Deconstructing Lottery Tickets: A Reproducibility Study

Numa Karolinski, Bartosz Miselis, Monisha Shcherbakova
McGill University
Montréal, Canada
{numa.karolinski,bartosz.miselis,monisha.shcherbakova}@mail.mcgill.ca

Abstract

The main aim of this project was to conduct a reproducibility study on a paper that was presented at the NeurIPS 2019 Conference. The paper chosen was Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask, by Hattie Zhou, Janice Lan, Rosanne Liu and Jason Yosinski [1]. An ablation track was followed for this study to determine whether the conclusions of the paper are still consistent when certain modifications to the original experiments are introduced; these include the effect of the sign with *magnitude increase* mask criterion. Additionally, the statement that "masking is training" was tested by applying the masks on the models that did not fully converged. Due to limited computational resources, the study was narrowed down to reproducing only the fully connected neural network, excluding all the convolutional architectures. The code for all the experiments and figures presented in this report is available on GitHub[1].

1 Introduction

The motivation of the *Deconstructing Lottery Tickets* paper [1] is to understand and investigate the "Lottery Ticket Hypothesis" that was proposed by Frankle and Carbin in [2]. The main takeaway from their work was that it is possible to create new sparse networks that only retain the weights that ended up having large final values in the original training procedure. Such networks can be successfully trained from scratch to approximately the same accuracy as the unpruned network when having poor weights masked and the remaining ones initialised to the starting values for the weights that converged to large final ones. It was also found that the performance of these sparse models exceeds non-sparse ones, with no clear intuitions and deeper reasoning behind it. The aim of the *Deconstructing Lottery Tickets* paper was to build on top of the Lottery Ticket hypothesis and understand precisely which components contribute to the results of the aforementioned hypothesis and what could potentially be further adapted to improve the performance of the resulting network. The main components of their paper are as follows:

1. The importance of pruning (the process of setting unimportant weights to zero)
2. The discovery that the sign of the weight is the most important factor in terms of retaining information during retraining of a reinitialized network
3. The use of weight masks during training
4. The existence of novel "supermasks", that can be applied to an untrained network to create a model with the performance significantly better than random

The authors conducted an ablation study on the original work [2], and analysed the variability in the results by trying different masking criteria and masking actions (importance of the initial weights).

[1] https://github.com/bmiselis/deconstructing-lottery-tickets

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Figure 1: **Left:** sample image from the MNIST dataset (zero digit). **Right:** sample image from the CIFAR-10 dataset (*frog* class).

1.1 Databases Used

The two databases used to train and validate the analyzed network in this paper are the MNIST[2] digit recognition dataset, and the CIFAR-10 [5], which is a dataset of 10 classes of images.

MNIST Dataset

The MNIST dataset is a collection of images of black and white handwritten digits, where the class of the image is the digit. An example of the handwritten digit can be found on the left side of Figure 1. The full dataset consists of 60,000 training and 10,000 test examples.

CIFAR-10 Dataset

The CIFAR-10 database contains 60,000 RGB images that are 32 x 32 pixels, labelled into 10 classes depending on the object that is depicted in the image. An example of the image can be found on the right side of Figure 1. The datasets consists of 50,000 and 10,000 test images.

1.2 Background

Pruning

Pruning is a technique used to prevent overfitting in various machine learning models, and generally involves a parameter or weight being removed or set to zero from given type of network or graph. In neural networks, pruning refers to setting weights to zero (weights signify how much an input impacts the output). As a result of neural network pruning ,some of its paths do not need to be evaluated, saving a lot of computation time. Additionally, overfitting tends to cause a decrease in validation accuracy, so pruning a neural network should generally both speed up computation and increase validation accuracy (if appropriate pruning strength applied). Neural networks with 50%-90% of original parameters sometimes tend to outperform their unpruned counterparts, and in some cases this can drop to as low as 99.5% of nodes pruned [2] with infinitesimal change in the accuracy.

