Analyzing the Robustness of Model Compression Techniques to Noise

CSC 591 (025) Course Project Report

Akhil Bukkapuram
ECE Department
NC State University
Raleigh, NC, USA
bakhil@ncsu.edu

Karthik Ganapathi Subramanian
ECE Department
NC State University
Raleigh, NC, USA
kganapa@ncsu.edu

ABSTRACT

Model compression techniques such as knowledge distillation and pruning are very popular methods used to decrease the memory size and computational resources required of large neural networks. These techniques are necessary when deploying the models in resource-constrained devices edge devices. Although these techniques help in reducing the memory required to deploy the models there are concerns related to their performance on the noisy input data, which is very common in real world scenarios.

In this project, we intend to analyze the robustness of various deep compression techniques to noisy / out of distribution data and find which models generalize well to unseen data. We will mainly compare three techniques: Knowledge distillation, unstructured pruning, and structured pruning. A ResNet-18 will be considered as the base model and we will experiment on different noise augmentation on the CIFAR-10 dataset. The model are compressed to the same number of parameters using the various techniques and their performance will be measured using noisy data. We test the compressed model in different types of noisy data such as change in brightness, contrast, defocus, blur, Gaussian noise. This way we can identify which compression techniques are resilient to certain types of noise. In the end our aim is to provide an analytical comparison of the robustness of the compression techniques to noisy data and their generalization capability.

KEYWORDS

Robustness, Model Compression, Knowledge Distillation, Pruning

1. BACKGROUND

1.1 Knowledge Distillation

Large neural network models can capture fine-grained features through the use of many layers and a large number of neurons. This allows the network to learn more complex and abstract representations of the input data. Moreover, large neural network models can use transfer learning to leverage pre-trained models that have already learned fine-grained features on a large dataset. This can help to improve the performance of the model on a smaller dataset or a more specific task, as the model has already learned to capture a broad range of features in the input data. However, deploying large neural network models on edge devices can result in several drawbacks, such as requiring significant computational resources, increasing energy consumption, occupying limited storage capacity, slowing down inference speed, being difficult to update, posing security risks, and having limited adaptability.

Knowledge distillation is a technique for compressing large neural network models into smaller, more efficient models. It is a method that involves training a smaller neural network model to imitate the behavior of a larger and more complex model. By minimizing the difference between the outputs of the two models, the student model can predict the same outputs as the teacher model but in a more efficient way. This technique helps to compress large models and reduce their computational and memory requirements, making them more suitable for use in applications with limited computing resources. In a neural network, the term "knowledge" usually refers to the learned weights and biases. However, in a large and complex neural network, there are many different sources of knowledge that can be used in knowledge distillation. For example, some methods use the logits as the source of teacher knowledge, while others focus on the weights or activations of intermediate layers. There are also other types of knowledge, such as the relationships between different types of activations and neurons, or the parameters of the teacher model itself, that can be useful in the compression process.

In our project we use response-based knowledge distillation, that focuses on the final output layer of the teacher model. Soft targets, which are probability distributions over output classes that are estimated using a SoftMax function, represent the response-based knowledge in computer vision tasks such as image classification. In this process, the soft targets' contribution to the knowledge is adjusted by a temperature parameter.

The goal is for the student model to replicate the behavior of the larger teacher model. To accomplish this, a loss function called the "distillation loss" is used to measure the difference between the logits produced by the student and teacher models. By minimizing this loss during training, the student model gradually learns to make the same predictions as the teacher model. We use

offline distillation, where the teacher model is first trained on a training dataset, and then the knowledge it has gained is transferred to the student model through the distillation process.

1.2 Pruning

Large models often perform well on complex datasets, however most models are heavily over parameterized for the task at hand. We can reduce the number of parameters of the network by a significant fraction and retain the performance of the model. Pruning connections is a popular approach that involves directly removing parameters by setting their weight values to zero within the parameter tensors. Pruning the weights of a network directly without changing the architecture of the network is known as unstructured pruning. But the limitation of unstructured pruning is that most frameworks and hardware cannot accelerate the computation of sparse matrices. Therefore, even if a large number of zeros are added to the parameter tensors, it does not reduce the actual computational cost of the network.

