Group Report For Final Project

Donghao Li
Department of Mathematics
HKUST
dlibf@ust.hk

Jiamin Wu
Department of Mathematics
HKUST
jwubz@ust.hk

Wenqi Zeng
Department of Mathematics
HKUST
wzengad@ust.hk

Yang Cao
Department of Mathematics
HKUST
ycaoau@ust.hk

1 Introduction

Nowadays, Deep learning have achieved surprising performance in many different tasks. However, researchers still have limit understanding on why it works. A recent article [6] put forward a new perspective on understanding neural networks. Specifically, they studied the correspondence between the teacher node and the student node in the teacher student framework. This will help us further understand the principles of neural networks. We thought this study was very interesting, so we first checked his theoretical proof, repeated some key experiments, and designed additional experiments to verify its conjecture.Finally, we found that when the learning rate is small, the experimental results meet the conjecture of the above article, but when the learning rate is large, the optimization algorithm can jump out of initialization, so initialization is not so important, which is in line with our intuition.

2 Related work

2.1 Open problems for deep learning

There are some open problems of deep learning, which could help us establish deeper insight on why it works. Here we list some of the problems:

Over-parameterization do not overfit but lead to better generalization? In traditional machine leanring theory, we should carefully keep balance between bias and varance of a esitmator, which corresponds to under fitting and over fitting. A model with zero training loss would overfit the training data therefore performs badly on the test set. However, in deep learning practice, the number of learnable parameters is much more than total training samples. And also we can reach zero training loss with over-parameterization. However, it does not hurt generalization and over-parameterization always leads to good generalization. Why that happens?

How can Stochastic Gradient Descent(SGD) algorithm sovle such a complicated non-convex optimization problem and find a good solution can generalize well? SGD is a quite simple optimazation algorithm while training neural network is a highly non-convex problem. How can SGD with over-parameterization achieve zero training loss easily in practice? Also, there are many global optima for the neural network, including many with terrible generalization power. We can design an algorithm to find a network with zero training loss but have large test loss.

Moreover, why we can find a sparse sub-network which have similiar or even better performance that a dense network? Why do lottery tickets sub-network exist?We will discuss this later.

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

2.2 Teacher student network framework

Teacher student framework is used for simulating training neural networks. In this framework, teacher is a neural network with fixed parameters. Then we want to train a student network to fit the teacher network. Specifically, we sample input X from some distribution and then use the teacher network to produce label Y. We use (X, Y) pair as data to train the student network.

Compared to using real world dataset, there are some benefits of using teacher student framework. First, since we know the parameter of teacher, we know that there exists a global optimal whose training loss and test loss are zero. Second, we can generate infinite data from this setting, which separates the finite sample issues from induction bias in the dataset. Finally, it could help theoretical analysis and lead to better generalization bound.[6]

2.3 Lottery ticket hypothesis

Lottery hypothesis [1] states that "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.". It gives a simple method to find sparse subnetworks and verifies it with sufficient experiments. It is shocking that this simple method has obvious effects, not only can reduce the parameters by 10-100 times, but also the training speed of the subnetwork is faster. It only gives the empirical results, without further theoretical analysis, which has set off a wave of discussion. Then [2] gives a trick to further develop this hypothesis to large scale applications like ImageNet dataset. The trick is called rewinding which retrain starts from early epoch parameters instead of initialization values.

Also, these is a paper trying to do more ablation study on this phenomenon, it deconstructe all the factors and found that setting weights to zero is important and only keep the sign of the mask can lead to good retrain results. They also found the exist of supermask, which can be applied to an untrained, randomly initialized network to produce a model with performance far better than chance (86% on MNIST, 41% on CIFAR-10)

Other group of researchers focus on the generalization power of winning ticket. [5] found that the winning ticket found on one dataset can generalized on different datasets and different optimizers. Moreover, winning tickets generated using larger datasets consistently transferred better than those generated using smaller datasets.

Later, some researchers found the phenomenon appears not only in computer vision (image classification) domain, but also in natural language processing and reinforcement learning. That suggests that this phenomenon could be nature of neural network.

However, there are also different opinions. [3] state that only the structure of the network is important in network pruning tasks. The initialization of the network is not important. They also conduct experiments to compare with the "Lottery Ticket Hypothesis". And found that with optimal learning rate, the "winning ticket" initialization as used in Frankle Carbin (2019) does not bring improvement over random initialization. It indicates the learning rate used in lottery hypothesis is too small, large initial learning rate could jump out from initialization and find better solutions.

