ARTICLE Communicated by Aapo Hyvarinen

A Correspondence Between Normalization Strategies in Artificial and Biological Neural Networks

Yang Shen
yshen@cshl.com
Julia Wang
julwang@cshl.edu
Saket Navlakha
navlakha@cshl.edu
Cold Spring Harbor Laboratory, Simons Center for Quantitative Biology, Cold Spring Harbor, NY 11724, U.S.A.

A fundamental challenge at the interface of machine learning and neuroscience is to uncover computational principles that are shared between artificial and biological neural networks. In deep learning, normalization methods such as batch normalization, weight normalization, and their many variants help to stabilize hidden unit activity and accelerate network training, and these methods have been called one of the most important recent innovations for optimizing deep networks. In the brain, homeostatic plasticity represents a set of mechanisms that also stabilize and normalize network activity to lie within certain ranges, and these mechanisms are critical for maintaining normal brain function. In this article, we discuss parallels between artificial and biological normalization methods at four spatial scales: normalization of a single neuron's activity, normalization of synaptic weights of a neuron, normalization of a layer of neurons, and normalization of a network of neurons. We argue that both types of methods are functionally equivalent—that is, both push activation patterns of hidden units toward a homeostatic state, where all neurons are equally used—and we argue that such representations can improve coding capacity, discrimination, and regularization. As a proof of concept, we develop an algorithm, inspired by a neural normalization technique called *synaptic scaling*, and show that this algorithm performs competitively against existing normalization methods on several data sets. Overall, we hope this bidirectional connection will inspire neuroscientists and machine learners in three ways: to uncover new normalization algorithms based on established neurobiological principles; to help quantify the trade-offs of different homeostatic plasticity mechanisms used in the brain; and to offer insights about how stability may not hinder, but may actually promote, plasticity.

Neural Computation **33, 3179–3203** (2021)
https://doi.org/10.1162/neco_a_01439

1 Introduction

Since the dawn of machine learning, normalization methods have been used to preprocess input data to lie on a common scale. For example, min-max normalization, unit vector normalization, z-scoring, and the like are all well known for improving model fitting, especially when different input features have different ranges (e.g., age versus salary). In deep learning, normalizing the input layer has also proved beneficial; for example, "whitening" input features so that they are decorrelated and have zero mean and unit variance leads to faster training and convergence (LeCun, Bottou, Orr, and Müller, 1998; Desjardins, Simonyan, Pascanu, & Kavukcuoglu, 2015). More recently, normalization has been extended to hidden layers of deep networks, whose activity can be viewed as inputs to a subsequent layer. This type of normalization modifies the activity of hidden units to lie within a certain range or to have a certain distribution, independent of input statistics or network parameters (Ioffe & Szegedy, 2015). While the theoretical basis for why these methods improve performance has been subject to much debate—for example, reducing covariate shift (Ioffe & Szegedy, 2015), smoothing the objective landscape (Santurkar, Tsipras, Ilyas, & Madry, 2018), decoupling the length and direction of weight vectors (Kohler et al., 2019), and acting as a regularizer (Wu et al., 2019; Luo, Wang, Shao, & Peng, 2018; Poggio, Liao, & Banburski, 2020)—normalization is now a standard component of state-of-the-art architectures and has been called one of the most important recent innovations for optimizing deep networks (Kohler et al., 2019).

In the brain, normalization has long been regarded as a canonical computation (Carandini & Heeger, 2011; Weber, Krishnamurthy, & Fairhall, 2019) and occurs in many sensory areas, including in the auditory cortex to varying sound intensities (Rabinowitz, Willmore, Schnupp, & King, 2011) and the olfactory system to varying odor concentrations (Olsen, Bhandawat, & Wilson, 2010). Normalization is believed to help generate intensity-invariant representations for input stimuli, which improve discrimination and decoding that occurs downstream (Carandini & Heeger, 2011). Gain control is a related mechanism commonly used at the sensory layer to ensure neural activity remains within bounds (Schwartz & Simoncelli, 2001; Priebe & Ferster, 2002). For example, gain control is used to ensure that neural responses in the retina remain within a narrow range of amplitudes, despite significant changes in light level from day to night (Shapley, 1997; Rodieck, 1998; Mante, Frazor, Bonin, Geisler, & Carandini, 2005).

But beyond the sensory (input) level, there is an additional type of normalization found ubiquitously in the brain, called *homeostatic plasticity* (Turrigiano, 2012). Homeostasis refers to the general ability of a system to recover to some set point after being changed or perturbed (Cannon, 1932). A canonical example is a thermostat used to maintain an average temperature in a house. In the brain, the set point can take on different forms at

different spatial scales, such as a target firing rate for an individual neuron or a distribution of firing rates over a population of neurons. This set point is typically approached over a relatively long period of time (hours to days). The changes or perturbations occur due to other plasticity mechanisms, such as long-term potentiation (LTP) or long-term depression (LTD), which modify synaptic weights and firing rates at much faster timescales (seconds to minutes). Thus, the challenge of homeostasis is to ensure that set points are maintained on average without "erasing" the effects of learning. This gives rise to a basic stability-versus-plasticity dilemma. Disruption of homeostasis mechanisms has been implicated in numerous neurological disorders (Laughlin & Sejnowski, 2003; Turrigiano & Nelson, 2004; Houweling, Bazhenov, Timofeev, Steriade, & Sejnowski, 2005; Yu, Sternad, Corcos, & Vaillancourt, 2007; Bakker et al., 2012; Wondolowski & Dickman, 2013), indicating their importance for normal brain function.

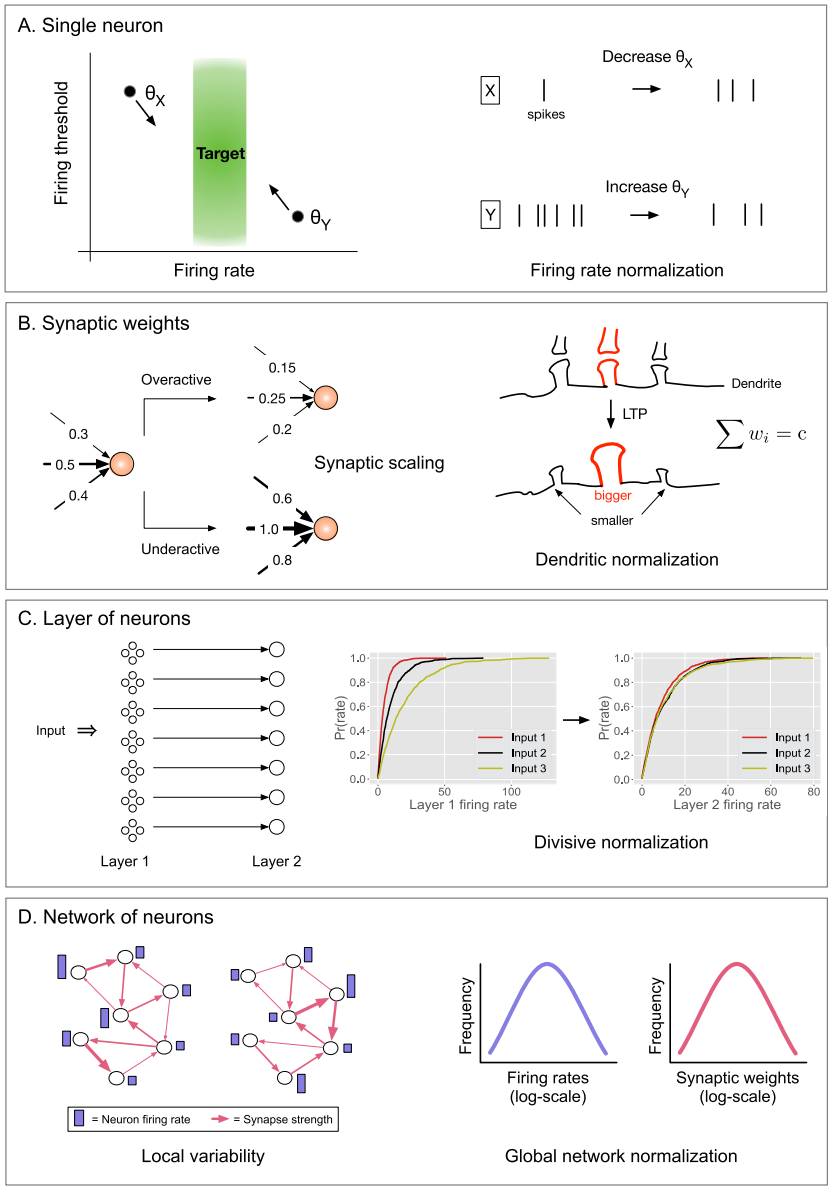
In this article, we highlight parallels between normalization algorithms used in deep learning and homeostatic plasticity mechanisms used in the brain. Identifying these parallels can serve two purposes. First, machine learners have extensive experience analyzing normalization methods and have developed a sense of how they work, why they work, and when using certain methods may be preferred over others. This experience can translate to quantitative insights about outstanding challenges in neuroscience, including the stability-versus-plasticity trade-off, the roles of different homeostasis mechanisms used across space and time, and whether there are parameters critical for maintaining homeostatic function that have been missed experimentally. Second, there are many normalization techniques used in the brain that have not, to our knowledge, been deeply explored in machine learning. This represents an opportunity for neuroscientists to propose new normalization algorithms from observed phenomena or established principles (Hassabis, Kumaran, Summerfield, & Botvinick, 2017) or to provide new perspectives on why existing normalization schemes used in deep networks work so well in practice.

