule maximizes robustness while preserving task performance

We have introduced the task-preserving manifold as the set of weight configurations accessible by a symmetry transformation which preserves a given deterministic input-output map. We have also shown that the sensitivity of neural dynamics to small perturbations in activity—the generally intractable quantity S—is well behaved when constrained to the task-preserving manifold. A simple question emerges: how might networks traverse the task-preserving manifold to maximize robustness while preserving underlying task performance? To address this question, we derive gradient descent dynamics that maximize network robustness in the coordinates of the task-preserving manifold. We find, perhaps surprisingly, that these dynamics are wholly implementable by biologically plausible local computations within neurons and synapses.

From (10), the problem of maximizing robustness S^{-1} is equivalent to that of minimizing the total cost:

$$C = \sum_{ij} c_{ij}, \tag{12}$$

where c_{ij} is the synaptic cost

$$c_{ij} = \sigma_j^2 |J_{ij}|^p \tag{13}$$

$$= c_{ij}^0 e^{p(h_j - h_i)}, \tag{14}$$

with $p = 2$, and in the second line we have rewritten the synaptic costs in terms of the initial cost $c_{ij}^0 = (\sigma_i^0)^2 |J_{ij}^0|^p$ and the coordinates $\mathbf{h}$, obtained by writing out (13) in terms of the task-preserving transformation $J_{ij} = J_{ij}^0 e^{h_j - h_i}$ (4).

We call the network connected if the directed graph whose edge weights are given by the initial synaptic cost matrix $\mathbf{C}^0$ is connected. We assume, without loss of generality, that the network is connected (if not, a similar theory applies to each connected component separately).

We turn to deriving synaptic dynamics that transforms $\mathbf{h}$ over time so as to maximize robustness on the task-preserving manifold. Until now we have treated the vector $\mathbf{h}$ as a free coordinate on the task-preserving manifold. Henceforth we consider it as a function of time, initializing at the origin and evolving under gradient descent on the total cost. Similarly, we now view $\mathcal{W} = (\mathbf{W}^{\text{in}}, \mathbf{J}, \mathbf{W}^{\text{out}}) = \pi_{\mathbf{h}}(\mathcal{W}^0)$ as a time-varying weight configuration, with initial value $\mathcal{W}^0$ corresponding to $\mathbf{h} = 0$.

Let the time derivative of $\mathbf{h}$ be denoted by $\dot{\mathbf{h}} = \mathbf{g}$, which we refer to as the neural gradient vector. The dynamics of gradient descent on C with respect to $\mathbf{h}$ is given by

$$\mathbf{g} = -\gamma \frac{\partial C}{\partial \mathbf{h}}, \tag{15}$$

where $\gamma \sim \mathcal{O}(1)$ is the descent rate. We assume a separation of timescales, such that neural dynamics (1) are much faster than the weight dynamics (15).

The synaptic costs (13) obey

$$\frac{\partial c_{ij}}{\partial h_k} = p\, c_{ij}\, (\delta_{kj} - \delta_{ki}), \tag{16}$$

where δ is the Kronecker delta. In the present regime where $\gamma \sim \mathcal{O}(1)$ and $p \sim \mathcal{O}(1)$, we are free to adopt throughout $\gamma p = 1$ for notational convenience. Using this, we evaluate the right hand side of (15) via (12) and (16) to find,

$$g_k = \sum_j c_{kj} - \sum_i c_{ik}. \tag{17}$$

Finally, to see how synaptic strengths are updated, we differentiate the task-preserving transformation $J_{ij} = J^0_{ij} e^{h_j - h_i}$ with respect to time, finding that

$$\dot{J}_{ij} = J_{ij}(g_j - g_i), \tag{18}$$

$$\dot{W}^{\text{in}}_{ik} = -W^{\text{in}}_{ik} g_i \tag{19}$$

$$\dot{W}^{\text{out}}_{kj} = W^{\text{out}}_{kj} g_j \tag{20}$$

Eqs (17) through (20), along with the definition of synaptic costs (13), are self-contained and collectively comprise the dynamics of our proposed update rule, which we call synaptic balancing.

Interestingly, we find that synaptic balancing is entirely implementable by local computations in a network. First, costs (13) are computed at the synapse as a product of the square of the synaptic weight and the presynaptic average gain. Second, neural gradients (17) are computed centrally in the neuron by aggregating and comparing the costs of incoming and outgoing synapses. Third, synaptic weights (18) are updated in proportion to the difference between the presynaptic and postsynaptic neural gradients.

In this scheme we assume that the synaptic weights, synaptic costs, and neural gradients can be represented as biophysical quantities inside neurons and synapses. We suppose, as depicted in Fig 2, that in one direction a neuron traffics its gradient from the soma to incoming and outgoing synapses, and, in the other, it traffics synaptic costs from synapses to the soma. Finally, we assume that each synapse is able to measure and store certain statistics of presynaptic activity, namely, the average presynaptic gain. This assumption is in line with other models of synaptic plasticity which introduce some form of leaky integration of neural activity, e.g. the sliding threshold of the BCM rule [34, p. 288].

Notably, although the coordinates $\mathbf{h}$ and the task-preserving transformation $\pi_{\mathbf{h}}$ are of instrumental value in deriving and analyzing our rule, the biological procedure just described implements synaptic balancing without an explicit representation of $\mathbf{h}$.

Next, we study the equilibrium state of synaptic balancing and offer a simple condition under which an equilibrium exists, thereby guaranteeing the stability of our rule.

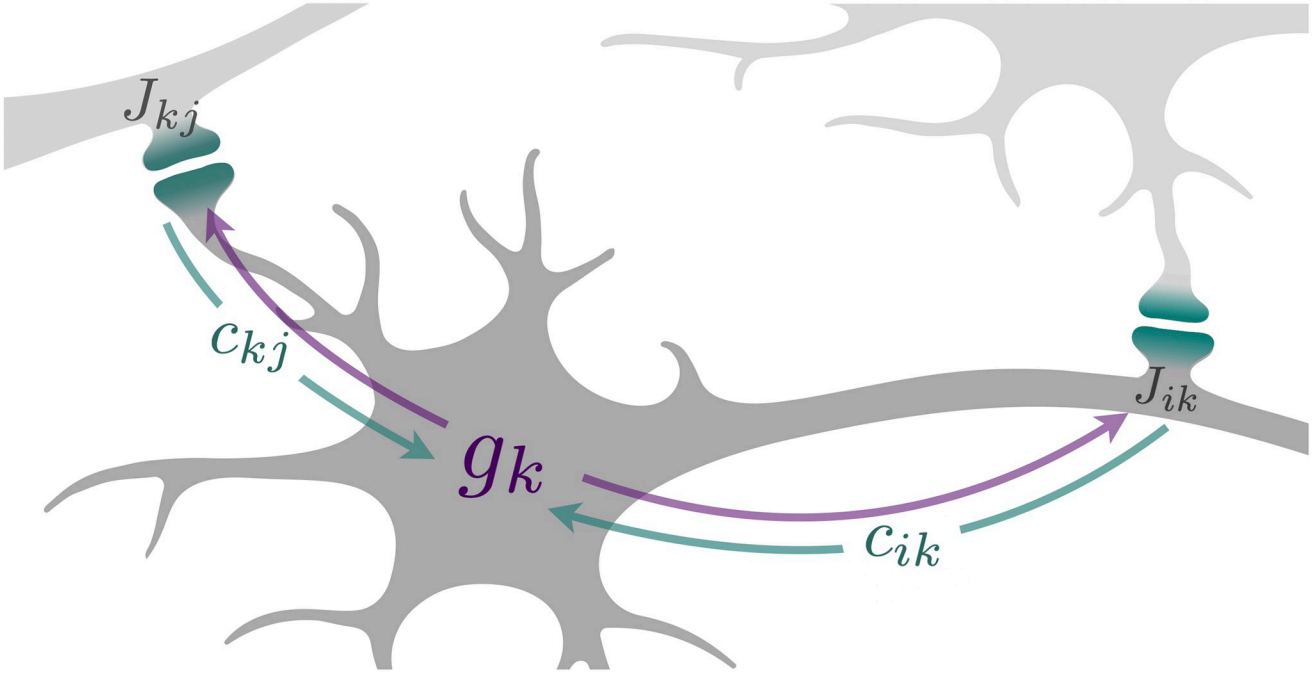

Fig 2. The local computations underlying synaptic balancing. Each synapse (indicated in teal) calculates a cost as a function of synaptic strength, as in (13). Neuron k receives signals of incoming synaptic cost c_{kj} and outgoing synaptic cost c_{ik} (teal arrows from synapses to soma) and computes the difference g_k as in (17). The signal g_k then propagates outwards (purple arrows from soma to synapses) to modify the strength of incoming and outgoing synaptic connections, as in (18), such that the total incoming and total outgoing costs are eventually balanced in every neuron.

https://doi.org/10.1371/journal.pcbi.1010418.g002

Strongly connected networks attain balanced equilibrium

Synaptic balancing asymptotically converges to a global minimum on the task-preserving manifold, by construction, because it is gradient descent on the convex function C. However, a global minimum may not exist at any finite value of the task-preserving manifold coordinates $\mathbf{h}$. For example, the convex function $f(h) = e^h$ on the real line possesses no global minimum at finite h, and the asymptotic convergence of gradient descent is not assured. It is therefore important to establish the criteria under which we expect synaptic balancing to stably converge to a (finite) minimizer $\mathbf{h}^*$. To do this, we provide two simple criteria: first, a balance condition which describes the set of equilibria of synaptic balancing, and second, a strongly-connected condition which implies the existence of an equilibrium on the task-preserving manifold that attains exponential convergence.

