Emergence of Hebbian Dynamics in Regularized Non-Local Learners

David Koplow DKOPLOW@MIT.EDU

MIT CBMM **Tomaso Poggio**
MIT CBMM **Liu Ziyin**
MIT RLE | NTT Research

Abstract

Stochastic gradient descent (SGD) is often viewed as biologically implausible, while local Hebbian rules dominate theories of synaptic plasticity in our brain. We derive and empirically demonstrate that SGD with weight decay can naturally produce Hebbian-like dynamics near stationarity, whereas injected gradient noise can flip the alignment to be anti-Hebbian. The effect holds for nearly any learning rule, even some random ones, revealing Hebbian behavior as an emergent epiphenomenon of deeper optimization dynamics during training. These results narrow the gap between artificial and biological learning and caution against treating observed Hebbian signatures as evidence against global error-driven mechanisms in our brains. For machine learning, our results shed light on how regularization and noise lead to feature-learning behaviors in the model.

1. Introduction

Hebbian and anti-Hebbian plasticity are the most common types of plasticity experimentally observed in the brain [6, 8, 10, 24]. It is a longstanding belief in neuroscience that Hebbian learning—often summarized as "cells that fire together wire together"–is fundamentally distinct from gradient descent [4]. While Hebbian learning is local and biologically plausible, gradient-based optimization is widely regarded as nonlocal, requiring access to global error signals and precise coordination across layers—properties not generally supported by neural circuits in the brain. As a result, gradient descent has been largely dismissed as biologically implausible [9, 16, 19], despite its centrality to modern machine learning.

However, this separation between artificial and biological learning may be less stark than previously thought. There is some apparent resemblance between Hebbian learning and SGD. Hebbian learning requires weight decay or forms of normalization to ensure convergence, as the core Hebbian principle functions primarily as a learning signal [14]. Similarly, in SGD with weight decay, SGD serves as the learning signal, while weight decay acts as a regularization mechanism that promotes robustness and generalization.

In this work, we discover deeper connections between SGD and Hebbian learning. We demonstrate that the standard training routines used in deep learning—specifically Stochastic Gradient Descent (SGD) with weight decay and noise—can give rise to learning dynamics that are phenomenologically indistinguishable from Hebbian and anti-Hebbian plasticity. Our contributions demonstrate that:

1. close to stationarity, almost any learning rule (including SGD) with weight decay will look like a Hebbian rule; and the correlation increases monotonically as we use a larger weight decay;

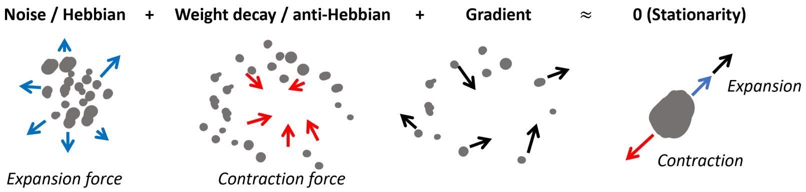

Figure 1: Close to the end of learning, the contraction forces must balance with the expansion forces. For deep learning, the noise and weight decay are, respectively, expansion and contraction forces. When they do not balance, the gradient must fill in the gap – if noise outweighs weight decay, the gradient must appear contractive; otherwise, it appears expansive. Similarly, for biology, the Hebbian dynamics is always expansive, and the anti-Hebbian dynamics is always contractive. Thus, the gradient will look like, and become aligned with, the Hebbian or anti-Hebbian rule depending on whether it is expansive or contractive.

2. when we inject noise into the learning rule, the learning rule will look like an anti-Hebbian rule, and the effect also becomes stronger as the noise gets stronger;

A visualization can be seen in Figure 1. The correlation we observe between weight motion and Hebbian/anti-Hebbian alignment is consistent with the idea that the network is starting to reshape its own intermediate features rather than staying static – a sign of feature learning [23]. Notably, stronger decay tends to coincide with Hebbian-like expansion that counter-balances shrinkage, whereas larger noise levels coincide with anti-Hebbian contraction that tempers stochastic blow-ups—patterns that match the qualitative balance predicted by simple theory and suggest that these macroscopic alignments might be useful markers of how far the dynamics has departed from the NTK manifold [2, 5].

These observations motivate a fundamental question: if Hebbian-like behavior can emerge from the dynamics of gradient-based optimization, are instances of apparent Hebbian learning truly distinct from gradient descent? Or might several cases of biological learning, long attributed to local mechanisms, in fact reflect the emergent behavior of a hidden and deeper optimization principle?

2. Related Works and Preliminaries

Consider a single hidden layer in an arbitrary network:

$$h_b = Wh_a(x), \tag{1}$$

where h_a is the postactivation of the previous layer, and h_b is the preactivation of the current layer. In the most conventional form, the simplest homosynaptic update[1] rule states that W is learned according to

$$\Delta W = s\eta h_b h_a^\top, \tag{2}$$

where $s \in \{-1, 1\}$ is the sign of learning. When $s = 1$, the rule is Hebbian, which states that if neuron i causes neuron j to fire, then their connection should be strengthened. Similarly, when $s = -1$, the rule is anti-Hebbian, as it tends to reduce correlation between neurons. η is a positive time constant, which we call the “learning rate."

There are a few works closely related to ours. Ref. [21] shows the equivalence of gradient descent to a form of contrastive Hebbian algorithm (CHA). However, CHA is not biologically Hebbian

[1] We use this term as a synonym of Hebbian learning.

