Lottery ticket hypothesis

BY- GRISHMA GUPTA, LOKIT PARAS

Motivation

Deep learning models have shown promising results in many domains. However, such models often have millions of parameters.

This has resulted in :

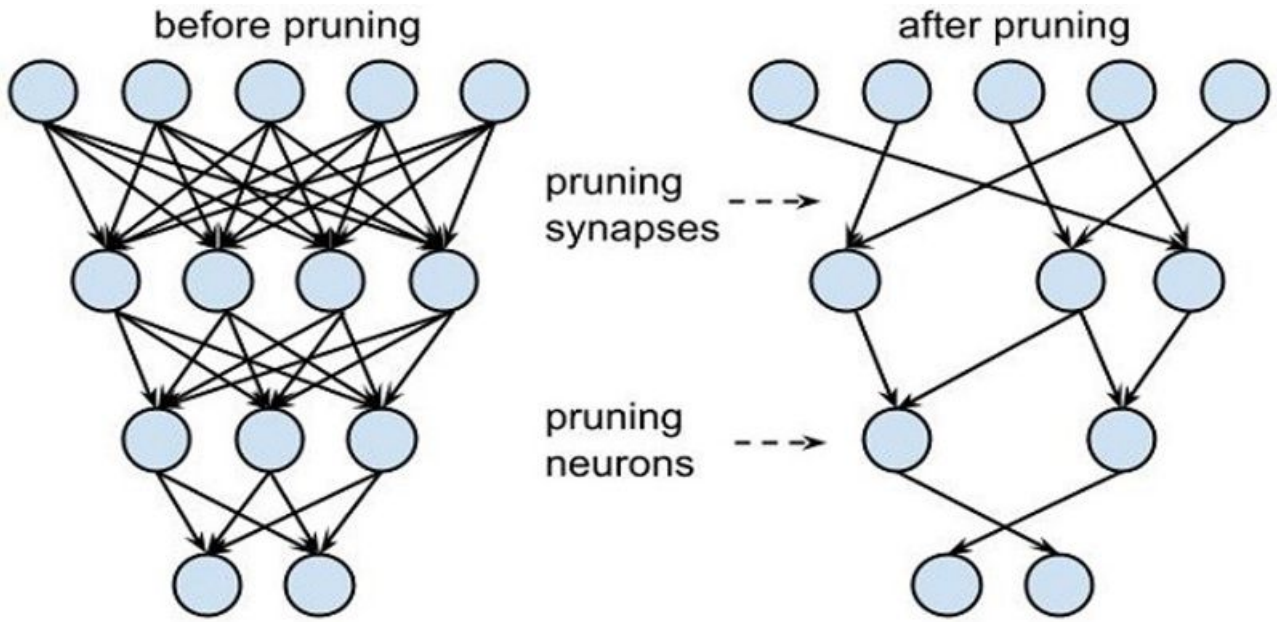
- Extremely long training periods (often days or weeks)
- Longer inference time
- Higher operational memory and computing requirements
- Increased storage requirements for deployed model.

Network pruning

- Technique in which unnecessary weights are removed from a neural network model after training

- Pruning can reduce model sizes by more than 90% without compromising on model accuracy while potentially offering a significant reduction in inference memory usage

Network pruning



Advantages

Larger Model => More Memory References => More Energy

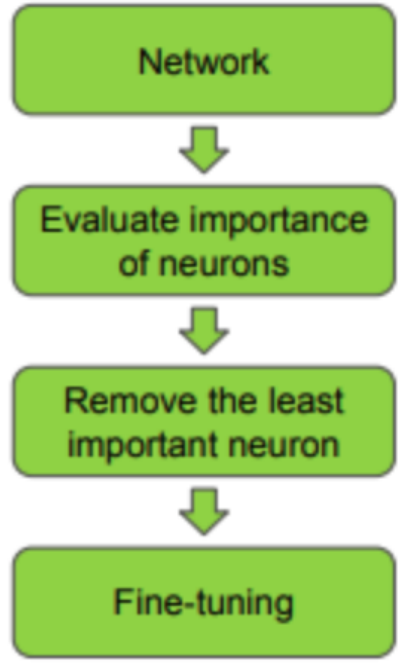
- models smaller in size
- more memory-efficient
- more power-efficient
- faster at inference with minimal loss in accuracy

One-shot pruning

The network connections are pruned only once

Steps:

- Randomly initialize a neural network.
- Train the network for certain iterations to find optimal weights.
- Prune p% of weights from each layer in the model.

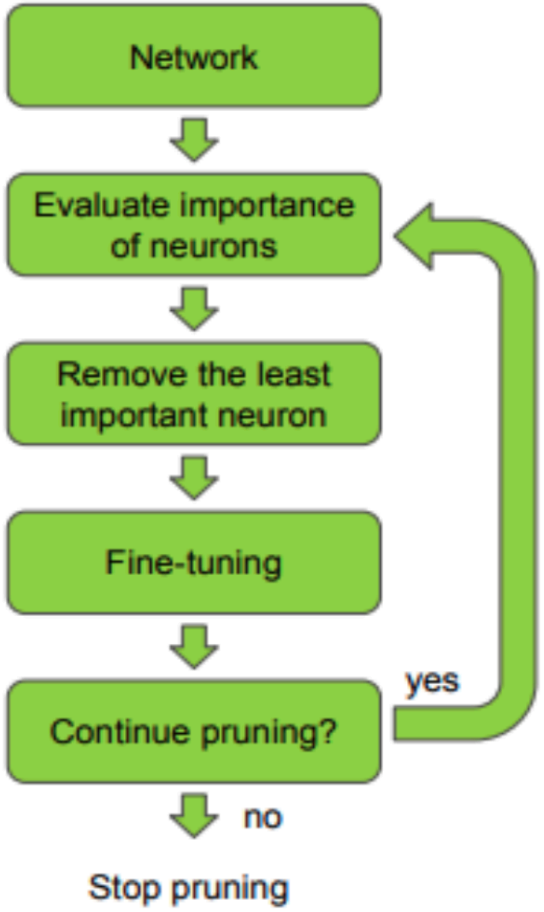


Iterative pruning

Pruning is done partially through multiple iterations

Steps

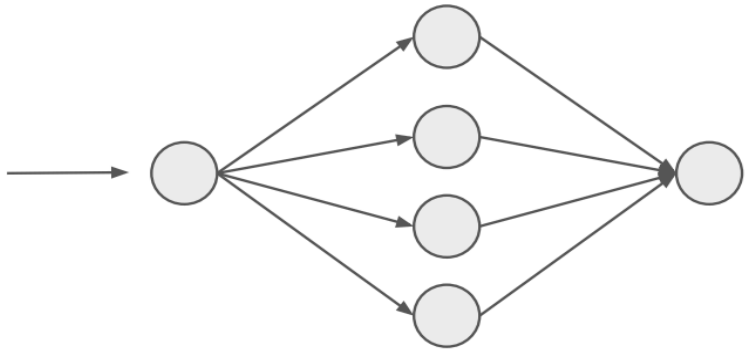
- Randomly initialize a neural network.
- Repeat for n rounds:
  - Train the network for certain iterations.
  - Prune $p^{(1/n)}$% of weights that survived previous pruning.



Is the pruned architecture enough?

