IE643: Deep Learning : Theory and Practice **Aug-Dec 2020**

End-term Project Report : Lottery Ticket Hypothesis

Team Name: C3POs *Team Members: 19i190005,19i190010,19i190011*

Abstract

Researchers across the world have come up with a large number of Deep Neural Networks to perform different tasks. However, the main concern about all those models is the large number of parameters that need to be trained to perform the task keeping in mind the risk of overfitting. Now for training those large number of parameters, it not only requires vast computation capacity, and it also requires a considerable amount of time. The research paper mainly tries to solve this question of overparametrization through pruning.

Different researchers have researched on the topic of pruning neural networks and has stuck to the question that, can the sparse model be trained from scratch or not. In this research, the authors have suggested a method to obtain a pruned network which can be trained in isolation to attain a commensurate accuracy. In the proposed method, they first initialized the network, trained it from some epochs to get a considerable accuracy and then performed unstructured pruning. Then, they reinitialized the unpruned model weights and retrained it. They performed this process recursively to obtain a subnetwork with a lesser number of parameters than the initial network model but have the same or better accuracy.

By applying their proposed technique, they have demonstrated a considerable number of significant results on the MNIST and CIFAR-10 dataset. They have worked on different convolutional architectures like LeNet, Conv-2/4/6. Their method has shown to have extensively high accuracies even when only 10-20 percent of original weights remain.

In this report, we make further progress in our project by reciprocating another work in similar terms. In this work, researchers have tried to use the same winning ticket irrespective of the hyperparameters taken, and also found the performance of the winning ticket on another data set.

1 Introduction

Over the years, immense development of Neural Networks has created a different number of architectures, each focusing on specific types of tasks. However, training each of those architectures requires not only time but a considerable space for a massive number of parameters. In recent past, there has been a massive rise in mobile devices. With the rise of smart mobile devices [3], those devices need to run some of the state-of-the-art neural network models to offer different services to people in daily needs.

Due to the huge space requirement and complexity of the model, it is impossible to run on those devices. So over the years, people have been considering pruning as a technique to run the models on smaller devices. People have come up with different measures of saliency to select the weights for pruning (Lecun et al [5]). Many researchers have provided many efficient algorithms for optimal pruning techniques like Song Han et al [3]. However, all researchers have come to a point asking the following questions: Can this ultimate pruned network can be retrieved at an early stage and can be trained from scratch?

Our project is based on a paper that tries to answer these questions with an algorithm. To solve the question, researchers have formulated the following hypothesis based on their algorithm. They propose Lottery Ticket Hypothesis [1] as the following,

'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 network after training for at most the same number of iterations'

They have considered an initial model, trained it and then pruned the model based on the magnitude of weights. Then, they initialized the weights of the pruned model to that of the original network and retrained it and pruned it again. They continued this process until they find the smallest sub-network that either outperforms or shows a similar output to that of the original network. They have also shown changes in performance over changing hyper-parameters ,techniques including Dropout, Warm-up and different re-initialization methods such as original re-initialization, random re-initialization and random sampling. They have shown their final subnetwork, or "winning ticket" as they have called, in different networks like ResNet,VGG, Conv-2/4/6 and have shown that even 10-20 percent of remaining parameters have given comensurate accuracy.

We have also tried on an extension work on that paper where researchers have discussed about the main problem that is finding these "winning ticket" initialization is computationally expensive. One potential solution they discussed is to reuse the same winning tickets across a variety of datasets and optimizers. However, the generalizability of winning ticket initializations remains unclear. Here, they attempt to answer this question by generating winning tickets for one training configuration (optimizer and dataset) and evaluating their performance on another configuration.Their method has worked on some specific changes that they have considered like training on one partition of the data and evaluating on another part.

We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details on experiments in Section 6. A description of future work is given in Section 8. We conclude with a short summary and pointers to forthcoming work in Section 9.