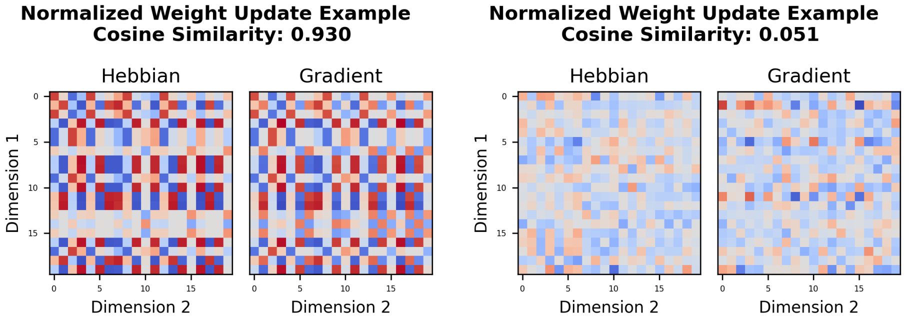

Figure 2: The **left** shows example weight updates with a high alignment between the gradient update ($-\nabla_W \ell$) and the Hebbian update at the end of training with high weight decay, while the **right** image displays an example update at the end of training with no weight decay which has very low alignment. This figure shows a 20x20 subset of the direction of the Hebbian and gradient updates for the second layer of an SCE after training with $\eta = 0.1$, and $\gamma = 0.05$, or $\gamma = 0.0$. Examples of low cosine similarity updates for $\gamma = 0.05$ at the start and end of training can be seen in Figure 10. In general, we find that stronger weight decay, larger learning rate, and larger batch size lead to better alignment (Figures 5 and 6).

because it is not a homosynaptic rule, required by the Hebbian principle, and there is no evidence that the brain can actually implement CHA. In contrast, the original Hebbian rule is easy to implement in the brain. Also, the key role of regularization and noise has not been clarified by this prior work. There have been several other adjacent ideas to modify the Hebbian rule to lead to learning performance similar to gradient descent or even mathematical equivalences to SGD in certain types of models [3, 17, 18, 20]. But all of these approaches stray in some way from the Hebbian rule or fail to provide a truly general relationship between arbitrary models trained with SGD and back propagation.

3. Learning-Regularization Balance Produces Hebbian Learning

Hebbian learning leads to the expansion of weights, and anti-Hebbian learning leads to the contraction of weights. There has been some thought that these two types of dynamics must balance for the brain to reach at least some form of homeostasis (stationarity) [1, 14, 21]. There is a similar effect in gradient-based training in neural networks. The use of weight decay contracts the weights to become smaller, but no useful representation can be learned if the weights are too small. Therefore, any model that reaches some level of stationarity in training must have a gradient signal that is expansive and opposed to the contractive effect of weight decay. This leads to the emergence of Hebbian learning as can be seen in the Proof in Appendix B.1.

Simulations We empirically find that this trend holds across a wide variety of different learning tasks. We ran simulations performing classification on CIFAR-10 and non-linear regression on synthetic data [7]. We tested both small MLPs and transformers, as well as a range of activation functions and optimizers. In some situations, the correlation between the two learning paradigms is very strong (e.g., in Figure 2). To get the alignment between the updates, we compute the cosine similarity of the direction of the gradient update from the loss function (the negative gradient in SGD) and the direction of the Hebbian update. For more details on our Standard Classification Experiment

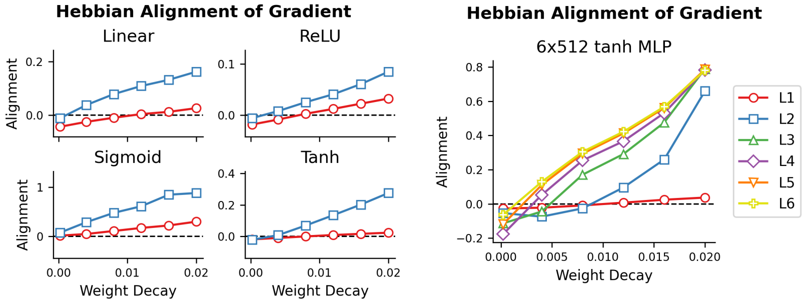

Figure 3: The diagram on the **left** shows that the trend of weight decay increasing Hebbian alignment of the gradient update is robust across different activation functions. The diagram on the **right** shows that the trend can generalize to deeper networks. The SCE MLPs were modified by varying the activation functions (**left**) or the number and size of the hidden vectors (**right**). For a small (or zero) weight decay, the learning sometimes has a weak anti-Hebbian alignment, signaled by a negative alignment to Hebbian learning. While we sometimes see a positive alignment in larger networks, often, some layers become more anti-Hebbian and others become more Hebbian with stronger weight decay; the deeper explanation as to why is a topic for future research. For larger weight decays than those plotted, the model's weights tend to collapse to zero.

(SCE) on CIFAR-10 and our Standard Regression Experiment (SRE) on synthetic data, see Appendix B.4. Any variations to SCEs and SREs will be reported when relevant.

