Lottery ticket hypothesis

By : Grishma Gupta, Lokit Paras

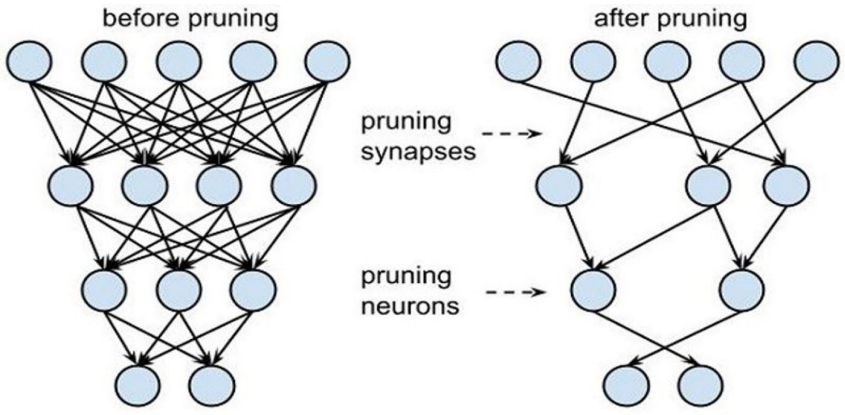
1. Motivation

Deep learning models have shown promising results in many domains. However, such models often have millions of parameters. The deep learning models face the following common issues :

- The models with large numbers of parameters have extremely long training periods (often days or weeks).
- The deep learning models have longer inference time
- Such models also need higher operational memory and computing requirements.
- This can lead to increased storage requirements for deployed models.

2. Network pruning

It is a 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.As we can see in the figure below, the number of parameters are drastically reduced after pruning.



3. Advantages

The deep learning model is large in size, needs more space to store the model, and requires more energy to deploy this model. Therefore, network pruning can be really helpful for the following reasons.

- Models will be smaller in size after pruning.
- Model will be more memory-efficient.
- Model will be more power-efficient.
- Models will be faster at inference with minimal loss in accuracy.

4. Types of Network Pruning

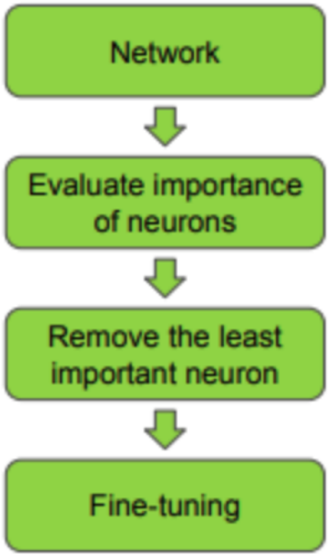
The paper discusses about two basic types of pruning:

- **<u>One-Shot Pruning</u>**

  The network connections in this type of pruning 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

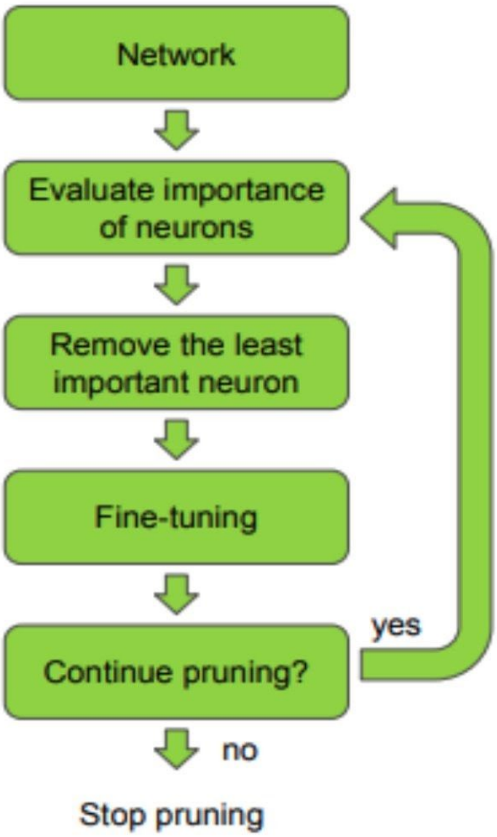


- **<u>Iterative pruning</u>**

  The network pruning is done partially through multiple iterations.