2 Contributions

The main research paper [1] on which our project is mainly based has a impact on different aspects, some of which are listed below :

1. We demonstrate that pruning uncovers trainable sub networks that reach test accuracy comparable to the original network from which they were derived in a comparable number of iterations.
2. We show that pruning finds winning tickets that learn faster than the original network while reaching higher test accuracy and generalizing better.
3. We propose the lottery ticket hypothesis as a new perspective on the composition of neural networks to explain these findings.

The research paper [7] based on which we extended our project contributes:

1. Winning tickets are capable of transferring across a variety of training configurations, suggesting that winning tickets drawn from sufficiently large datasets are not overfit to a particular optimizer or dataset, but rather feature inductive biases which improve training of sparsified models more generally.
2. Winning tickets generated against datasets with more samples and more classes consistently transfer better, suggesting that larger datasets encourage more generic winning tickets.

Thus, the above two conditions imply that ticket initializations satisfy a necessary precondition (generalizability) for the eventual construction of a lottery ticket initialization scheme.

3 Literature Survey

A lot of work has been done regarding pruning of the neural networks.One of the most significant work has been done by Yan Lecun et al[5]. They have proposed the Optimal Brain damage technique. Optimal Brain Damage technique comes into the picture for reducing the size by selecting and reducing the size. It is found that networks with too many weights do not generalise well, whereas systems with too few values will not be able to represent the data clearly. So a trade off between the training error and network complexity is considered. One technique is to consider a cost function made off ordinary training error and plus some measure of network complexity. They have made a technique that uses second-order derivatives of the objective function with respect to parameters to compute the saliencies. Their proposed method was as follows:

1. Choose reasonable network architecture
2. Train network until a reasonable solution is achieved
3. Compute $h_{kk} = \frac{\delta^2 E}{\delta u_k^2}$ for each parameter u_k
4. Compute the saliencies for each parameter $s_k = hkk * u_k/2$
5. Sort the parameters by saliency and delete some low saliency parameters.
6. Iterate step 2

In later years, as mobile devices were becoming more and more needful, the requirement for training sophisticated neural networks were important but it became impossible. So Song Han et al[3]. came up with Learning both Weights and Connections for Efficient Neural Networks where they proposed a method to prune network connections in a manner that preserves the original accuracy. After an initial training phase, we remove all connections whose weight is lower than a threshold. This pruning converts a dense, fully-connected layer to a sparse layer. This first phase learns the topology of the networks — learning which connections are important and removing the unimportant connections. We then retrain the sparse network so the remaining connections can compensate for the connections that have been removed.In the following years Hao Li et al[6]. proposed Pruning filters for efficient covnets where he discussed about a technique of acceleration method for CNNs, where we prune filters from CNNs that are identified as having a small effect on the output accuracy. By removing whole filters in the network together with their connecting feature maps, the computation costs are reduced significantly. Their technique was :

1. For each filter $F_{i,j}$, calculate the sum of its absolute kernel weights $s_j = \sum_{l=1}^{n_i} |K|$
2. Sort the filters by s_j
3. Prune m filters with the smallest sum values and their corresponding feature maps. The kernels in the next convolutional layer corresponding to the pruned feature maps are also removed.
4. A new kernel matrix is created for both the i^{th} and $i + 1^{th}$ layers, and the remaining kernel weights are copied to the new model.

In coming years, it was realized by the authors of the paper Lottery ticket [1]that it is very time taking to get the winning lottery ticket. So they came up with a follow up work where their main contributions[2] were:

1. They propose a small but critical change to Frankle and Carbin's procedure for finding winning tickets that makes it possible to overcome the scalability challenges with deeper networks. After training and pruning the network, do not reset each weight to its initialization at the beginning of training; instead, reset it to its value at an iteration very close to the beginning of training. They termed this

practice late resetting since the weights that survive to prune are reset back to their values at an iteration slightly later than initialization.

2. Finally, they studied a possible mechanism for the efficacy of late resetting and—more broadly—the lottery ticket hypothesis: stability to pruning. The stability of a network to pruning is the extent to which pruning produces a subnetwork that follows a similar optimization trajectory regardless of whether the surviving weights are trained in isolation or as part of the full network.

