Biologically realistic supervised deep learning in spiking LIF neurons

Eric Hunsberger, Chris Eliasmith

{ehunsber,celiasmith}@uwaterloo.ca

UNIVERSITY OF WATERLOO

CTN Waterloo

CNRG computational neuroscience research group

compneuro.uwaterloo.ca

Motivation

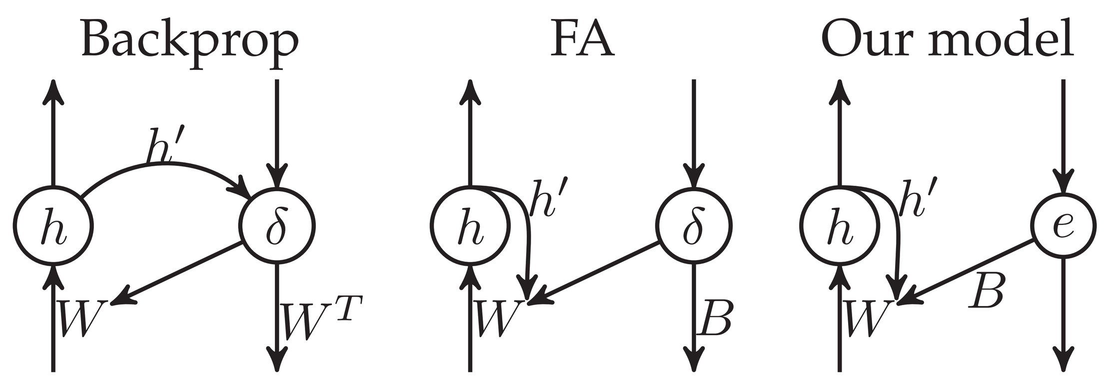
Backprop (BP) is not biologically plausible [1]:

- Error pathway is purely linear
- Error pathway uses present derivatives of forward neurons
- Error pathway uses weights symmetric to those of forward pathway (aka. tied weights)
- Forward and error pathways synchronous
- Treats neurons as non-spiking, differentiable

Feedback Alignment (FA)

A more biologically plausible alternative to backpropagation developed by [2]:

- Uses random feedback weights, instead of symmetric ones
- Forward weights in later layers "align" so that feedback weights carry useful information
- Hidden neurons pushed towards random targets, but only when there is error (c.f. unsupervised learning)
- Hidden neurons pushed in different directions depending on "type" of error (i.e. vector difference between output and target)



Limitations:

- Feedforward neurons are sigmoidal, stochastically spiking
- Feedback neurons are sigmoidal, non-spiking, allow negative firing rates

References

[1] Y. Bengio, D.-h. Lee, J. Bornschein, and Z. Lin, "Towards Biologically Plausible Deep Learning," *arXiv preprint*, vol. 1502, pp. 1–10, 2015.

[2] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random feedback weights support error backpropagation for deep learning," *Nat. communications*, vol. 7, pp. 1–10, 2016.