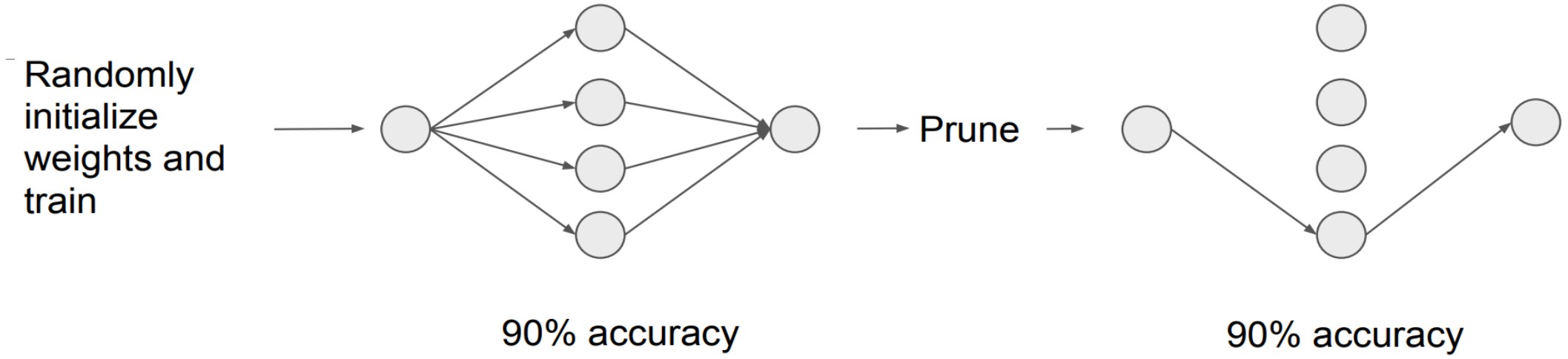
Re-initializing a pruned network

Randomly initialize weights and train

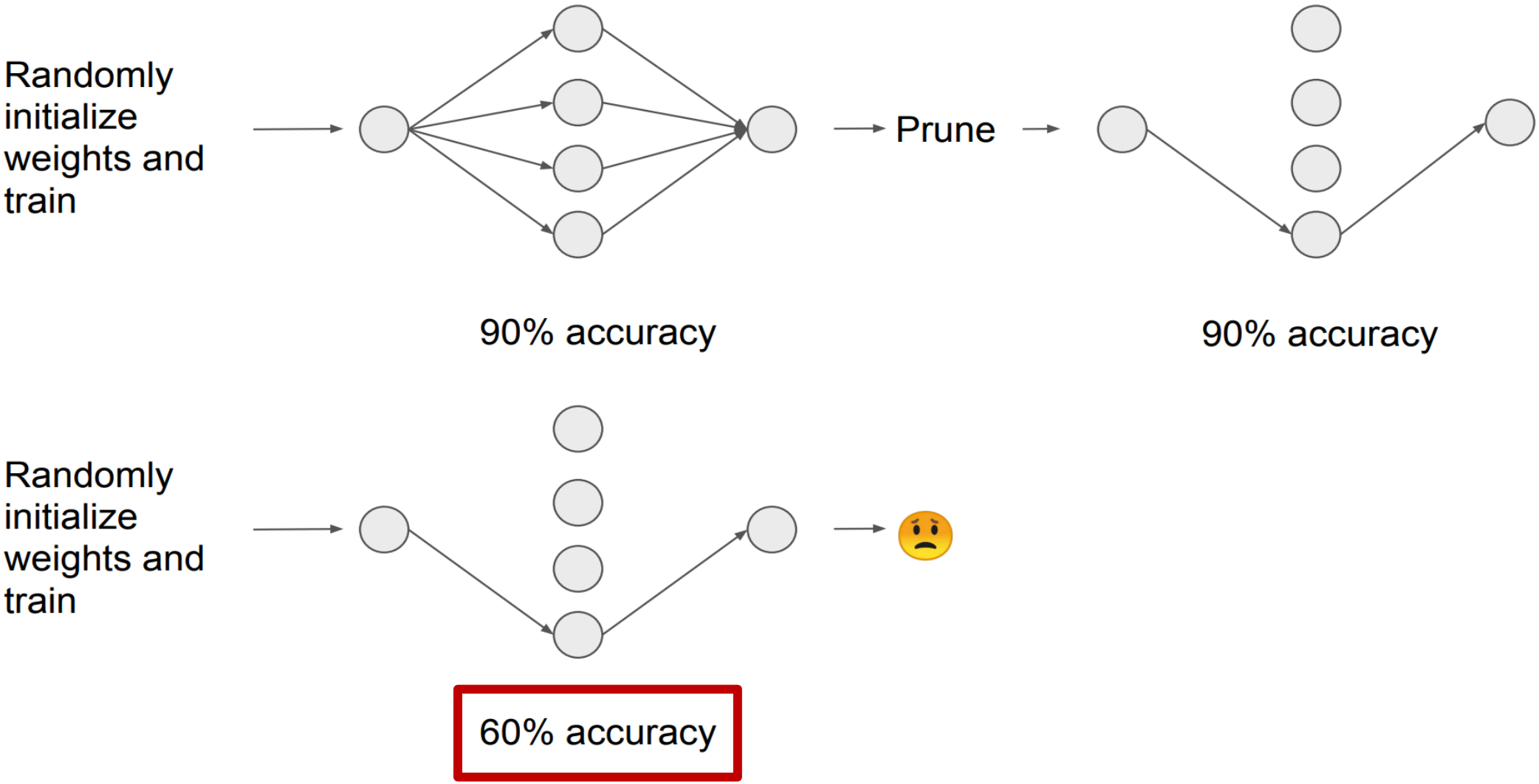


90% accuracy

Re-initializing a pruned network



Re-initializing a pruned network



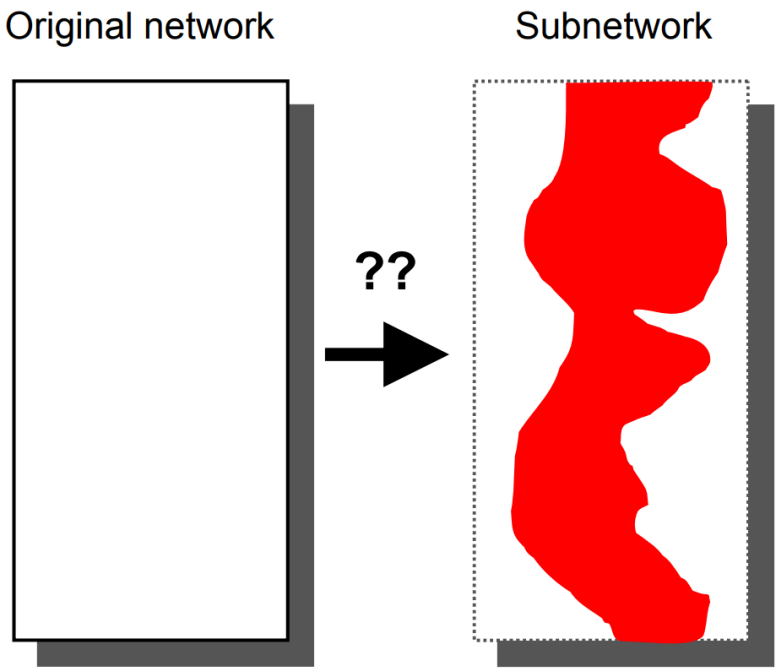
So, only the pruned architecture is not enough!

So, only the pruned architecture is not enough!

The initialization weights have a role to play

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.

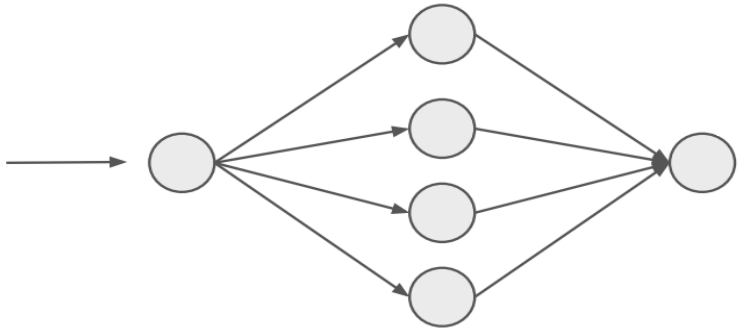


What is the Lottery Ticket Hypothesis about?

- Is there a subnetwork with better results
- Shorter training time
- Notably fewer parameters
- Trainable from beginning?

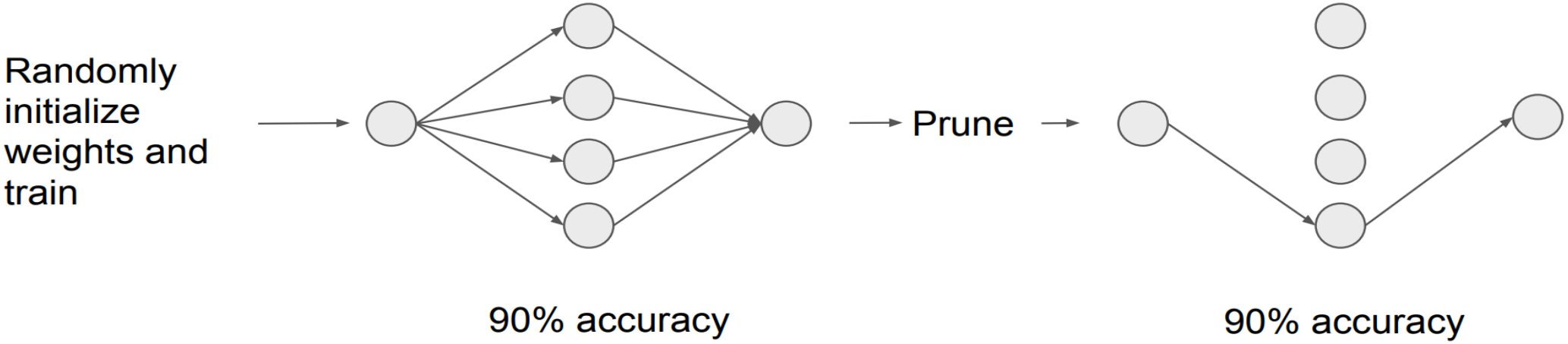
The hypothesis

Randomly initialize weights and train



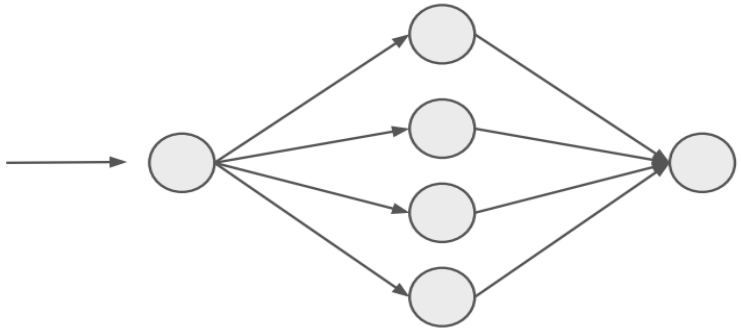
90% accuracy

The hypothesis



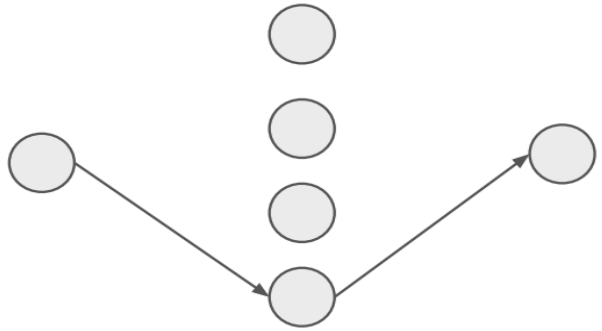
The hypothesis

Randomly initialize weights and train



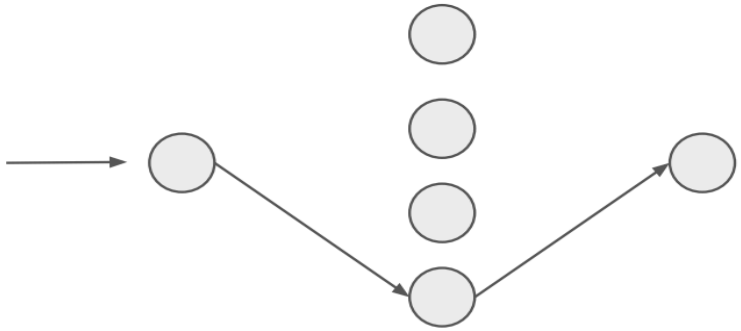
90% accuracy

Prune



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

The Lottery Ticket Hypothesis

Consider a **dense feed-forward neural network $f(x;\theta)$** with **initial parameters $\theta = \theta_0 \sim D_\theta$**

Where θ_0 is the chosen initialization parameters from the parameter space D_θ

f reaches a **minimum validation loss l** at **iteration j** with **test accuracy a**

The Lottery Ticket Hypothesis

Consider a **dense feed-forward neural network $f(x;\theta)$** with **initial parameters $\theta = \theta_0 \sim D_\theta$**

Where θ_0 is the chosen initialization parameters from the parameter space D_θ

f reaches a **minimum validation loss l** at **iteration j** with **test accuracy a**

In addition, consider training another **network $f'(x; m \odot \theta)$**

with a **mask $m \in \{0, 1\}^{|\theta|}$** on its parameters such that initialization parameters are now $m \odot \theta_0$.

