Robust Supervised Hebbian Learning via Confidence Weighting

Student Erin Tan

SID 3036882210

Abstract

While Hebbian learning rules have proven foundational in computational neuroscience, their fundamental assumption that "neurons that fire together, wire together" becomes problematic when confronted with noisy, corrupted, or anomalous data. Traditional Hebbian learning and its variants (e.g., Oja's rule, Sanger's rule) treat all co-activations with equal importance, lacking mechanisms to discriminate between meaningful and spurious correlations, which is not consistent with human learning mechanisms. In this work, we propose a novel modification to Sanger's Rule that incorporates a Bayesian confidence-weighting mechanism, leveraging likelihood as a proxy for confidence to assign greater weight to more reliable data points. Our approach also introduces a supervised framework that preserves the biological plausibility of Hebbian updates while also providing the ability to learn from a concrete error signal, allowing us to bridge the gap between existing Hebbian and gradient-based learning algorithms, such as backpropagation. We call the method Robust Supervied Hebbian Learning. We evaluate this method on benchmark datasets, demonstrating its effectiveness in improving convergence time and accuracy. The code for this project is published at `https://github.com/etan18/confidence-weighted-hebbian`.

1 Introduction

Today, backpropagation is the most commonly used learning algorithm for training deep neural networks. However, one of the greatest criticisms of the technique is that it is not biologically plausible for a number of reasons, particularly because it is a global training procedure that requires weight transport across many hidden layers [Bengio et al., 2016]. In practice, these qualities also serve as limitations for the ability to parallelize training operations and lead to higher latency. To address these issues, recent works have looked towards biology as inspiration for designing efficient learning algorithms that can compete with the performance outcomes of backpropagation [Miconi, 2017] [Konishi et al., 2023]. In this project, we build off of this line of work to test our own biologically-plausible learning rule, which we call Robust Supervised Hebbian Learning.

1.1 Hebbian Learning

Hebbian learning rules provide a class of biologically-plausible, local update mechanisms for learning model weights. These capabilities are enabled by Donald Hebb's original idea that synaptic weight updates relied only on the correlation between pre- and post-synaptic activity. For a linear neuron $y = \sum_i w_i \cdot x_i$, we can explicitly write out Hebb's rule as follows:

$$w_i \propto \langle y \cdot x_i \rangle \propto \langle \sum_j w_j x_j x_i \rangle \tag{1}$$

$$= \sum_j w_j \langle x_j x_i \rangle \tag{2}$$

Evidently, this class of learning rules becomes unsupervised in nature, looking only at the correlation between neuron inputs to learn patterns from the data, without considering the output label. As it turns out, the learned weight vectors under this single-neuron case of linear Hebbian Learning align with the eigenvectors of the covariance matrix of the input features. Future variants of Hebb's Rule, such as Oja's Rule and Sanger's Rule, model the generalized, multi-neuron case, wherein we are able to learn multiple eigenvectors of the underlying data.

Supervised variations of Hebb's Rule have been proposed to more closely compare to the application of backpropagation [Alemanno et al., 2023]. These methods are able to leverage a defined error signal, providing a clear optimization objective to guide learning. The most prominent instance of this is Contrastive Hebbian Learning (CHL), a generalization of the Hebbian Learning rule which updates the synaptic weights proportionally to the difference in cross-products of activations in a clamped and free running phase [Movellan, 1991]. CHL laid the foundations for showing that, under specific conditions, supervised Hebbian Learning can be comparable to backpropagation in terms of the learned error signal, as well as overall performance [Xie and Seung, 2003].

1.2 Robustness via Confidence Weighting

In the realm of learning theory, robustness is, broadly, the ability of a model to maintain performance under different conditions, particularly in the face of uncertainty [Nobandegani et al., 2019]. Popular learning algorithms like predictive coding [Aitchison and Lengyel, 2017] and delta rule [Worthy et al., 2018] incorporate mechanisms for weighting each point proportionally to their prediction error.

Recent works have sought to improve model robustness by incorporating an understudied aspect of human learning—confidence weighting [Meyniel, 2020]. The premise of confidence weighting is to assign greater influence to more reliable or certain data points during learning to minimize the impact of noise or uncertainty. One method of measuring confidence in a prediction is using surprise, originally proposed by [Shannon, 2001], mathematically defined as the log-improbability of an observation x_i:

$$\text{surprise}(x_i) = -\log p(x_i|x_{1:i-1}) \tag{3}$$

Intuitively, surprise can be thought of as a measure of how unexpected a prediction compared to all previous examples, and can be used to effectively detect outliers in data.