Masking

While the reason for pruning was made very clear in [2] (it lowers computation time and increases validation accuracy), it was not the case for which weights to actually remove. It might seem intuitive that the weights that are "bad" or "poor performing" would be set to zero, but these definitions are not rigorous. Much of the purpose of the *Deconstruction Lottery Tickets* paper [1] was to investigate this question in detail, with the idea of various masks criteria arising. Masks are rules that say which weights to prune and which ones to keep. These rules can be applied to any mathematical concept related to the weights. This means that the value, magnitude, and change of weights before and after the training can be used as the rules. The most straightforward one is probably the one saying that "weights with small magnitudes should be pruned" (large final magnitude implies that the input either increases or decreases the output significantly, having a large influence on the results). In the *Deconstructing Lottery Tickets* paper [1], multiple masking criteria were investigated, with vast majority of them being analyzed in this report. The masking criteria with short descriptions can be found in Table 1.

[2] http://yann.lecun.com/exdb/mnist/

Mask Criterion	Formula	Description
large_final	$\|w_f\|$	Large final magnitude
small_final	$-\|w_f\|$	Small final magnitude
large_init	$\|w_i\|$	Large initial magnitude
small_init	$-\|w_i\|$	Small initial magnitude
large_init_large_final	$min(\alpha\|w_f\|, \|w_i\|)$	Large Initial or final magnitude
small_init_small_final	$-max(\alpha\|w_f\|, \|w_i\|)$	Small initial or final magnitude
movement	$\|w_f - w_i\|$	Magnitude of difference
magnitude_increase	$\|w_f\| - \|w_i\|$	Difference in magnitudes
large_final_same_sign	$\|w_f\| * sign(\frac{w_f}{w_i})$	Large final magnitude (same sign)
large_final_diff_sign	$\|w_f\| * -sign(\frac{w_f}{w_i})$	Large final magnitude (different sign)
magnitude_increase_same_sign	$(\|w_f\| - \|w_i\|) * sign(\frac{w_f}{w_i})$	Difference in magnitudes (same sign)
magnitude_increase_diff_sign	$(\|w_f\| - \|w_i\|) * -sign(\frac{w_f}{w_i})$	Difference in magnitudes (different sign)
random	0	Random

Table 1: The variety of masks were analyzed in this report and the *Deconstructing Lottery Tickets* paper [1]. In the first column are the names of the masks that were used across all the figures. In the second column is the mathematical formula of the criterion (the larger this value is, the less likely it is to be pruned). In the final column is a verbal description of the criterion (which weights to keep).

1.3 Related Works

The Lottery Ticket Hypothesis

As mentioned in the introduction, the original idea analyzed in this report comes from the "Lottery Ticket Hypothesis" paper by Frankle and Carbin [2]. In the past, many neural network pruning methods were developed, in an attempt to reduce memory requirements while at the same time improving computational performance without compromising on accuracy. The issue was that highly pruned networks were hard to retrain from the initial values, and the retrained networks suffered great reductions in performance. In [2] the authors found an algorithm that reveals a subnetwork inside a trained network, whose initial weights are enough to train the entire network from scratch, achieving a comparable accuracy. In other words the "Lottery Ticket Hypothesis" can be stated as follows:

"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" [2].

The work that was accomplished in this paper had multiple contributions and implications. After successfully proving the above stated hypothesis, the authors concluded that the subnetworks can be trained much faster and reach test accuracies comparable to the original networks. The winning ticket (or the best subnetwork) can not only be trained fast but also reach higher test accuracy when compared to the original network. This finding can help researchers understand the underlying working of these complex algorithms and lead to even better network design [2].

