Experiments on Lottery Ticket Hypothesis

Kevin Martin Jose
Universität des Saarlandes
s8kejose@stud.uni-saarland.de

Nisha George
Universität des Saarlandes
s8nngeor@stud.uni-saarland.de

Navami Kairanda
Universität des Saarlandes
s8nakair@stud.uni-saarland.de

Abstract

The recent "Lottery ticket hypothesis" by Frankle and Garbin [4] demonstrated a way to find trainable subnets of neural networks that achieve same or better accuracy as the original unpruned network. These networks, dubbed winning tickets, are identified by training a neural net, pruning smallest-magnitude weights and resetting the remaining weights to their original initializations. We examine if these tickets are trainable only because it has seen the same training data in the previous pruning iteration. As the process of uncovering a ticket is slow and tedious, we explore a faster alternative by using a fraction of the dataset for pruning iterations and examine its performance when retrained with the entire dataset. We observe that a speed-up of 7.5x can be achieved by using subset (10%) of training data to generate winning tickets while achieving the same accuracy when retrained on the full dataset. We also discover a winning ticket for Shufflenet, a network architecture with 48 layers, that makes use of depthwise separable convolutions.

1. Introduction

Most modern neural networks optimize millions of parameters and the general consensus is that the bigger the network, the better. Even though there have been efforts to reduce the parameter count of these networks while preserving accuracy, most of these techniques [[11], [3], [9], [8], [14]] start from a trained network and do not focus on decreasing the parameter-count during training. The Lottery Ticket Hypothesis by Franklin and Carbin [4] tells us that a trainable sparse network (winning ticket) can be found within a dense neural network. This means that a network with a given random initialization and the associated mask (structure) would give the same accuracy as the dense network. This is of great scientific interest as it sheds more light into how and why neural networks work the way they do. However, discovering a winning ticket is a computationally heavy process even for a relatively small network such as LeNet[3]. In this paper, we explore the possibilities of (a) finding a winning ticket faster for fully connected and convolutional neural networks[15], (b) finding a winning ticket for a subset of the dataset which performs equally well when retrained on the full network and (c) finding a winning ticket for a different neural architecture - namely ShuffleNet[17].

We find that discovering a winning ticket is possible with a small fraction of the dataset instead of training on the whole dataset. This makes the process of finding a winning ticket much faster. Another implication of this result is that a pruned network for one set of data is useful for another set of data with the same distribution. In other words, the trainability of a winning ticket is independent of the data it has seen before. Also, we successfully find a winning ticket for the ShuffleNet architecture. This is of interest because ShuffleNet makes use of batch normalization, depthwise separable convolutions [2] and channel shuffle to achieve faster training and these techniques might respond to pruning differently.

2. Related work

There have been some substantial earlier work to reduce the memory footprint of a neural network through pruning [[11], [7]]. Most of these works focus on taking a trained network and then pruning it with minimal deterioration of test accuracy. Creating a trainable neural networks was not the objective of these works. However, the overarching goal of our work is to produce pruned networks that both train faster and at the same time reach a higher accuracy. The precursor to our experiments is the paper by Frankle and Carbin where they formulate the lottery ticket hypothesis: *A randomly-initialized, dense neural network contains a sub-network that is initialized such that when trained in isolation it can match the test accuracy of the original net-*

work after training for at most the same number of iterations. Such a sub-network, along with the weights with which it was initialized and a mask, is known as a winning ticket since it achieves better performance than the original unpruned network. Iterative magnitude pruning is used to discover such winning tickets and it proceeds as follows:

1. Train a neural network from scratch
2. Take the bottom p% weights with the lowest magnitude and prune them (i.e create a mask)
3. Reset the unpruned weights to their initial values before training
4. Repeat steps 1 to 3 for n iterations

After each pruning step, the sub-network is trained on the full data set again (i.e step 1) and its test accuracy is measured. If it is found to be equal to or higher than the baseline test accuracy (of the original unpruned network) and we reach early stopping in fewer iterations, then that indicates a winning ticket.

