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

Jimmy Gammell   Sae Woo Nam   Adam N. McCaughan

July 28, 2020

Motivation

- Seek to implement deep learning in neuromorphic analog hardware

Motivation

- Seek to implement deep learning in neuromorphic analog hardware
- Want learning framework usable on simple hardware

Motivation

- Seek to implement deep learning in neuromorphic analog hardware
- Want learning framework usable on simple hardware
  - Neurons and connections perform few distinct tasks

Background: equilibrium propagation

- Equilibrium propagation: a biologically motivated learning framework

Scellier and Bengio, Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation

Background: equilibrium propagation

- Equilibrium propagation: a biologically motivated learning framework
  - Gradient descent on cost function (alternative to backpropagation)

Scellier and Bengio, Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation

Background: equilibrium propagation

- Equilibrium propagation: a biologically motivated learning framework
  - Gradient descent on cost function (alternative to backpropagation)
  - Evolve to minimum of energy function (equilibrium)

Scellier and Bengio, Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation

Background: equilibrium propagation

- Equilibrium propagation: a biologically motivated learning framework
  - Gradient descent on cost function (alternative to backpropagation)
  - Evolve to minimum of energy function (equilibrium)
- Each neuron has state - function of states of neighbors

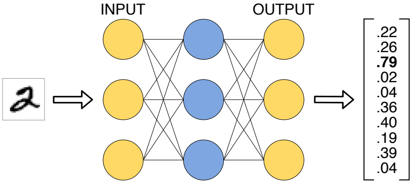
Scellier and Bengio, Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation

Background: equilibrium propagation

- Equilibrium propagation: a biologically motivated learning framework
  - Gradient descent on cost function (alternative to backpropagation)
  - Evolve to minimum of energy function (equilibrium)
- Each neuron has state - function of states of neighbors
  - Like in Hopfield network

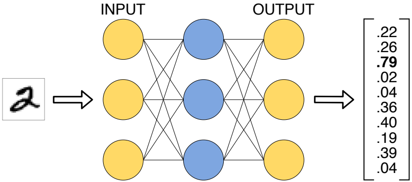
Scellier and Bengio, Equilibrium Propagation: Bridging the Gap Between Energy-Based Models and Backpropagation

Background: equilibrium propagation



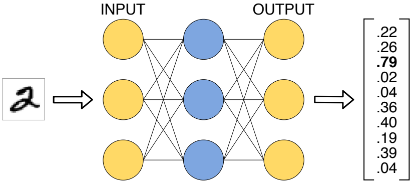
- First phase of training: free phase

Background: equilibrium propagation



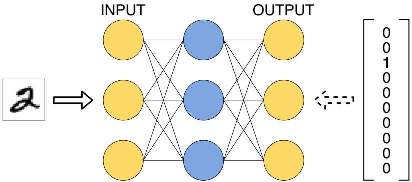
- First phase of training: free phase
  - Evolve to equilibrium for input

Background: equilibrium propagation



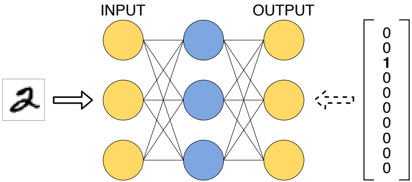
- First phase of training: free phase
  - Evolve to equilibrium for input
  - Prediction: output activations at equilibrium

Background: equilibrium propagation



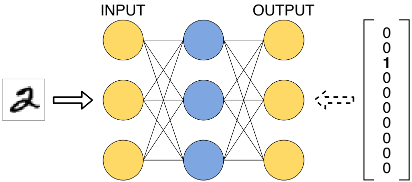
- Second phase of training: weakly-clamped phase

Background: equilibrium propagation



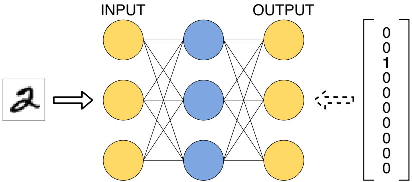
- Second phase of training: weakly-clamped phase
  - Perturb output activations towards target output

Background: equilibrium propagation



- Second phase of training: weakly-clamped phase
  - Perturb output activations towards target output
  - Evolve to equilibrium