[Meyniel, 2020] found that the mechanism for confidence-weighted learning in the brain conforms with a Bayesian inference approach. They formalize prediction confidence as the log-precision of the posterior distribution with parameters θ over the predicted point:

$$\text{confidence}_\theta(x_i) = -\log\big(\text{Var}[\theta|x_i]\big) \tag{4}$$

Compared to surprise, confidence plays a more regulatory role in learning, and can better stabilize synaptic weight updates [Bounmy et al., 2023]. This paper investigates how incorporating these aforementioned confidence-weighting mechanisms into a supervised Hebbian learning framework can help make the algorithm more robust, making the following contributions:

1. Proposing a modification to the traditional Hebbian Learning rule which incorporates a Bayesian confidence weighting mechanism.

2. Evaluating the efficacy of this method against exiting learning algorithms on several benchmark datasets.

2 Method

The generalized Hebbian algorithm, also known as Sanger's Rule [Sanger, 1989], updates the synaptic weights $\mathbf{w}$ for a set of n neurons such that each neuron learns a different principal component of the input data $\mathbf{x}$. The algorithm achieves this by taking the output projection of the j-th neuron, $y_j = \mathbf{w}_j^\top \mathbf{x}$, and subtracting the learned principal components of all preceding upstream neurons i, where $i \leq j$.

$$\Delta \mathbf{w}_j = \eta \cdot \mathbf{x}\left(y_j - \sum_{k=1}^{j} y_k \mathbf{w}_k^\top \mathbf{x}\right) \tag{5}$$

Under this rule, neurons learn principal components sequentially, such that $\mathbf{w}_1$ will converge to the first principal component, without being affected by downstream neurons. When $j = 1$, Sanger's Rule is equivalent to Oja's Rule, which it extends from. We also introduce a learning rate parameter η.

In this project, we propose a modification to Sanger's Rule:

$$\Delta \mathbf{w}_j = \eta \cdot c \cdot \mathbf{x}\left(y_j - \sum_{k=1}^{j} y_k \mathbf{w}_k^\top \mathbf{x}\right) \tag{6}$$

Here, we introduce a new **confidence** term c to incorporate plasticity into our learning rule, such that each new piece of information may affect the weights differently. We take a Bayesian approach to defining the confidence term, using the likelihood of an input-label pair $z_i = (\mathbf{x}_i, y_i)$ with dimensions $d = \dim(\mathbf{x}) + \dim(y)$, assuming a Gaussian distribution with mean μ and variance σ^2. Through this confidence term, we are also able to introduce a supervised framework to guide learning. This is done by capturing the likelihood of the input-label pair z_i, rather than only looking at the inputs. By also incorporating the label information only in the confidence term, we preserve the biological plausibility of the Hebbian Learning rule in the rest of the equation.

$$c_i := P[z_i|\mu, \sigma^2] = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{||z_i - \mu||_2^2}{2\sigma^2}\right) \tag{7}$$

Importantly, we use the likelihood as a proxy for the probabilistic confidence defined in Equation (4), which offers several practical advantages. Firstly, there is a computational advantage to using likelihood, as precision

calculations require performing a matrix inversion operation on the variance, which is expensive. We can also guarantee numerical stability by bounding our confidence values $c \in [0, 1]$.

The relationship between probabilistic confidence (defined in Equation (4)) and likelihood (defined in Equation (7)) arises from their shared dependence on the variance σ^2 in a Gaussian distribution [Pawitan and Lee, 2021]. Precision, by definition, is inverse variance $\beta = 1/\sigma^2$, and confidence is defined as log-precision, $\log(\beta)$. Confidence reflects global uncertainty in the model, so as σ^2 increases, β and confidence both decrease, representing greater uncertainty. Likelihood, as shown in Equation (7), captures the fit of a data point z_i to the model, considering both its proximity to the mean μ and the spread of the distribution σ^2. As σ^2 increases, likelihood decreases because the underlying distribution is less sharply concentrated.

