A Two-regime Model Of Network Pruning: Metrics And Scaling Laws

Yefan Zhou
Steven Gunarso
Chen Wang
Yaoqing Yang
Michael Mahoney

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-51
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-51.html

May 1, 2023

Acknowledgement

We want to thank Joe Zou, Alan Pham, Arin Chang and Zhuang Liu for helpful discussions and valuable feedback.

A TWO-REGIME MODEL OF NETWORK PRUNING: METRICS AND SCALING LAWS

CAPSTONE FINAL REPORT

Yefan Zhou
Master of Engineering in EECS
University of California, Berkeley
yefan0726@berkeley.edu

Steven Gunarso
Master of Engineering in EECS
University of California, Berkeley
y_steven_gunarso@berkeley.edu

Chen Wang
Master of Engineering in EECS
University of California, Berkeley
chw@berkeley.edu

Yaoqing Yang
Capstone Advisor
Department of EECS
University of California, Berkeley
yqyang@berkeley.edu

Michael Mahoney
Capstone Advisor
Department of Statistics
University of California, Berkeley
mmahoney@stat.berkeley.edu

May, 2022

EXECUTIVE SUMMARY

Due to the increasing memory and computation cost of large-scale deep neural networks, there is a new trend in the AI community to design principled network compression approaches. In this Capstone Project, we find a dichotomous phenomenon of a widely-studied compression approach, network pruning, which removes unimportant network weights while maintaining the test-time performance of neural networks. We find that, depending on the relative size of the pruned weights and the training data, early stopping the training process before pruning can significantly help or harm the test-time performance of pruned models. Having shown the existence of this dichotomous phenomenon on both computer vision (CV) and natural language processing (NLP) benchmarks, we study the following two questions to quantify the phenomenon. (1) How do we determine the transition line between using early stopping or training to convergence (in the two-dimensional space of data size and pruned model size)? (2) If we need early stopping, how do we determine the optimal early stopping time? We show that metrics derived from the recently proposed Heavy-Tailed Self-Regularization (HT-SR) theory and the analysis of neural network loss landscapes can answer these questions. In particular, we show that the `rand_distance` metric, which measures the distance in the spectral domain between trained and randomly initialized weights, can be used to determine the early stopping time. We also show preliminary results that the model similarity, a newly proposed metric related to the loss landscapes of neural networks, can determine the transition line between using early stopping or training to convergence.

Contents

Part I
Project Background

1 Context

Before diving into the technical details, we briefly discuss in Part 1 the application context of our research to show where we can apply our research outcomes and where our research can be the most useful.

In recent years, we have witnessed that neural network models are increasingly used to solve complex real-world problems. For example, the BERT model [4], which proposes a new pretraining-fine-tuning pipeline for large-scale Transformer-type models, has significantly improved the ability of natural language processing (NLP) models, and it has revolutionized the NLP field. After the BERT model was released, most of the current machine translation engines started to use large transformer models. For the common computer vision (CV) tasks, such as image object detection [31], image object recognition [7], and image classification [15], various computer vision models have helped accomplish human-level performance. In addition, deep neural networks for speech recognition [24] also brought huge improvements in applications, such as automatic movie captioning [3] and faster voice input [9].

Deep neural networks are also used in mobile applications and portable devices. With the latest version of iOS, the texts in the photos stored on the device can be extracted automatically within a second. This feature does not need Internet connection, and the entire character recognition algorithm runs on the local machine. Some mobile phones utilize image augmentation techniques for deep learning models to enhance the image quality of the photos [13]. There are also many other application scenarios of deep learning on portable devices, especially in the past five years.

However, despite the widespread applications of deep neural networks on portable devices, their high cost and massive energy consumption have always been an issue. In order to make it possible for a neural network model to run on portable devices, researchers proposed neural network pruning to make the network smaller [27, 26]. However, the process of incorporating network pruning into the model's training/inference pipeline is challenging due to the nature of the problem, i.e., the performance of pruning is sensitive to different training settings like the model size, training data size, the learning rate of the model to be pruned, and the hyperparameters of pruning algorithms.

Due to the large hyperparameter space, achieving optimal performance for network pruning is an open research problem. An ad-hoc pruning setting is unlikely to produce optimal performance of the pruned model. We also recognize that the objective function when training the large, dense model is to minimize the training loss for the pre-trained model, which is not necessarily optimized for the performance of the pruned model. One of the possible solutions to this problem is to perform a grid search on the different hyperparameters and training settings to find the combination that will result in the best pruning results. However, this process is highly costly as it requires repeating the computationally expensive training-pruning-fine-tuning cycle for many iterations. There is a need for a cost-effective method to select the optimal training settings and hyperparameters.

Our research proposes the two-regime model (see Part II) on network pruning to provide a divide-and-conquer way of improving network pruning. By carefully examining the optimal settings of network pruning, our research builds a general framework to evaluate network pruning and diagnose the failure of pruning. Depending on specific goals and constraints of pruned model size, the framework serves as a guidance to determine the best setting to improve pruning performance.