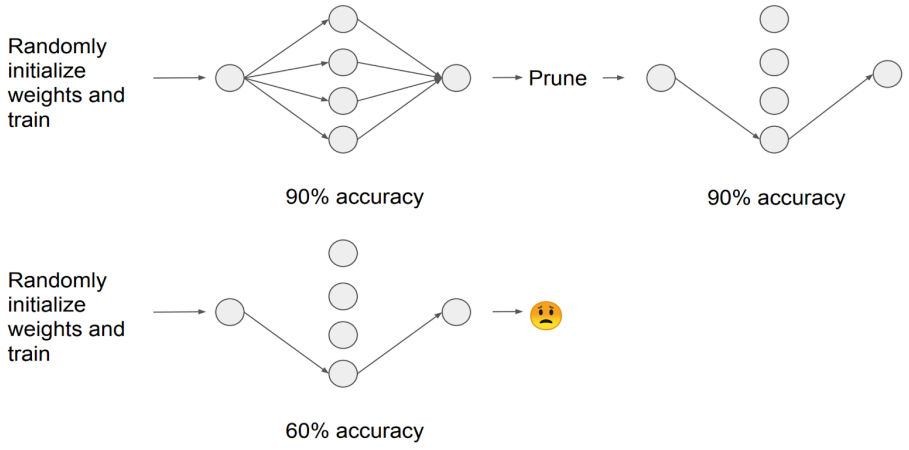
  Steps:

  - 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.



5. Is the pruned architecture enough?

We are trying to understand if the pruned architecture is enough for reducing parameters and maintaining accuracy. Here in the figure below we can see an architecture which is pruned to 90% but when the model is re-initialized with different weights the accuracy drops to 60%. This shows us that the ***<u>pruned architecture is not enough</u>***, and initialization of weights has a very important role.

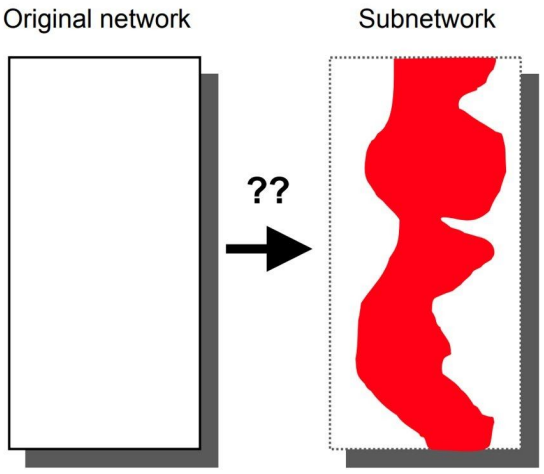


6. Lottery ticket hypothesis

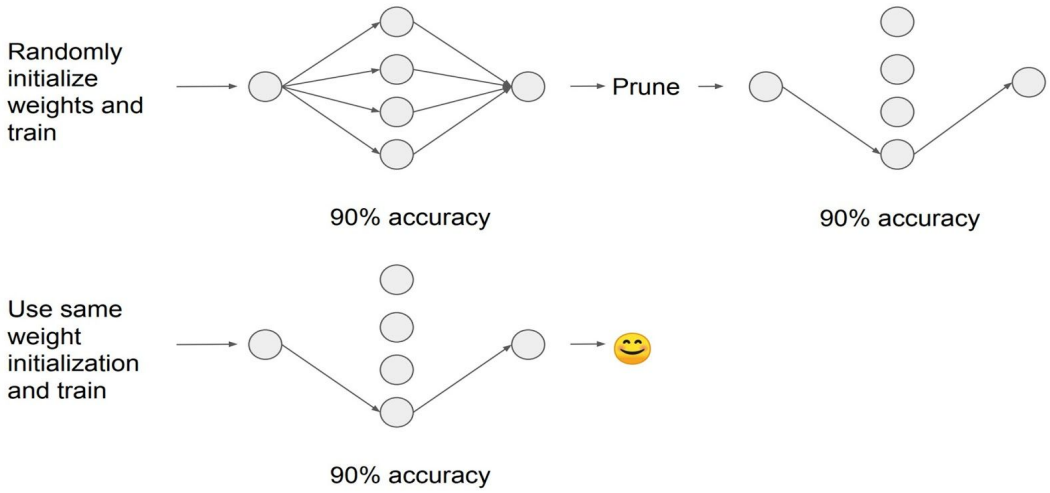
A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

Lottery ticket hypothesis aims to find a subnetwork which has following properties:

- Is there a subnetwork with better results
- Shorter training time
- Notably fewer parameters



The figure below shows how the neural network model when re-initialized with same weights after pruning maintains the 90% accuracy even with parameters reduced.