In this paper one of the most important concepts that they came up was of stability. One starting point is the observation that networks found by iterative pruning without late resetting perform similarly to winning tickets that have been randomly reinitialized. We describe the stability of training to pruning (with respect to a specific pruning mask) as the extent to which the optimization trajectories of the unmasked weights produced by training without the mask (i.e., the entire network) and with the mask (i.e., the pruned network) are similar. Final values of the unpruned weights in training the entire network or the pruned network might be the same or not. If same then the network is stable to pruning otherwise unstable. From this concepts and experiments they also hypothesized that winning tickets identifiable by pruning emerges after the network is stable to pruning i.e. the final weights obtained after training of the network and further pruned network is close, at which point late resetting is effective.
Researchers at[1] shows that there exist winning tickets (small but critical subnetworks) for dense, randomly initialized networks, that can be trained alone to achieve a comparable accuracy to the latter in a similar number of iterations. However, the identification of these winning tickets still requires the costly train-prune-retrain process, limiting their practical benefits. In new research [8], researchers discover for the first time that the winning tickets can be identified at a very early training stage, which they term as Early-Bird (EB) tickets, via low cost training schemes (e.g., early stopping and low-precision training) at large learning rates. Their finding on the existence of EB tickets is consistent with recently reported observations that the key connectivity patterns of neural networks emerge early. Furthermore, they propose a mask distance metric that can be used to identify EB tickets with a low computational overhead, without needing to know the true winning tickets that emerge after the full training.

4 Methods and Approaches

Generalizing the entire literature discussed above, neutral network pruning typically follows a relatively standard process. Initially we begin with a neural network where each connection has been set to a random weight and first we train the whole network. Until recently, nearly all neural network pruning strategies in the literature actually pruned after the network has gone through substantial training or was fully trained. Secondly, redundant structures are removed from the network according to some heuristic. In most of the earlier work, pruning strategy of removing weights with lowest magnitudes (sparse pruning) was followed. Once the network was pruned, the underlying function learnt was damaged somewhat and to recover that function, the network was trained some more (fine-tuning). Finally, the above steps were repeated iteratively to prune more and more connections each round.

Based on the above steps for network pruning , it was found that for several common vision networks the parameter count of the respective networks could be reduced by an order of magnitude after training without losing any accuracy [3]. Thus, the pruned network can represent an equally accurate function. Following in the footsteps of training a pruned network from start with an aim to reduce cost of network training, the two ideas that played a central role in the development of [1] were:

- “During retraining, it is better to retain the weights from the initial training phase for the connections that survived pruning than it is to re-initialize the pruned layers.” [3]

- "Training a pruned model from scratch performs worse than retraining a pruned model, which may indicate the difficulty of training a network with a small capacity." [6]

Eventually, Frankle and Carbin begged the questions that served as motivation for their research work along the lines,

- Can sparsely pruned networks be trained from scratch ?
- Do networks have to be overparameterized to learn the underlying complexity ?

Frankle and Carbin found that the above two motivating questions could indeed be answered and contrary to previous findings, sparse networks can indeed be trained from scratch and networks doesn't have to be overparameterized to learn given one specific condition holds. The condition being, one can't just randomly re-initialize the pruned network viz. we need to give the connections the exact weight it had when it was part of the full network. Based upon the experiments and the results that they obtained, they formulated the **"Lottery Ticket Hypothesis"**.

Mathematically, the lottery ticket hypothesis can be formulated in the following way,

Let us consider a dense feed-forward neural network $f(x, \theta_0)$ with initial parameters $\theta = \theta_0 \sim D_0$
When optimizing with stochastic gradient descent (SGD) on a training set,

1. $f(x, \theta_0)$ reaches minimum validation loss l at iteration j with test accuracy a (commensurate accuracy) and,
2. $f(x, m \, \Theta \, \theta_0)$ reaches minimum validation loss $l^{'}$ at iteration $j^{'}$ with test accuracy $a^{'}$ where $m \in \{0,1\}^{|\theta|}$

Then, the Lottery Ticket Hypothesis predicts that $\exists$ m for which for which $j' \leq j$ (commensurate training time), $a \leq a'$ (commensurate accuracy), $\|m\|_0 <| \theta |$ (fewer parameters).