2 Normalization Methods across Four Spatial Scales

We begin by describing artificial and neural normalization strategies that occur across four spatial scales (see Figure 1 and Table 1): normalization of a single neuron's activity via intrinsic neural properties, normalization of synaptic weights of a neuron, normalization of a layer of neurons, and normalization of an entire network of neurons.

2.1 Normalization of a Single Neuron's Activity. Here, we focus on normalization methods that directly modify the activity level of a neuron via intrinsic mechanisms.

In deep learning, the current most popular form of single neuron normalization is *batch normalization* (Ioffe & Szegedy, 2015). It has long been


A. Single neuron
Firing threshold
θX
Target
θY
Firing rate
Decrease θX
X
spikes
Increase θY
Y
Firing rate normalization
B. Synaptic weights
Overactive
0.15
0.25
0.2
0.3
0.5
0.4
Synaptic scaling
0.6
1.0
0.8
Underactive
Dendrite
LTP
bigger
smaller
Dendritic normalization
C. Layer of neurons
Input
Layer 1
Layer 2
Pr(rate)
Input 1
Input 2
Input 3
Layer 1 firing rate
Layer 2 firing rate
Divisive normalization
D. Network of neurons
= Neuron firing rate
= Synapse strength
Local variability
Frequency
Firing rates
(log-scale)
Synaptic weights
(log-scale)
Global network normalization

known that z-scoring the input layer—that is, shifting and scaling the inputs to have zero mean and unit variance—speeds up network training (LeCun, Bottou, Orr, & Müller, 1998). Batch normalizaton essentially applies this idea to each hidden layer by ensuring that for every batch of training examples, the activation of a hidden unit over the batch has zero mean and unit variance.

Mathematically, let $\{z_1, z_2, \ldots, z_B\}$ be the activations of hidden unit z for each of the $i = 1 : B$ inputs in a training batch. Let μ_B and σ_B^2 be the mean and variance of all the z_i's, respectively. Then the batch-normalized activation of z for the ith input is

$$\hat{z}_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}},$$

where ϵ is a small constant.

In practice, the effect of this simple transformation is profound: it leads to significantly faster convergence (larger learning rates) and improved stability (less sensitivity to parameter initialization and learning rate) (Ioffe and Szegedy, 2015; Bjorck, Gomes, Selman, & Weinberger, 2018; Santurkar et al., 2018; Arora, Li, & Lyu, 2019). Numerous extensions of this method have since been proposed with various tweaks and perks on a similar underlying idea (see Table 1).

Figure 1: Neural homeostatic plasticity mechanisms across four spatial scales. (A) Normalization of a single neuron's activity. Left: Neuron X has a relatively low firing rate and a high firing threshold, θ_X, and vice versa for neuron Y. Right: Both neurons can be brought closer to their target firing rate by decreasing θ_X and increasing θ_Y. (B) Normalization of synaptic weights. Left (synaptic scaling): If a neuron is firing above its target rate, its synapses are multiplicatively decreased, and vice versa if the neuron is firing below its target rate. Right (dendritic normalization): If a synapse size increases due to strong LTP, its neighboring synapses decrease their size. (C) Normalization of a layer of neurons. Left: Two layers of neurons with feedforward connections and other feedback inhibitory connections (not shown). Right: The cumulative distribution of firing rates for neurons in the first layer is exponential with a different mean for different inputs. The activity of neurons in the second layer is normalized such that the means of the three exponentials are approximately the same. (D) Left: Example of a neural circuit with the same units and connections but different activity levels for neurons (purple bars) and different weights (pink arrow thickness) under two different conditions. Right: Despite local variability, the global distributions of firing rates and synaptic weights for the network remains stable (log-normally distributed) under both conditions.

Table 1: Correspondences Between Normalization Mechanisms in Artificial and Biological Neural Networks across Four Spatial Scales.

Scale	Deep learning	References	Neural circuits	References
Single neuron	Batch normalization	Ioffe and Szegedy (2015)	Firing thresholds	Bienenstock et al. (1982); Zhang and Linden (2003); Turrigiano (2011)
			Ion channel density	Marder and Goaillard (2006); Davis (2013)
			Cell-type specificity	Turrigiano (2012)
	Group normalization	Wu and He (2018)	—	—
	Instance normalization	Ulyanov, Vedaldi, and Lempitsky (2016)	—	—
	Self-normalization	Klambauer, Unterthiner, Mayr, and Hochreiter (2017)	—	—
	Normalization propagation	Arpit, Zhou, Kota, and Govindaraju (2016)	—	—
Synaptic weights	Weight normalization	Salimans and Kingma (2016)	(Post) synaptic scaling	Turrigiano (2008); Turrigiano (2012)
	—	—	(Pre) release probability	Murthy et al. (2001); Turrigiano (2012)
	—	—	(Branch) dendritic normalization	Royer and Pare (2003); Rabinowitch and Segev (2008); Chistiakova et al. (2015)
Layer of neurons	Whitening	LeCun, Bottou, Orr, and Müller (1998); Raiko, Valpola, and Lecun (2012); Desjardins et al. (2015)	Decorrelation / whitening	Pitkow and Meister (2012); Graham, Chandler, and Field (2006); Pozzorini, Naud, Mensi, and Gerstner (2013); Huang, Zhang, and Chacron (2016); Wanner and Friedrich (2020)
	Layer normalization	Lei Ba et al. (2016)	Divisive normalization	Carandini and Heeger (2011)
Network of neurons	—	—	Network homeostasis	Maffei and Fontanini (2009); Slomowitz et al. (2015)

See also reviews: Turrigiano (2017); Keck et al. (2017); Fox and Stryker (2017); Davis (2006).

In the brain, normalizing the activity of a neuron has long been appreciated as an important stabilizing mechanism (Pozo & Goda, 2010). For example, if neuron u drives neuron v to fire, the synapse between them may get strengthened by Hebbian plasticity. Then the next time u fires, it is even more likely that v fires, and this positive feedback loop can lead to excessive activity. Similarly, if the synapse undergoes depression, it is less likely for v to fire in the future, and this negative feedback can lead to insufficient activity. The job of homeostasis is to prevent neurons from being both overutilized (hyperactive) and underutilized (hypoactive) (Turrigiano, 2008).

Modifying a neuron's excitability (e.g., its firing threshold or bias) represents one intrinsic neural mechanism used to achieve homeostasis (Bienenstock, Cooper, & Munro, 1982; Zhang and Linden, 2003; Turrigiano, 2011). The idea is simple (see Figure 1A); each neuron has an approximate target firing rate at which it prefers to fire. A neuron with sustained activity above its target rate will increase its firing threshold such that it becomes harder to fire, and likewise, a neuron with depressed activity below its target will decrease its firing threshold, thus becoming more sensitive to future inputs. The net effect of these modifications is that the neuron hovers around its target firing rate, on average, over time. Several parameters are involved in this process, such as the rate at which thresholds are adjusted (which affects how quickly homeostasis is approached) and the value of the target itself, which may be cell-type-specific. Other intrinsic mechanisms, such as modifying ion channel density, can also be used to intrinsically regulate firing rates (see Figure 1).

Both of these methods are unsupervised; they adjust the activity of a neuron to lie within a preferred, narrow range with respect to recently observed data.

