Exploring Bias and Fairness in Resource-Efficient Neural Networks: A Comparative Study

Akruti Sinha, Max Groover

Abstract

This term paper explores the complex interplay between resource efficiency, bias, and fairness in neural network models. The study aims to understand the impact of resource efficiency on bias and fairness, as well as to compare the performance of various techniques for reducing bias and increasing fairness in resource-efficient neural networks. Moreover, the study aims to investigate whether a fair neural network is preferable over a resource-efficient neural network. Our hypothesis is that there exists a tradeoff between resource efficiency, accuracy, and fairness in deep neural networks. Specifically, when compressing a network to improve resource efficiency, there will be a decrease in either accuracy or fairness, with a balance between the two needing to be struck. We will be using the CIFAR-10 and UTKFace datasets to train Resnet-32 and VGG16 models. Our aim is to use pruning and knowledge distillation techniques to improve the resource-efficiency of the models while also tracking fairness metrics such as demographic parity, equal opportunity, and predictive parity. The results of this study will provide valuable insights into the challenges and opportunities for creating more fair and accurate resource-efficient neural network models.

1. Introduction

Neural networks have emerged as a heavily relied on tool in the world of machine learning and problem solving. Through their use, we have achieved impressive accuracy on a wide array of tasks such as image recognition and natural language processing. One weakness of successful deep neural networks is their computational demand, which can hurt deployability and stagnate research.

As a means to solve this problem, researchers have developed ways to prune the network and reduce size and complexity. As it is difficult to understand exactly what is occurring as we prune a network, the effects of this compression on the bias and fairness within the network are not well understood.

In this paper, we seek to observe these effects and offer some insight into easily overlooked consequences of compression on a network. We present two experiments in which we train a ResNet-34 and VGG-16 network on CIFAR-10 and FairFace respectively. We prune the networks of 50% and 80% of their weights and measure several bias and fairness metrics at each stage. We add a few intermediate sparcities as well for the FairFace experiment after noticing interesting results. The results, while not large enough to be conclusive, do offer a greater understanding of how pruning a neural network can impact its bias, as well as the relationship between a pruned model's accuracy and fairness.

2. Related Work

The authors of paper [1] present a solution to the challenge of balancing accuracy and fairness in deep neural network (DNN) models. The authors observe that this trade-off is not only present in the overall model, but also on an individual neuron level during backward propagation-based training. To address this, the authors propose "FairNeuron", an automatic repairing tool that combines path analysis and selective dropout to improve fairness while maintaining accuracy. The tool detects neurons with conflicting optimization directions between accuracy and fairness, and balances the trade-off through dropout training. The authors evaluate FairNeuron on three datasets and find that it has better performance compared to state-of-the-art methods, with the exception of Convolutional Neural Network (CNN) models, which require improvement in future work.

The paper [2] studied the effect of model compression techniques on the size, accuracy and fairness of facial expression recognition (FER) models. Three compression techniques (pruning, weight clustering, and post-training quantization) were implemented and evaluated on two datasets (Extended Cohn-Kanade Dataset (CK+DB) and the Real-World Affective Faces Database (RAF-DB)). The results showed that compression techniques can reduce the size of the models with minimal impact on accuracy. However, the findings showed that compression can amplify existing biases in terms of gender for FER models trained on the CK+DB dataset. The impact of compression on fairness varied across different compression techniques. The post-training quantization had no visible effect on fairness, while pruning and weight clustering amplified biases. The results on the

RAF-DB showed that the impact on fairness was not negative for the sensitive attributes of gender, race and age.

The paper [3] presents an overview of the current (2021) state of research on bias deep neural networks used for Natural Language Processing (NLP) models. The authors open with a discussion on the types of bias that are often present in NLP models. These types of biases consist of gender, racial, and socioeconomic bias. They gather information from various studies that investigate the biases that exist in varying NLPs in the areas of machine translation, sentiment analysis, and named entity recognition. The paper, in addition to investigating bias, also discusses different methods for mitigating this bias in NLP models. They outline some of the currently implemented ways we identify and deal with bias, but also provide a more general approach that all should keep in mind when developing a deep neural network. The conclusion of the paper was an emphasis on the importance of additional research bias and addressing it within NLP models.

