frontiers

Layer-skipping connections facilitate training of layered networks using equilibrium propagation.

Jimmy Gammell [1,*], **Sae Woo Nam** [1] **and Adam N. McCaughan** [1]

[1] *National Institute of Standards and Technology, Boulder, CO, United States*

Correspondence*:
Jimmy Gammell
jimmy.gammell@colorado.edu

ABSTRACT

Equilibrium propagation is a learning framework that marks a step forward in the search for a biologically-plausible implementation of deep learning, and is appealing for implementation in neuromorphic analog hardware. However, previous implementations on layered networks encountered a vanishing gradient problem that has not yet been solved in a simple, biologically-plausible way. In this paper, we demonstrate that the vanishing gradient problem can be overcome by replacing some of a layered network's connections with random layer-skipping connections. This approach could be conveniently implemented in neuromorphic analog hardware, and is biologically-plausible.

Keywords: equilibrium propagation, deep learning, small-world, layer-skipping connections, neuromorphic computing, biologically-motivated

1 INTRODUCTION

Equilibrium propagation Scellier and Bengio [2016] is a learning framework for energy-based networks such as the continuous Hopfield network [Hopfield, 1984]. It is appealing relative to backpropagation because it is more biologically-plausible, and as a side-effect could be implemented more-easily in neuromorphic analog hardware.

Implementation of equilibrium propagation in [Scellier and Bengio, 2016] was hindered by a vanishing gradient problem whereby networks with as few as 3 hidden layers trained slowly on MNIST [LeCun and Cortes, 1998] - a serious issue given that network depth is critical to performance on difficult datasets [Simonyan and Zisserman, 2014; Srivastava et al., 2015b] and that convergence to a low error rate on MNIST is a low bar to meet. The problem was overcome in [Scellier and Bengio, 2016] by independently tuning a unique learning rate for each layer in the network; however, this approach is unappealing because (1) it introduces additional hyperparameters to tune, (2) it would be inconvenient to implement in neuromorphic analog hardware, and (3) it has not been observed in biological systems.

The purpose of this paper is to demonstrate that in this context the vanishing gradient problem can instead be solved by randomly replacing some of a layered network's connections with layer-skipping connections.

[1] Through this modification we have achieved 0% training error (out of 50,000 examples) and $\lesssim$2.5% test error (out of 10,000 examples) on MNIST using a network with three hidden layers and no regularization term in its cost function. These error rates are comparable to those of other biologically-motivated networks [Bartunov et al., 2018] and are roughly the same as those of the layered network with unique, manually-tuned learning rates in [Scellier and Bengio, 2016]. Our method could be implemented with relative ease in any system with configurable connectivity. Layer-skipping connections have been observed in biological brains [Bullmore and Sporns, 2009], so the approach is biologically-plausible. Similar techniques have seen success in convolutional [He et al., 2015; Srivastava et al., 2015a] and multilayer feedforward [Xiaohu et al., 2011; Krishnan et al., 2019] networks. Our findings outlined in this paper suggest that layer-skipping connections are effective-enough to be appealing in contexts where simplicity and biological plausibility are important.

2 METHODS

2.1 Equilibrium propagation

Similarly to backpropagation, equilibrium propagation [Scellier and Bengio, 2016] trains networks by approximating gradient descent on a cost function. Equilibrium propagation is applicable to any network with dynamics characterized by evolution to a fixed point of an associated energy function; our implementation is a recreation of that in [Scellier and Bengio, 2016], which applies it to a continuous Hopfield network [Hopfield, 1984]. The mathematical formulation of the framework can be found in [Scellier and Bengio, 2016].

A major reason backpropagation is not biologically-plausible is that to implement it, each neuron would need two distinct mechanisms for information transmission: one to transmit its activation to shallower neurons during the forward-propagation phase, and another to transmit error-correction information to deeper neurons during the backward-propagation phase [Bengio et al., 2015]. While this is easy in a digital computer that can oversee and manipulate an entire network, it would be cumbersome in hardware (biological or otherwise) consisting of many simple, independent computational nodes with limited ability to share information. In contrast, equilibrium propagation consists of a free phase (comparable to forward-propagation) and a weakly-clamped phase (comparable to backward-propagation) during which each neuron only needs to know the activations of neighboring neurons, so only one mechanism for information transmission is needed [Scellier and Bengio, 2016]. In a similar vein, to implement backpropagation each neuron would need mechanisms to compute both an activation value using activations of deeper neurons and error-correction information using that of shallower neurons. For equilibrium propagation a neuron would only need the ability to compute an activation given those of adjacent neurons [Scellier and Bengio, 2016].