2.2 Normalization of Synaptic Weights. Here, we focus on normalization methods that indirectly modify the activity of a neuron by changing its weights.

In deep learning, one popular way to (postsynaptically) normalize the inputs to a hidden unit is *weight normalization* (Salimans & Kingma, 2016). The idea is to reparameterize the conventional weight vector $\mathbf{w}$ of a unit into two components,

$$\mathbf{w} = \frac{c}{\|\mathbf{v}\|}\mathbf{v},$$

where c is a scalar and $\mathbf{v}$ is a parameter vector, both of which are learned. This transformation fixes the length (Euclidean norm) of the weight vector, such that $\|\mathbf{w}\| = c$, for any $\mathbf{v}$. Backpropagation is then applied to c and $\mathbf{v}$ instead of to $\mathbf{w}$. Thus, the length of the weight vector (c) is decoupled from the direction of the weight vector ($\mathbf{v}/\|\mathbf{v}\|$). Such "length-direction"

decoupling leads to faster learning and exponential convergence in some cases (Kohler et al., 2019). The concept of weight normalization has also been proposed as a means to enforce optimization constraints in neural map formation (Wiskott & Sejnowski, 1998).

In the brain, the best-studied type of weight normalization is called *synaptic scaling* (Turrigiano, 2008; see Figure 1B, left). If a neuron is on average firing above its target firing rate, then all of its incoming excitatory synapses are downscaled (i.e., multiplied by some factor, $0 < \alpha < 1$) to reduce its future activity. Similarly, if a neuron is firing far below its target, then all its excitatory synapses are upscaled ($\alpha > 1$); in other words, prolonged inactivity leads to an increase in synaptic size (Murthy, Schikorski, Stevens, & Zhu, 2001). These rules may seem counterintuitive, but remember that these changes are happening over longer timescales than the changes caused by standard plasticity mechanisms. Indeed, it is hypothesized that one way to resolve the plasticity-versus-stability dilemma is to temporally segregate Hebbian and homeostatic plasticity so that they do not interfere (Turrigiano, 2017). This could be done, for example, by activating synaptic scaling during sleep (Tononi & Cirelli, 2012; Krishnan, Tadros, Ramyaa, & Bazhenov, 2019).

Interestingly, synapse sizes are scaled on a per neuron basis using a multiplicative update rule (see Figure 1B, left). For example, if a neuron has four incoming synapses with weights 1.0, 0.8, 0.6, and 0.2 and if the neuron is firing above its target rate, then the new weights would be downscaled to 0.5, 0.4, 0.3, and 0.1, assuming a multiplicative factor of $\alpha = 1/2$. Critically, multiplicative updates ensure that the relative strengths of input synapses are preserved, which is believed to help maintain specificity of the neuron's response caused by learning. The value of the multiplicative factor need not be constant and could depend, for example, on how far away the neuron is from reaching its target rate. Thus, synaptic scaling keeps the firing rate of a neuron within a range while preserving the relative strength between synapses.

Another form of weight normalization in the brain is called *dendritic normalization* (Royer & Pare, 2003; Rabinowitch & Segev, 2008; Chistiakova, Bannon, Chen, Bazhenov, & Volgushev, 2015); it occurs locally on individual branches of a neuron's dendritic arbor (see Figure 1B, right). The idea is that if one synapse gets strengthened, then its neighboring synapses on the arbor compensate by weakening. This process is homeostatic because the total strength of all synapses along a local part of the arbor remains approximately constant. This process could be mediated by a shared resource, for example, a fixed number of postsynaptic neurotransmitter receptors available among neighboring synapses (Li, Li, Lei, Wang, & Guo, 2013; Triesch, Vo, & Hafner, 2018). Computationally, this process creates sharper boundaries between spatially adjacent synapses receiving similar inputs, which could enhance discrimination and contrast.

2.3 Normalization of a Layer of Neurons. Here, we focus on normalization schemes that modify the activity of an entire layer of neurons, as opposed to just a single neuron's activity.

In deep learning, *layer normalization* (Lei Ba, Kiros, & Hinton, 2016) was proposed to overcome several drawbacks of batch normalization. In batch normalization, the mean and variance statistics of each neuron's activity is computed across a batch of training examples, and then each neuron is normalized with respect to its own statistics over the batch. In layer normalization, the mean and variance are instead computed over an entire layer of neurons for each training example, and then each neuron in the layer is normalized by the same mean and variance. Thus, layer normalization can be used online (i.e., batch size of one), which makes it more amenable to training recurrent neural networks (Lei Ba et al., 2016).

In the brain, layer-wise normalization has most prominently been observed in sensory systems (see Figure 1C, left). For example, in the fruit fly olfactory system, the first layer of (receptor) neurons encode odors via a combinatorial code, in which, for any individual odor, most neurons respond at a low rate and very few neurons respond at a high rate (Stevens, 2015). Specifically, the distribution of firing rates over all receptor neurons is exponential with a mean that depends on the concentration of the odor (higher concentration $\rightarrow$ higher mean). In the second layer of the circuit, projection neurons receive odor excitation from receptor neurons, as well as inhibition from lateral inhibitory neurons (Olsen et al., 2010). The result is that the concentration dependence is largely removed; that is, the distribution of firing rates for projection neurons follows an exponential distribution with approximately the same mean, for all odors and all odor concentrations (Stevens, 2016; see Figure 1C, right). Thus, while an individual neuron's firing rate can change depending on the odor, the distribution of firing rates over all neurons remains nearly the same for any odor. This process is dubbed *divisive normalization* and is believed to help fruit flies identify odors independent of the odor's concentration. Divisive normalization has also been studied in the visual system, for example, for light adaptation in the retina or contrast adjustment in the visual cortex (Carandini & Heeger, 2011; Sanchez-Giraldo, Laskar, & Schwartz, 2019).

Overall, layer normalization divides the responses of individual neurons by a factor that relates to the summed activity of all the neurons in the layer. These normalizations can be considered "homeostatic" because they preserve, for any input, certain properties of a distribution of firing rates, such as the mean or variance. In the brain, other nonlinear transformations are also used alongside these transformations, for example, to adjust saturation rates of individual neurons and amplify signals prior to normalization (Carandini & Heeger, 2011).

2.4 Normalization of a Network of Neurons. In the brain, recent work has challenged the conventional view that homeostasis applies only at the

level of a single neuron or a strict layer of neurons and has instead attributed homeostasis properties to a broader network of neurons. In one experiment, the firing rates of individual neurons in a hippocampal network were monitored for two days after applying baclofen, a chemical agent that suppresses neural activity. After two days, the distribution of firing rates over the population was compared to the distribution of firing rates for a control group of neurons that received no baclofen; strikingly, both were approximated by the same log-normal distribution. Moreover, the firing rates of many individual neurons, in both conditions, significantly changed from day 0 to day 2 (Slomowitz et al., 2015; Ziv et al., 2013) (Figure 1D). Thus, individual neurons may deviate from their "preferred" firing rate, but the distribution, and hence the sum, of firing rates over the population is well preserved. Similar observations have been made in the stomatogastric ganglion of crabs and lobsters, where rhythmic bursting is robustly maintained despite many different configurations of the circuit (Prinz, Bucher, & Marder, 2004). This remains a beautiful yet mysterious property of network stability implemented by neural circuits, and the mechanisms driving this level of network regulation remain poorly understood (Buzsaki & Mizuseki, 2014).

In deep learning, we are not aware of a normalization strategy that is applied across an entire network of units or even across a population of units beyond a single layer. Network homeostasis could in principle be an emergent property from local homeostasis rules implemented by individual units or could be a global constraint intrinsically enforced by some unknown mechanism. Either way, we hypothesize that network homeostasis may be attractive in deep networks because it allows for more flexible local representations while still providing stability at the network level.

3 The Computational Benefits of Homeostasis (Load Balancing)

In computer science, the term *load balancing* means to distribute a data processing load evenly over a set of computing units, such that efficiency is maximized and the amount of time that units are idle is minimized (Lynch, 1996; e.g., load balancing of servers that handle traffic from users on the Internet). For neural networks, we define load balancing based on how frequently a set of neurons is activated and how similar their mean activation levels are on average. Why might load balancing in neural networks be attractive computationally? Three reasons come to mind.

