

Meta-Learning Biologically Plausible Semi-Supervised Update Rules

Keren Gu, Sam Greydanus, Luke Metz, Niru Maheswaranathan, Jascha Sohl-Dickstein
Google Research, Brain Team
{kerengu, sgrey, lmetz, nirum, jaschasd}@google.com

Abstract

The question of how neurons embedded in a network update their synaptic weights to collectively achieve behavioral goals is a longstanding problem in systems neuroscience. Since Hebb’s hypothesis [10] that cells that fire together strengthen their connections, cellular studies [6] have shed light on potential synaptic mechanisms underlying learning. These mechanisms have directly driven the careful hand design of biologically plausible models of learning and memory in computational neuroscience [1]. However, these hand designed rules have yet to achieve satisfying success training large neural networks, and are dramatically outperformed by biologically implausible approaches such as backprop. We propose an alternative paradigm for designing biologically plausible learning rules: using meta-learning to *learn* a parametric synaptic update rule which is capable of training deep networks. We demonstrate this approach by meta-learning an update rule for semi-supervised tasks, where sparse labels are provided to a deep network but the majority of inputs are unlabeled. The meta-learned plasticity rule integrates bottom-up, top-down, and recurrent inputs to each neuron, and generates weight updates as the product of pre- and post- synaptic neuronal outputs. The way in which the inputs to each neuron are combined to produce a learning signal, however, is itself a *meta-learned* function, parameterized by a neural network. Critically, the meta-learned update rule integrates only neuron-local information when proposing updates—that is, our learning rule is spatially localized to individual neurons. After meta-learning, the resulting synaptic update rule is capable of driving task-relevant learning for semi-supervised tasks. We demonstrate this capability on two simple classification problems. In general, we believe meta-learning to be a powerful approach to finding more effective synaptic plasticity rules, which will motivate new hypotheses for biological neural networks, and new algorithms for artificial neural networks.

1 Introduction

Our approach is to directly optimize a synaptic weight update rule using a meta-objective that maximizes performance on a suite of semi-supervised learning tasks. Additionally, we design our learning rule to be biologically plausible: it is *neuron local*, meaning updates to weights depend only on the corresponding pre- and post- synaptic units; it further does not use weight tying between feedforward and feedback passes during training [16]. These design features enable the update rule to be applied to train models with different depths or widths, and to be applied to new datasets. In contrast to previous work [18], our approach is more biologically plausible, simpler, and tackles semi-supervised rather than unsupervised learning. We show that meta-learned biologically plausible learning rules for semi-supervised learning can outperform supervised learning on 2D toy tasks, and have the ability to generalize to unseen tasks.

2 Related Work

2.1 Learning local learning rules

Understanding how networks of interconnected neurons can learn to solve tasks using only information that is spatially and temporally local to individual neurons is an active area of research in both neuroscience and machine learning [23]. Prior research along these lines have proposed and studied specific local learning rules [3] or approximated existing non-local algorithms (such as backpropagation) with more biologically plausible variants [16, 25, 20].

Our work here follows a different thread, which uses meta-learning to learn a local learning rule. Learning a parameterization of a local synaptic learning rule was originally proposed in the early 1990s [5, 4]. Despite differences in scale (for example, these early learning rules had just 7 or 16 meta-parameters [4]), the overall setup is largely similar: write down a parameterized learning rule that is a function of neuron local variables, specify a set of training task(s), and finally optimize the learning rule for those tasks using either 1st (gradient descent) or 0th order optimization methods (simulated annealing, genetic algorithms). Concurrent work along these lines uses multilayered fully connected networks to parameterize the learning rule [17]. A distinct line of research does not focus on biological plausibility, that is, they use meta-learning for learning non-local learning algorithms [12, 21, 22, 2, 24].