2.1.1 Implementation in a continuous Hopfield network

2.2 Vanishing gradient problem

2.3 Implementation

2.3.1 Layered topology with per-layer rates

2.3.2 Layered topology with global learning rate

2.3.3 Our topology

[1] This modification was inspired by small-world topology [Watts and Strogatz, 1998]; however, we have not observed a strong correlation between network performance and common metrics of small-worldness (characteristic path length, clustering coefficient, small-world coefficient).

3 RESULTS

3.1 Network performance comparison

3.2 Training rates of individual pairs of layers

3.3 Effect of p

4 DISCUSSION

4.1 Comparing the computational complexity of equilibrium propagation and backpropagation

4.1.1 Comparison

4.2 Related work

4.3 Directions for Future Research

CONFLICT OF INTEREST STATEMENT

REFERENCES

FIGURES

TABLES

Bartunov, S., Santoro, A., Richards, B. A., Hinton, G. E., and Lillicrap, T. P. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *CoRR* abs/1807.04587

Bengio, Y., Lee, D., Bornschein, J., and Lin, Z. (2015). Towards biologically plausible deep learning. *CoRR* abs/1502.04156

Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature*

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR* abs/1512.03385

Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America* 81, 3088–92. doi:10.1073/pnas.81.10.3088

[Dataset] Krishnan, G., Du, X., and Cao, Y. (2019). Structural pruning in deep neural networks: A small-world approach

[Dataset] LeCun, Y. and Cortes, C. (1998). The mnist database of handwritten digits

[Dataset] Scellier, B. and Bengio, Y. (2016). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation

[Dataset] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015a). Highway networks. *CoRR* abs/1505.00387

[Dataset] Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015b). Training very deep networks

Watts, D. and Strogatz, S. (1998). Collective dynamics of 'small-world' networks. *Nature*

Xiaohu, L., Xiaoling, L., Jinhua, Z., Yulin, Z., and Maolin, L. (2011). A new multilayer feedforward small-world neural network with its performances on function approximation. In *2011 IEEE International Conference on Computer Science and Automation Engineering*