First, load balancing increases the coding capacity of the network—that is, the number of unique stimuli that can be represented using a fixed number of resources (neurons). Suppose that under standard training, a certain fraction (say, 50%) of the hidden units are not used; that is, they are never, or rarely, activated. This wasted capacity would reduce the number of possible patterns the network could represent and would introduce unnecessary parameters that can prolong training. Load balancing of neurons could avoid these problems by pressing more hidden units into service. In the brain, equal utilization of neurons also promotes distributed representations, in

which each stimulus is represented by many neurons, and each neuron participates in the representation of many stimuli (often called a combinatorial code; Stevens, 2015; Malnic, Hirono, Sato, & Buck, 1999). This property is particularly attractive when such representations are formed independent of input statistics or structure. Importantly, load balancing is not in conflict with sparse coding, another common coding scheme in the brain (Olshausen & Field, 2004). In generating a sparse code, only a small percentage of neurons are activated per input. However, after averaging over many inputs over time, each neuron is activated roughly the same number of times, and this is a form of load balancing.

Second, load balancing can improve fine-grained discrimination. If a neuron has a sigmoidal activation function, normalization keeps the neuron in its nonsaturated regime. This means the neuron will be sensitive to small changes in the input, which is believed to help the neuron be maximally informative and discriminative (Wang, Stocker, & Lee, 2016; Ganguli & Simoncelli, 2010, 2014; Wang, Stocker, & Lee, 2012; Laughlin, 1981). Moreover, if input statistics shift over time, the neuron will continue to respond within a narrow range of amplitudes, and thus continue to have a useful dynamic range.

Third, load balancing can serve as a regularizer, which is commonly used in deep networks to constrain the magnitude of weights or the activity levels of units. Regularizers typically improve generalization and reduce overfitting (Kukacka, Golkov, & Cremers, 2017) and can be specified explicitly or implicitly (Neyshabur, Tomioka, Salakhutdinov, & Srebro, 2017). There are many forms of regularization used in deep learning—for example, Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), in which a random fraction of the neurons is set inactive during training, or weight regularization, in which ℓ_1 or ℓ_2 penalties are applied to the loss function to limit how large weight vectors become (Lang & Hinton, 1990; Zou & Hastie, 2005). Although regularization is a powerful tool to build robust models, regularization alone is not guaranteed to generate load-balanced (homeostatic) representations.

4 Empirically Testing the Benefits of Homeostasis

The empirical results in this section serve two purposes. The first is to show that two popular normalization methods (batch normalization and weight normalization) generate homeostatic representations and demonstrate the three benefits of homeostasis discussed in the previous section. The second is to show that a method inspired by synaptic scaling also demonstrates these benefits and performs competitively against existing normalization methods.

These results are not meant to represent a full-fledged comparison between normalization methods across multiple architectures, data sets, or hyperparameter settings. Rather, these results are simply meant to demonstrate a proof-of-concept of the bidirectional perspective argued here.

Table 2: Normalization Algorithms.

Algorithm	Equations	Notation
Batch normalization	$z_i = \boldsymbol{w}\boldsymbol{x_i} + b$ $\hat{z}_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ $y_i = \text{ReLU}(\gamma \hat{z}_i + \beta)$	i: ith example in a batch of size B z_i: value of the unit (before activation) for i y_i: value of the unit (after activation) for i
Weight normalization	$y_i = \text{ReLU}(\boldsymbol{w}\boldsymbol{x}_i + b)$ $\boldsymbol{w} = \frac{c}{\|\|\boldsymbol{v}\|\|}\boldsymbol{v}$	$\boldsymbol{w}$: incoming weights to the unit $\boldsymbol{x_i}$: inputs to the unit for i
Synaptic scaling	$\tilde{\boldsymbol{w}} = \alpha \boldsymbol{w}$ $z_i = \tilde{\boldsymbol{w}}\boldsymbol{x}_i + b$ $y_i = \text{ReLU}(z_i - \mu_B)$	b: bias of the unit μ_B: average of z_i's over the batch σ_B^2: variance of z_i's over the batch
Mean-only	$z_i = \boldsymbol{w}\boldsymbol{x}_i + b$ $y_i = \text{ReLU}(z_i - \mu_B)$	γ, β: trainable parameters (BatchNorm) c, $\boldsymbol{v}$: training parameters (WeightNorm)
Scale-only	$\tilde{\boldsymbol{w}} = \alpha \boldsymbol{w}$ $z_i = \tilde{\boldsymbol{w}}\boldsymbol{x}_i + b$ $y_i = \text{ReLU}(z_i)$	α: trainable parameter (Synaptic Scaling) ϵ: a small constant

Notes: All equations show the forward-pass update equations for a single hidden unit. For Weight normalization, backpropagation is performed on c and $\boldsymbol{v}$, instead of $\boldsymbol{w}$.

4.1 Experimental Setup. For our basic architecture, we used the original LeNet5 (LeCun, Bottou, Bengio, & Haffner, 1998) with two convolutional layers and three fully connected layers with ReLU activation functions.

We experimented with two data sets. The first is CIFAR-10, a standard benchmark for classification tasks, which contains 60,000 color images, each of size 32×32, and each belonging to one of 10 classes (e.g., airplanes, cats, trucks). The second data set is SVHN (Street House View Numbers), which contains 73,257 color images, each of size 32×32, and each belonging to one of 10 classes (digits 0–9). SVHN is analogous to MNIST but is more difficult to classify because it includes house numbers in natural scene images taken from a street view.

Each normalization method is applied to every layer, except the input and output layers, with all affine parameters trainable. All methods used Adam optimization in PyTorch with default parameters. Additional hyperparameters were fixed for each data set: CIFAR-10 (batch size of 32, learning rate of 0.003, train for about 45,000 iterations) and SVHN (batch size of 256, learning rate of 0.01, train for about 8,000 iterations). Batch statistics are calculated using training data during training and using testing data during testing. Table 2 provides the equations for each normalization algorithm.

There are three benefits of homeostasis, measured as follows:

1. *Coding capacity*: The information entropy of the binarized activation values over hidden units. Entropy is highest when the probability

that a random unit is activated (i.e., it outputs a value greater than zero) for an input is 50%.
2. *Discrimination*: The classification accuracy on the test set.
3. *Regularization*: The range of the response magnitudes of hidden units; a narrower range implies better regularization.

4.2 A Synaptic-Scaling-Inspired Normalization Algorithm. Of the many normalization strategies discussed above, we choose to model synaptic scaling because it is one of the best studied and most widely observed mechanisms across brain regions and species.

We propose a simplified model of synaptic scaling that captures two key aspects of the underlying biology: multiplicative scaling of synaptic weights and constraining a node to be activated around a target activation probability on average. In the first step, the incoming weight vector $\boldsymbol{w}$ for a hidden unit is multiplied by a factor α, that is, $\boldsymbol{w} = \alpha \boldsymbol{w}$. Each hidden unit has its own α value, which is made learnable during training. The α values are initialized to 1. In the second step, for each hidden unit, we subtract its mean activation (over a batch) from its actual activation for each input in the batch. This process ensures that each unit has a mean activation (before ReLU) of 0 and, hence, a probability of activation (output value > 0) of around 50%, and thus resembles the biological observation that no neuron is over- or under utilized. This step is also the same as mean-only batch normalization (Salimans & Kingma, 2016). One advantage of this synaptic scaling model compared to batch normalization is that it removes the division by the variance term, which can lead to exploding gradients when the variance is close to zero.

An "ideal" model of synaptic scaling might only multiplicatively scale the weights of a hidden unit such that a given target activation probability is achieved on average. Instead, we first scale the weights by a learnable parameter (α), which allows the network to learn the optimal range of activation values for the unit, and we then constrain the unit to hit its target. Similarly, batch normalization does not simply use z-scored activation values for each hidden unit (see Table 2), but rather includes two learnable parameters (γ, β) per unit to shift and scale its normalized activation. In both cases, this flexibility likely increases the representation power of the network (Ioffe & Szegedy, 2015).

Mathematically, for each hidden unit, the forward-pass operations for synaptic scaling are

$$\tilde{\boldsymbol{w}} = \alpha \boldsymbol{w},$$

$$z_i = \tilde{\boldsymbol{w}} \boldsymbol{x}_i + b,$$

$$y_i = \mathrm{ReLU}(z_i - \mu_B),$$

where the subscript i indicates the ith example in a batch of size B ($i = 1 : B$); $\boldsymbol{w}$, $\boldsymbol{x_i}$, b, z_i, y_i are the incoming weights to the hidden unit, the inputs for the

ith example from the previous layer, the bias of the unit, the value of the unit before activation, and the output of the unit, respectively; μ_B is the average of all z_i's over a batch.

