Lottery Ticket Initialization: Exploring the Uniqueness of Mask Topology and Weight Distribution

Anonymous Authors[1]

Abstract

Previous works have hypothesized and empirically shown the existence of so-called "lottery ticket" subnetworks within deep neural networks (Frankle & Carbin, 2018). These subnetworks have less than 20% the parameter size of the original model yet perform better with regards to training convergence, inference time, and test accuracy. The existence of such networks begs further exploration on their unintuitively high performance. In this paper, we look at two specific parameters of the lottery ticket: the distribution of non-zero weights after a lottery ticket has been found and the specific mask of the lottery ticket. With regards to these parameters, we test four separate network architectures: the original full network, the pruned lottery ticket network, the lottery ticket reinitialized with the empirical lottery ticket weight distribution, and a network with parameter size equal to the lottery ticket initialized with the empirical lottery ticket weight distribution. These architectures were experimentally tested on a fully-connected network and convolutional network for MNIST and CIFAR10. The test accuracy of these new architectures were worse than both the full network and the lottery ticket network, and the number of epochs to train to that accuracy increased significantly. At a high level, this shows that the distribution of weights and topology of the pruning mask do not provide adequate information to create a performance-equivalent subnetwork to the original network. Our findings imply that generating pruned networks highly depends on the initialization of the original network, both with regards to the exact pruning topology as well as the placement of model weights.

1. Introduction

In an ideal machine learning setting, the ability to remove parameters from a neural network whilst maintaining accuracy and other valuable metrics would be groundbreaking (Frankle & Carbin, 2018). The smaller network could achieve the same test accuracy, convergence speed, and train loss as the full network, but could potentially provide various benefits in model storage space, inference time, and interpretability. Efforts in this idea (*pruning*) (Cun et al., 1990) have shown that though it is possible to achieve such results, even when pruning to below 20% of the initial model capacity, in practice this is quite hard to do. Specifically, though there might exist such subnetworks, finding and training these networks is quite difficult (Li et al., 2016).

Recent work (Frankle & Carbin, 2018) has found a novel way to construct a specific subnetwork, labeled the lottery ticket, that achieves the same performance whilst maintaining the training speed of the original network. The lottery ticket subnetwork is created by first finding a mask that indicates which weights to include in the subnetwork, then applying that mask onto the original initialized weights of the network. Intuitively, this network can be thought of as a part of the original network that contains a 'good' initialization. This subnetwork won the initalization 'lottery', hence the term lottery ticket. Empirically, these lottery ticket subnetworks have shown equal performance to the original network, even when over 80% of network parameters are pruned.

When compared to other networks of the same parameter size, the lottery ticket performs significantly better (Frankle & Carbin, 2018). Seeing that the only difference between the lottery ticket and these comparison networks is their initialization, these results imply that lottery ticket initialization is heavily reliant on the original non-pruned network. However, this begs further exploration on the uniqueness of these lottery tickets: what is so special about their initialization? We notice that, through the pruning process, the non-zeroed weights of the lottery ticket induce a new distribution of values, and conjecture that this distribution could potentially provide meaningful initialization information for sparse subnetworks to converge quickly.

Intuitively, there is something special about this distribution of non-zeroed weights, as when subnetworks are initialized with standard random initialization, convergence results are poor. This paper explores in more depth how much information this distribution of weights carry. We first observe the distribution of weights that are non-zeroed, then see how

models initialized with this distribution perform. Through this process, we see to what extent the distribution of lottery ticket weights impacts how lottery tickets perform so well.

From our results, we find that exact topology as drawn from the pruned mask and permutation of weights from the initial model do in fact contribute to the accuracy, training loss, and iterations taken to achieve optimal results on these metrics.

2. Background

Let us define our initially dense network $f(x;\theta)$, with x and θ representing the input and parameters of the network, respectively. Before any pruning, we start with our model initialized with weights $\theta_0 \sim D_\theta$, $f(x;\theta_0)$. We then use an iterative pruning process, Algorithm 1, to generate a pruning mask m for the lottery ticket. The lottery ticket is then the sparse network $f(x; m \odot \theta_0)$, where $\odot$ applies element-wise multiplication.