Deep Compression

With increasingly more complicated machine learning models, there is a growing need for model compression to fit complex models into the available hardware. In 2016, Han et al. released a paper on the method called *deep compression* [3]. Deep compression is a three step procedure for compressing neural networks. The three steps are: pruning, quantization, and Huffman coding. This compression method was applied to two different image classification models and was able to decrease the model memory footprint by the factors of 35x and 49x. The first step in this procedure is a straightforward pruning of weights (reducing the number of connections between nodes) as has been explained earlier in this report. The second step includes weights sharing and quantization. Here, the number of effective weights is decreased by having multiple connections share the same weights. Additionally, this reduces the number of bits necessary to store all of the weights, decreasing the memory consumption further. Finally, Huffman coding is applied; this involves an optimal variable length encoder (if a variable is used more commonly, it gets encoded with a smaller number of bits so that less memory is used). Most importantly though, each of the three steps does not decrease the accuracy of the model, yet reduces the memory footprint by as high as $\simeq 50$x [3]. Potentially, the masks investigated in [1] could further reduce the storage requirements if applied as the pruning step.

Optimal Brain Surgeon

The optimal brain surgeon procedure [4] refers to a weight pruning approach that optimally removes weights by looking at second order derivatives. Just like in [3], the purpose of this method is to minimize neural network complexity and reduce storage requirements. Hassibi et al. explored the second order derivative of classification error with respect to the neural network weights. The procedure of the optimal brain surgeon algorithm involves initially training a neural network to minimal error. Then, the inverted Hessian is solved for; the Hessian can be defined by:

$$\mathbf{H}^{-1} = \left(\frac{d^2 E}{d^2 \mathbf{w}}\right)^{-1}, \tag{1}$$

where E is the error, and $\mathbf{w}$ is the weight vector; the Hessian contains all second order derivatives. It is used to find a value q that minimizes the saliency L_q, where:

$$L_q = \frac{w_q^2}{2[\mathbf{H}^{-1}]_{qq}}. \tag{2}$$

The q value is then used to update the weights in the neural network. The key point of this paper is that the usage of the second order error derivative performs much better than the standard magnitude-based methods of weight pruning; the ordinary methods often remove the weights mistakenly while the optimal brain surgeon (theoretically) does not [4].

2 Ablation Study

An ablation study on the "Deconstructing Lottery Tickets" paper [1] was performed in order to reproduce part of the presented results. Specifically, we focused on analysing the results from Figure 2, which displays the results obtained after training four different networks (Fully-Connected, 2-layer, 4-layer and 6-layer CNN) with multiple masking criteria and various pruning rates. The remaining weights for each iterative pruning step (defined as a percentage of the original number of weights of the model) denote the horizontal axis, while the test accuracy (at early stopping iteration) denotes the vertical axis. Each of the plot lines is a different masking criteria, and all the experiments were performed 5 times to obtain an average value with the uncertainty bands around each line. The various masking criteria form complimentary pairs, for instance `large_final` and `small_final`. It was noticed in [1] that such pairs result in one of the lines being above the *random* masking criterion, with the other one below it.

