Riley DeHaan
Dr. McClelland
3/23/19 - PSYCH209

Error-driven Hebbian Learning

Biological learning in neural populations is an area which continues to have significant potential to inform both systems-level learning theory as well as studies of the lower level implementational details of human (and artificial) learning. While considerable understanding of the mechanisms modulating synaptic plasticity has been reached, credit assignment to individual neurons for their contributions to network performance remains an open question. Computational studies of optimization in artificial neural networks suggest that detailed credit assignment, particularly as performed by backpropagation (BP), may be necessary for effective learning in large, complex networks of neural units (Lillicrap et al, 2016). However, the mechanism of BP has generally been considered physiologically implausible due to the requirement that downstream synaptic weight values be explicitly accounted for by upstream feedforward (FF) neurons (Richards et al, 2019). Several solutions to this “weight transport” problem have been proposed, including feedback (FB) networks for transmitting error or reward signals separately from FF propagation as well as the use of relatively isolated distal apical dendritic compartments to process those feedback signals separately from FF inputs (Guerguiev et al, 2017). Additionally, random feedback weights, even with significant degrees of random sparsity, have been found in some cases to enable effective learning, showing that the symmetric feedback weights used in ordinary biological models of BP may not be strictly required for detailed neural credit assignment (Lillicrap et al, 2016).

While these computational considerations of the information transfers necessary for tractable, effective learning in complex neural networks (along with possible underlying biological mechanisms) give support for "hill climbing" local optimization methods, these considerations do not directly attempt to account for the correlative or associative Hebbian learning rules observed extensively in real neurons. It may be that biologically realistic learning rules have so far had comparatively little success in network-level optimization relative to gradient-based methods. However, attempts to understand and develop methods of learning have benefited significantly from study of the human brain, and we believe the empirically observed Hebbian learning rules likely play an important role in human intelligence beyond serving solely as a convoluted biological implementation of BP. It does appear however that some form of gradient-based optimization is likely operating at least locally in the brain, given the computational advantages demonstrated by the technique. We therefore propose an approach to neural learning that melds these two forms of learning and in this work, give an initial exploration of a simple combination of gradient-based and Hebbian learning.

We begin by reviewing Hebbian learning and the specific formulation of it used in our study. Hebbian "fire-together-wire-together" learning may be outlined by the tendency for input synapses correlated with the postsynaptic activity of a given neuron to grow in connection strength, tending to increase the ability of correlated input synapses to stimulate postsynaptic activity in the future. Simple correlative Hebbian learning schemes, such as the Cooper neuron, in which input synaptic weight updates are made solely on the basis of correlation between the weighted input signals and the net postsynaptic activity of the neuron, have been found to lead to synaptic stimulus selectivity in some cases, a significant learning task in biological systems (Bear, 1996). In addition to stimulus selectivity, the general effect of strengthening the connections between related signals is considered a significant factor in the associative memory of human learning (Amari, 1977). However, simple schemes such as the Cooper neuron can lead to runaway (or vanishing) connection weight changes (positive or negative) when input-output correlations causing weight changes in turn cause statistically stronger correlations (Bear, 1996). One common solution to the issue of runaway weights is applying a simple weight decay to all synapses. This technique additionally provides for some degree of input stimulus selectivity in limiting the total synaptic weight strength that can be distributed among the different synapses, leading to competition among different input patterns for weighting. However, weight decay alone has limited ability to lead to input selectivity (McClelland, 2015). The BCM rule, another widely studied Hebbian learning scheme, was designed specifically to account for the formation of orientation selectivity in the visual system and has strong dynamical properties leading to stimulus selectivity (Bienenstock et al). We use this rule in our exploration of combined correlative Hebbian and error-driven learning. The BCM rule adds, in addition to input-output correlation, a second learning factor of a modification threshold for controlling synaptic plasticity of a given neuron. This modification threshold θ_m is in general a nonlinear function of some time integration of the postsynaptic activity y of a given neuron. In our work, we use a common formulation of the modification threshold in which the square of the postsynaptic activity is integrated in time (Mariani, 2017), in our case by an exponential moving average:

$$\theta_{m,\ t+1} = \gamma * \theta_{m,\ t} + (1 - \gamma) * y_t^2$$

with γ giving the exponential decay constant and with the postsynaptic activity y being the synapse-weighted sum of the inputs to the neuron x_j:

$$y = \sum_j w_j x_j$$

Our BCM update rule may then be expressed as

$$\frac{dw_j}{dt} = \alpha \phi(y, \theta) x_j$$

where α gives the BCM learning rate, and ϕ is a nonlinear function of the postsynaptic activity and the modification threshold. We use a standard quadratic form for this function (Mariani, 2017):