Classification We trained a series MLPs using our SCE setup to classify CIFAR-10. As can be seen in Figure 3, as weight decay increases, so too does the alignment of the SGD gradient with the Hebbian update. The trend persists across different activation functions. However, we find that we can still detect this trend in larger MLPS (Figure 3), though often we see some layers behave in an anti-Hebbian direction as weight increases. Unsurprisingly, when one trains an identical model using a Hebbian learning rule such as weight-normalized Hebbian updates or Oja's rule, the model performs poorly and there is no alignment at convergence with the SGD gradient as seen in Figure 9 [14]; there is no reason to expect unsupervised learning to produce similar dynamics near convergence as supervised learning.

Regression We also evaluate the generalization of this trend to student-teacher regression problems as described in SRE. We explored both MLP and Transformer models and evaluated the Hebbian alignment for learning rules outside of SGD. Since a key prediction of the theory is that almost any learning rule with weight decay can look Hebbian, we test update rules beyond SGD. Namely, we considered SGD, Adam, and Direct Feedback Alignment (DFA) [13]. To demonstrate that this observation is universal, we also run a setting with a randomly initialized neural network whose output is used as a learning signal to each layer of the student network, based entirely on the input data (Random NN). To have non-trivial dynamics (always converging or always collapsing to zero), we choose the sign of the weight update to be whichever direction pushes the norm of the updated weights closer to some constant greater than zero. Results are shown in Table 1. All of these simulations produced Hebbian phenomena that scale with weight decay near convergence. The universality of this relationship for so many learning rules suggests that perhaps our brain is using a non-Hebbian learning rule as well, even if the neurophysiological data have so far seemed to support Hebbian-like plasticity

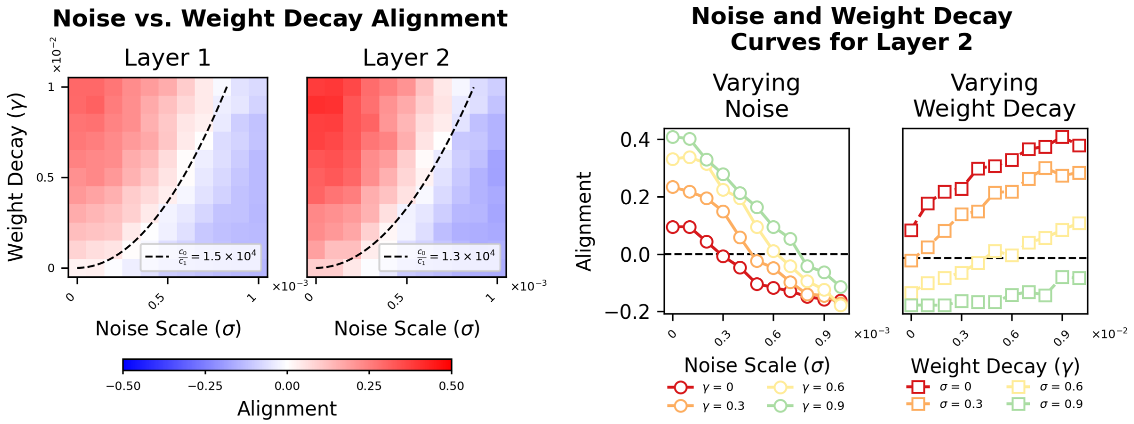

Figure 4: As the noise increases, the Hebbian alignment decreases and higher weight decays lead to higher Hebbian alignment (**right**). The figure on the **left** displays a heatmap of the Hebbian alignment of the gradient at convergence for a number of different additive noises and weight decays; there is a clear quadratic curve at zero-alignment as predicted by the theory. The SRE was augmented by adding noise to each parameter at the start of each iteration with a mean of zero and the specified standard deviation on the diagram. The trend is even clearer when we follow the behavior of varying the noise of a specific weight decay (Varying Noise) or the weight decay of a specific noise standard deviation (Varying Weight Decay).

4. Learning-Noise Balance Leads to Anti-Hebbian Learning

We have (at least partially) answered the question of when Hebbian learning can be an emergent and phenomenological byproduct. A remaining question is when we will see anti-Hebbian learning, as both Hebbian and anti-Hebbian learning are ubiquitous in the brain. Can anti-Hebbian learning also be a byproduct of more complicated learning rules? It turns out that it can, as described in the proof in Appendix B.2.

Simulations We ran experiments to validate the noise prediction using SREs and varying both the variance of the Gaussian noise added to the parameters at each training step, as well as the weight decay. There is a very smooth alignment trend with SGD, as can be seen in Figure 4. The white region shows the the phase boundary between the Hebbian phase and anti-Hebbian phase, and shows a shape in accordance with the quadratic curve $\gamma \approx \sigma^2$. At convergence, the Hebbian alignment of the gradient is higher in low noise environments, and becomes more aligned with anti-Hebbian as the noise increases (Figure 4). Interestingly, we found that solutions with high generalization generally had low Hebbian and anti-Hebbian alignment (Figure 11).

While we observed this trend with other optimizers such as Adam (Figure 7), we struggled to robustly reproduce it outside of the last few layers of much larger networks or those doing different learning tasks. We hypothesize this could be because the learning tasks for those models aren't as sensitive to the magnitude of the weights. A more rigorous explanation is a topic for further research. We also found that adding other types of biologically plausible constraints during learning, such as a sparsification term on layer activations, can lead to a stronger anti-Hebbian alignment of the gradient.

References

[1] Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

[2] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[3] Maxence M Ernoult, Fabrice Normandin, Abhinav Moudgil, Sean Spinney, Eugene Belilovsky, Irina Rish, Blake Richards, and Yoshua Bengio. Towards scaling difference target propagation by learning backprop targets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5968–5987. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ernoult22a.html.