2 Market Insights

There are a broad range of applications of deep learning models for portable devices. However, portable devices have many limitations in computational power, RAM, and battery that make running larger models costly or prohibitive. Large dense neural network models put high demand and constraints on the portable devices. For example, the feature map [22] generated by model inference requires large amounts of memory. However, the RAM of a portable device may only be less than 1GB, which means there will be no room to run other processes if we want to run a deep learning model on a portable device. Even the most advanced iPhone has a RAM of 6GB, which means it could not load a giant neural network model while running many other applications. Nevertheless, some applications require deep neural networks to make consecutive model predictions in real-time. For example, voice recognition software is used to generate texts according to the input sound and is a typical usage scenario of neural network models. Such software can work quite well when the workload is light, but can be disastrous in heavy use cases. For example, using the deep

learning-based virtual background technology [23] on portable devices for long hours may drain the battery much faster than using the camera alone. When we perform consecutive predictions, it will burden the CPU and battery [8].

We can use network pruning to trade the model's performance for efficiency, which allows portable devices to effectively use these models. However, given the constraints of the device, we have to search for the best configuration to maximize the tradeoff between model's performance and efficiency. To do this, industry engineers typically utilize a grid search method, which takes enormous time and computational resources [1].

Searching for optimal pruning configurations could be expensive. Consider a scenario where a deep learning researcher wants to search for the best combination of the learning rate, batch size, and early stopping epoch to train the large models and then prune them. Suppose we have 100 different (learning rate, batch size) combinations and 100 early stopping checkpoints for a deep learning model and have three different compression techniques for each of the early stopping checkpoints on each hyperparameter combination. The price for renting an AWS p3 machine is $3/hour. If a model takes 1 hour to train, prune, and fine-tune fully, it will cost $90,000 of compute time to search for all 30,000 combinations. This shows that pruning a model needs careful large-scale studies to improve the performance, which is daunting for the research engineers and even small enterprises.

Considering the resource intensive processes of both training deep learning models and running inference on devices in real-time, our work in deep learning model pruning can save high costs on both developers' and users' sides.

Part II
Technical Project Outcomes

3 Overview

In this work, we propose the two-regime model on network pruning: early stopping and training to convergence. Early stopping means that we stop the training early short of convergence during the training stage. Training to convergence means that we train the model to minimize training loss within an error range during the training stage. Depending on the various problem configurations of network pruning, either early stopping or training to convergence is more useful to improve the prediction performance of the pruned model. These are the two different regimes of network pruning that we consider.

In this part, we first show the existence of the two regimes with our experimental results on the CV and NLP models. Then, we utilize HT-SR and loss landscape metrics to quantify the two regimes. These metrics utilize the numerical values of the weights in a neural network to determine the characteristics of the model. The percentage of parameters that are removed is called the prune ratio. In this paper, we identify heavy-pruning as the regime where early stopping is optimal in a model compression setting and light-pruning as the regime where training to convergence is optimal. The open problem is to determine the transition line between the two regimes in a combination of hyper-parameters and the early stopping epochs.

4 Related Work

4.1 Pruning

One of the most popular model compression techniques is pruning. This technique removes unimportant connections (weights) in a neural network based on specific pruning criteria [19]. There are many different aspects that factor to the method of pruning, such as structure, scoring, scheduling, and fine-tuning.

Different pruning methods may affect the structure of the weights and the sparsity of the neural network. Unstructured pruning zeroes out the connections between neurons (weights), which effectively makes it a sparse network [1]. There are also different ways to score or select which weights to prune. The most common scoring technique is scoring the weights based on its magnitude. In this research endeavor, we are mainly focused on unstructured weight-level magnitude pruning [10], where for each layer weight matrix, a certain percentage of the parameters are removed based on its magnitude.

4.2 Early Stopping For Pruning

Most prior pruning work [19, 16] typically trains the model to convergence and then prune. This approach is based on the assumption that the well-trained model can still preserve its superior performance after pruning. However, the assumption is not rigorously or empirically proved by any prior work. Our findings in this work show contradictory results to the assumption, i.e., one can stop the training stage early to get better pruned models. There is a recent study [18] proposing that early stopping improves the pruning performance of initially trained models with large sizes. Although it gives a valuable heuristic on early stopping, it is still unclear when early stopping is necessary and when to apply early stopping to reach the optimal pruning results, which are questions that our study addresses. Another study [30] proposes a metric that measures the similarity between pruned sub-network architecture in consecutive training epochs to indicate when to stop training and start to draw the "lottery tickets" [6]. However, our work answers the question of whether early stopping benefits pruning.

4.3 Generalization Measures

As noted earlier, there is a need for machine learning practitioners to gain valuable insights before any experiments or any computationally expensive tasks are performed. Generalization is a model's ability to react to data outside of the training set. Generalization measures attempt to characterize some aspects of generalization through evaluating the properties of the trained model, the optimizer, or the training set. These analyses are often drawn from empirical correlations and do not necessarily prove causation [11]. In this research, we use two generalization measures that can represent a model's innate characteristics and evaluate its ability to generalize across different training settings. These settings may vary in terms of model architecture, dataset size, or hyperparameters.

Empirical Spectral Density (ESD) is the eigenvalues' distribution of the layers' weights of a model. We know from the Heavy-Tailed Self-Regularization theory (HT-SR) [20] that well-trained, well-correlated models can be fit into a (truncated) power law distribution where their distribution of eigenvalues are different than a model with randomized weight matrices. By evaluating the similarity between the ESD of a trained model with the ESD of a random model for each layer's weight matrix, we can measure how non-random the trained model is. Ideally, the more information a neural network learns through training, the less random its ESD should be.