Zhu et al. [18] deconstructed the lottery ticket hypothesis to try and understand why masking weights would result in better accuracy. They pruned the weights based on the magnitude of the gradient instead of the weights and concluded that the weights that were pruned were already moving towards zero.

3. Trainability on Unseen Data

According to Frankle and Cabins baseline experiments, lottery hypothesis requires iterative pruning on the whole dataset to uncover a sparse winning subnet. The ticket thus obtained is retrained on the same training set to evaluate if it can be trained from scratch. Since other pruning techniques like SqueezeNet[11] do not produce trainable pruned networks, we would like to examine if the ability to train a winning ticket exists simply because the network saw the same dataset in the previous pruning iteration. In other words, is the ability to be trained preserved if we attempt to train a winning ticket on new data? This motivates us to conduct an experiment by splitting the training set into two halves one for finding ticket and the other to examine if the ticket is trainable on unseen data.

1. Split the training data set into two parts, S1 = $\{(x, y)_m : m = 1, 2...\frac{n}{2}\}$ and S2 = $\{(x, y)_m : m = \frac{n}{2}, \frac{n}{2} + 1, ..., n\}$
2. Train the neural network on S1.
3. Prune the network by masking the connections with lowest weights.
4. Train the pruned network on S2.

Network	LeNet	Conv2	Shufflenet
Convolutions	-	64, 64, pool	48 conv layers
FC Layers	300, 100, 10	256, 256, 10	576
All weights	266K	4.3M	946K
Mini-batch size	100	200	100
Iterations	30K	30K	30K
Prune percent for Conv layers	20	20	10
Prune percent for FC layers	10	10	5
Optimizer	Adam	Adam	Adam
Dataset for evaluation	MNIST	CIFAR - 10	CIFAR - 10

Table 1. Underlying architecture of Fully connected, Convolutional and Shufflenet networks in all experiments. This setup is used for evaluating trainability on unseen data as well as for identifying winning tickets faster.

5. Report the test accuracy.
6. Iterate steps 2-5 for every pruning iteration.

We evaluate trainability of winning tickets on two networks, viz fully-connected architecture (LeNet) for MNIST and convolutional architecture for CIFAR10. The details about network configuration are provided in Table 1. As seen in Figure 1, we observe that winning tickets identified on S1 exhibit lottery ticket pattern on S2. For fully connected network, test accuracy at early stopping iteration matches the test accuracy of unpruned network at every stage of pruning up to 80%. In case of conv2, the accuracy is maintained as long as less than 85% weights are pruned. This is in line with the baseline results from Franklin and Carbin's paper. This provides evidence that a winning network from a dataset can be trained on unseen data as long as the data distribution remains similar.

4. Faster Winning Tickets

In the previous section, we observed that one half of the training data is sufficient to uncover a sparse trainable network. This leads to the question, Can we use much smaller splits of dataset to generate winning tickets? Identifying a winning ticket is a rather slow and cumbersome process. If we use less data in every pruning iteration, we would be able to discover winning tickets faster. However, do such tickets discovered using sparse data exhibit the lottery ticket pattern (i.e same or higher accuracy as unpruned network) when re-trained with the full dataset?