When it comes to existing research on **mitigating bias** in compressed neural networks, there has been little work. The few studies that have talked about mitigating bias in compressed neural networks, are quite commendable.
The authors of [4] proposed one of the best contributions for analyzing and minimizing generated bias in pruned neural networks. To perform a quantitative evaluation of pruned neural networks, they offer two metrics viz. Combined Error Variance (CEV) and Symmetric Distance Error (SDE). It's worth noting that, in comparison to previous metrics like Pruning Identified Exemplars (PIEs) which was proposed by Google, CEV and SDE have several benefits. The most important of which is that, while they evaluate any compression method, they are superior at providing a clear measurement of model quality. They can also be compared directly because they do not require vast populations of models to be trained like PIEs normally require. SDE and CEV were created to consider both the spread of the classification error as well as how the model makes mistakes. The authors show through a series of experiments that knowledge distillation is one of the most proven ways for mitigating generated bias in neural networks that have been compressed by eliminating weights from a trained model. This holds true even for unbalanced datasets. When the authors analyze the influence of the same on model bias, they find that while unbalanced datasets typically aggravate bias issues, particularly in pruned neural networks, knowledge distillation's effectiveness in reducing bias remains unchanged. Finally, the authors propose that model similarity, which has substantial correlations with pruning generated bias, can explain why bias emerges in pruned neural networks at all. To reach this conclusion, the authors have used Singular Vector Canonical Correlation Analysis (SVCCA) [5]. The use of SVCCA to examine how the non-pruned and pruned models' layer representations are similar, produced the conclusion that the pruned models (that are extremely similar to the non-pruned models) generate less bias [4]. Also, while SDE and CEV's concept overcomes the limitations of PIEs, the model population size of [4] is nowhere near that of Google's initial paper proposing PIEs [6].

SVCCA [5] analyzes representations using a combination of "Singular Value Decomposition and Canonical Correlation Analysis (CCA)". Canonical Correlation Analysis provides comparisons without any necessary alignment and Singular Value Decomposition determines the directions to the original dimensions. According to [5], the integration of these two techniques provides insights into different neural networks. Authors have suggested some new techniques that are very less suffering from overfitting and measure the dimensions of the layers.

3. Background Knowledge

The authors of [7] have done excellent work in understanding the relationship between fairness and compression methods. They [7] show that compression can cause significant accuracy loss and bias, particularly for disadvantaged groups. Furthermore, they demonstrate that present model compression methods, such as pruning and quantization, can worsen bias and that new fairness-aware compression techniques are required.

There definitely has been more work in this as, for example, the authors of [8] investigates the possibility of pruning approaches to introduce bias in transformer models. Their findings suggest that pruning can generate bias, especially when done selectively to specific layers or tokens. They also propose a way for minimizing this bias during training by using a fairness regularization term. The authors suggest that taking bias into account while compressing models is critical for developing fair and equitable AI systems.

4. Problem Description

The problem we are attempting to address is the often overlooked topic of bias and fairness within deep neural networks and how the pursuit of resource-efficiency affects this topic. It is a relatively untapped field of research that is starting to receive the attention it needs. While we cannot feasibly come to a conclusive resolution within the scope of this observational study, we can still provide additional information and context for compression and bias. If nothing else, our study can influence those creating and compressing models to consider fairness to a greater extent. Our study may also act as a general guide for those same people to measure the bias in their models.

5. Challenges Faced

In the process of researching this topic and running experiments, we faced several challenges that we either had to overcome or simply accept. Some of the more prevalent challenges we faced are as follows:

- Minimizing extraneous bias: Like any experiment, we attempt to mitigate bias introduced by factors we are not observing. The primary way we do this is by using datasets with as little bias as we can. The dataset a model trains on can greatly influence the bias within the model.
- Bias and fairness are complex topics: There are no encompassing bias and fairness metrics for all situations, thus choosing the right ones for the problem at hand is important and difficult. Two different metrics may provide two different conclusions, so we must determine which is more reasonable.
- Interpretation of results: Even when we observe data with a clear and concise conclusion, it is still difficult to understand why we observe what we do. Typically the best we can do is generate reasonable inferences as to the causes of the patterns.
- Choosing the most generalizable experiments and datasets: Our goal was to choose datasets and architectures that would best generalize our results. With more time and freedom, we would ideally perform experiments across all major domains with architectures of greatly varying structure.
- Obtaining enough data to draw strong conclusions: With as broad a research topic as ours, it is hard to derive conclusive evidence to support any patterns we observe. Recognizing we cannot feasibly address this issue, we instead try to broaden our experimental results and offer our findings as an insight into this topic.