6.1 Feed-forward neural network

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

6.2 <u>Subnetwork with lottery ticket hypothesis</u>

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

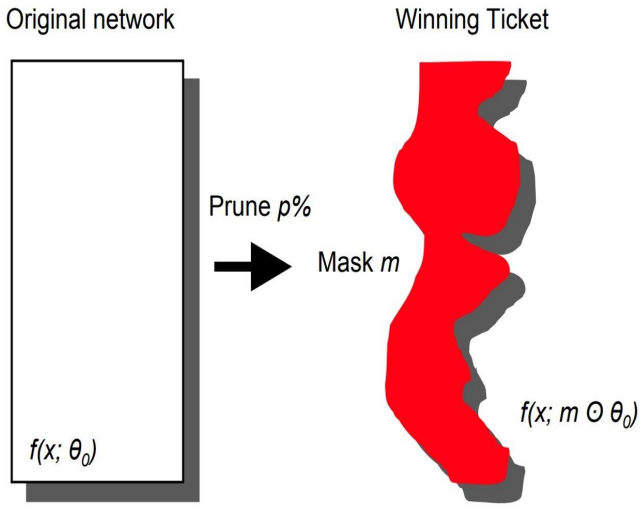
6.3 <u>The lottery ticket hypothesis predicts that</u>

$\exists$ m (mask) for which

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

7. 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*



These winning tickets give:

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

7.1 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 $f(x; m \odot \theta_0)$.

7.2 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 p1/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 $f(x; m \odot \theta_0)$

8. Experimental with fully-connected

The paper conducts various experiments to prove this hypothesis. To test their hypothesis, the authors applied it to fully-connected networks trained on MNIST. The architecture used for experiments is Lenet-300-100.

8.1 Pruning heuristic:

- Remove a percentage of weights layer-wise,
- Magnitude based (remove lower magnitude)

8.2 Pruning Rate and Sparsity:

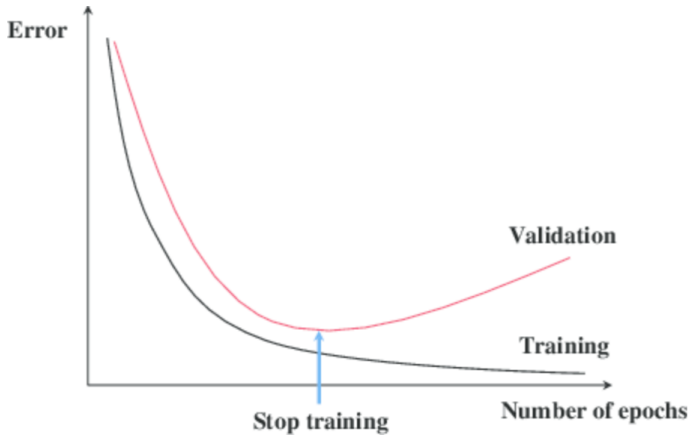
p% is the Pruning Rate

P_m is the Sparsity of the pruned network (mask)

E.g. P_m = 25% when p% = 75% of weights are pruned

8.3 Early stop

To prevent the model from overfitting, the authors use early stopping as the convergence criteria. The iteration for early stopping is decided on the basis of validation loss.