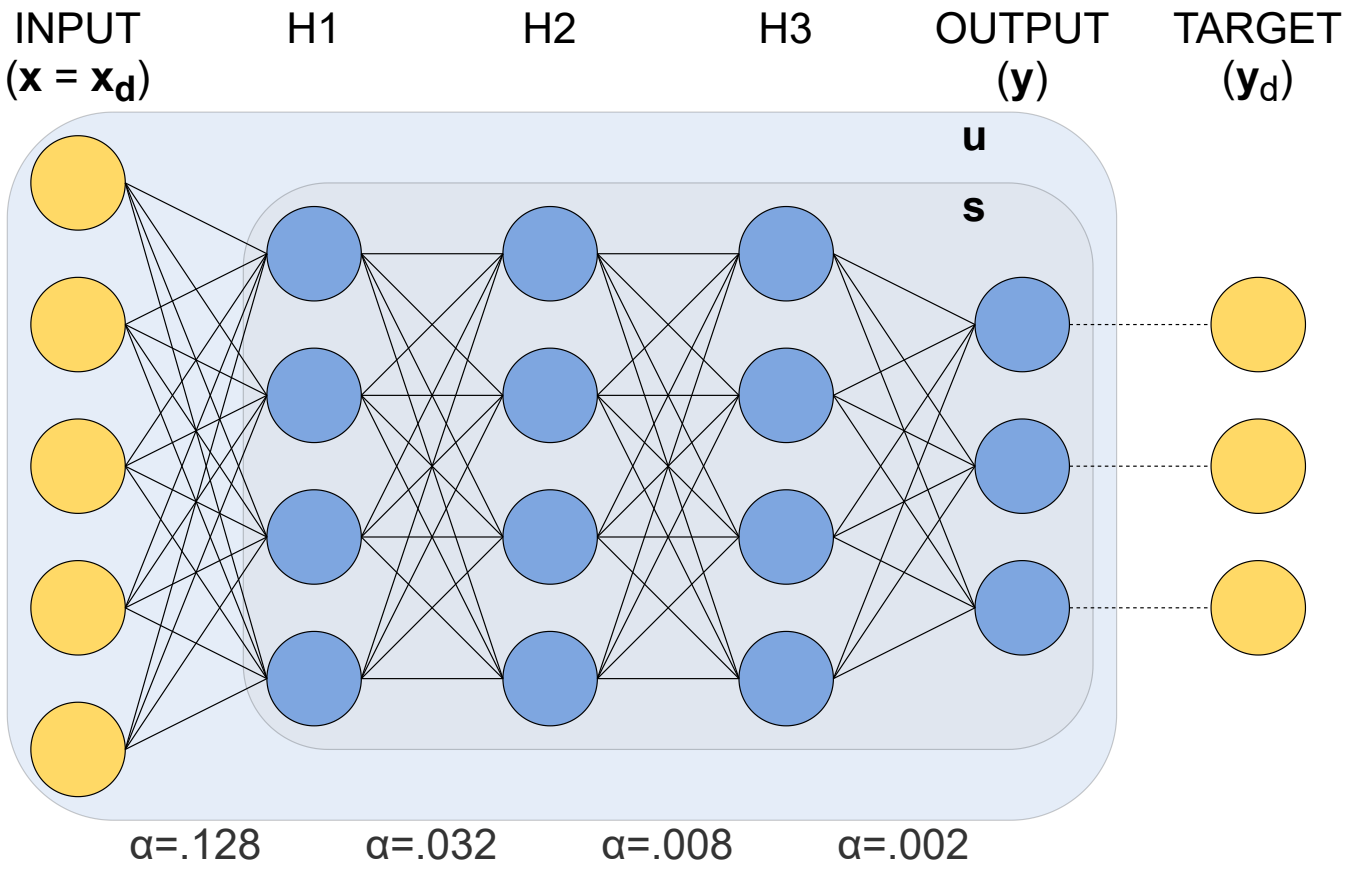

Figure 1. Topology of the layered network tested in [Scellier and Bengio, 2016]. All pairs of neurons in adjacent layers are connected. All connections are bidirectional. To compensate for the vanishing gradient problem, the learning rate is reduced by a factor of 4 each time distance from the output decreases by one layer.