[4] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[5] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[6] Giacomo Koch, Viviana Ponzo, Francesco Di Lorenzo, Carlo Caltagirone, and Domenica Veniero. Hebbian and anti-hebbian spike-timing-dependent plasticity of human cortico-cortical connections. *Journal of Neuroscience*, 33(23):9725–9733, 2013.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

[8] Karri P Lamsa, Joost H Heeroma, Peter Somogyi, Dmitri A Rusakov, and Dimitri M Kullmann. Anti-hebbian long-term potentiation in the hippocampal feedback inhibitory circuit. *Science*, 315(5816):1262–1266, 2007.

[9] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[10] John Lisman. A mechanism for the hebb and the anti-hebb processes underlying learning and memory. *Proceedings of the National Academy of Sciences*, 86(23):9574–9578, 1989.

[11] Kangqiao Liu, Liu Ziyin, and Masahito Ueda. Noise and fluctuation of finite learning rate stochastic gradient descent, 2021.

[12] Michael London, Arnd Roth, Lisa Beeren, Michael Häusser, and Peter E Latham. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466(7302): 123–127, 2010.

[13] Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[14] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267—-273, 1982.

[15] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[17] Benjamin Scellier and Yoshua Bengio. Equivalence of equilibrium propagation and recurrent backpropagation. *Neural Computation*, 31(2):312–329, 2019.

[18] Benjamin Scellier, Anirudh Goyal, Jonathan Binas, Thomas Mesnard, and Yoshua Bengio. Generalization of equilibrium propagation to vector field dynamics. *arXiv preprint arXiv:1808.04873*, 2018. URL https://arxiv.org/abs/1808.04873.

[19] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

[20] Will Xiao, Honglin Chen, Qianli Liao, and Tomaso Poggio. Biologically-plausible learning algorithms can scale to large datasets. In *International Conference on Learning Representations, (ICLR 2019)*, 2019.

[21] Xiaohui Xie and H Sebastian Seung. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454, 2003.

[22] Mengjia Xu, Tomer Galanti, Akshay Rangamani, Lorenzo Rosasco, and Tomaso Poggio. The janus effects of sgd vs gd: high noise and low rank. 2023.

[23] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[24] Friedemann Zenke and Wulfram Gerstner. Hebbian plasticity requires compensatory processes on multiple timescales. *Philosophical transactions of the royal society B: biological sciences*, 372(1715):20160259, 2017.

[25] Liu Ziyin, Isaac Chuang, Tomer Galanti, and Tomaso Poggio. Formation of representations in neural networks. *arXiv preprint arXiv:2410.03006*, 2024.

Appendix A. Reproduction

All experiments were run on MIT's OpenMind cluster using Quadro RTX 6000 GPUs and cumulatively took under 50 hours of compute time. Proof of concept code can be run on a single GPU in Google Collab for about 20 minutes. Example code can be found on our GitHub: [Code is in Supplementary Material in ZIP File].

Appendix B. Theory

B.1. Weight Decay Leads to Hebbian Dynamics

For the layer defined in Eq. (1), the dynamics of SGD states that

$$\Delta W = -\eta(\nabla_{h_b(x)} \ell h_a^T(x) + \gamma W), \tag{3}$$

where η is the learning rate, and γ is the strength of weight decay. At stationarity, the update should be zero in expectation:

$$\mathbb{E}_x[\nabla_{h_b(x)} \ell h_a^T(x)] + \gamma W = 0, \tag{4}$$

where $\mathbb{E}$ denote averaging over the training set. Multiplying W^T on both sides and taking the trace of the matrix, we obtain that

$$\mathrm{Tr}\mathbb{E}_x[\nabla_{h_b(x)} \ell h_b^T(x)] = \mathbb{E}_x[\nabla_{h_b(x)}^T \ell h_b(x)] = -\gamma \mathrm{Tr}[WW^T] < 0 \tag{5}$$

Therefore, on average, $\nabla_{h_b(x)}^T \ell h_b(x)$ is negative.

Now, as in [25], we assume a weak decoupling condition: $\|h_a\|^2 = \mathbb{E}[\|h_a\|^2]$, which states that norms of all representations are rather close to each other. This is certainly satisfied when, for example, there is a neural collapse [15] or when the representations are normalized. This means that the expected alignment between the SGD update and the Hebbian update is given by

$$\mathbb{E}\left[\mathrm{Tr}[\underbrace{-\nabla_{h_b(x)} \ell h_a^T(x)}_{\text{SGD update}}\, \underbrace{h_a(x) h_b^T(x)}_{\text{Hebbian update}}]\right] = -\mathbb{E}[\|h_a\|^2]\mathbb{E}_x[\nabla_{h_b(x)}^T \ell h_b(x)] > 0. \tag{6}$$

Namely, SGD updates will have a positive correlation with the Hebbian update, and the alignment becomes stronger as γ increases. See Figure 2 for an example of such alignment. It turns out that this result is true not only for SGD but also for any update rule, precisely because the weight decay is always aligned with the anti-Hebbian update. Any update rule that reaches a stationarity with weight decay needs to look anti-Hebbian.

