Exploring the Impact of Network Pruning Techniques on Model Efficiency and Layer Similarity in Medical Image Analysis

Tauqeer Akhtar
2021csm1021@iitrpr.ac.in
Dr. Deepti R. Bathula
bathula@iitrpr.ac.in

Indian Institute of Technology
Ropar, Punjab, India

Abstract

Network pruning is a technique used to reduce the size of deep neural networks by removing unimportant or redundant connections and parameters. This technique make large models more computationally efficient without sacrificing accuracy. Inspired by this technique, we explore the performance of different network pruning techniques that are one-shot, layer-wise, and iterative pruning, on state-of-the-art models VGG16 and DenseNet169. In our work, we present an approach in which we observe that iterative pruning outperforms one-shot and layer-wise pruning both in terms of qualitative and quantitative performance. We also visualized the features learned by the layers of the model while implementing these pruning techniques and observed that the features learned by one-shot and iterative pruning are similar but different compared to layer-wise pruning. Extensive experiments were conducted to evaluate the performance of our proposed approach, using two datasets: CIFAR100 and HAM10000.

Keywords: Pruning , Centered Kernel Alignment(CKA) , Fine Tuning.

1 Introduction

Model compression is an important technique for reducing the size and computational complexity of deep learning models without compromising their accuracy. Medical image analysis is an area where deep learning has shown great promise, but the use of deep models in medical imaging tasks often involves processing large amounts of high-resolution image data, which can be computationally expensive and time-consuming.

There are several reasons why model compression is particularly relevant for medical image analysis:

- Resource Constraints: Medical imaging often involves limited computational resources, so compressed models enable efficient on-device analysis without requiring a high-end infrastructure.
- Latency: Medical image analysis is often time-critical, so model compression can reduce inference time and enable fast and real-time analysis.

- Privacy: Medical data is sensitive and requires careful handling, so compressed models can be designed to process data locally and reduce the risk of data breaches.

- Data Scarcity: Deep learning models often require large amounts of data, but labeled medical imaging data is often scarce, so compressed models with fewer parameters can achieve good performance with limited data.

In this study, we explored and compared the impact of three different pruning methods: iterative, layerwise, and one-shot pruning, on two different models: VGG16 and DenseNet169, and two datasets: CIFAR100 and HAM10000. Additionally, we evaluated the layer-wise feature similarity between the different pruning methods using the Centered Kernel Alignment (CKA) metric.

Our results revealed the following:

- Iterative pruning outperformed one-shot and layer-wise pruning in terms of model efficiency for both VGG16 and DenseNet169 architectures on CIFAR100 and HAM10000 datasets.

- Layer-wise pruning resulted in the least diverse feature representation among different layers compared to one-shot and iterative pruning.

- Pruned models showed better training performance in terms of accuracy and convergence speed compared to the unpruned models, especially for large models like VGG16.

- The student model distilled from the pruned teacher model learned similar features as the teacher in the first few layers.

- As more layers were pruned, we observed a significant decrease in the model's performance.

Overall, our study highlights the importance of careful selection of pruning techniques to balance model accuracy and provides insights into the differences in feature learning between different pruning methods

2 Literature Survey

Many approaches have been proposed for compressing models in order to reduce the number of parameters without compromising performance. One approach to model compression is the combination of weight pruning and knowledge distillation, as proposed in a paper[1].The method involves first pruning unimportant weights from the network and then distilling the knowledge from the pruned network to a smaller student network. Experimental results have shown that this method can achieve significant compression while maintaining high accuracy.

Another approach[6] argues that pruning should be performed before knowledge distillation, as it can improve the quality of the distilled model. The authors propose a method for pruning the teacher network based on the sensitivity of each weight to the final output and show that the resulting distilled models can achieve higher accuracy than those distilled from unpruned networks.

Additionally, a comprehensive approach[4] to model compression that combines weight pruning, weight quantization, and knowledge distillation. The proposed method first prunes the network to remove unimportant weights, then quantizes the remaining weights, and finally distills the knowledge from the pruned and quantized network to a smaller student network. Experimental results show that the proposed method can achieve significant compression while maintaining high accuracy.

For pruning before training there is a concept of "lottery tickets,"[7] is proposed, which are subnetworks of a larger network that can be trained in isolation to achieve similar accuracy with much fewer parameters. The authors use iterative pruning to find these subnetworks, and show that they can be used as the initializations for much smaller networks that can achieve similar accuracy to the original network. Experimental results show that the proposed method can achieve significant compression without sacrificing accuracy.