An alternate approach is to alter the architecture of the network through pruning to reduce the computational cost , which is referred to as Structured pruning. For instance, by removing entire filters from convolutional layers, not only are the networks lighter to store, but they also require fewer computations and generate lighter intermediate representations, which require less memory during runtime. This makes it a preferred kind of pruning, especially for tasks like semantic segmentation or object detection, where intermediate representations can be more memory-consuming than the network itself.

2. RELATED WORK

In [1] Maroto et. al explore the use of knowledge distillation (KD) to improve the adversarial robustness of neural networks. The paper demonstrates that KD can improve the robustness of neural networks to adversarial attacks, even when compared to other techniques such as adversarial training and defensive distillation. The authors also investigate different variations of KD and show that they can further improve adversarial robustness. Liebenwein et. al in [2] evaluate the impact of neural network pruning on various performance metrics beyond just test accuracy, such as training time, convergence rate, robustness to adversarial attacks, transfer learning, and energy efficiency. The paper highlights that aggressive pruning can lead to slower convergence and reduced robustness to adversarial attacks, while more conservative pruning can improve transfer learning and energy efficiency. The authors suggest that the impact of pruning on neural network performance is complex and depends on various factors, and that more research is needed to optimize pruning strategies for multiple performance metrics simultaneously. In [3] the authors investigate the robustness and transferability of universal adversarial attacks on compressed neural network models. The authors demonstrate that compressed models can be more vulnerable to universal attacks than their uncompressed counterparts. Moreover, they show that the transferability of these attacks is lower for compressed models, suggesting that targeted attacks may be more effective in this scenario. The paper emphasizes the importance of considering the robustness and transferability of adversarial attacks on compressed models, especially in resource-constrained environments.

3. METHODOLOGY

3.1 Noise

In our study, we conducted experiments to evaluate the robustness of deep learning models against various types of noise augmentations. Specifically, we utilized five different noise augmentations, namely:

- Gaussian Noise(mean =0, variance=(0,2))
- ISO Noise (color_shift=(0.01, 0.05), intensity=(0.1, 0.5)
- RGB Shift (r_shift_limit=20, g_shift_limit=20, b_shift_limit=20)
- Pixel Dropout (dropout_prob=0.01)
- Random Fog (fog_coef_lower=0.3, fog_coef_upper=1, alpha_coef=0.08)

on our test dataset.

We then measured the accuracy of each compressed model on the noise-augmented dataset, comparing the performance of each model to a noiseless dataset and to the baseline model's performance on each noisy dataset. Our findings can provide insights into the effectiveness of these compressed models in handling noise, which can be useful in real-world applications where data is often corrupted or noisy.

3.2 Compressed Models

In our study, we have developed five compressed models with different sizes: 10%, 40%, 60%, 75%, and 90% compared to the baseline model. To ensure knowledge distillation, we have maintained the same architecture of the student models as that of the teacher model, including the same number of residual connections across all models. The only changes made were in the number of layers and channels.

We use distillation loss, which is a weighted combination of cross entropy loss and KL divergence loss to train the student model. The KL divergence loss measures the difference between the probability distribution of the teacher and student logits.

The KL divergence loss is defined as:

$$L(y_{Teacher}, y_{Student}) = y_{student} * \log\left(\frac{y_{student}}{y_{teacher}}\right)$$

To make a fair comparison of pruning and knowledge distillation's impact on robustness we pruned the baseline models for the appropriate sparsities such that the number of trainable parameters in the pruned models and the student models were the same. We have conducted our tests with three different types of pruning paradigms, namely: Unstructured Global Pruning, whether it is global or local pruning. In case of structured pruning, we remove the channels with the lowest L2 norm across all the layers. After pruning the weights we train the models again to fine-tune the model, this ensures better performance of the pruned model.

We have ensured that none of our models were trained on the noise-corrupted dataset meaning that the data that all the models are tested on is out of distribution. This ensures a fair comparison of the robustness of each model compression technique.