Background: equilibrium propagation



- Second phase of training: weakly-clamped phase
  - Perturb output activations towards target output
  - Evolve to equilibrium
- Differences between equilibrium states can be used to compute gradient

Background: equilibrium propagation

- Advantageous due to simplicity of neurons and connections

Background: equilibrium propagation

- Advantageous due to simplicity of neurons and connections
  - Implementable in neuromorphic analog hardware

Background: equilibrium propagation

- Advantageous due to simplicity of neurons and connections
  - Implementable in neuromorphic analog hardware
  - Biologically-plausible (relative to backpropagation)

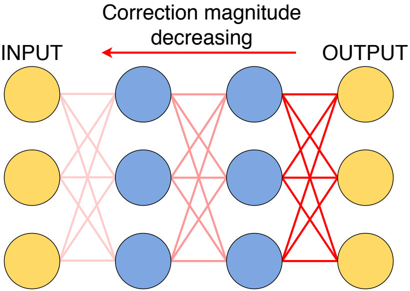
Background: equilibrium propagation

- Advantageous due to simplicity of neurons and connections
  - Implementable in neuromorphic analog hardware
  - Biologically-plausible (relative to backpropagation)
  - Neurons perform same task in both phases of training

Background: equilibrium propagation

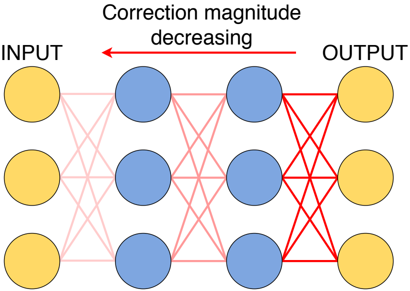
- Advantageous due to simplicity of neurons and connections
  - Implementable in neuromorphic analog hardware
  - Biologically-plausible (relative to backpropagation)
  - Neurons perform same task in both phases of training
  - Same connections usable in both phases of training

Background: vanishing gradient problem



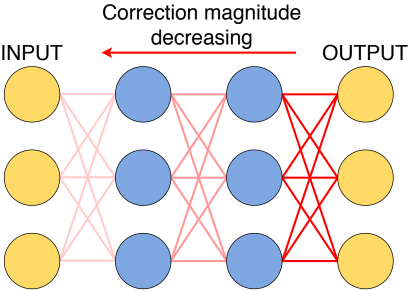
- Problem: vanishing gradients in layered networks

Background: vanishing gradient problem



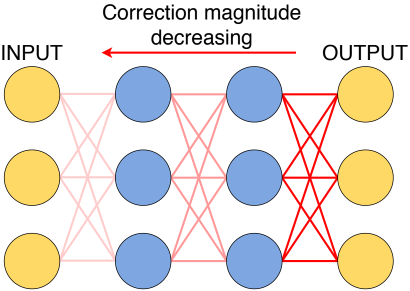
- Problem: vanishing gradients in layered networks
  - Slow training

Background: vanishing gradient problem



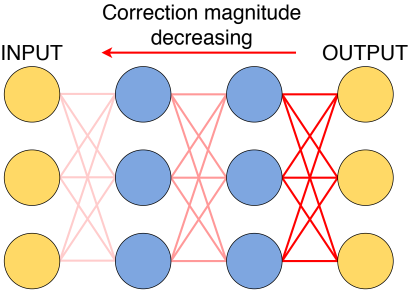
- Problem: vanishing gradients in layered networks
  - Slow training
  - Bit-depth issues

Background: vanishing gradient problem



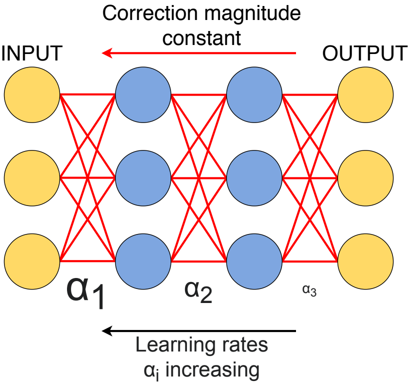
- Problem: vanishing gradients in layered networks
  - Slow training
  - Bit-depth issues
- Need to solve - deep networks better than shallow networks

