The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural networks

Overview

Md Shahbaz Siddiqi

Indian Institute of Science, Bangalore

Introduction

In 2019, Jonathan Frankle and Michael Carbin from CSAIL MIT lab presented a paper that was one of the two best paper submitted in ICLR that year.

As stated in the paper: *“A randomly-initialized, dense neural network contains a subnetwork which when trained in isolation — it can match the test accuracy of the original network after training for at most the same number of iterations.”*

- Through out their experiment their main hypothesis was to explain why it is easier to train dense neural network than their respective sparse counterpart. The reason is that dense neural network contain large number of “winning tickets”.

Motivation

- Pruning algorithms provide a link between overparameterized models and appropriately parameterized models. Thus, these algorithms may provide explanation for success of overparameterized models.

- From a more practical perspective, overparameterized networks require more storage capacity and are computationally more expensive than their pruned counterparts.

why don't we train a generalized prune network from start?

Even though one could generalize the pruned network, after reinitializing the weights and retraining, its was observed that there was a significant drop in accuracy.

Related Work and Initial discoveries

Pruning is only possible in neural network, because of **sparsity** of weights within the network. Sparsity has long been part of neuroscience research since the activity of neuron in brain is sparse in nature.

The first major paper advocating sparsity in neural networks dates back from 1990, written by LeCun et al. while working at AT&T Bell Laboratories. The paper was named “Optimal Brain Damage”, it introduced a then novel mechanism of pruning weight using second derivative of the objective function with respect to the parameters as a pruning metric to approximate the saliency.

In 2015 Han et al. published a paper proposing a three step method to prune neural networks:

- Train the Network
- Remove superfluous structures.
- Fine-tune the network

Winning Lottery Ticket

What is a winning lottery ticket?

- If a dense network can reach an accuracy **a** in **t** iterations, **W** being the total weights.

- A subnetwork of the dense network is a winning ticket if it can reach an accuracy of **a’** in **t’** iteration, with only **K** weights.

- **Here necessary condition is a’ >= a, t’ < t and K < W.**

Generating Winning ticket

The paper presents two pruning strategies which can find winning tickets:

One-shot pruning

- Randomly initialize a neural network with weights W.
- Train the network.
- Set p% of weights with the lowest magnitude from each layer to 0 (this is the pruning).

Iterative Pruning

- Iterative pruning just iteratively applies the steps of one-shot pruning.
- Reset the pruned network weights to their original random initializations before training begun that is W.

The authors found that iterative pruning yielded smaller pruned subnetworks than one-shot pruning. However there are two variants of iterative pruning continuous training and resetting.

How weight reacts to success of winning ticket

- If we consider the magnitude of the difference between initial and final weights, there are two possible rationale how weights can react to success of winning tickets.

- The weights are so initialized that they already happen to be close to the optimum that gradient descent eventually finds.

- Another possible rationale is that 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.

- In general, 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.

Experimentations performed

The authors experimented with MNIST and CIFAR-10 dataset. They choose to implement the method on fully connected architecture for MNIST dataset while Convolution Neural Network architecture for CIFAR-10 dataset.

They used several optimization strategy like SGD, Adam and momentum with techniques like dropout and weight decay.

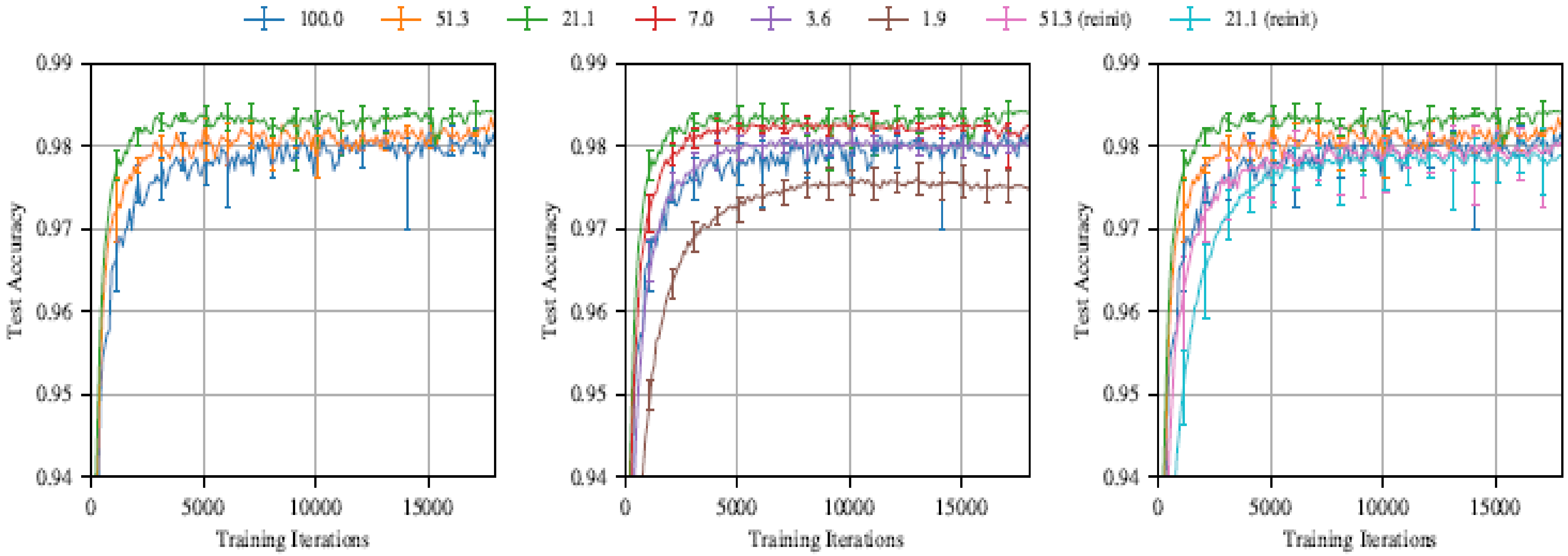
The author experimented with following architecture