Another type of generalization measures we consider is obtained from the loss landscape of a model. Loss landscape is the representation of the loss function around the weight space of the network [17]. The shape of the loss landscape is dependent on the weights or parameters of the network. As a pre-trained model is pruned, its loss landscape changes as its weights are perturbed. This weight perturbation often decreases the quality of the loss landscape [1], where training models to convergence will often find very different global minimas [28]. By evaluating the loss landscape at every prune ratio, we can infer the early stopping epoch during training to achieve optimal performance.

5 Two-Regime Model of Network Pruning

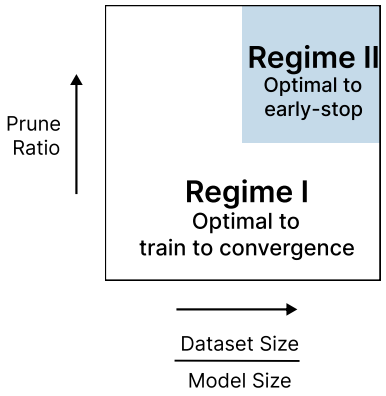


Figure 1: Caricature of the two-regime model of network pruning. Regime I: training a model to convergence is optimal to achieve the best pruned model performance. Regime II: applying early stopping during training is optimal to achieve the best pruned model performance.

We propose a two-regime model to answer the question if early stopping is needed in network pruning. Furthermore, we are interested in the following three questions that rise to the two-regime model.

1) When do we need to do early stopping?

2) How do we determine the transition line on the space of prune ratio between the two regimes?

3) How do we determine the optimal early stopping epoch during training in Regime II?

5.1 Definition and Preliminaries

We define our two-regime model as the distinction between different training settings within the context of network pruning, which is pictorially represented in Figure 1. Regime I represents settings where training a neural network model to convergence is optimal to achieve the best pruned model performance. Regime II represents settings where applying early stopping when training a large model will result in the best pruned model performance. One of the main objectives of this research is to identify the conditions where a machine learning setting belongs in either Regime I or Regime II. Our findings suggest that there are two conditions which have to be satisfied for a training setting that is optimal when applying early stopping. These conditions are:

1) Model is **heavily pruned**

2) Model is trained with **a smaller model size** and **more data**

[1] It is well-known that the loss landscape becomes less "connected" when the network size becomes smaller [5, 28]. Therefore, heavy pruning can lead to worse connectivity of the loss landscape.

While the exact prune ratio, model size, and data size that determines the transition between the two regimes can vary by model architecture and other factors, our findings show that the two-regime pattern is consistent in all of the experiment results we collected.

5.2 Metrics

We also provide useful metrics to quantify the conditions when we should apply early stopping and at what step to stop training in the two-regime model.

To determine the transition line between the prune ratios where we should apply early stopping as opposed to training to convergence, we propose to use model similarity as defined in the equation below.

Model similarity is calculated by taking an already-pruned model and fine-tuning that model with two different seeds [12]. The predicted probability distributions between the two resulting models are then compared for its similarity. Intuitively speaking, model similarity represents the distance between the position of the models with respect to the loss landscape [28].

$$\text{Model Similarity} = \frac{1}{N}\sum_{i=1}^{N} \text{JS}\left(\mathbf{d}_i, \mathbf{d}_i'\right). \tag{1}$$

Here JS represents the Jensen-Shannon Divergence to calculate the similarity between two probability distributions [11]. The two distributions $\mathbf{d}_i$ and $\mathbf{d}_i'$ represent the predicted probability distributions of a pruned model fine-tuned with one seed and of the same pruned model fine-tuned with another seed on the i-th data sample in the training set. N is the number of training samples. We calculate this Jensen-Shannon Divergence on the whole training set.

We also propose to use `random_distance` as an indicator to determine the best early stopping epoch to stop training.

$$\texttt{random\_distance} = \frac{1}{L}\sum_{i=1}^{L} \text{JS}\left(\mathbf{p}_i, \mathbf{p}_i^{rand}\right). \tag{2}$$

Here JS represents the Jensen-Shannon Divergence. Consider a trained model with L weight matrices. The notation $\mathbf{p}_i$ represents the ESD obtained by normalizing the squared singular values of the i-th weight matrix $\mathbf{W}_i$, and we use $\mathbf{p}_i^{\text{rand}}$ to denote the distribution obtained in the same way but using the randomized weight matrix. We calculate this Jensen-Shannon Divergence for every layer weight matrix and take the mean as the `random_distance` score [29].

6 Empirical Results

We first provide details of the experimental setup. Then, we provide empirical results to show the existence of the two-regime model. Finally, we show how to use generalization metrics to quantify the two regimes.

6.1 Experimental Setup

We conduct experiments for both CV and NLP tasks. In the CV experiments, we use CIFAR-10 dataset [14] and the PreResNet architecture implemented in [19]. For the standard setting in Figure 2, 3a and 6, we use the full train set and PreResNet-20. For scaling experiments in Figure 4 and 5, we study four cases with subsampling {1, 5, 10, 100} % of CIFAR-10, and four cases with PreResNet-{20, 38, 56, 110} architectures. For each experiment, we train the model for 160 epochs with a batch size of 64. We use an SGD optimizer with an initial learning rate of 0.1, learning rate decay by 0.1 at epoch 80, 120, momentum 0.9, and weight decay 1e-4. We save the checkpoints at every 10 epochs during training. For each setting in CV, we train and prune a model for one seed, fine-tune each pruned models for three random seeds, and report the average classification accuracy on the test set.