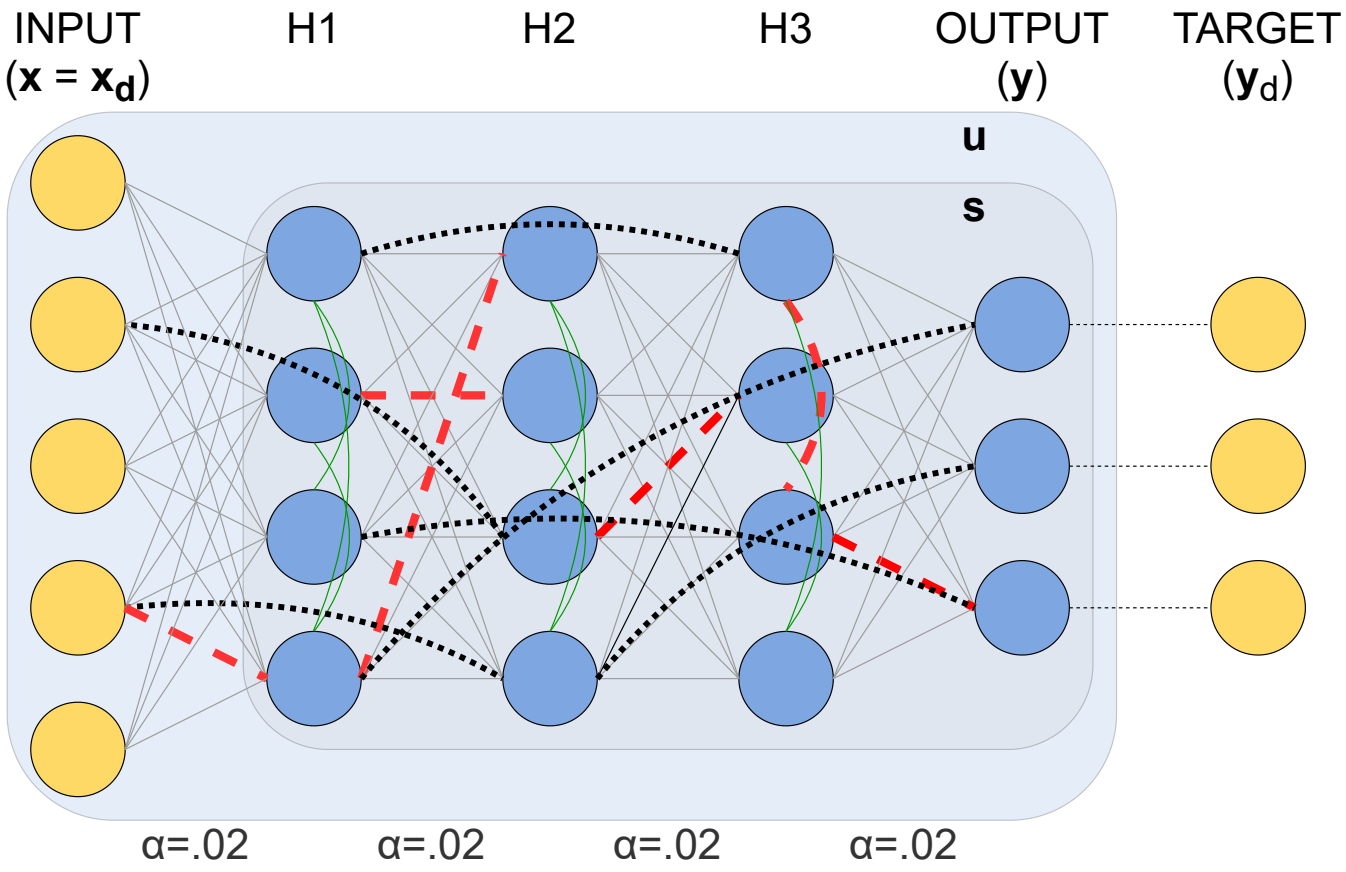

Figure 2. Our modifications to the topology of figure 1 to avoid a vanishing gradient while using a global learning rate. Red dotted lines denote connections that have been removed, black dotted lines denote their replacements, and green solid lines denote added intralayer connections. All connections are bidirectional. This illustration shows a network with $p = 8\%$.

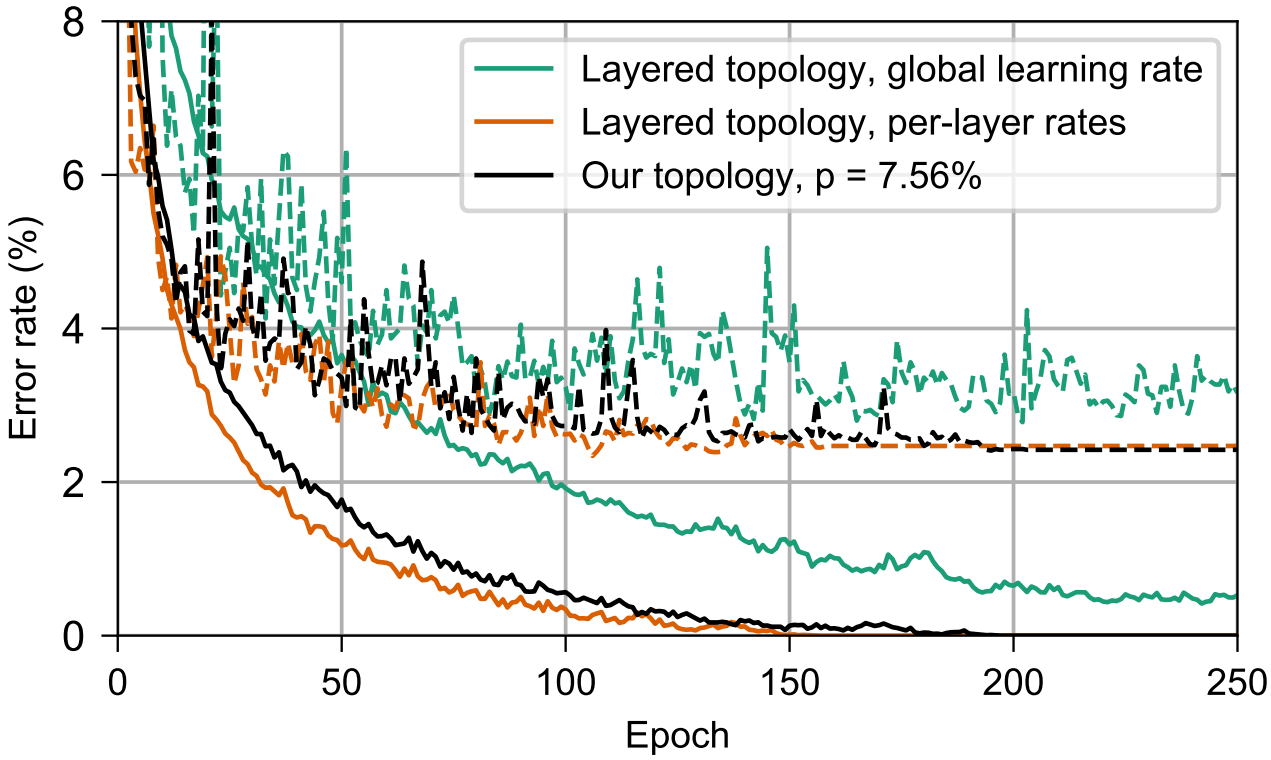

Figure 3. Performance on MNIST of the networks in section 2.3. Dashed lines show the test error and solid lines show the training error. In green is a layered network with a global learning rate (section 2.3.2), in orange is a layered network with per-layer rates individually tuned to counter the vanishing gradient problem (section 2.3.1), and in green is a network with our topology, $p = 7.56\%$ (section 2.3.3). Observe that our topology is almost as effective as per-layer rates at countering the vanishing gradient problem that impedes training of the layered network with a global learning rate.