$$\phi(y, \theta) = y(y - \theta)$$

giving an overall BCM learning rule of

$$\frac{dw_j}{dt} = \alpha y(y - \theta)x_j$$

The threshold controls synaptic updates such that input synapses active when the postsynaptic activity is above the modification threshold (always positive by our chosen form for the threshold) tend to be strengthened in the direction of the sign of x_j ($y(y - \theta) > 0$) while those active when the postsynaptic activity is positive but below the modification threshold are conversely weakened, tending to cause lower postsynaptic activities for that particular input pattern in the future. For negative postsynaptic activities (potentially corresponding biologically to a firing rate below a standard undriven firing rate for that neuron), the nonlinear factor $y(y - \theta)$ is positive, driving weight changes in the direction of the sign of x_j and therefore pushing future activations for that input pattern in general from a negative postsynaptic activation toward zero. Overall, these dynamics have fixed points with a few input patterns dominating while the synaptic activity $w_j x_j$ for most others tending toward zero (with fixed points of all weights being driven to zero also being possible) (Bienenstock et al). The BCM rule then tends to result in synaptic stimulus selectivity.

In the combination of Hebbian and error-based learning explored here, we make several simplifying assumptions. First, we assume error signals are provided by a feedback network, and for simplicity, we implement these error signals with standard backprop. While standard backprop is broadly considered biologically implausible as discussed above, we note that the work of Lillicrap on random feedback weights (considered far more plausible) demonstrates performance of random feedback comparable to backprop on the simple learning tasks used in this work, which were adopted from (Lillicrap et al, 2016), and therefore using backprop will likely give performance comparable to that which would have been obtained with random feedback weights. The particular implementation for the feedback network in that paper used random feedback weights which were statically set throughout the course of experimentation. We assume however that, although biologically plausible feedback networks likely have asymmetrical feedback weights, these feedback weights also evolve in time with the feedforward network weights to achieve some additional degree of symmetry beyond that shown to be necessary by Lillicrap, and we simplify this to using backprop. The evolution of the

feedback weights over the course of training possibly allows for the Hebbian and error-based learning rules to interact with each other more directly than with static feedback weights, since Hebbian strengthening of particular inputs over others allows for error signals to correspondingly flow back through the network with varied magnitudes, potentially favoring flows of errors through input synapses selected by the feedforward neurons through Hebbian learning. Biologically, we can model the error signals as being input to distal apical dendritic compartments as discussed above, which have been found to have significant and somewhat separate effects on plasticity from those of the basal feedforward dendrites in a wide variety of settings (Richards et al, 2019; Gambino et al, 2014). The BCM component of our learning is linearly separated in this initial analysis from error-driven learning by backprop, with the BCM rule and backprop producing separate weight updates which are interpolated between:

$$\Delta w_{j,total} = \epsilon \Delta w_{j,BP} + (1 - \epsilon)\Delta w_{j,BCM} \ , \ 0 \leq \epsilon \leq 1.$$

In this initial analysis of combinations between Hebbian and gradient-based methods, we explore some simple learning processes involving different interpolation schedules between BP and BCM learning updates (the BP updates are implemented by standard stochastic gradient descent). The first interpolation schedule “BCM initialization” places all learning weight for about the first 10% of training samples on the BCM weight update, followed by all weight being placed on BP for the remainder of training samples. The intuition for this schedule was to examine whether BCM, an unsupervised learning rule in effect generally in neural networks, with or without the presence of error signals, could initialize or “prime” the network for faster BP once an error signal arrives, potentially by generating “error channels” in the feedback weights (which again are modeled as having significant symmetry with the feedforward weights being updated by BCM). The error signal is assumed in this case to dominate plasticity once present. The second interpolation schedule “BCM-GD instant switch” is similar to the first, but has places all learning on the BP and BCM updates in alternating blocks of about 10% of the total training samples. This schedule similarly models the case where a network goes through multiple periods with and without error signals. The third interpolation schedule “cyclical interpolation” shows simultaneous learning from error signals and BCM updates using a sinusoidal interpolation weighting with a sinusoidal period of 10% of the total training samples. A baseline of stochastic gradient descent was also included. Unit biases in all cases were not updated by BCM, but only by SGD.

Our learning tasks as mentioned above were adopted from Lillicrap’s work on random feedback weights. We train our network to approximate a linear target function with 30 input units, 10 output units, and no hidden layers. The target function weights are drawn uniformly from (-1, 1) and fixed for all remaining simulations. Training (2200) and test (1000) samples were generated with the target function from random inputs generated by multivariate Gaussian distribution with zero mean and identity covariance and set for all simulations. All losses shown are normalized squared errors (NSE), which normalize

ordinary mean square error with the loss obtained by using the sample mean as the model (Lillicrap et al, 2016). For this linear task, the BCM initialization schedule started with 200 samples of pure Hebbian learning, followed by pure GD for the remaining 2000 samples. The BCM-GD instant switch schedule alternated between GD and BCM learning phases with 200 sample intervals per phase. The cyclical interpolation schedule had a 200 sample sinusoidal period.