2.2 Semi-supervised learning

Semi-supervised learning (SSL) refers to a class of approaches that use unlabeled data to improve supervised learning. One approach to SSL starts by learning an unsupervised representation using unlabeled data, followed by fine tuning the model on a small labeled dataset [7, 8, 11]. Another common approach to SSL uses unlabeled data during training by adding additional loss terms, for example via generative modeling [13, 14], entropy minimization [9], small perturbations [19], or creating labels for nearby unsupervised examples [15].

3 Methods

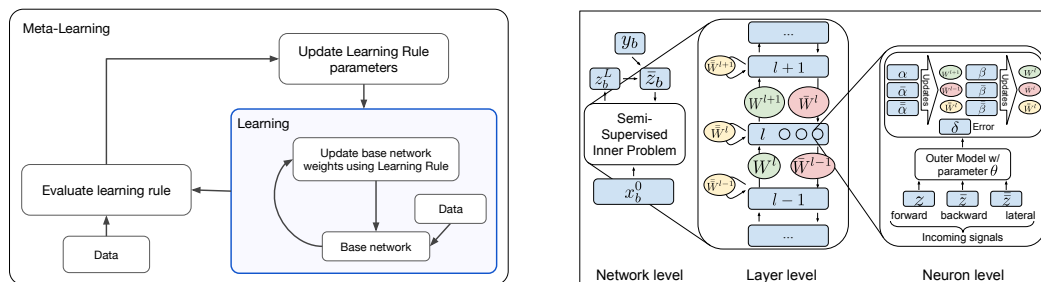


Figure 1: **Left:** Schematic of inner and outer learning loops in meta-training. **Right:** Schematic showing how the learned neuron-local semi-supervised learning rule produces weight updates.

Our meta-training procedure consists of a learning loop (applying the learning rule to update the weights of a network), which itself is wrapped in a meta-learning loop (Fig. 1, Left). The training procedure involves iteratively improving the learning rule in the meta-learning loop, where each iteration involves training and evaluation of the learning rule on a sampled task. In a single iteration of meta-training, we sample data from a semi-supervised learning task (discussed below), initialize a base network with random weights, and perform a fixed number of learning iterations where the learning rule is applied to update the base network weights. The inner model (the model the learned update rule is training) is a 1-hidden layer fully connected neural network in all experiments, with feed-forward weights W . We introduce two additional sets of weights \bar{W} and \tilde{W} in addition to the feed forward weights W (Fig. 1, Right). These two sets of auxiliary weights help propagate error top-down in the network (\bar{W}) and communicate lateral signals between neurons in the same layer (\tilde{W}), and are exclusively used for learning and not prediction. These are loosely inspired by

the numerous feedback and recurrent connections present in cortex. All three sets of weights are initialized at random and updated using the learned update rule.

Instead of computing the gradient of the parameters with respect to the loss (as would be done in backprop), we query the learned update rule for updates to the inner-parameters. Specifically, we compute updates one layer at a time from the top of the network down. For each neuron i , we use the forward activation (z), reverse signal (\bar{z} obtained from \bar{W}), and lateral signal ($\bar{\bar{z}}$ obtained from $\bar{\bar{W}}$) to query the outer model and compute weight updates ΔW , $\Delta \bar{W}$, and $\Delta \bar{\bar{W}}$ for pre- and post- synaptic neurons j and i :

$$\begin{aligned} (\delta_i, \alpha_i, \beta_i, \bar{\alpha}_i, \bar{\beta}_i, \bar{\bar{\alpha}}_i, \bar{\bar{\beta}}_i) &= \text{LearnedPlasticityRule}((z_i, \bar{z}_i, \bar{\bar{z}}_i); \theta), \\ \Delta W_{ij} &= \alpha_i \beta_j, \Delta \bar{W}_{ij} = \bar{\alpha}_i \bar{\beta}_j, \Delta \bar{\bar{W}}_{ij} = \bar{\bar{\alpha}}_i \bar{\bar{\beta}}_j. \end{aligned}$$

The output δ from the outer model is a learned top down error signal, and θ denotes the meta-parameters that parameterize the learning rule. See the Appendix A for a detailed description for how weight updates, and reverse and lateral signals, are computed.