Algorithm 1 Iterative Pruning Algorithm (Frankle & Carbin, 2018)

Initialize weights with $\theta = \theta_0 \sim D_\theta$
Initialize the mask m with an all one's vector of the same size as θ_0
Set the network to $f(x; m \odot \theta)$
for $i = 1, 2, \dots n$ **do**
  Train the network for j iterations, creating parameters $m \odot \theta'$
  Remove $s\%$ of non-pruned parameters in θ' with lowest magnitude by zeroing the respective indices of m
  Reinitialize $\theta = \theta_0$
end for
Return mask m

With hyperparameters j, s, and n, this pruning scheme creates a mask m with $100(1 - \frac{s}{100})^n\%$ of weights kept from the original network.

By construction, the non-zero weights in the lottery ticket after n iterative prunings, $\theta_{LT} = nonzero(m \odot \theta_0)$, do not come from the distribution that our full network was initialized with, D_θ, but rather a new distribution, which we shall label D_n. This paper specifically explores the role of D_n in initialization, and whether or not this new distribution contains sufficient information to create successful subnetworks. By default, our lottery ticket has mask m from Algorithm 1 and weights $\theta = \theta_0$. We test whether both the weight and mask initializations are important, by testing two new subnetworks: one with the same mask as the lottery ticket but new weights, $\theta \sim D_n$, and another with both a new mask (randomly created with the same size as the lottery ticket) and new weights, $\theta \sim D_n$. The first configuration solely tests whether the weight initialization leads to well-performing subnetworks, and the second tests both the mask and weight initialization.

Where this differs from previous work is that instead of comparing new configurations with $\theta \sim D_\theta$, we create new models with $\theta \sim D_n$, which is explicitly constructed via network pruning. Implicitly, we are testing whether D_n provides adequate initialization information for fast subnetwork training in comparison to the lottery ticket initialization with ordering θ_0.

To test this hypothesis, we would ideally generate a probability density function for D_n that captures the empirical distribution of non-zero weights in the lottery ticket network. However, as this distribution is explicitly formulated through the process of Iterative Pruning, we do not create such a pdf for our experiments. Instead, We quantize D_n with θ_{LT}, and generate samples from D_n by choosing uniformly at random from θ_{LT}. We leave the theoretical formulation of D_n to future work.

For creating a randomized mask, we did so on a layer-by-layer basis. For example, say we had a 3-layer fully connected network with 200 in the first hidden layer and 100 in the second, for a total of 300 parameters. A 20%-capacity mask might remove 175 parameters from the first layer and 85 parameters from the second. To create a randomized mask, we permute the mask of each layer individually, thus the permuted mask would also contain 175 and 85 parameters on the first and second layers, respectively.

3. Related Work

Lottery Ticket Hypothesis. The Lottery Ticket Hypothesis states that when randomly initialized, feed-forward neural networks contain a much smaller subnetwork, hypothesized as a "winning ticket," which can be trained to achieve a similar accuracy as the original network in a similar number of training iterations (Frankle & Carbin, 2018).

The process undergone to train a Lottery Ticket Subnetwork is to randomly initialize the parameters of a neural network, train the network over some number of iterations, prune a specific percentage of the parameters, then reset the parameters to their initial values. This process of training and pruning is iteratively repeated on the remaining parameters to find a winning ticket.

In the original and subsequent papers on the Lottery Ticket Hypothesis, the main focus in successful tests for initializing parameters within pruned networks were solely focused on reinitializing the weights to their values before any pruning had taken place (Frankle & Carbin, 2018) and transferring successful weight values from a previously pruned model (Morcos et al., 2019). There currently exists no in-depth analysis of the distribution created through iterative pruning and how reinitializing according to this distribution would

affect the accuracy of the subnetwork.

Masking. The process of pruning weights creates a mask of 1s and 0s, signifying which parameters are still being considered. Empirically, subnetworks created from a mask have not been able to be trained to a comparable accuracy when the parameters are reinitialized to random values rather than their initial values in the full network (Frankle & Carbin, 2018). This means the original initialization of the parameters is somehow tied to the pruning mask used on the network. Through experimentation with different masks, it has been shown that the with regards to Lottery Ticket networks, the reason this occurs is that the sign of the initialized weights must be the same as they originally were when the mask was found (Zhou et al., 2019).

Knowing that there exist different initializations under the same mask which can be trained to achieve results comparable to the full network, we raise the question of what happens when we fix the percentage of weights pruned but randomize our pruning mask, so different topologies of weights all coming from the same distribution can be tested.