For layer similarity visualization purpose, CKA[5], which is a metric for comparing the similarity of neural network representations. CKA is calculated by mapping the representations of two networks to a high-dimensional feature space using a non-linear function. Then, the kernel matrices of the mapped representations are centered, normalized, and compared using the Hilbert-Schmidt Independence Criterion (HSIC) to measure their alignment. The resulting CKA score ranges from 0 to 1, where higher scores indicate greater similarity between the representations. Compared to other methods for measuring representation similarity, CKA has been shown to be more consistent and robust, and it is now widely used in deep learning research.

3 Problem Statement

In this study, we aim to investigate the effectiveness of three different pruning methods, namely iterative, layerwise, and one-shot pruning, on the efficiency of neural networks.

- Iterative pruning: Pruning is done in steps, where after each step, the network is fine-tuned to recover its lost performance. This process is repeated until the desired sparsity level is achieved.

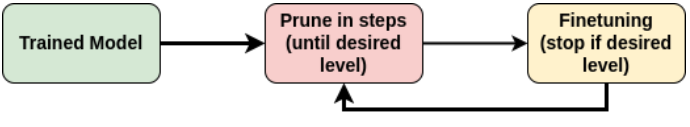

Figure 1: Iterative Pruning

- One-Shot pruning: The entire network is pruned at once to the desired sparsity level then fine-tuned only once to recover its lost performance.

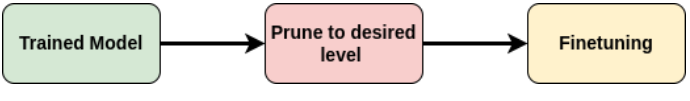

Figure 2: One-Shot Pruning

- Layerwise pruning: Each layer is pruned to the desired level of sparsity, and the model is finetuned after each layer pruning. The overall desired global pruning level is achieved when all layers have been pruned.

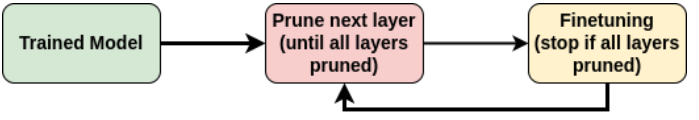

Figure 3: Layerwise Pruning

In our study, we adopted the L1 unstructured pruning technique for network pruning.

3.1 Why L1 unstructures pruning?

L1 unstructured pruning is a type of weight pruning that is commonly used in deep learning because of its ability to reduce the size of a neural network while maintaining or even improving its accuracy. L1 unstructured pruning works by removing the weights with the smallest absolute values in the network.

L1 unstructured pruning is a widely adopted method for reducing the size of neural networks due to its computational efficiency and ease of implementation. By producing sparser networks, it can reduce memory requirements and improve inference speed. Furthermore, L1 unstructured pruning can be combined with other optimization techniques to further enhance network performance. This technique can be utilized for both pre-training and fine-tuning stages and allows for fine-grained control over the amount of pruning. Additionally, L1 unstructured pruning is suitable for large-scale neural network pruning, as it can effectively handle millions of parameters while maintaining or even improving network accuracy.

For this study, we utilized VGG16 and DenseNet169 architectures for our experimental evaluations.

3.2 Why VGG16 and DenseNet169?

The choice of VGG16 and DenseNet169 for our experiment was based on their architectural properties. VGG16 is a deep convolutional neural network with 138 million parameters, making it a suitable candidate for comparing the performance of different pruning methods, especially weight pruning, which can be easily applied to this architecture. In contrast to the ResNet architecture, which is prone to model collapse with pruning, VGG16 is known to be robust to L1 unstructured pruning.

DenseNet169, on the other hand, is a smaller network with only around 13 million parameters, providing a suitable comparison to VGG16 in terms of performance after pruning. Additionally, DenseNet169 has been shown to be less susceptible to pruning-induced collapse compared to other architectures. Therefore, using DenseNet169 as a second model in our experiment allowed us to evaluate the effectiveness of pruning methods across different network sizes.

3.3 Models and Datasets

The following is a list of the combinations of models and datasets that we utilized in our study:

Model	Dataset
VGG16	Cifar100
VGG16	HAM10000
DenseNet169	Cifar100
DenseNet169	HAM10000

Figure 4: Model and Datset for Pruning

Teacher	Student	Dataset
VGG16	VGG11	CIFAR100

Figure 5: Model and Datset for Knowledge Distillation

CIFAR-100 is a well-known image classification dataset that is commonly used for benchmarking computer vision algorithms. It consists of 60,000 32x32 color images in 100 classes, with 600 images per class. The classes are grouped into 20 superclasses, each containing 5 subclasses.– (LINK TO DATASET)

HAM10000 is a publicly available skin lesion classification dataset that contains 10,015 dermatoscopic images of skin lesions from over 7,000 patients. The images are labeled with one of seven diagnostic categories, including basal cell carcinoma, melanoma, and seborrheic keratosis. The dataset is challenging because it includes a wide variety of skin lesion types and appearances, and many of the categories are visually similar. HAM10000 is often used as a benchmark dataset for developing and evaluating machine learning models for skin lesion classification and diagnosis. – (LINK TO DATASET)

3.4 Experimental Setup

We first pruned the pre-trained models using L1 unstructured pruning technique, and then fine-tuned them on the respective datasets. The models were trained for 200 epochs with a learning rate of 0.01 and a momentum of 0.9, using the SGD optimizer. We observed that the loss and accuracy stabilized after around 200 epochs.