A weight configuration globally minimizes the total cost on the task-preserving manifold if and only if it satisfies the balance condition

$$\sum_i c_{ik} = \sum_j c_{kj} \tag{21}$$

for each neuron k. To see this, observe that because the total cost C is convex and differentiable as a function of $\mathbf{h}$, a coordinate $\mathbf{h}^*$ achieves a global minimum if and only if $\frac{\partial C}{\partial \mathbf{h}}(\mathbf{h}^*) = 0$, which, from (15), holds if and only if $\mathbf{g} = 0$. Solving for $\mathbf{g} = 0$ in (17) yields the balance condition (21).

It is because of (21) that we use the term balancing to describe our rule: the total cost is minimized, and a weight configuration is a stable fixed point of the weight dynamics (18), if and only if the total synaptic costs of each neuron's incoming and outgoing synapses are equal. It

remains unclear, however, whether the balance condition (21) is attainable in every case, or put differently, whether synaptic balancing is stable from every initial condition.

We now state a simple topological criterion which (we will show) implies the stability of synaptic balancing. A recurrent network is said to be strongly connected if between every pair of neurons i and j there exists a path of synapses, each with positive synaptic cost, from i to j. Conceptually, this condition simply means that the recurrent connectivity does not possess any embedded directed acyclic structure, such as purely feedforward connections.

With regard to biology, it is of course generally challenging to show that any given biological network is strongly connected. With that said, we believe that the highly recurrent nature of the central nervous system suggests that this assumption may not be far from reality in certain cortical regions. Long-range feedback projections as well as local recurrence in cortex suggest that non-negligible sub-networks of neurons participating in, for example, the ventral visual stream are indeed strongly connected [35, 36]. Such networks are strong candidates for synaptic balancing.

Mathematically, it is known that the connected components of any balanced matrix, i.e., a matrix satisfying (21), are strongly connected [37], and that, conversely, such matrices may be made balanced through a positive diagonal similarity transformation [38].

Before formally describing the stability of our rule, we make the preliminary observation that the synaptic costs in (14) are invariant under the transformation $\mathbf{h} \mapsto \mathbf{h} + b\mathbf{1}$, for $b \in \mathbb{R}$. Therefore the gradient of the total cost C is perpendicular to the all ones vector $\mathbf{1}$, and gradient descent on C does not explore that direction: From (17), $\mathbf{1}^T\dot{\mathbf{h}} = \sum_i g_i = 0$. Assuming as before that $\mathbf{h}^0 = 0$, synaptic balancing exclusively yields solutions that satisfy

$$\mathbf{1}^T\mathbf{h} = \mathbf{1}^T\mathbf{h}^0 = 0. \tag{22}$$

We may now connect these remarks to our weight dynamics, showing that synaptic balancing converges exponentially to a balanced configuration whenever the initial cost matrix is strongly connected. Fig 3 illustrates this point in two-neuron networks, showing the time course of synaptic balancing and the geometry of the cost function. In the first case of a single feedforward connection, a stable equilibrium is not reached at any finite $\mathbf{h}$. In the second case of recurrent connections but asymmetric initial synaptic costs, the network converges to a stable equilibrium at finite $\mathbf{h}$ and positive total cost C, with symmetric final costs. A short proof is provided in the following proposition.

Proposition 3. *In a strongly connected network, synaptic balancing converges to a globally exponentially stable equilibrium, which is the unique solution to* (22) *and* (21) *on the task-preserving manifold*.

Proof (Sketch). Because synaptic balancing does not increase the total cost $C = \sum_{ij} c_{ij}$ (12), and because the synaptic costs are nonnegative, we have that $c_{ij} \leq C^0$ along the trajectory of synaptic balancing for all i, j. If $c_{ij}^0 > 0$, then since $c_{ij} = c_{ij}^0 e^{p(h_j - h_i)}$ (14), it follows directly that $e^{h_j - h_i} \leq (C^0/c_{ij}^0)^{1/p} < \infty$. If $c_{ij}^0 = 0$, then an "indirect" upper bound for $e^{h_j - h_i}$ is obtained by collapsing the direct upper bounds along a path of positive synaptic costs from j to i: In the case of a single intervening neuron k, for example, $e^{h_j - h_i} = e^{h_j - h_k} e^{h_k - h_i} \leq (C^0/c_{kj}^0)^{1/p} (C^0/c_{ik}^0)^{1/p}$. By the strong connectedness assumption, such a path exists between every i and j, so we may in this way obtain upper bounds on $h_j - h_i$, and therefore on $|h_j - h_i|$, for all i, j. Combined with the constraint $\sum_i h_i = 0$ (22), it follows that $\mathbf{h}$ is contained in a compact set over the trajectory of synaptic balancing. We conclude that the minimizer of C, a convex function of $\mathbf{h}$, is attained in this set. To show that the minimizer is unique and globally exponentially stable, it is sufficient

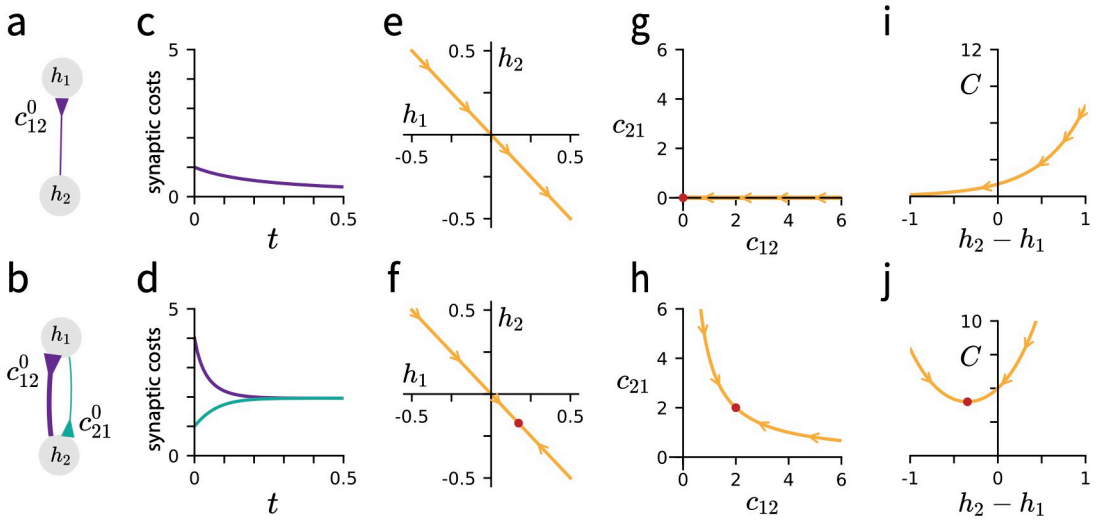

Fig 3. Network topology determines the geometry of the task-preserving manifold and the dynamics of synaptic balancing. Top row: ReLu network with two hidden units connected by a single synapse, corresponding to initial synaptic cost c_{12}^0. Bottom row: ReLu network network with two hidden units connected reciprocally, with initial synaptic costs c_{12}^0 and c_{21}^0. **(a-b)** Network diagrams showing topology and initial synaptic costs, indicated by line thickness. Input and output neurons are not shown. **(c-d)** Trajectory of synaptic costs over the course of synaptic balancing. Line colors match synapse colors in panels (a-b). Panel (c) matches (31) and panel (d) matches (29). **(e-f)** The feasible set of **h** satisfying (22), with flow lines indicating trajectory of synaptic balancing. Panel (f): Red point indicates the (finite) minimizer of the total cost. **(g-h)** Tradeoff between c_{12} and c_{21} as a function of $h_2 - h_1$, with red point indicating the optimal value of total cost $C = c_{12} + c_{21}$. **(i-j)** Total cost C as a function of $h_2 - h_1$. (i) Optimal cost is zero, attained at an infinite value of **h**. (j) Optimal cost is positive, attained at finite **h**.

https://doi.org/10.1371/journal.pcbi.1010418.g003

to show that under strong connectedness, C is strongly convex on the sublevel set of C^0. This is shown in S1 Appendix 2 with a more detailed proof.

In summary, synaptic balancing consists of local synaptic updates which are stable in any recurrent network that does not possess directed acyclic structure. By minimizing a convex cost on the task-preserving manifold, the rule maximizes the robustness of neural dynamics to noise while maintaining underlying task performance. In the following section we will mention a few useful generalizations of our model.

Generalizations and a connection to Lax dynamical systems

We have derived synaptic balancing as gradient descent on a behaviorally-relevant convex function—the sensitivity of the network to neural noise—and characterized the stability and equilibria of the rule. We now observe that our framework for synaptic balancing admits several generalizations: first, a broader class of synaptic costs, of which sensitivity is a particularly salient example, and second, a yet more general class of matrix-valued dynamical systems known as Lax dynamics, which have found widespread applications in fields from physics [9] to optimization and numerical linear algebra, where they are known as isospectral flows [10, 11].