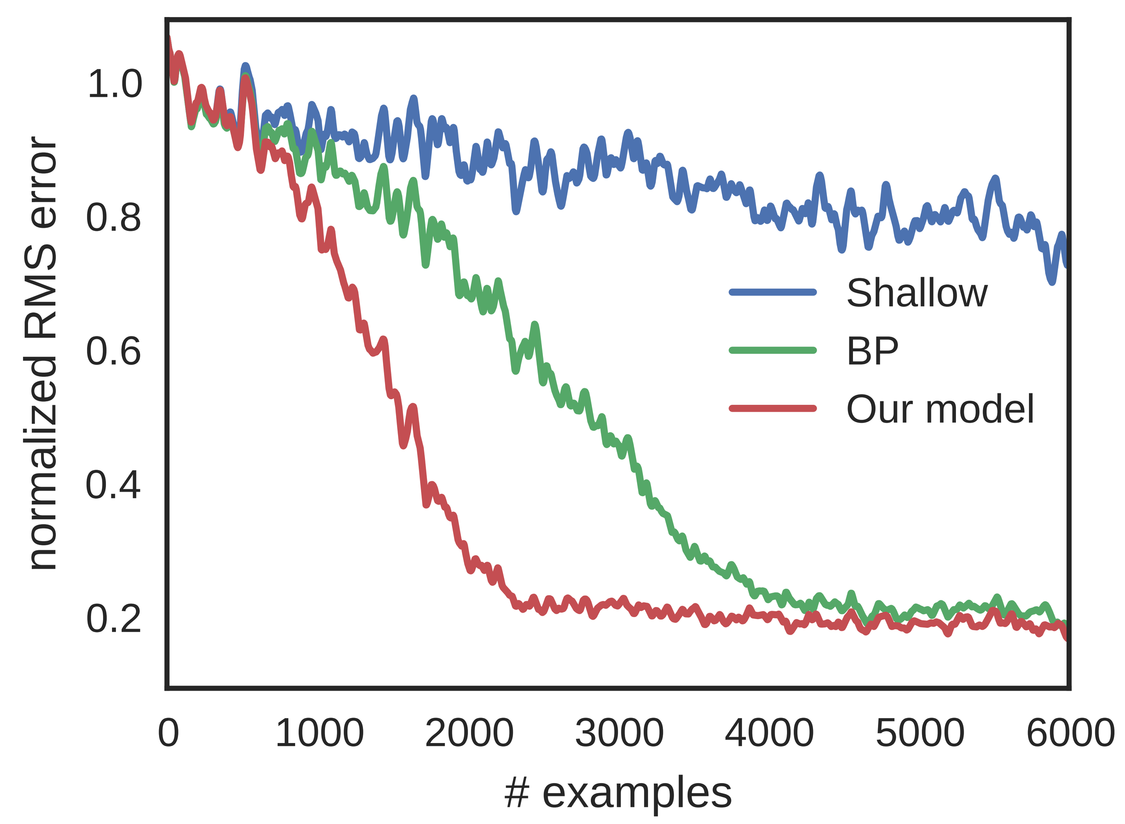
Solution

- Use population coding to transmit final-layer error backwards. This allows the encoding of negative errors.
- Use spiking LIF neurons throughout, with a surrogate derivative for learning

Validation problem:

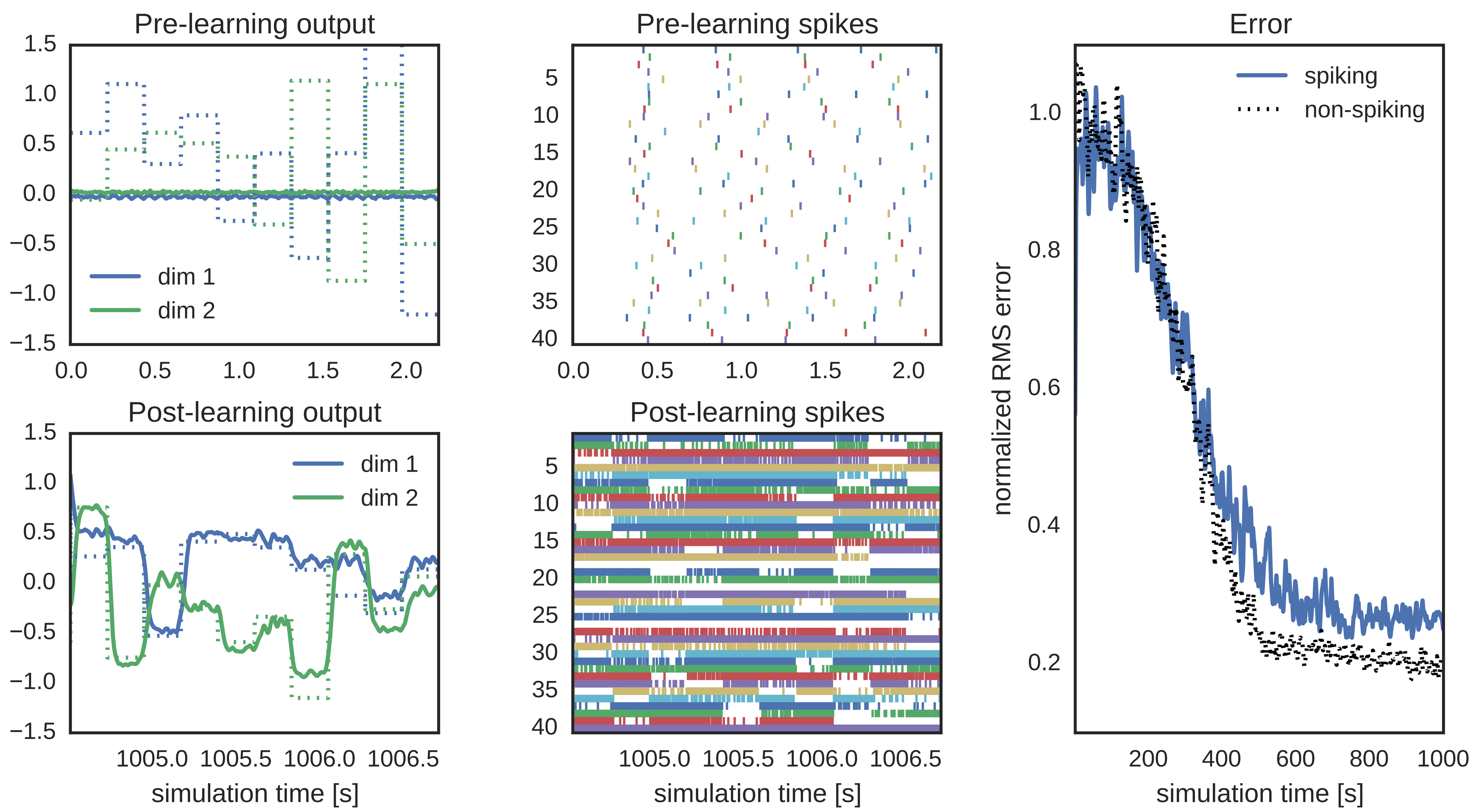
- Data: Linear mapping from 30-D to 10-D, normally distributed. Nontrivial to learn with nonlinear neurons.
- Network: Two hidden layer (30-80-80-10) network. Demonstrates ability to learn deep networks. Spiking network has 3 ms alpha synapses between layers.
- Training: Trained both non-spiking and spiking versions. For spiking network, each stimulus is presented for 220 ms.