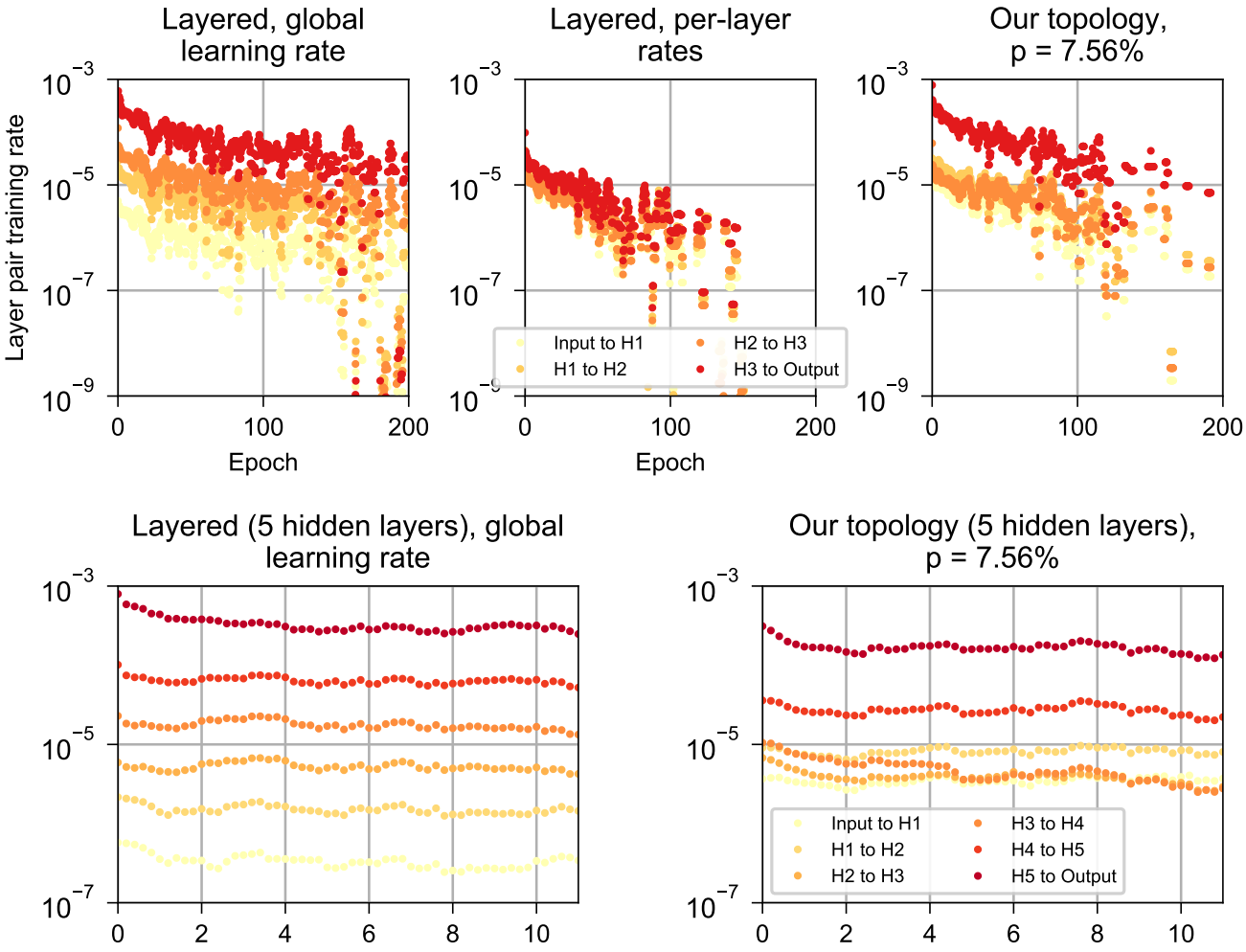

Figure 4. Root-mean-square corrections to weights in different layers while training on MNIST, for the networks in section 2.3. For clarity, values were subjected to an 11-point centered moving average. (top left) A layered network with a single global learning rate (section 2.3.2). (top center) A layered network a unique, individually-tuned learning rate for each layer (section 2.3.1). (top right) A network with our topology, $p = 7.56\%$ (section 2.3.3). (bottom left) A layered network with 5 100-neuron hidden layers and a single global learning rate (section **??**). (bottom right) A network with our topology, $p = 7.56\%$, and 5 100-neuron hidden layers (section **??