The reason for the existence of lottery tickets has not been answered so far, but analyzing this phenomenon may be of great help to further our understanding of neural network principles.

2.4 Luck matters

[6] provide a new view to understand open questions about deep learning. They state that "During optimization, student nodes compete with each other to explain teacher nodes. And lucky student nodes which have greater overlap with teacher nodes converge to those teacher nodes at a fast rate, resulting in winner-take-all behavior." Then they think that only those luck nodes are important and other would go to zeros during training process. Then it gives intuitive explanations to those open problems. Since over-parameterized student network would have more nodes closed to teacher network, it would help the training process. But, the "winner-take-all" behavior would aviod overfitting. It can also explan lottery ticket hypothesis because those winning tickets are corresponding to lucky student nodes.

They give theoretical guarantee of their framework and we checked them and found there is something still need to be improved. In the beginning of their work, when they are trying to figure out the gradient dynamics, the assumption they use could be a little bit strong.

The notation setting of that article is as the following. Denote the input of student network to be x. For each note j, denote $f_j(x)$ as the activation, $f'_j(x)$ as the ReLU gating, and $g_j(x)$ as the backpropagated gradient. They use $^\circ$ to represent a teacher node. And they use ω_{jk} to represent weight between node j and k.

Their assumption is

$$E_x[\beta^*_{jj^\circ}(x)f'_j(x)f'_{j^\circ}(x)f_k(x)f_{k^\circ}(x)] = E_X[\beta^*_{jj^\circ}(x)]E_X[f'_j(x)f'_{j^\circ}(x)]E_X[f_k(x)f_{k^\circ}(x)] \quad (1)$$

However, in their Theorem 1, they prove that all student nodes follows

$$g_j(x) = f'_j(x)[\sum_{j^\circ} \beta^*_{jj^\circ}(x)f_{j^\circ}(x) - \sum_{j'} \beta_{jj'}(x)f_{j'}(x)] \quad (2)$$

We know that $g_j(x)$ is highly related with $f_{j^\circ}(x)$ and $f_j(x)$. Specially, if j is at the top layer $g_j(x) = f_{j^\circ}(x) - f_j(x)$. Thus, since Equation 2 holds, Equation 1 would seem to be a little bit strong. However, Equation 1 seems to be an important tool to simplize the problem, thus I think future work could be done here to let the result to be even more convincing.

3 Reproducing key results

We will first reproduce the key results of related papers, which can help us determine whether the experimental results are credible and verify our experimental code. And in the experiments we will get a deeper understanding, dig out hidden information to help us design new experiments.

3.1 Over-parameterization helps optimization

In this part we want to reproduce Figure 2 Bottom in [4] to show the benefit of over-parameterization.

Here is detail of experiments. Teacher network: A randomly initialized two-layer neural network with 60 hidden neurons, the output layer is 10-dimensional, and the activation function is ReLU. Student network: It is also a two-layer neural network with n * 60 hidden neurons, the activation function is ReLU, and the output layer is 10-dimensional. In the experiment, MSE was used as the loss function, and SGD was used for optimization. The momentum parameter was 0.9, weightdecay was 0, and batchsize = 256. Each batch was resampled and inputed through the teacher network. The reported MSE is the value in training, but since the data is regenerated each time, it can be considered as test MSE.

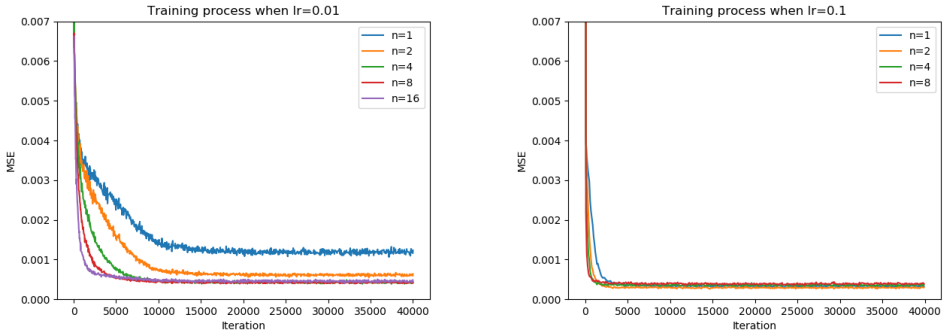
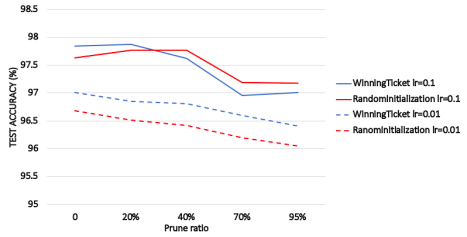