A general class of well-behaved synaptic costs. The expression for synaptic costs (13) was derived to maximize robustness by minimizing the behaviorally relevant sensitivity function (6). Generally, however, the framework presented here admits a broad class of synaptic cost functions. If the total cost is the sum of synaptic costs, as in (12), then the synaptic cost c_{ij} may be an arbitrary fixed function of J_{ij}. Importantly, the stability of the resulting weight dynamics, and the optimality of any equilibria, are not guaranteed in general.

A sensible class of synaptic costs which are convex in $\mathbf{h}$ are the power-law costs

$$c_{ij} = \alpha_{ij}|J_{ij}|^p, \tag{23}$$

where $\alpha_{ij} \geq 0$ for all i, j, and $p > 0$. This definition includes sensitivity (13) as a special case, with $\alpha_{ij} = \sigma_j^2$ and $p = 2$. Other costs of the power-law form include the ℓ_1 and ℓ_2 matrix penalties studied in statistics and machine learning [39, Ch. 3.4]. Notably, the power-law costs (23): *i)* obey (14), *ii)* correspond to neural gradients of the form (17), and *iii)* inherit the stability result of Prop. 3, extending the findings of this work to a broader class of cost function. Robustness, specifically, is only maximized with the particular choice of costs (13).

Synaptic Lax dynamics. A yet further generalization of our synaptic update framework dispenses with a cost function entirely. A Lax dynamical system is an evolution on an $N \times N$ matrix $\mathbf{A}$ such that

$$\dot{\mathbf{A}} = [\mathbf{A}, k(\mathbf{A})]; \quad \mathbf{A}(0) = \mathbf{A}_0, \tag{24}$$

where $k : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is a matrix-valued function which depends on time only through its argument $\mathbf{A}$, and $[\cdot, \cdot]$ denotes the Lie bracket, which is defined as

$$[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}.$$

An important property of Lax dynamics is that the evolution of $\mathbf{A}$ takes the form of a time-varying similarity transformation. To see this, consider the $N \times N$ matrix $\mathbf{K}$ which evolves in time according to

$$\dot{\mathbf{K}} = \mathbf{K}k(\mathbf{A}); \quad \mathbf{K}(0) = \mathbf{I}.$$

It is straightforward to show that $\mathbf{K}(t)$ is nonsingular for all t and that the similarity transformation

$$\mathbf{A}(t) = \mathbf{K}(t)^{-1}\mathbf{A}_0\mathbf{K}(t)$$

is the unique solution of (24). As a consequence, Lax dynamical systems conserve the entire spectrum of the matrix $\mathbf{A}$ and are sometimes referred to as isospectral flows.

Synaptic balancing is a specific instance of Lax dynamics. We may rewrite (18) as

$$\dot{\mathbf{J}} = [\mathbf{J}, \operatorname{diag}\{\mathbf{g}\}], \tag{25}$$

noting that the elements of $\mathbf{g}$ depend on time only through a dependence on the elements of $\mathbf{J}$. Thus this dynamics is a special case of the general definition (24).

The Lax dynamics of synaptic balancing (25) encompass a very general form of task-preserving local learning rule. If the neural gradient g_k is allowed to depend arbitrarily on the incoming and outgoing weights of neuron k, then synaptic Lax dynamics remains confined to the task-preserving manifold and is fully locally computable in the sense of Fig 2. Further, all quantities which are conserved on the task-preserving manifold—including the spectrum of the recurrent weight matrix and the product of synaptic weights along every directed closed loop—are conserved by the synaptic Lax dynamics as well.

Synaptic Lax dynamics, if chosen in such a way to be stable, might be used to regulate any number of structural properties of the network. Any equilibrium of the synaptic Lax dynamics satisfies the equality condition $g_i = g_j$ for all pairs (i, j), which generalizes the balance condition (21). For choices of neural gradient for which this equilibrium is attainable and stable, the synaptic Lax dynamics offer a flexible framework for adapting network weights without affecting task performance. For example, [40] studies the problem of stably balancing the maximum

incoming and outgoing synaptic strengths, and [41] studies the problem of balancing the incoming and outgoing products of synapses. Each of these aims could be implemented by the synaptic Lax dynamics through proper choice of g_k. Our work unifies these efforts under a general framework of continuous-time weight dynamics (25), and, unlike previous approaches, derives from functional considerations a particular form of neural gradient that provably minimizes a behaviorally-relevant cost function on the task-preserving manifold.

Regularized networks are nearly balanced

Our focus so far has been on establishing theoretical links between the balance condition, noise-robust computation and the proposed local plasticity rule. We now turn to studying the interaction of these concepts with other forms of plasticity, such as task-relevant learning. In this section we present evidence that commonly studied classes of network are, in fact, generically in the vicinity of equilibrium, and that the balance condition (21), far from being an obscure edge case of network configurations, is in some sense a generic state. In particular, we find that the balance condition is approximately attained by any learning algorithm minimizing a very general class of regularized loss functions.

Recall that we denote by $F_{\text{RNN}}(\mathcal{W})$ the input-output map of the network (3), which takes input trajectories $\{\mathbf{u}(t)\}$ to output trajectories $\{\mathbf{y}(t)\}$ and which is parameterized by the weight configuration $\mathcal{W} = (\mathbf{W}^{\text{in}}, \mathbf{J}, \mathbf{W}^{\text{out}})$. Consider an arbitrary loss function L on the input-output map F_{RNN}. We are not concerned with the particular functional form of L—just that it depends on the weight configuration only through the input-output map. For example, the loss might measure the performance of the network on some task, with weight configurations that attain lower loss achieving better task performance.

We consider a regularized loss function L_{reg} which is the sum of the loss L and element-wise regularization of the recurrent weight matrix:

$$L_{\text{reg}}(\mathcal{W}) = L(F_{\text{RNN}}(\mathcal{W})) + \sum_{ij} \alpha_{ij} |J_{ij}|^p, \tag{26}$$

where p, $\alpha_{ij} > 0$ for all i, j. The expression for L_{reg} encompasses both ℓ_1 and ℓ_2 regularization of the recurrent weight matrix, for example, by setting $p = 1$ and $p = 2$ respectively.

Proposition 4. *Suppose L_{reg} has a local minimum at the weight configuration $\mathcal{W}^*$. Then $\mathcal{W}^*$ satisfies the balance condition (21), with synaptic costs $c_{ij} = \alpha_{ij}|J_{ij}|^p$.*

Proof. We provide a proof by contradiction. Assume that there exists a weight configuration $\mathcal{W}^0$ which is a local minimum of L_{reg} but which is not an equilibrium of synaptic balancing. Let $\mathbf{h}(t)$, $t \geq 0$, describe the trajectory of synaptic balancing initialized at $\mathcal{W}^0$, with synaptic costs $c_{ij} = \alpha_{ij}|J_{ij}|^p$. Because synaptic balancing preserves the input-output map F_{RNN}, the loss function $L(F_{\text{RNN}}(\pi_{\mathbf{h}(t)}(\mathcal{W}^0)))$ is constant as a function of t, i.e.,

$$\frac{d}{dt} L(F_{\text{RNN}}(\pi_{\mathbf{h}(t)}(\mathcal{W}^0))) = 0. \tag{27}$$

Because we have assumed that $\mathcal{W}^0$ is not an equilibrium of synaptic balancing, the total cost $C = \sum_{ij} \alpha_{ij}|J_{ij}|^p$ is strictly decreasing at $t = 0$, i.e.,

$$\frac{d}{dt} C(\pi_{\mathbf{h}(t)}(\mathcal{W}^0)) < 0. \tag{28}$$

Combining (27) and (28), we have that

$$\begin{aligned}\frac{d}{dt}L_{\text{reg}}(\pi_{\mathbf{h}(t)}(\mathcal{W}^0)) = & \frac{d}{dt}L(F_{\text{RNN}}(\pi_{\mathbf{h}(t)}(\mathcal{W}^0))) \\ & + \frac{d}{dt}C(\pi_{\mathbf{h}(t)}(\mathcal{W}^0)) \\ < & 0\end{aligned}$$

for all t and in particular, $t = 0$. Thus, L_{reg} is a decreasing function on the trajectory of synaptic balancing initialized at $\mathcal{W}^0$, and $\mathcal{W}^0$ is not a local minimum of L_{reg}. We conclude that every local minimum of L_{reg} satisfies the balance condition (21).

An inverse result—that in unregularized networks trained with gradient descent, the initial (generally nonzero) neural gradients are exactly conserved by training—has been noted in, e.g., [42, 43].

In practice, numerical optimization algorithms for network training terminate before achieving true local minima, so we do not expect every network trained with regularization to exactly exhibit the balance condition (21). In Fig 4a we show, however, that trained, ℓ_2-regularized networks exhibit neural gradients that are substantially lower than would be attained by chance, under random perturbation of rows of the weight matrix.