The following settings hold for all the above discussed interpolation schedules with this linear function approximation learning task. The learning networks were 30-20-10 linear feedforward networks with initial weights drawn uniformly from (-0.1, 0.1). The stochastic gradient descent learning rate was manually tuned and set to 0.01. Both the learning networks had bias weights initialized to zero for this task while the target network had no biases. A BCM decay rate γ of 0.8 and a BCM learning rate α of 5.0e-05 were found through manual search with the neural unit selectivity metric developed by (Bienenstock et al):

$$Sel_{d(N)} = 1 - \frac{mean\ response\ of\ unit\ N\ with\ respect\ to\ d}{maximum\ response\ of\ unit\ N\ with\ respect\ to\ d}.$$

This metric approaches 1 for a neural unit N selecting for or responding strongly to only a few input patterns in the input space d and approaches 0 for neurons responding to all input patterns equally. These hyperparameters of the BCM decay and learning rates were manually selected to lead to as rapid evolution as possible in the mean network unit selectivity while maintaining stability in smaller networks as well as the linear learning network. After tuning these hyperparameters with the selectivity metric on smaller networks as well as the linear learning network, we stopped examining the selectivity metric, which would commonly saturate trivially for larger networks as well as result in spiking behavior when involved with significant probability of negative activations, such as those involved with the linear target network, and was found to be less informative for larger networks and larger data sets with larger numbers of samples over which the maximum response would be computed (resulting in saturation due to the statistical spread of the input distribution). Instead, we examined another, better behaved metric of concentration, the Gini coefficient, which was recorded in several ways throughout training on the network activations in response to the train set. First, the Gini coefficient of each individual layer was recorded throughout training for each network. Second, the mean of those coefficients was taken across layers throughout training. Third, the mean Gini coefficients of unit activations were recorded across all train set samples for each neuron and then averaged over neurons to give another measure of neural selectivity and concentration of activation.

Below are given our results for the linear training task described above with the BCM learning rule applied to all layers, hidden and output. All results shown for the linear learning task are averages over 10 trials with new random weight initializations on each

trial. In Fig. 1, we see the layer activation Gini coefficients for the GD baseline and all interpolation schedules. These Gini coefficients overall appear qualitatively similar for all learning methods with respective layers. The Gini coefficients show some spiking early in training and then level out for the output layer 1 with all learning methods, while some increase in Gini coefficient and input selectivity can be seen in the hidden layer, layer 0, for the cyclical interpolation and BCM-GD instant switch schedules, the schedules having persistent Hebbian learning presence.