Figure 1: Reproduce results of Figure 2 Bottom in [4]. Left is training process with learning rate 0.01. It shows over-parameterization helps speed up training speed and get smaller loss. Right is training process with learning rate 0.1. It shows over-parameterization can not lead to better generalization



Learning Rate	Dataset	Model	Prune Ratio	WinningTicket	Random Initial
0.1	MNIST	fc	0	97.84	97.63
			20%	97.88	97.77
			40%	97.62	97.77
			70%	96.96	97.19
			95%	97.01	97.18
0.01			0	97.01	96.68
			20%	96.85	96.51
			40%	96.81	96.42
			70%	96.6	96.2
			95%	96.41	96.05

Figure 2: Test Accuracy for Full Connected Network

It can be seen from the picture that when lr is small (0.01), we get similar conclusions as in the paper. Over-parameterization can make the training process easier, the training speed is faster, and the resulting solution has smaller loss. But when the learning rate becomes larger (0.1), we find that no matter what value of n, the convergence speed and the final value are better than when the learning rate is small. At the beginning of training, we found that parameterization improved the training speed, but did not show the advantage when it was converged. When n> 2, loss is slightly affected by over-parameterization.

3.2 Reproduction and rethinking of the lottery ticket hypothesis

In lottery ticket hypothesis by Frankle (2018)[1], authors prune parameters to 10 to 20 percent compared to original models. To verify and rethink the results of Frankle (2018)[1], we read the paper by LIU (2018)[3] and reproduced the results. The dataset used to do reproduction is MNIST, which consists of 70,000 hand writing digital figures from 0 to 9. In the paper by LIU (2018)[3], methods of VGG16 and ResNet50 have already been tested. As a result, we do the test of fully connected neural network for unstructured pruning. By the process of reproduction, what we want to verify is whether the parameter of learning rate affecting the results get by Frankle (2018)[1] or not and how it affecting. Therefore, we do the experiments with learning rate being 0.1 and 0.01 on winning ticket and random initialization, respectively. The following table and figure show the result we get.

As we can observe in the figures2, the blues lines represent the test accuracy on winning ticket hypothesis with different prune ratio while the red ones are for random initialization. The solid lines denote the result with large learning ratio 0.1 and the dot lines means the relatively small learning ratio equals to 0.01. First, from the graphs2, test accuracy with higher learning ratio performs better. Second, when learning ratio being small, that is 0.01, winning ticket hypothesis is more useful than random initialization. Nevertheless, when learning rate getting larger, equaling to 0.1 in our experiment, the difference is not such obvious.

According to LIU (2018)[3], for the structured pruning, model trained from random initialization can produce results as good as winning tickets. As a result, achitecture but not weights is more important during the pruning. From the result we reproduced, parameters actually affecting the results and performance. The parameter of learning ratio in the paper by Frankle (2018)[1] is more or less by testing or choosing empirically, which may be hard for lower level users. Frankle (2018)[1] also state in their paper that under low learning ratio, the lottery ticket hypothesis performs better.

4 Experiment design

We have designed several comparative experiments on the basis of teacher-student-net.

First we want to prove that one-shot pruning is consequential. The method for one-shot pruning is to sort the pre-trained model weights by descending absolute value and set the smaller ones to 0 to achieve the corresponding sparse ratio. At the same time, another set of weights whose certain positions are randomly set to 0 by the same sparse ratio will act as a baseline. The expected performance of randomly sparseness will be far worse than one-shot pruning. On the other hand, we use the dense model as a benchmark to verify the lottery ticket hypothesis.

Our second goal is to verify whether the lottery network hypothesis is established in the teacher student framework. If the luck matters article is correct, according to the explanation in the article, the lottery network hypothesis will hold. We will experiment by decomposing the conditions assumed by the lottery network. Specifically, we will compare several cases.

(a) we keep initialization of student network and teacher network.

(b) we keep initialization of student network but reinitialize teacher network.

(c) we reinitialize student network but keep teacher network.

(d) we reinitialize both student network and teacher network.

We expect that if the conclusion of " Luck Matters: Understanding Training Dynamics of Deep ReLU Networks " holds, then the results of (a) will outperform the others and make the model work well because reinitialization would destroy the correspondence between teacher and student networks. However, if the conclusion of "Rethinking the Value of Network Pruning" prevails, who criticized the luck matters theory, then the results of the above four settings should be similar.