In understanding these results, the reader should keep in mind that under our terminology, a network may be balanced with respect to one cost function (say, the ℓ_2 cost), but not balanced with respect to another cost function (say, the robustness cost). Similarly, the form of regularization that is used during training affects the balance properties of the local minimum

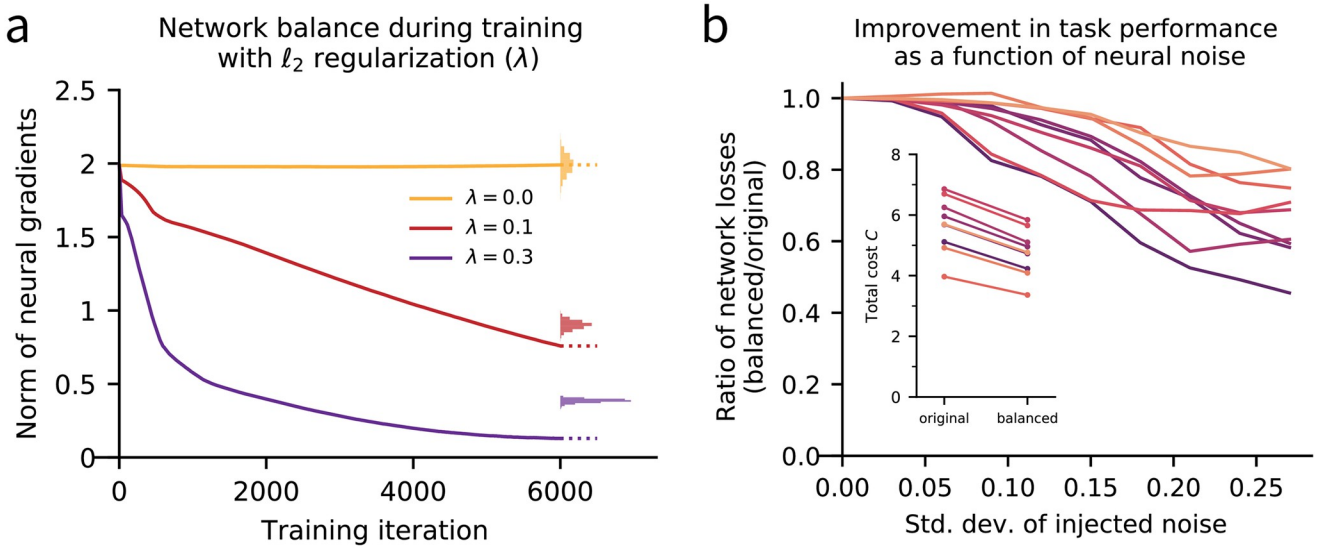

Fig 4. The balance condition in trained networks. (a) ReLu networks with $N = 256$ neurons are trained via gradient descent on a context-dependent integration (CDI) task modeled after [23], with varying levels of ℓ_2 penalty $\lambda \sum_{ij} J_{ij}^2$. The norm of neural gradients, $\|\mathbf{g}\|$, is shown over the course of training. When $\lambda = 0$, neural gradients are fixed by gradient descent dynamics. When $\lambda > 0$, trained solutions tend towards the balance condition (21), with $c_{ij} = J_{ij}^2$. Histograms at right denote the empirical null distribution of $\|\mathbf{g}\|$ under permutation of the rows of $\mathbf{C}$ at the final training iteration. For positive values of λ, the actual value of $\|\mathbf{g}\|$ (dotted line) falls significantly below the null distribution. **(b)** Synaptic balancing with the robustness cost function (13) is applied to several ReLu networks trained with $\lambda = 0.3$ (corresponding to the purple curve in panel **a**). The original and balanced networks are run on the CDI task at varying levels ε of Gaussian noise injected into hidden dynamics. For each network pair the ratio is plotted of task loss of the balanced network to that of the original network, as a function of ε. As dynamics become more noisy, the performance of the original networks (as measured by loss on the task) degrades faster than that of the corresponding balanced networks. Inset: total cost C of the original vs. balanced networks (12).

https://doi.org/10.1371/journal.pcbi.1010418.g004

reached; a network trained with ℓ_2 regularization need not satisfy the robustness balance condition.

Nonetheless, we have showed that any regularized network will become exactly balanced (in some sense) through training. This observation suggests that attaining the equilibrium condition of synaptic balancing is not only functionally desirable from a robustness standpoint but also is a generic consequence of learning rules which optimize a loss function of the form (26).

Balancing empirically improves task performance

To experimentally measure the effect of synaptic balancing on task performance, we trained networks via gradient descent on a context-dependent integration task modeled after [23]. This process yielded a trained network which adequately performed the task. We applied synaptic balancing to the trained network, using the robustness cost function (13), and stored the equilibrium weight configuration as our balanced network. We simulated the trajectories of the original (trained) versus balanced network and observed task performance while varying the levels of additive Gaussian noise in the neural dynamics (1). Details of the task and network training are provided in S1 Appendix 5.

As expected, both the original and balanced networks perform identically in the absence of noise, due to the task-preserving nature of synaptic balancing. We also found that the task performance, as measured by the trial-averaged task loss on held-out test data, deteriorated in both the trained and balanced networks as the noise level increased. However, this deterioration was noticeably attenuated in the balanced networks, and trained networks that had not been balanced proved more sensitive to higher levels of Gaussian noise. The decay in relative performance of original versus balanced networks is illustrated across several network instantiations in Fig 4b.

The improvement exhibited by the balanced network is remarkable in part because synaptic balancing does not make use of a task-specific error signal, but merely uses summary statistics of the average neuronal gain during the task. These results confirm that in networks which are already performing a task, the variability of neural responses can be suppressed simply by shifting synaptic weights towards a balanced configuration via the task-preserving transformation. As noted above, this task preserving transformation can be implemented by local synaptic learning rules that require no knowledge of the task.

Exact and approximate trajectories of synaptic balancing

We have shown that the ordinary differential Eq (18) is a member of a widely studied class of dynamical systems called Lax dynamics. In the interest of better understanding the action of our dynamics on the recurrent weight matrix as a whole, we now turn to closed-form solutions to (18) that specify the evolution of synaptic balancing over time. When the network has just two neurons, our solution is exact; in general, we derive a quadratic approximation taking the form of a heat equation.

Exact trajectory with two neurons. In a two-neuron network, the balancing dynamics admit an exact analytical solution for the time course of the synaptic costs c_{12} and c_{21}, when these costs take the power-law form (23). If both initial synaptic costs are positive, we find that they evolve as

$$\begin{aligned} c_{12}(t) &= \hat{c}_{12}\, q(t) \\ c_{21}(t) &= \hat{c}_{12}\, q(t)^{-1}, \end{aligned} \tag{29}$$

where

$$\hat{c}_{ij} = \sqrt{c_{ij}^0 c_{ji}^0} \tag{30}$$

and

$$q(t) = \tanh\left[2p^2(\hat{c}_{12})t + \tanh^{-1}(c_{12}^0/c_{21}^0)^{1/2}\right]$$

If just a single initial synaptic cost is positive—suppose it is c_{12}—then it evolves as

$$c_{12}(t) = \frac{c_{12}^0}{2c_{12}^0\gamma p^2\, t + 1} \tag{31}$$

while c_{21} is fixed at zero. In agreement with Prop. 3, the latter case suffers from unbounded growth of the input and output weights over time as $J_{12} \to 0$, and there is no stable equilibrium of the dynamics. The trajectories in (29) and (31) are illustrated in panels (g-h) of Fig 3.

Approximate trajectory via the heat equation. Except in the two-neuron scenario, we are not aware of a general analytic solution to the trajectory of synaptic balancing. Here we show that a closed-form approximate solution is attainable in general, however, shedding light on the macroscopic patterns of weight modification that synaptic balancing induces in a network. Our approach is to consider the time evolution of the neural gradients, which we tie to solutions of the heat equation on a graph. This then yields a closed-form approximate local solution for the trajectory of the coordinates $\mathbf{h}$.

The time evolution of the neural gradients (17) is given by

$$\dot{\mathbf{g}} = \frac{\partial \mathbf{g}}{\partial \mathbf{h}}\mathbf{g}. \tag{32}$$

Since $\mathbf{h}$ evolves under gradient descent (15), the Jacobian of the dynamics of $\mathbf{h}$ is a sign-flipped version of the Hessian of the cost function (12),

$$\frac{\partial \mathbf{g}}{\partial \mathbf{h}} = -\gamma \frac{\partial^2 C}{\partial \mathbf{h}^2}. \tag{33}$$

A short calculation via (16) finds the Hessian to be

$$\frac{\partial^2 C}{\partial \mathbf{h}^2} = p^2\mathbf{L}, \tag{34}$$

where $\mathbf{L}$ takes the form of a Laplacian matrix corresponding to a graph with edge weights given by what we call the conductance matrix $\bar{\mathbf{C}}$, which has ijth element

$$\bar{c}_{ij} = c_{ij} + c_{ji}, \tag{35}$$

and which is evidently symmetric. Explicitly, the Laplacian $\mathbf{L}$ has elements drawn from $\bar{\mathbf{C}}$ as follows:

$$L_{ij} = \begin{cases} -\bar{c}_{ij}, & i \neq j, \\ \sum_{k\neq i}\bar{c}_{ki}, & i = j. \end{cases} \tag{36}$$

Combining (32), (33) and (34), and maintaining $\gamma = p^{-1}$ as before, the time derivative of the neural gradients is

$$\dot{\mathbf{g}} = -p\mathbf{L}\mathbf{g}. \tag{37}$$

Eq (37) is the heat equation of the graph corresponding to $\mathbf{L}$, in analogy to the heat equation in continuous environments [44]. This suggests an intuitive physical description, and approximate mathematical solution, of the evolution of neural gradients.