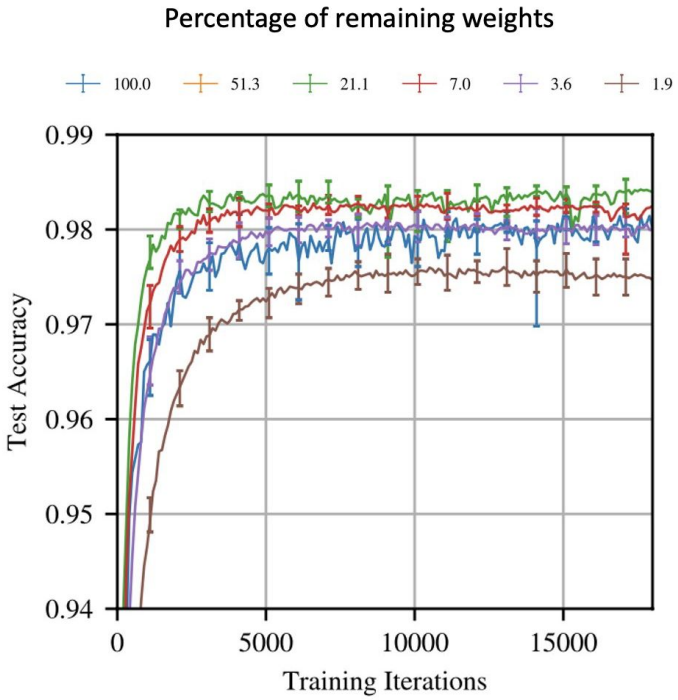
Validation and test loss follow a pattern where they decrease early in the training process, reach a minimum, and then begin to increase as the model overfits to the training data.

9. Results

9.1 Effect of pruning on accuracy

The below experiment shows that a winning ticket comprising 51.3% of the weights (i.e., Pm = 51.3%) reaches higher test accuracy faster than the original network but slower than when Pm = 21.1%. When Pm = 3.6%, a winning ticket regresses to the performance of the original network.

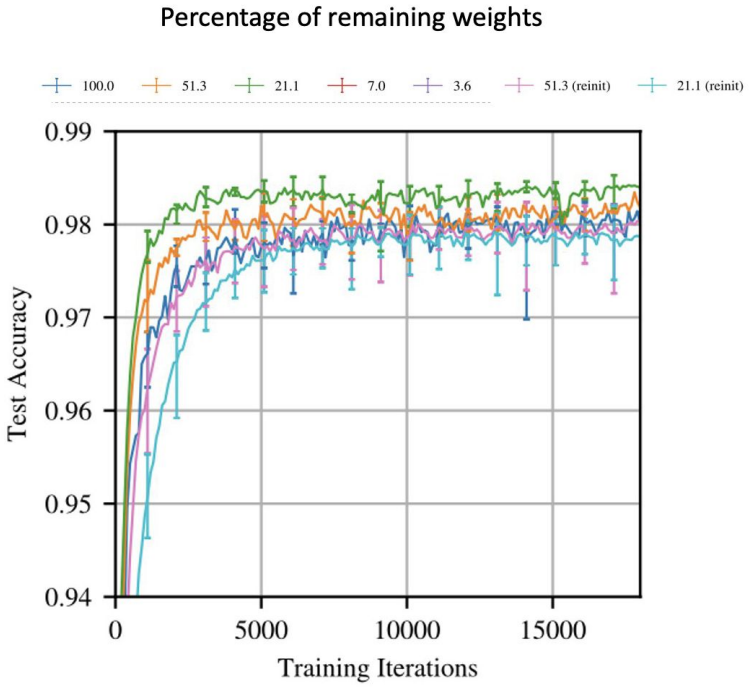
The experiment below shows us that the winning tickets we find learn faster than the original network and reach a higher test accuracy than original network. But Beyond a certain percentage, pruning starts reducing the model's accuracy.



9.2 Pruning + Re-initialization

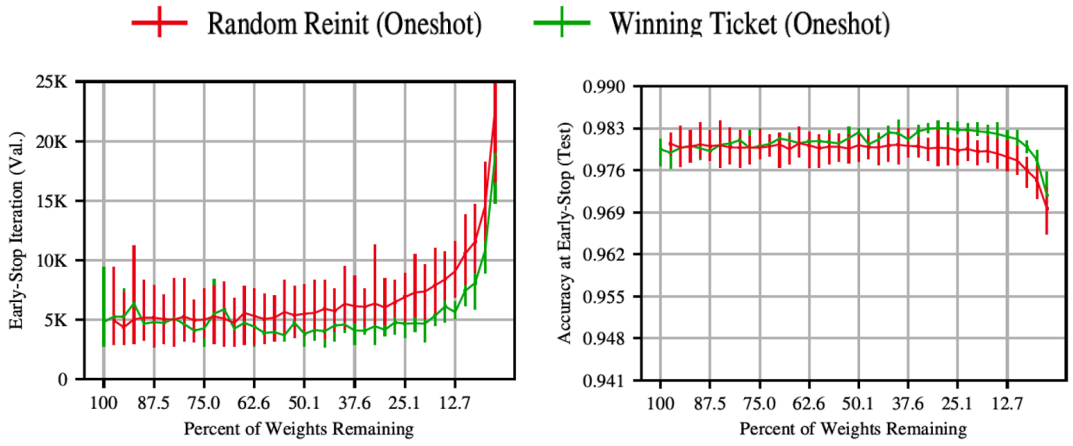
To measure the importance of a winning ticket's initialization, we retain the structure of a winning ticket (i.e. the mask m) but randomly sample a new initialization $\theta_{0.}$