We will discuss how we addressed these challenges in the experiments, but we cannot fully address the interpretation of the results challenge beyond the inference described above.

6. Proposed Method

The proposed approach is divided into two sections. The ResNet-34 model was deployed on the CIFAR-10 dataset in the first half and obtained a test accuracy of 86%. Weight pruning was then applied to the architecture, and six fairness measures were calculated before and after the pruning procedure, including the confusion matrix, training and test data split, precision, recall, kappa statistic, and false positive rate (FPR). The VGG-16 model was implemented for gender categorization in the second section. The different race groups were then retroactively compared to determine bias. The model had a test accuracy of 92%, and numerous metrics were used to evaluate it, including precision, recall, the area under the curve (AUC), FPR per race group, false negative rate (FNR) per race group, positive predictive value (PPV), and differential impact (DI).

7. Experiments

For the sake of experimentation, we divided our project into two parts in an attempt to generalize our findings. In the first part, we implemented training a ResNet-34 neural network model on the CIFAR-10 dataset; after which we applied weight pruning on the model. In the second part, we implement VGG-16 on the FairFace dataset; then apply weight pruning on the full model of varying sparsity.

A. Part 1: ResNet-34 on CIFAR-10

As previously stated, in part 1 of our study, we trained a ResNet-34 model using the CIFAR-10 dataset, modified the classification layer, and tested its accuracy.

After the model was effectively trained (before compression techniques were applied) and after the techniques were applied (pruning 50% and 80%), we calculated six fairness metrics:

- Confusion Matrix
- Training and Test Data Split
- Precision - which measures the percentage of true positives among all positives
- Recall - measures the percentage of real positives among all actual positives.
- Kappa Statistic - is a measure of inter-rater agreement. It is useful for assessing the classification model's performance when the distribution of test data is skewed. Furthermore, it is an excellent measure for dealing with class imbalance issues. Kappa Statistic also provides a score between -1 and 1, with 1 indicating perfect agreement and 0 indicating agreement equal to chance.

- False Positive Rate (FPR) - is the ratio of negative instances that are wrongly projected as positive to the total number of negative instances. It is a critical statistic for assessing the performance of a binary classification model.

To begin, we imported the required libraries, which included *torch, torch.nn, torch.optim*, *torch.nn.utils.prune, torchvision, and DataLoader.* Following the definition of the training parameters, such as the number of epochs (140), batch size (128), and learning rate (0.01), the data augmentation transform is defined using *torchvision.transforms.Compose()* performs several operations such as random cropping, horizontal flipping, and normalizing. *Torchvision.datasets.CIFAR10() and DataLoader()* are used to load the CIFAR-10 training dataset. Finally, the pre-trained ResNet-34 model is loaded with *torchvision.models.resnet34()* and transported to the GPU if one is available. We computed the six previously specified fairness metrics after successfully training the model and attaining an acceptable level of accuracy.

After having calculated the fairness metrics, we worked on pruning the network 50% and then 80%.

We defined the pruning and training parameters after importing the appropriate libraries, including PyTorch and torchvision. The prune.ln_structured method is then used to prune the ResNet-34 model on the weights of a single layer. The pruning settings are configured as follows: pruning_percent is 50, n is 2, and dim is 0. This means that 50% of the weights in the chosen layer will be pruned using L2-norm structured pruning with two elements per block along the provided dimension (dim=0 is the channel dimension). The model's classification layer is altered by replacing the fully connected layer with a nn.Linear layer with 512 features that generates 10 classes, the number of classes in the CIFAR-10 dataset.

The pruned model is then trained using the modified classification layer. The following training settings are set: num_epochs is 80, batch_size is 128, and learning_rate is 0.01. The DataLoader function is used to load the training data, and the images and labels are transported to the GPU before training.

During training, SGD with momentum of 0.9 is utilized as the optimizer, while nn.CrossEntropyLoss is used as the loss function. The model is trained for the number of epochs chosen, and the loss is displayed every 100 steps. The testloader and the model.eval() function are used to assess the correctness of the pruned and re-trained model on the CIFAR-10 test set.
We computed the six previously specified fairness metrics after successfully training the model and attaining an acceptable level of accuracy.