On the NLP side, we consider the machine translation dataset WMT14 [2] German to English corpus. We use the original "base" transformer architecture from [25] with 6 layers, 8 attention heads per layer, and an embedding dimension of 512. We train the model from scratch on a subsampled dataset of 1.28 million sentence pairs for 19 epochs. We use a batch size 1500 and an Adam optimizer with an inverse square-root learning schedule, dropout probability of 0.1, and 10% label smoothing. We save the checkpoints at every epoch during training and report BLEU scores on the validation set.

For pruning method, we use unstructured weight-level magnitude-based pruning [10] on CV and NLP models with a uniform layerwise sparsity [32], i.e., each layer is pruned to have identical layerwise sparsity levels. Specifically, we only prune weights in convolution layers in CV models following the setting in [19] and only prune weights in linear layers (transformer layers and prediction layer) in NLP models. After pruning, we fine-tune the CV models for 40

epochs with learning rate 1e-3 and fine-tune the NLP models for 5 epochs with the same learning rate schedule as training.

For generalization metrics, we use the open-source `WeightWatcher` tool [21] to measure the `random_distance` of the ESDs. When measuring the model similarity, we fine-tune each pruned model for three random seeds and average the score across the three pairs.

6.2 Existence of Two Regimes

We provide results to show the existence of the two-regime models. By evaluating the results by changing prune ratio and training data/model size, we are able to examine the transition between the two regimes along two axis: prune ratio and training data/model size. Based on the transition between the two regimes, we demonstrate that early stopping is necessary for optimal pruning performance when the model is heavily pruned and trained with more data and smaller model size.

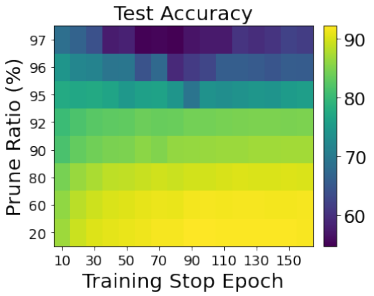


Figure 2: Test accuracy of pruned models trained with varying epochs and pruned with varying prune ratios. Models are trained with PreResNet-20 on CIFAR-10.

Illustration of Result Format Figure 2 shows a 2D diagram we get from the experiment, using PreResNet-20 on CIFAR-10. The x-axis represents how many epochs this model gets trained. In the CV experiment, we train a model for 160 epochs and save the model every 10 epochs. The y-axis represents the prune ratio and we apply the same pruning algorithm with these ratios to all the saved models during training. Therefore, each pixel in this 2D diagram represents a single experiment, i.e., a model is trained for x epochs, pruned $y\%$ of parameters, and then fine-tuned. Note that this should be distinguished from the setting when the entire row of the 2D diagram represents a training run. We plot the test accuracy of the pruned models after fine-tuning in the 2D diagram. We consider two quantities to differentiate the regimes: (1) the optimal stop epoch in each prune ratio. (2) the difference between the optimal epoch and other epochs. The first quantity is important because we use that to differentiate the two regimes, i.e., the regime is defined based on whether early stopping is necessary. The second quantity is also important because it affects our perception of the regime diagram. To better visualize the two quantities, we use a normalization scheme where we subtract the grid values in each row by the value of the optimal stop epoch for each prune ratio[2]. In addition, we annotate the optimal epochs in each row using red star markers. Figure 3a shows 2D plot after the normalization and annotation.

Transition between Regimes on Prune Ratio We show the transition between regimes on prune ratio in a fixed dataset and model size setting. This transition illustrates that we should do early stopping when the model is heavily pruned.

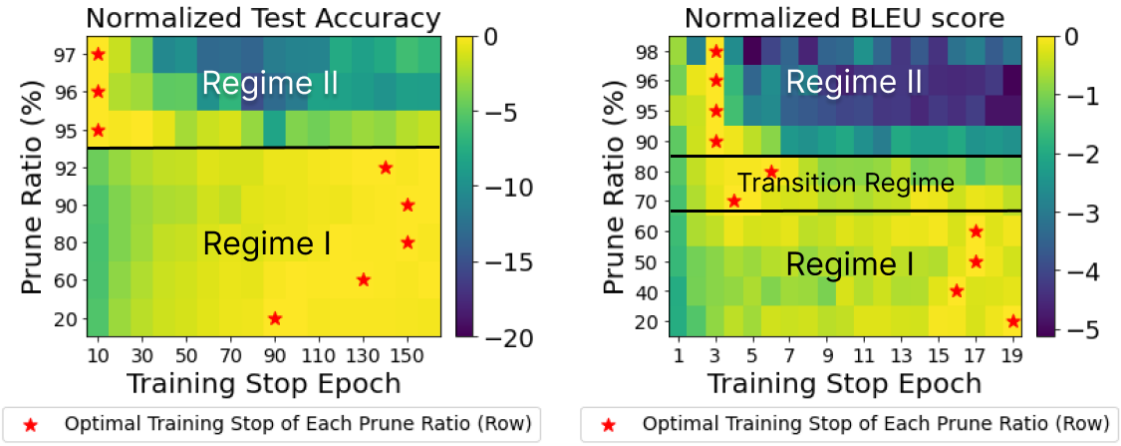
In Figure 3a, we can see that an abrupt color change partitions the 2D plots into two regimes, as we defined in Section 5.1. In Regime I, when using small prune ratios, the optimal epochs (x-axis value of red markers) are late in the training, located after 90 epochs. Note that our training also converged after 90 epochs. The training curve is shown in Figure 10 in Appendix. In Regime I, we need to train to convergence (at least 90 epochs) to get the optimal pruning results. In Regime II, when using large prune ratios, we can see the red markers are early on the left, and there are dark bands after the red markers. It means that we need to do early stopping at the red markers, and training more will decrease the test accuracy of pruned models.