From the experiments unlike winning tickets, the reinitialized networks learn increasingly slower than the original network and lose test accuracy after little pruning. The experiments shows that the initialization is crucial for the efficacy of a winning ticket



9.3 One-shot pruning to find winning tickets

Although iterative pruning extracts smaller winning tickets, repeated training means they are costly to find. One-shot pruning makes it possible to identify winning tickets without this repeated training.

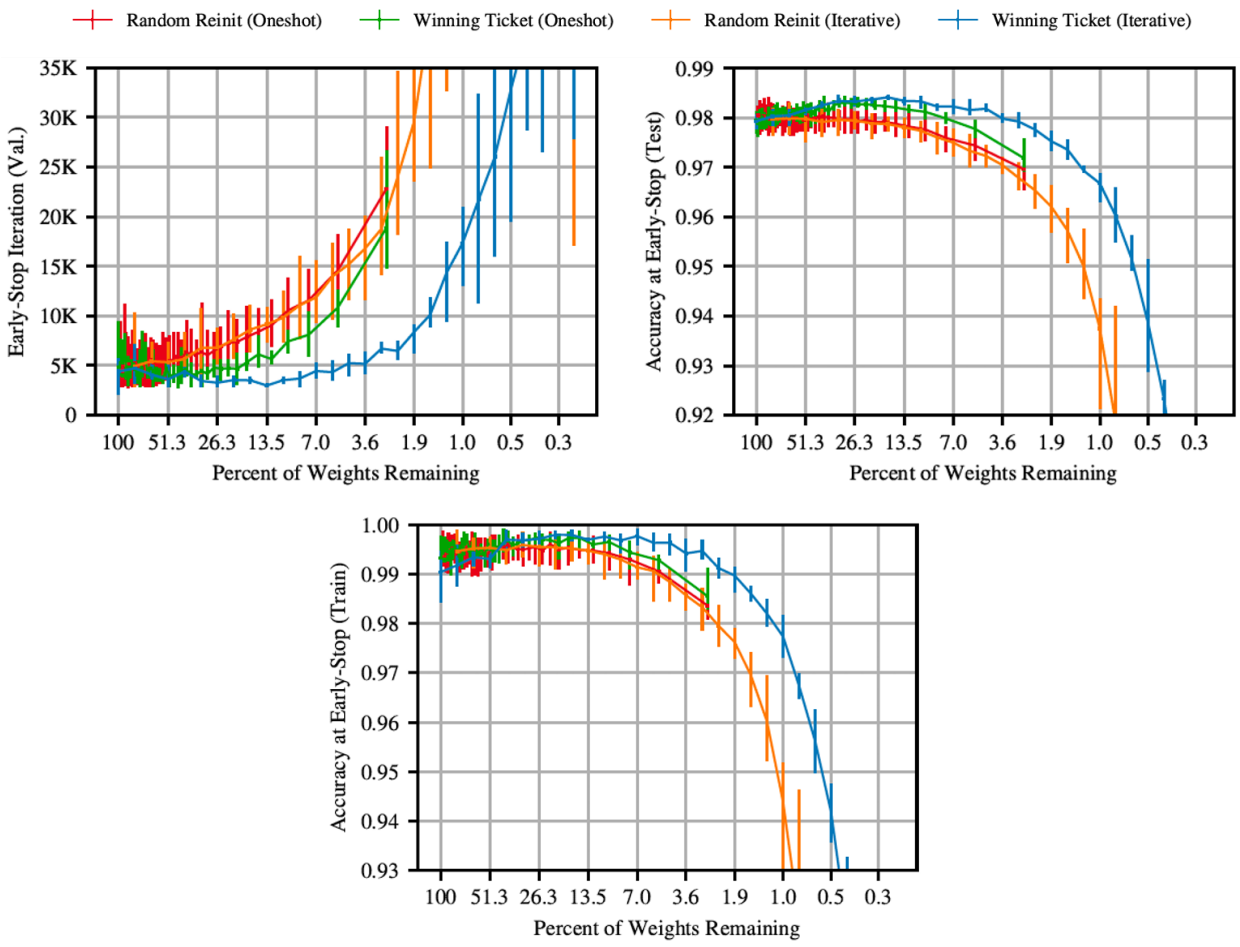


With 67:5% > P_m > 17:6%, the average winning tickets reach minimum validation accuracy earlier than the original network.

With 95:0% > P_m> 5:17%, test accuracy for the winning tickets is higher than the original network

This highlights that winning tickets can outperform original network while having a smaller network size. However, if we randomly re-initialize the winning ticket, we lose the leverage as the performance drops.

Convergence and Accuracy with Iterative Pruning



With iterative pruning, we find that the winning tickets learn faster as they are pruned. However, upon random re-initialization, they learn progressively slower.