Since $\mathbf{L}$ is itself time-varying, the dynamics (37) do not admit a straightforward exact solution. However, in the vicinity of a weight configuration $\tilde{\mathcal{W}}$, we may take $\mathbf{L}$ to be fixed at $\tilde{\mathbf{L}}$ to obtain a closed-form expression which approximates the exact solution to (37). This corresponds to gradient descent dynamics on the quadratic Taylor approximation of C in $\mathbf{h}$ evaluated at $\tilde{\mathcal{W}}$. Under fixed $\tilde{\mathbf{L}}$, (37) is a (time-invariant) linear dynamical system, and the solution at time t may be expressed in terms of the heat kernel $e^{-p\tilde{\mathbf{L}}t}$:

$$\begin{aligned}\mathbf{g}(t) &= e^{-p\tilde{\mathbf{L}}t}\mathbf{g}^0 \\ &= \sum_{i:\lambda_i>0} e^{-p\lambda_i t}\mathbf{v}^i(\mathbf{v}^i)^T\mathbf{g}^0,\end{aligned} \tag{38}$$

where $\mathbf{g}^0$ denotes the neural gradient at time $t = 0$, and λ_i, $\mathbf{v}^i$ denote the ith eigenvalue and eigenvector of $\tilde{\mathbf{L}}$. The sum is taken over all i such that λ_i is positive, since if $\lambda_i = 0$ for some i then $\mathbf{v}^i$ is an indicator vector of a connected component, which must be orthogonal to any realizable neural gradient $\mathbf{g}^0$.

Integrating (38) yields a closed-form approximate expression for the evolution of the coordinates $\mathbf{h}$ on the task-preserving manifold:

$$\mathbf{h}(t) = \sum_{i:\lambda_i>0} (p\lambda_i)^{-1}(1 - e^{-p\lambda_i t})\mathbf{v}^i(\mathbf{v}^i)^T\mathbf{g}^0, \tag{39}$$

where we have chosen boundary conditions such that $\mathbf{h}(0) = 0$. As $t \to \infty$, the network reaches an equilibrium $\mathbf{h} \to \mathbf{h}^*$, given by

$$\mathbf{h}^* = p^{-1}\tilde{\mathbf{L}}^\dagger\mathbf{g}^0. \tag{40}$$

The notation $\tilde{\mathbf{L}}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\tilde{\mathbf{L}}$.

To illustrate these concepts, Fig 5 shows how a 12-neuron, ring topology network, redistributes synaptic cost by synaptic balancing dynamics in response to a perturbation of a single synaptic cost (a—b), as well as the evolution of $\mathbf{h}$ in the Laplacian matrix eigenmode basis (c-d). The approximate closed-form dynamics derived in this section, and especially the role of the Laplacian matrix $\mathbf{L}$, help shape intuition about the behavior of our rule: Neural gradients diffuse in a network akin to heat diffusing on a graph, with higher spatial-frequency modes of the graph decaying more quickly than lower spatial-frequency modes. As shown in Fig 5(d), numerical simulations verify that the quadratic approximate dynamics presented here accurately describe synaptic trajectories near equilibrium.

General bounds on equilibrium cost. We have provided exact and approximate closed-form dynamics of synaptic balancing. To complement these results, we now turn to results on the optimal value of the cost function, stated in terms of the initial cost matrix $\mathbf{C}^0$.

Proposition 5. *Suppose synaptic costs are of the power-law form (23). Then given initial neural gradients $\mathbf{g}^0$ and initial total cost C^0, the minimum total cost C^* on the task-preserving manifold is bounded above by*

$$C^* \leq C^0 - \frac{1}{8C^0}\|\mathbf{g}^0\|_2^2, \tag{41}$$

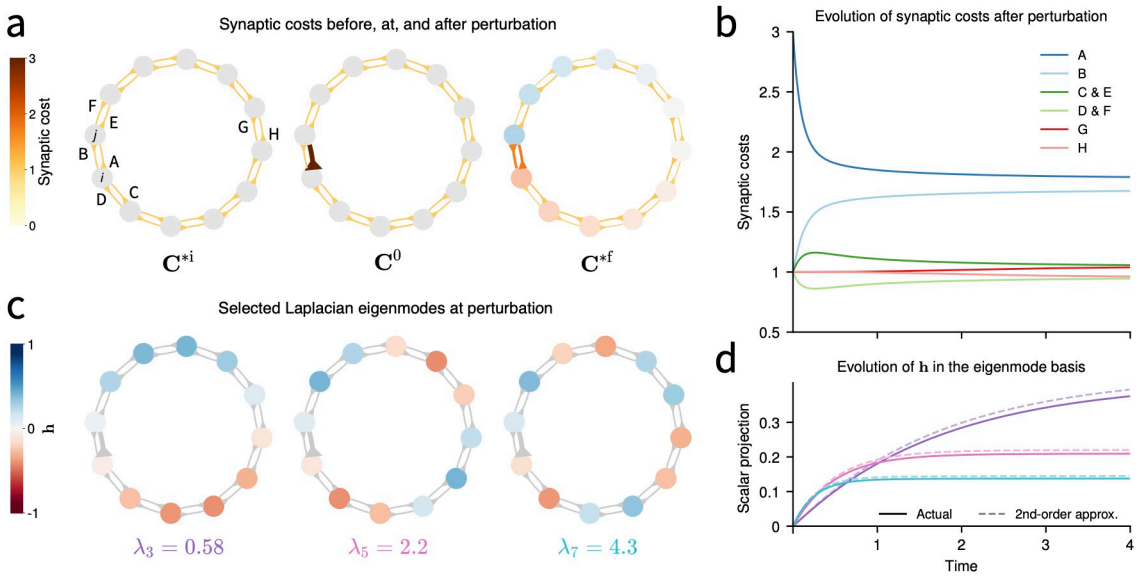

Fig 5. Dynamics of synaptic balancing in a 12-neuron ring network with single perturbed synapse. (a). Left: A ring network is at an initial equilibrium $\mathbf{C}^{*i}$ with all synaptic costs equal to 1. Nodes indicate neurons; arrows indicate directed synapses. Center: Synapse A, from neuron j to i, is instantaneously potentiated to a synaptic cost of $c_{ij}^0 = 3$. Right: Synaptic balancing relaxes to a new equilibrium $\mathbf{C}^{*f}$. Synapse colors and thickness indicate synaptic cost. Neuron colors indicate value of $\mathbf{h}^{*f}$ according to color scheme of panel (c). **(b)** Time course of synaptic balancing following perturbation. Incoming synapses to neuron i (A and D) are weakened and outgoing synapses from neuron i (B and C) are strengthened. Incoming synapses to neuron j (B and E) are strengthened and outgoing synapses from neuron j (A and F) are weakened. Synapses (H and G) that are distant from the site of perturbation respond more slowly than proximate synapses though they reach the same equilibrium values. **(c)** Three eigenmodes $\mathbf{v}_3$, $\mathbf{v}_5$, $\mathbf{v}_7$, with eigenvalues λ_3, λ_5, λ_7, of the Laplacian matrix $\mathbf{L}^0$ corresponding to the conductance matrix at the moment of perturbation. Color indicates mode value at each neuron. **(d)** Dynamics of $\mathbf{h}$ approximately decompose into the basis of Laplacian eigenmodes. The scalar projection of $\mathbf{h}$ onto each mode $\mathbf{v}$ is shown along with the quadratic approximation (39), using $\mathbf{L}^0$ as Laplacian. Line color matches eigenvalue color in panel (c).

https://doi.org/10.1371/journal.pcbi.1010418.g005

and below by

$$\sum_{ij} \hat{c}_{ij} \leq C^*, \tag{42}$$

where $\hat{c}_{ij} = \sqrt{c_{ij}^0 c_{ji}^0}$ *for all i, j, as in* (30).

A proof is given in S1 Appendix. Intuitively, the upper bound (41) says that the greater the initial deviation from the balance condition (in the form of large neural gradients), the greater the guarantee that synaptic balancing will improve the total cost. The lower bound (42) says that the more symmetric the initial cost matrix, the less that synaptic balancing is able to improve the costs.

Recall that in a two-neuron network, the equilibrium cost matrix is symmetric (29). In fact, we find that symmetry is preferred by synaptic balancing, and the equilibrium cost matrix will be symmetric in any circumstance in which symmetry is attainable on the task-preserving manifold. If $\mathbf{C}^0$ is the initial matrix of synaptic costs, and $\mathbf{C}^0$ is positive diagonally symmetrizable, then the minimum is attained at the cost matrix $\hat{\mathbf{C}} = e^{-\mathbf{H}^*}\mathbf{C}^0 e^{\mathbf{H}^*}$, whose i, jth element is the geometric mean of initial costs (30), and equality is attained in (42).

As an example of symmetric equilibrium beyond the case of $N = 2$, consider a rank-one network whose initial costs factor as $\mathbf{C}^0 = \mathbf{a}\mathbf{b}^T$. This network is strongly connected only if every

element of **a** and **b** is positive, in which case the equilibrium cost matrix is $\mathbf{C}^* = \mathbf{a}^*\mathbf{b}^{*T}$, where $a_i^* = b_i^* = \sqrt{a_i b_i}$.