Let the weight W be updated according to

$$\Delta W = g(x, \theta) - \gamma W, \tag{7}$$

where g is the learning rule and θ is the entirety of all trainable (plastic) parameters. At stationarity, we have that

$$\mathbb{E}_x[g(x, \theta)] = \gamma W. \tag{8}$$

Now, consider the cosine similarity of the learning rule with the Hebbian rule:

$$\begin{aligned}\mathrm{Tr}\left[\mathbb{E}_x[g(x,\theta)]\mathbb{E}_x[h_a h_b^T]\right] &= \gamma \mathrm{Tr}\left[W\mathbb{E}_x[h_a h_b^T]\right] &(9)\\ &= \gamma \mathrm{Tr}\left[W\mathbb{E}_x[h_a h_b^T]\right] &(10)\\ &= \gamma \mathbb{E}[\|h_b\|^2] > 0. &(11)\end{aligned}$$

Thus, the update must have a positive alignment with the Hebbian rule on average. This shows an intriguing and yet surprising fact: any algorithm with weight decay may look like a Hebbian rule, and the Hebbian rule may just be a “universal" projection of more complicated algorithms.

B.2. Noise Leads to Anti-Hebbian dynamics

The analysis in the previous section does not take into account the existence of noise in learning. In reality, noise is always non-negligible both in biological learning and in artificial learning. That a strong noise leads to an anti-Hebbian gradient can already be explained by looking at a simple linear regression problem:

$$\ell(w) = (w^T x - y)^2, \tag{12}$$

where $x \in \mathbb{R}^d,\ y \in \mathbb{R}$ are sampled from some underlying distribution at every training step. Here, we inject noise $\epsilon \in \mathcal{N}(0, \sigma I)$ to the weight before every optimization step so that

$$w = v + \epsilon, \tag{13}$$

where v is the weight before noise injection. This is nothing but thermal noise that can exist ubiquitously in the brain [12]. It can also be seen as an approximate model of the SGD noise, which causes w to fluctuate around the mean [11]. The updates are

$$\Delta_{\mathrm{SGD}} w = -x(w^T x - y), \tag{14}$$

$$\Delta_{\mathrm{Hebb}} w = x w^T x. \tag{15}$$

The alignment between the two is

$$\begin{aligned}\mathbb{E}_\epsilon[(\Delta_{\mathrm{SGD}} w)^T(\Delta_{\mathrm{Hebb}} w)] &= -\|x\|^2 \mathbb{E}_\epsilon\left[(w^T x)^2 - w^T x y\right] &(16)\\ &= -\|x\|^2\left[(v^T x)^2 + \sigma^2\|x\|^2 - v^T x y\right], &(17)\end{aligned}$$

which is negative for sufficiently large σ^2 and any $\|x\| \neq 0$. Thus, large noise leads to anti-Hebbian learning. An interesting question is how this effect competes and trades off with weight decay. When there is a weight decay,

$$\Delta_{\mathrm{SGD}} w = -x(w^T x - y) - \gamma w, \tag{18}$$

and so

$$\mathbb{E}_\epsilon[(\Delta_{\mathrm{SGD}} w)^T(\Delta_{\mathrm{Hebb}} w)] \approx -\sigma^2 c_0 + \gamma c_1, \tag{19}$$

where c_0 and c_1 are positive. Thus, one expects a phase transition boundary at $\gamma \propto \sigma^2$. This scaling law is verified in the experiments (Figure 4), which justifies this simple analysis.

B.3. RandomNN Formulation

The RandomNN was a MLP with 3 hidden vectors of size 128 and tanh activations. The MLP took the same input as the student model but outputted a vector of length 4. The output was averaged across the batch and then and then multiplied by a random projection matrix unique to each parameter and reshaped to be the dimensions of that parameter. No parameters of RandomNN change after initialization. The resulting learning signal for W is a deterministically random low rank matrix, W^*.

The full weight update is given by:

$$\Delta W = \eta \left(g(x, \theta) - \gamma W \right)$$

where

$$g(x, \theta) = p(W^*) s_{dir}(W) s_{red}(W, W^*) W^*$$

and where,

$$s_{red}(W, W^*) = \text{sign} \left(\|W\|_2 - \|W - W^*\|_2 \right)$$

$$s_{dir}(W) = \text{sign} \left(100 - \|W\|_2 \right)$$

$$p(W^*) = \begin{cases} 1 & \text{if } \|W^*\|_2 \leq 1 \\ \frac{1}{\|W^*\|_2 + \epsilon} & \text{otherwise} \end{cases}$$

The minimal requirements to have non-zero weights and reach stationarity require $g(x, \theta)$ to be some forcing function that wants to make the weights larger than zero. This is the case with any descent learning algorithm, as with zero weights, one can not learn or express anything besides 0. However, it is not only true of learning algorithms.

There are a number of trivial constructions that satisfy this condition, such as setting $f(x, \theta) = A$ where A is a random matrix defined at initialization. This will naturally be an expanding force and become aligned with the Hebbian rule, however, it will do this even without regularization. But is there a way to make a non-learning model that doesn't behave Hebbian at all without regularization, but does with regularization?

RandomNN is one such construction. In it, we produce random weight update vectors in a subspace of the possible directions of the student model's weight updates. This means that after some number of updates, the value of the weight is not orthogonal to the random update vectors, and in fact becomes highly aligned to them. Thus, for a given weight update, the norm of the weights will either increase or decrease, not strictly increase. We can make an attractor to push the norm of the weights to a specific non-zero value by choosing to either add or subtract the random update, depending on which one will move it closer to the target value. Thus, without any regularization, the model's weights will converge to have the target norm and will, on average, not increase or decrease, resulting in no Hebbian alignment. However, once a weight decay term is added, the attractor will try to strictly increase to approach the target, and thus align with the Hebbian update. We also apply a weight update norm clip for stability.