Thus the experiment supports the lottery ticket hypothesis' emphasis on initialization:

The original initialization withstands and benefits from pruning, while the random reinitialization's performance immediately suffers and diminishes steadily.

Comparing the graphs for training and testing accuracy, we can see that even with training accuracy of ~100%, the testing accuracy of the winning tickets increases with some pruning. However, if we randomly re-initialize the winning ticket, there is no such improvement in accuracy with pruning.

Thus, we can say that the winning tickets generalize substantially better than when randomly reinitialized.

10. Winning tickets for Convolutional Networks

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

Experimental setup:

The authors use a scaled down version of the VGG network. There are three variants:

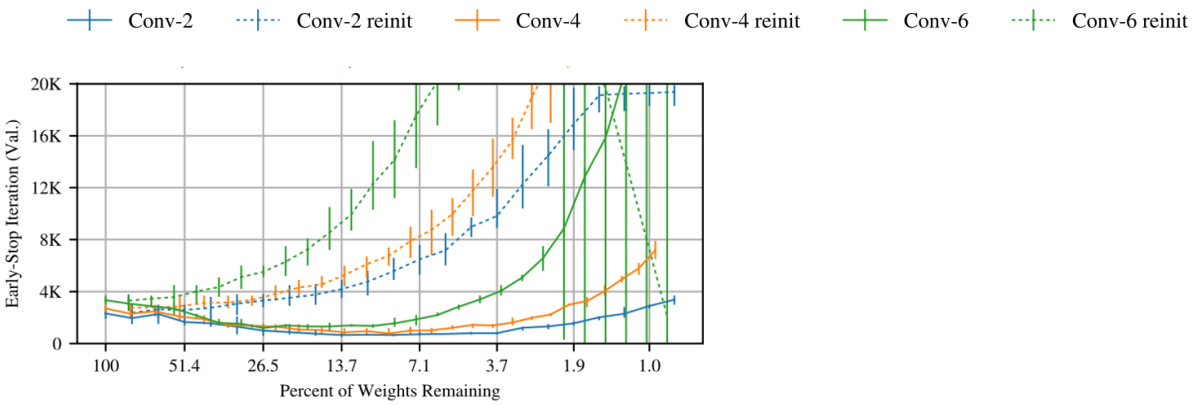
◦ Conv-2: 2 convolutional layers
◦ Conv-4: 4 convolutional layers
◦ Conv-6: 6 convolutional layers

Convergence and accuracy with iterative pruning

Here we see a pattern similar to the results from the LeNet architect, only even more pronounced.