Figure 3b shows a similar two-regime phenomenon in the NLP experiment. Interestingly, we find a transition region between the two regimes in the NLP regime plot. In this region, we can apply early stopping at epoch 4 or 6, but if we train more, the performance will be similar. It indicates that we can either do early stopping or train to convergence based on our needs in the transition regime.

Transition between Regimes on Training Data/Model Size We show the transition between two regimes on the training dataset size and model size. This transition demonstrates that we only need to do early stopping when training with smaller models and more data.

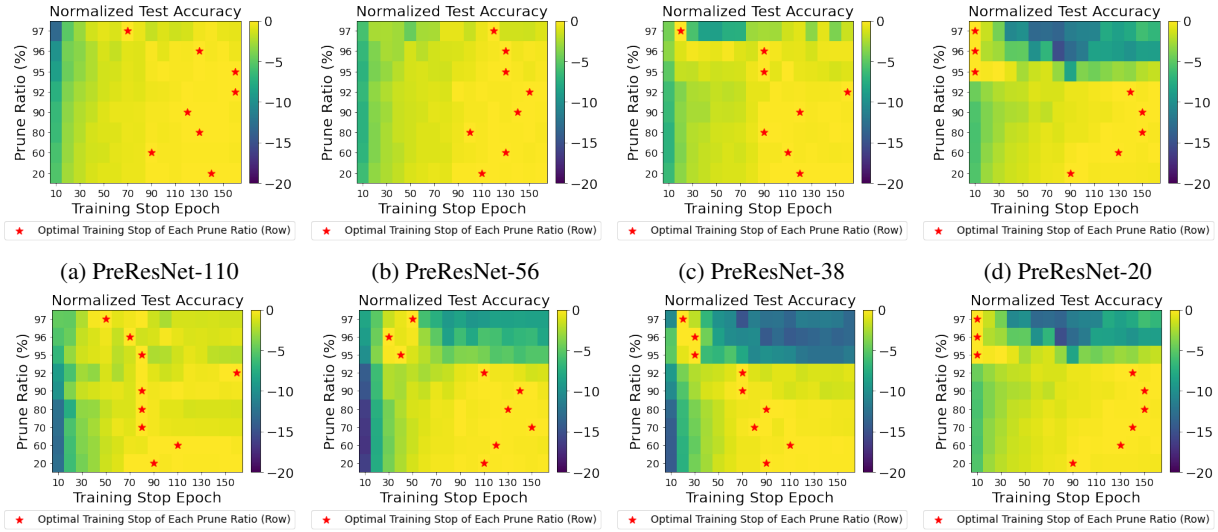
In the first experiment, we scale the model depth of PreResNet by {110, 56, 44, 20 }. We show their regime plots in the first row of Figure 4. We can see that in Figure 4a, when we prune a large model with high prune ratios, the optimal epochs are late in the training, and there is no dark band. However, for the small models in Figure 4b and 4d, the red

[2]We note that this normalization method sounds like an ad-hoc way to process data. However, the normalization is only for the visualization purposes, and we show both the normalized and the original (unnormalized) results in this report. For unnormalized results, see Section A.



(a) (b)

Figure 3: Partitioning the 2D prune ratio—training epoch diagram into two regimes of pruning. (a) Models are trained with PreResNet-20 on CIFAR-10. (b) Models are trained with Transformer-base on WMT14. BLEU score is reported on validation set.



(a) PreResNet-110 (b) PreResNet-56 (c) PreResNet-38 (d) PreResNet-20

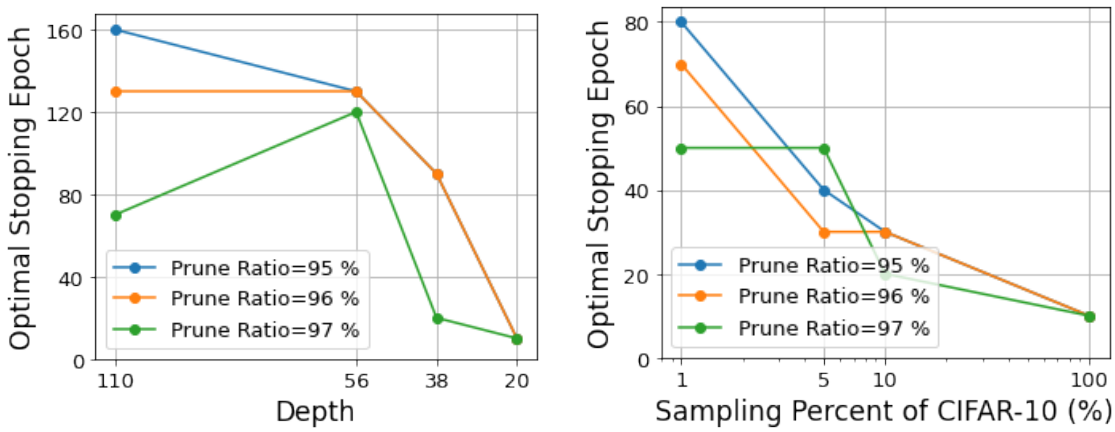
(e) 1% of CIFAR-10 (f) 5% of CIFAR-10 (g) 10% of CIFAR-10 (h) 100% of CIFAR-10

Figure 4: Partitioning the 2D prune ratio — training epoch diagram into two regimes of pruning. **(First row)**. Varying depth of model trained on CIFAR-10. **(Second row)**. Varying amount of training data to train PreResNet-20. All plots are on the same set of axes.

markers move left, and the dark band gradually appears late in the training. It indicates that when using smaller models, we should apply early stopping. Further training beyond the early stopping epoch will give poor results.