It is not just symmetric matrices which are fixed points of synaptic balancing: all normal matrices—including symmetric matrices as a special case—are equilibria of our rule, i.e., they satisfy the balance condition (21). This more general result is shown in S1 Appendix.

In purely linear networks, arbitrary similarity transformations of the recurrent matrix—not just positive diagonal transformations—are task-preserving. The positive diagonal similarity transformation (4), then, will generically not achieve the optimal sensitivity among task-equivalent *linear* networks. It would be interesting to explore how optimizing over similarity transforms involving all invertible matrices might perform relative to our purely diagonal similarity transformation. However, we note that such an optimization will not generically lead to a biologically plausible local learning rule.

Synaptic balancing predicts heterosynaptic plasticity

Previous sections demonstrated that the balance condition (21) is not only functionally useful for noise robustness but also in a sense generic in trained, regularized recurrent networks. These findings suggest that synaptic balance may be a widespread phenomenon, with many networks attained through regularized training naturally exhibiting the equilibrium condition of synaptic balancing (21). In a biological context, where we hypothesize that synaptic balancing would continually operate alongside other weight dynamics, we predict that the network is generically in a state fluctuating around the balance condition. For example, fast-acting, inherently unstable plasticity rules—Hebbian or otherwise—might temporarily bring synapses and firing rates away from equilibrium, while synaptic balancing slowly tunes the network in response to those modifications, so that the balance condition is always approximately maintained.

This suggests that synaptic balancing might most commonly play a role in fine-tuning networks that are already in the vicinity of a balanced equilibrium. To explore this scenario in a specific, experimentally testable regime, we consider the response of synaptic balancing to a single synaptic potentiation or depression near equilibrium, as depicted for a ring network case in Fig 5.

Synaptic perturbations induce compensatory heterosynaptic plasticity. Suppose the that network begins at an initial equilibrium configuration $\mathcal{W}^{*\mathrm{i}}$ satisfying the balance condition (21), and that a perturbation is applied. Specifically, let the i, jth synapse ($i \neq j$) be instantaneously modified by a small factor $\eta \approx 0$ while every other synapse is unchanged. Call the perturbed weight configuration $\mathcal{W}^0$. Then

$$J_{kl}^0 = J_{kl}^{*\mathrm{i}}(1 + \eta\delta_{ik}\delta_{jl}). \tag{43}$$

If $\eta > 0$, this perturbation corresponds to synaptic potentiation of J_{ij}, and if $\eta < 0$, it corresponds to synaptic depression.

By first-order expansion of the power-law synaptic costs (23), we have that the costs adjust as

$$c_{kl}^0 = c_{kl}^{*\mathrm{i}}(1 + \eta p\delta_{ik}\delta_{jl}). \tag{44}$$

Plugging the perturbed synaptic costs (44) into the expression for the neural gradient (17), and using that the network was initialized at equilibrium, i.e. that $g_k^{*\mathrm{i}} = 0$ for all k, we have,

$$g_k^0 = \eta p c_{ij}^{*\mathrm{i}}(\delta_{ik} - \delta_{jk}). \tag{45}$$

Finally, by plugging (43) and (45) into (18), the learning rule becomes, to first order,

$$\dot{J}_{kl} = \eta p J^{*\mathrm{i}}_{kl} c^{*\mathrm{i}}_{ij} (\delta_{il} - \delta_{jl} + \delta_{jk} - \delta_{ik}) \tag{46}$$

This rule predicts that a perturbation from equilibrium at a single synapse from j to i will lead to multiplicative plasticity at the outgoing and incoming synapses of both neurons. Specifically, if J_{ij} is potentiated, i.e., if $\eta > 0$, then the response of synaptic balancing is potentiation of the presynaptic neuron's incoming synapses (J_{jl}) and of the postsynaptic neurons' outgoing synapses (J_{ki}), as well as depression of the presynaptic neuron's outgoing synapses (J_{kj}) and the postsynaptic neuron's incoming synapses (J_{il}).

Following the perturbation from the initial equilibrium configuration $\mathcal{W}^{*\mathrm{i}}$, the system will relax under synaptic balancing to a final equilibrium configuration $\mathcal{W}^{*\mathrm{f}}$ (Fig 5(a) and 5(b)). To calculate the change in synaptic strength of neuron ij from its initial equilibrium to its final equilibrium value, we plug the value of $\mathbf{g}^0$ from (45) into (40) to find

$$\mathbf{h}^{*\mathrm{f}} = \eta c^{*\mathrm{i}}_{ij} \mathbf{L}^{\dagger} (\mathbf{e}_i - \mathbf{e}_j), \tag{47}$$

where $\mathbf{e}_k$ is the kth standard basis vector of $\mathbb{R}^N$, and we adopt the Laplacian matrix corresponding to the perturbed weight configuration $\mathcal{W}^0$. Writing the task-preserving transformation $J^{*\mathrm{f}}_{ij} = J^0_{ij} e^{h^{*\mathrm{f}}_j - h^{*\mathrm{f}}_i}$ in terms of (47), then the log change in synaptic weight, normalized by the perturbation η, is

$$\begin{aligned} \frac{1}{\eta} \log\left(\frac{J^{*\mathrm{f}}_{ij}}{J^0_{ij}}\right) &= -c^0_{ij}((L^{\dagger})_{ii} + (L^{\dagger})_{jj} - 2(L^{\dagger})_{ij}) \\ &= -c^0_{ij} R^0_{ij} \end{aligned} \tag{48}$$

where $R^0_{ij} = (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - 2(L^{\dagger})_{ij} \geq 0$ is the resistance distance (alternately called effective resistance) between neurons i and j, on the graph whose edges have conductances $\bar{\mathbf{C}}^0$. Resistance distance generalizes to arbitrary graphs the formulas for resistance in series and parallel, and R_{ij} is greater where the paths of conductivity between neurons i and j are fewer or weaker [45]. We may interpret (48) to mean that synaptic balancing counteracts a potentiation or depression of J_{ij} by a factor equal to the fraction of overall conductance between neurons j and i that is attributable to the synaptic cost c^0_{ij}, versus to other paths of synaptic costs in the network.

Slow, compensatory, and network-wide heterosynaptic plasticity. We have described synaptic balancing near equilibrium as a slow, compensatory mechanism that adjusts a neuron's input and output synapses in response to a single potentiation or depression, in a manner closely resembling known phenomena of heterosynaptic plasticity [12, 13]. Here we discuss experimental evidence in connection with the predictions of synaptic balancing.

Studies inducing Hebbian plasticity at one or more target synapses have repeatedly observed compensatory heterosynaptic modification at nearby synapses on the same dendrite [14–16, 46, 47]. For example, two-photon *in vivo* imaging of synaptic spines revealed that heterosynaptic depression of inputs to mouse V1 layer 2/3 pyramidal neurons follows functionally induced potentiation (LTP) of synapses on the same dendrite—a result exactly predicted by Eq (46).

Our specific predictions about the magnitude and direction of heterosynaptic effects also find experimental support. Eq (46) predicts heterosynaptic modifications that are multiplicative, with the change in weight proportional to the original size of the synapse, and

independent of the synapse's sign (i.e., inhibitory or excitatory). Patch-clamp recordings of synapses in cortical slice after induction of spike-timing-dependent Hebbian plasticity in paired synapses matched the multiplicative principle, with the heterosynaptic effect applying more strongly to the initially larger unpaired synapses than to the initially smaller ones [16]. The same work found that heterosynaptic effects could be explained solely in terms of the absolute strengths of the paired and unpaired synapses: both E and I synapses experienced compensatory heterosynaptic depression when the paired synapse was potentiated, and both E and I unpaired synapses experienced compensatory potentiation when the paired synapse was depressed.

Our rule may be distinguished from existing concepts of compensatory heterosynapstic plasticity [13, 28] through a further prediction, to our knowledge not yet experimentally explored: in response to the potentiation of a single neuron's inputs, not only do input synapses experience heterosynaptic depression, but also output synapses experience potentiation. This concept is demonstrated in Fig 5 for a 12-neuron ring network that experiences an instantaneous synaptic perturbation.

Mechanisms supporting the input-synapse half of our conjectured balancing dynamics are well studied. In addition to the known heterosynaptic effects already mentioned, neurons are known to multiplicatively and bidirectionally scale all incoming synapses through synaptic scaling [25, 26, 48, 49]. While the particular homeostatic trigger posited by synaptic scaling—deviations from a set-point firing rate—differs from the trigger considered in our model, synaptic scaling lends evidence to the hypothesis that a neural mechanism exists to distribute a negative feedback signal to incoming synapses and induce coordinated compensatory plasticity, such as is predicted by synaptic balancing.

This work introduces a new view on the possible functional roles of compensatory heterosynaptic plasticity. Existing literature has largely focused on the important role it may play in constraining the inherent instability of Hebbian plasticity [12, 13, 28, 50]. In our model, the role of heterosynaptic plasticity is to maintain a functional state of noise robustness even as other learning processes homosynaptically modify synapse strength. We believe that these traits are suitably thought of as a novel model of homeostatic plasticity: one whose aim is to maintain, through negative feedback, the functionally-relevant balance condition (21).

