Project Report: Activation-Aware Weight Pruning for Efficient 3D Reconstruction

Apoorva Molukuvan Krishnamurthy
apoorvamk@gatech.edu

Abstract

Neural Radiance Fields or NeRFs[8] are an excellent technique for obtaining highly realistic 3D scene renderings however they are computationally expensive due to several MLP inference calls for creating a single pixel and time-consuming volumetric rendering. Further, NeRFs utilize a dense, fully-connected neural network as a function to approximate the neural radiance field. The MLP further requires mapping the simple 5D coordinates into a more complex higher dimensional space to represent higher frequency functions. The ray-marching process in the training of NeRFs also needs several points to be sampled on the ray and evaluated on the network which gets expensive with high-resolution images. The entire NeRF pipeline has several bottlenecks that require optimization. This research project explores the idea of post-training Activation-aware weight pruning. The activations from a calibration dataset are used to find which weight channels are most important and the remaining are pruned. After pruning the weights, the pruned network is evaluated on the MSE Loss, PSNR, and visual fidelity of the scenes reconstructed.

1 Introduction

Neural Radiance Fields (NeRF) is a technique for recreating realistic 3D scenes from a set of 2D images. The algorithm uses a fully connected dense deep neural network or MLP which takes a 5-dimensional input (the spatial location (x, y, z) and the viewing direction $(\theta, \phi)$and outputs the and volume density $\sigma$and emitted color $c = (r, g, b)$. NeRFs can be used to render high-fidelity visual scenes with several objects and outperform previous techniques in 3D reconstruction. While NeRFs have gained massive popularity in industry and performance due to their ability to generate realistic scenes, they are computationally very expensive as each scene needs multiple network calls to generate each pixel. Efficiently re-constructing 3D scenes is important as it is widely used in applications such as AR-VR devices, medical imaging, and product design which usually don't have access to a large amount of compute.

2 Motivation and Background

There have been several previous works that address inefficiencies in the NeRF pipeline. For example, in Neural Sparse Voxel Fields [6], the idea is to reduce the number of sampled points by combining NeRFs with sparse voxel grids to exploit the sparsity in 3D space. In KiloNeRF [10], the idea is to replace the single large MLP with thousands of tiny MLPs after sub-dividing the 3D scene into a grid. Each grid's MLP is then trained separately so that they can be run in parallel at inference. Another recent work, Neural Radiance Distribution Field (NeRDF)[13] and R2L: Distilling Neural Radiance Field [12] applies Knowledge Distillation to achieve 254x speedup but struggles to generate 360-degree views. While many of the existing works are promising there are few works that take into consideration the MLP activations and utilize that information in making the network more efficient. Traditional pruning and quantization techniques, disregard the activations which ultimately define

the output of the network. There have been recent works [11], [5] that use this approach with LLMs pruning and quantization, and in this project, I apply it to NeRFs and assess the performance. Another work [14] utilizes activations to structurally prune and find the winning ticket iteratively.

Pruning is an effective technique [1], [4], [3] for compressing deep neural networks with several parameters and achieving inference speedups [9], [2] The motivation behind applying activation information for pruning the networks is that ultimately the outputs from the activation are what determine the final scenes, so preserving the most important weights related to the activations will preserve visual fidelity. Further, NeRFs are trained on one scene so there will likely be some pattern in the activation pattern which can be exploited when we prune a network.

3 Data

The dataset used to train the network and then perform post-training pruning is the Local Light Field Dataset [7], specifically the Fern scene and images. The training dataset consists of 20 images shot from different viewpoints.

Figure 1: Dataset images captured at different angles

4 Methodology

Instead of using traditional magnitude-based pruning, I explore post-training activation-aware weight pruning. Given a weight matrix W and input activations X, I compute the activations given by $Y = W \cdot X$ and the use the activation values to retain the most important weights. The method is motivated by the idea that NeRFs are typically trained on a set of images pertaining to only one scene and thus the images follow a similar input distribution and activate the same neurons. By retaining the neurons that produce the highest activations, we can ensure lesser deviation in the output.

Activation-aware Weight Pruning Method:

1. Complete training of the NeRF model on a particular scene. I trained the original NeRF model for 200,000 epochs.
2. Create a calibration dataset that will be used to accumulate the activations. I chose a random sample of the training images to assess the activation distributions.
3. In each forward pass, I store the activations for each DNN layer and average them across the samples.
4. After accumulating the activations, I prune the weight channels that correspond to the lowest activations. The number of channels pruned is in proportion to the percentage of the model that we wish to prune.
5. The pruned model is then evaluated on its overall model size, MSE loss, PSNR and visual fidelity of the images generated.

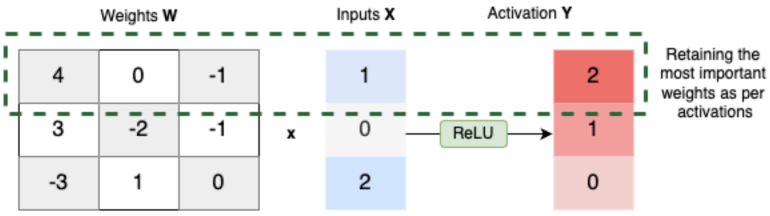

Figure 2: Activation-aware pruning method

5 Experiments and Results

I varied the percentage of the MLP pruned and evaluated the performance with different metrics.