Non-spiking Results



Both BP and our model are able to solve the problem using rate-based LIF neurons. Because the initial forward weights are small, our model learns quicker than BP because it has larger feedback weights.

Spiking Results



Here, the fully spiking algorithm learns a deep network. Before learning, the network has no output, and spikes in the second hidden layer (shown) are sparse and independent of the input. After learning, the network produces the target output, and hidden-layer neurons are selective to different inputs. The accuracy is comparable to the non-spiking network.

LIF neuron derivatives

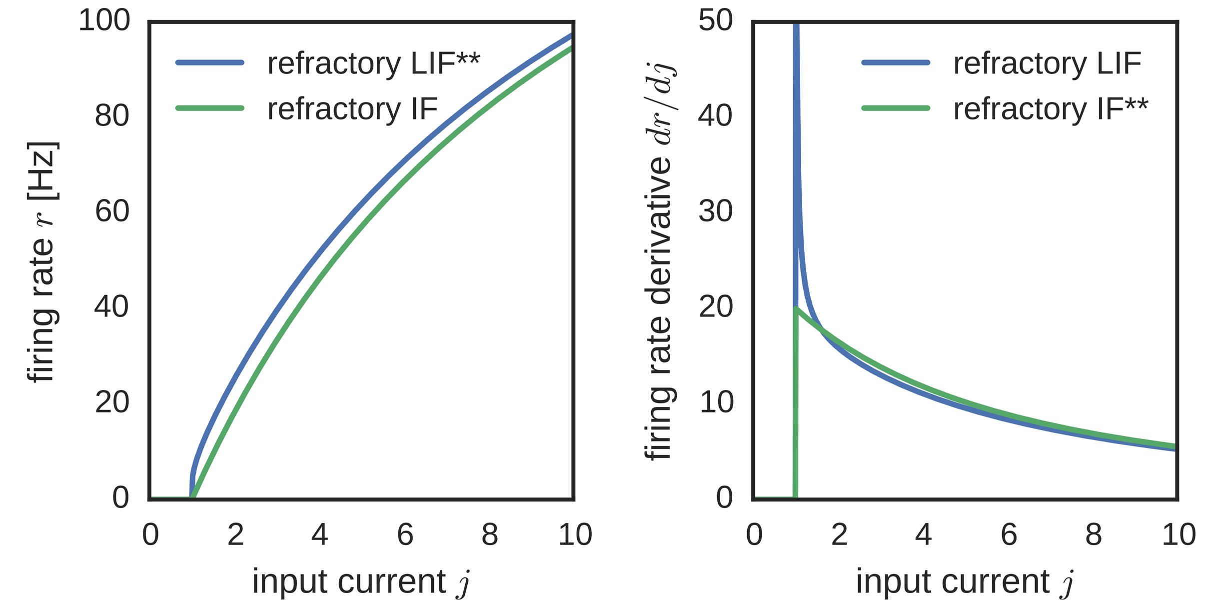
The leaky integrate-and-fire (LIF) neuron dynamics:

$$\tau_{RC}\frac{dV}{dt} = -V + J(t)$$

Spikes when $V > 1$, then $V = 0$ for t_{ref} seconds. The instantaneous firing rate in Hertz is:

$$h(u) = \left(t_{ref} + \tau_{RC}\log\left(1 + \frac{1}{\max(u-1, 0)}\right)\right)^{-1}$$

- Problem: $h' \to \infty$ as $u \to 1^+$
- Solution: Replace h' with surrogate derivative function (derivative of IF neuron with refractory period)
- Derivative no longer matches nonlinearity, but learning still works



** used in model

Discussion

- Fully-spiking FA-based network, using more realistic LIF neurons instead of sigmoid neurons
- Population coding used to transmit negative errors
- Surrogate derivative used for the LIF neurons when learning. Used derivative of refractory IF, but plain IF derivative (i.e., step function) also works.
- Network includes synaptic delays. Each stimulus must be presented long enough for the network to learn from it, but short enough to expose the network to a variety of stimuli.