Deep Models. In general, Lottery Ticket approaches have failed on deeper neural networks when using iterative pruning with similar percentages for pruning on smaller-scale networks (Frankle & Carbin, 2018). These deeper networks have subnetworks that can be shown to have a comparable accuracy to the original network while being 50 to 90 percent smaller, through pruning more earlier in training the network and through initializing to a former iteration in the training process rather than the original initialization before the network had begun to be pruned (Frankle et al., 2019). These results imply that depending on the size of the network, the pruning amounts of the network and the initialization of the parameters need to be changed to suit these values. In our work, we mostly focus on shallow networks due to these findings.

4. Experiment Results

4.1. Setup

All experimental runs use cross-entropy loss, the Adam optimizer with 10^{-4} weight decay, and a batch size of 1024 for training. Two networks were tested: a fully connected (FC-1) architecture on the MNIST dataset (LeCun & Cortes, 2010), and the LeNet5 (Lecun et al., 1998) architecture on the CIFAR-10 (Krizhevsky, 2012) dataset. The FC-1 architecture employed two hidden layers with 300 and 100 nodes, all separated by ReLU activation functions. The FC-1 network was trained for 100 epochs per run, and the LeNet5 network was trained for 300 epochs per run.

After an initial Xavier initialization (Glorot & Bengio, 2010), pruning was done using the iterative pruning strategy (Algorithm 1) with parameters of $j = 5$, $s = 20$, $n = 6, 10$. The pruning n of 6 and 10 respectively correlate to 26.2% and 10.7% network capacity.

As a base comparison point, we first trained the full-capacity original network. For each pruning amount, we record the mask and weight distribution of the pruned networks. Then, we trained three networks: the standard lottery ticket, the pruned network reinitialized from the lottery ticket weight distribution, and a randomly-masked network (with same capacity) reinitialized from the lottery ticket weight distribution. These are referenced by pruned/reinitialized/remasked-{sparcity amount}, respectively.

4.2. Fully-Connected Network on MNIST

Figure 2 plots the average test accuracy and train loss at each epoch across five trials for the FC-1 models. The pruned-10 and reinitialized-10 models achieve similar accuracy to the full network, as do all the 26% models. The remasked-10 achieves the lowest max accuracy throughout its training, and is .85% lower than the fully connected network. Each of the 26% network also achieve a better max accuracy than their 10% counterpart, but takes longer to reach this maximum accuracy except in the remasked case. We also see the trend that changing more of the initial conditions of the model, leads to a lower maximum accuracy obtained.

We can see similar patterns of the 26% network having better performance with respect to training loss (Figure 2). Each model achieves their minimum training loss near or at the end of training. Again, as we change the initial conditions of the pruned networks the training loss increases. Table 1 summarizes these three figures with max average accuracy, min average training loss and what epochs they occur.

4.3. LeNet5 on CIFAR-10

Figure 4 displays the average of five trials of each LeNet5 model's test accuracy. The full network achieves its maximum accuracy at epoch 89 and then begins to degrade in performance. Unlike before, only the full model achieved its maximum accuracy significantly before training stopped. As with FC-1, The 26% models perform much better than their 10% counterparts and the more the initial conditions are changed (weight initialization and mask) the lower the max accuracy the model achieves.

Figure 4 displays the train loss, where only the full network shows a significant decreasing trend by the end of training. Table 3 summarizes the test accuracy and train loss for each LeNet5 model.

Figure 1. Summary of FC-1 Performance. Values come from average of five trials. Includes max average accuracy, min average training loss, and what epochs they are obtained in

Model	Max Average Accuracy	Epoch	Min Average Training Loss	Epoch
Full	.9068	96	0.656e-3	95
Pruned-26%	.9055	91	1.707e-3	91
Pruned-10%	.9043	75	3.915e-3	96
Reinit-26%	.9045	100	2.395-3	100
Reinit-10%	.9036	67	4.471e-3	100
Remask-26%	.9028	95	2.964e-3	96
Remask-10%	.8983	100	10.04e-3	100