Now that we have laid the theoretical groundwork for the proposed rule, we need to specify mechanisms to determine the parameters of the Gaussian model, μ and σ^2. We compute these value empirically, replacing them with parameter estimates $\hat{\mu}$ and $\hat{\sigma}^2$, respectively. We store the running emprical mean and variance and update these values as each batch of inputs is processed. Concretely, for a size-N batch $\mathbf{Z} \in \mathbb{R}^{N \times d}$, we find the batch mean and variance:

$$\mu_{\text{batch}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \tag{8}$$

$$\sigma^2_{\text{batch}} = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{z}_i - \mu_{\text{batch}}||_2^2 \tag{9}$$

To update the overall running statistics, we also store a running count of samples processed, $n_{\text{total}} = n_{\text{running}} + N$. We use this to update our parameter estimates:

$$\hat{\mu} \leftarrow \hat{\mu} + \frac{N}{n_{\text{total}}}(\mu_{\text{batch}} - \hat{\mu}) \tag{10}$$

$$\hat{\sigma}^2 \leftarrow \frac{n_{\text{running}} \cdot \hat{\sigma}^2 + N \cdot \sigma^2_{\text{batch}}}{n_{\text{total}}} \tag{11}$$

Under this framework, we expect to effectively estimate the true underlying distribution of the input data as $n_{\text{total}} \to \infty$. Putting this all together, we are left with our final Robust Supervised Hebbian Learning rule:

$$\Delta \mathbf{w}_j = \eta \cdot \frac{1}{(2\pi\hat{\sigma}^2)^{d/2}} \exp\left(- \frac{||z_i - \hat{\mu}||_2^2}{2\hat{\sigma}^2} \right) \cdot \mathbf{x} \left(y_j - \sum_{k=1}^{j} y_k \mathbf{w}_k^\top \mathbf{x} \right) \tag{12}$$

3 Experiments

In this section, we outline the experimental setup, including the datasets used, the baseline learning rules for comparison, and the model each algorithm was used to train.

3.1 Datasets

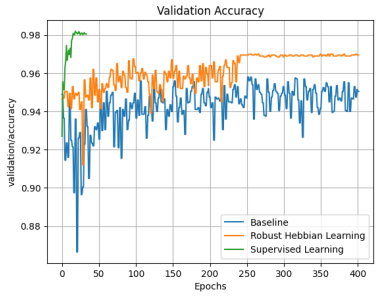
We evaluated Robust Hebbian Learning on two datasets for image classification tasks: MNIST [Deng, 2012] and MNIST-Fashion [Xiao et al., 2017].

Both datasets contain a standard training dataset of 60,000 examples. We randomly split these examples into a 50,000-example training set and 10,000-example validation. The final performance of the model was evaluated on the standard held-out dataset of 10,000 examples.

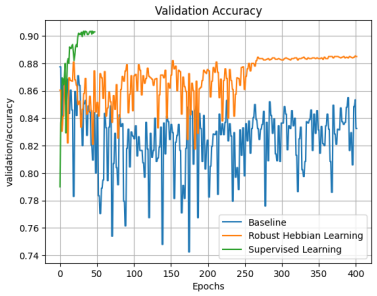
3.2 Baselines

To assess the efficacy of Robust Supervised Hebbian Learning, we compare it with two established learning mechanisms: the Krotov-Hopfield Learning Rule and backpropagation.

Krotov-Hopfield Learning Rule, proposed in [Krotov and Hopfield, 2019], is a Hebbian-based learning rule that is able to achieve remarkable performance despite the locality constraint. The rule incorporates anti-Hebbian plasticity with global lateral inhibition to guide learning of diverse feature representations in an unsupervised manner. In their original paper, [Krotov and Hopfield, 2019] found that the Krotov-Hopfield rule outperformed several other biologically-plausible learning algorithms on MNIST and achieved a test-error performance comparable to backpropagation despite not being trained end-to-end.



(a) MNIST Digit results



(b) MNIST-Fashion results.

Figure 1: Comparison of validation accuracy as a function of epochs trained, using Krotov-Hopfield learning rule (baseline), Robust Hebbian Learning (proposed method), and backpropagation (supervised learning). Results are shown across the MNIST dataset (left) and MNIST-Fashion dataset (right). Each algorithm was used to train the same base model (described in Section 3.3), which incorporated EarlyStopping, which halts training once weight values stabilize in order to demonstrate convergence time. It should also be noted that in these plots, Robust Hebbian Learning was trained through all 400 epochs without EarlyStopping for completeness.