On the same training set (with m fixed), **f' reaches minimum validation loss l' at iteration j' with test accuracy a**

The Lottery Ticket Hypothesis

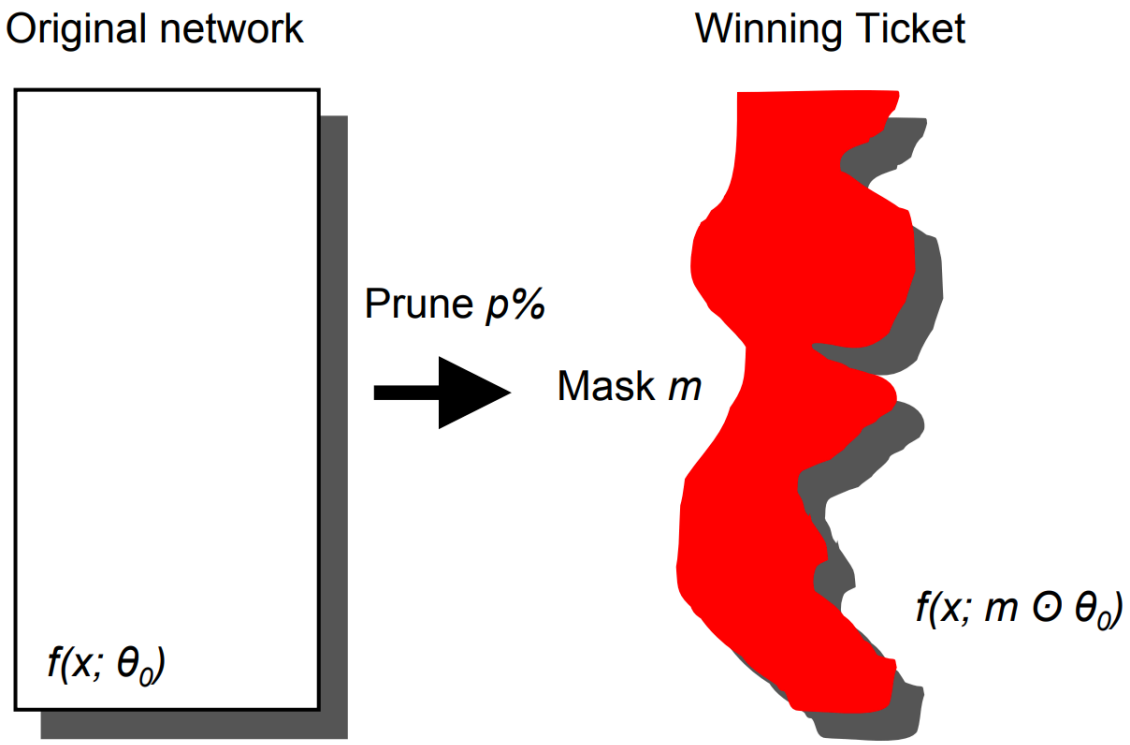
The lottery ticket hypothesis predicts that

$\exists$ m (mask) for which

- $j' \leq j$ (***commensurate training time***)
- $a' \geq a$ (***commensurate accuracy***)
- $\|m\|_0 \ll |\theta|$ (***fewer parameters***)

What are winning tickets?

*We designate these trainable subnetworks, **winning tickets**, since these subnetworks have **won the initialization lottery** with a combination of weights and connections capable of learning*



Winning Tickets

These winning tickets give:

- Better or same results
- Shorter or same training time
- Notably fewer parameters
- Is trainable from the beginning

Winning tickets - One-shot pruning

Steps:

1. Randomly initialize a neural network $f(x; \theta_0)$, with initial parameters θ_0

2. Train the network for j iterations, arriving at parameters θ_j

3. Prune p% of the parameters in θ_j , creating a mask m

4. Reset the remaining parameters to their value in θ_0 , creating the winning ticket $f(x; m \odot \theta_0)$.

Winning tickets - Iterative pruning

Steps:

1. Randomly initialize a neural network $f(x; \theta_0)$, with initial parameters θ_0
2. Train the network for j iterations, arriving at parameters θ_j
3. Prune $p^{1/n}$% of the parameters in θ_j , creating a mask m
4. Reset the remaining parameters to their value in θ_0 , creating network $f(x; m \odot \theta_0)$
5. Repeat n times from 2
6. Final network is a winning ticket $f(x; m \odot \theta_0)$

Empirically testing the hypothesis

Winning ticket for fully connected networks

To test their hypothesis, the authors applied it to fully-connected networks trained on MNIST.

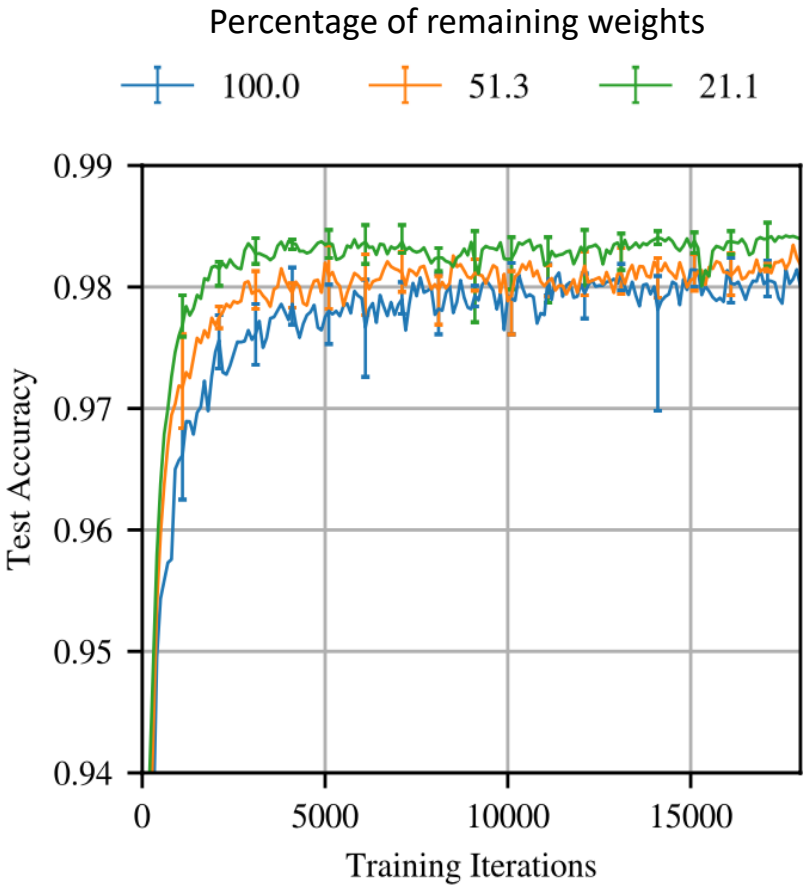
Experimental setup:

- Architecture: Lenet-300-100
- Pruning heuristic:
  - Remove a percentage of weights layer-wise,
  - Magnitude based (remove lower magnitude)

Pruning Rate and Sparsity

- p% is the Pruning Rate
- P_m is the Sparsity of the pruned network (mask)
- E.g. P_m = 25% when p% = 75% of weights are pruned

Effect of pruning on accuracy

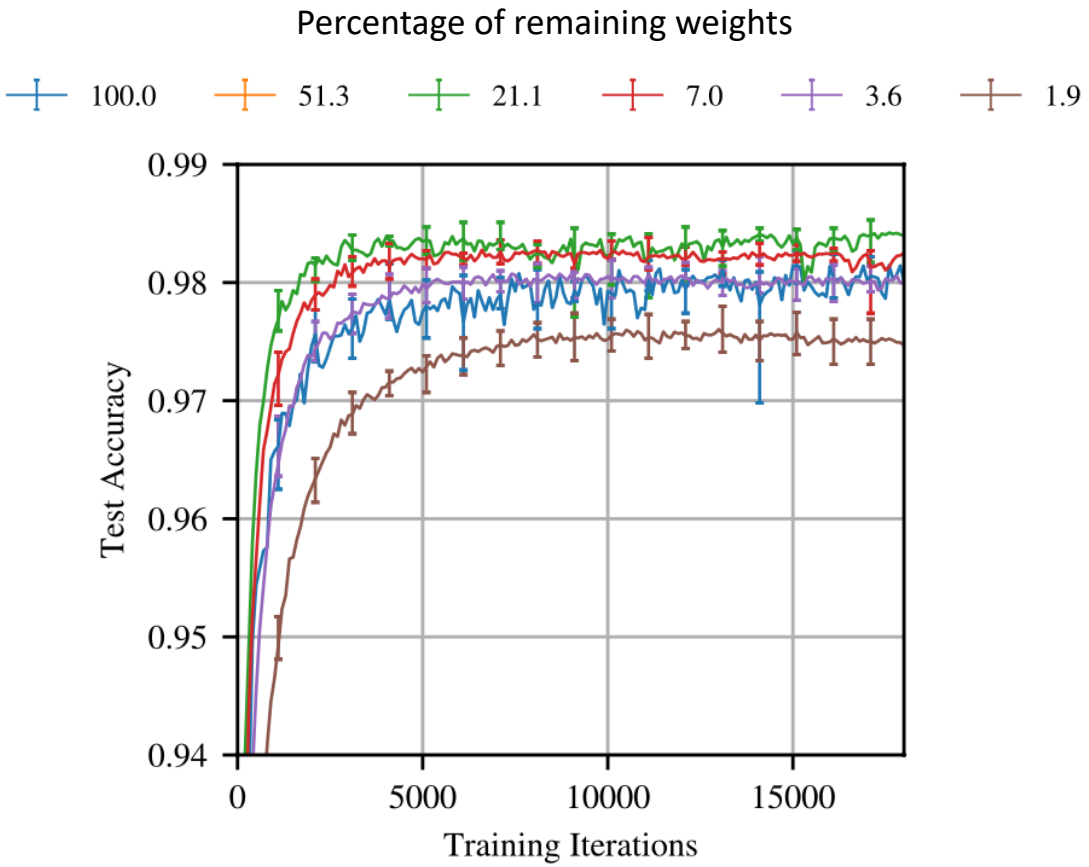


Observation:

- The winning tickets we find learn faster than the original network.
- Winning tickets also reach a higher test accuracy than original network.

