How to prune the network ?

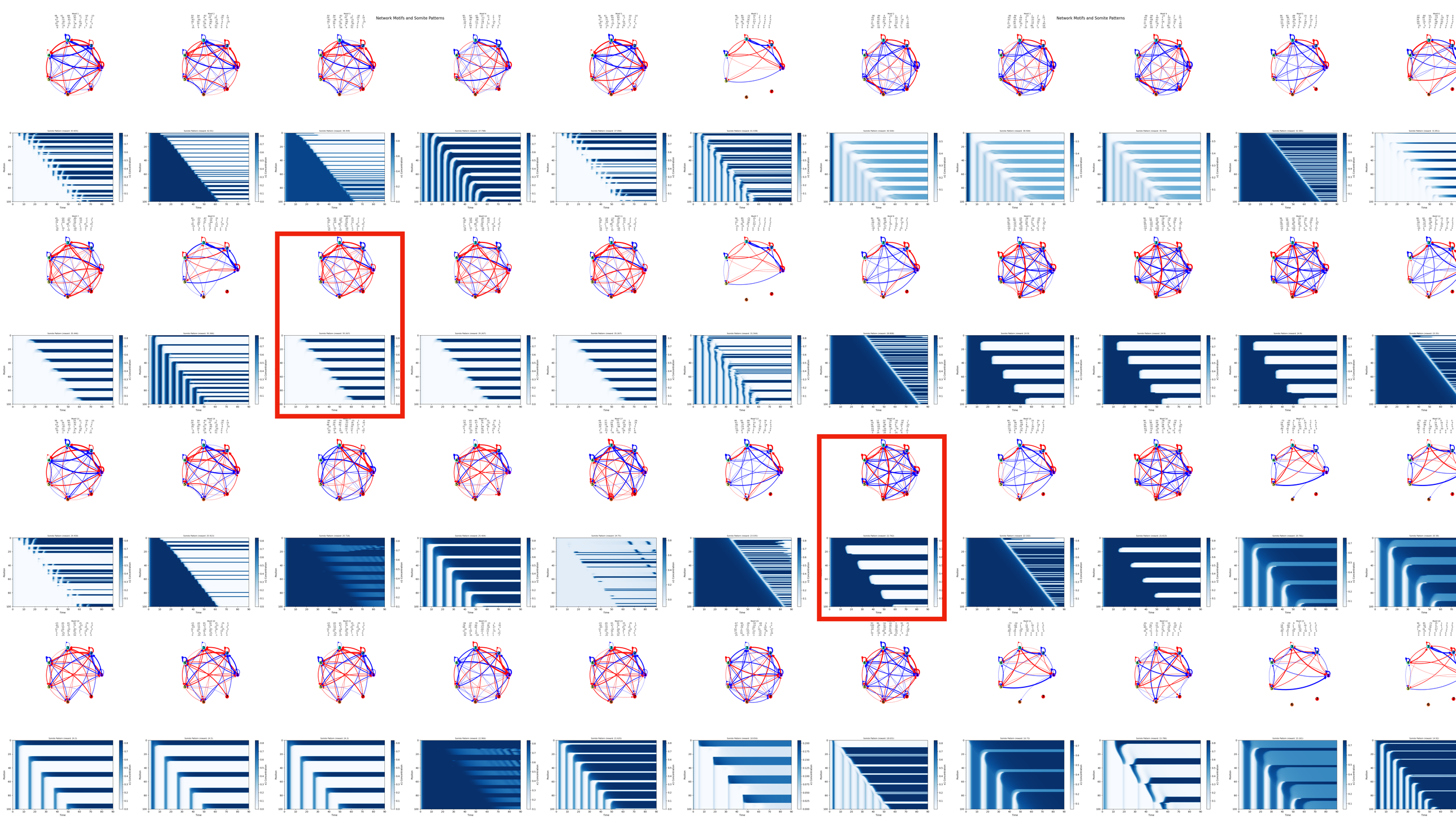
THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

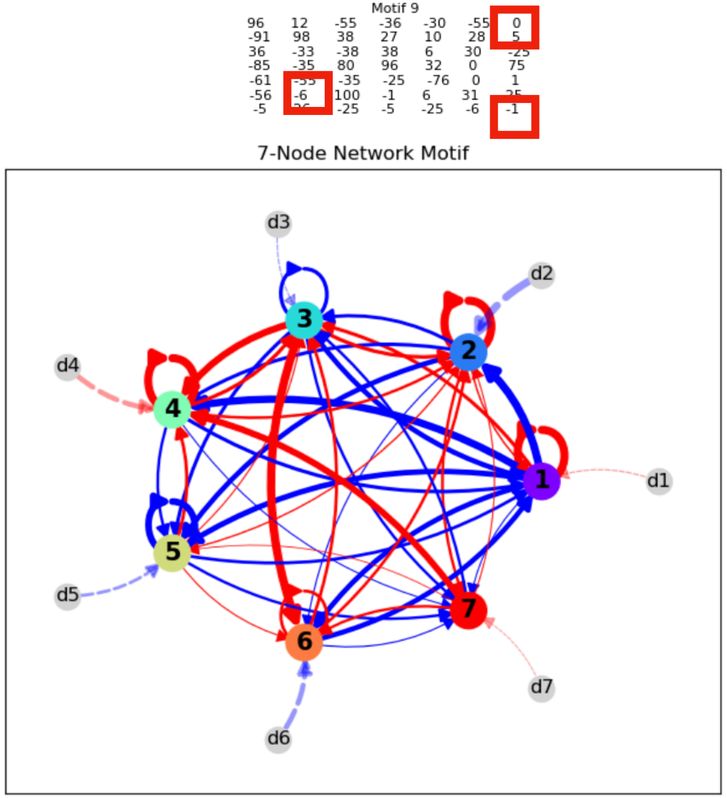
Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

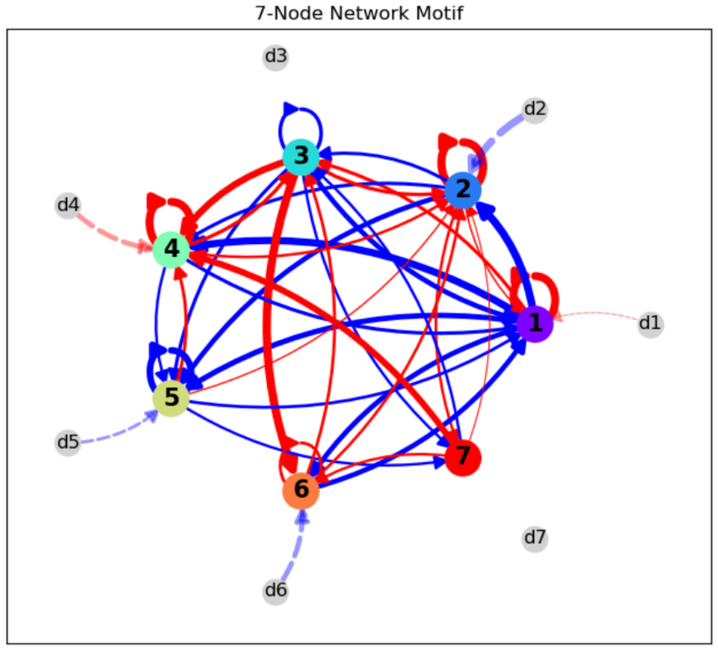
Journal Club - 06.02
Qichen Huang
Paul François Lab