To explore how the two steps independently affect classification performance, we tested each of them without the other as two additional control experiments. We call these models "mean-only" and "scale-only," respectively (see Table 2).

4.3 Existing Normalization Methods Generate Homeostatic Representations. First, we confirmed that two state-of-the-art normalization methods—batch normalization (BatchNorm) and weight normalization (WeightNorm)—improve discrimination (i.e., classification accuracy) on CIFAR-10: from $59.3 \pm 1.4\%$ for the original version of LeNet5 without normalization (Vanilla) to $63.8 \pm 0.9\%$ (WeightNorm) and $65.8 \pm 0.5\%$ (BatchNorm) (see Figure 2A). Normalized networks also learned faster; they required fewer training iterations to achieve high accuracy.

Second, we show that BatchNorm and WeightNorm have higher coding capacity; units are relatively equally utilized, each with an activation probability close to 0.50. Figure 2B shows that hidden units in normalized networks had more similar activation probabilities than in Vanilla: the coefficients of variation of activation probabilities across hidden units were 0.20 (BatchNorm) and 1.38 (WeightNorm) compared to 1.65 (Vanilla). Further, individual units in networks with BatchNorm and WeightNorm had activation probabilities closer to 0.50 compared to Vanilla. For example, in the first fully connected layer, units in BatchNorm and WeightNorm had activation probabilities of 0.41 ± 0.05 and 0.17 ± 0.23, respectively, compared to Vanilla (0.10 ± 0.15) (see Figure 2C). For BatchNorm, the probability of activation forms a near gaussian distribution, whereas for Vanilla, the distribution forms a huge peak at 0.0 with a long right tail, indicating that many units in Vanilla never get activated, while a few units are "overused." The distribution of WeightNorm lies between Vanilla and BatchNorm.

Third, for regularization, Figure 2D shows that when active, the values of hidden units have a narrower distribution when using normalization compared to without normalization; the coefficients of variation of activation values across hidden units were 0.16 (BatchNorm) and 0.30 (WeightNorm) compared to 0.55 (Vanilla). The average output value for hidden units was also significantly reduced in BatchNorm (0.89 ± 0.14) and WeightNorm (0.76 ± 0.23) compared to Vanilla (13.36 ± 7.36). By reducing the activation values of hidden units and confining them to a narrower range, BatchNorm and WeightNorm demonstrate regularization.

4.4 Synaptic Scaling Performs Load Balancing and Obtains Competitive Performance. We next tested the Synaptic Scaling method and found that its classification accuracy ($66.0 \pm 0.7\%$) was very similar to BatchNorm ($65.8 \pm 0.5\%$) on CIFAR-10 (see Figure 2A). In contrast, mean-only

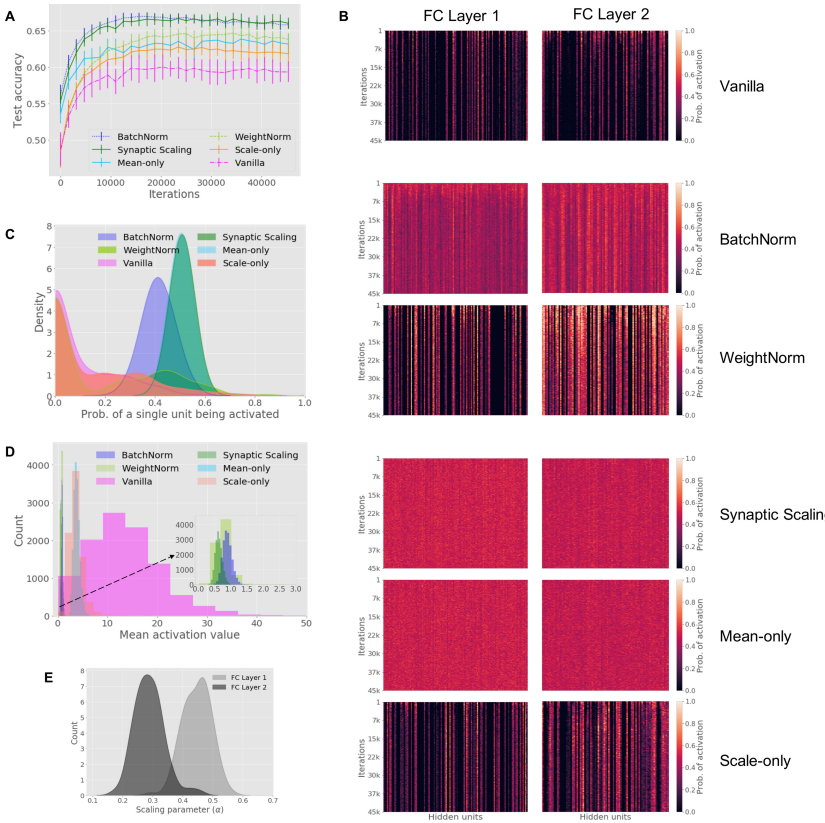


Figure 2: Data set: CIFAR-10: Normalization increases performance and drives neural networks toward a "homeostatic" state. (A) Test accuracy (y-axis) versus training iteration (x-axis). Error bars show standard deviation over 10 random initializations. BatchNorm and Synaptic Scaling achieve higher accuracy at the beginning and the end of training compared to all other methods, including Vanilla. (B) The probability of each hidden unit (columns) being activated over all inputs in a batch, computed on every 100th training iteration (rows). Heat maps are shown for hidden units in both fully connected (FC) layers. (C) Distribution of the probabilities that each unit in the first FC layer is activated per input. (D) Histogram of the mean activation values for hidden units in the first FC layer, calculated using the test data set. (E) Distribution of the trained α parameters for Synaptic Scaling, for each FC layer.

and scale-only performed worse than Synaptic Scaling, suggesting that both steps—multiplicative scaling of synapses and setting target activation probabilities—are better when combined than by themselves.

The coding capacity of Synaptic Scaling is on par with or even slightly better than BatchNorm. Figure 2B shows that each hidden unit had a similar probability of being activated—a coefficient of variation of 0.11 for Synaptic Scaling and 0.20 for BatchNorm, compared to 1.65 for Vanilla. Synaptic Scaling activated hidden units with a probability of 0.51 ± 0.02, slightly higher than BatchNorm (0.41 ± 0.05) and much higher than Vanilla (0.10 ± 0.15) (see Figure 2C).

Finally, for regularization, Figure 2D shows that the activation values across hidden units were similar after normalization with a coefficient of variation of 0.17 for Synaptic Scaling and 0.16 for BatchNorm, compared to 0.55 for Vanilla. The average output value for hidden units was also reduced in Synaptic Scaling and BatchNorm (0.63 ± 0.11 versus 0.89 ± 0.14, respectively) compared to Vanilla (13.36 ± 7.36).

Interestingly, the learned α parameters for Synaptic Scaling are all positive, meaning no weights flipped sign during training, and all the $\alpha < 1$, meaning the weights are all scaled down (see Figure 2E). We did not set any upper or lower bounds on α, and the fact that the learned values stay within $[0, 1]$ indicates that downscaling of weights, which in turn reduce activation values, may generally be beneficial for this classification task.

4.5 Validation on a Second Data Set. To ensure these results were not specific to one data set, we ran all methods on a second data set (SVHN) and found similar trends (see Figure 3). To summarize, Synaptic Scaling and BatchNorm improve classification accuracy (see Figure 3A), coding capacity (see Figures 3B and 3C), and regularization (see Figure 3D), compared to all other methods.

5 Discussion

We showed that widely used normalization methods in deep learning are functionally equivalent to homeostatic plasticity mechanisms in the brain. While the implementation details vary, both ensure that the activity of a neuron is centered around some fixed value or lies within some fixed distribution, and both are temporally local in the sense that changes only depend on recent behavior (recent firing rate or recent data observed). In summary, both attempt to stabilize and bound neural activity in an unsupervised manner, and both are critical for efficient learning.