In the second experiment, we scale the dataset sizes by sampling the CIFAR-10 train set with {1, 5, 10, 100 } %. The regime plots are shown in the second row of Figure 4. From left to right, we can also see that in the upper regions, the dark band gradually appears and red markers move left.

In Figure 5, we use a line plot to visualize how the optimal epoch changes in the two scaling experiments. The curves show decreasing trend in both figures, which means decreasing model depth or increasing data amount will make the optimal stop epoch earlier. Therefore, we should stop early when the model is trained with a larger dataset size and a smaller model size.



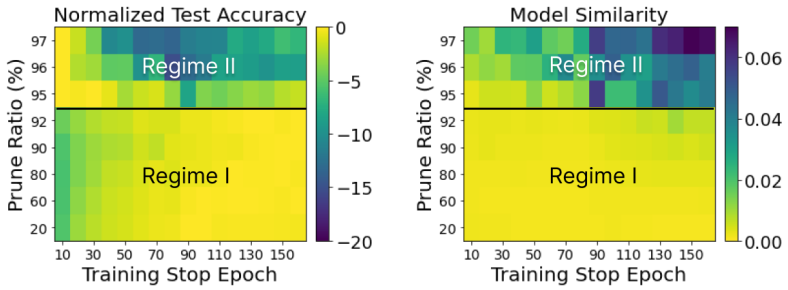
(a) (b)

Figure 5: Optimal early stopping epoch versus training settings in high prune ratios. (a) Varying model depth. (b) Varying amount of training data.

6.3 Quantification of Two Regimes

We study the following two questions to quantify the two-regime model using generalization metrics: (1) how do we determine the transition line on prune ratios between using early stopping or training to convergence? (2) how to determine the optimal early stopping epoch during training?

Transition Line on Prune Ratio We propose to measure model similarity on pruned models to predict the transition line. As shown in Figure 6, we can see that both the regime plots of normalized test accuracy and model similarity show an abrupt change between upper and lower regions, and the transition line between prune ratios are aligned. We believe that this result implies a hypothesis on neural network loss landscapes. We state this hypothesis but leave the formal investigation to future work. We believe our results imply that, when pruning well-trained models, we are likely to prune the model into two different types of loss landscapes depending on prune ratios. When using a high prune ratio, we could prune it into a poor loss landscape, in which trained models are less similar. In this case, it is more likely that one will get poor pruning results. One of the treatments is to use early stopping to retrain the original dense model and prune the model again for better results.



(a) Normalized Test Accuracy (b) Model Similarity

Figure 6: Partitioning the 2D prune ratio — training epoch diagram into two regimes of pruning and use model similarity to predict the transition line. Model are trained with PreResNet-20 on CIFAR-10. All plots are on the same set of axes.

Optimal Early Stopping We propose to measure `random_distance` during training and use it to indicate when to apply early stopping. The results are shown in Figure 7. The first row considers CV experiments, and the second row considers NLP experiments. These sub-figures show models pruned in different prune ratios, but these are all heavy pruning identified in 2D regime plots. The blue line is the test accuracy of the pruned model, which is the quantity we want to predict, and it is unknown during training because acquiring it needs two more steps: pruning and fine-tuning. The red line is `random_distance` that can be measured during training. We can see that the peaks of red lines and peaks of blue lines are well-aligned across the x-axis for all cases we show, which shows that it is possible to use the