4 Results

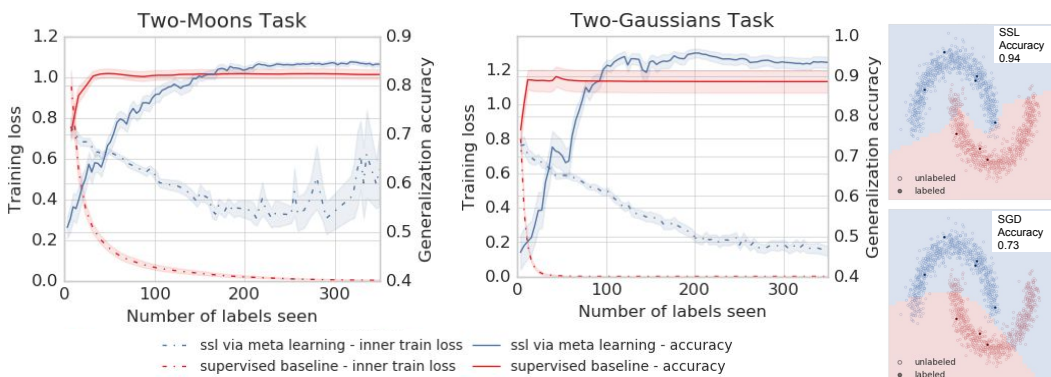


Figure 2: Meta-trained Semi-Supervised update rule outperforms gradient descent on toy two-moons task and generalizes to two-Gaussians. We plot the inner-accuracy and inner-training loss with respect to the number of labeled examples seen for both a supervised baseline (red) and semi-supervised learning with a learned update rule (blue), outer-trained on random two-moons tasks. The total number of labeled examples is 10, while the number of unlabeled examples available during semi-supervised learning is unlimited. Shaded regions represent standard error over 100 trials. Additionally, we show example decision boundaries on two-moons from supervised learning (bottom-right) and semi-supervised learning using a meta-learned update rule (top-right).
†cosyne

To generate our preliminary results, we train a fully connected neural network with one hidden layer. We train and evaluate semi-supervised learning rules on toy classification tasks. We train on tasks formed by re-sampling labeled examples from the two-moons dataset.

For the inner loop, we use a batch size of 10 and train for 50 steps. We compare supervised learning from 10 labeled examples (using backprop) against semi-supervised learning using the same labeled examples and unlimited unlabeled examples using our learned update rule (Fig. 2, Left). We see that the learned update rule has learned to leverage unlabeled examples, achieving higher generalization accuracy. In Figure 2 (Right), the decision boundaries resulting from applying the learned update rule are nonlinear and better aligned to the underlying data distribution.

We further show that the learning rule generalizes from a two moons task to random two-Gaussian tasks that it never experienced during outer-training. On this new semi-supervised learning task, it continues to outperform the supervised baseline. Note that the learned update rule solves the task in a qualitatively different way than backpropagation, and despite doing better on test accuracy performs dramatically worse on training loss.

References

- [1] Larry F Abbott and Sacha B Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3(11s):1178, 2000.
- [2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pp. 3981–3989, 2016.
- [3] Pierre Baldi and Peter Sadowski. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks*, 83:51–74, 2016.
- [4] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pp. 6–8. Univ. of Texas, 1992.
- [5] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, Département d’informatique et de recherche ..., 1990.
- [6] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472, 1998.
- [7] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pp. 3079–3087, 2015.
- [8] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.
- [9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.
- [10] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
- [11] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [12] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pp. 87–94. Springer, 2001.
- [13] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543. ACM, 2008.
- [14] Julia A Lasserre, Christopher M Bishop, and Thomas P Minka. Principled hybrids of generative and discriminative models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 1, pp. 87–94. IEEE, 2006.
- [15] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, pp. 2, 2013.
- [16] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7:13276, 2016.
- [17] Jack Lindsey. Learning to learn with feedback and local plasticity. 2019.
- [18] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Meta-learning update rules for unsupervised representation learning. *International Conference on Learning Representations*, 2019.