4.1 Work done before mid-term project review

With a purpose of designing a method for obtaining a **winning ticket** viz. the subnetwork obtained after pruning such that it aligns with the lottery ticket hypothesis , the following algorithm was used [1],

1. Randomly initialize a neural network $f(x, \theta_0)$ with initial parameters $\theta = \theta_0 \sim D_0$
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 where $m \in \{0,1\}^{|\theta|}$
4. Reset the remaining parameters to their values in θ_0, creating the winning ticket $f(x; m \, \Theta \, \theta_0)$

Presuppositions of the above proposed method are,

- There are number of pruning strategies like structured and unstructured that are being practiced from a long time. In the research paper, they have considered low final weight based unstructured pruning.
- There are two types of pruning – One-shot and iterative. In this paper [1], the authors focused on iterative pruning, which repeatedly trains, prunes, and resets the network over n rounds; each round prunes $p^{1/n}$ % of the weights that survive the previous round.

4.2 Work done after mid-term project review

Aligning in the footsteps of the work done before mid-term project review on finding winning tickets for a few specific well known architectures, the following observations served as a motivation for the post mid-term review work [1] , which also serves as a significance of the methods proposed later in this section.

- The success of lottery ticket initializations suggests that small, sparsified networks can be trained so long as the network is initialized appropriately. Unfortunately, finding these "winning ticket" initializations is computationally expensive.
- One potential solution is to reuse the same winning tickets across a variety of datasets and optimizers. We can attempt to answer this question by generating winning tickets for one training configuration (optimizer and dataset) and evaluating their performance on another configuration.

We divided the work done post mid-term review into **three tasks** for clarity. We proceed subsequently, defining these tasks and proposing a solution for each upon careful investigation of a few papers in the literature.

Task 1. *Finding out **winning ticket** on one data subset, training on another subset of the original data and evaluating the obtained Ticket's performance on this new data subset.*

Methodology:

- By training and performing iterative pruning on data subsets S_1 and S_2 we obtain winning tickets T_1 and T_2 respectively.
- The weights of the winning ticket T_1 are reset to their original initializations, the exact weights it had when it was part of the full network. Subsequently, we train the winning ticket T_1 on data subset S_2
- Finally, the performances of the winning tickets (T_1) and (T_2) were evaluated and compared.

Task 2. *Finding out **winning ticket** on one data subset having a set of labels, training on a new data set having different class labels and evaluating the obtained Ticket's performance on this new data subset.*

Methodology:

- By training and performing iterative pruning on data sets S_1 and S_2 we obtain winning tickets T_1 and T_2 respectively.
- The weights of the winning ticket T_1 are reset to their original initializations, the exact weights it had when it was part of the full network. Subsequently, we train the winning ticket T_1 on new data set S_2
- Finally, the performances of the winning tickets T_1 and T_2 were evaluated.

Task 3. *To check whether **winning ticket** generated using one optimizer will generalize to the performance obtained by a winning ticket generated based on another optimizer.*

Methodology:

- By training and performing iterative pruning, based on optimizers O_1 and O_2 we obtain winning tickets T_1 and T_2 respectively.
- Finally, the performances of the winning tickets T_1 and T_2 were evaluated and compared.

5 Data set Details

The **MNIST** dataset has been used in all the experiments performed. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.

6 Experiments

Framework : For work done both before and after mid-term review, the **PyTorch** framework was considered exclusively for all the experiments and testing performed as part of the project.

Hardware configurations : The entire work was done on Google Colab, with each file consuming ∼4GB of the assigned RAM. The work was done using GPU hardware accelerator, whose exact configuration varies over time and include Nvidia K80s, T4s, P4s and P100s.