3.3 Training Procedure

The baseline ResNet-18 model was trained on the noiseless training dataset for 75epochs. With this model, we achieve a test accuracy of 87.59%.

Unstructured Local Pruning and Structured Pruning (Channel Pruning).

For Unstructured pruning we have prune the neurons with the lowest L1 norm either globally or in each layer depending on

The student models were trained using knowledge distillation from the teacher model for 75 epochs. For pruning techniques, one shot pruning strategy was used where we pruned the model in a single instance to the required sparsity and finetuned the pruned model for 15 epochs on the noiseless train dataset. Once we had all the models, the test dataset was created with various noise augmentation strategies and the models were tested to compare their performance.

4. RESULTS

The performances of each model compression technique with respect to the different noise are shown in Fig 1.

A comparison of each compression techniques with respect to various noise is given in Fig 2.

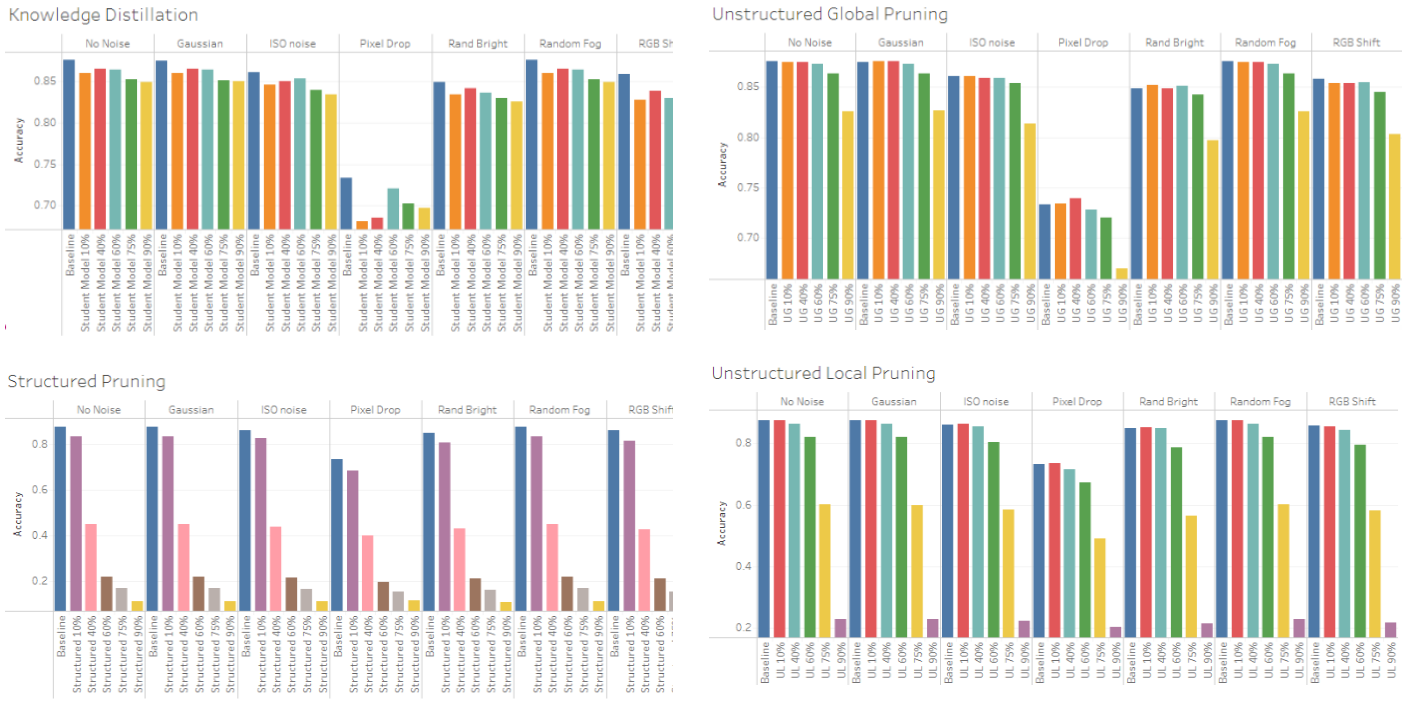

Fig 1: Accuracy of different compression techniques on various noise augmentations

We also calculated different metrics for the compressed models. We calculated the inference time and the throughput of the compressed models. The results for the various models are shown in tables 1, 2 and 3.