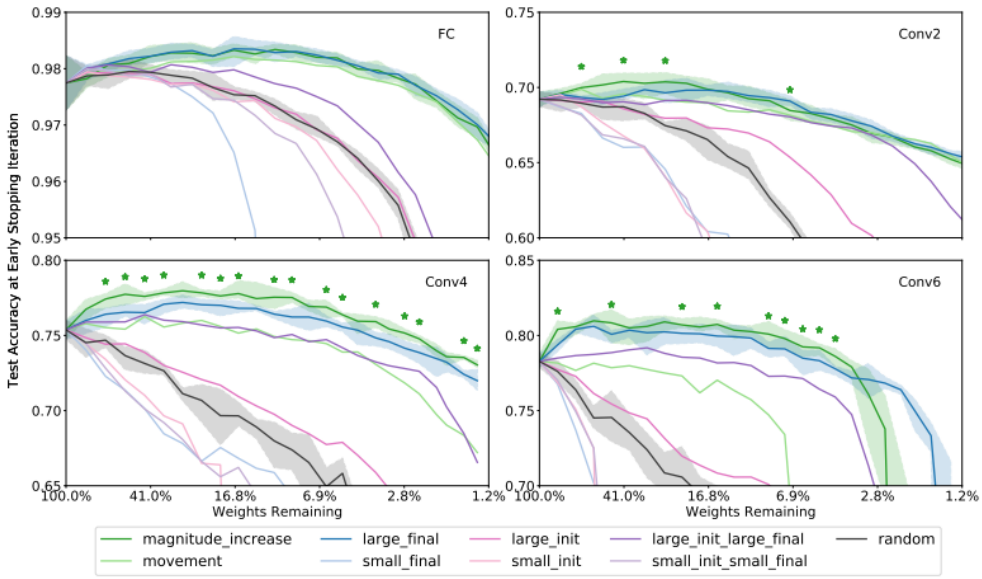


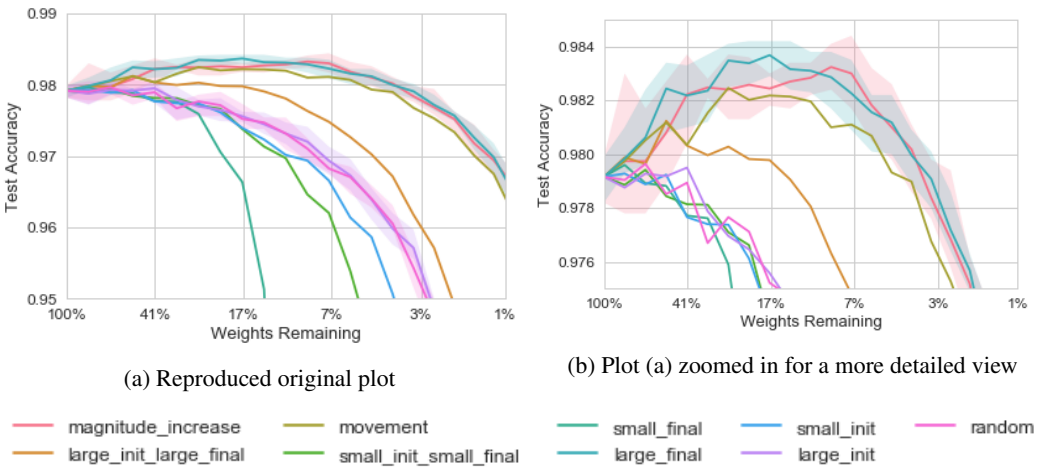
Figure 2: Original figure retrieved from [1].

The FC network was trained on MNIST data, while all the CNNs were trained on CIFAR-10, a detail that was not mentioned in the original figures clearly enough. The reasoning behind this could be that a simple FC network is not good enough for the CIFAR-10 data, hence the use of simpler MNIST.

As the CNN networks used in this paper require significant amount of time to run all the experiments (hardware available for this reproducibility study would require $\simeq$ 4 weeks to complete), we decided to focus solely on the FC network that was feasible to run across all the experiments in the time given.

2.1 Masking Criteria with Ablated Random Seeds

In our first experiment (that served as a baseline), we decided to reproduce the experiment that analyses the behavior of various masking criteria using the FC network, trained on the MNIST dataset. We ran the code shared with [1] paper, using arbitrary random seeds (the original paper does not mention the seeds used), to see if we can get similar results to the ones reported in the paper.



(a) Reproduced original plot

(b) Plot (a) zoomed in for a more detailed view

Figure 3: Results obtained for the FC model experiment using various masking criteria were reproduced. Arbitrary random seeds were used compared to the ones used in the paper. As can be seen from the above plots, they appear similar to the one reported by the authors. The masking criteria exist in complimentary pairs, one of them performs better than random and the other one worse. (the lines lie either above or below the random criterion one). The bands surrounding the plots depict the standard deviation, and the solid line is the average obtained over 5 runs with different random seeds.