In this experiment,
A winning ticket comprising 51.3% of the weights (i.e., P_m = 51.3%) reaches higher test accuracy faster than the original network but slower than when P_m = 21.1%

Effect of pruning on accuracy



Observation:

- Beyond a certain percentage, pruning starts reducing the model's accuracy.

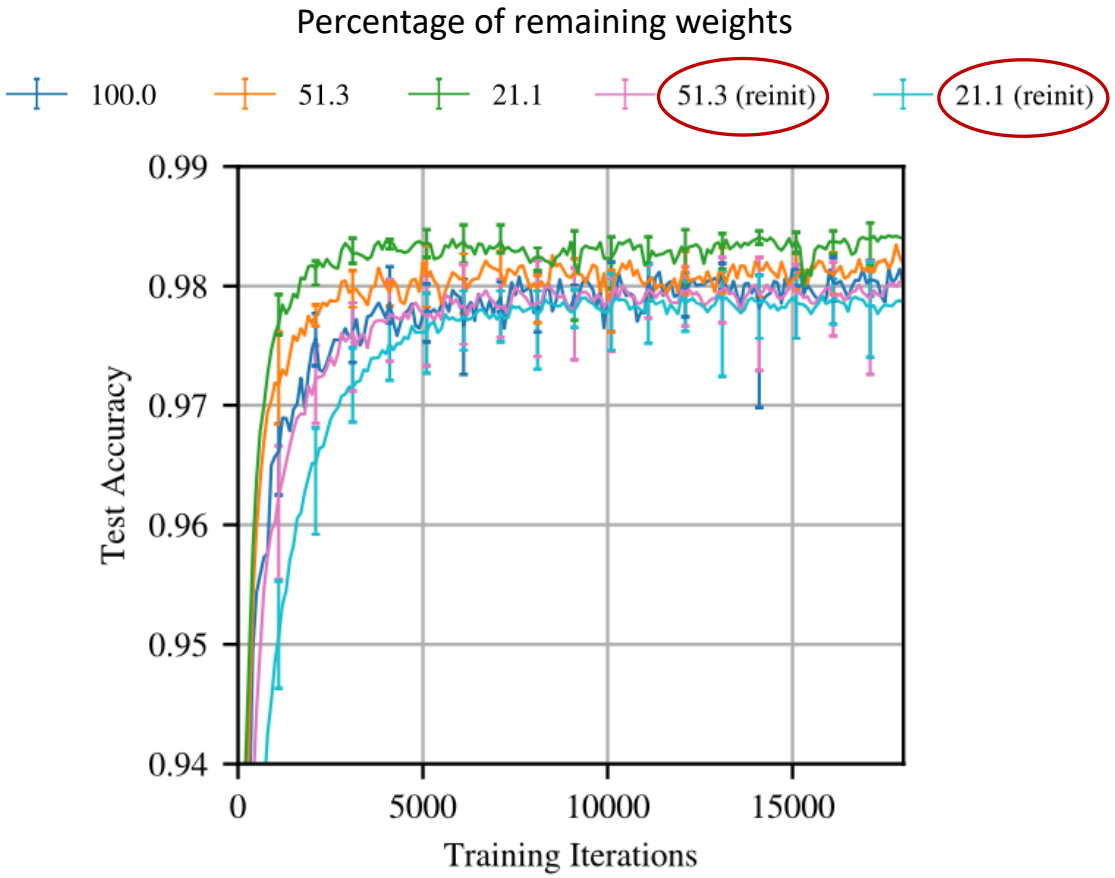
In this experiment,
When P_m = 3.6%, a winning ticket regresses to the performance of the original network.

Pruning + Re-initialization

To measure the importance of a winning ticket's initialization, we retain the structure of a winning ticket (i.e. the mask m)

but randomly sample a new initialization θ_0

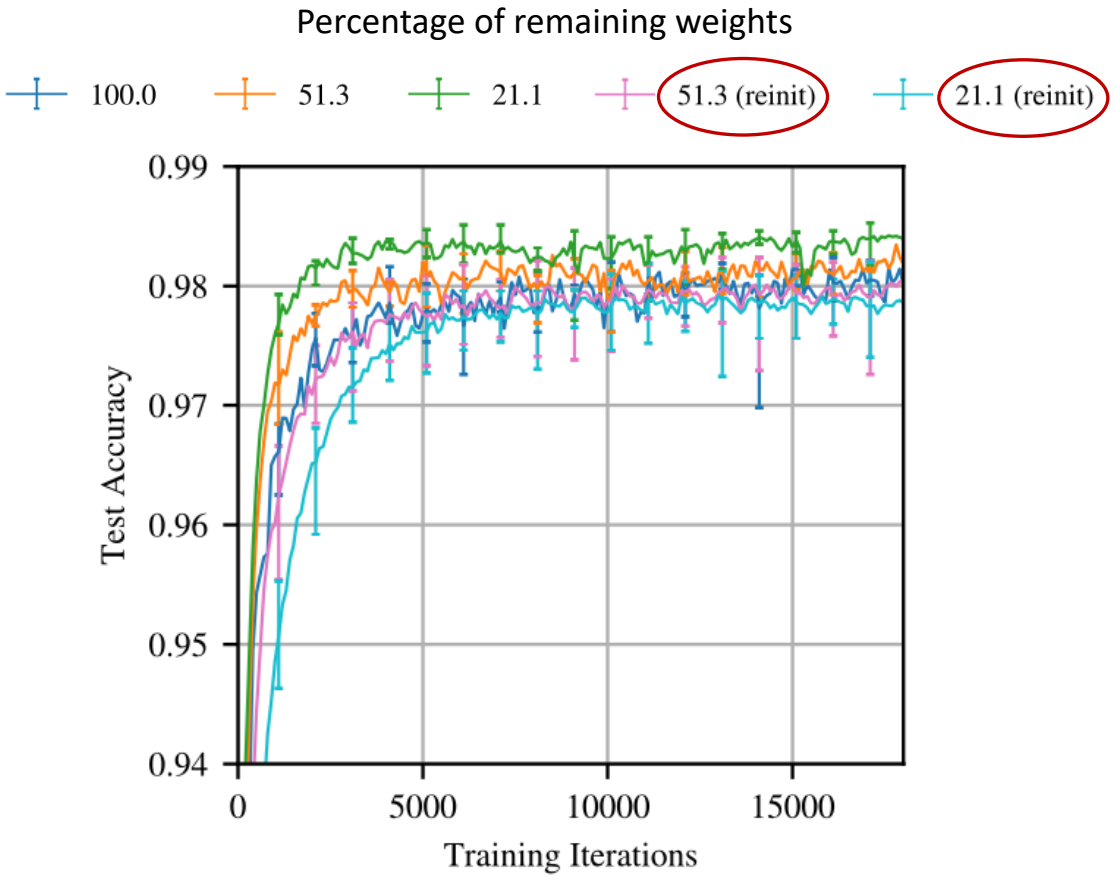
Pruning + Re-initialization



Observation:

- Unlike winning tickets, the reinitialized networks learn increasingly slower than the original network and lose test accuracy after little pruning.

Pruning + Re-initialization



Observation:

- Unlike winning tickets, the reinitialized networks learn increasingly slower than the original network and lose test accuracy after little pruning.

Conclusion:

The initialization is crucial for the efficacy of a winning ticket

Early-stopping Criteria

Early stopping criteria used in the paper is –> minimum validation loss

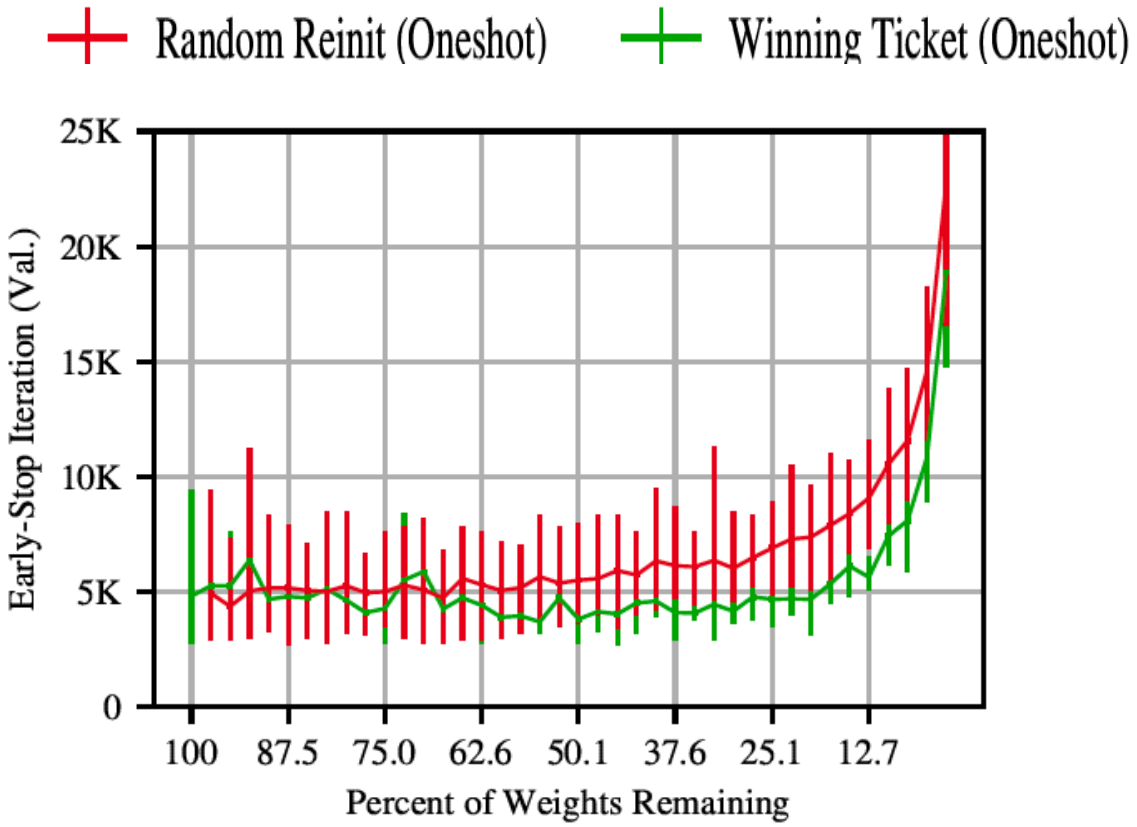
Validation and test loss follow a pattern where they decrease early in the training process, reach a minimum, and then begin to increase as the model overfits to the training data.