Knowledge Distillation	Baseline	Student Model 10%	Student Model 40%	Student Model 60%	Student Model 75%	Student Model 90%
Inference Time(ms)	3.7	3.67	3.12	2.396	2.81	2.41
Throughput(batch size 256)	12798.44	13962.30	15306	22294.98	24462.08	26654.93
Model Size(MB)	42.66	38.15	24.647	18.726	10.738	5.101
GFLOPs	0.14	0.12	0.12	0.065	0.039	0.03

Table 1: Performance metrics of student models

Structured Pruning	Baseline	Structured 10%	Structured 40%	Structured 60%	structured 75%	Structured 90%
Inference Time(ms)	3.7	4.18	4	3.91	3.95	3.88
Throughput (batch size 256)	12798.44	12774.56	13193.27	13356.24	13452.36	13619.1

Table 2: Performance metrics of models compressed using structured pruning

Unstructured Pruning	Baseline	Global 10%	Global 40%	Global 60%	Global 75%	Global 90%	Local 10%	Local 40%	Local 60%	Local 75%	Local 90%
Inference Time(ms)	3.7	3.88	3.90	4.07	4.23	4.12	3.89	3.95	3.98	4.08	4.21
Throughput (batch size 256)	12798	13452	13509	12715	12586	12766	12792	12684	12640	12592	12315