As can be observed, the obtained results are very similar to the ones presented in [1]. All the masking criteria have an accuracy of about 98% when all the weights are retained (no pruning) and then either slightly increase for best-performing criteria (quite surprising effect that is probably the result of iteratively pruning less useful weights, implicitly transferring more knowledge from the original network) or gradually decreases for worse-performing ones. The best criteria appear to be `magnitude_increase` , `movement` and `large_final`, with the accuracy of the model quite well retained, even with strongly pruned weights. It suggests that these criteria successfully keep only the most important weights, allowing the rest to be dropped without hurting the accuracy of the model.

2.2 *Large Final Magnitude* Criterion with Ablated Random Seeds and Sign Analysis

The three masking criteria studied in details in this section, namely `large_final`, `large_final_same_sign` and `large_final_diff_sign` were compared to reproduce the results presented in top-left subplot of Figure S5 from the original paper [1]. These criteria examine if the weights had large final values, and whether they retained their original sign or not. As can be seen from the Figure 4, the `large_final_diff_sign` masking criteria performed quite poorly, with the test accuracy dropping rapidly as the number of weights being pruned increased. The `large_final` and `large_final_same_sign` performed relatively well, with the accuracy remaining quite high between 96-98%, even as the number of the remaining weights dropped significantly.

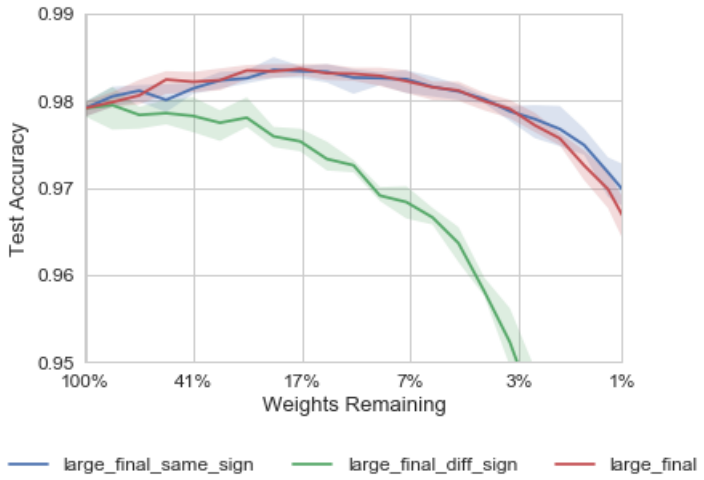


Figure 4: Test accuracy at early stopping iteration for `large_final` variants of masking criteria.

2.3 *Magnitude Increase* Criterion with Ablated Random Seeds and Sign Analysis

There were two masking criteria that were implemented in the original code, but not mentioned in the paper at all: `magnitude_increase_same_sign` and `magnitude_increase_diff_sign`. The focus here is on how the weights magnitudes change before and after the training. The `magnitude_increase_same_sign` criteria keeps the weights that increased in magnitude and retained their sign, while the `magnitude_increase_diff_sign` keeps the weights that increased in magnitude but flipped their sign. We analyzed these masking criteria by training the models in the same way as in the `large_final` case, visualizing the results in Figure 5.

It is worth noticing that the `magnitude_increase_same_sign` criterion performed very well, with the performance similar to `magnitude_increase`, staying at a very high accuracy of about 98% (even as the number of pruned weights gradually increased).

When applying `magnitude_increase_diff_sign` criterion, the accuracy of the model dropped sharply as the number of weights remaining reduced (keeping only the ones with high final magnitude and different sign), indicating that such weights do not carry valuable information for the future model training. However, the weights that increase in magnitude and retain their sign seem to be very informative: even with the majority of the weights being removed, it still maintains high accuracy.