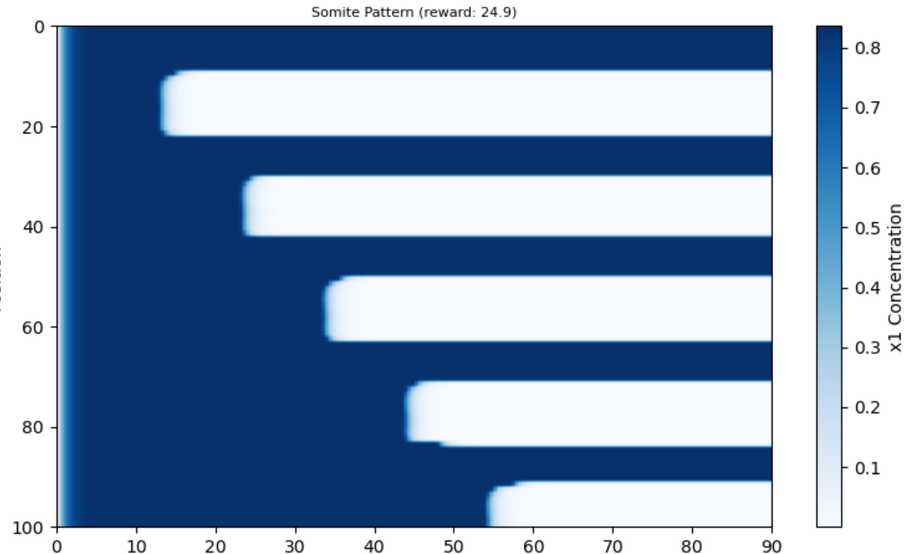
Network Motifs and Somite Patterns

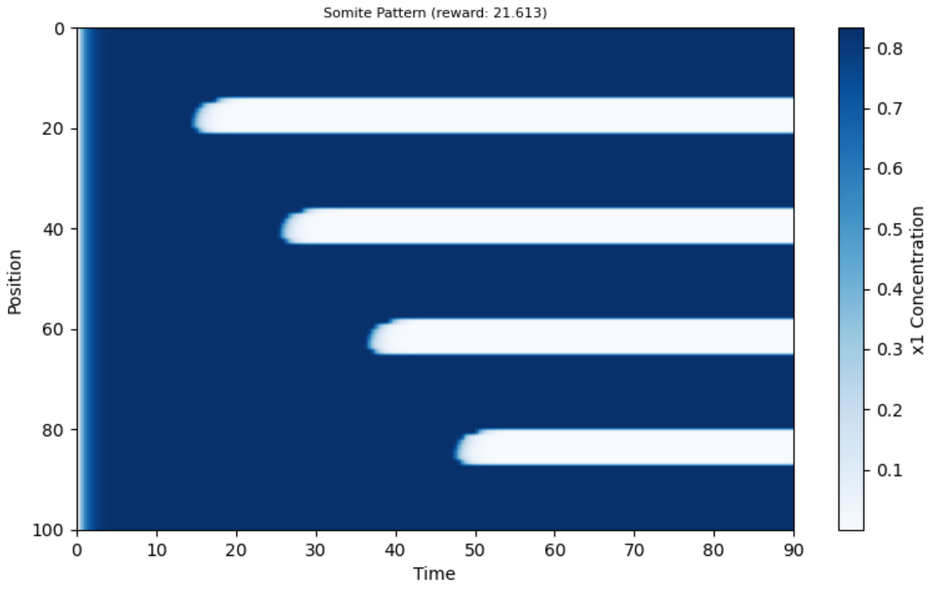



Motif 9
96 12 -55 -36 -30 -55 0
-91 98 38 27 10 28 5
36 -33 -38 38 6 30 -25
-85 -35 80 96 32 0 75
-61 -55 -35 -25 -76 0 1
-56 -6 100 -1 6 31 25
-5 -25 -5 -25 -6 -1
7-Node Network Motif
d3
d2
d4
d1
d5
d7
d6
1
2
3
4
5
6
7

prunning


7-Node Network Motif
d3
d2
d4
d1
d5
d7
d6
1
2
3
4
5
6
7


Somite Pattern (reward: 24.9)
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80
90
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
x1 Concentration

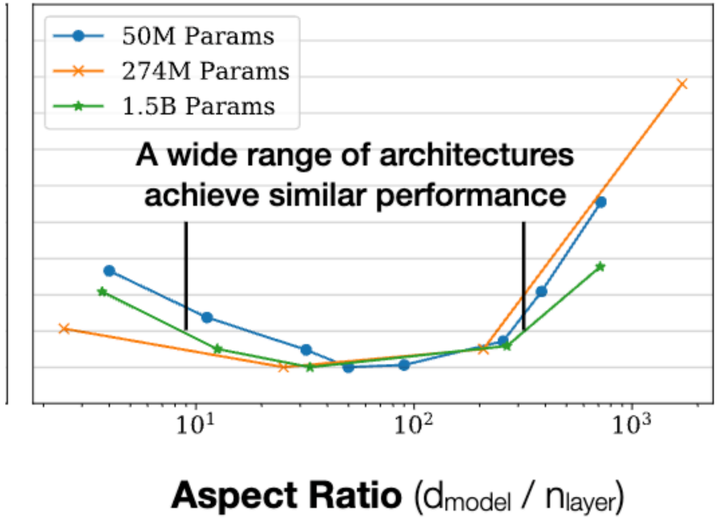

Somite Pattern (reward: 21.613)
Position
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80
90
Time
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
x1 Concentration

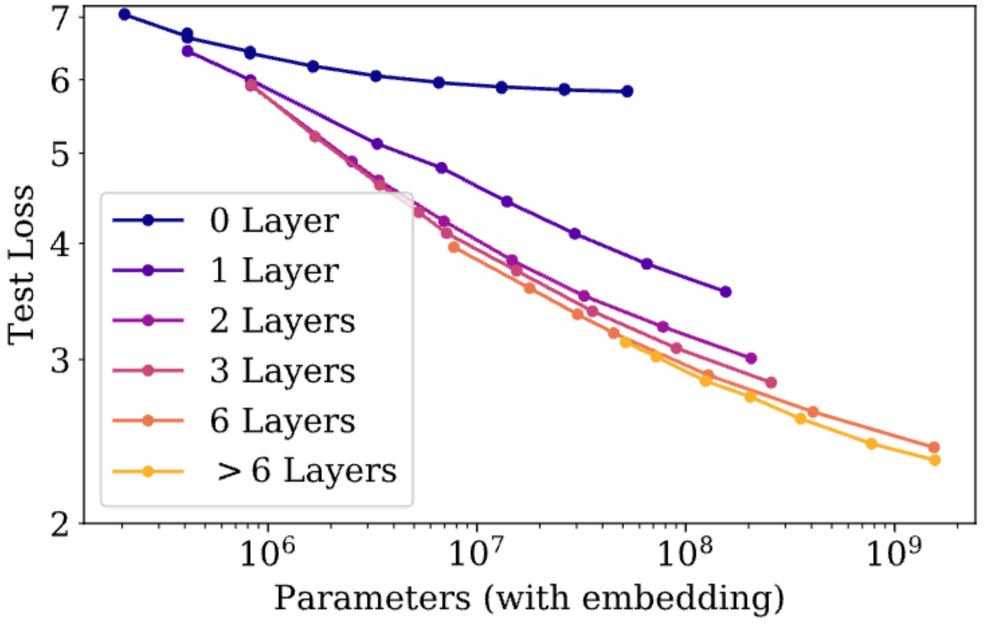
How to prune the network ?