We showed that two state-of-the-art normalization methods (BatchNorm and WeightNorm), as well as a new normalization algorithm inspired by synaptic scaling, demonstrate the three benefits of load balancing: compared to Vanilla, all three methods (1) increase coding capacity (i.e., per input, each unit has a probability closer to 50% of being activated), (2) increase discrimination (classification accuracy), and (3) act as a regularizer (narrowing the range of activation levels for each unit). Interestingly, WeightNorm achieves lower accuracy and generates representations that are less

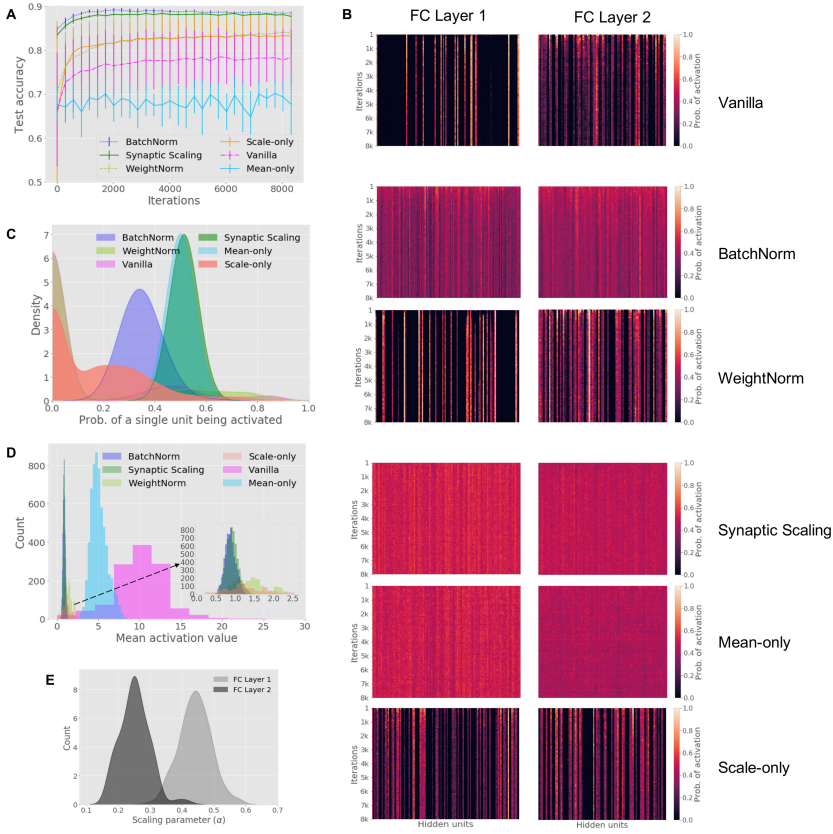


Figure 3: Data set: SVHN: Similar benefits of normalization on a second data set. Synaptic Scaling and BatchNorm have the highest classification accuracy (A), increase coding capacity (B,C: probability of each hidden unit being activated), and increase regularization (D: mean activation values for hidden units). See Figure 2 caption for detailed panel descriptions.

homeostatic, compared to both BatchNorm and Synaptic Scaling (see Figures 2 and 3). This suggests that learning algorithms are more efficient when coupled with homeostatic load balancing, and either without the other degrades performance. This article contributes to the growing list of explanations for why normalization is so useful in deep networks (Ioffe & Szegedy, 2015; Santurkar et al., 2018; Kohler et al., 2019; Wu et al., 2019; Poggio et al., 2020). A natural next step is to develop a theoretical understanding for why stability (i.e., creating homeostatic representations) may actually promote plasticity (i.e., improving classification accuracy and learning efficiency), as opposed to being in conflict.

While Synaptic Scaling performed similar to BatchNorm on the two data sets examined here, there are several differences between these two algorithms, both computationally and biologically. First, BatchNorm is less sensitive to the learning rate compared to Synaptic Scaling. For example, on the CIFAR-10 data set, BatchNorm achieves reasonable accuracy ($\sim$58%) with learning rates as large as 0.10, whereas the accuracy of Synaptic Scaling drops significantly for learning rates greater than 0.01. Specifically, BatchNorm achieves an accuracy of $67 \pm 1\%$ and $67 \pm 0.4\%$ for learning rates of 0.01 and 0.02, respectively, whereas Synaptic Scaling achieves an accuracy of $64 \pm 2\%$ for learning rate 0.01 but drops to $56 \pm 3\%$ for learning rate 0.02. Thus, BatchNorm is generally more robust than Synaptic Scaling. Second, BatchNorm divides the mean-centered activation by the standard deviation over the batch, which could lead to exploding activations when the standard deviation is close to zero. Synaptic Scaling scales the activations indirectly by multiplying the weights by a learnable parameter, which does not require any division or calculation of the standard deviation. Third, Synaptic Scaling achieves higher coding capacity than BatchNorm; for example, for Synaptic Scaling, the node activation probability (0.51 ± 0.02) is closer to 0.5 than BatchNorm (0.41 ± 0.05) (see Figure 2C), and the coefficient of variation of node activation probabilities is lower for Synaptic Scaling (0.11) than BatchNorm (0.20) (see Figure 2B). This suggests that the amount of information (entropy) present in the representation for Synaptic Scaling is slightly higher than that of BatchNorm. While this did not translate to better performance on the simple data sets examined here, more work is needed to test these algorithms in other scenarios (e.g., fine discrimination between very similar stimuli or testing on data sets with many more classes). We also emphasize that even though achieving good classification performance often implies that representations are homeostatic, achieving homeostasis alone does not guarantee good performance, and more theoretical work is needed to untangle these dependencies. Fourth, after normalization, BatchNorm requires two additional learnable parameters for each node (γ and β), whereas Synaptic Scaling requires one new parameter per node (α; see Table 2). Hence, Synaptic Scaling requires half of the parameters as BatchNorm, and more work is needed to quantify the trade-offs between fewer parameters, training efficiency, and representation power. Fifth, we are not aware of a neural mechanism capable of implementing BatchNorm (particularly the division by the standard deviation), whereas synaptic scaling is a well-established neural mechanism whose study may provide better feedback to neuroscience.

Moving forward, there are several challenges that remain in bridging the gap between understanding normalization in artificial and biological neural networks. First, the implementation details of both types of networks are well acknowledged to be different (Lillicrap, Santoro, Marris, Akerman, & Hinton, 2020). For example, unlike most artificial networks, the brain has a strict division of excitatory and inhibitory neurons, which means

different homeostasis rules can be applied to excitatory and inhibitory synapses (Joseph & Turrigiano, 2017). Second, our model of synaptic scaling assumed that each hidden unit had the same target fixed point, whereas in reality, adjustable fixed points might further improve performance. Indeed, batch normalization allows the fixed points to be learned through the affine parameter, β. In artificial networks, fixed points could vary based on the data set, network architecture, or other hyperparameters. In the brain, different cell types may use different fixed points, or fixed points of a single cell may change during different phases of training. Third, it is unclear how the timescales of homeostasis in the brain map to timescales of learning in artificial networks. Normalization is typically applied per input or per batch in deep learning, but other timescales remain unexplored (Tononi & Cirelli, 2012; Krishnan et al., 2019; Zenke & Gerstner, 2017). Similarly, normalization that operates simultaneously across different spatial scales (e.g., combining batch normalization and layer normalization) has only recently been explored (Ren, Liao, Urtasun, Sinz, & Zemel, 2017). Fourth, there are different constraints between what a hidden unit can store and compute and what a neuron can (likely) store and compute. For example, it seems plausible for a neuron to track its own mean firing rate over a given time window, but tracking its own variance seems trickier.

There are also several challenges in understanding the neuroscience of homeostasis that remain outstanding. For example, network-wide homeostasis, which goes beyond fixed points for individual neurons, has been observed in the brain, but the circuit mechanisms that give rise to these effects remain elusive. Further, it remains unclear what the advantages and disadvantages of different homeostatic mechanisms are and when to use which. For example, many homeostatic plasticity mechanisms seek to set a neuron's average firing rate to some target rate, but when would it be appropriate to achieve this goal by modifying intrinsic excitability versus modifying pre- or postsynaptic weights? Indeed, there may be multiple means toward the same end, and it remains unclear what the trade-offs are among these different paths.

We hope these insights provide an avenue for building future collaborations, where computer scientists can use quantitative frameworks to evaluate how different plasticity mechanisms affect neural function. In return, neuroscientists can provide new perspectives on the benefits of normalization in neural networks and inspiration for designing new normalization algorithms based on neurobiological principles.

Acknowledgments

We thank Sanjoy Dasgupta, Alexei Koulakov, Vishaal Krishnan, Ankit Patel, and Shyam Srinivasan for helpful comments on the manuscript. S.N. was supported by the Pew Charitable Trusts, the National Institutes of Health under awards 1R01DC017695 and 1UF1NS111692, and funding from the

Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory. Y.S. was supported by a Swartz Foundation Fellowship.