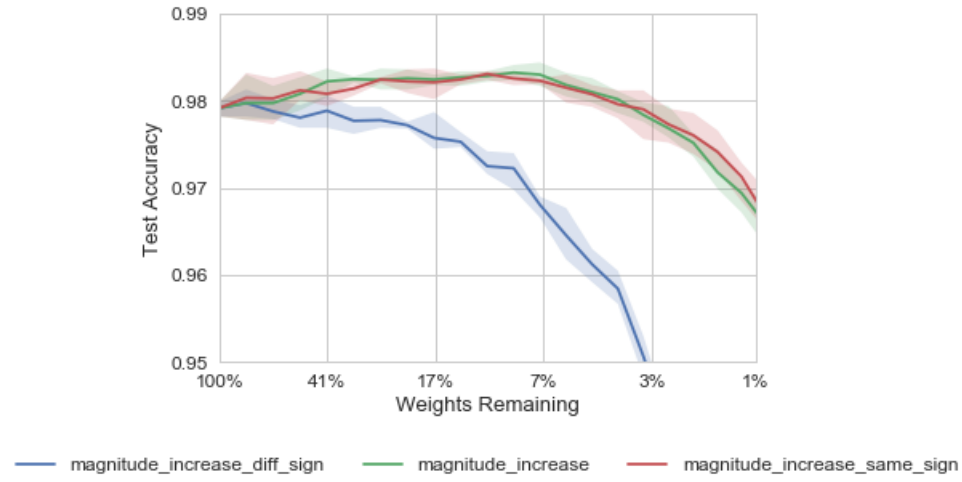
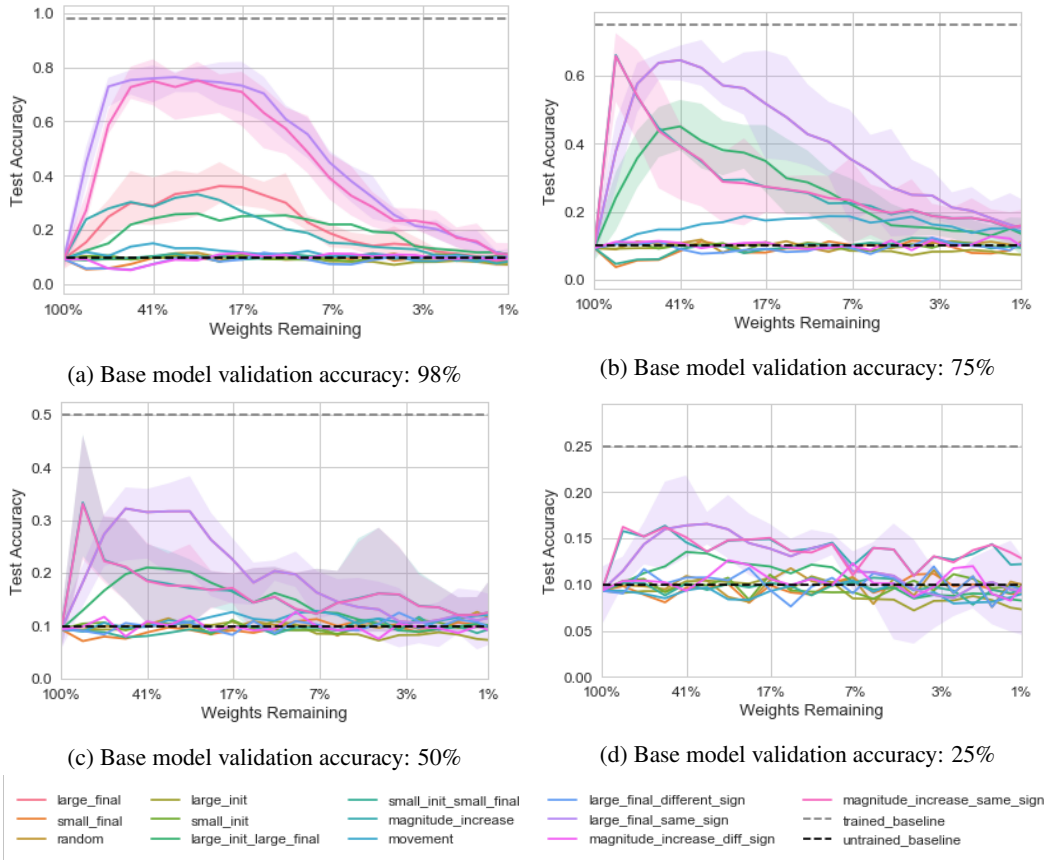


Figure 5: Test accuracy at early stopping iteration for `magnitude_increase` variants of masking criteria.