To further evaluate the effectiveness of iterative pruning, we pruned the models by 10% of their weights after every 20 epochs. We compared the pruned models with the original models at a pruning level of 80%.

4 Results and Discussions

Our study compared three different types of pruning methods - iterative pruning, layerwise pruning, and one-shot pruning on pruning level 80% and 50%.

Model: VGG16 Dataset : CIFAR100		
Iterative	**One-shot**	**LayerWise**
Pruning Level : 80%		
65.79	62.28	55.35
Pruning Level : 50%		
67.97	63.40	56.27

Figure 6: VGG16 and CIFAR100

Model : VGG16 Dataset : HAM10000		
Iterative	**One-shot**	**LayerWise**
Pruning Level : 80%		
88.59	88.09	83.02
Pruning Level : 50%		
88.26	88.16	87.56

Figure 7: VGG16 and HAM10000

Model : DenseNet169 Dataset : CIFAR100		
Iterative	**One-shot**	**LayerWise**
Pruning Level : 80%		
64.24	63.58	59.76
Pruning Level : 50%		
63.28	59.88	60.26

Figure 8: DenseNet169 and CIFAR100

Model : DenseNet169 Dataset : HAM10000		
Iterative	**One-shot**	**LayerWise**
Pruning Level : 80%		
86.60	**88.23**	87.99
Pruning Level : 50%		
88.96	86.10	88.19

Figure 9: DenseNet169 and HAM10000

Our findings indicate that iterative pruning generally outperformed the other two pruning methods, consistently achieving better results across various sparsity levels. On the other hand, layerwise pruning, which constrains the percentage of non-zero parameters in each layer, performed the worst among the three methods as shown in 6,7,8 and 9

We observed a significant decrease in the performance of the pruned models as we increased the number of layers being pruned. This implies that pruning a large number of lay-

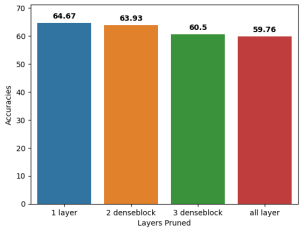

Figure 10: DenseNet169 and CIFAR100

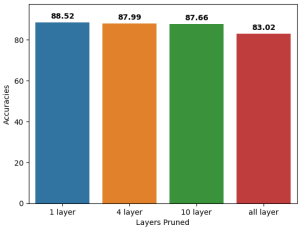

Figure 11: VGG16 and HAM10000

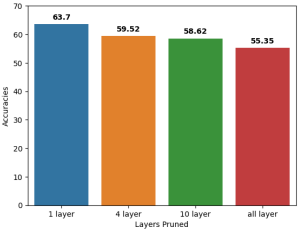

Figure 12: VGG16 and CIFAR100

ers can lead to a significant loss of important features, resulting in a decrease in the model's performance as shown in bar graph 10, 11 and 12.

Additionally, we performed a comparative analysis of the features learned by different layers of the neural networks after applying all three types of pruning methods, using the CKA metric.

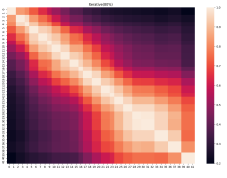

Figure 13: Iterative(CKA)

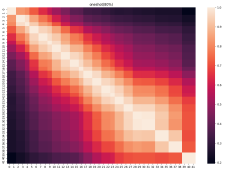

Figure 14: One-Shot(CKA)

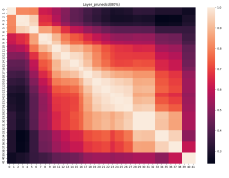

Figure 15: Layerwise(CKA)

Our findings reveal that layerwise pruning resulted in maximum feature similarity between the layers, suggesting that this type of pruning may result in the model learning fewer distinct features compared to iterative and one-shot pruning as shwonn in CKA plots 13, 14 and 15.

We conducted a comparative study of the features learned by different layers of the models pruned by iterative and one-shot pruning, iterative and layerwise pruning, and one-shot and layerwise pruning.