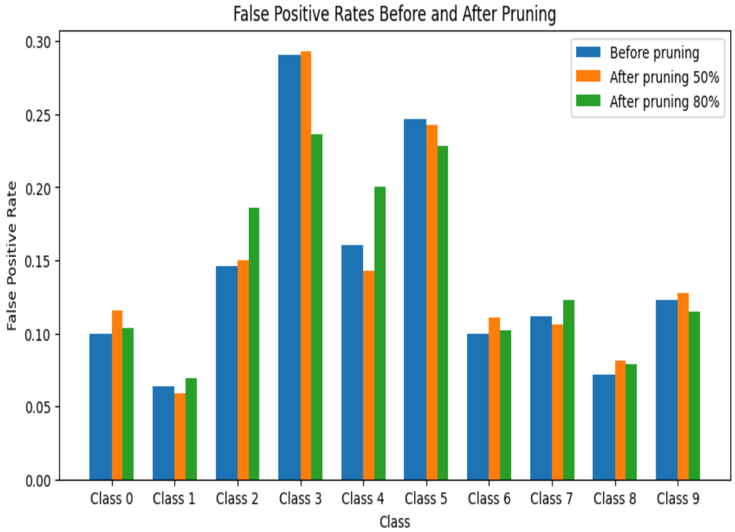
The same process is then repeated for pruning_percent = 80 and computed the six previously specified fairness metrics.

We computed and obtained the results of six fairness metrics. These findings are given in a tabular format, as shown below:

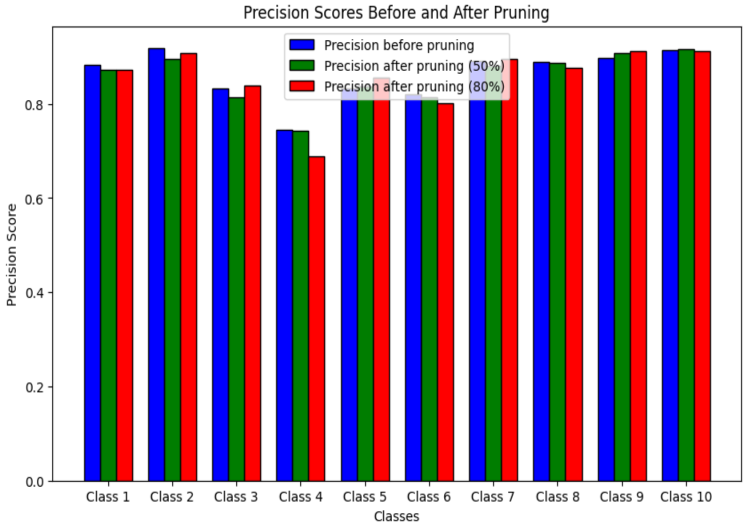
	Before Pruning	After Pruning (50%)	After Pruning (80%)
Accuracy	.862	.845	.826
FPR	[0.1000, 0.0640, 0.1460, 0.2910, 0.1600, 0.2470, 0.1000, 0.1120, 0.0720, 0.1230]	[0.1160, 0.0590, 0.1500, 0.2930, 0.1430,0.2430, 0.1110, 0.1060, 0.0810, 0.1280]	[0.1040. 0.0690, 0.1860, 0.2360, 0.2000, 0.2280, 0.1020, 0.1230, 0.0790, 0.1150]
Precision	[0.882, 0.919, 0.832, 0.746, 0.833, 0.820, 0.892, 0.890, 0.897, 0.914]	[0.872, 0.896, 0.813, 0.744, 0.838, 0.815, 0.880, 0.888, 0.909, 0.917]	[0.872, 0.909, 0.840, 0.688, 0.855, 0.802, 0.895, 0.876, 0.912, 0.912]
Recall	[0.873, 0.934, 0.848, 0.722, 0.888, 0.761, 0.896, 0.916, 0.942, 0.892]	[0.899, 0.927, 0.847, 0.686, 0.844, 0.757, 0.892, 0.901, 0.922, 0.864]	[0.895, 0.934, 0.811, 0.74, 0.808, 0.762, 0.889, 0.883, 0.914, 0.889]