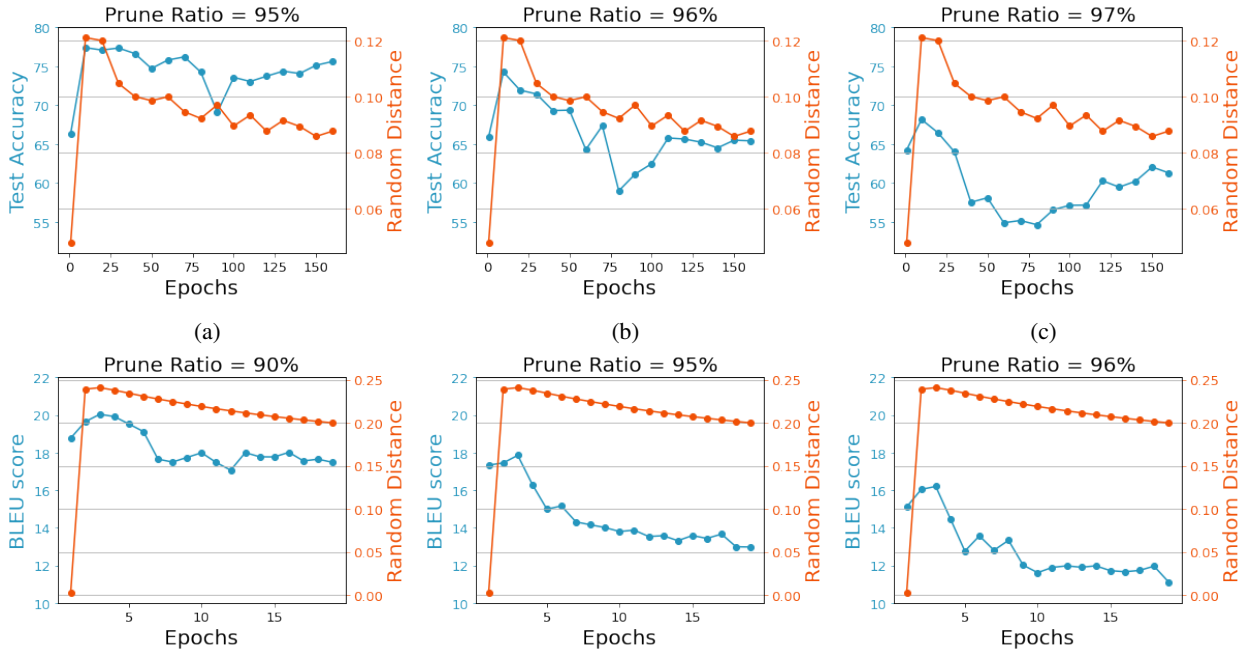


Figure 7: Predicting optimal early stopping epochs using `random_distance` measured during training. **(First row).** PreResNet-20 trained on CIFAR-10 pruned with high prune ratios. **(Second row).** Transformer trained on WMT14 pruned with high prune ratios. In all cases, the peaks of `random_distance` are aligned with the peaks of accuracy/BLEU score in x-axis.

`random_distance` to monitor the training. We can apply early stopping and start pruning when `random_distance` reaches its maximum.

7 Conclusion

In this work, we propose a two-regime model that can determine if we need to do early stopping or train to convergence when training a model to get the optimal pruning results. First, we provide results to show two regimes in CV and NLP experiments and identify the conditions for the two-regime model. Early stopping is necessary for optimal performance when the model is heavily pruned and trained with more data and smaller model sizes. Second, we show that HT-SR and loss landscape metrics can be used to quantify the two regimes. Specifically, we show results of using model similarity to determine the transition line on prune ratio and using `random_distance` to determine the optimal early stopping epoch during training. Future work includes finding better metrics that accurately determine the transition line on the prune ratio and investigating if the two-regime model exists in other similar deep learning problems like network quantization and transfer learning.

Acknowledgements

We want to thank Joe Zou, Alan Pham, Arin Chang and Zhuang Liu for helpful discussions and valuable feedback.

References

[1] Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., and Guttag, J. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

[2] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, June 2014.

[3] Chen, S., Yao, T., and Jiang, Y.-G. Deep learning for video captioning: A review. In *IJCAI*, volume 1, pp. 2, 2019.

[4] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[5] Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

[6] Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

[7] Fujiyoshi, H., Hirakawa, T., and Yamashita, T. Deep learning-based image recognition for autonomous driving. *IATSS research*, 43(4):244–252, 2019.

[8] García-Martín, E., Rodrigues, C. F., Riley, G., and Grahn, H. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88, 2019.

[9] Gavat, I. and Militaru, D. Deep learning in acoustic modeling for automatic speech recognition and understanding-an overview. In *2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pp. 1–8. IEEE, 2015.

[10] Han, S., Pool, J., Tran, J., and Dally, W. J. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

[11] Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

[12] Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

[13] Koskinen, S., Yang, D., and Kämäräinen, J.-K. Reverse imaging pipeline for raw rgb image augmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2896–2900. IEEE, 2019.

[14] Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[16] Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *ICLR*, 2017.

[17] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Conference on Neural Information Processing Systems*, pp. 6389–6399, 2018.

[18] Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D., and Gonzalez, J. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, pp. 5958–5968, 2020.

[19] Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *ICLR*, 2019.

[20] Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[21] Martin, C. H., Peng, T. S., and Mahoney, M. W. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):1–13, 2021.

[22] O'Shea, K. and Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[23] Ryu, S., Ko, K., and Hong, J. W.-K. Performance analysis of applying deep learning for virtual background of webrtc-based video conferencing system. In *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 53–56. IEEE, 2021.

[24] Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *SSW*, 125:2, 2016.

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[26] Wang, T., Wang, K., Cai, H., Lin, J., Liu, Z., Wang, H., Lin, Y., and Han, S. Apq: Joint search for network architecture, pruning and quantization policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2078–2087, 2020.

[27] Wu*, Z., Liu*, Z., Lin, J., Lin, Y., and Han, S. Lite transformer with long-short range attention. In *International Conference on Learning Representations (ICLR)*, 2020.

[28] Yang, Y., Hodgkinson, L., Theisen, R., Zou, J., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. Taxonomizing local versus global structure in neural network loss landscapes. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[29] Yang, Y., Theisen, R., Hodgkinson, L., Gonzalez, J. E., Ramchandran, K., Martin, C. H., and Mahoney, M. W. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data. *arXiv preprint arXiv:2202.02842*, 2022.

[30] You, H., Li, C., Xu, P., Fu, Y., Wang, Y., Chen, X., Lin, Y., Wang, Z., and Baraniuk, R. G. Drawing early-bird tickets: Toward more efficient training of deep networks. In *International Conference on Learning Representations*, 2020.