Pruned %	Model Size	MSE Loss	PSNR
Original Model	2.273MB	0.0859	14.4434
15%	1.995MB	0.0695	14.9187
30%	1.717MB	0.0714	14.9187
50%	1.336MB	0.0847	14.5010
65%	1.058MB	0.0883	14.6097
80%	0.780MB	0.0998	14.5723
95%	0.495MB	0.1034	15.038

Table 1: Evaluation losses and other metrics after 20 epochs

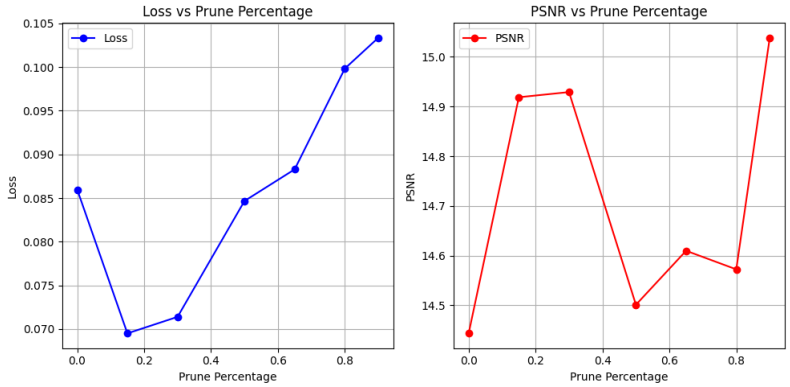

Figure 3: Effect on MSE loss and PSNR vs Prune percentage

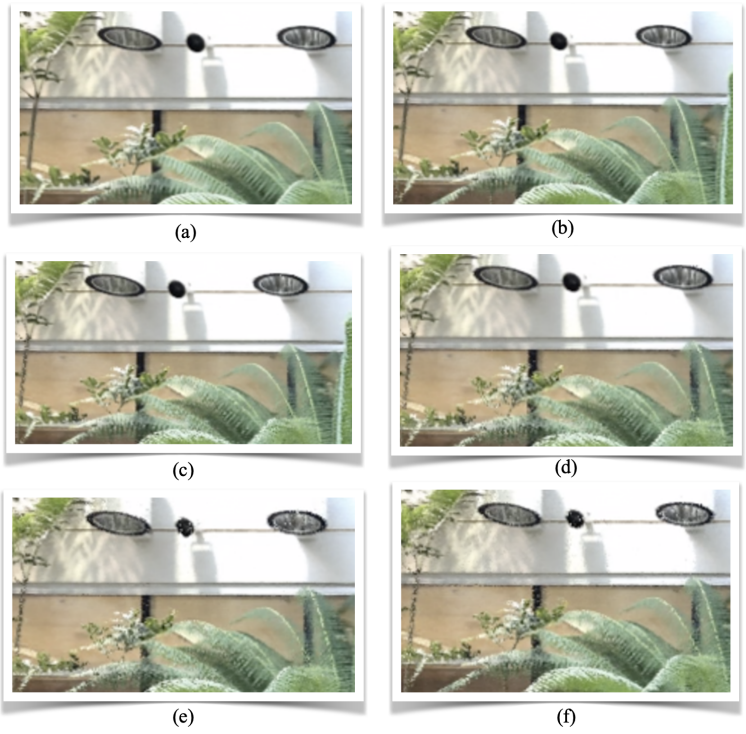

Figure 4: Comparison of visual fidelity of the different scenes for different pruned percentages. (a) original model (b) 15% pruned (c) 30% pruned (d) 50% pruned (e) 80% pruned (f) 95% pruned

6 Conclusion

1. From the results we see that up to 50% pruning good visual fidelity is maintained and after that the image becomes noisy. Some denoising techniques and post-pruning training can boost the quality of the rendering.
2. In the images generated, areas without details and with less variance continue to have the same visual fidelity as opposed to areas with more details, textures, and lighting effects. Future work can focus on pruning the image such that activations in such regions is preserved to a greater extent.
3. The MSE loss increases gradually with more percentage being pruned however there is no noticeable trend in the PSNR values. This shows that MSE loss is a better metric to quantify performance over PSNR. The high PSNR value might be due to the weight pruning which causes very high and low output values leading to a higher PSNR value.
4. Overall, the idea of Activation-aware weight pruning shows decent performance with NeRFs and can be adopted with further fine-tuning.

7 Link to Code

The project codebase can be found at https://github.com/apoorva-mk/nerf-pytorch

References

[1] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016.

[2] Yihui He. Pruning very deep neural network channels for efficient inference, 2022.

[3] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets, 2017.

[4] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks, 2020.

[5] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration, 2024.

[6] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields, 2021.

[7] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines, 2019.

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[9] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference, 2017.

[10] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps, 2021.

[11] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2023.

[12] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis, 2022.

[13] Yushuang Wu, Xiao Li, Jinglu Wang, Xiaoguang Han, Shuguang Cui, and Yan Lu. Efficient view synthesis with neural radiance distribution field, 2023.

[14] Kaiqi Zhao, Animesh Jain, and Ming Zhao. Adaptive activation-based structured pruning, 2023.