**). Observe that for the shallower networks the layered topology with a global learning rate has a vanishing gradient problem, which is almost completely solved by tuning an individual learning rate for each layer. Our topology improves the situation by making training uniform among the deeper layers, although the shallowest layer still trains more-quickly than the deeper layers. For the deeper networks, the same trend is apparent but not as strong; we believe this is due to sub-optimal hyperparameter settings.

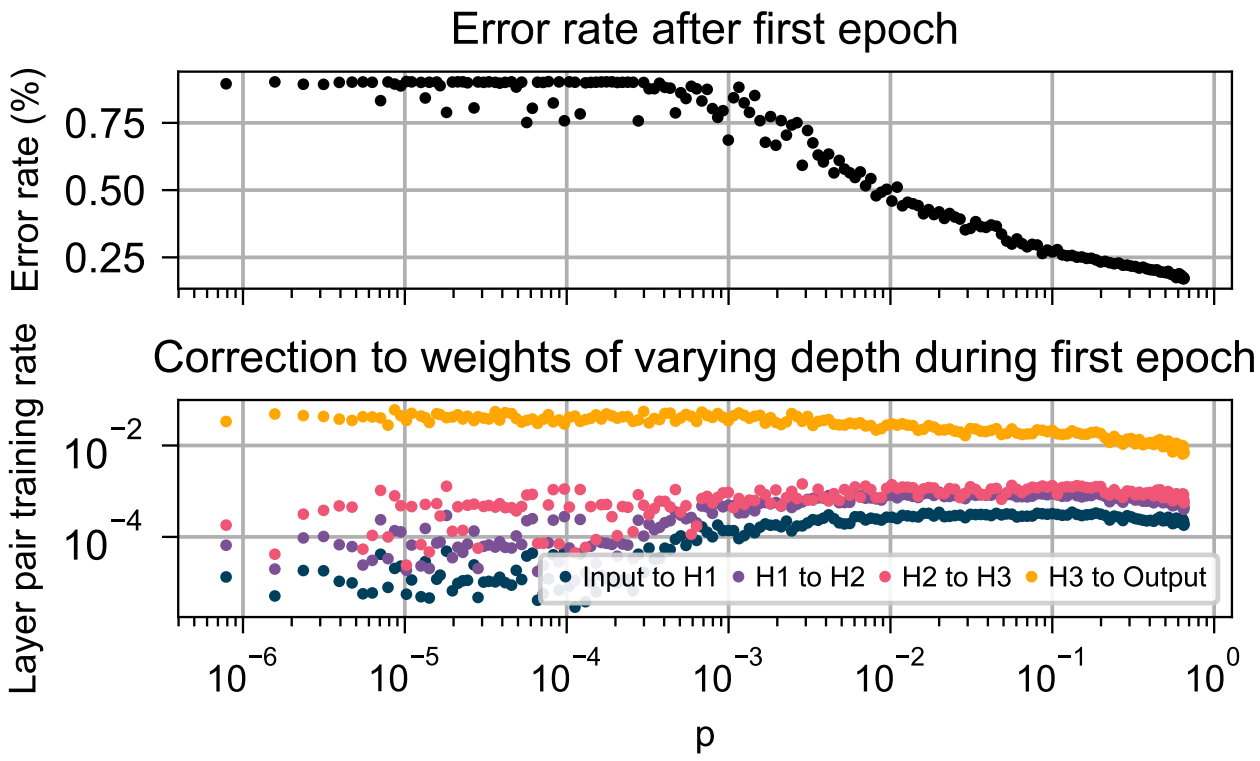

Figure 5. Behavior of our network (section 2.3.3) with varying p, during the first epoch of training. (top) The training error after one epoch. (bottom) Root-mean-square correction to weights in different layers during the first epoch. Observe that as p is increased, the error rate decreases and the root-mean-square corrections to each layer become more-uniform.