One-shot pruning to find winning tickets

1. Although iterative pruning extracts smaller winning tickets, repeated training means they are costly to find.

2. One-shot pruning makes it possible to identify winning tickets without this repeated training.

Early-stopping iterations for one-shot pruning.



Observation:

- 67:5% > P_m > 17:6%, the average winning tickets reach minimum validation accuracy earlier than the original network.

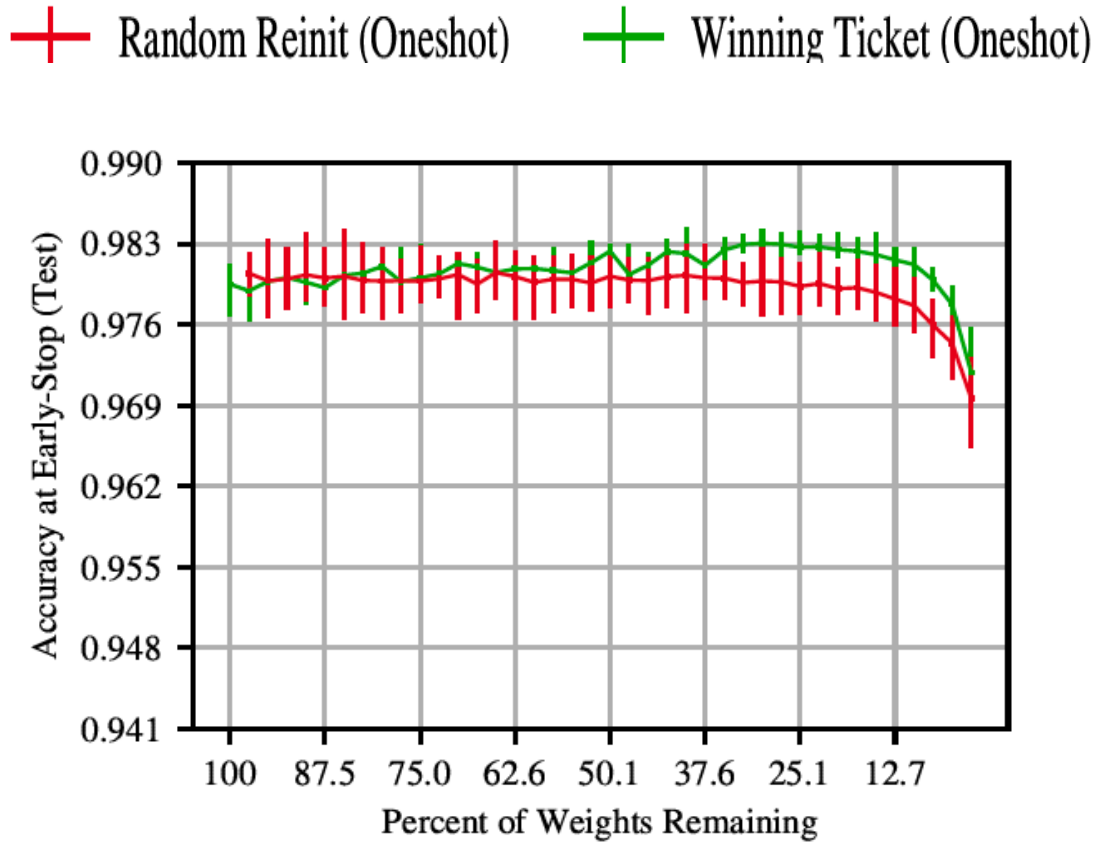
Accuracy for one-shot pruning.

Observation:

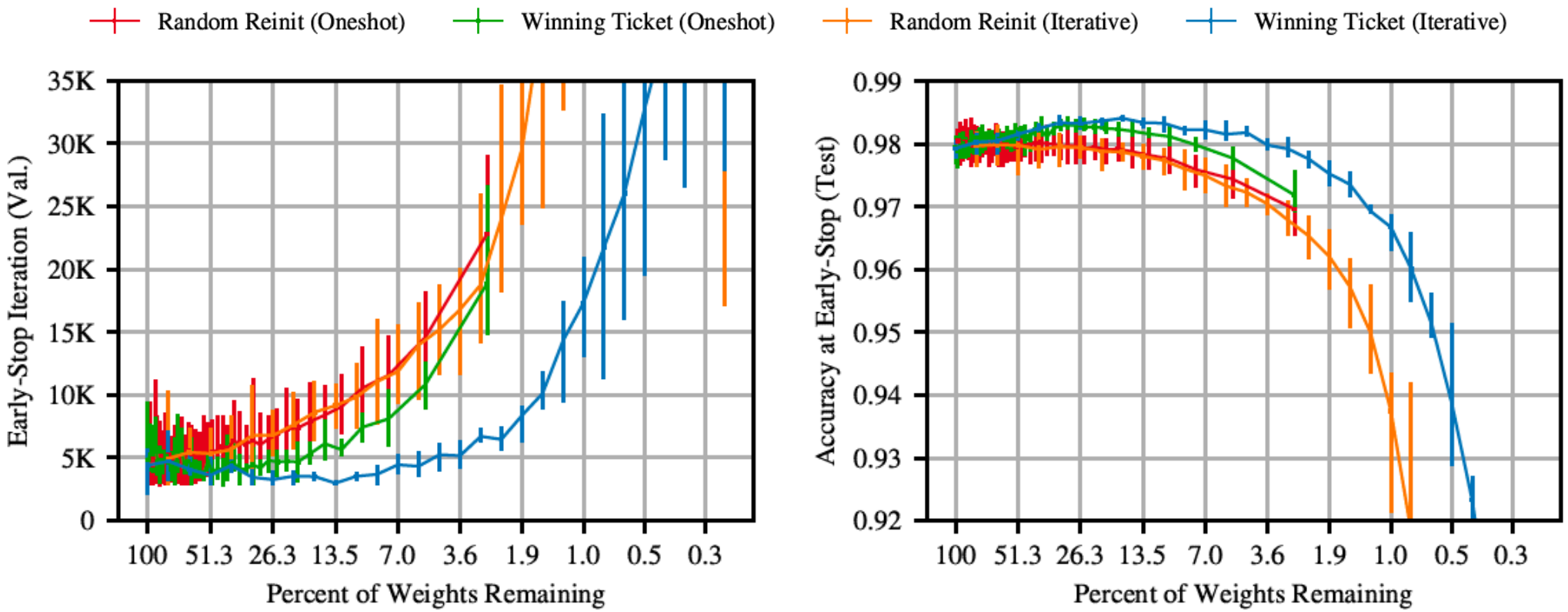
- 95:0% > P_m> 5:17%, test accuracy is higher than the original network

Conclusion:

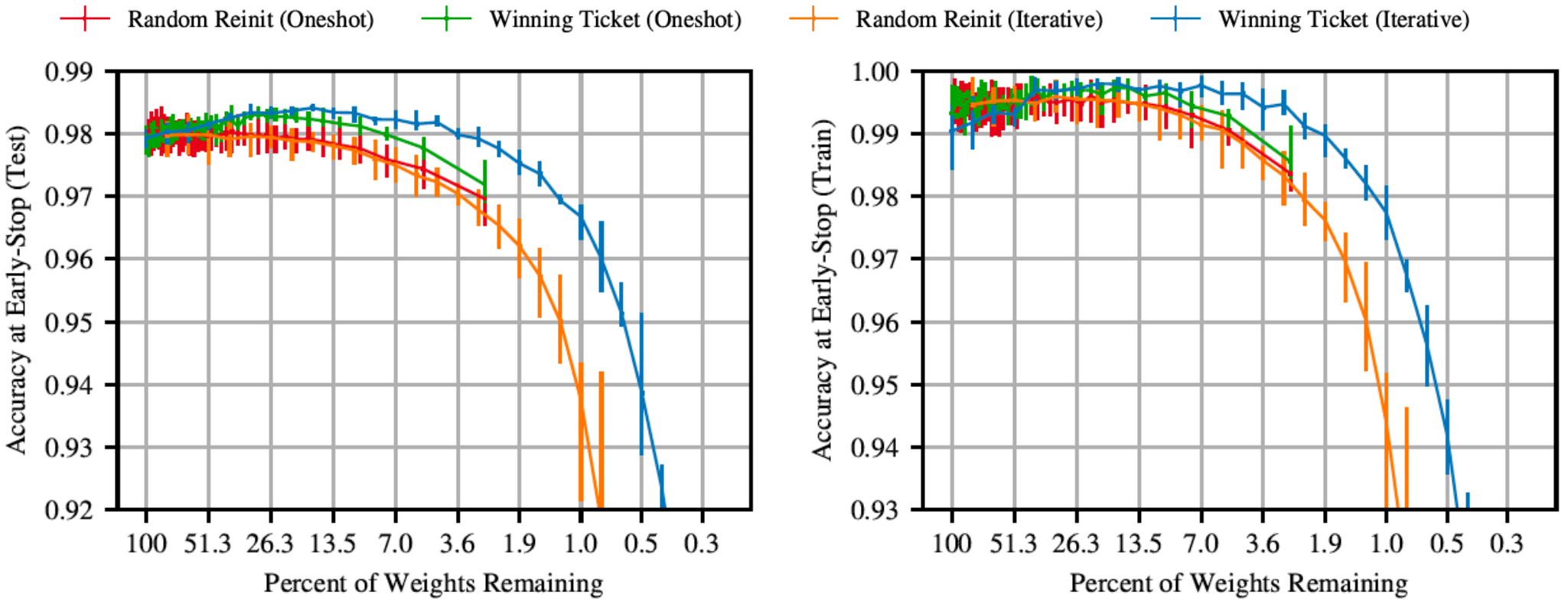
With one-shot pruning, winning ticket shows a trend of faster convergence and higher test accuracy than random re-initialization.