Table 3: Performance metrics of models compressed using unstructured pruning

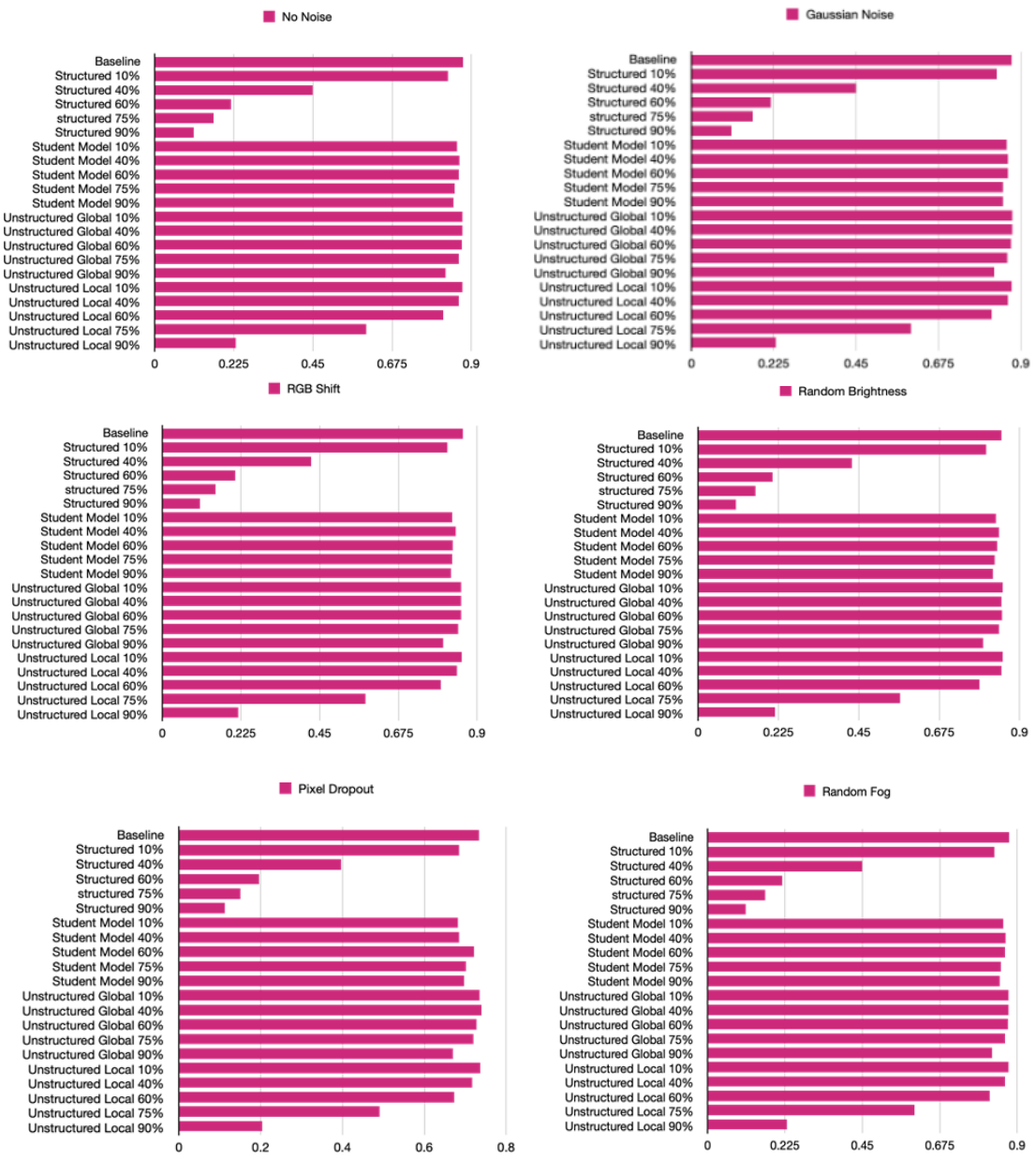

Fig 2: Comparison of each model on the various Noise Augmentations

5. CONCLUSIONS

From our experiments, we observed that student models produced via knowledge distillation had performance similar to that of the teacher model without any major drop in accuracy. Further more we could observe tangible reduction in computational efficiency, in the form of reduced inference time, FLOPs required and storage.

Among the pruning mechanisms, we observed that unstructured pruning gave better results compared to structured pruning.

Unstructured global pruning outperforms local unstructured pruning at higher pruning ratios showing that unstructured global pruned models have better generalization capabilities. However due to the fact that we need specialized software and libraries to take advantage of the sparse structure offered by pruned models, it is more practical and easier to deploy knowledge distilled student

models as they offer more compute gains and are robust in handling out of distribution and noisy data.

Hence, we can conclude that model compression does not impact generalizability with respect to noise. Most compressed models exhibit performance on par with baseline models on unseen/noisy data.

REFERENCES

[1] Maroto, J., Ortiz-Jiménez, G. and Frossard, P., 2022. *On the benefits of knowledge distillation for adversarial robustness*. arXiv preprint arXiv:2203.07159.

[2] Liebenwein, L., Baykal, C., Carter, B., Gifford, D. and Rus, D., 2021. *Lost in pruning: The effects of pruning neural networks beyond test accuracy*. Proceedings of Machine Learning and Systems, 3, pp.93-138.

[3] Matachana, A.G., Co, K.T., Muñoz-González, L., Martinez, D. and Lupu, E.C., 2020. *Robustness and transferability of universal attacks on compressed models*. arXiv preprint arXiv:2012.06024.

[4] Xie, H., Xiang, X., Liu, N. and Dong, B., 2020. *Blind Adversarial Training: Balance Accuracy and Robustness*. arXiv preprint arXiv:2004.05914.

[5] https://leimao.github.io/blog/PyTorch-Pruning/

[6] https://blog.paperspace.com/writing-resnet-from-scratch-in-pytorch/

[7] https://towardsdatascience.com/the-correct-way-to-measure-inference-time-of-deep-neural-networks-304a54e5187f

[8] https://github.com/1adrianb/pytorch-estimate-flops

[9] https://stackoverflow.com/questions/71851474/how-to-find-the-size-of-a-deep-learning-model

[10] https://albumentations.ai/docs/api_reference/augmentations/transforms/#albumentations.augmentations.transforms.RandomBrightnessContrast