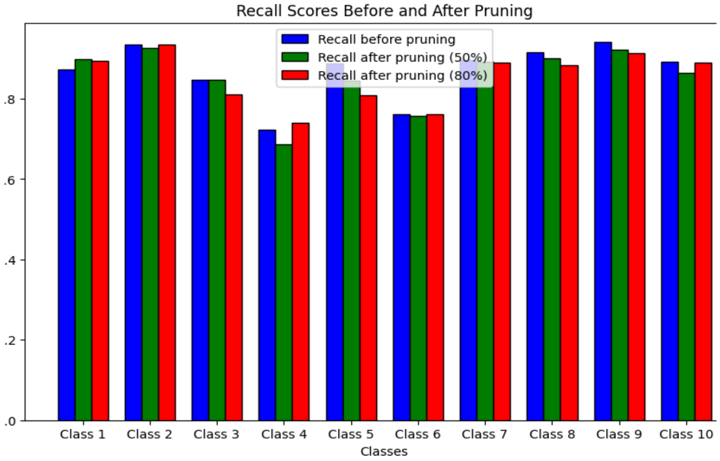
Kappa Statistic	0.8456	0.8397	0.8356

As is clear from the results above, the false positive rates across the ten classes of the CIFAR-10 dataset have increased slightly after pruning.



The precision scores of the model have also decreased after pruning





B. Part 2: VGG-16 on FairFace

For the second experiment, we wanted to use a dataset with faces to perform face identification and classification. Initially, we had intended to use the well-known UTKFace dataset. However, this dataset is also well-known for being considerably biased against non-white individuals. As we are attempting to observe the effect of compression on the bias and fairness in a model, we want to eliminate extraneous causes of bias such as that in the training/validation data.

To mitigate this bias, we instead used the FairFace dataset, which is a considerably larger dataset (UTKFace - 23,708, FairFace - 97,698) that is significantly more representative of the different races than UTKFace. It is important to note that it is not realistic to assume this dataset is entirely free from bias. In each of the models we create, we take the following metrics to compare:

- Precision - overall precision is considering male as the positive class, but this metric is calculated for both positive and negative classes.
- Recall - overall recall is considering male as the positive class, but this metric is calculated for both positive and negative classes.
- Area Under the Curve (AUC) - a binary classification performance metric. We look at the ROC curve as a tradeoff between TPR and FPR. The AUC summarizes the performance of the model on distinguishing between negative and positive cases.
- FPR per race group - we look at proportion of actual female predictions that are classified as male by race group.
- FNR per race group - we consider the proportion of actual male predictions that are classified as female by race group.
- Positive Predictive Value (PPV) - the proportion of male predictions that are actually male.
- Disparate Impact (DI) - the PPV of each race group divided by the PPV of the race group with the highest. This is a helpful fairness metric to determine how similar the positive accuracy is across all race groups relative to each other.

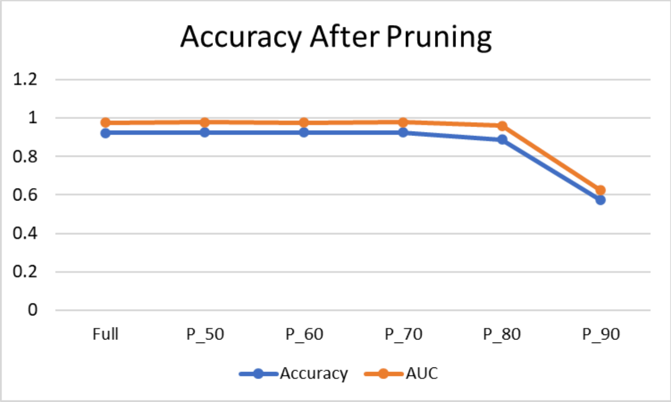
As mentioned previously, we used VGG-16 for this experiment, which is an architecture known for its facial classification. For this experiment, we relied on the keras module in the tensorflow library. The initial model was based on the VGG16 implementation within keras.applications with an input size of (224, 224, 3). We added two layers to this structure. One was a simple Flatten() layer and the other easy a Dense() layer using an activation of ‘sigmoid’ and kernel and bias regularizers of L2 with values 0.01 each. The FairFace images are randomly divided into training, validation, and test sets at a 0.7/0.15/0.15 split respectively.

After applying some simple data augmentation of slight rotations, shifts, rescaling, and horizontal flips, we train the model with a learning_rate of 0.01, 30 epochs, while fine-tuning an additional 20 epochs after initial training. Due to the classification type, we use binary_crossentropy as our loss function. While this would at first glance appear to be a small number of epochs for the size of the data, we did experiment with a larger number of epochs, but this led to some overfitting and poorer test accuracy. The other benefit to a relatively small epoch number is reduced time requirements for training. We ended up with a test accuracy of about 92% on the full model.