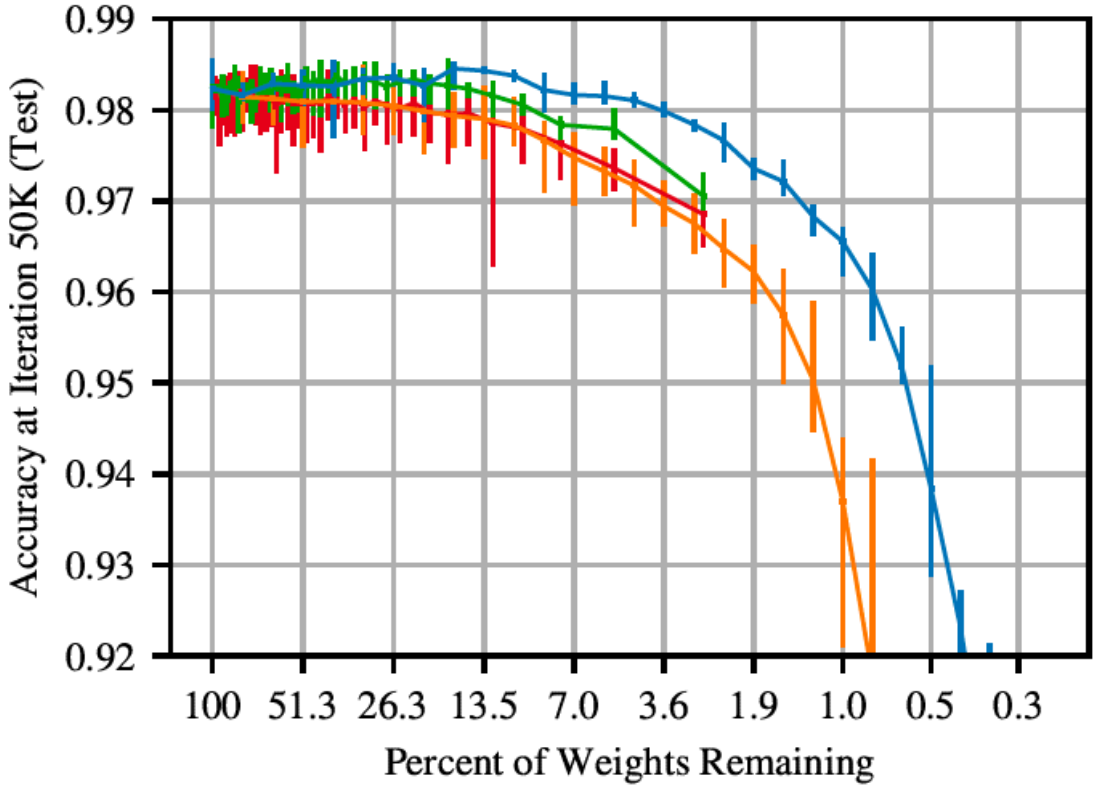
Iterative Pruning

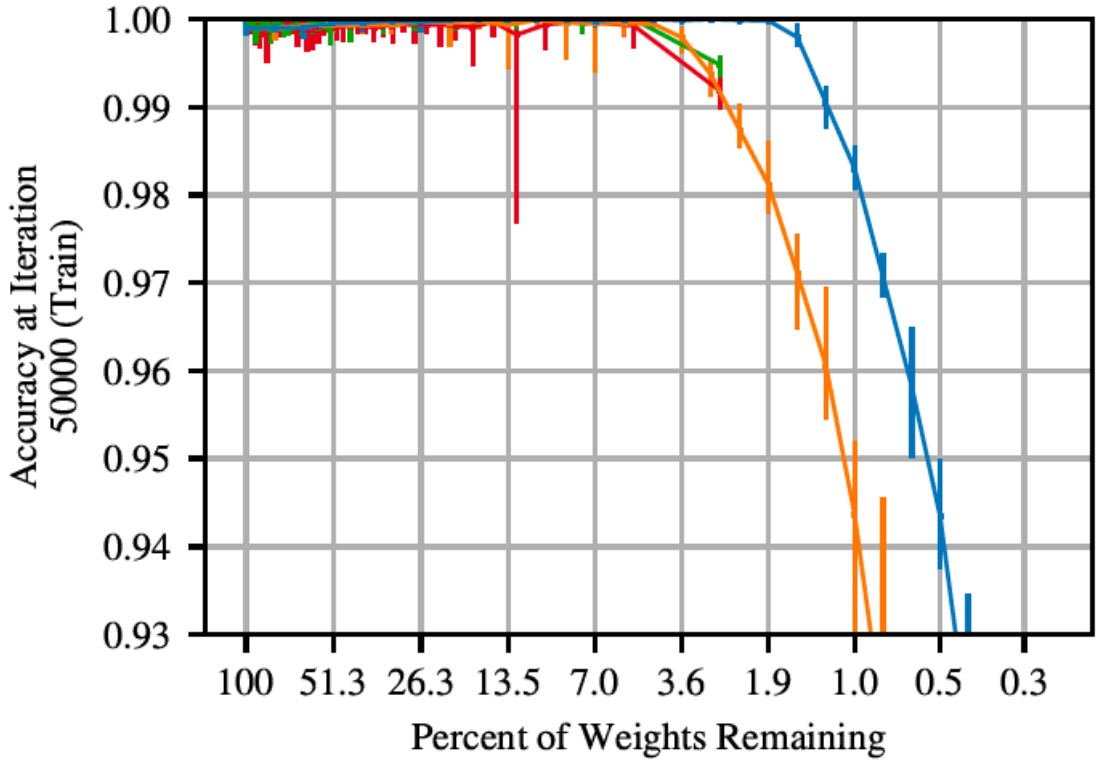


Iterative Pruning



Iterative Pruning - 50,000 iterations





Winning tickets for Convolutional Networks

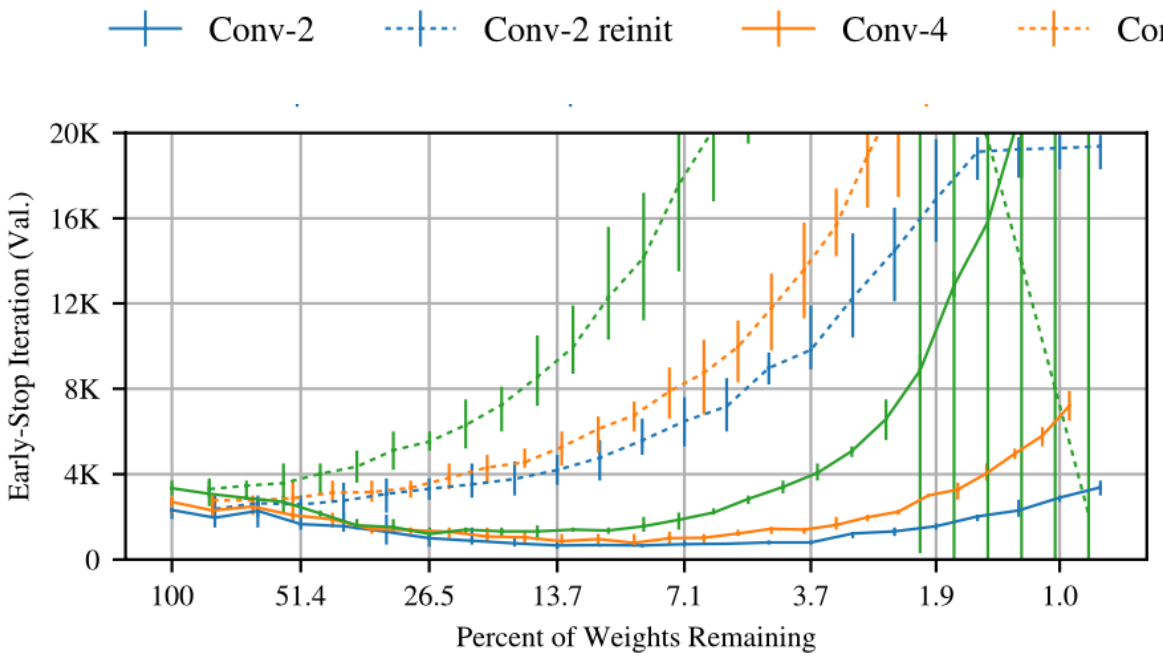
Winning ticket in Conv Nets

Here we test the hypothesis on convolutional networks trained on CIFAR 10 dataset.