Why prune the network ? / Why not to prune the network

Language Models are Few-Shot Learners

Scaling Laws for Neural Language Models





Learning both Weights and Connections for Efficient Neural Networks

Song Han
Stanford University
songhan@stanford.edu

Jeff Pool
NVIDIA
jpool@nvidia.com

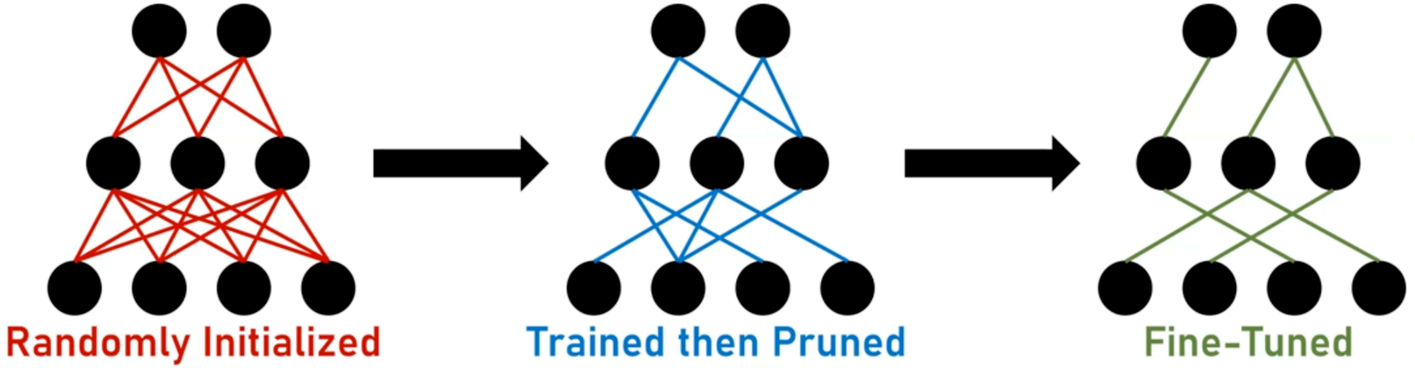
John Tran
NVIDIA
johntran@nvidia.com

William J. Dally
Stanford University
NVIDIA
dally@stanford.edu

2015

Prunning **weights** with lowest magnitude

1) Train the network
2) Remove superfluous structure
3) Fine-tune the network
4) Optionally: prune and fine-tune iteratively



Accuracy dropped

ICLR 2019

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

The Lottery Ticket Hypothesis :

The neural networks we typically train have **subnetworks** (at non-trivial sparsities) at **initialization** that can train to full accuracy in the same number of steps as the original network.

ICLR 2019

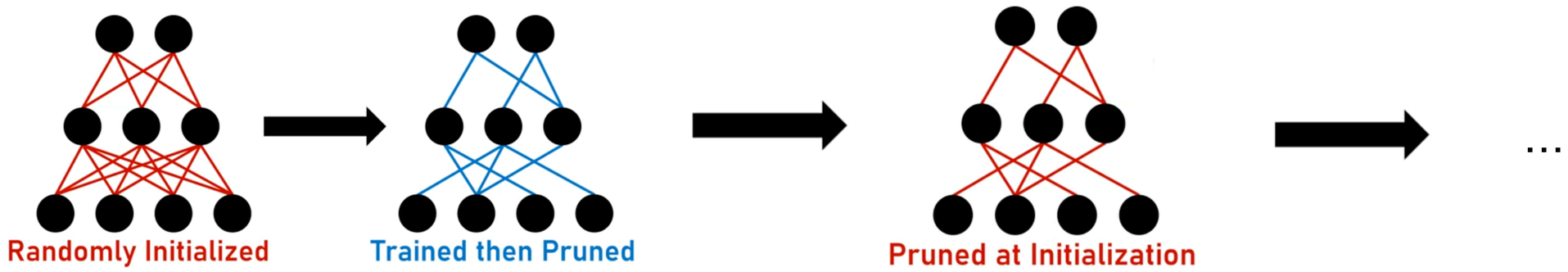
THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

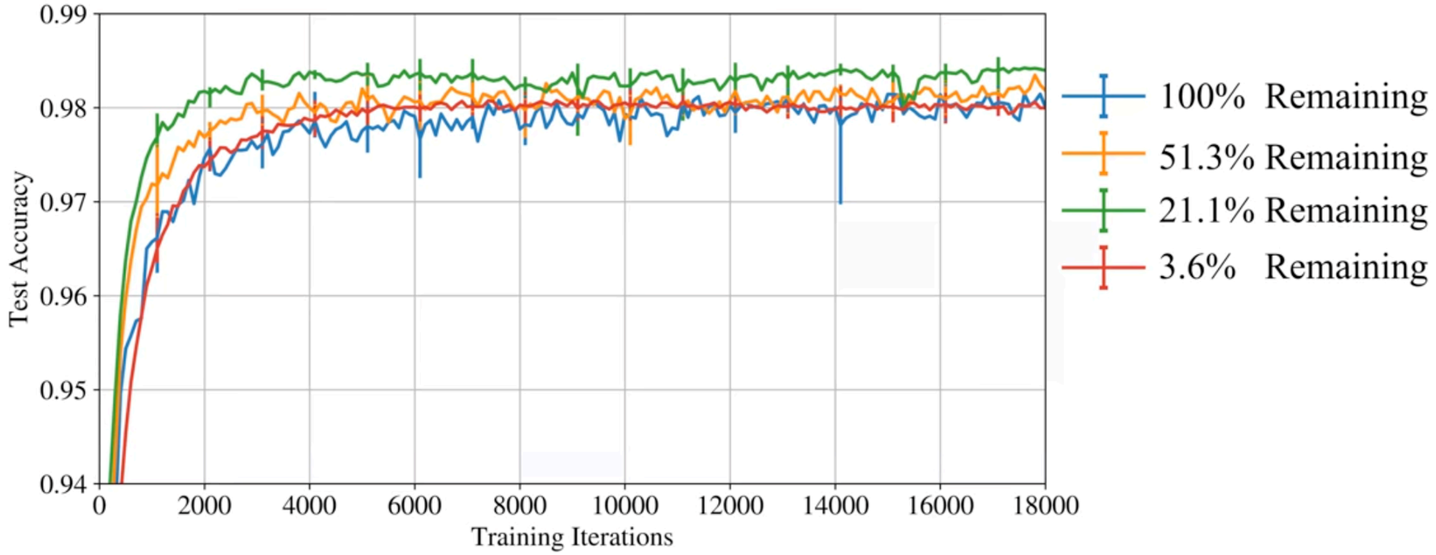
The Importance of Initialization

1) Train the network
2) Remove superfluous structure
3) Reset each weight to its original initialization
4) Train it to convergence

