THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

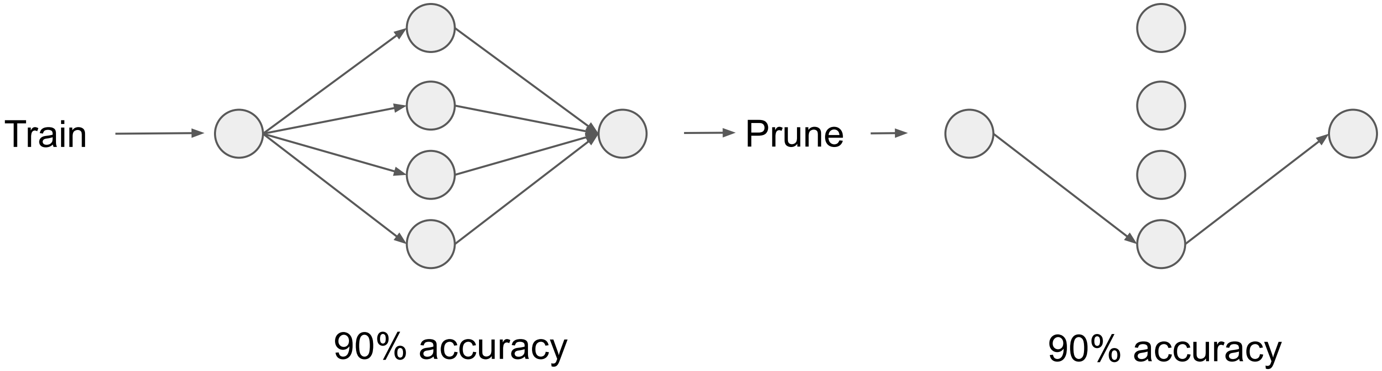
Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Slides prepared for reading club by Nolan Dey