3.3 Model

Each learning algorithm was used to train the convolutional layer of a single-layer convolutional neural network (CNN) with 400 filters. Once learned, the trained layer is frozen, and the output layer is then trained with backpropagation using cross-entropy loss and the Adam optimizer. The convolutional layer has 5×5 kernels with stride 1 and the pooling layer uses 2×2 kernels with stride 2. We also use a learning rate scheduler that linearly decreases the learning rate parameter until it reaches 0 in the final epoch. Finally, we incorporate an early stopping mechanism to terminate training after 15 epochs of minimal weight updates. This serves to prevent overfitting. All image pre-processing steps, as well as the model training pipeline, are handled by PyTorch-Hebbian, a lightweight and open-source framework designed specifically for implementing Hebbian Learning [Talloen et al., 2021].

3.4 Evaluation Metrics

To assess the performance of these methods, we will use two different measures: validation accuracy and convergence time. Validation accuracy is computed after every epoch using the held-out, 10,000-example validation set described in Section 3.1. This measure is used to evaluate the overall performance of each learning algorithm in their ability to learn useful features.

Convergence time is defined as the number of epochs trained before weight convergence. Weight convergence was tracked using an EarlyStopping handler provided by PyTorch-Ignite with patience of 15 epochs and minimum delta parameter of 0.0001. This means that training would stop after 15 consecutive epochs where change in validation accuracy was less than 0.0001. We took this number of epochs as a measure of convergence time, as our hypothesis with Robust Hebbian Learning is that the robustness to outlying data points would allow for training convergence in fewer epochs.

4 Results

The results of the experiments are shown in Figure 1. Robust Hebbian Learning clearly outperforms Krotov-Hopfield across both datasets, achieving a final validation accuracy of 0.8824 and 0.8325, respectively, on MNIST-Fashion. The final validation accuracies on MNIST digit were 0.969 for Robust Hebbian Learning and 0.9505 for Krotov-Hopfield. Robust Hebbian Learning also converged in fewer epochs, as shown in Table 1, with EarlyStopping invoked after 252 epochs for MNIST and 278 epochs for MNIST-Fashion. Krotov-Hopfield never converged after 400 epochs on either dataset.

Evidently, though, backpropagation still prevailed as the fastest-converging and highest-performing method, ending with a validation accuracy of 0.9804 after just 34 epochs trained on MNIST digit. Similarly, on MNIST-Fashion, backpropagation achieved a final validation accuracy of 0.9029 after 48 epochs.

These performance disparities are not unexpected, given the differences in constraints that each algorithm must operate under. For example, Krotov-Hopfield is a purely unsupervised method which must operate under the locality constraint imposed by Hebb's rule. Robust Hebbian Learning similarly operates under the Hebbian locality constraint, but incorporates error signals from labelled data to guide learning. In contrast, backpropogation not only leverages labelled data, but also does not operate under any constraints of locality

Epochs Trained before EarlyStopping Invoked			
	Robust Hebbian	Krotov-Hopfield	Backpropagation
MNIST	252	400+	34
MNIST-Fashion	278	400+	48

Table 1: Number of epochs that the base model (described in section 3.3) was trained for before EarlyStopping (described in Section 3.4) was invoked.

nor biological plausibility, and instead optimizes loss by propagating the error signal across the entire network and updating using a gradient-based calculation [Bengio et al., 2016].

5 Discussion

This set of experiments shows that overcoming the locality constraint of Hebbian Learning is an extremely challenging open problem that for now remains unsolved. This work demonstrates that incorporating a Bayesian likelihood-based measure of confidence can help in narrowing the performance gap between Hebbian learning algorithms and artificial learning algorithms like backpropagation while maintaining biological plausibility.

5.1 Future Work and Limitations

The findings of this paper raise many promising extensions and questions to be investigated in greater detail. Firstly, the results of the experiment described in Section 4 show that the Robust Hebbian Learning method has promise in bridging the performance gap between biologically-plausible learning algorithms and artifical learning algorithms. However, it is not immediately clear what aspects of the method are responsible for the observed performance gains. Ablation studies should be performed to independently assess the benefits of (1) introducing a Bayesian-likelihood confidence weighting mechanism and (2) incorporating label information to make this a supervised Hebbian approach.