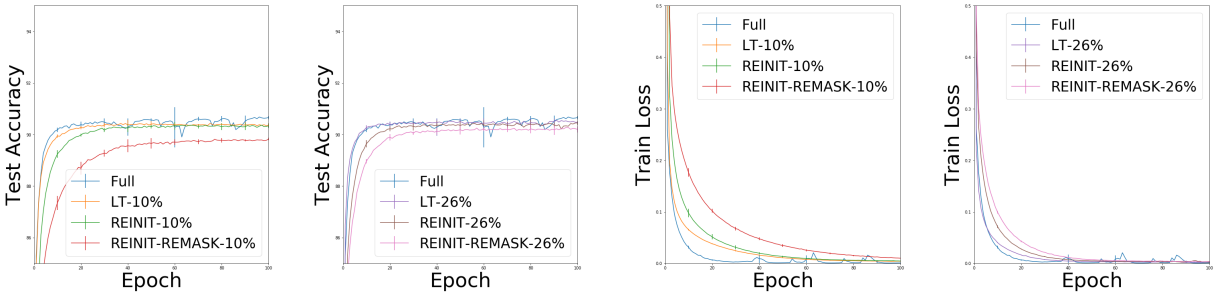

Figure 2. FC-1 training data. Test accuracy (two left) and train loss (two right) of model pruned to 10.7% (left) and 26.2% (right). Each line represents average of five trials with error bars representing max/min across the trials.

Figure 3. Summary of LeNet5 Performance. Values come from average of five trials. Includes max average accuracy, min average training loss, and what epochs they are obtained in

Model	Max Average Accuracy	Epoch	Min Average Training Loss	Epoch
Full	.5861	89	.2341	300
Pruned-26%	.5689	298	.8001	298
Pruned-10%	.5218	300	1.121	297
Reinit-26%	.5499	292	.8902	300
Reinit-10%	.5167	300	1.156	299
Remask-26%	.5478	269	.8884	298
Remask-10%	.5089	291	1.183	298

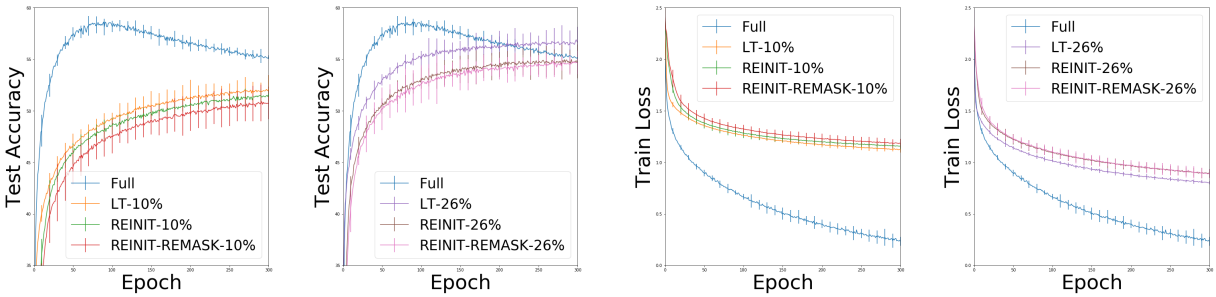

Figure 4. LeNet5 training data. Test accuracy (two left) and train loss (two right) of model pruned to 10.7% (left) and 26.2% (right). Each line represents average of five trials with error bars representing max/min across the trials.

4.4. Distributions at Initialization

Figure 5 shows histograms of θ_{LT} for the two networks at each of the pruning levels. We notice that though the exact distribution of weights is different between them, the rough shape of the distribution is quite similar. We see four peaks, of which the middle two are much higher than the outer two. There is a strong divot near 0 - this by itself is quite intuitive, as these are the weights that we remove via pruning. There is a rough symmetry for these distributions, though the exact scaling is different between the FC-1 and LeNet5 runs. This scaling offset is probably due to the difference in scaling in the original Xavier initializations.

5. Conclusion

This paper sought to explore the influence of the lottery ticket non-zero weight distribution, D_n, on the power of lottery ticket performance. In doing so, we tested various configurations of networks initialized with weights sampled from D_n against both the standard lottery ticket and the base non-pruned model. For FC-1, we see each model achieves a similar accuracy as the full network, although the full network still outperforms in that regard. Examining the train loss curves, we see the full network and lottery tickets train much faster than the other networks (Figure 2).