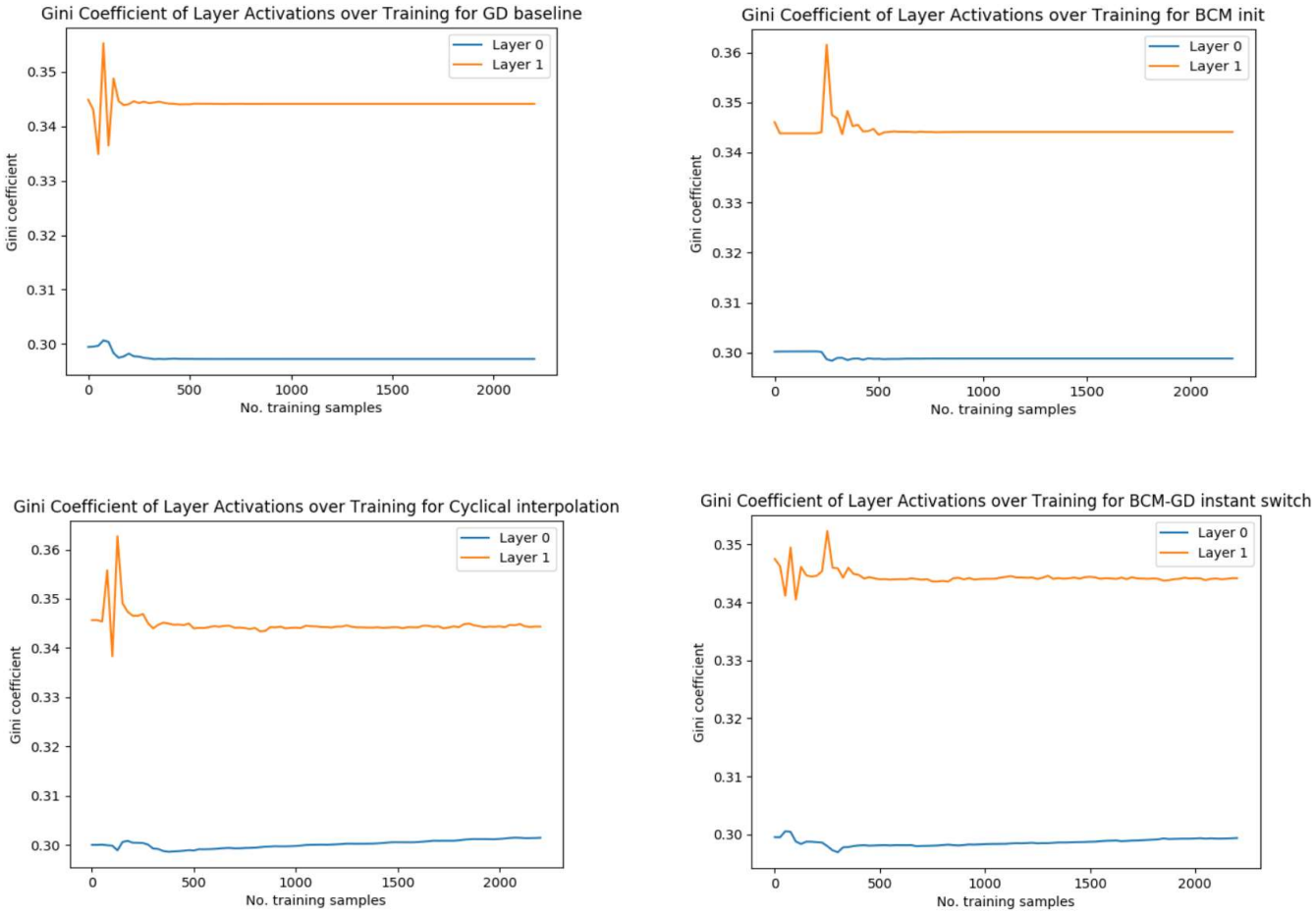

Fig. 1. Layer activation Gini coefficients on training set over training training process for linear target task. GD baseline, BCM initialization, cyclical interpolation, BCM-GD instant switch shown from left to right, top to bottom.

The hidden layers in particular might be expected to develop more selectivity than the output layers because they are exposed to a stationary distribution of inputs, whereas the output layer is learning from a non-stationary distribution of hidden activations, which changes with the hidden layer weights. It is expected the output layer would develop increased selectivity as the hidden layer stabilizes. The mean layer Gini coefficients shown in Fig. 2 below tell a similar story and are included to provide simpler comparison across interpolation schedules. The mean Gini coefficient across unit activations gives

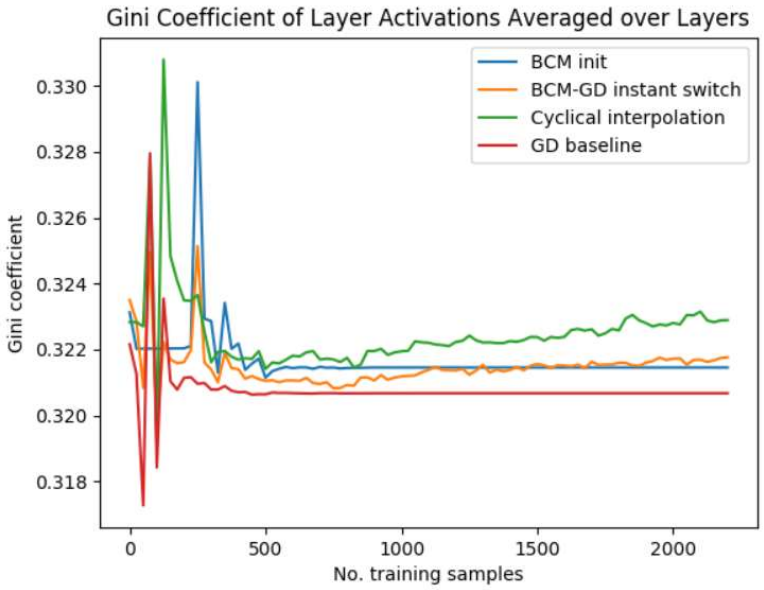

Fig. 2. Average layer Gini coefficients for all Hebbian interpolation schedules applied to all layers and GD baseline for linear target task.