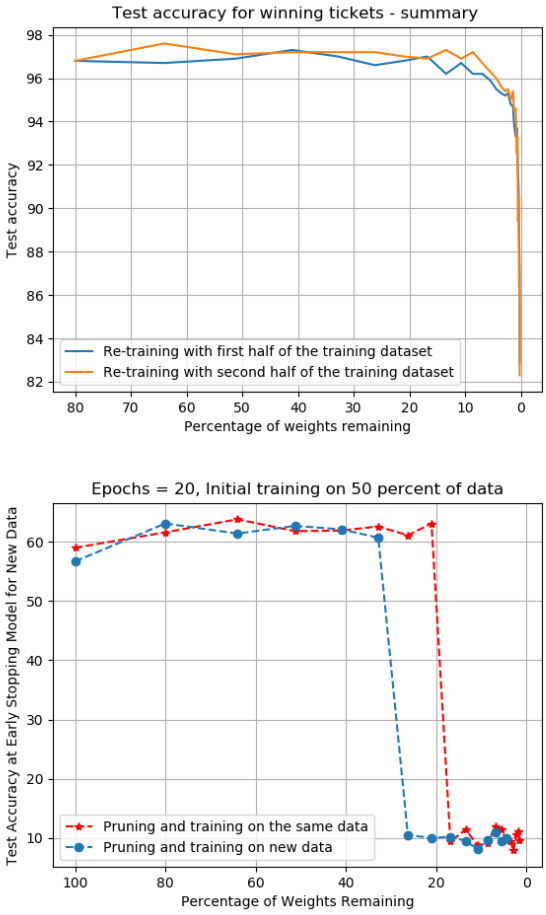

Figure 1. Test accuracy for tickets identified from S1, evaluated on S1 and S2 for fully connected network-MNIST and conv2-CIFAR10

We observe that using subsets of training data will provide fast and efficient ways to identify tickets which continue to exhibit the lottery ticket behavior that match the behavior for a subnet uncovered using the full dataset. The method remains same as the trainability experiment but with two modifications S1 is an s% subset of dataset, S2 is the full training set. We evaluate the method for varying size of subsets.

Figure 2 shows the test accuracy behavior of winning tickets after training from scratch with 100% of data. Here, we note that the winning tickets were identified with only a small fraction of the data. This confirms our hypothesis that a subset of data is adequate for effective pruning and uncovering of tickets. This result is promising as it demonstrates an accelerated technique for finding winning tickets. In Figure 3, we see that a subset size of 10% reaches earl stopping in around $\sim$1200 iterations, whereas $\sim$ 9000 iterations are required to early stop when using 100% dataset.

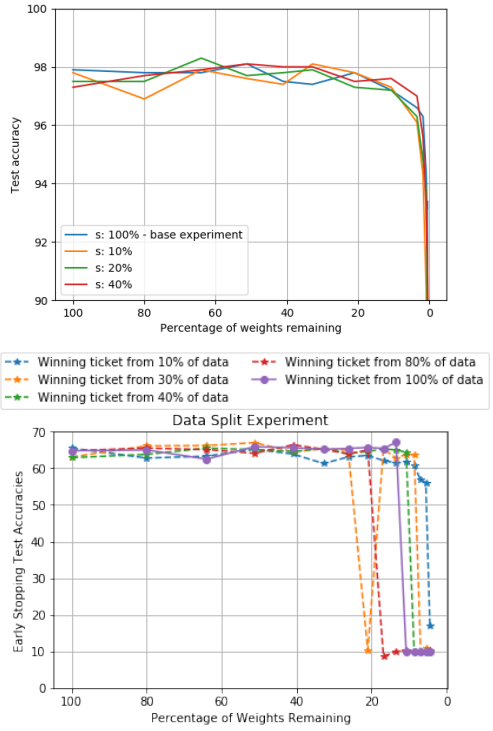

Figure 2. Test accuracy for tickets identified from smaller subsets, evaluated for fully connected (above) and Conv2 (below) network

Thus, the 10% model is 7.5X faster in discovering winning tickets while maintaining same test accuracy when retrained using the whole dataset. Similar behaviour is observed for Conv2 network as well. The 10% model takes $\sim$500 iterations as compared to $\sim$4500 for the dense model, while giving the same accuracy, as can be observed from Figure 2.