B.4. Experimental Design

In our experiments, we used a default learning rate of $\eta = 0.01$ and trained for 50 epochs, which reached convergence. Since this trend only holds near stationarity–a condition achievable in full gradient descent but obscured in SGD by noise–we found it best to use larger batch sizes to compute both the gradient and Hebbian update as suggested in [22]. We found a batch size of 256 to generally show Hebbian phenomena while being small enough to converge to good solutions quickly (Figure 5).

(1) Standard Classification Experiment (SCE): In these experiments, we trained a small MLP with 2 layers of 128 dimensions and tanh activation using cross-entropy loss to classify CIFAR-10.

(2) Standard Regression Experiment (SRE): In these experiments, we trained a small MLP with two hidden layers of 128 units each and `tanh` activation, using mean squared error to predict the output of a teacher model. The teacher has the same architecture but is initialized with different random parameters. Both the input and output vectors are 32-dimensional. The training dataset consisted of 20,000 randomly generated training examples, and the validation dataset contained 2,000 examples. For the transformer variant of the SRE, we used a transformer with 32-dimensional token embeddings, a vocabulary size of 16, and a maximum sequence length of 32. The encoder consists of 2 layers with 4 attention heads and 32-dimensional feed-forward blocks using ReLU activations. The average of the output token embeddings is passed through the same MLP described above and compared to the teacher output.

Appendix C. Experiments

In the following document, we provide additional figures and explanations that were referenced in the main text.

C.1. Additional Influences on Hebbian Alignment and Generalization

C.1.1. BATCH SIZE

See Figure 5.

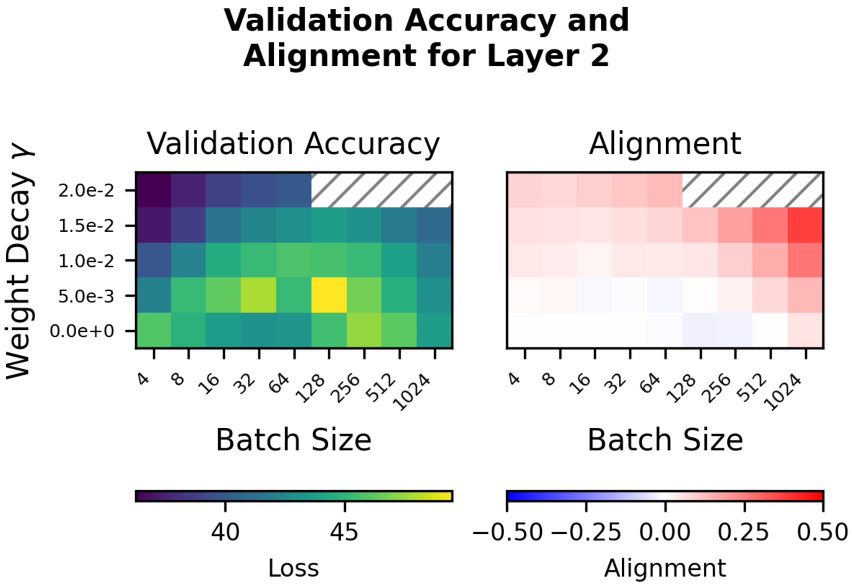

Figure 5: The optimal performance seems to be at a critical position between Hebbian and anti-Hebbian gradient alignment when varying batch size and weight decay. This shows the accuracy (**left**) and the Hebbian alignment of gradient update (**right**) for SCEs with a variety of weight decays and batch sizes. The striped background indicates NaN values.

C.1.2. LEARNING RATE

See Figure 6.

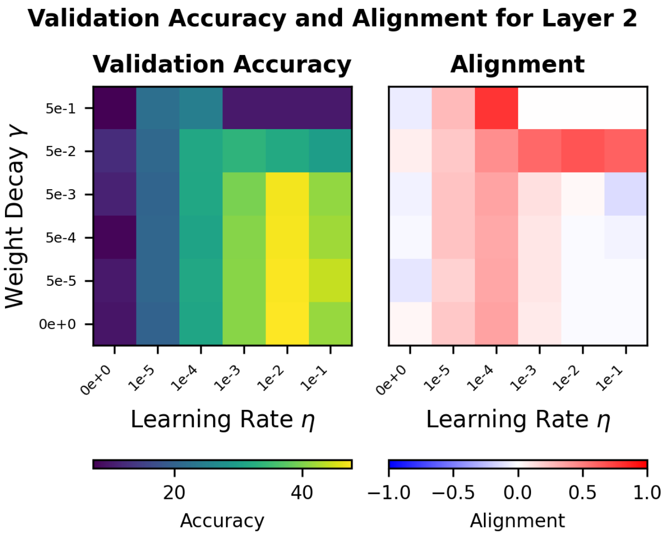

Figure 6: The optimal performance seems to be at a critical position between Hebbian and anti-Hebbian gradient alignment when varying learning weight and weight decay. This shows the accuracy (**left**) and the Hebbian alignment of gradient update (**right**) for SCEs with a variety of weight decays and learning rates.

C.1.3. NOISE

See Figure 7.