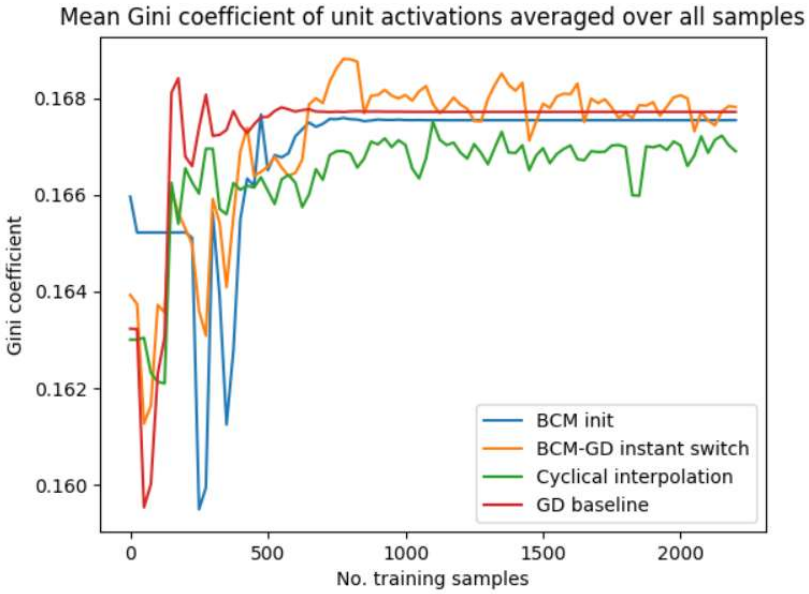

Fig. 3. Mean Gini coefficients across unit activations for all train samples averaged then across all units for each Hebbian interpolation schedule applied to all layers and GD baseline for linear target task.

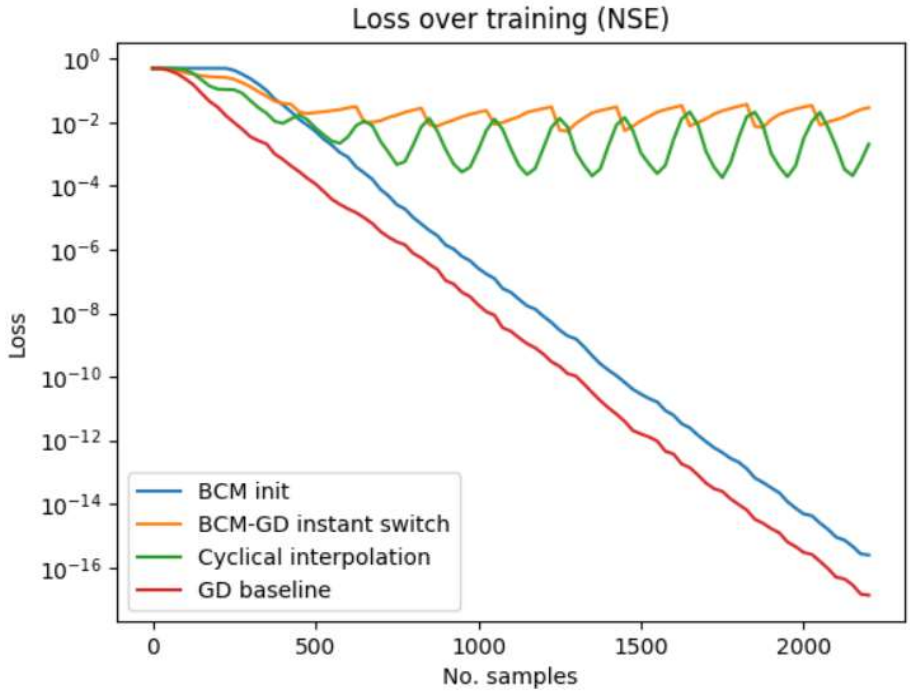

Fig. 4. Normalized square error over test set for the linear target task throughout training for all error-based-Hebbian learning interpolation schedules applied to all layers and the GD baseline.

another perspective and shows Gini coefficients, again after an initial phase of instability where Gini coefficients in general rose significantly, more or less level out later in training with BCM-GD instant switch and cyclical interpolation continuing to oscillate. The differences between layer activation Gini coefficients and Gini coefficients across samples suggest that while the neurons within layers are differentiating from each other, while the neural units overall are not tending to become selective with training, at least beyond early training. Further studies of combinations of Hebbian and error-based learning may need to consider more effective means of stably inducing selectivity, which was difficult in our experience.

Moving on to the learning performance (Fig. 4), we see first of all that BCM-GD instant switch and cyclical interpolation methods quickly converge to oscillating about a much higher error than the methods finishing with pure, unrestricted GD in the GD baseline as well as the BCM initialization interpolation schedule. We might expect this result, given the competing tendencies of these two learning methods, particularly for such a simple learning task for which a local optimization method like GD will excel (as shown by the exponentially vanishing losses with GD and BCM initialization methods). The Hebbian learning methods in general will interfere with the learning completed by GD, resulting in oscillation that settles to a sinusoidal steady state. This behavior may not necessarily be without function. One could imagine Hebbian learning possibly performing a form of regularization which prevents networks from optimizing too tightly to any given solution.