Observations

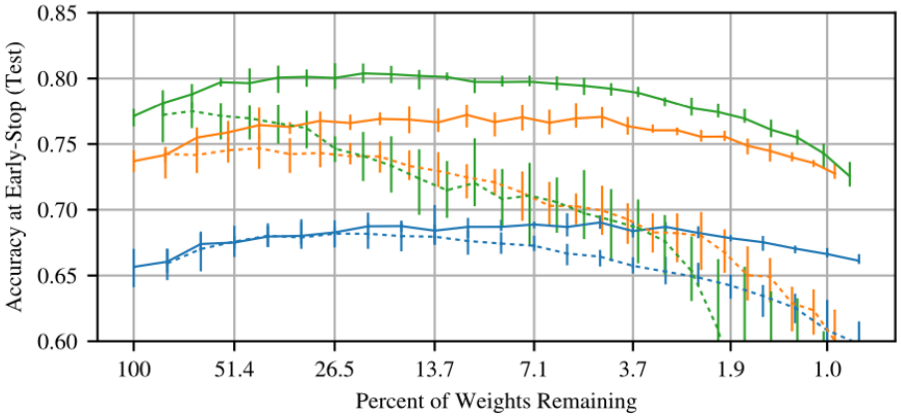
1. **Winning tickets learns faster.**



Winning tickets reach minimum validation loss at best

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

However, re-initialization of the winning ticket leads to increase in the required number of iterations.



2. **Winning tickets have a higher test accuracy**

Winning tickets show an improvement in test accuracy

- 3.4% for Conv-2 (Pm = 4.6%)
- 3.5% for Conv-4 (Pm = 11.1%)

- 3.3% for Conv-6 (Pm = 26.4%)

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

11. Using dropout to find winning ticket

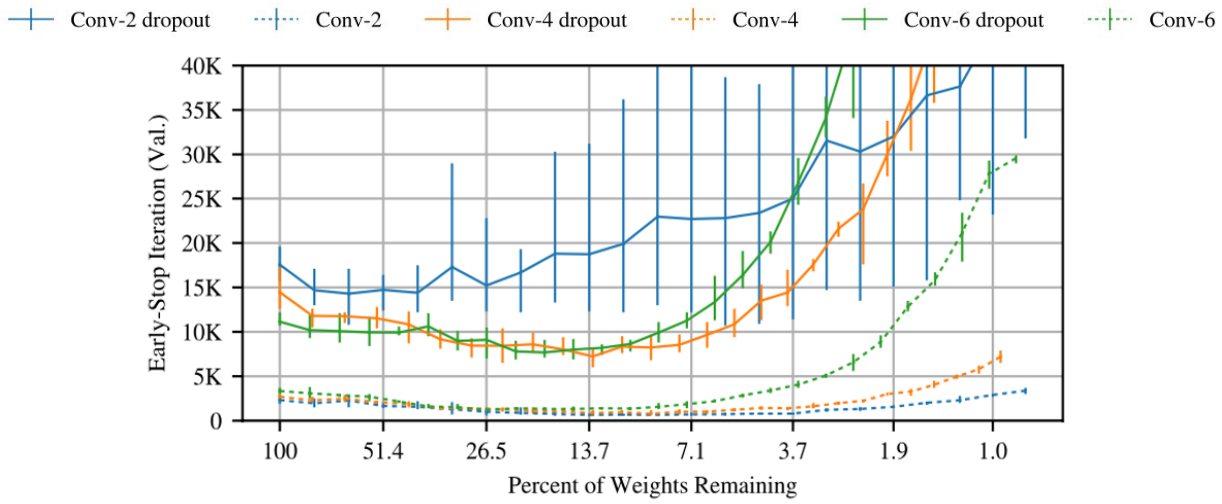
Dropout is a strategy that improves the model's accuracy by randomly disabling a fraction of the units (i.e., randomly sampling a subnetwork) on each training iteration.