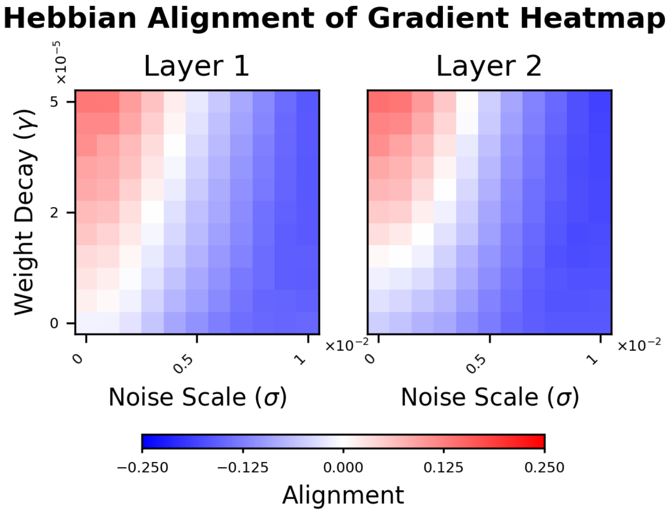

Figure 7: Again, there is a clear trend that even for the Adam optimizer, as noise increases, alignment decreases, and as weight decay increases, so too does alignment. Adam was very sensitive to the parameter ranges for which we'd see the trend, so we used a different weight decay and standard deviation range than the prior experiment. However, the rest of the architecture and experimental setup is identical to that described in Figure 4.

C.1.4. OTHER REGULARIZATION TECHNIQUES

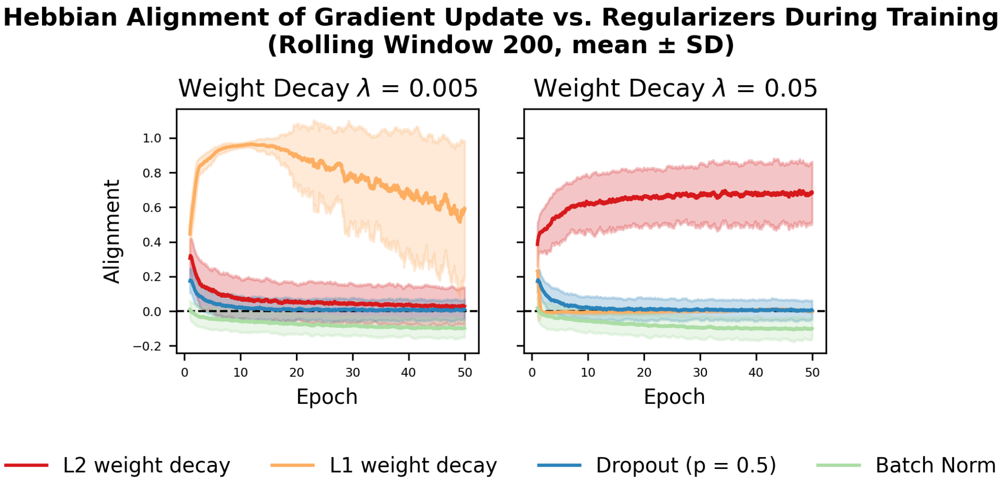

Figure 8: Other regularization techniques have a variety of effects on the Hebbian alignment of the gradient update. While we only developed a theory for L2 weight decay, the alignment seems to exist for some other regularizers as well, such as L1 weight decay and an anti-Hebbian alignment for batch normalization when used to augment SCEs. Batch normalization seems to have an anti-Hebbian effect, while both L1 and L2 weight decay can have a Hebbian effect.

C.2. Example Alignments During Training

C.2.1. HEBBIAN LEARNING RULES DON'T ALIGN WITH SGD

See Figure 9.

C.2.2. LOW ALIGNMENT UPDATE AT END OF TRAINING

See figure 10

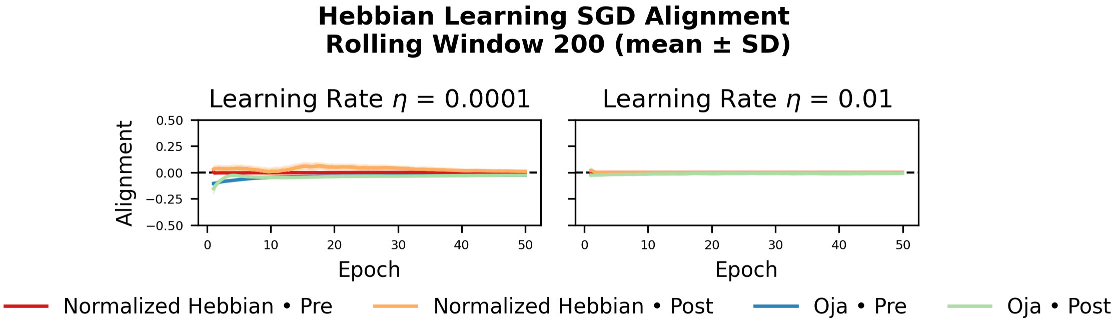

Figure 9: No standard interpretation of Hebbian learning produces alignment with SGD at convergence. The graph shows the mean SGD alignment of the second layer's updates, ± the standard deviation over a 200-iteration window, when trained with various versions of the Hebbian learning rule for two different learning rates. The SCE was modified to learn with various versions of the Hebbian learning rule. The *Normalized Hebbian* learning rule is the generic Hebbian algorithm with weight standardization after every step. The second algorithm is Oja's rule. We also tested the pre-activation and post-activation versions of both. The average alignment of every combination approaches zero.