Background: vanishing gradient problem



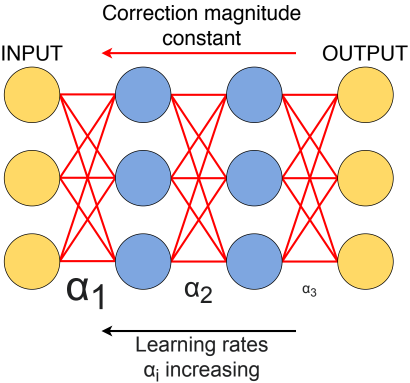
- Problem: vanishing gradients in layered networks
  - Slow training
  - Bit-depth issues
- Need to solve - deep networks better than shallow networks
- Solved for digital computers, but no simple solution for analog implementations

Background: vanishing gradient problem



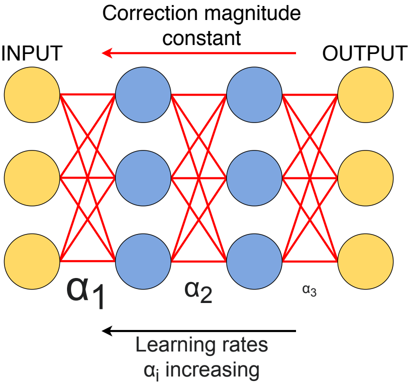
- Original paper: independent learning rates for each layer

Background: vanishing gradient problem



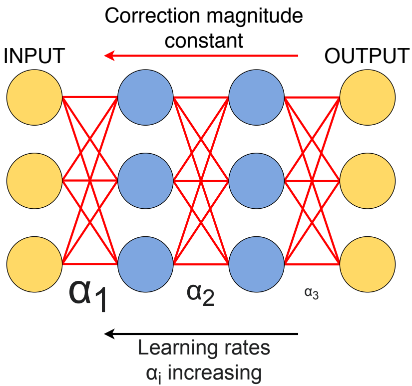
- Original paper: independent learning rates for each layer
  - Increase with depth to compensate

Background: vanishing gradient problem



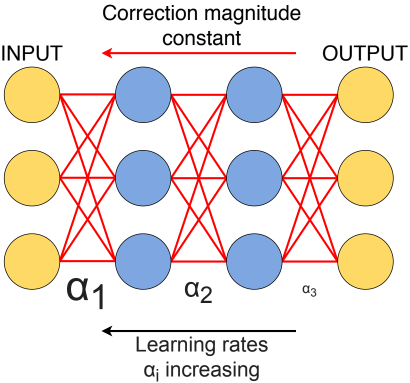
- Original paper: independent learning rates for each layer
  - Increase with depth to compensate
- Unappealing for following reasons:

Background: vanishing gradient problem



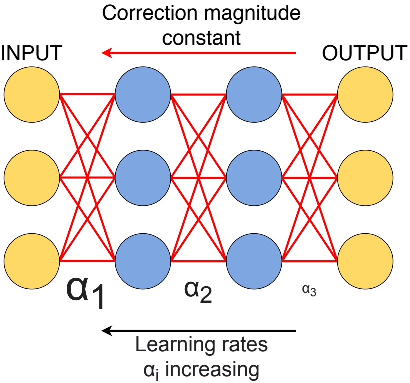
- Original paper: independent learning rates for each layer
  - Increase with depth to compensate
- Unappealing for following reasons:
  1. More hyperparameters to tune

Background: vanishing gradient problem



- Original paper: independent learning rates for each layer
  - Increase with depth to compensate
- Unappealing for following reasons:
  1. More hyperparameters to tune
  2. Inconvenient in neuromorphic hardware

Background: vanishing gradient problem



- Original paper: independent learning rates for each layer
  - Increase with depth to compensate
- Unappealing for following reasons:
  1. More hyperparameters to tune
  2. Inconvenient in neuromorphic hardware
  3. Seems unlikely in biological systems

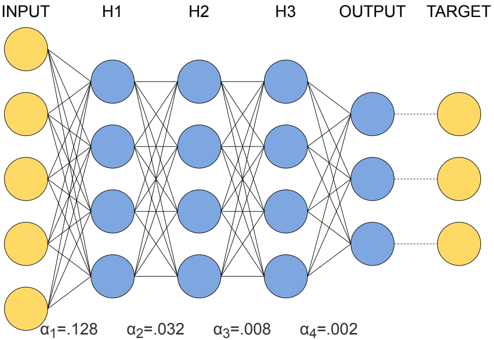
Our solution: layer-skipping connections