Experimental setup:

- Architecture (scaled-down variants of VGG)
  - Conv-2
  - Conv-4
  - Conv-6

Convergence and accuracy with iterative pruning



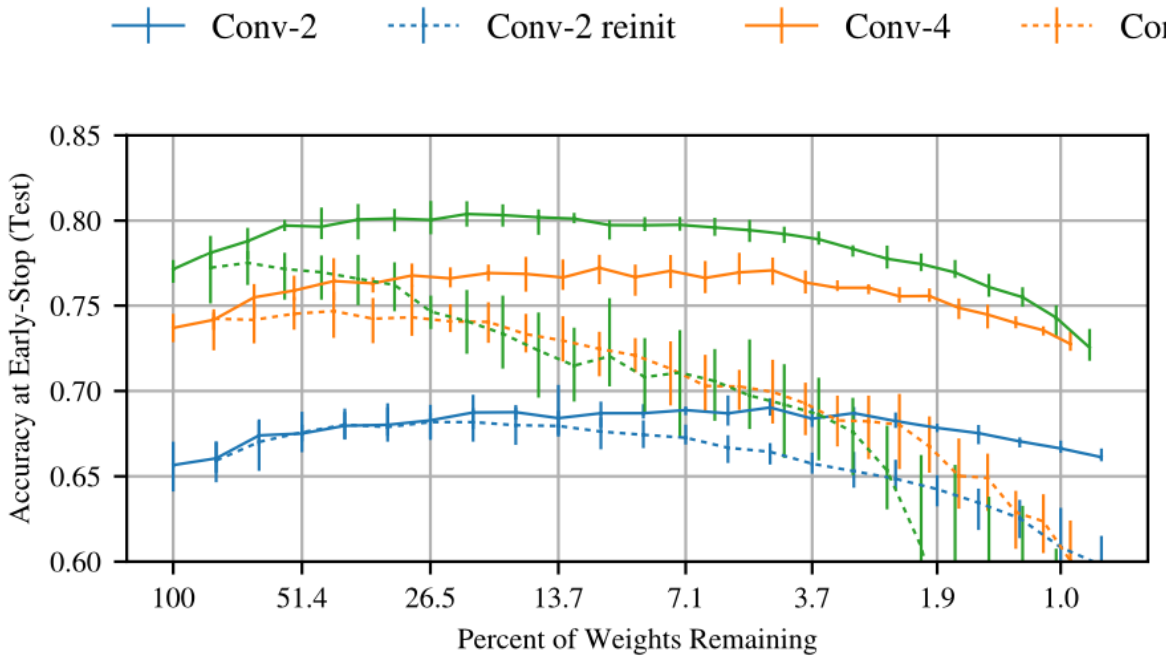
The same pattern from LeNet becomes more pronounced:

Winning tickets learns faster.

Winning tickets reach minimum validation loss at best

- 3.5x faster for Conv-2 (P_m = 8.8%),
- 3.5x for Conv-4 (P_m = 9.2%),
- 2.5x for Conv-6 (P_m = 15.1%).

Convergence and accuracy with iterative pruning



The same pattern from LeNet becomes more pronounced:

Winning tickets have a higher test accuracy.

Winning tickets improve the test accuracy

- 3.4% for Conv-2 (P_m = 4.6%)
- 3.5% for Conv-4 (P_m = 11.1%)
- 3.3% for Conv-6 (P_m = 26.4%)

All three networks remain above their original model's average test accuracy when P_m > 2%.

How about adding Dropout?

What is dropout?

Dropout is a strategy that improves the model's accuracy by randomly disabling a fraction of the units (i.e., randomly sampling a subnetwork) on each training iteration.

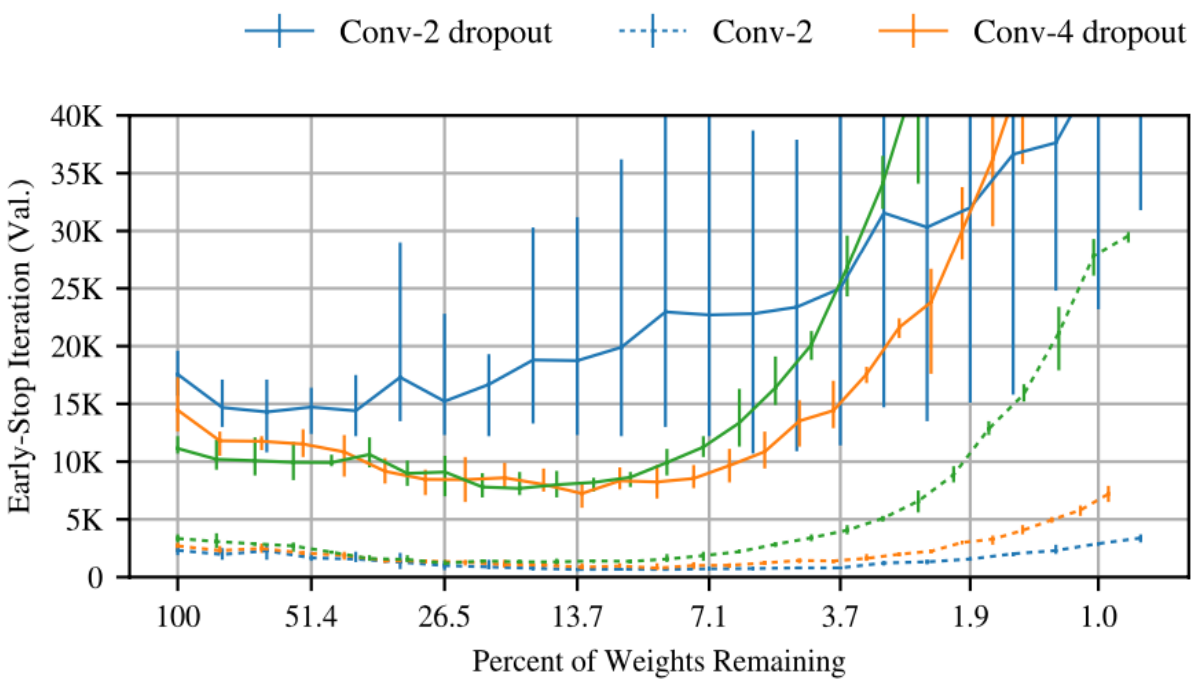
What is dropout?

Dropout is a strategy that improves the model's accuracy by randomly disabling a fraction of the units (i.e., randomly sampling a subnetwork) on each training iteration.

Since the lottery ticket hypothesis suggests that one of these subnetworks comprises a winning ticket, it is natural to ask:

How does dropout and pruning to find winning tickets interact?

Effect of dropout + pruning

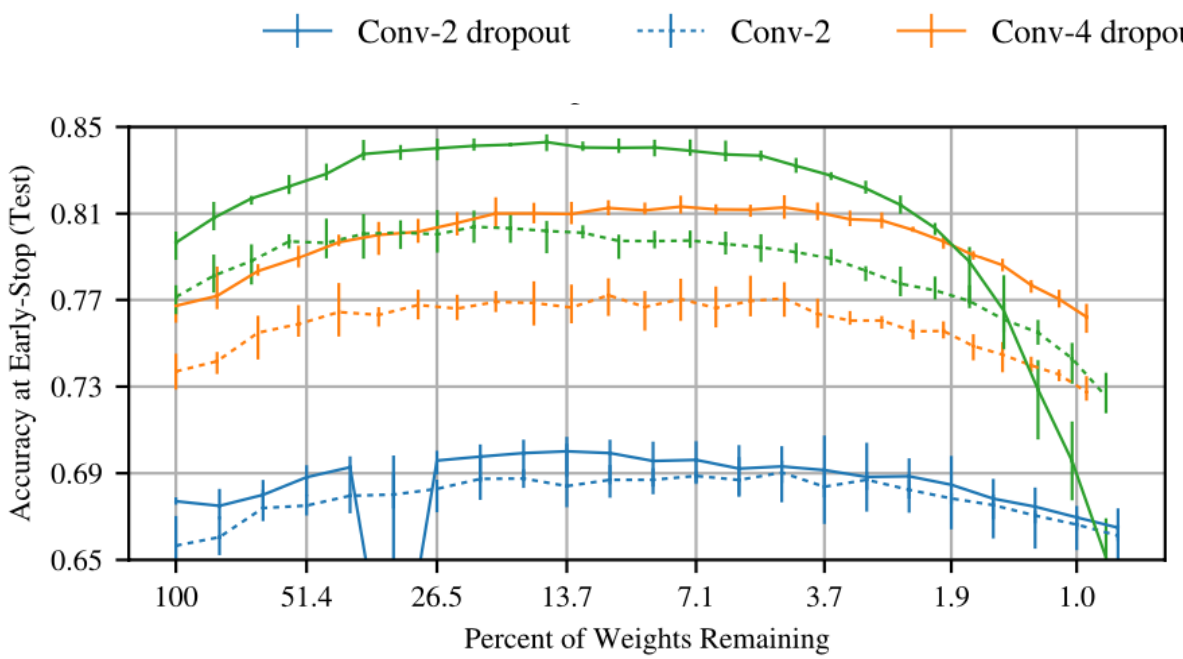


The experiment uses a dropout rate of 0.5

Models using dropout require more Iterations than models with only pruning.

However, learning becomes faster with iterative pruning as before.

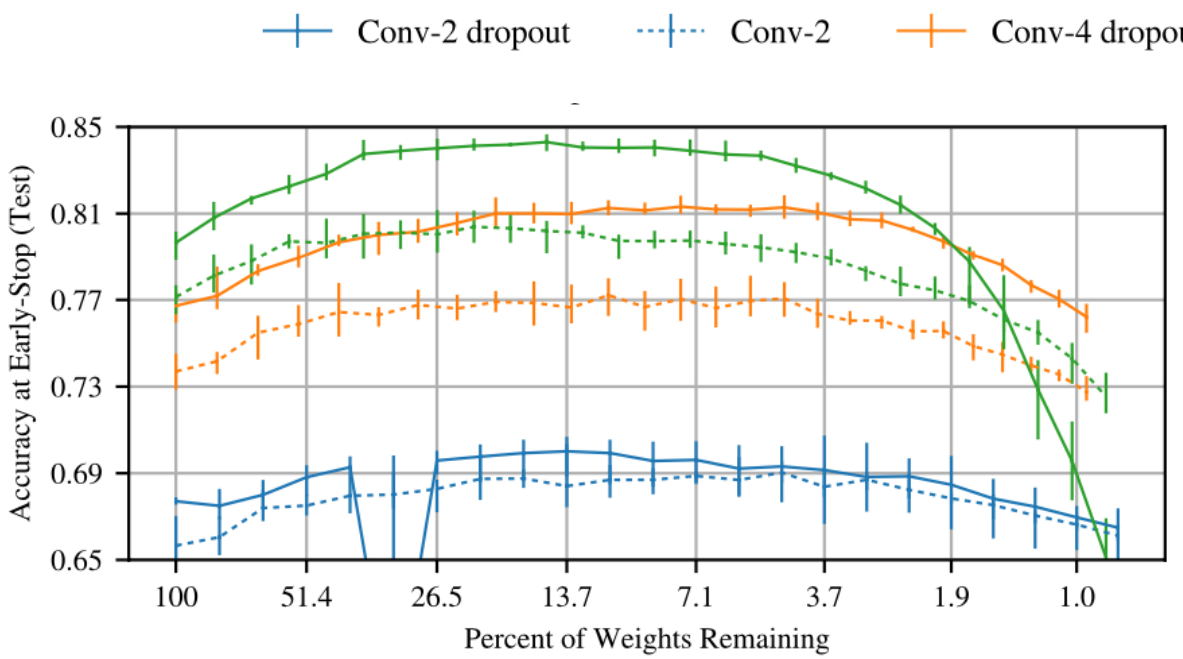
Effect of dropout + pruning



Dropout increases initial test accuracy (2.1, 3.0, and 2.4 % on average for Conv-2, Conv-4, and Conv-6)

Iterative pruning increases it further (up to an additional 2.3, 4.6, and 4.7 % on average).