This behavior may not be advantageous for common machine learning tasks with stationary target data sets. However, for non-stationary data sets, we conjecture that a learning rule such as BCM could possibly allow for faster reaction to changing situations in a learning agent's environment by preventing over-investment in one particular situation, while also potentially providing for associative structure in the learned experience. Non-stationary effects such as these are beyond the scope of this work, but may comprise an interesting future direction of research.

Interestingly, while subtle in Fig. 4, there is a slight increase in GD learning speed for the BCM initialization interpolation schedule after the initial pure Hebbian learning period of 200 samples for about 800 samples. Allowing for a hundred samples for both GD methods to "accelerate" and move out of the initial learning phase of small weights (and corresponding small gradients), and accounting for the 200 samples of Hebbian learning during which the network did not improve its error (but which again, from a biological perspective, is unsupervised and therefore "for free" in a sense), we compare the logistic slopes of the learning curves between the BCM initialization interpolation schedule and the GD baseline from 300 samples to 1000 samples, and obtain a logistic slope of -0.344 dB/sample for GD (R^2=0.998) compared to a -0.399 dB/sample (R^2=0.999) logistic slope of loss reduction for BCM initialization over the same period during training. This advantage disappears as training continues, but suggests Hebbian learning may have potential to "prime" a network with Hebbian structure for more standard gradient-based methods. This learning task again may not well-suited for observing this effect, as discussed previously, and problems with more structure, possibly sequential structure, may be more appropriate for future examinations of combinations between Hebbian and gradient-based NN learning.

We similarly analyze the slope of the loss curve of the cyclical interpolation schedule over the periodic intervals of steepest descent once sinusoidal steady state is reached at around 1000 samples. Taking linear regressions of the intervals from 1050 to 1150, 1250 to 1350, 1450 to 1550, 1650 to 1750, 1850 to 1950, 2050 to 2150 samples and then averaging over the slopes from these intervals, we obtain an average slope of -0.923 dB/sample (R^2=0.978), significantly larger than the general slope obtained for the GD baseline (which is more or less constant over the entire training process). These intervals from 1050 to 1150, 1250 to 1350 samples, etc. represent the steady state intervals in each sinusoidal interpolation period during which the GD weighting is greater than the BCM weighting (i.e. the updates lean toward GD rather than BCM). We then again see some evidence of a "priming" effect for this simple task in which a period of mixed but predominantly BCM learning leads to higher rates of error reduction in subsequent periods of mixed but predominantly error-based GD learning. This BCM cyclical interpolation shows sinusoidal steady state behavior with higher average downward slope magnitudes (in the presence of a strong error signal or equivalently when the updates are weighted toward GD), or higher reactivity, than baseline GD learning, which again could be advantageous for non-stationary distributions. We note

that while the instant switch interpolation schedule on the other hand does not exhibit this increased rate of error reduction following the BCM learning periods sustained over the GD periods, it appears this may be in part due to phase shifts between different runs leading to partial cancellation of the effect.

We additionally present loss curves for the linear learning task with the BCM learning rule applied to only the hidden layers (Fig. 5) to allow for the output network to better achieve the learning task with less disturbance from the BCM rule, which tends to work against negative activations in producing pattern selectivity. These loss curves are qualitatively similar to the results with BCM applied to the output layer. The BCM initialization interpolation schedule is essentially unchanged as might be expected given full GD learning is unaffected by the BCM rule. However the curves for BCM-GD instant switch and BCM initialization interpolation schedules have about an order of magnitude and a half lower loss at steady state compared to the results with the BCM rule applied to the output layer. These results might be expected since these schedules involve the BCM rule on an ongoing basis, which disrupts to some degree GD error-based learning while removing the BCM rule from the output layer learning reduces the effect of the