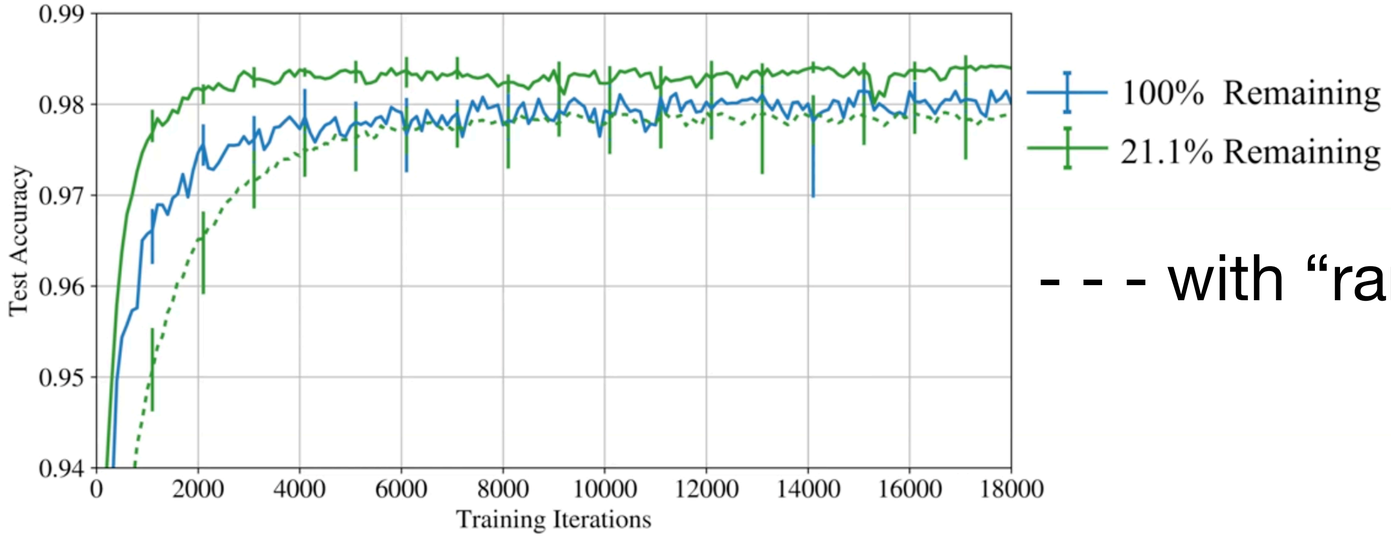


Reset weights to its original initialization

Retrain

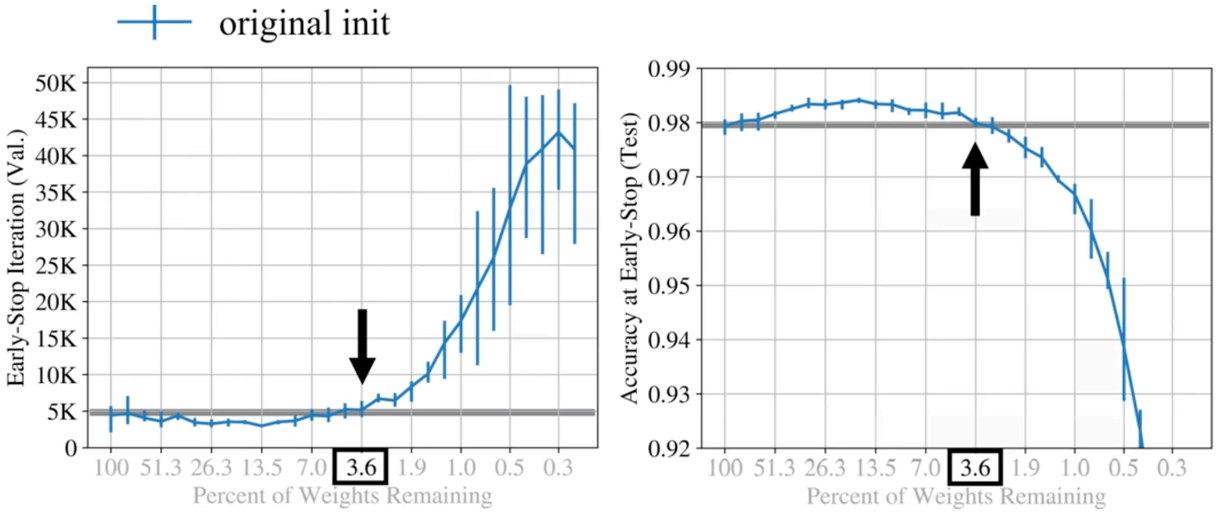


LeNet 300-100-10 for MNIST fully-connected 300K parameters

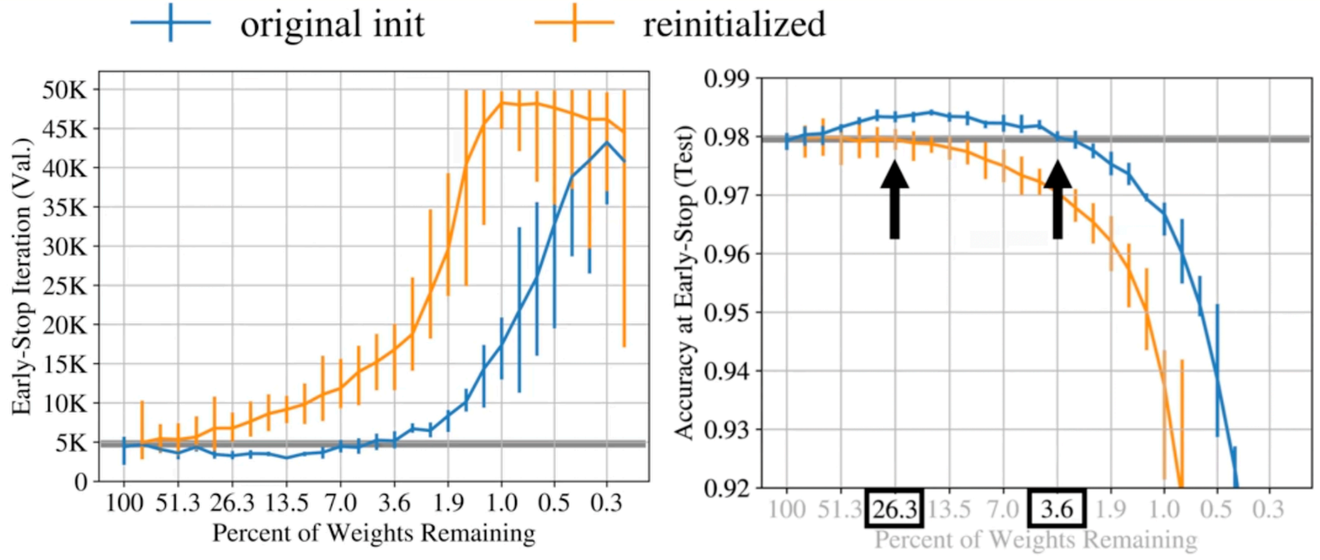


- - - with “random reinit”