Pruning strategy : The *iterative unstructued pruning* technique was considered throughout the project.

Loss function : Cross entropy loss function was considered in the implementation.

For any other related hyperparameter,refer to the corresponding table of hyperparameters.

6.1 Pre-Midsem

6.1.1 Status of experiment :

Implemented a dense Multi-Layer Perceptron (MLP) network and LeNet-5 and found winning tickets on each architecture.

6.1.2 Architecture Details :

MLP Input images of size 28 x 28 were taken, when were then fed into a hidden layer with 300 neurons, which in turn is again fed into another hidden layer with 100 neurons, which are connected to the output layer of 10 neurons, each neuron in output layer denoting the digits from 0-9. Softmax was used as the output layer activation function to obtain the required results.

Notationally, the MLP architecture can be written as: 28 x 28 - 300 - 100 - 10

LeNet-5 The standard LeNet [4] architecture was used throughout in all the experiments conducted.

6.1.3 Hyperparameters :

Hyperparamter	MLP	LeNet-5
Learning rate	0.0012	0.0012
Batch size	60	60
Train Epochs	75	50
Pruning %	30	20
Prune iteration	20	20

Table 1: Table of hyperparameter values considered for each architecture

6.2 Post-Midsem

6.2.1 Status of experiment :

Delineated on the **generalizability** of winning tickets obtained by implementing **Task 1, Task 2 and Task 3** as explained above.

6.2.2 Architecture details:

The architectures considered for the work done post mid-sem review are the same as the architectures used for pre-midsem review, namely dense Multi-Layer Perceptron(MLP) network and LeNet-5.

For the experiments performed after mid-sem review, the configuration of the experiments performed for Tasks stated above has been summarized in the next subsection.

6.2.3 Hyperparameters

Hyperparameter	Task-1	Task-2	Task-3
Learning rate	0.0012	0.0012	0.0012
Train Epoch	50	50	50
Pruning %	50	35	50
Prune iteration	5	15	10
Batch size	50	50	50
Optimizer	Adam	Adam	Adam and SGD

Table 2: Table of hyperparameter values and optimizers considered for each task

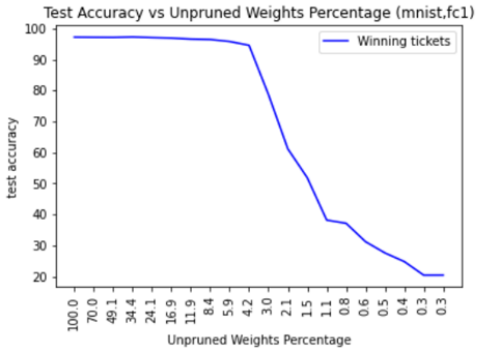
7 Results

We divide this section into 2 halves summarizing the results obtained before midsem, and those obtained after the midsem comments were received. The following subsections would follow this order: Plot, Relevant table (if any), Comment on the findings.

7.1 Pre-Midsem:

The implementation of the algorithm to find out the Lottery Ticket and observing the accuracy of the network even at a significantly pruned stage yielded the following plots.

(a) Test Accuracy vs. Unpruned Weights in the network



(b) Test Accuracy vs. Training Iterations

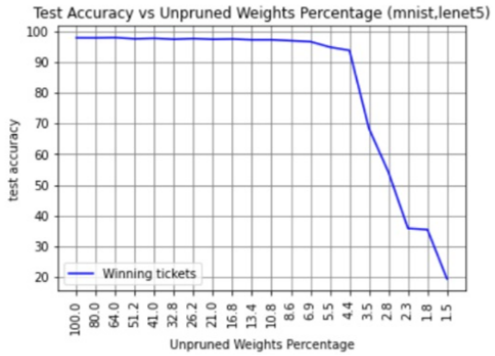
Figure 1: Using FC1 architecture on the MNIST data set

Comments on the plot: The plots obtained for the fully connected network provide the following observations:

- We achieve commensurate accuracy level even at a significantly pruned stage. We observe that even after pruning more than 90 % of the weights, we get accuracy almost same as the original dense network.
- Accuracy level falls drastically to a level below 80% when the network gets pruned by more than 95%.