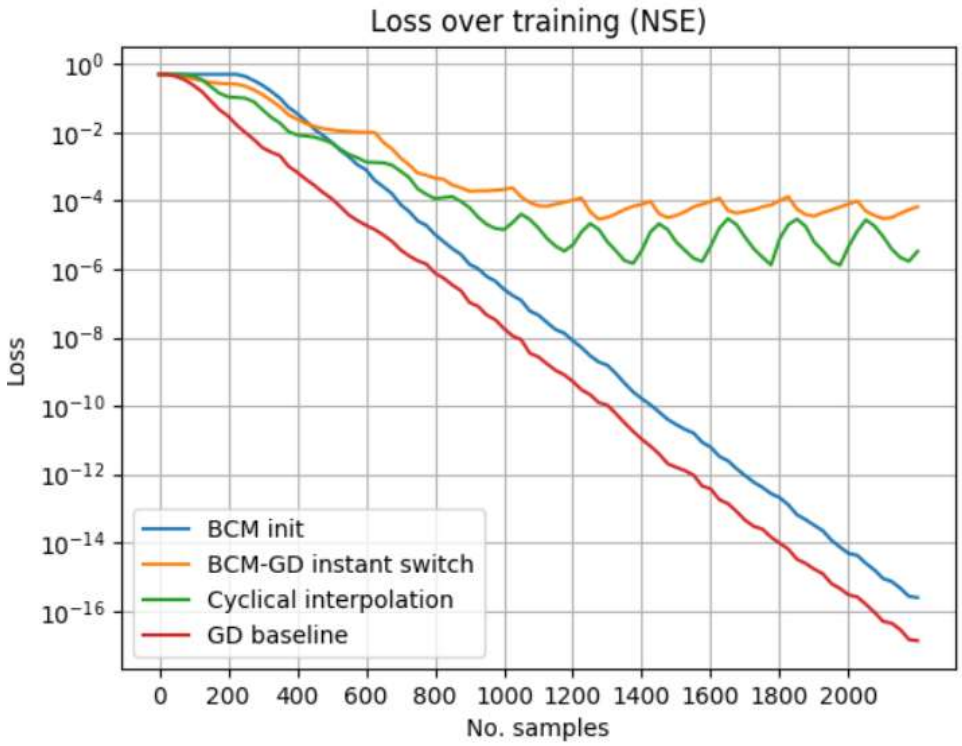

Fig. 5. Normalized square error over test set for the linear target task throughout training for all error-based-Hebbian learning interpolation schedules applied to the hidden layers only and the GD baseline.

BCM rule overall and allows the output layer to produce negative activations more freely, leading to reduced loss. The priming effect can still be seen between the BCM

initialization interpolation schedule and the GD baseline, with nearly identical logistic slopes of the learning curves between the BCM initialization interpolation schedule and the GD baseline from 300 samples to 1000 samples of -0.344 dB/sample for GD (R^2=0.998) compared to -0.399 dB/sample (R^2=0.999) for BCM initialization. We similarly examine the average slope of the loss curve of the cyclical interpolation schedule over the periodic intervals of steepest descent at sinusoidal steady state reached around 1200 samples. Taking linear regressions over the intervals of steepest descent from 1250 to 1350, 1450 to 1550, 1650 to 1750, 1850 to 1950, 2050 to 2150 samples and then averaging over slopes, we obtain an average slope of -0.532 dB/sample (R^2=0.995). These rates of loss reduction (higher than the GD baseline) again support a "priming" effect for this simple task and the cyclical interpolation schedule.

In addition to the linear learning task, we examined the two-hidden layer tanh-tanh-linear function approximation problem also explored in (Lillicrap et al, 2016). These results were averaged over only one trial, and therefore are not particularly convincing, but we present the training loss curves here for comparison in Fig. 6 below. The target function for this task was a 30-20-10-10 tanh-tanh-linear network, with weights and biases uniformly drawn from the range (-0.5, 0.5). A data set of 60000 training samples and of 2000 test samples (a small test set was chosen for speed with the Hebbian updates) was computed again from a zero-mean, identity covariance multivariate Gaussian. The learning networks were also 30-20-10-10 tanh-tanh-linear networks with weights initialized uniformly on (-0.1, 0.1), biases initialized to zero, BCM learning rates of 5.0e-05, BCM decay rate of 0.8, and GD learning rate of 0.05. The BCM hyperparameters were taken directly from the linear network. The GD learning rate was tuned for the problem directly without BCM learning in play. All interpolation schedules were again tested with the BCM learning rule applied to all layers, hidden and output. The BCM initialization schedule started with 5000 samples of pure Hebbian learning, followed by pure GD for the remaining samples. The BCM-GD instant switch schedule alternated between GD and BCM learning phases with 1000 sample intervals per phase. The cyclical interpolation schedule had a 6000 sample sinusoidal period. This learning task represents a significantly more complex task than the linear problem presented earlier. Eventually, both the GD baseline and the BCM initialization methods again win out while the other interpolation methods approach asymptotic oscillation.

We do not see as much of a “priming” effect for this task, in part because the GD baseline reaches convergence early (and the simulation cuts out just as BCM initialization appears about to pass up the GD baseline, although the significance of this would not be clear regardless; we again stress the fact that these results represent a single randomized trial).