LeNet 300-100-10 for MNIST fully-connected 300K parameters

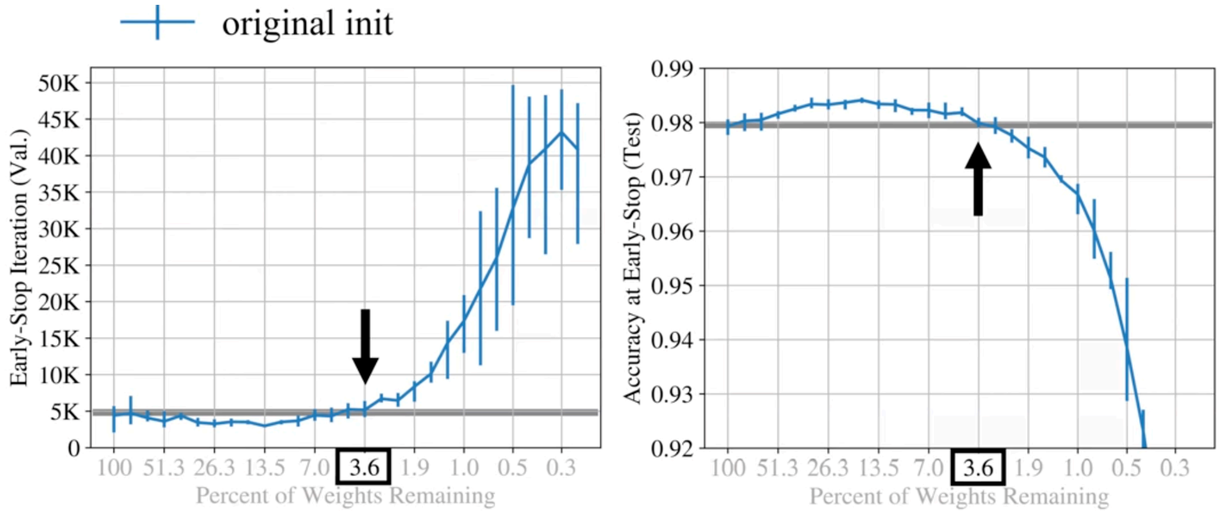


LeNet 300-100-10 for MNIST fully-connected 300K parameters

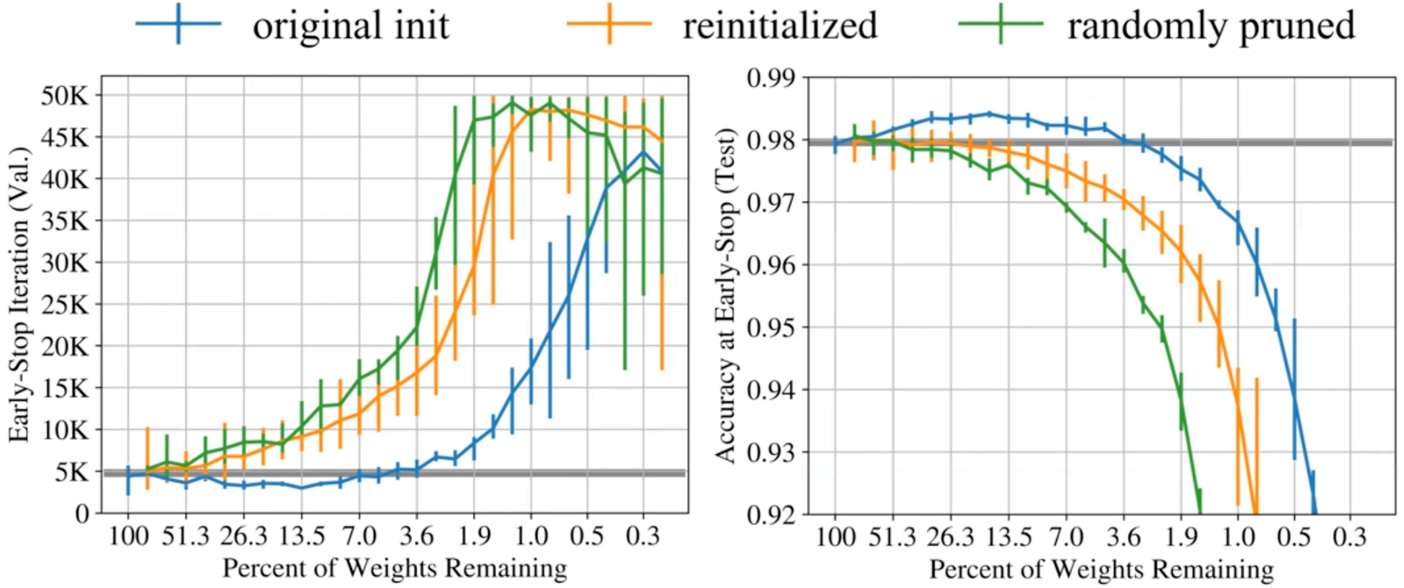


LeNet 300-100-10 for MNIST fully-connected 300K parameters

initialization matters



LeNet 300-100-10 for MNIST fully-connected 300K parameters



LeNet 300-100-10 for MNIST fully-connected 300K parameters

subnetwork
architecture matters

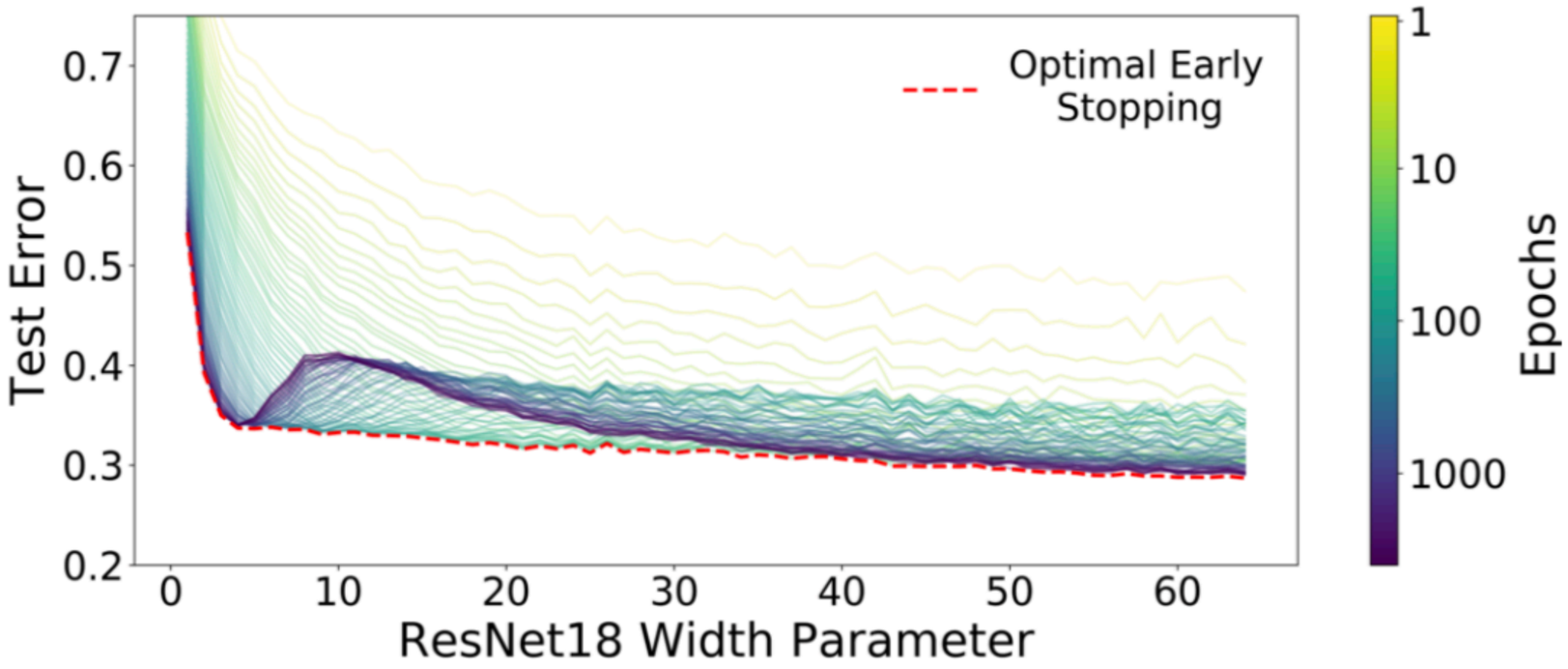
🎟 The Lottery Ticket Hypothesis :