- [19] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [20] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *arXiv preprint arXiv:1901.09948*, 2019.
- [21] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.
- [22] Jürgen Schmidhuber. A ‘self-referential’ weight matrix. In *International Conference on Artificial Neural Networks*, pp. 446–450. Springer, 1993.
- [23] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 2019.
- [24] Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3751–3760. JMLR. org, 2017.
- [25] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

A Inner-Training details

Below we provide update equations describing how the inner-model weight updates are produced by the outer model:

$$\begin{aligned}
 x_{bi}^0 &\equiv \text{unit } i \text{ of input example } b \text{ in batch} \\
 y_b &\equiv \text{label for example } b \text{ in batch. Need not be defined for all } b. \\
 N^l &\equiv \text{width of layer } l \\
 W^l &\in \mathcal{R}^{N^{l-1} \times N^l} \equiv \text{input weights for layer } l, \text{ initialize } \sim \mathcal{N}(0, \frac{1}{N^{l-1}}) \\
 \bar{W}^l &\in \mathcal{R}^{N^{l+1} \times N^l} \equiv \text{top down weights for layer } l, \text{ initialize } \sim \mathcal{N}(0, \frac{1}{N^{l+1}}) \\
 \bar{\bar{W}}^l &\in \mathcal{R}^{N^l \times N^l} \equiv \text{lateral weights for layer } l, \text{ initialize } \sim \mathcal{N}(0, \frac{1}{N^l})
 \end{aligned}$$

$$\begin{aligned}
 z_{bi}^l &= \sum_j x_{bj}^{l-1} W_{ji}^l \equiv \text{feedforward pre-activations} \\
 x_{bi}^l &= \phi(z_{bi}^l), \text{ for } l > 0, \text{ and where } \phi \text{ is a nonlinearity such as ReLU} \\
 z^L &\equiv \text{logits at top of } L \text{ layer network} \\
 J &= \sum_{b \in [\text{supervised examples}]} \text{loss}(z_b^L, y_b) \equiv \text{total loss} \\
 \bar{z}_{bi}^L &= \text{gradient of loss w.r.t. logits, } \frac{\partial J}{\partial z_{bi}^L} \\
 \bar{\bar{z}}_{bi}^l &= \sum_j x_{bj}^l \bar{\bar{W}}_{ji}^l \equiv \text{lateral inputs}
 \end{aligned}$$

$$\begin{aligned}
 &\theta \equiv \text{parameters of per-neuron MLP} \\
 &\left(\delta, \alpha, \beta, \bar{\alpha}, \bar{\beta}, \bar{\bar{\alpha}}, \bar{\bar{\beta}}\right)_{\cdot i}^l = \text{MLP} \left((z, \bar{z}, \bar{\bar{z}})_{\cdot i}^l; \theta \right), \text{ where the MLP can accumulate information along the batch dimension} \\
 &\bar{z}_{bi}^l = \sum_j \delta_{bj}^{l+1} \bar{W}_{ji}^l \equiv \text{top-down learning signal for } l < L \\
 &B \equiv [\text{batch size}] \\
 &\Delta W_{ji}^l = \frac{1}{B} \sum_b \frac{\alpha_{bj}^{l-1} \beta_{bi}^l}{LN^{l-1} \sqrt{N^l}}, \Delta \bar{W}_{ji}^l = \frac{1}{B} \sum_b \frac{\bar{\alpha}_{bj}^{l+1} \bar{\beta}_{bi}^l}{L \sqrt{N^{l+1}}}, \Delta \bar{\bar{W}}_{ji}^l = \frac{1}{B} \sum_b \frac{\bar{\bar{\alpha}}_{bj}^l \bar{\bar{\beta}}_{bi}^l}{L \sqrt{N^l}}
 \end{aligned}$$