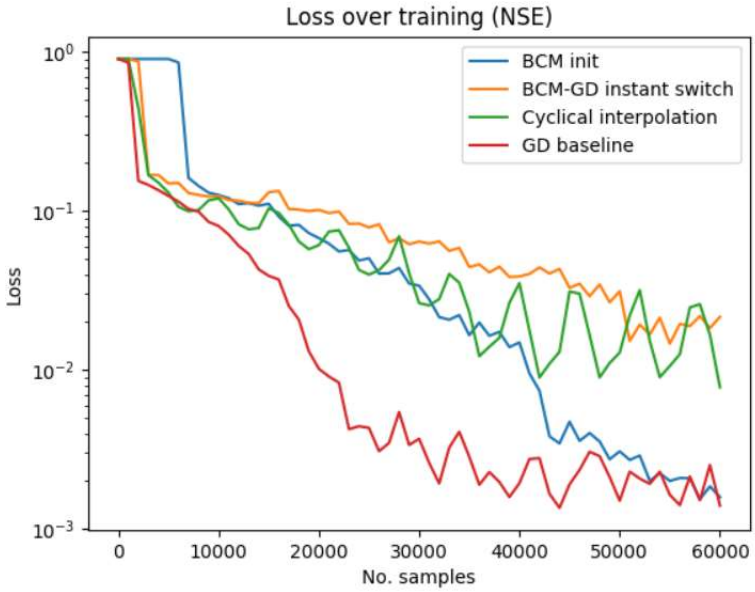

Fig. 6. Learning curves for the nonlinear learning task with a tanh-tanh-linear target function with the BCM learning rule applied to all layers, hidden and output.

In this project, we made an initial exploration of a combination of Hebbian and gradient-based learning methods. The initial results do not provide strong support for advantages to applying Hebbian learning in tandem with more established GD approaches for neural network training, but some support can be seen. The overall idea of using Hebbian learning to provide for unsupervised, neural-level statistical structure in support of learning through local optimization is intriguing. In addition there are many intuitive advantages for the technique, as discussed above, namely the notions of Hebbian regularization, as well as Hebbian correlative associativity providing for structural guidance for gradient propagation. Future work in this area could take many directions, including exploration of Hebbian-error-based combinations with non-stationary distributions having episodic or associative structure, as well as more effective, stable methods of encouraging Hebbian stimulus selectivity. Other task structures that could be well-suited for exploring Hebbian learning include scenarios with unsupervised clustering structure. Another possible idea would be an entropic regularization term between neurons in the same layer, as well as potentially between neurons in different layers, to encourage the learning of diversity in Hebbian stimulus selectivity. Mutual information regularization could also potentially be used to encourage more efficient statistical association between inputs and outputs in Hebbian learning. Overall, while associative learning in the brain has not been seen by many researchers as of late as effective enough to be taken as a primary driver of learning in the brain, to largely dismiss it and focus more or less completely on gradient-based methods would likely be to ignore a critical factor in human learning.

References

Guerguiev J, Lillicrap TP, Richards BA: Towards deep learning with segregated dendrites. eLife 2017, 6:e22901 http://dx.doi. org/10.7554/eLife.22901.

Bear M: A synaptic basis for memory storage in the cerebral cortex. Proc. Natl. Acad. Sci. USA Vol. 93, pp. 13453â€“13459, November 1996 Colloquium Paper.

Gambino F, Pages S, Kehayas V, Baptista D, Tatti R, Carleton A, Holtmaat A: Sensory-evoked LTP driven by dendritic plateau potentials in vivo. Nature 2014, 515:116-119 http://dx.doi.org/ 10.1038/nature13664

Lillicrap TP, Cownden D, Tweed DB, Akerman CJ. Random synaptic feedback weights support error backpropagation for deep learning. Nat. Commun. 7, 13276 doi: 10.1038/ncomms13276 (2016).

Richards BA, Lillicrap TP: Dendritic Solutions to the credit assignment problem. Current Opinion in Neurobiology 2019, 54:28-36. 8 https://doi.org/10.1016/j.conb.2018.08.003.

Bienenstock EL, Cooper, LN, Munro PW: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Journal of Neuroscience 1 January 1982, 2 (1) 32-48; DOI: https://doi.org/10.1523/JNEUROSCI.02-01-00032. (1982).

Amari SI: Neural Theory of Association and Concept-Formation. Biol. Cybernetics 26, 175-185 (1977).

McClelland, JL: Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises. Stanford University, PDP Lab. (2015) https://web.stanford.edu/group/pdplab/pdphandbook/handbook.pdf.

Mariani T: Comparison between Oja's and BCM neural networks models in finding useful projections in high-dimensional spaces. Thesis. Universita di Bologna, Scuola di Scienze Dipartimento di Fisica e Astronomia Corso di Laurea in Fisica. (2017).