References

Arora, S., Li, Z., & Lyu, K. (2019). Theoretical analysis of auto rate-tuning by batch normalization. In *Proceedings of the 7th International Conference on Learning Representation*. OpenReview.net.

Arpit, D., Zhou, Y., Kota, B. U., & Govindaraju, V. (2016). Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In *Proceedings of the 33rd International Conference on Machine Learning* (pp. 1168–1176).

Bakker, A., Krauss, G. L., Albert, M. S., Speck, C. L., Jones, L. R., Stark, C. E., . . . Gallagher, M. (2012). Reduction of hippocampal hyperactivity improves cognition in amnestic mild cognitive impairment. *Neuron*, *74*(3), 467–474. 10.1016/j.neuron.2012.03.023, PubMed: 22578498

Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.*, *2*(1), 32–48. 10.1523/JNEUROSCI.02-01-00032.1982, PubMed: 7054394

Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, R. (Eds.), *Advances in neural information processing systems*, *31* (pp. 7694–7705). Red Hook, NY: Curran.

Buzsaki, G., & Mizuseki, K. (2014). The log-dynamic brain: How skewed distributions affect network operations. *Nat. Rev. Neurosci.*, *15*(4), 264–278. 10.1038/nrn3687, PubMed: 24569488

Cannon, W. (1932). *The wisdom of the body*. New York: Norton. 10.1038/133082a0

Carandini, M., & Heeger, D. J. (2011). Normalization as a canonical neural computation. *Nat. Rev. Neurosci.*, *13*(1), 51–62. 10.1038/nrn3136, PubMed: 22108672

Chistiakova, M., Bannon, N. M., Chen, J. Y., Bazhenov, M., & Volgushev, M. (2015). Homeostatic role of heterosynaptic plasticity: models and experiments. *Front. Comput. Neurosci.*, *9*, 89. 10.3389/fncom.2015.00089, PubMed: 26217218

Davis, G. W. (2006). Homeostatic control of neural activity: From phenomenology to molecular design. *Annu. Rev. Neurosci.*, *29*, 307–323. 10.1146/annurev.neuro.28.061604.135751, PubMed: 16776588

Davis, G. W. (2013). Homeostatic signaling and the stabilization of neural function. *Neuron*, *80*(3), 718–728. 10.1016/j.neuron.2013.09.044, PubMed: 24183022

Desjardins, G., Simonyan, K., Pascanu, R., & Kavukcuoglu, K. (2015). Natural neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*, *28* (pp. 2071–2079). Red Hook, NY: Curran.

Fox, K., & Stryker, M. (2017). Integrating Hebbian and homeostatic plasticity: Introduction. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, *372* (1715), 20160413. 10.1098/rstb.2016.0413, PubMed: 28093560

Ganguli, D., & Simoncelli, E. P. (2010). Implicit encoding of prior probabilities in optimal neural populations. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems*, *23* (pp. 658–666). Red Hook, NY: Curran.

Ganguli, D., & Simoncelli, E. P. (2014). Efficient sensory encoding and Bayesian inference with heterogeneous neural populations. *Neural Comput.*, *26*(10), 2103–2134.

Graham, D. J., Chandler, D. M., & Field, D. J. (2006). Can the theory of "whitening" explain the center-surround properties of retinal ganglion cell receptive fields? *Vision Res.*, *46*(18), 2901–2913. 10.1016/j.visres.2006.03.008, PubMed: 16782164

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, *95*(2), 245–258. 10.1016/j.neuron.2017.06.011, PubMed: 28728020

Houweling, A. R., Bazhenov, M., Timofeev, I., Steriade, M., & Sejnowski, T. J. (2005). Homeostatic synaptic plasticity can explain post-traumatic epileptogenesis in chronically isolated neocortex. *Cereb. Cortex*, *15*(6), 834–845. 10.1093/cercor/bhh184, PubMed: 15483049

Huang, C. G., Zhang, Z. D., & Chacron, M. J. (2016). Temporal decorrelation by SK channels enables efficient neural coding and perception of natural stimuli. *Nat. Commun.*, *7*, 11353. 10.1038/ncomms11353, PubMed: 27088670

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning* (pp. 448–456).

Joseph, A., & Turrigiano, G. G. (2017). All for one but not one for all: Excitatory synaptic scaling and intrinsic excitability are coregulated by caMKIV, whereas inhibitory synaptic scaling is under independent control. *Journal of Neuroscience*, *37*(28), 6778–6785. 10.1523/JNEUROSCI.0618-17.2017

Keck, T., Toyoizumi, T., Chen, L., Doiron, B., Feldman, D. E., Fox, K., . . . van Rossum, M. C. (2017). Integrating Hebbian and homeostatic plasticity: The current state of the field and future research directions. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, *372*(1715). 10.1098/rstb.2016.0158, PubMed: 28093552

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In I. Guyon, Y. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, *30* (pp. 971–980): Red Hook, NY: Curran.

Kohler, J., Daneshmand, H., Lucchi, A., Hofmann, T., Zhou, M., & Neymeyr, K. (2019). Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In K. Chaudhuri & M. Sugiyamav (Eds.), *Proceedings of Machine Learning Research* (pp. 806–815).

Krishnan, G. P., Tadros, T., Ramyaa, R., & Bazhenov, M. (2019). *Biologically inspired sleep algorithm for artificial neural networks.* arXiv:1908.02240.

Kukacka, J., Golkov, V., & Cremers, D. (2017). Regularization for deep learning: A taxonomy. *CoRR*, abs/1710.10686.

Lang, K. J., & Hinton, G. E. (1990). *Dimensionality reduction and prior knowledge in e-set recognition*. San Mateo, CA: Morgan Kaufmann.

Laughlin, S. (1981). A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c*, *36*(9–10), 910–912. 7303823

Laughlin, S. B., & Sejnowski, T. J. (2003). Communication in neuronal networks. *Science*, *301*(5641), 1870–1874. 10.1126/science.1089662, PubMed: 14512617

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, *86*(11), 2278–2324. 10.1109/5.726791

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop. In G. Orr & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 9–50). Berlin: Springer-Verlag. 10.1007/978-3-642-35289-8_3

Lei Ba, J., Kiros, J. R., & Hinton, G. E. (2016). *Layer normalization*. arXiv:1607.06450.

Li, H., Li, Y., Lei, Z., Wang, K., & Guo, A. (2013). Transformation of odor selectivity from projection neurons to single mushroom body neurons mapped with dual-color calcium imaging. In *Proc. Natl. Acad. Sci. U.S.A.*, *110*(29), 12084–12089.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*, 335–346. 10.1038/s41583-020-0277-3

Luo, P., Wang, X., Shao, W., & Peng, Z. (2018). *Towards understanding regularization in batch normalization*. arXiv:1809.00846.

Lynch, N. A. (1996). *Distributed algorithms*. San Mateo, CA: Morgan Kaufmann.

Maffei, A., & Fontanini, A. (2009). Network homeostasis: A matter of coordination. *Curr. Opin. Neurobiol.*, *19*(2), 168–173. 10.1016/j.conb.2009.05.012, PubMed: 19540746

Malnic, B., Hirono, J., Sato, T., & Buck, L. B. (1999). Combinatorial receptor codes for odors. *Cell*, *96*(5), 713–723. 10.1016/s0092-8674(00)80581-4, PubMed: 10089886

Mante, V., Frazor, R. A., Bonin, V., Geisler, W. S., & Carandini, M. (2005). Independence of luminance and contrast in natural scenes and in the early visual system. *Nat. Neurosci.*, *8*(12), 1690–1697.

Marder, E., & Goaillard, J. M. (2006). Variability, compensation and homeostasis in neuron and network function. *Nat. Rev. Neurosci.*, *7*(7), 563–574. 10.1038/nrn1949, PubMed: 16791145

Murthy, V. N., Schikorski, T., Stevens, C. F., & Zhu, Y. (2001). Inactivity produces increases in neurotransmitter release and synapse size. *Neuron*, *32*(4), 673–682. 10.1016/s0896-6273(01)00500-1, PubMed: 11719207

Neyshabur, B., Tomioka, R., Salakhutdinov, R., & Srebro, N. (2017). *Geometry of optimization and implicit regularization in deep learning*. arXiv:1705.03071.