5. ShuffleNet

Frankle and Carbin successfully demonstrates lottery ticket hypothesis for relatively shallow networks, viz LeNet and Conv-2,4,6. However, they had to employ learning rate warmup in order to obtain winning tickets for deeper networks such as VGG[15] and ResNet[10]. Here we try to reproduce a winning lottery ticket for a network architecture that lies between LeNet and VGG in terms of the number of layers. The motivations behind choosing ShuffleNet are:

1. It has more layers than LeNet, but is not as big as to induce prohibitively large iterative pruning times

2. It makes use of depth-wise separable convolutions, batch norm layers and channel shuffle and the behavior of the associated kernels under pruning is unknown

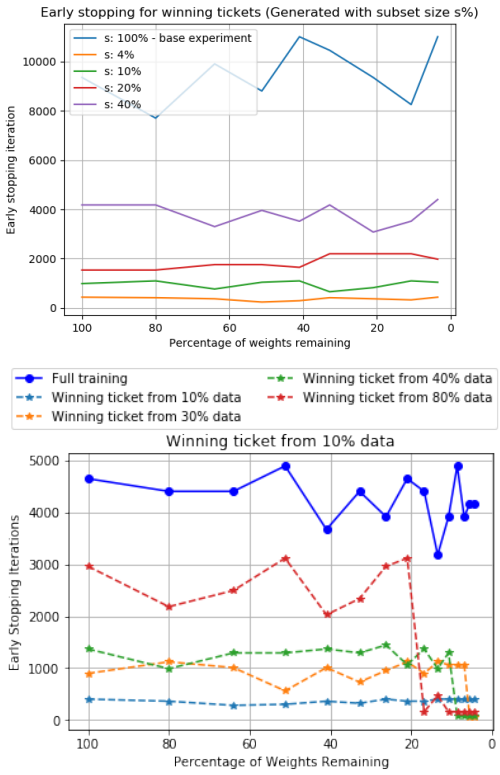

Figure 3. Early stopping iterations for tickets identified from smaller subsets, evaluated for fully connected (above) and Conv2 (below) network

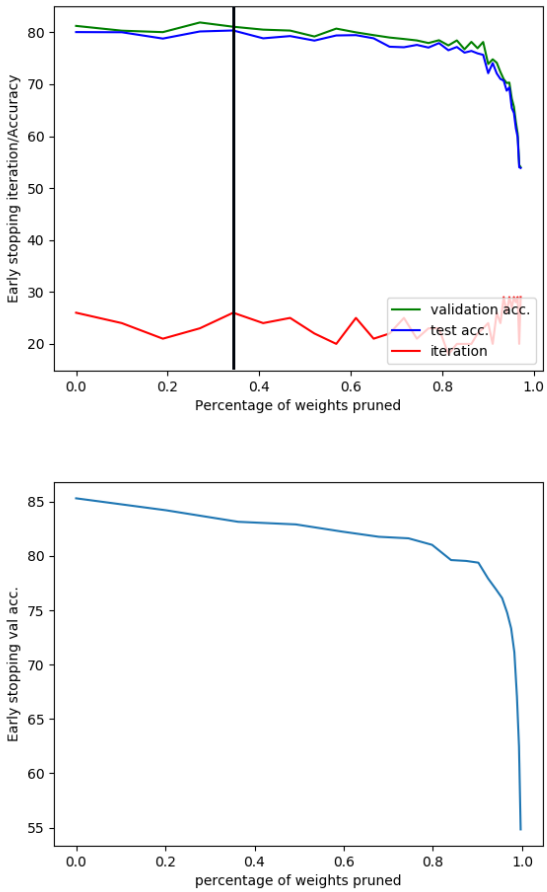

Figure 4. (Top) ShuffleNet exhibiting a winning ticket when around 40% are pruned. The vertical black line indicates one of the points where we observe a winning ticket. It can be seen that the sub-network did not take more iterations than the unpruned network to reach early stopping, and test accuracy is the same as when 100% weights were remaining. (Bottom) The validation accuracy of shufflenet decreases linearly with pruning if we prune the batch norm layers as well