Effect of dropout + pruning



Dropout increases initial test accuracy (2.1, 3.0, and 2.4 % on average for Conv-2, Conv-4, and Conv-6)

Iterative pruning increases it further (up to an additional 2.3, 4.6, and 4.7 % on average).

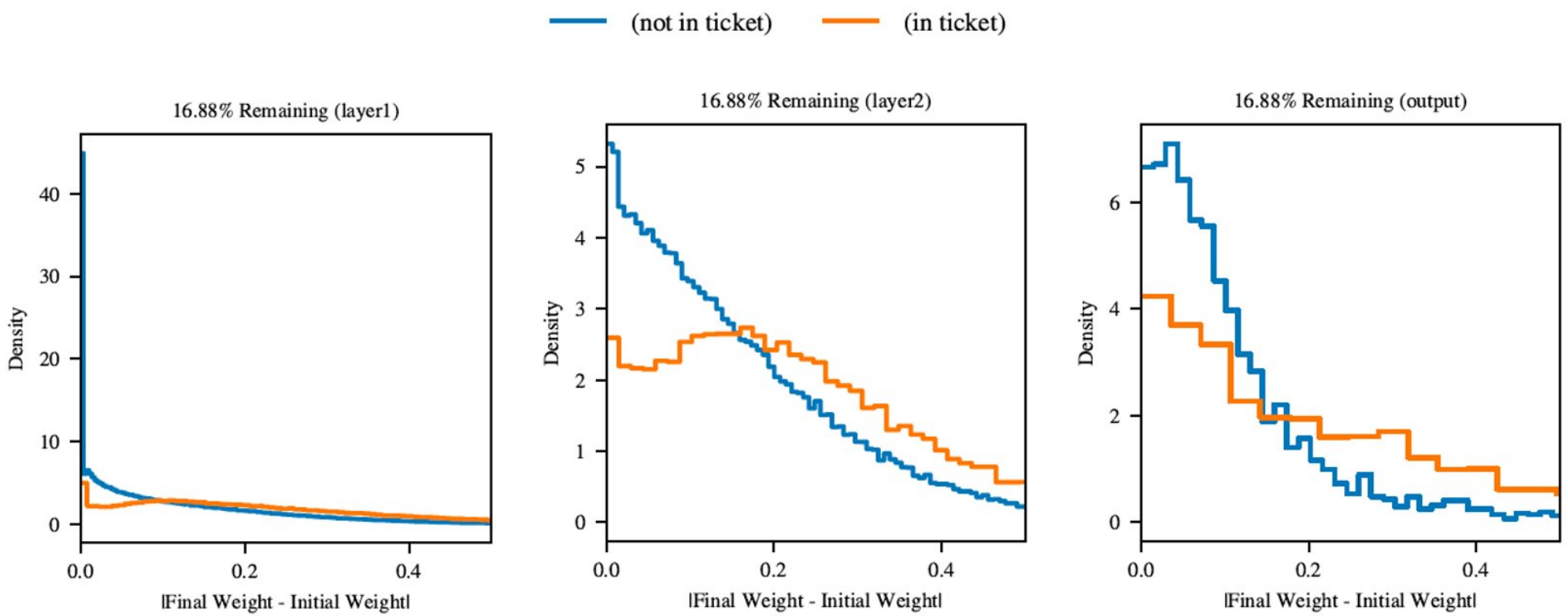
These improvements suggest that the iterative pruning strategy interacts with dropout in a **complementary** way when finding winning tickets.

COMPARING INITIAL AND FINAL WEIGHTS IN WINNING TICKETS?

Comparing initial and final weights in winning tickets?

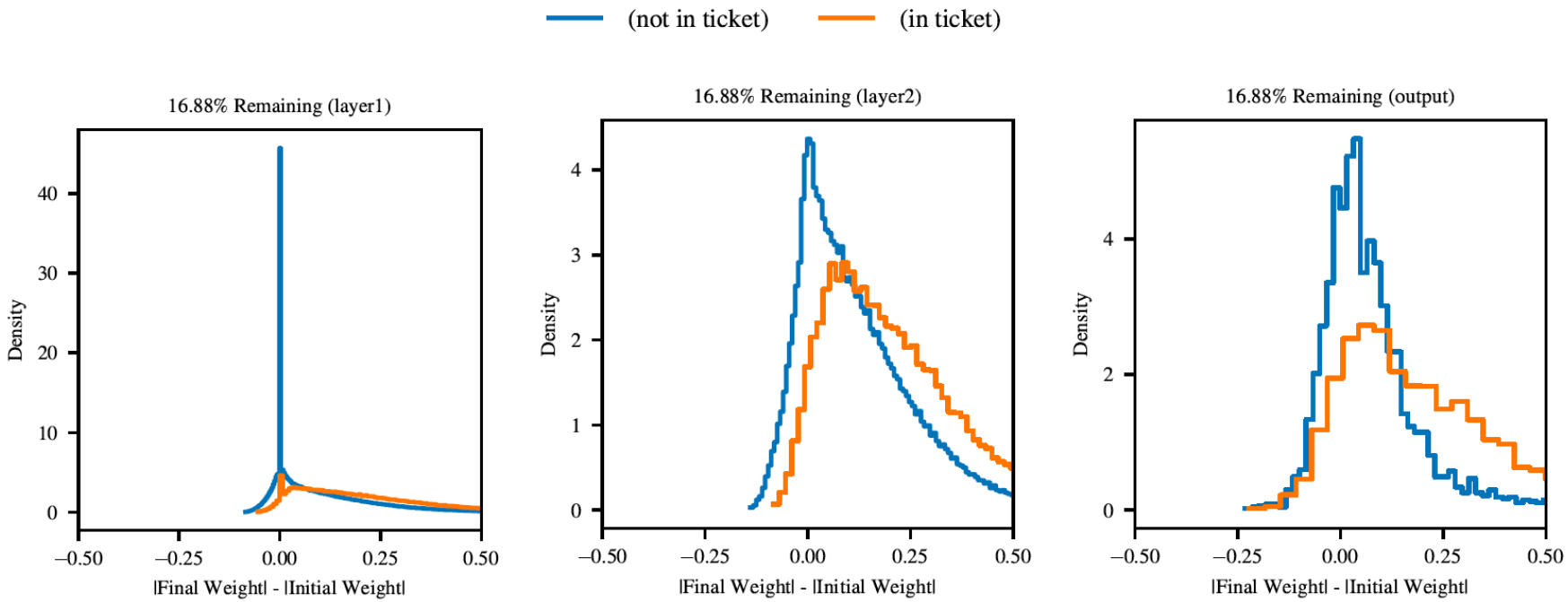
- One possible reason for success of winning tickets is that they already happen to be close to the optimum that gradient descent eventually finds, meaning that winning ticket weights should change by a smaller amount than the rest of the network.

Comparing initial and final weights in winning tickets?



Comparing initial and final weights in winning tickets?

Winning ticket weights are more likely to increase in magnitude (that is, move away from 0) than are weights that do not participate in the eventual winning ticket.



Comparing initial and final weights in winning tickets?

***Conclusion**: Winning tickets are well placed in the optimization landscape for gradient descent to optimize productively, meaning that winning ticket weights should change by a larger amount than the rest of the network.*

Importance of winning ticket structure

- The initialization that gives rise to a winning ticket is arranged in a particular sparse architecture.

- The paper uncover winning tickets through heavy use of training data and hypothesize that the structure of winning tickets encodes an inductive bias customized to the learning task at hand.

Improved generalization of winning tickets

- The paper shows that the winning tickets that generalize better, exceeding the test accuracy of the original network while matching its training accuracy.
- Test accuracy increases and then decreases as the network is pruned where the original, overparameterized model has too much complexity (perhaps. overfitting) and the extremely pruned model has too little.
- The conventional view of the relationship between compression and generalization is that compact hypotheses can better generalize.
- The lottery ticket hypothesis offers a complementary perspective on this relationship—that larger networks might explicitly contain simpler representations.

Why do lottery ticket hypothesis?

1. ***Improve training performance.***
Since winning tickets can be trained from the start in isolation, a hope is that we can design training schemes that search for winning tickets and prune as early as possible.

2. Design better networks

Winning tickets reveal combinations of sparse architectures and initializations that are particularly adept at learning. We can take inspiration from winning tickets to design new architectures and initialization schemes with the same properties that are conducive to learning. We may even be able to transfer winning tickets discovered for one task to many others.

3. Improve our theoretical understanding of neural networks.

We can study why randomly-initialized feed-forward networks seem to contain winning tickets and potential implications for theoretical study of optimization and generalization.

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

Thanks

Quiz questions

Question 1

What are the potential benefits of finding a winning ticket?

- **Achieving faster training (Correct)**
- **Winning tickets allow for better generalization (Correct)**
- The pruned architecture and weights of a model's winning ticket are universal in nature and can be applied to any setting.
- **Winning tickets occupy less storage space. (Correct)**

Question 2

Using dropout with iterative pruning while finding winning ticket leads to

- Lower early-stop iterations than without dropout.
- **Higher early-stop iterations than without dropout. (Correct)**
- **Better early-stop test accuracy than without dropout. (Correct)**
- Worse early-stop test accuracy than without dropout.

Question 3

For finding winning tickets, which of the following statements are correct?

- **One-shot pruning is faster than iterative pruning (Correct)**
- Iterative pruning is faster than one-shot pruning
- One-shot pruning results in smaller possible winning tickets without impacting accuracy.
- **Iterative pruning results in smaller possible winning tickets without impacting accuracy. (Correct)**

Extra Slides

Early-stopping Criteria