We also implemented the algorithm on the MNIST data set using the LeNet-5 architecture, and the plots that were obtained are as follows:

(a) Test Accuracy vs. Unpruned Weights in the network



(b) Test Accuracy vs. Training Iterations

Figure 2: Using LeNet-5 architecture on the MNIST data set

Comments on the plot: The plots for the LeNet-5 architecture reveal the following:

- We achieve commensurate accuracy level even at a significantly pruned stage. We observe that even

after pruning more than 85 % of the weights, we get accuracy almost similar to the original dense network.

- Accuracy level falls drastically to a level below 80% when the network gets pruned by more than 95%.

In conclusion to these observations over the same data set MNIST using two different architectures, we can infer that the overparametrized networks can be significantly pruned, however keeping their original initializations, to provide highly comparable accuracy as to the original network. The initialization is an important criteria for the Winning Tickets to perform with such high level of accuracy.

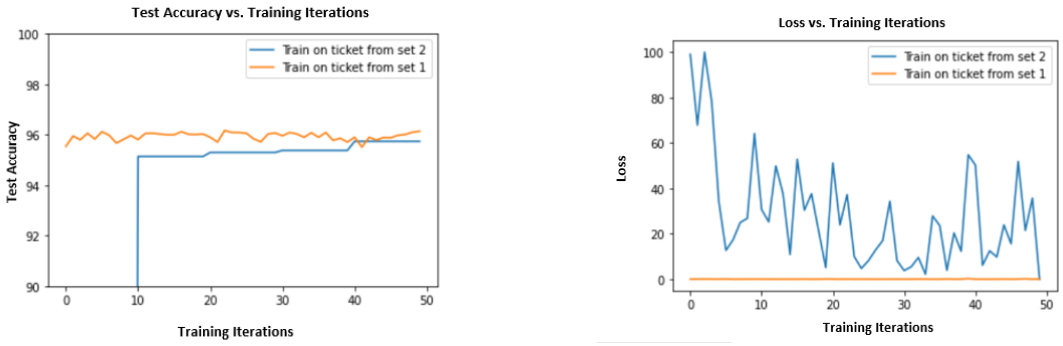
Now, we discuss the results that we obtained from the experiments performed post midterm review.

7.2 Post-Midsem:

As defined in Sec. 4.2, we looked at 3 tasks, the results of which are being stated below:

Task 1 Results

The plots corresponding to the test accuracy and loss against training iterations are given below:



(a) Test Accuracy vs. Training iterations

(b) Test Loss vs. Training Iterations

Figure 3: Task 1: Using MLP architecture on the MNIST data set

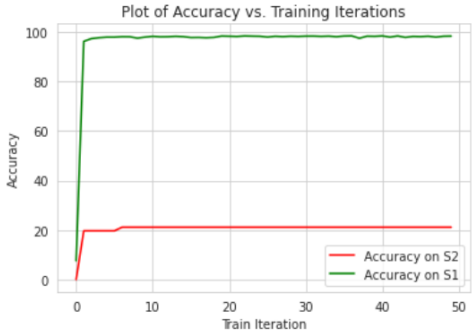
We observed the following:

- Test Accuracy on S_1 (one half of data containing all labels)= 95.74
- Observed Test Accuracy on S_2 (remaining half of data containing all labels) = 96.14

We conclude that the winning ticket T_1 trained over S_1 generalized well when trained over S_2.