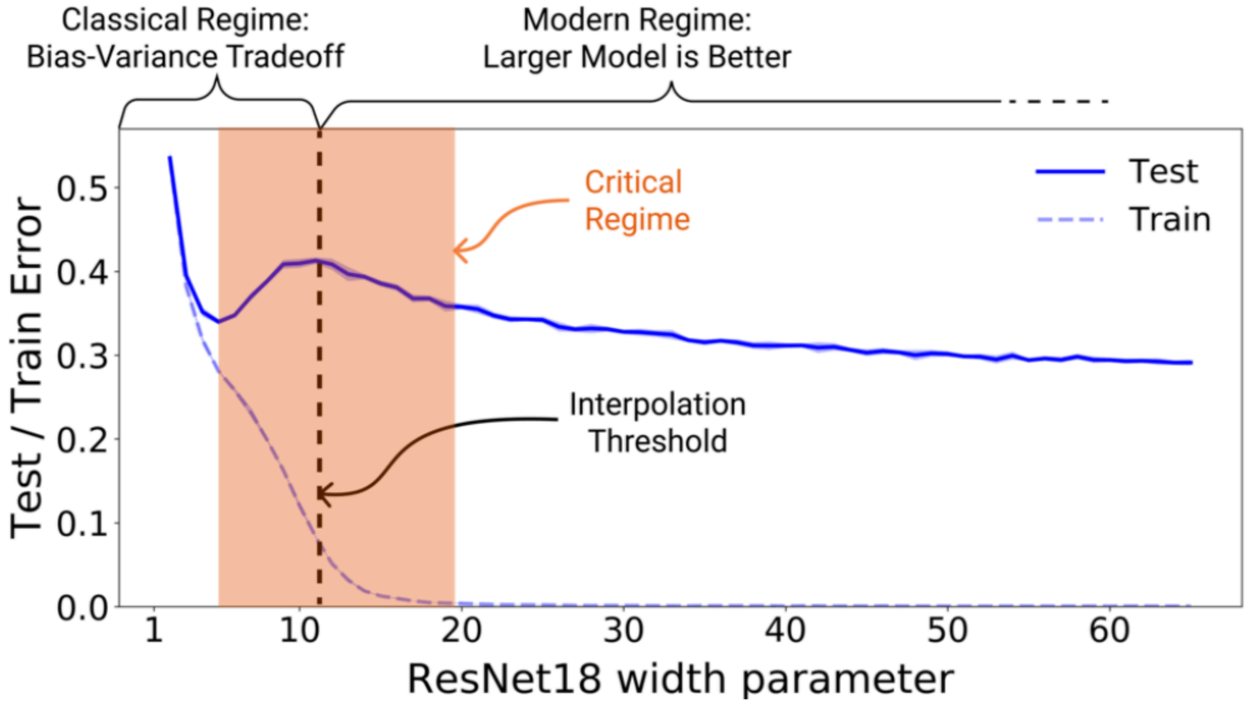
The neural networks we typically train have **subnetworks** (at non-trivial sparsities) at **initialization** that can train to full accuracy in the same number of steps as the original network.

If we can use this smaller network,

then why do we even bother with the larger one ?

double descent





statistical learning theory
The simplest model really is the best generalizer.

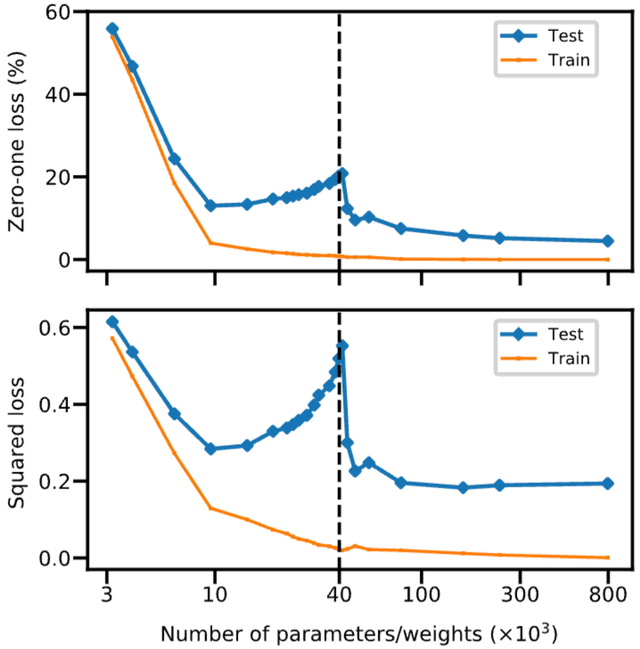
Occam's razor, also known as the principle of parsimony, suggests that when faced with multiple explanations for a phenomenon, the simplest one is usually the most likely to be correct.