- LeNet
- Conv-2, Conv-4, Conv-6 which are scaled down variant of VGG network
- ResNet 18
- VGG 19

- Early stopping criteria is consider as a proxy of how fast a network learns.
- For VGG 19 and ResNet 18 different learning rate was used
- A warm-up weight equivalent to initial weight was used along low learning rate in VGG 19.

Results and Observations

LeNet



Pruned network achieves a higher test accuracy and also learn faster. However, if we prune the network beyond a certain percentage, the learning slow down and test accuracy reduces.

Key Take Away

- When randomly initialized the sub networks or winning tickets obtained from the dense network seems to learn slowly and achieves lower test accuracy. This implies that initialization is important for success.

- Test accuracy increases and then decreases as we prune, where the original, overparameterized model has too much complexity (perhaps overfitting) and the extremely pruned model has too little.

- Winning tickets can reach accuracy equivalent to that of the original, unpruned network, but with significantly fewer parameters . This indicates that presence of a winning ticket is necessary or sufficient for SGD to optimize a neural network to a particular test accuracy.

Limitations

The results have helped us understand a little bit more about how deep neural networks learn. Unfortunately the iterative pruning strategy they propose does not provide significant practical benefits.

- Iterative pruning is computationally expensive as it involves training a network multiple times per trial. This made it hard for the authors to study larger datasets like ImageNet.

- By reducing the number of parameters by 80–90%, the storage requirements for the network are decreased. The weight matrices still have the same dimensions but are just more sparse.

Different Point of View

What could be the outcome if pruning is done in a structured manner?

It was observed (Lui et al.) that if we conduct this experiment using all major structured pruning approaches retraining the network instead of fine-tuning the subnetwork tends to achieve an equivalent accuracy as that of dense counterpart, even if **randomly reinitialized**.

- **What is the relevance of the learnt Weights?**

In one of the experiments performed (Zhou et al), it was observed that there is no relevance of magnitude of learnt weights.

If we break the weights into two parts sign (+ or -) and magnitude, reinitializing the pruned network to a fixed constant magnitude of weight keeping same sign as the original initialization would help subnetwork to outperform the accuracy of dense network.

Proposed Improvements

One of key implication as discussed in the paper is:

- Design better networks
- Improve our theoretical understanding of neural network.

One can perform following experiments in order to further improve on the implications discussed:

- If we use a constant weight that is a single predefined parameter to learn the new weights during training. This will allow us to understand the weight distribution and weight changes occurring in the network better, because there is no randomness in initialization.
- When dealing with dense neural network with multiple layer, one can sequentially re- initialize the layers as a pruning mechanism, in order to understand the impact a layer have on the outcome.
- An experiment conducted by Zhang et al. indicates than in a dense multi-layer network few layers have more role to play in achieving accuracy than others.

Final Review

The motivation behind analyzing this literature was to understand how we can create an appropriately parameterized network since over parameterized dense network is computationally complex and use huge resources.

The lighter, robust and less parametrized network is essential as machine learned model are rapidly pushed into embedded systems. With the advancement in Internet of Things low parameterized models that can obtain a test accuracy equivalent to that of dense neural network, apart from getting trained faster is desirable.

This paper has enable researchers to leverage further on sparsity aspect of network and come up with better pruning algorithms.

Some areas where research could further be carried out using this literature as starting point are :

- Finding the winning ticket without training a dense model.
- Training a sparse network to learn the optimal set of connections so we can have sparse models during training as well.
- Elimination of any possibility of overfitting the subnetwork.

Thank You