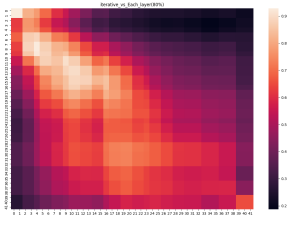

Figure 16: Iterative vs Layerwise(CKA)

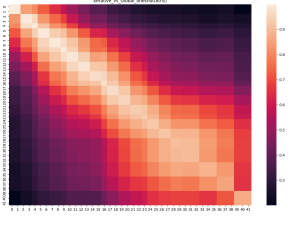

Figure 17: Iterative vs One-Shot(CKA)

Our results showed that the layers of the models pruned by one-shot and iterative pruning

learned similar types of features, but differed significantly from those learned by the layers of the models pruned by layerwise pruning as shown in 16 and 17.

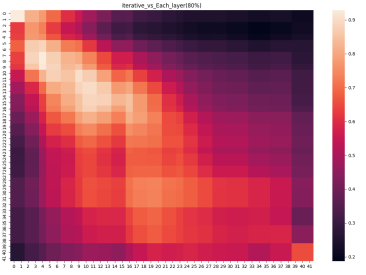

Figure 18: Student vs Teacher(CKA)

In addition to comparing different types of pruning, we also investigated the similarity of features learned by a pruned teacher and student distilled[3] using that teacher. Our analysis revealed that the initial layers of the student model learn similar features as those learned by the teacher model as shown in 18

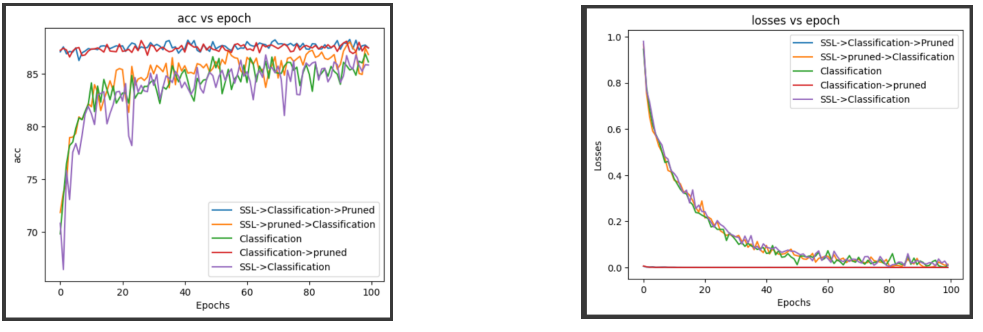

Figure 19: VGG16 and HAM10000

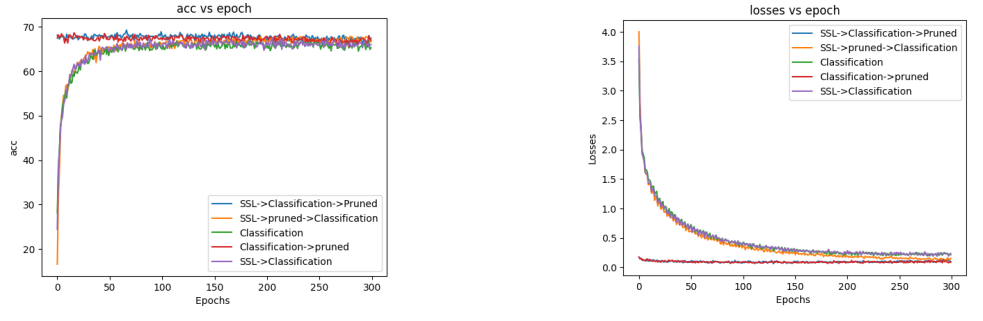

Figure 20: VGG16 and CIFAR100

Our experiments also revealed that the pruned VGG16 model performs better during training than the unpruned model, regardless of the dataset used (HAM10000 and CIFAR100)

as shown in 19 and 20. This suggests that pruning can be an effective technique for improving the efficiency of large models without compromising performance.

5 Conclusion

In conclusion, we conducted a comparative study on different pruning methods, and evaluated their effects on both performance and feature similarity. Our experiments demonstrated that iterative pruning is a promising technique for model compression, outperforming one-shot and layer-wise pruning in terms of efficiency. Moreover, we found that layer-wise pruning resulted in the least diverse feature representation among different layers. We also observed that Pruned models showed better training performance compared to unpruned models if it has a large number of parameters. However, it is important to note that as more layers were pruned, we observed a significant decrease in the model's performance, highlighting the need for careful selection of pruning techniques and compression ratios.

References

[1] Nima Aghli and Eraldo Ribeiro. Combining weight pruning and knowledge distillation for cnn compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3191–3198, 2021.

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

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[4] Jangho Kim, Simyung Chang, and Nojun Kwak. Pqk: Model compression via pruning, quantization, and knowledge distillation. *arXiv preprint arXiv:2106.14681*, 2021.

[5] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[6] Jinhyuk Park and Albert No. Prune your model before distill it. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 120–136. Springer, 2022.