Task 2 Results

The plots corresponding to the test accuracy and loss against training iterations are given below:



(a) Test Accuracy vs. Training iterations

(b) Test Loss vs. Training Iterations

Figure 4: Task 2: Using MLP architecture on the MNIST data set

We observed the following:

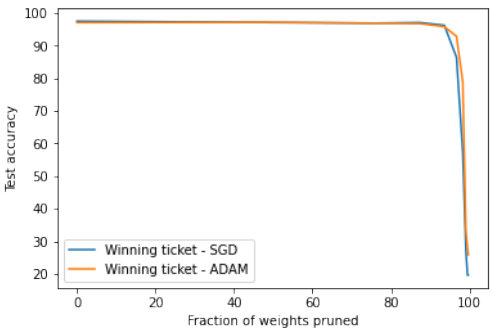
- Test accuracy of winning ticket (T_1) on S_1 (data set containing labels 0-4) = 97.23
- Test accuracy of T_1 trained over S_2 (data set containing labels 5-9): 21.1479

We believe that the model from S_1 being pruned significantly is not able to learn the features of the data set S_2 efficiently, and it results in such poor results.

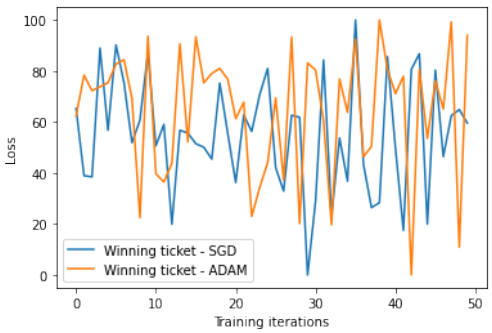
Task 3 Results

From the plots given below, we observed the following:

- Test accuracy of winning ticket on O_1(SGD) = 97.52
- Observed test accuracy of winning ticket on O_2(ADAM) = 97.17



(a) Test Accuracy vs. Unpruned Weights in the network



(b) Test Loss vs. Training Iterations

Figure 5: Task 3: Using MLP architecture on the MNIST data set

From the obtained plots and the observed test accuracies, we can conclude that the winning tickets are not overfit to the particular optimizer and hence winning tickets are optimizer-independent.

8 Future Work

Researchers at [1] shows that there exist winning tickets (small but critical sub networks) for dense, randomly initialized networks, that can be trained alone to achieve a comparable accuracy to the latter in a similar number of iterations. However, the identification of these winning tickets still requires the costly train-prune-retrain process, limiting their practical benefits. In new research [8], researchers discover for the first time that the winning tickets can can be identified at a very early training stage, which they term as Early-Bird (EB) tickets, via low cost training schemes (e.g., early stopping and low-precision training) at large learning rates. Their finding on the existence of EB tickets is consistent with recently reported observations that the key connectivity patterns of neural networks emerge early. Furthermore, they propose a mask distance metric that can be used to identify EB tickets with a low computational overhead, without needing to know the true winning tickets that emerge after the full training.

9 Conclusion

In the first half of the project, we looked into the works done in [1], where they proposed a novel method of finding winning tickets (which are sparse, trainable subnetworks of the original dense network), which gives commensurate accuracy in comparable number of iterations when trained in separation as well. We demonstrated performance of subnetworks at different pruning levels and found that the subnetworks at a specific pruning level perform better than the original network. However, the caveat was the severe complexity of obtaining these winning tickets and their generalizability.

In the second half of the project, we tried to look into the caveats and existing literature on addressing them. We looked into the works done in [7] where the authors proposed the idea of using the winning ticket obtained from one data set on another dataset having similar distribution. We considered 3 subtasks along this line of work, and summarized the results obtained in the Results section (Sec. 7).

Finally, a critical question remains unanswered: what makes winning tickets special? While our results shed a vague light on this by suggesting that whatever makes these winning tickets unique is somewhat generic, what precisely makes them special is still unclear. Understanding these properties will be critical for the future development of further research inspired by lottery tickets.

References

[1] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019.

[2] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Stabilizing the lottery ticket hypothesis, 2020.

[3] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks, 2015.

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

[5] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 598–605. Morgan-Kaufmann, 1990.

[6] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and H.P. Graf. Pruning filters for efficient convnets. 08 2016.

[7] Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers, 2019.

[8] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks, 2020.