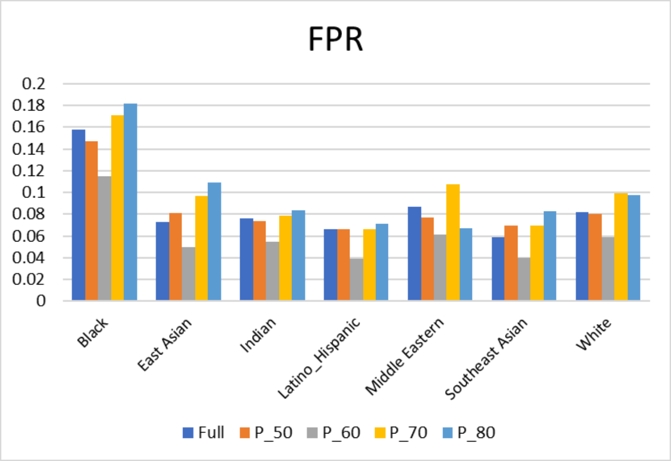
For compression, we started by loading the full model and setting the pruning parameters. This weight pruning included PolynomialDecay() with an initial_sparsity of 0 and a final_sparsity of 0.50-0.90 in intervals of 0.10. The pruning for this experiment was an unstructured approach using the tensorflow_model_optimation function prune_low_magnitude() with 20 epochs of iterative pruning and 10 epochs of fine-tuning. This function iteratively removes the lowest magnitude individual weights to achieve the sparsity desired at any particular step. Again, we experimented with different epoch values here and settled on the values that led to consistently higher resulting accuracy. For both the pruned and full models, we used the keras optimizer Adam, but for the pruned models, we used a learning_rate of 0.0001 (same loss function).

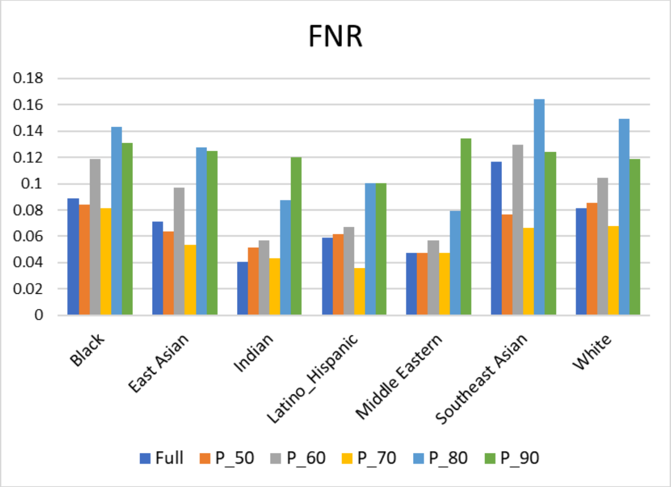
After taking the metrics listed above for all final models, we observed the following:

The accuracy and AUC remained high until intense pruning, with a significant drop-off at 90% weight pruning. Test accuracies were as follows - full: 0.9213, P_50: 0.9239, P_60: 0.9331, P_70: 0.9229, P_80: 0.8873, P_90: 0.5731.

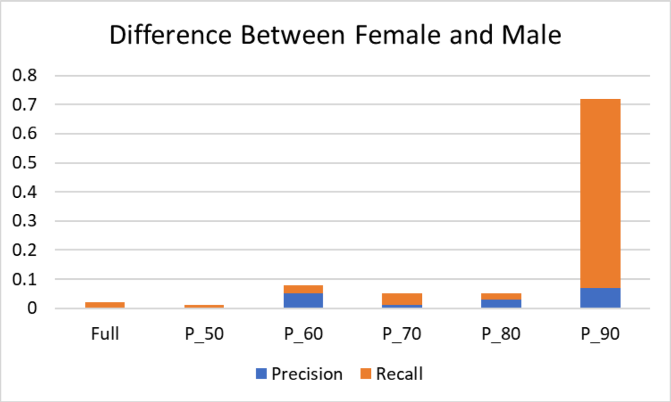


The FPR of the model as it is pruned ends up with a U-shape curve across all race groups. Surprisingly, the FPR first increases for pruning of 50% and 60%, then increases for greater pruning. 90% is removed from this chart below as it is significantly higher than all other models and makes the chart hard to interpret. The FNR, however, followed a slightly more linear trend, with the 70% pruned model performing the best. The other thing to note is that the 90% pruned model’s FNR is comparative to the other models whereas it was far higher for FPR.