2.4 Does *Masking* = *Training* when Models Trained Poorly?

For the ablation study, it was decided to look into the author's following statement: "For certain mask criteria, masking is training". In section 5 of the paper [1] (called *"Supermasks"*), the authors investigated the presence of so-called "supermasks", which are masking criteria that lead to better than random accuracy at initialization when applied to an untrained network.

In the original study, the authors trained a base model **until convergence** and recorded its initial and final weights to compute the pruning masks. Then, the model weights were rewound to their initial values. Next, iterative pruning was performed using the aforementioned masks. No training was done at this point–only the pruning and masking were performed to get the final model.



(a) Base model validation accuracy: 98%

(b) Base model validation accuracy: 75%

(c) Base model validation accuracy: 50%

(d) Base model validation accuracy: 25%

Figure 6: Results of the ablation study. Different unpruned networks were used to produce individual subfigures. Each line denotes different mask criterion applied to unpruned models that were trained until convergence (top-left plot) or until the fixed validation accuracy (remaining plots). The test accuracy (vertical axes of the plots) was calculated at the initialisation (none of the models in this ablation study were trained). The figure is best seen digitally.

Figure 6 contains the results of the investigation on what the results would be if these masks were applied to a model that did not fully converge (until 98% validation accuracy) but was instead early-stopped at 25%, 50% and 75% validation accuracy. The various masking criteria were applied to these three models and then pruned iteratively to see whether the pruning may improve the validation accuracy of sub-optimal models.

The conducted experiments show that there exists a positive correlation between the performance of the base and the reinitialized models. From the plots in Figure 6 it can be observed that the higher the accuracy of the base model the better the performance of the model pruned with the top masking criteria (`large_final_same_sign` and `magnitude_increase_same_sign`).

For the top masking criteria, the pruning procedure can be split into two phases: the one that results in the model's accuracy being improved (retaining information acquired while training the original model) and the over-pruning (removing too many weights which starts to hurt the accuracy). It is essential to stop pruning when it starts to hurt the validation accuracy. Potentially early-stopping for the pruning process could be used to obtain the best results.

It is important to note that better convergence of the base model leads to smoother pruning results. When the accuracy of the unpruned network is not high enough (the base model converged to noisy final weights)–as can be seen in Figure 6(d)–the masking criteria are not able to keep informative weights, which leads to poor overall performance.

To sum up, the conclusions are as follows: masking **is** training (when base model converged to reasonable final weights) and the better the base model, the smoother the results.

Statement of Contributions

1. **Numa Karolinski: Analysis of code, write-up.**
2. **Bartosz Miselis: Analysis and running of code, generation of data and data crawlers.**
3. **Monisha Shcherbakova: Write-up, generation of plots, figures and analysis of code.**

Acknowledgments

The members of this team would like to thank Dr. Will Hamilton and McGill University for giving us a chance to participate in the NeurIPS Reproducibility Challenge 2019.

References

[1] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. arXiv:1905.01067v3 [cs.LG], September, 2019.

[2] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In International Conference on Learning Representations (ICLR), volume abs/1803.03635, 2019.

[3] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. CoRR, abs/1510.00149, 2015.

[4] Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, Advances in Neural Information Processing Systems 5, pages 164–171. Morgan-Kaufmann, 1993

[5] Krizhevsky, A., Hinton, G. (2009). Learning multiple layers of features from tiny images (Vol. 1, No. 4, p. 7). Technical report, University of Toronto.