Ockham chooses a razor

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

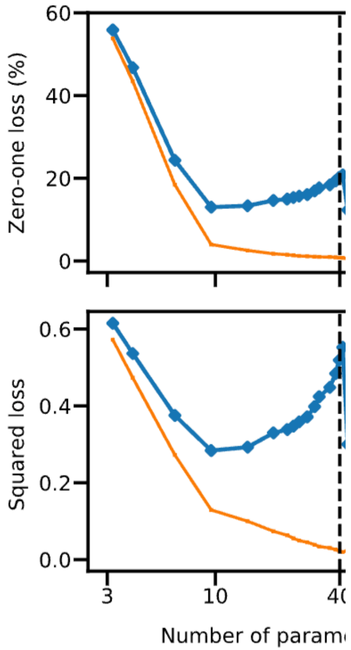
Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

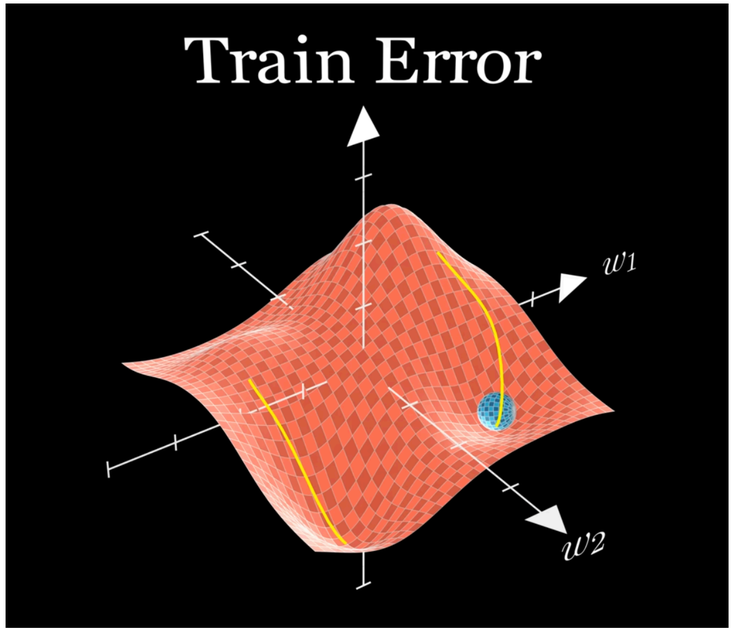
Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

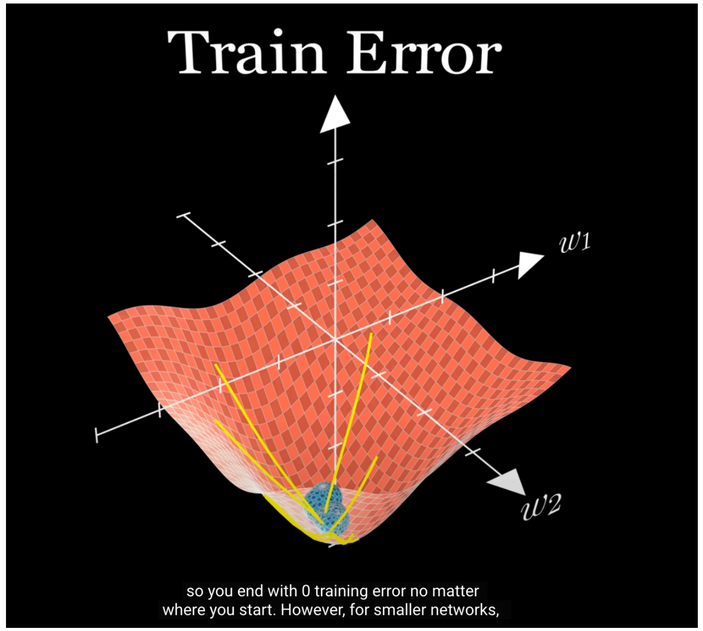


The lottery ticket hypothesis:
randomly-initialized NN contain subnetworks (winning tickets) can reach test accuracy **comparable to** the original network in a similar number of iterations.

The winning tickets we find have won the initialization lottery: their connections have initial weights that make training particularly effective.

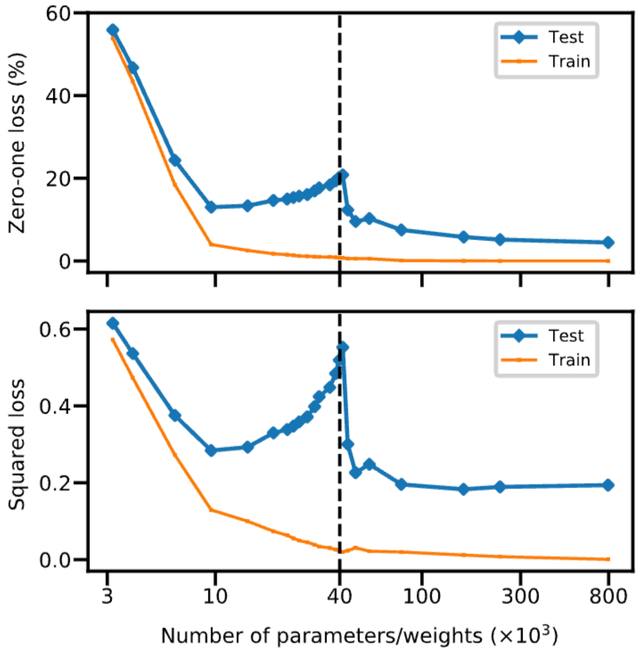


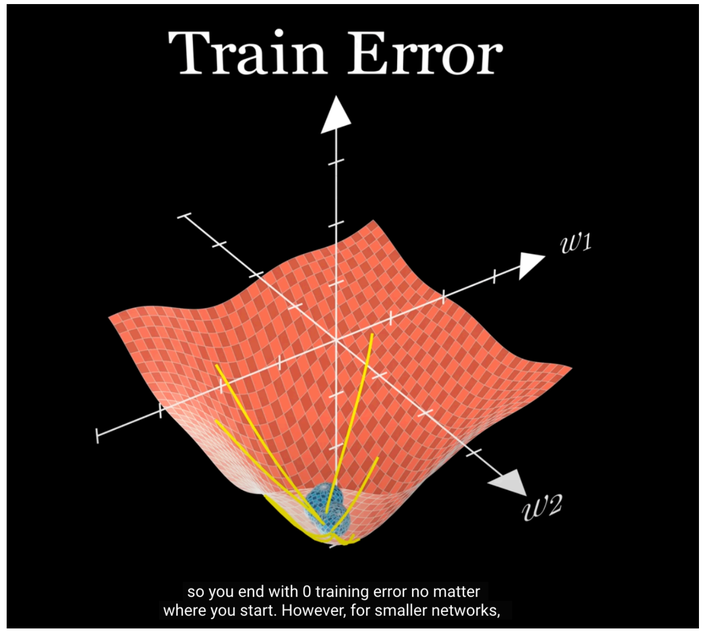




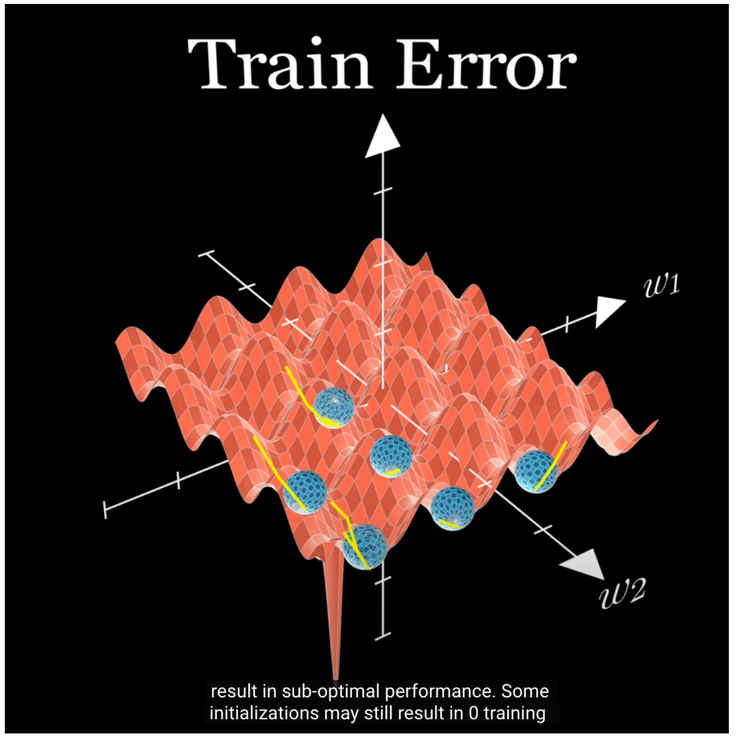
Receive ~ 0 training error no matter where you start.