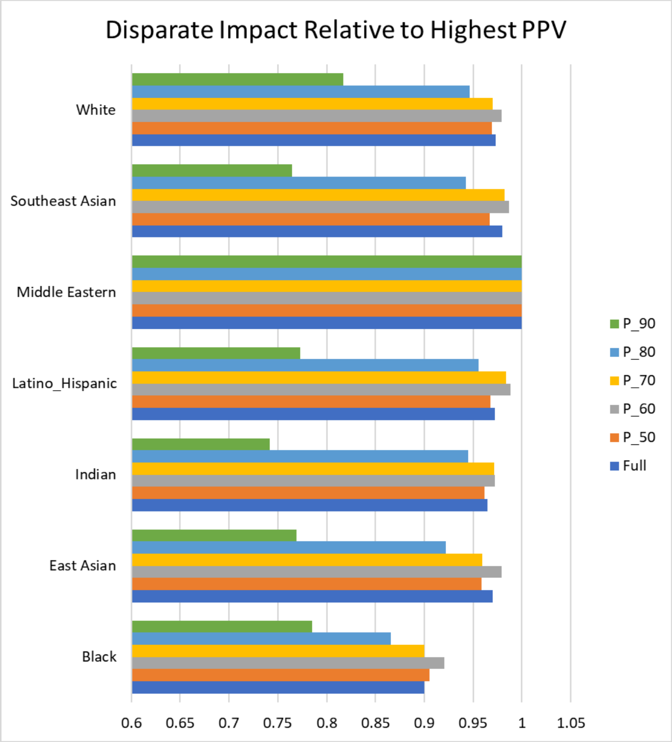




To indicate the increase in bias for the genders during classification instead of only on race group, we also compare the precision and recall between male and female classifications as the positive class across all models. We see a minor decrease in both for 50% pruning. From there, we observe an inconsistent fluctuation between the two differences. By the 90% pruned model, we see a massive difference in recall but still a small difference in precision.



As a different approach to measuring fairness, we compare the PPVs of each race group to that of the highest group. Across all models, the Middle Eastern group had the highest and Black had the lowest PPV (other than 90% pruned). By this metric we observe the 60% pruned model as the fairest. There is first a slight decrease, then a peak, then a steady decline as we prune more. This trend is seen in other metrics as well.



The metrics we took from the data shed some light on some trends of bias with compression. We will discuss our takeaways from this data below.

8. Discussion

From the Part 1 experiments, we can conclude that the model's fairness before and after pruning is mixed, and it may be affected by the specific fairness metric utilized. Here's a breakdown of what we can conclude:

- **Accuracy**: As we increase the pruning rate, the accuracy decreases, suggesting that the model becomes less accurate after pruning. However, the difference in accuracy before and after pruning is not substantial, implying that pruning had little effect on the model's overall performance.
- **False Positive Rate (FPR)**: The FPR changes depending on the class and pruning level. After pruning, the FPR declines in some circumstances and increases in others. In class 2, for example, the FPR

drops from 0.0640 to 0.0590 after 50% pruning but rises to 0.0690 after 80% pruning. Similarly, after 50% pruning, the FPR in class 4 falls from 0.1600 to 0.1430 then rises to 0.2000 after 80% pruning. This implies that pruning may have varied implications on the model's fairness for different classes.

- **Precision**: Precision varies depending on the class and level of pruning. In certain circumstances, pruning enhances precision, while in others, it diminishes. In class 1, for example, precision rises from 0.882 to 0.919 after 50% pruning but falls to 0.872 after 80% pruning. Similarly, after 50% pruning, precision increases from 0.833 to 0.838, but lowers to 0.802 after 80% pruning. This implies that pruning may have varied effects on model precision for various classes.
- **Recall**: The recall varies depending on the class and pruning level. In certain circumstances, pruning enhances recall while in others, it diminishes. In class 1, for example, recall rises from 0.873 to 0.934 after 50% cutting but falls to 0.895 after 80% pruning. Similarly, after 50% pruning, recall increases from 0.888 to 0.844 but reduces to 0.808 after 80% pruning. This implies that pruning may have varied effects on the model's recall for different classes.
- **Kappa Statistic**: After pruning, the Kappa Statistic lowers slightly, indicating that pruning may have a negative impact on the agreement between anticipated and real labels.

In this experiment, the diverse effects of pruning on several fairness metrics imply that the model's fairness before and after pruning is nuanced and context-dependent.

We get some insightful and interesting results from Part 2 as well. While we do not observe as clear a pattern as would be conclusive, we still can see certain trends with the compression and the metrics we measured.