Secondly, more concrete experiments must be performed to evaluate the robustness capabilities of the method. Through our experiment, we demonstrated that Robust Hebbian Learning converges in fewer epochs than Krotov-Hopfield Learning. However, a more precise measure of weight convergence is needed. Several such metrics exist already, including those used in [Bottou, 2010], [Nocedal and Wright, 2006], and [Srivastava et al., 2015]. Additionally, MNIST and MNIST-Fashion are both known to be well-defined datasets which don't require very advanced architectures to be able to achieve extremely high accuracy on. In order to better gauge the robustness of the proposed method, additional experiments should be performed on more complex, noisier datasets.

5.2 Conclusion

Overall, I am very pleased with the outcome of this paper. There is plenty of opportunity for extending this more rigorously in a way that could better represents the underlying human mechanism of confidence weighting, and also concretely bridges the gap between backpropagation and previously-proposed biologically-plausible learning mechanisms. This proposed Robust Hebbian Learning rule provides a more complete model of the human learning mechanism, which is *not* purely unsupervised and *does* incorporate feelings of confidence.

References

L. Aitchison and M. Lengyel. With or without you: predictive coding and bayesian inference in the brain. *Current Opinion in Neurobiology*, 46:219–227, 2017. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2017.08.010. URL https://www.sciencedirect.com/science/article/pii/S0959438817300454. Computational Neuroscience.

F. Alemanno, M. Aquaro, I. Kanter, A. Barra, and E. Agliari. Supervised hebbian learning. *Europhysics Letters*, 141(1):11001, Jan. 2023. ISSN 1286-4854. doi: 10.1209/0295-5075/aca55f. URL http://dx.doi.org/10.1209/0295-5075/aca55f.

Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin. Towards biologically plausible deep learning, 2016. URL https://arxiv.org/abs/1502.04156.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.

T. Bounmy, E. Eger, and F. Meyniel. A characterization of the neural representation of confidence during probabilistic learning. *NeuroImage*, 268:119849, 2023. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2022.119849. URL https://www.sciencedirect.com/science/article/pii/S1053811922009703.

L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

M. Konishi, K. M. Igarashi, and K. Miura. Biologically plausible local synaptic learning rules robustly implement deep supervised learning. *Frontiers in Neuroscience*, 17, 2023. ISSN 1662-453X. doi: 10.3389/fnins.2023.1160899. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1160899.

D. Krotov and J. J. Hopfield. Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences*, 116(16):7723–7731, 2019. doi: 10.1073/pnas.1820458116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1820458116.

F. Meyniel. Brain dynamics for confidence-weighted learning. *PLOS Computational Biology*, 16:1–27, 06 2020. doi: 10.1371/journal.pcbi.1007935. URL https://doi.org/10.1371/journal.pcbi.1007935.

T. Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *eLife*, 6:e20899, feb 2017. ISSN 2050-084X. doi: 10.7554/eLife.20899. URL https://doi.org/10.7554/eLife.20899.

J. R. Movellan. Contrastive hebbian learning in the continuous hopfield model. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Connectionist Models*, pages 10–17. Morgan Kaufmann, 1991. ISBN 978-1-4832-1448-1. doi: https://doi.org/10.1016/B978-1-4832-1448-1.50007-X. URL https://www.sciencedirect.com/science/article/pii/B978148321448150007X.

A. S. Nobandegani, K. da Silva-Castanheira, T. Odonnell, and T. Shultz. On robustness: An undervalued dimension of human rationality. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 41, 2019. URL https://escholarship.org/uc/item/22k276rs.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

Y. Pawitan and Y. Lee. Confidence as Likelihood. *Statistical Science*, 36(4):509 – 517, 2021. doi: 10.1214/20-STS811. URL https://doi.org/10.1214/20-STS811.

T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90044-0. URL https://www.sciencedirect.com/science/article/pii/0893608089900440.

C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, Jan. 2001. ISSN 1559-1662. doi: 10.1145/584091.584093. URL https://doi.org/10.1145/584091.584093.

R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

J. Talloen, J. Dambre, and A. Vandesompele. Pytorch-hebbian: facilitating local learning in a deep learning framework. *CoRR*, abs/2102.00428, 2021. URL https://arxiv.org/abs/2102.00428.

D. Worthy, A. Otto, A. Cornwall, H. Don, and T. Davis. A case of divergent predictions made by delta and decay rule learning models. *CogSci ... Annual Conference of the Cognitive Science Society. Cognitive Science Society (U.S.). Conference*, 2018:1175–1180, 07 2018.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL `https://arxiv.org/abs/1708.07747`.

X. Xie and H. Seung. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15:441–54, 03 2003. doi: 10.1162/089976603762552988.