- Vanishing gradient problem can be mitigated with layer-skipping connections

Our solution: layer-skipping connections

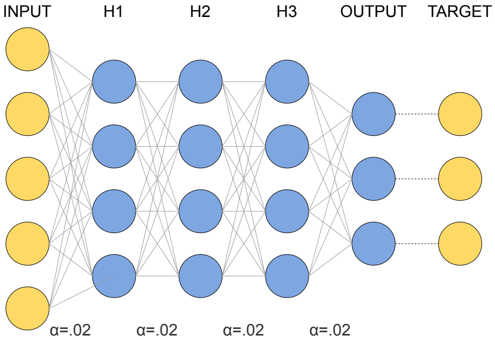
- Vanishing gradient problem can be mitigated with layer-skipping connections
- Topology inspired by small-world networks

Original layered topology



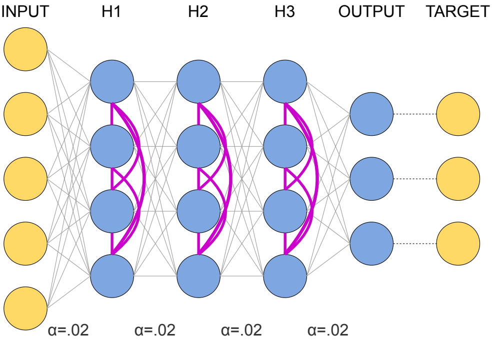
- From original paper
- Per-layer learning rates

Our topological modifications



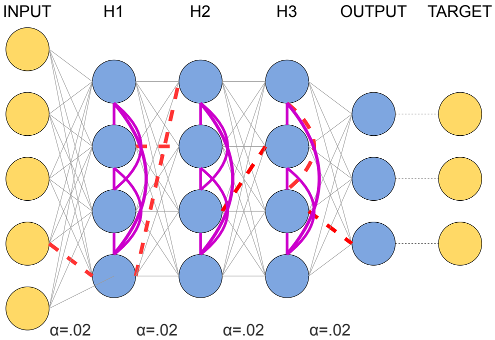
- Starting point: original topology
- One learning rate for all layers

Our topological modifications



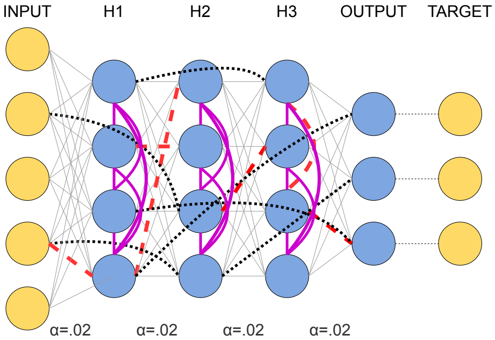
- Hidden layers fully connected

Our topological modifications



- Consider each connection
- Remove with probability p

Our topological modifications



- For each removed connection, randomly connect a different pair
- No connections within input or output layers

Results

- Experiments on MNIST with networks using our topology

Results

- Experiments on MNIST with networks using our topology
  - Fast training

Results

- Experiments on MNIST with networks using our topology
  - Fast training
  - Able to observe effect of variety of changes

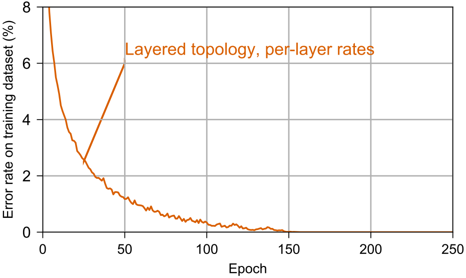
Results

- Experiments on MNIST with networks using our topology
  - Fast training
  - Able to observe effect of variety of changes
- Speeds up training

Results

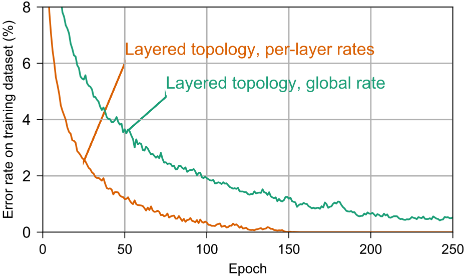
- Experiments on MNIST with networks using our topology
  - Fast training
  - Able to observe effect of variety of changes
- Speeds up training
- Mitigates vanishing gradient problem