Methods

Network training details

We constructed a context-dependent integration task with three input variables: the context $\mathbf{a}$ and two signals $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$. Each variable was encoded as a pair of indicator-style inputs, yielding six dimensions of network inputs. Each pair (a_1, a_2), $(s_1^{(1)}, s_2^{(1)})$, and $(s_1^{(2)}, s_2^{(2)})$ independently took the value (0, 1) or (1, 0), plus Gaussian noise, for the duration ($T = 50$) of each trial. Three Boolean variables corresponded to eight total trial conditions.

The task target $\mathbf{z}$ (also encoded as indicator variables z_1, z_2) was the integral of the noisy signal over time, as gated by the context:

$$z_i(t) = \begin{cases} \int_0^t s_i^{(1)}(t')dt', & \mathbf{a} = (0,1), \\ \int_0^t s_i^{(2)}(t')dt', & \mathbf{a} = (1,0). \end{cases}$$

for $i = 1, 2$.

Networks of the form (1), with rectified linear activation functions, were initialized with i.i.d. weights $J_{ij} \sim \mathcal{N}(0, N^{-1})$ and were trained via gradient descent to minimize an objective

consisting of the squared-error loss function $\sum_{i,t}(y_i(t) - z_i(t))^2$, plus a regularization term $\lambda\sum_{ij} J_{ij}^2$. Networks in Fig 4, with $N = 256$ neurons, learned the task after being trained with a fixed learning rate of 0.003 for 1,600 training iterations.

Average gains $\{\sigma_j^2\}_{j=1}^N$ were calculated across conditions and trials on a separate trial dataset not used in the training or evaluation of the network. The balanced network was taken to be the final time step of a numerical solution to the ODE (18), using neural gradients (17) and synaptic costs (13), with the original (trained) network as initial condition. Fig 4b was attained by running the original and balanced networks with varying levels of additive Gaussian noise injected into the recurrent neural dynamics.

Discussion

Lastly, we would like to discuss some important observations, consequences, and limitations of this work. One particularly notable drawback of this work is the lack of any theoretical guarantees of improvement in the robustness to noise of the output trajectory $\{\mathbf{y}(t)\}$ to fluctuations in either the input trajectory or the internal units. Instead, we study the sensitivity of the neural dynamics to noise in the internal units, a quantity that is generally considered crucial to task performance but allows us to study noise robustness in a task-agnostic way. This paper has intentionally focused on the case of recurrent networks, where task performance is dominated by the compounded dynamics of the recurrent matrix feeding into itself over time. In the limit of large trial time, we postulate that it is these recurrent dynamics which contribute most essentially to task performance. In this light, we believe the apparent feedforward structure of $\mathbf{W}^{\text{in}} \to \mathbf{J} \to \mathbf{W}^{\text{out}}$ distracts from the essential computations at hand. In other words, since noise in the recurrent dynamics leads to compounding errors over time, we expect that the sensitivity of the neural dynamics $S^{\mathbf{x}\to\mathbf{x}}$ (using the notation of section) matters more than $S^{\mathbf{x}\to\mathbf{y}}$ in the limit of large trial time.

As we show that regularized networks lead to more balanced solutions, one may ask if a biological system would be better off implementing a form of regularization rather than our local synaptic update rule. Synaptic balancing explores a limited manifold with N degrees of freedom, where the underlying input-output function is fixed. In contrast, network training may optimize over the entire input, recurrent, and output matrices. Even highly regularized loss functions, which effectively constrain the degrees of freedom available to the optimization, allow the network to optimize over different input-output functions. Our analysis of synaptic balancing as a dynamical system reveals a connection between learning in noise-robust systems and heterosynaptic plasticity. This connection is new to our knowledge and merits further investigation. Furthermore, we believe our learning rule will be significantly better than weight decay after a task is learned and no more training data for that task is arriving to further maintain task performance (i.e. our memory of how to ride a bike when we haven't ridden a bike in some time). If we use weight decay in the post learning regime, then weight decay will clearly destroy task performance (we will forget how to ride bikes during periods when we are not riding bikes). In contrast, a task-preserving plasticity rule, such as synaptic balancing, would move trained circuits to solutions that are more robust to internal noise even when the task is not actively being trained without destroying any learned information (causing us to forget how to ride a bike).

Lastly, the crucial aspects of synaptic balancing that it provably preserves the exact input-output relation and to do so explores a limited manifold with N degrees of freedom has consequences for the efficacy of this synaptic update rule for high-dimensional networks. We show in appendix Fig 1 that the relative reduction in total cost (a proxy for sensitivity on the task-preserving manifold) decreases with the rank of the initial synaptic weight matrix. It is possible

that this effect of modest improvements for high-dimensional networks could be mitigated by relaxing this requirement slightly, and transforming the network in a way that only approximately preserves the task. Perhaps an approximate rule inspired by this work could result in larger improvements in robustness as a function of initial rank, while any damage done to the task performance could be corrected with a small amount of retraining. We believe this would be an interesting future research direction.

Summary

In this work, we introduced a positive diagonal similarity transformation of the recurrent weight matrix that preserves task performance in nonlinear recurrent networks with homogeneous nonlinearities. We showed that a simple class of cost functions, notably including the sensitivity of the neural dynamics to noise, are convex in the coordinates of the symmetry. From this observation, we derived a local learning rule—synaptic balancing—that globally maximizes network robustness whenever the recurrent network is strongly connected.

We found that the synaptic cost matrix is balanced at equilibrium, and that this balance condition arises at every local minimum of a very general class of regularized loss functions. To further understand how synaptic balancing dynamics shape network connectivity in the vicinity of equilibrium, we approximated the dynamics of our rule through a heat equation and described the diffusion of synaptic modifications throughout the network according to its Laplacian eigenmodes. We found that near equilibrium, synaptic balancing is well summarized as slow, compensatory, and heterosynaptic, and that experimental evidence of heterosynaptic plasticity is consistent with our predictions.

Overparameterization in neural network models may be linked to the biological processes which sustain task performance under noisy conditions—a possibility known to experimental neuroscience for some time [51]. Here, we have provided a concrete, analytically tractable example of this concept, in which an identifiable symmetry in network parameterization gives rise to a corresponding local process for maintaining stable task performance. We hope that this work may provide a fruitful framework for future research relating homeostatic processes to the mathematical structures underpinning neural network redundancy.

Supporting information

S1 Appendix. Additional proofs. Full proofs of Propositions 1, 3 and 5.
(PDF)

Acknowledgments

The authors wish to thank Brandon Benson, Subhaneil Lahiri, and Jonathan Timcheck for helpful discussions and feedback. C.H.S. thanks the Blavatnik Family Foundation. S.E.H. thanks the National Defense Science and Engineering Graduate Fellowship and the Stanford Graduate Fellowship. S.G. thanks the James S. McDonnell and Simons Foundations, NTT Research, and an NSF CAREER Award for support.

Author Contributions

Conceptualization: Christopher H. Stock, Samuel A. Ocko.

Data curation: Christopher H. Stock.

Formal analysis: Christopher H. Stock, Sarah E. Harvey.

Funding acquisition: Surya Ganguli.

Investigation: Christopher H. Stock, Sarah E. Harvey, Samuel A. Ocko.

Methodology: Christopher H. Stock, Sarah E. Harvey, Samuel A. Ocko.

Project administration: Christopher H. Stock, Sarah E. Harvey.

Resources: Surya Ganguli.

Software: Christopher H. Stock.

Supervision: Samuel A. Ocko.

Validation: Christopher H. Stock.

Visualization: Christopher H. Stock, Sarah E. Harvey.

Writing – original draft: Christopher H. Stock.

Writing – review & editing: Christopher H. Stock, Sarah E. Harvey, Samuel A. Ocko, Surya Ganguli.

References

1. Faisal AA, Selen LP, Wolpert DM. Noise in the nervous system. Nature reviews neuroscience. 2008; 9 (4):292–303. https://doi.org/10.1038/nrn2258 PMID: 18319728
2. Tolhurst DJ, Movshon JA, Dean AF. The statistical reliability of signals in single neurons in cat and monkey visual cortex. Vision research. 1983; 23(8):775–785. https://doi.org/10.1016/0042-6989(83)90200-6 PMID: 6623937
3. Mainen ZF, Sejnowski TJ. Reliability of spike timing in neocortical neurons. Science. 1995; 268 (5216):1503–1506. https://doi.org/10.1126/science.7770778 PMID: 7770778
4. Shadlen MN, Newsome WT. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. Journal of neuroscience. 1998; 18(10):3870–3896. https://doi.org/10.1523/JNEUROSCI.18-10-03870.1998 PMID: 9570816
5. Rumyantsev OI, Lecoq JA, Hernandez O, Zhang Y, Savall J, Chrapkiewicz R, et al. Fundamental bounds on the fidelity of sensory cortical coding. Nature. 2020; 580(7801):100–105. https://doi.org/10.1038/s41586-020-2130-2 PMID: 32238928
6. Ganguli S, Huh D, Sompolinsky H. Memory traces in dynamical systems. Proceedings of the national academy of sciences. 2008; 105(48):18970. https://doi.org/10.1073/pnas.0804451105 PMID: 19020074
7. Ganguli S, Sompolinsky H. Short-term memory in neuronal networks through dynamical compressed sensing. In: Neural Information Processing Systems (NIPS); 2010.
8. Kadmon J, Timcheck J, Ganguli S. Predictive coding in balanced neural networks with noise, chaos and delays. Advances in neural information processing systems. 2020; 33.
9. Lax PD. Integrals of nonlinear equations of evolution and solitary waves. Communications on pure and applied mathematics. 1968; 21(5):467–490. https://doi.org/10.1002/cpa.3160210503
10. Helmke U, Moore J. Optimization and dynamical systems. Springer-Verlag London Ltd.; 1994.
11. Chu MT. Linear algebra algorithms as dynamical systems. Acta numerica. 2008; 17:1–86. https://doi.org/10.1017/S0962492906340019
12. Chistiakova M, Bannon NM, Bazhenov M, Volgushev M. Heterosynaptic plasticity: multiple mechanisms and multiple roles. The Neuroscientist. 2014; 20(5):483–498. https://doi.org/10.1177/1073858414529829 PMID: 24727248
13. Chistiakova M, Bannon NM, Chen JY, Bazhenov M, Volgushev M. Homeostatic role of heterosynaptic plasticity: models and experiments. Frontiers in computational neuroscience. 2015; 9:89. https://doi.org/10.3389/fncom.2015.00089 PMID: 26217218
14. Oh WC, Parajuli LK, Zito K. Heterosynaptic structural plasticity on local dendritic segments of hippocampal CA1 neurons. Cell reports. 2015; 10(2):162–169. https://doi.org/10.1016/j.celrep.2014.12.016 PMID: 25558061
15. El-Boustani S, Ip JP, Breton-Provencher V, Knott GW, Okuno H, Bito H, et al. Locally coordinated synaptic plasticity of visual cortex neurons in vivo. Science. 2018; 360(6395):1349–1354. https://doi.org/10.1126/science.aao0862 PMID: 29930137