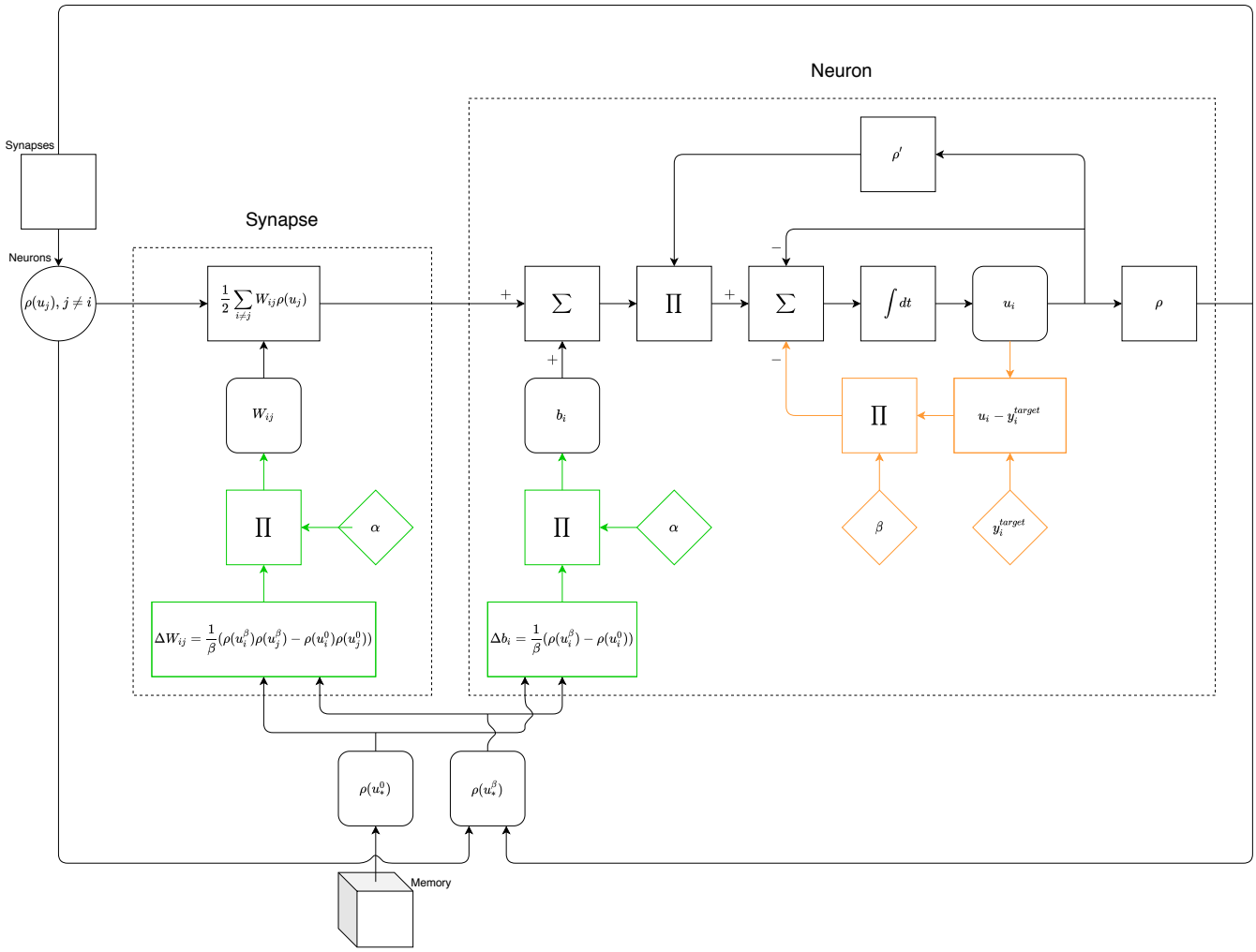

Figure 6. Illustration of the functionality needed to implement equilibrium propagation in hardware. Black lines denote functionality needed in the free phase. Green lines denote functionality to correct parameters. Orange lines denote functionality needed only by output neurons, that is unique to the weakly-clamped phase. There is no functionality unique to the weakly-clamped phase that is needed by all neurons.