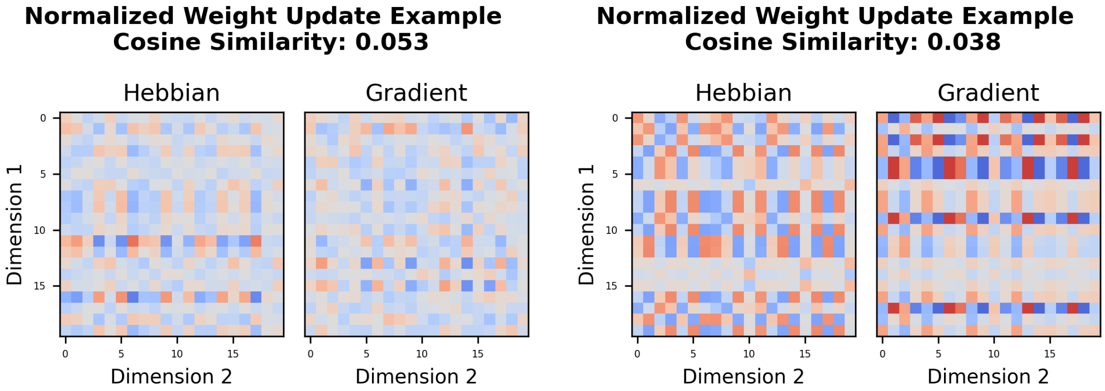

Figure 10: With weight decay, even after the first epoch (**left**), there starts to be an alignment of the directions; at convergence (**right**), even when specific steps have low cosine similarity, there is still clearly a lot of similar structure. At the end of training, many weight updates with low Hebbian alignment still share a surprising amount of structure. The plots above are from a SCE with $\eta = 0.1$ and $\gamma = 0.05$.

C.3. Model Performance is Correlated with Low Alignment

See Figure 11

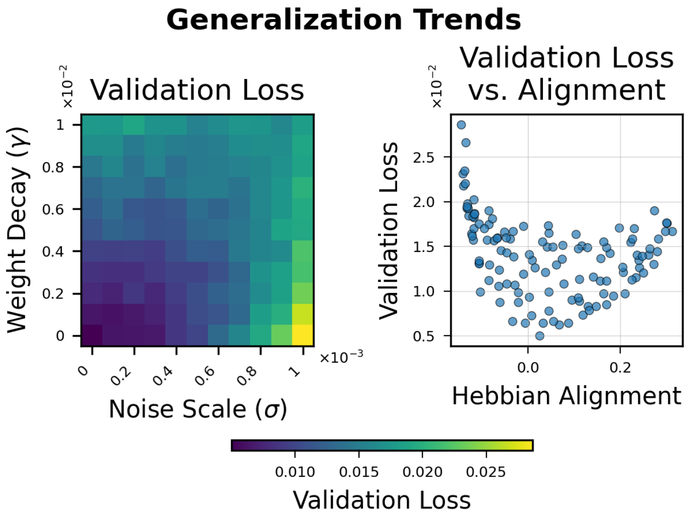

Figure 11: Best performance of the model is achieved when it is not Hebbian or anti-Hebbian on average. The **left** image displays the student validation loss for the experiment in Figure 4, while the **right** image shows a scatter plot of the validation loss vs. Hebbian alignment of the gradient. There seems to be some weak saddle phenomena in loss that occur at the phase transition boundary of Hebbian alignment with respect to noise and scale. The validation loss reduces as both weight decay and noise get smaller.

Table 1: For all models, optimizers and learning rules, Hebbian alignment rises with increasing weight-decay γ. Hebbian alignment (mean ± SD, $n = 10$) at convergence is shown for the 2nd-layer gradient in a regression MLP and a sequence-to-vector transformer (1st layer for DFA). All experiments were SREs with a few modifications outside of the learning rule and weight decay specified in the table. DFA used $\eta = 0.1$ with gradient-norm $clip = 5$ and, as in the original implementation, used biases. RandomNN used gradient-norm $clip = 1$ and a target weight L2 norm of 100 to determine the sign of the update as explained in Section B.3 of the Appendix.

Model	Learning Rule	Weight Decay (γ)			
		0	5×10^{-5}	5×10^{-4}	5×10^{-3}
Regression MLP	Adam	-0.02 ± 0.00	0.10 ± 0.00	$\mathbf{0.66 \pm 0.01}$	–
	SGD	-0.10 ± 0.01	-0.06 ± 0.01	0.17 ± 0.01	$\mathbf{0.59 \pm 0.01}$
	DFA	0.45 ± 0.05	0.45 ± 0.04	0.68 ± 0.05	$\mathbf{0.87 \pm 0.00}$
	RandomNN	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.00	$\mathbf{0.50 \pm 0.00}$
Transformer	Adam	-0.02 ± 0.02	0.50 ± 0.24	$\mathbf{0.99 \pm 0.02}$	–
	SGD	0.00 ± 0.01	0.04 ± 0.01	0.47 ± 0.06	$\mathbf{0.88 \pm 0.03}$
	DFA	0.08 ± 0.03	0.07 ± 0.02	0.11 ± 0.02	$\mathbf{0.12 \pm 0.02}$
	RandomNN	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	$\mathbf{0.09 \pm 0.01}$

Appendix D. Experiment Table

See Table 1.