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

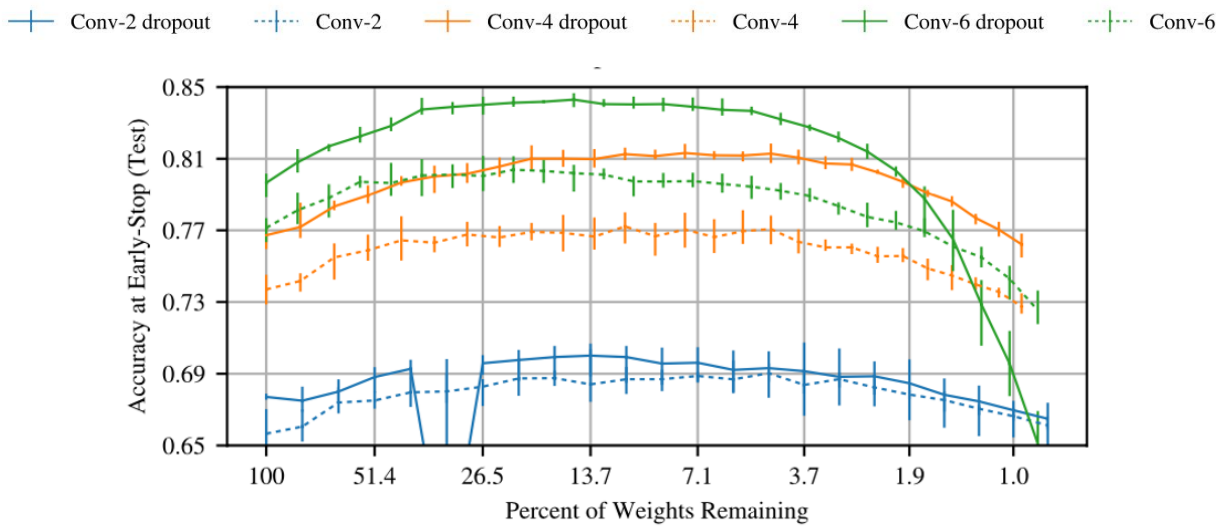
How does dropout and pruning to find winning tickets interact?

Effect of dropout

The experiment uses a dropout rate of 0.5



Models using dropout require more iterations than models with only pruning. However, as seen before, learning becomes faster in the initial rounds of iterative pruning.



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

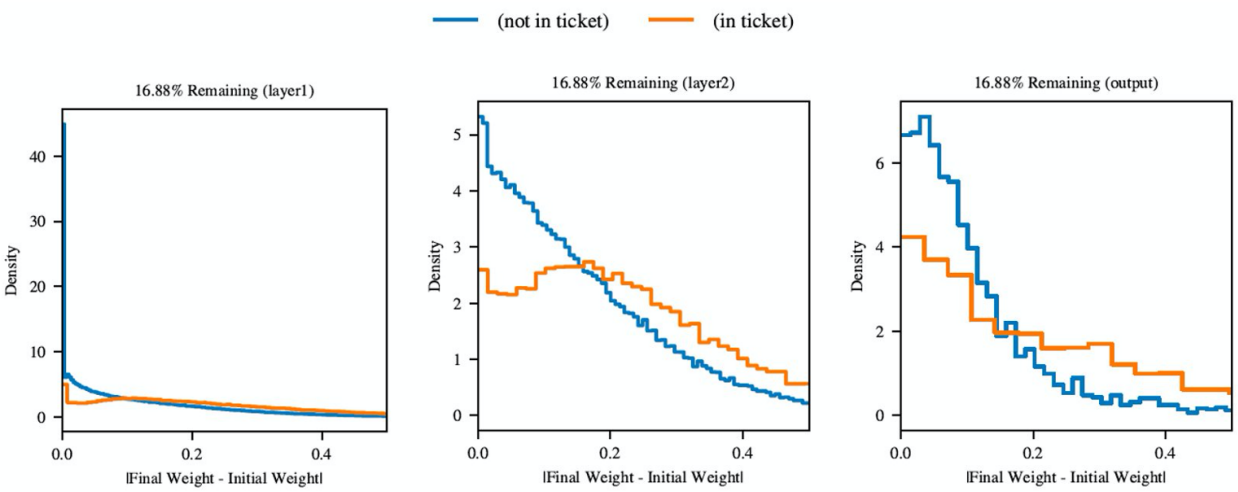
Iterative pruning increases it further (up to an additional 2.3, 4.6, and 4.7 % on average).

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

12. Closer analysis of winning ticket

Study of initial and final weights in winning tickets?

There is a possible intuition, that 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.



However, experimentally we find that 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.

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.

Winning tickets generalize better

- 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.

13. Why the 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.

14. Limitations

Iterative pruning is computationally intensive as it involves training a network multiple times per trial. This makes it hard to study the hypothesis for larger datasets like ImageNet.

Future work: We can try finding more efficient methods of finding winning tickets.

The current algorithm for finding winning tickets are not optimized for modern libraries or hardware