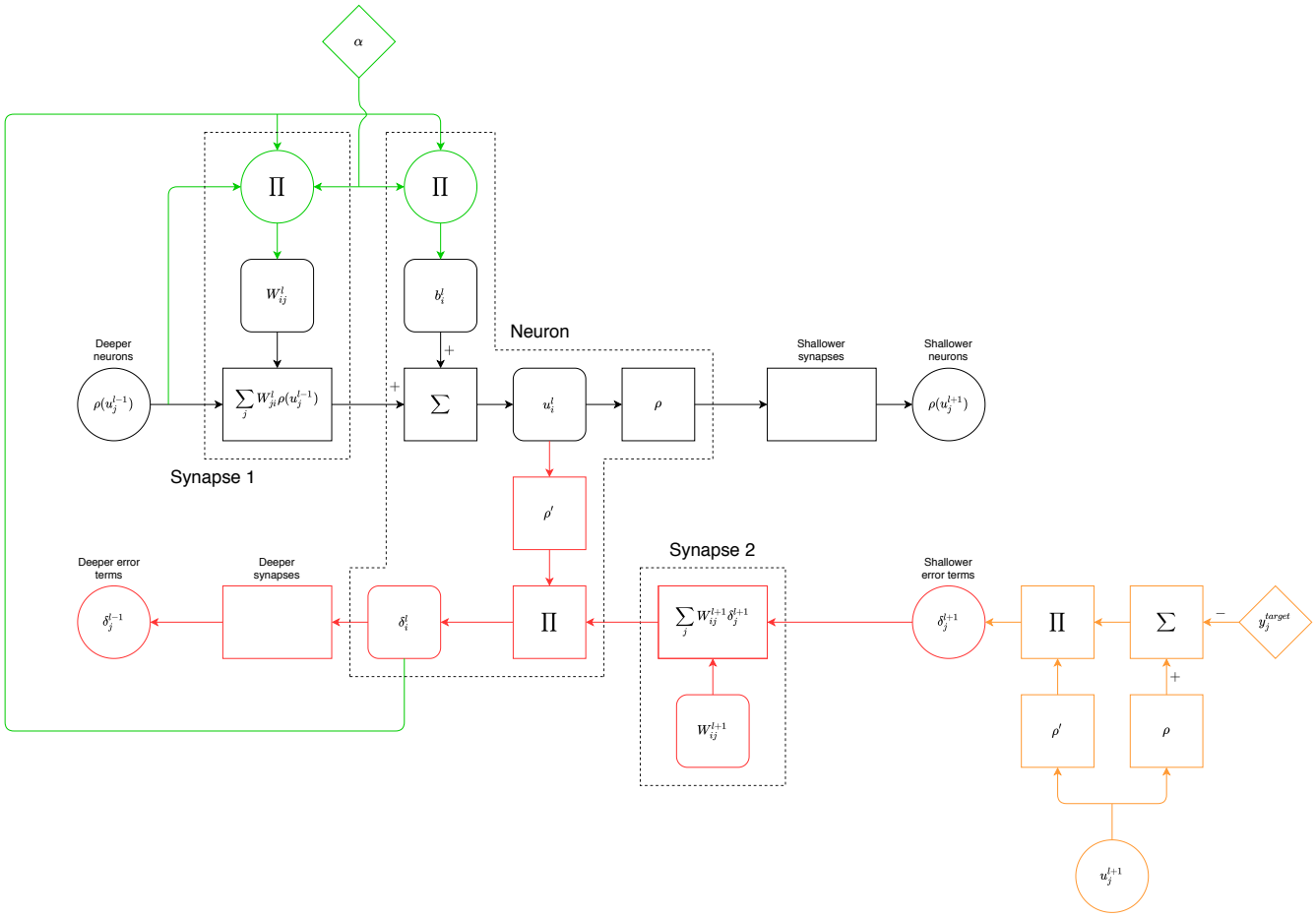

Figure 7. Illustration of the functionality needed to implement backpropagation in hardware. Black lines denote functionality needed in the forwards phase. Green lines denote functionality to correct parameters. Red lines denote functionality unique to the backwards phase that is needed by all neurons. Orange lines denote functionality needed only by output neurons, that is unique to the backwards phase.

	Backpropagation	Equilibrium Propagation
Memory	Space to store activation and error term for each neuron	Space to store free and weakly-clamped activations for each neuron
Nonlinear activation function	Yes	Yes
Derivative of nonlinear activation function	Yes	Yes
Number of distinct computations	2 - computations during forwards and backwards phases are distinct	≈ 1 - hidden neurons perform same computation in both phases. Output neurons perform a similar but modified version of the same computation.
Types of connections	Unidirectional to transmit activation to shallower neighbors and error to deeper neighbors	Bidirectional to each neighbor
Correction computation	Corrections require dedicated circuitry unique from that implementing propagation	Corrections require dedicated circuitry unique from that implementing evolution
Order of computations	Forwards propagation phase where layers are computed from deepest to shallowest; backwards propagation phase where layers are computed from shallowest to deepest; parameter update phase	Free phase where all neurons evolve simultaneously; weakly-clamped phase where all neurons evolve simultaneously; parameter update phase

Table 1.