Motivation

- Pruning techniques can reduce parameter counts by 90% without harming accuracy



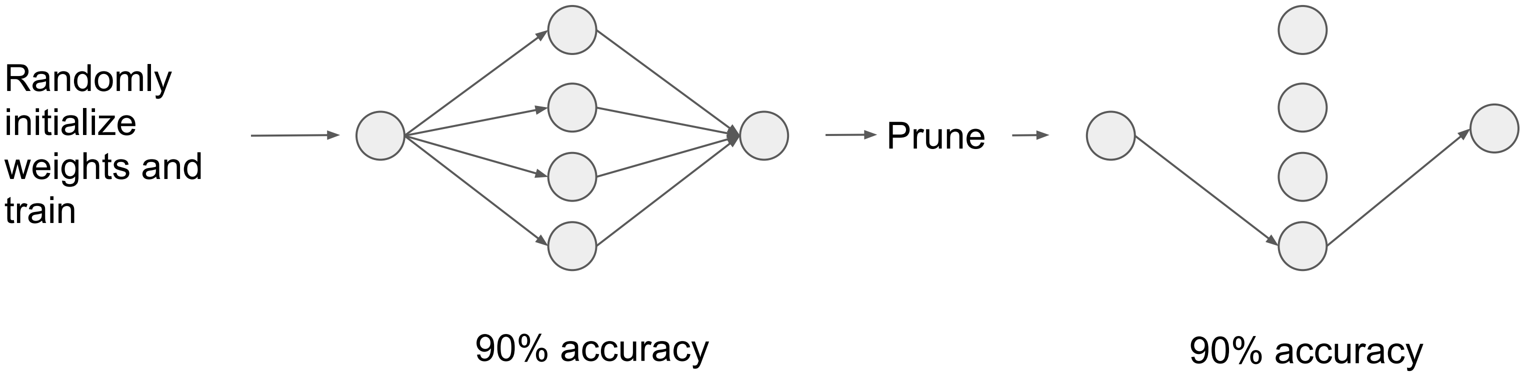
Train → → Prune →

90% accuracy

90% accuracy

Motivation

Pruning techniques can reduce parameter counts by 90% without harming accuracy



Randomly initialize weights and train

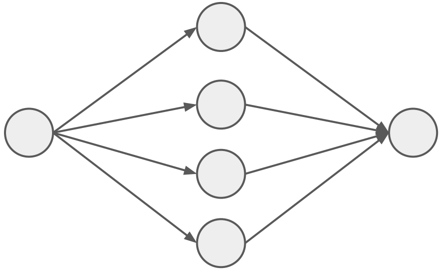
Prune

90% accuracy

90% accuracy

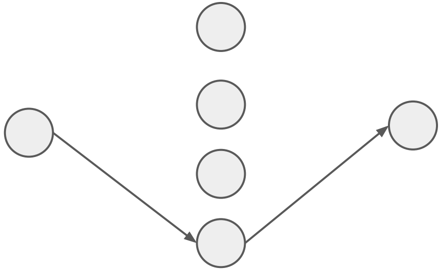
Motivation

Randomly initialize weights and train →



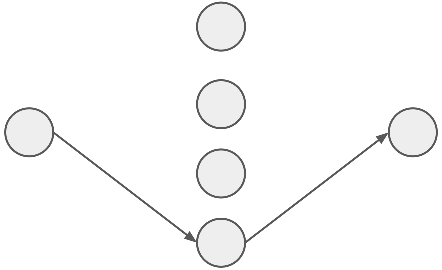
90% accuracy

→ Prune →



90% accuracy

Randomly initialize weights and train →



60% accuracy

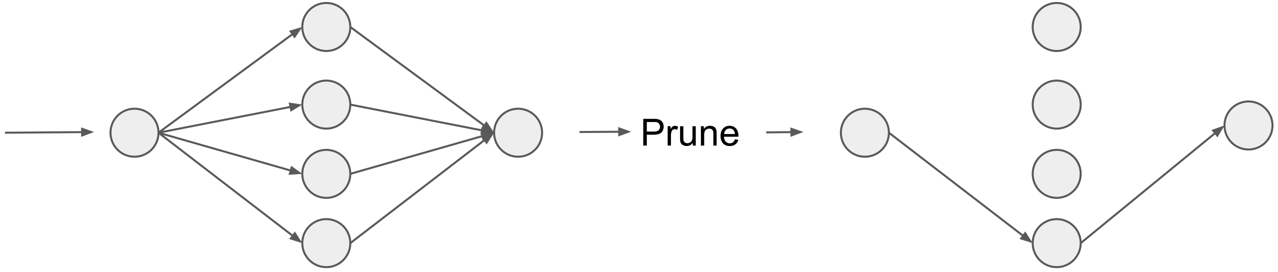
→ 😟

The Lottery Ticket Hypothesis

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

The Lottery Ticket Hypothesis

Randomly initialize weights and train

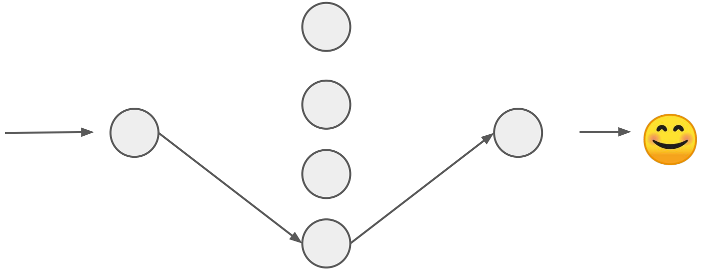


Prune

90% accuracy

90% accuracy

Use same weight initialization and train



😊

90% accuracy

Lottery Analogy

- If you want to win the lottery, just buy a lot of tickets and some will likely win
- Buying a lot of tickets = having an overparameterized neural network for your task
- Winning the lottery = training a network with high accuracy
- Winning ticket = pruned subnetwork which achieves high accuracy

Identifying Winning Tickets

One-shot pruning

1. Randomly initialize a neural network
2. Train the network
3. Prune p%** of weights with lowest magnitude from each layer (set them to 0)
4. Reset pruned network parameters to the original random initialization

Iterative pruning

- Iteratively repeat the one-shot pruning process
- Yields smaller networks than one-shot pruning

**Connections to outputs are pruned at 50% of the pruning rate

Results

- Tested with fully connected, convolutional, and ResNet on MNIST and CIFAR-10
- Pruned subnetworks are 10-20% smaller than the original and meet or exceed original test accuracy in at most the same number of iterations
- Works with different optimizers (SGD, momentum, Adam), dropout, weight decay, batchnorm, residual connections
- Sensitive to learning rate: requires a number of “warmup” iterations to find winning tickets at higher learning rates

Discussion

- Are winning initializations already close to fully-trained values?
    - No! They actually change more during training than the other parameters
    - Perhaps winning initializations might land in a region of the loss landscape that is particularly amenable to optimization
- They conjecture that SGD seeks out a trains a winning ticket in an overparameterized network
- Pruned subnetworks generalize better (smaller difference between train and test accuracies)

Limitations

- Iterative pruning is computationally intensive -> involves training a network 15 times per trial
    - Hard to study larger datasets like ImageNet
    - Future work: find more efficient methods of finding winning tickets
- Their winning tickets are not optimized for modern libraries or hardware
    - Future work: maybe non-magnitude based pruning methods could find smaller winning tickets earlier

Let’s Discuss