Since the initial learning rate may have a large impact on the lottery ticket hypothesis, we compared the larger learning rate and the smaller learning rate in the experiments.

5 Results and analysis

We first compare one shot pruning proposed in [1] and random pruning, the results is in 3.

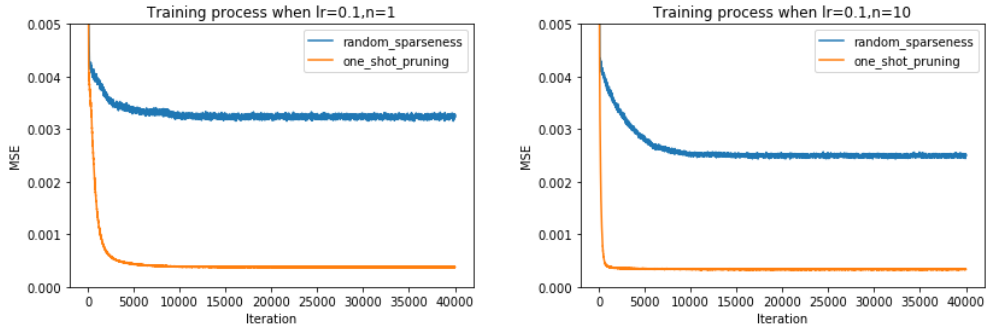


Figure 3: We first test if one shot pruning is better than random pruning, we find that regardless of the student net size and learning rate, the one-shot pruning is always significantly better than the randomly selected ones with an overwhelming performance. This indicates that one-shot pruning is applicable and meaningful.

Then we compare differnt pruning rate and dense model, the results is in 4.

In Figure 5 when learning rate = 0.01, it can be seen that (a) which retained both pre-trained networks is better than partially retained situations (b) and (c), and the latter two surpass (d) who does not retain parameters at all. The experimental results now is consistent with " Luck Matters: Understanding Training Dynamics of Deep ReLU Networks ".

When learning rate = 0.1, the anaylsis above is not applicable. All the loss values is very closed and much more batter than small learning rate case. So we can say that large learning rate could help the optimizer jump out of the initialization and find a better solution. When n=10, we find that the results obtained are surprisingly divided into two groups. The upper group consists only keep student initialization and keep none. The lower group consists the rest. We speculate that such a gap is caused by whether to change the teacher net, which causes completely different label y and therefore affect the loss.

To sum up, we divide the factors that affect the performance of the model into two. The first is due to the impact of initialization on model capabilities, and the other is due to the impact of teacher changes. When the learning rate is small, the first factor dominates, and when the learning rate is large, the initialization impact decreases, and the second one dominates.