This trend is far more pronounced in the LeNet5 data. Though all of the models did not achieve a significantly high accuracy (less than 60%), in comparing the various runs, the full model and lottery tickets performed far better than the new configurations we proposed. Just in the 26.7%-capacity runs, the lottery ticket scored 1.90% higher than the reinitialized run with same mask, and 2.21% better than the remasked run. Though the training curves for these runs were comparable, overall the lottery ticket still outperformed the runs with same capacity.

These empirical results show that initializing a subnetwork with distribution D_n is *not* sufficient in producing a high-performing subnetwork. Indeed, both the specific configuration of the mask and the exact permutation of non-zero weights are crucial for subnetwork performance, and it seems that the lottery ticket is highly unique with respect to this. Ideally, we would not want to go through the process of pruning a larger network, but these findings imply that generating a performant subnetwork might rely on this process.

In analyzing the specific lottery ticket weights, we find surprising trends on the similarity among the distributions. In comparison to the original initialization, the lottery-ticket initialization D_n highly favours non-zero weights, and seemingly does not have a bias on positive or negative weights. Moreover, between the two model types, these distributions were formulated after pruning separate models on different underlying training data, so the similarity between these distributions is quite surprising. We intuit that this is caused by the pruning process, which iteratively removes values closer to zero. More work would need to be done to capture a mathematical model of an arbitrary θ_{LT} over any model/dataset, given a starting initialization distribution D_θ.

6. Future Work

There are multiple angles that this research should be extended on. In Section 4.4, we empirically found that the D_n distributions were similar between runs for the FC-1 and LeNet5 models, even though they were trained on completely separate datasets. This implies that, given a starting D_θ, it might be possible to generate a mathematical formulation of D_n. Though the direct use of this distribution is unclear, it is interesting how the distribution is more or less the same between different architectures and datasets. Moreover, our analysis only compared pruned models, which inherently depend on an artificial mask m. Though we have shown that for pruned models, θ_0 initialization outperforms D_n and D_θ initialization, we have yet to discover what happens for the full model; i.e. could D_n initialization perform better than D_θ initialization for a non-masked model.

References

Cun, Y. L., Denker, J. S., and Solla, S. A. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pp. 598–605. Morgan Kaufmann, 1990.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Training pruned neural networks. *CoRR*, abs/1803.03635, 2018. URL http://arxiv.org/abs/1803.03635.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. The lottery ticket hypothesis at scale. *CoRR*, abs/1903.01611, 2019. URL http://arxiv.org/abs/1903.01611.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9: 249–256, 01 2010.

Krizhevsky, A. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

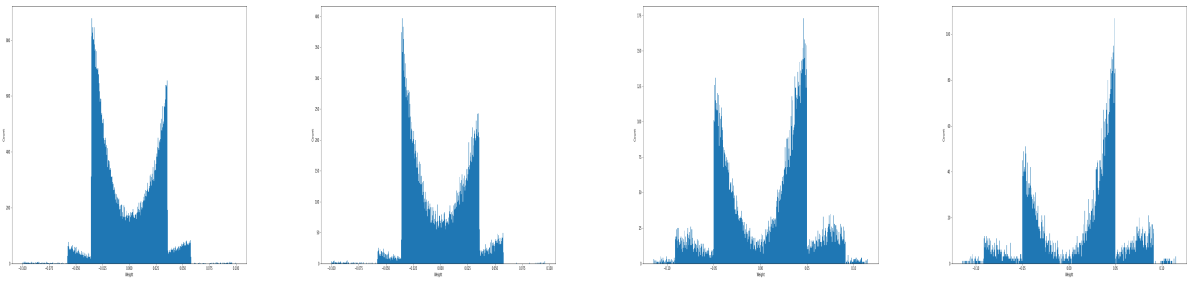

Figure 5. Initial non-zero weight distribution of lottery tickets for FC-1 (two left) and LeNet5 (two right), for subnetwork capacity 26.2% (left) and 10.7% (right).

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *CoRR*, abs/1608.08710, 2016. URL http://arxiv.org/abs/1608.08710.

Morcos, A., Yu, H., Paganini, M., and Tian, Y. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 4932–4942. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a4613e8d72a61b3b69b32d040f89ad81-Paper.pdf.

Zhou, H., Lan, J., Liu, R., and Yosinski, J. Deconstructing lottery tickets: Zeros, signs, and the supermask. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 3597–3607. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1113d7a76ffceca1bb350bfe145467c6-Paper.pdf.