16. Field RE, D'amour JA, Tremblay R, Miehl C, Rudy B, Gjorgjieva J, et al. Heterosynaptic Plasticity Determines the Set Point for Cortical Excitatory-Inhibitory Balance. Neuron. 2020;. https://doi.org/10.1016/j.neuron.2020.03.002 PMID: 32213321
17. Prinz AA, Bucher D, Marder E. Similar network activity from disparate circuit parameters. Nature neuroscience. 2004; 7(12):1345–1352. https://doi.org/10.1038/nn1352 PMID: 15558066
18. Naumann EA, Fitzgerald JE, Dunn TW, Rihel J, Sompolinsky H, Engert F. From whole-brain data to functional circuit models: the zebrafish optomotor response. Cell. 2016; 167(4):947–960. https://doi.org/10.1016/j.cell.2016.10.019 PMID: 27814522
19. Maheswaranathan N, Williams AH, Golub MD, Ganguli S, Sussillo D. Universality and individuality in neural dynamics across large populations of recurrent networks. Advances in neural information processing systems. 2019; 2019:15629–15641. PMID: 32782422
20. Gao P, Ganguli S. On simplicity and complexity in the brave new world of large-scale neuroscience. Current opinion in neurobiology. 2015; 32:148–155. https://doi.org/10.1016/j.conb.2015.04.003 PMID: 25932978
21. Barak O, Sussillo D, Romo R, Tsodyks M, Abbott L. From fixed points to chaos: three models of delayed discrimination. Progress in neurobiology. 2013; 103:214–222. https://doi.org/10.1016/j.pneurobio.2013.02.002 PMID: 23438479
22. Sussillo D, Churchland MM, Kaufman MT, Shenoy KV. A neural network that finds a naturalistic solution for the production of muscle activity. Nature neuroscience. 2015; 18(7):1025–1033. https://doi.org/10.1038/nn.4042 PMID: 26075643
23. Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature. 2013; 503(7474):78–84. https://doi.org/10.1038/nature12742 PMID: 24201281
24. Barak O. Recurrent neural networks as versatile tools of neuroscience research. Current opinion in neurobiology. 2017; 46:1–6. https://doi.org/10.1016/j.conb.2017.06.003 PMID: 28668365
25. Turrigiano GG. The self-tuning neuron: synaptic scaling of excitatory synapses. Cell. 2008; 135(3):422–435. https://doi.org/10.1016/j.cell.2008.10.008 PMID: 18984155
26. Turrigiano GG. The dialectic of Hebb and homeostasis. Philosophical transactions of the royal society B: biological sciences. 2017; 372(1715):20160258. https://doi.org/10.1098/rstb.2016.0258 PMID: 28093556
27. Chen JY, Lonjers P, Lee C, Chistiakova M, Volgushev M, Bazhenov M. Heterosynaptic plasticity prevents runaway synaptic dynamics. Journal of Neuroscience. 2013; 33(40):15915–15929. https://doi.org/10.1523/JNEUROSCI.5088-12.2013 PMID: 24089497
28. Zenke F, Gerstner W. Hebbian plasticity requires compensatory processes on multiple timescales. Philosophical transactions of the royal society B: biological sciences. 2017; 372(1715):20160259. https://doi.org/10.1098/rstb.2016.0259 PMID: 28093557
29. Vogels TP, Rajan K, Abbott LF. Neural network dynamics. Annual review of neuroscience. 2005; 28:357–376. https://doi.org/10.1146/annurev.neuro.28.061604.135637 PMID: 16022600
30. Sompolinsky H, Crisanti A, Sommers HJ. Chaos in random neural networks. Physical review letters. 1988; 61(3):259. https://doi.org/10.1103/PhysRevLett.61.259 PMID: 10039285
31. Maheswaranathan N, Williams A, Golub M, Ganguli S, Sussillo D. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. Advances in neural information processing systems. vol. 32. Curran Associates, Inc.; 2019. p. 15696–15705.
32. Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: Advances in neural information processing systems; 2014. p. 2933–2941.
33. Boyd S, Vandenberghe L. Convex optimization. Cambridge university press; 2004.
34. Dayan P, Abbott LF. Theoretical neuroscience: computational and mathematical modeling of neural systems. Computational Neuroscience Series; 2001.
35. Kravitz DJ, Saleem KS, Baker CI, Ungerleider LG, Mishkin M. The ventral visual pathway: an expanded neural framework for the processing of object quality. Trends in cognitive sciences. 2013; 17(1):26–49. https://doi.org/10.1016/j.tics.2012.10.011 PMID: 23265839
36. Nayebi A, Sagastuy-Brena J, Bear DM, Kar K, Kubilius J, Ganguli S, et al. Goal-driven recurrent neural network models of the ventral visual stream. bioRxiv. 2021;.
37. Hooi-Tong L. On a class of directed graphs–with an application to traffic-flow problems. Operations research. 1970; 18(1):87–94. https://doi.org/10.1287/opre.18.1.87

38. Eaves BC, Hoffman AJ, Rothblum UG, Schneider H. Line-sum-symmetric scalings of square nonnegative matrices. In: Mathematical programming essays in honor of George B. Dantzig Part II. Springer; 1985. p. 124–141.
39. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media; 2009.
40. Schneider H, Schneider MH. Max-balancing weighted directed graphs and matrix scaling. Mathematics of operations research. 1991; 16(1):208–222. https://doi.org/10.1287/moor.16.1.208
41. Rothblum UG, Zenios SA. Scalings of matrices satisfying line-product constraints and generalizations. Linear algebra and its applications. 1992; 175:159–175. https://doi.org/10.1016/0024-3795(92)90307-V
42. Tanaka H, Kunin D, Yamins DL, Ganguli S. Pruning neural networks without any data by iteratively conserving synaptic flow. Advances in neural information processing systems. 2020;.
43. Du SS, Hu W, Lee JD. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In: Advances in neural information processing systems; 2018. p. 384–395.
44. Chung FR, Graham FC. Spectral graph theory. 92. American Mathematical Soc.; 1997.
45. Klein DJ, Randić M. Resistance distance. Journal of mathematical chemistry. 1993; 12(1):81–95. https://doi.org/10.1007/BF01164627
46. Dunwiddie T, Lynch G. Long-term potentiation and depression of synaptic responses in the rat hippocampus: localization and frequency dependency. The Journal of physiology. 1978; 276(1):353–367. https://doi.org/10.1113/jphysiol.1978.sp012239 PMID: 650459
47. Royer S, Paré D. Conservation of total synaptic weight through balanced synaptic depression and potentiation. Nature. 2003; 422(6931):518–522. https://doi.org/10.1038/nature01530 PMID: 12673250
48. Turrigiano GG, Leslie KR, Desai NS, Rutherford LC, Nelson SB. Activity-dependent scaling of quantal amplitude in neocortical neurons. Nature. 1998; 391(6670):892–896. https://doi.org/10.1038/36103 PMID: 9495341
49. Hengen KB, Lambo ME, Van Hooser SD, Katz DB, Turrigiano GG. Firing rate homeostasis in visual cortex of freely behaving rodents. Neuron. 2013; 80(2):335–342. https://doi.org/10.1016/j.neuron.2013.08.038 PMID: 24139038
50. Zenke F, Gerstner W, Ganguli S. The temporal paradox of Hebbian learning and homeostatic plasticity. Current opinion in neurobiology. 2017; 43:166–176. https://doi.org/10.1016/j.conb.2017.03.015 PMID: 28431369
51. Marder E, Goaillard JM. Variability, compensation and homeostasis in neuron and network function. Nature Reviews Neuroscience. 2006; 7(7):563–574. https://doi.org/10.1038/nrn1949 PMID: 16791145