We were able to reproduce a winning ticket for the ShuffleNet architecture by skipping pruning of the batch normalization layer (Figure 4). We get a winning ticket (i.e same test accuracy as the unpruned network) when around 40% of the weights are pruned, and the test accuracy of the network hovers over 75% until 80% of the weights are pruned. We can also see that for the winning ticket that we discovered, the network does not take any more iterations to reach early stopping than the original unpruned network. When we masked the weights of the batch norm layers, validation accuracy decreased linearly with the percentage of weights pruned (Figure 4 bottom). This is expected since the weights of a batch normalization layer depend on the activations of the previous layer, and setting some weights in the batch normalization layer to be always zero is incorrect. If the associated weight in the previous layer is zero (i.e masked), the network would adjust the corresponding batch norm weight in the further iterations. Tweaking the learning rate also plays a major role in discovering winning tickets. With a constant learning rate of $2e^{-4}$, we obtained no winning tickets for the shufflenet. However, starting with a higher learning rate of $1e^{-2}$ and then using a multiplicative step decay of 0.1 every 10 epochs yielded better results.

6. Methodology

The experiments were conducted using the PyTorch library and the associated code is hosted on Github [[13] [6] [12]]. In every forward pass, masking is done by applying the Hadamard product of the mask and associated weights. In each pruning iteration, the lowest $p\%$ of the weights in each layer are masked. p is a hyperparameter, and is different for different layer types. In all our experiments, the final fully connected layer is pruned at half the rate of the convolutional layers present in the network. The intuition is that if we prune the fully connected layer too fast it would bring down the accuracy since there would be too few weights left in the final layer to do classification. The

experiments on LeNet uses MNIST dataset [[16]], which is a dataset of handwritten digits. It has 60,000 training examples and 10,000 examples for testing. For the other experiments, the CIFAR-10 dataset is used. The CIFAR-10 dataset [[1]] consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

7. Conclusion

The above experiments conclude that a winning ticket that is obtained using a fraction of the training data is trainable on unseen data. Hence this validates that the trainability of a pruned network is not due to the fact that it has seen the same data in the previous pruning iteration. Furthermore, we observed that winning tickets obtained using a small subset of the dataset provide the same or better accuracy when the pruned network is retrained with the entire dataset. Using this technique, it is possible to speed up the discovery of winning tickets significantly without impacting the accuracy. It was also demonstrated that unlike deep networks like VGG and ResNet, shufflenet produces winning tickets without requiring a learning rate warmup. A more recent paper by Franklin et. al [5] demonstrates that instead of resetting unpruned weights to their initial values, setting them to their value at kth iteration produces better winning tickets for deeper networks. It is reasonable to believe that this modified iterative magnitude pruning with rewinding to the kth iteration would produce better winning tickets for Shufflenet as well. Moreover, it would be interesting to examine how our data sparsity and speedup experiments would behave under this setting, but we leave this to future work.

References

[1] Vinod Nair Alex Krizhevsky and Geoffrey Hinton. Canadian institute for advanced research dataset. *https://www.cs.toronto.edu/ kriz/cifar.html*.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[3] Yann Le Cun, John S. Denker, and Sara A. Solla. Advances in neural information processing systems 2. chapter Optimal Brain Damage, pages 598–605. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[4] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[5] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2019.

[6] Nisha George. Code for experiments on covn2 network. *https://github.com/nisha1729/lottery-ticket*, 2019.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[9] Babak Hassibi and David G Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pages 164–171, 1993.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[12] Kevin Martin Jose. Code for experiments on shufflenet. *https://github.com/lonesword/lottery-ticket-experiments*, 2019.

[13] Navami Kairanda. Code for experiments on fully connected network. *https://github.com/NavamiK/lottery-ticket-nn*, 2019.

[14] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] Christopher J.C. Burges Yann LeCun, Corrina Cortes. Modified national institute of standards and technology database. *http://yann.lecun.com/exdb/mnist/*.

[17] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[18] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *arXiv preprint arXiv:1905.01067*, 2019.