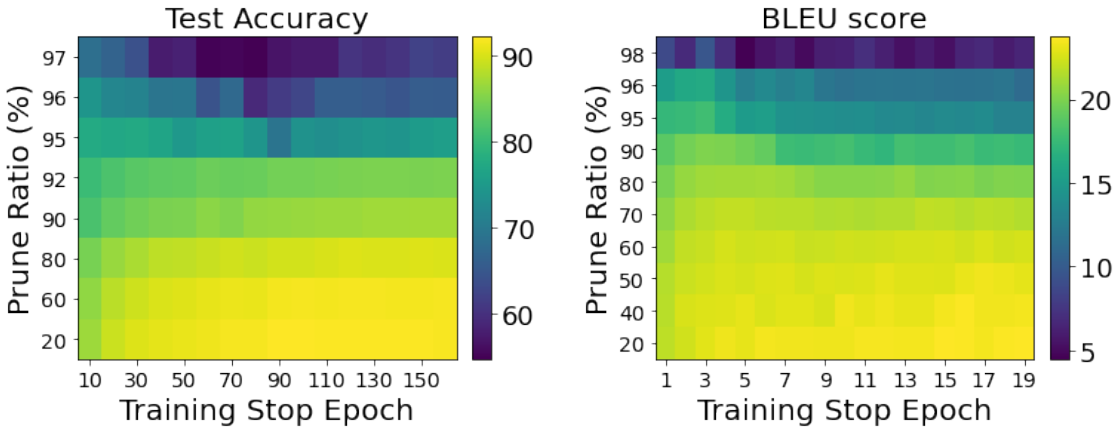
[31] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[32] Zhu, M. and Gupta, S. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=Sy1iIDkPM`.

Appendices

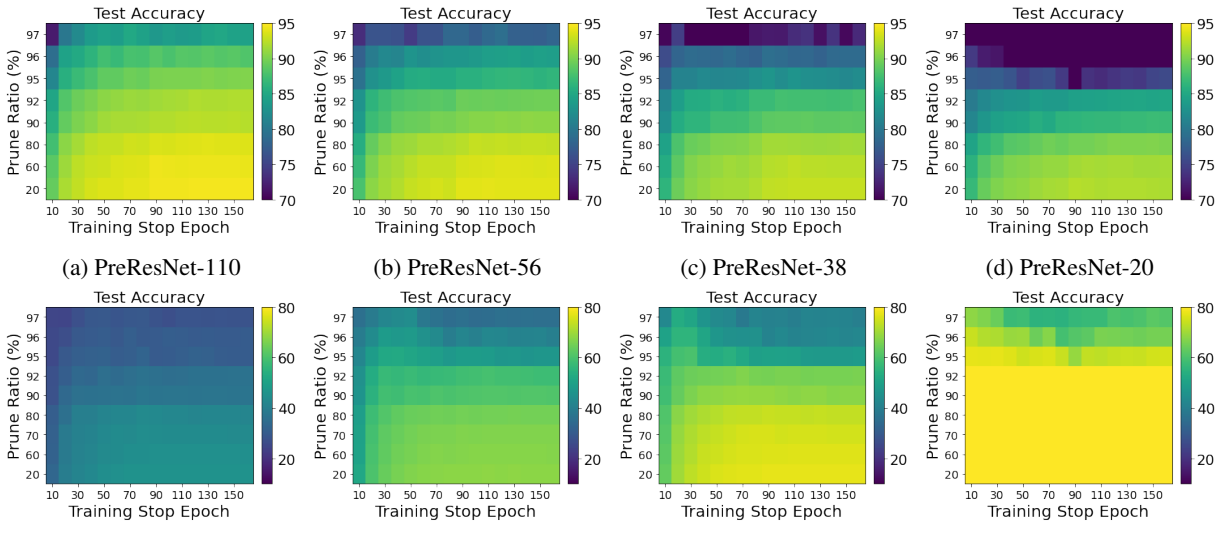
A Unnormalized 2D Regime Plots

In this section, we provide unnormalized 2D regime plots as the complementary results for the normalized plots shown in the main paper. We show the results of CV and NLP standard setting in Figure 8. We show the results of scaling training setting experiments in Figure 9.



(a) (b)

Figure 8: Original (unnormalized) test accuracy / BLEU score of pruned models trained with varying epochs and pruned with varying prune ratios. (a) Models are trained with PreResNet-20 on CIFAR-10. (b) Models are trained with Transformer-base on WMT14. BLEU score is reported on validation set.



(a) PreResNet-110 (b) PreResNet-56 (c) PreResNet-38 (d) PreResNet-20

(e) 1% of CIFAR-10 (f) 5% of CIFAR-10 (g) 10% of CIFAR-10 (h) 100% of CIFAR-10

Figure 9: Original (unnormalized) test accuracy of pruned model in the experiments of scaling training settings. **(First row)**. Varying depth of model trained on CIFAR-10. **(Second row)**. Varying amount of training data to train PreResNet-20. All plots are on the same set of axes.

B Training Curve

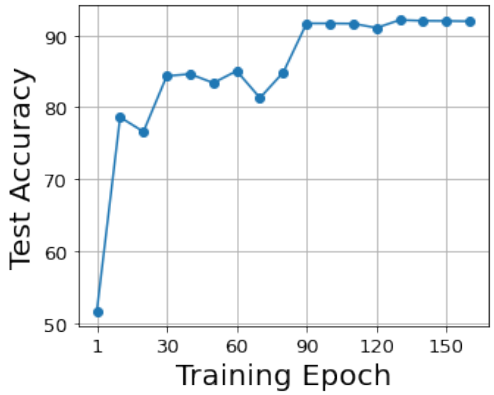


Figure 10: Training curve of PreResNet-20 on CIFAR-10. The training converges after 90 epochs.