Results: training error of layered network with per-layer learning rates



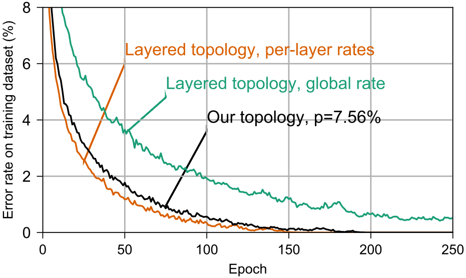
- Network with per-layer rates
- Evaluated on MNIST

Results: training error of layered network with single global learning rate



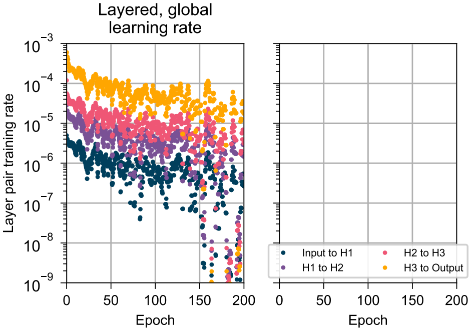
- Network with one global learning rate
- Training slows down

Results: training error of network with our topology



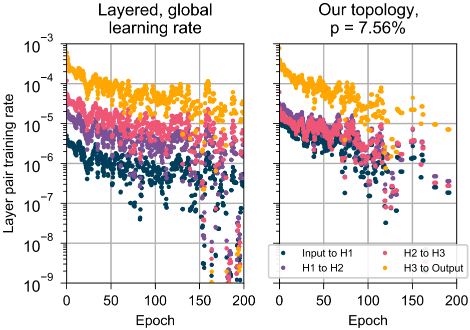
- Network with our topology (still one global learning rate)
- Trains significantly faster than layered network
- Performance similar to original network

Results: vanishing gradient in layered network with single global learning rate



- Vanishing gradient problem when one learning rate is used

Results: vanishing gradient in layered network with our topology



- Our topology mitigates vanishing gradient problem
- Shallowest weights train faster due to lack of layer-skipping connections to target

Conclusions

- Our topology mitigates vanishing gradient problem

Conclusions

- Our topology mitigates vanishing gradient problem
- Avoids issues with per-layer rates

Conclusions

- Our topology mitigates vanishing gradient problem
- Avoids issues with per-layer rates
    1. Only two new hyperparameters; constant with depth

Conclusions

- Our topology mitigates vanishing gradient problem
- Avoids issues with per-layer rates
    1. Only two new hyperparameters; constant with depth
    2. Small-world networks have been observed in biological brains

Conclusions

- Our topology mitigates vanishing gradient problem
- Avoids issues with per-layer rates
  1. Only two new hyperparameters; constant with depth
  2. Small-world networks have been observed in biological brains
  3. Easy to implement in networks with configurable connectivity

Conclusions

- Our topology mitigates vanishing gradient problem
- Avoids issues with per-layer rates
  1. Only two new hyperparameters; constant with depth
  2. Small-world networks have been observed in biological brains
  3. Easy to implement in networks with configurable connectivity
- Good solution where simplicity, biological plausibility important

Directions for future research

- Try on harder datasets (e.g. CIFAR, ImageNet) where depth is very important

Directions for future research

- Try on harder datasets (e.g. CIFAR, ImageNet) where depth is very important
- Effect of p on network performance

Directions for future research

- Try on harder datasets (e.g. CIFAR, ImageNet) where depth is very important
- Effect of p on network performance
- Effectiveness on deeper networks

Directions for future research

- Try on harder datasets (e.g. CIFAR, ImageNet) where depth is very important
- Effect of p on network performance
- Effectiveness on deeper networks
- Try training a network with added layer-skipping connections, then removing them afterwards

Acknowledgments

- Adam N. McCaughan
- Sae Woo Nam
- Sonia Buckley
- Alex Tait
- Zach Grey

Thanks!

- Questions? Email me at `jimmy.i.gammell@gmail.com`