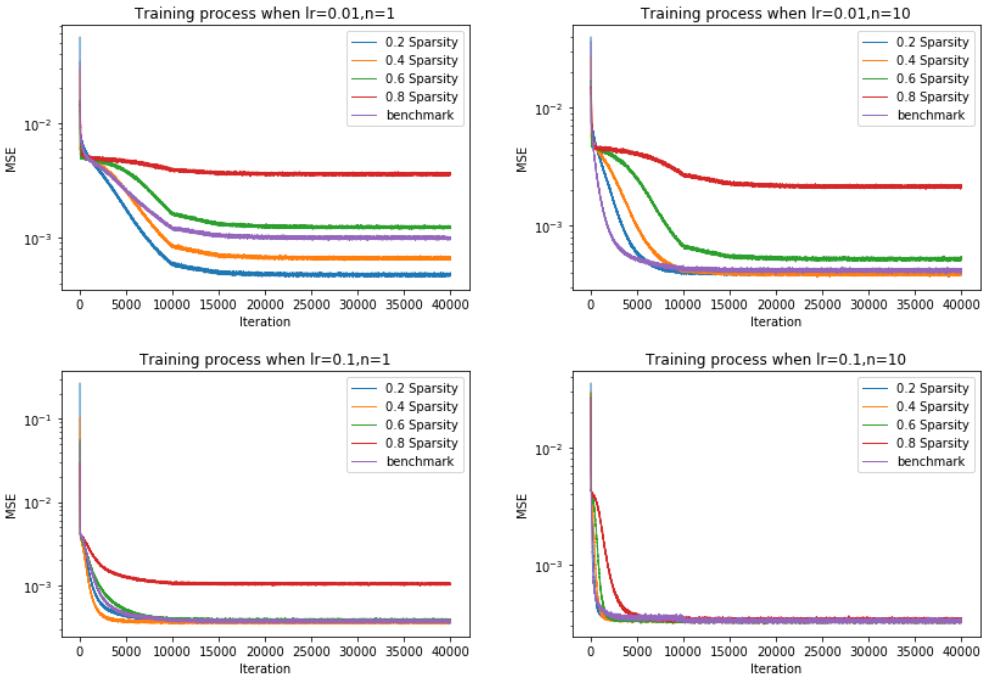


Figure 4: These figures shows training process in different settings. Benchmark shows dense model performance. From the figure we can find three phenomena. The first is that sparse networks can achieve the same performance as dense networks, and even achieve better performance. The second is that the training rate is faster and the end result is better when the learning rate is large. When the learning rate is small, the model performs poorly, but the sparse network has a greater advantage than the dense network. The third is that the over-parameterized model outperforms the network of the same size as the teacher in different degrees of sparsity.

6 Conclusion

In this report, we review work with sparse networks and attempts to explain neural networks, and reproduce key results from the paper. In addition, we also independently designed and implemented some experiments to try to analyze the impact of learning rate on initialization. We found that when the learning rate is small, the initialization has a greater impact on model training. Specifically, the student initialization includes some nodes that overlap with the teacher node and are important for training the neural network. However, a larger learning rate can help the optimization algorithm jump out of initialization and achieve better results.

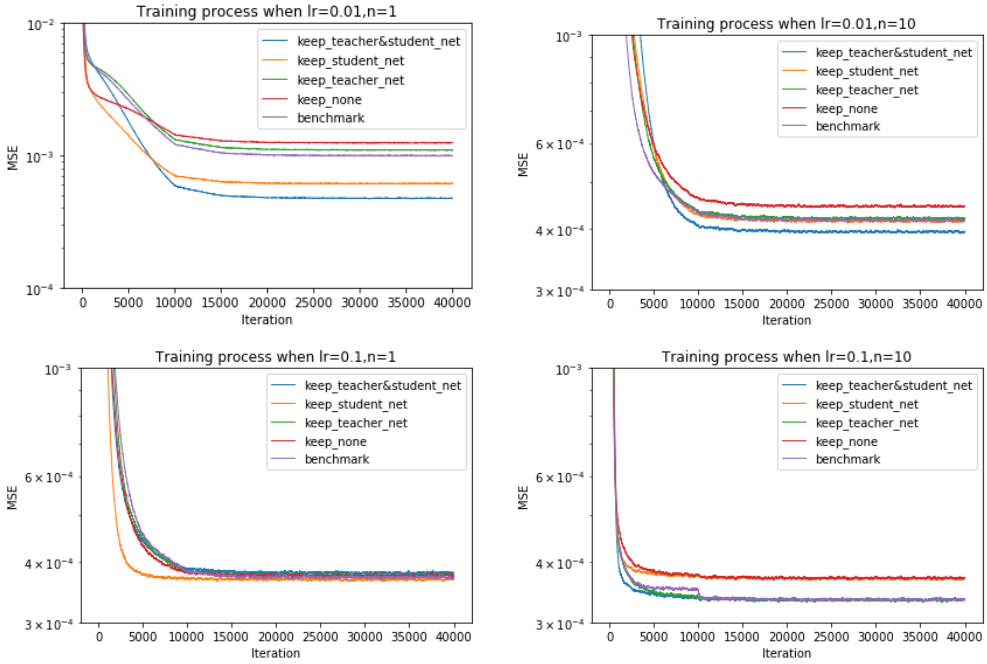


Figure 5: These pictures show the effect of different initializations on model performance when we save 80% of the parameters.

7 Contribution

Donghao Li:Introduction, Related Work, Reproducing overparameter.

Jiamin Wu:Reproducing lottery hypothesis.

Wenqi Zeng:Experiment Design and Results and analysis.

Yang Cao:Theoretical proof check.

References

[1] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2018.

[2] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Stabilizing the lottery ticket hypothesis, 2019.

[3] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *ArXiv*, abs/1810.05270, 2018.

[4] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks, 2014.

[5] Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *ArXiv*, abs/1906.02773, 2019.

[6] Yuandong Tian, Tina Jiang, Qucheng Gong, and Ari Morcos. Luck matters: Understanding training dynamics of deep relu networks, 2019.