However, for smaller networks,
the initialization becomes crucial, as more possible initializations result
in sub-optimal performances. (but tickets exist)





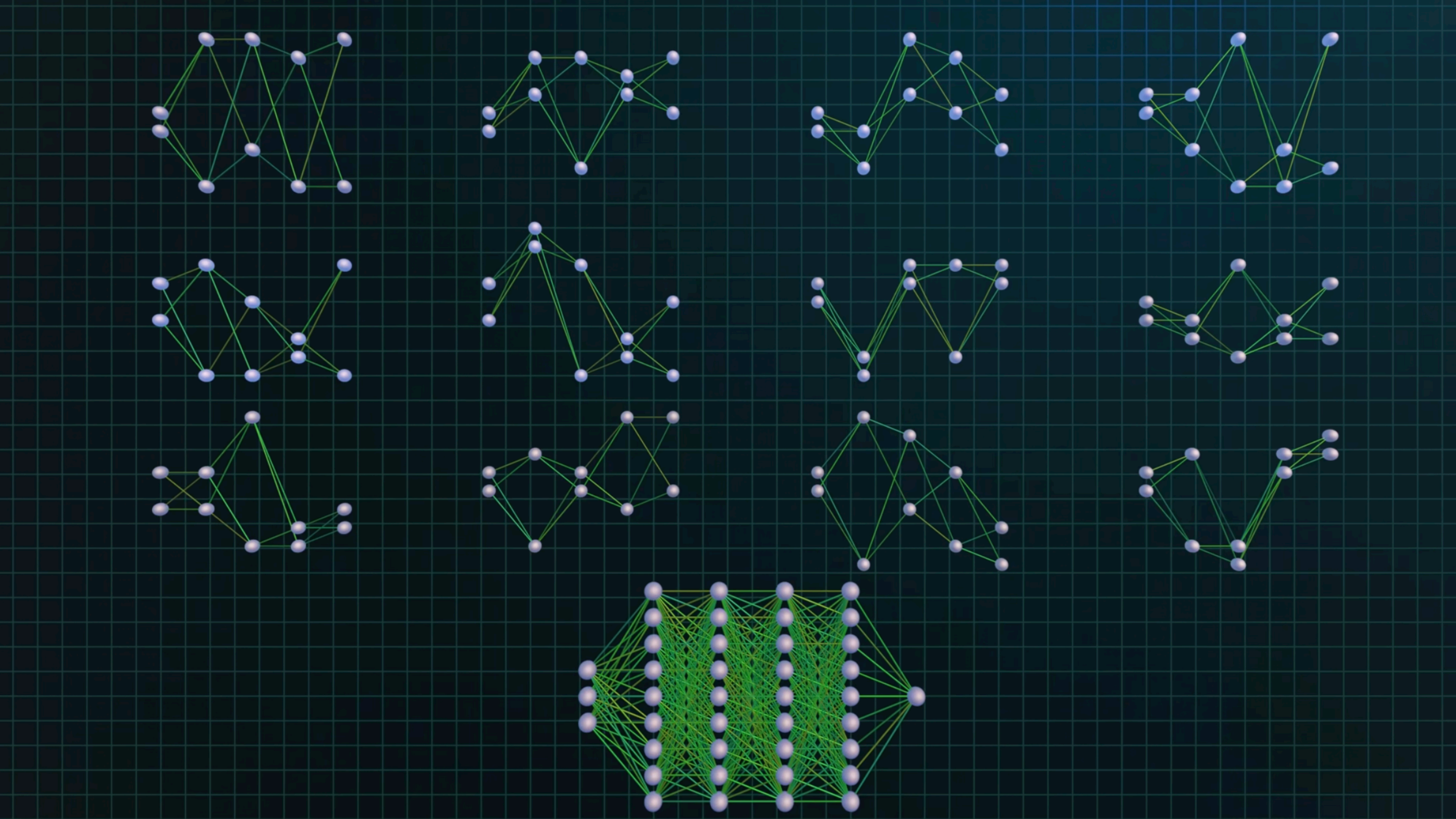
A large network



Many small sub-networks

Receive ~ 0 training error no matter where you start.

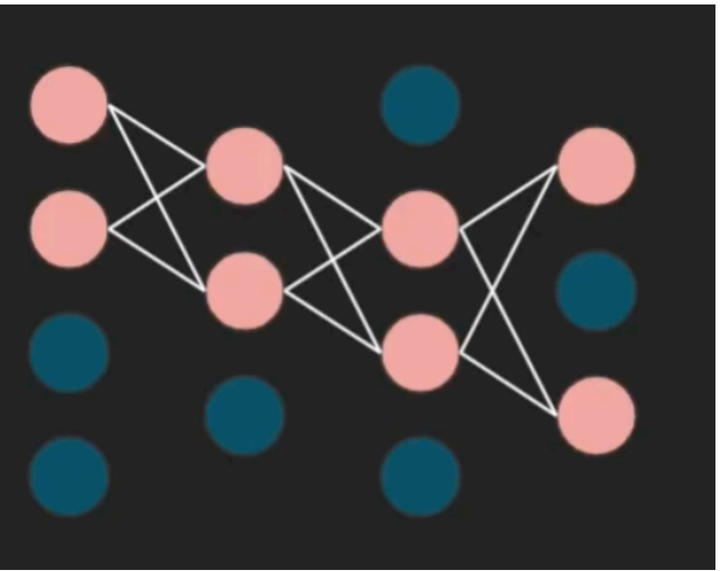
However, for smaller networks,
the initialization becomes crucial, as **more possible initializations** result in sub-optimal performances. (but 🎟️s **exist** in some rare initializations)



Hidden inside of a large neural net, is a much smaller **sub-network** that is actually doing all of the work. The rest is just useless fluff.

But why ?

why does it generalize better than a small network of the same size as this subnetwork.



🎟️ The Winning Ticket(s)

Two ways to find a sub-network:

pruning procedure —> **uncovering the sub-network** with the better initialization.

larger network —> more 🎟️s to choose from to win the lottery
—> **uncovering the sub-network** with the better initialization.

Prunning procedure =

they were essentially uncovering the sub-network with the best initialization.

Questions to take away

- fitness landscape
- GRNs are diverse
- GFN finds diversity