Olsen, S. R., Bhandawat, V., & Wilson, R. I. (2010). Divisive normalization in olfactory population codes. *Neuron*, *66*(2), 287–299. 10.1016/j.neuron.2010.04.009, PubMed: 20435004

Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, *14*(4), 481–487. 10.1016/j.conb.2004.07.007

Pitkow, X., & Meister, M. (2012). Decorrelation and efficient coding by retinal ganglion cells. *Nat. Neurosci.*, *15*(4), 628–635. 10.1038/nn.3064, PubMed: 22406548

Poggio, T., Liao, Q., & Banburski, A. (2020). Complexity control by gradient descent in deep networks. *Nat. Commun.*, *11*(1), 1027.

Pozo, K., & Goda, Y. (2010). Unraveling mechanisms of homeostatic synaptic plasticity. *Neuron*, *66*(3), 337–351. 10.1016/j.neuron.2010.04.028, PubMed: 20471348

Pozzorini, C., Naud, R., Mensi, S., & Gerstner, W. (2013). Temporal whitening by power-law adaptation in neocortical neurons. *Nat. Neurosci.*, *16*(7), 942–948. 10.1038/nn.3431

Priebe, N. J., & Ferster, D. (2002). A new mechanism for neuronal gain control (or how the gain in brains has mainly been explained). *Neuron*, *35*(4), 602–604. 10.1016/s0896-6273(02)00829-2, PubMed: 12194862

Prinz, A. A., Bucher, D., & Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nat. Neurosci.*, *7*(12), 1345–1352. 10.1038/nn1352, PubMed: 15558066

Rabinowitch, I., & Segev, I. (2008). Two opposing plasticity mechanisms pulling a single synapse. *Trends Neurosci.*, *31*(8), 377–383. 10.1016/j.tins.2008.05.005

Rabinowitz, N. C., Willmore, B. D., Schnupp, J. W., & King, A. J. (2011). Contrast gain control in auditory cortex. *Neuron*, *70*(6), 1178–1191. 10.1016/j.neuron.2011.04.030, PubMed: 21689603

Raiko, T., Valpola, H., & Lecun, Y. (2012). Deep learning made easier by linear transformations in perceptrons. In N. D. Lawrence & M. Girolami (Eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*.

Ren, M., Liao, R., Urtasun, R., Sinz, F. H., & Zemel, R. S. (2017). Normalizing the normalizers: Comparing and extending network normalization schemes. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net.

Rodieck, R. (1998). *The first steps in seeing*. Sunderland, MA: Sinauer.

Royer, S., & Pare, D. (2003). Conservation of total synaptic weight through balanced synaptic depression and potentiation. *Nature*, *422*(6931), 518–522. 10.1038/nature01530, PubMed: 12673250

Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*, *29* (pp. 901–909). Red Hook, NY: Curran.

Sanchez-Giraldo, L. G., Laskar, M. N. U., & Schwartz, O. (2019). Normalization and pooling in hierarchical models of natural images. *Curr. Opin. Neurobiol.*, *55*, 65–72. 10.1016/j.conb.2019.01.008, PubMed: 30785005

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, (Eds.), *Advances in neural information processing systems*, *31* (pp. 2483–2493. Red Hook, NY: Curran.

Schwartz, O., & Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nat. Neurosci.*, *4*(8), 819–825. 10.1038/90526, PubMed: 11477428

Shapley, R. (1997). Retinal physiology: Adapting to the changing scene. *Curr. Biol.*, *7*(7), R421–423.

Slomowitz, E., Styr, B., Vertkin, I., Milshtein-Parush, H., Nelken, I., Slutsky, M., & Slutsky, I. (2015). Interplay between population firing stability and single neuron dynamics in hippocampal networks. *eLife*, *4*. 10.7554/eLife.04378, PubMed: 25556699

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

Stevens, C. F. (2015). What the fly's nose tells the fly's brain. In *Proc. Natl. Acad. Sci. U.S.A.*, *112*(30), 9460–9465.

Stevens, C. F. (2016). A statistical property of fly odor responses is conserved across odors. In *Proc. Natl. Acad. Sci. U.S.A.*, *113*(24), 6737–6742.

Tononi, G., & Cirelli, C. (2012). Time to be SHY? Some comments on sleep and synaptic homeostasis. *Neural Plast.*, *2012*, 415250.

Triesch, J., Vo, A. D., & Hafner, A. S. (2018). Competition for synaptic building blocks shapes synaptic plasticity. *eLife*, *7*. 10.7554/eLife.37836, PubMed: 30222108

Turrigiano, G. G. (2008). The self-tuning neuron: Synaptic scaling of excitatory synapses. *Cell*, *135*(3), 422–435. 10.1016/j.cell.2008.10.008, PubMed: 18984155

Turrigiano, G. (2011). Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Annu. Rev. Neurosci.*, *34*, 89–103. 10.1146/annurev-neuro-060909-153238, PubMed: 21438687

Turrigiano, G. (2012). Homeostatic synaptic plasticity: Local and global mechanisms for stabilizing neuronal function. *Cold Spring Harb. Perspect. Biol.*, *4*(1), a005736. 10.1101/cshperspect.a005736, PubMed: 22086977

Turrigiano, G. G. (2017). The dialectic of Hebb and homeostasis. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, *372*(1715). 10.1098/rstb.2016.0258, PubMed: 28093556

Turrigiano, G. G., & Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.*, *5*(2), 97–107. 10.1038/nrn1327, PubMed: 14735113

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). *Instance normalization: The missing ingredient for fast stylization*. arXiv:1607.08022.

Wang, Z., Stocker, A. A., & Lee, D. D. (2012). Optimal neural tuning curves for arbitrary stimulus distributions: Discrimax, infomax and minimum Lp loss. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 2168–2176). New York: Curran.

Wang, Z., Stocker, A. A., & Lee, D. D. (2016). Efficient neural codes that minimize Lp reconstruction error. *Neural Comput.*, *28*(12), 2656–2686. 10.1162/NECO_a_00900, PubMed: 27764595

Wanner, A. A., & Friedrich, R. W. (2020). Whitening of odor representations by the wiring diagram of the olfactory bulb. *Nat. Neurosci.*, *23*(3), 433–442. 10.1038/s41593-019-0576-z

Weber, A. I., Krishnamurthy, K., & Fairhall, A. L. (2019). Coding principles in adaptation. *Annu. Rev. Vis. Sci.*, *5*, 427–449. 10.1146/annurev-vision-091718-014818, PubMed: 31283447

Wiskott, L., & Sejnowski, T. (1998). Constrained optimization for neural map formation: A unifying framework for weight growth and normalization. *Neural Computation*, *10*(3), 671–716. 10.1162/089976698300017700, PubMed: 9527838

Wondolowski, J., & Dickman, D. (2013). Emerging links between homeostatic synaptic plasticity and neurological disease. *Front. Cell. Neurosci.*, *7*, 223. 10.3389/fncel.2013.00223, PubMed: 24312013

Wu, X., Dobriban, E., Ren, T., Wu, S., Li, Z., Gunasekar, S., Ward, R., & Liu, Q. (2019). *Implicit regularization of normalization methods*. arXiv:1911.07956.

Wu, Y.. & He, K. (2018). Group normalization. In *Proceedings of the European Conference on Computer Vision*. Berlin: Springer.

Yu, H., Sternad, D., Corcos, D. M., & Vaillancourt, D. E. (2007). Role of hyperactive cerebellum and motor cortex in Parkinson's disease. *NeuroImage*, *35*(1), 222–233. 10.1016/j.neuroimage.2006.11.047, PubMed: 17223579

Zenke, F., & Gerstner, W. (2017). Hebbian plasticity requires compensatory processes on multiple timescales. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, *372*(1715). 10.1098/rstb.2016.0259, PubMed: 28093557

Zhang, W., & Linden, D. J. (2003). The other side of the engram: Experience-driven changes in neuronal intrinsic excitability. *Nat. Rev. Neurosci.*, *4*(11), 885–900. 10.1038/nrn1248, PubMed: 14595400

Ziv, Y., Burns, L. D., Cocker, E. D., Hamel, E. O., Ghosh, K. K., Kitch, L. J., . . . Schnitzer, M. J. (2013). Long-term dynamics of CA1 hippocampal place codes. *Nat. Neurosci.*, *16*(3), 264–266. 10.1038/nn.3329

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *67*(2), 301–320. https://www.jstor.org/stable/3647580

Received March 15, 2021; accepted June 14, 2021.