- **FPR and FNR**: We observe a small decrease in both FPR and FNR per race group after pruning 50%. This, as well as the other performance gains, can likely be attributed to improved generalization. More interestingly, we start to see a bit of a tradeoff between FPR and FNR as we reach 60 and 70% pruning. The 60% pruned model achieves the lowest FPR for all groups, but also sees a spike in FNR. The complete opposite observation can be seen with the 70% model. The 80% pruned model also maintains a similar FPR to the other models and a more noticeable spike in FNR. The 90% pruned model only observes a slight increase in FNR, but a massive increase in FPR. These observations indicate a clear tradeoff between these values as the pruning gets more intense.
- **Precision and Recall by Gender**: We observe another minor pattern when comparing the difference in precision and recall on both male and female classification. There are some minor fluctuations in this difference when pruning additional percentages of weights, but after 90% pruning, the gap is immense. This indicates that the model ends up heavily favoring one classification over the other. However, as stated, this discrepancy remains minimal through 80% weight pruning.
- **Disparate Impact**: One of the more interesting measures of fairness, this metric is independent of accuracy and other performance measures. Despite that, we still observe a similar pattern between the models as we do with the accuracy. This pattern resembles more of a bell curve, where fairness remains fairly constant and even slightly increases after moderate pruning. Once we reach a sparsity of 80%, this fairness quickly decreases. Surprisingly, even the 70% model has similar fairness to the full model by this metric. Since this measure is independent of overall accuracy, it provides greater evidence to the patterns we observe and enables us to infer that the other observations did not only occur due to patterns in overall performance.

It is clear from the data we gathered that the bias and fairness in a model and how it is affected by compression is not entirely comprehensive. Through various metrics, we show that moderate compression does not necessarily lead to any increase in bias. In fact, to an extent, this pruning can actually decrease this bias. We attribute this detail to the same characteristic as the increase in accuracy after moderate pruning: generalization. We see fluctuations between various sparsity models regarding the FPR and FNR tradeoff as well as precision and recall on both gender classifications. These observed fluctuations indicate that the bias does not necessarily persist from the full model to pruned models regarding classification. In terms of fairness on race groups, very similar patterns are seen through all models, potentially indicating some bias of this feature in the training data.

9. Conclusion

In conclusion, our original hypothesis that there exists a tradeoff between resource efficiency, accuracy, and fairness in deep neural networks was partially confirmed through our proposed approach. In Part 1 of our approach after weight pruning was applied, we noticed a decrease in accuracy, precision, and kappa statistic, as well as an increase in false positive rates, suggesting the model may not perform as well in identifying certain classes, potentially leading to biased outcomes. In Part 2 of our approach, we found that moderate pruning did not result in significant drops in fairness, likely due to generalization. In fact, up to 60% (and 70% in some cases) pruning, the performance and fairness actually improved by some metrics. However, 80% and 90% pruning resulted in large drops in both accuracy and fairness. Even when isolating the influence of overall accuracy and performance with Disparate Impact, the moderate pruning up to 70% actually increased or maintained similar fairness to the full model. Overall, our results suggest that there is a correlation between accuracy and fairness, rather than a trade-off, in the context of our experiment. This conclusion is most greatly supported by the pattern in Disparate Impact by race showing similarities to those of bias metrics that rely on performance. Further research is needed to determine if this holds true in other settings and with other datasets. While pruning can assist in increasing the model's efficiency, it is critical to closely check the fairness measures to avoid unintentional biases. As machine learning continues to impact our environment, it is critical to work toward establishing models that are fair and unbiased in order to benefit all individuals and communities. We hope our study may contribute to understanding this topic and influence researchers to extensively measure bias and fairness in the creation and compression of their models.

10. References

[1] Xuanqi Gao, Juan Zhai, Shiqing Ma, Chao Shen, Yufei Chen, and Qian Wang. 2022. FairNeuron: improving deep neural network fairness with adversary games on selective neurons. In Proceedings of the 44th International Conference on Software Engineering (ICSE '22). Association for Computing Machinery, New York, NY, USA, 921–933. https://doi.org/10.1145/3510003.3510087

[2] Stoychev, S., & Gunes, H. (2022). The effect of model compression on fairness in facial expression recognition. arXiv preprint arXiv:2201.01709.

[3] Garrido-Muñoz I, Montejo-Ráez A, Martínez-Santiago F, Ureña-López LA. A Survey on Bias in Deep NLP. Applied Sciences. 2021; 11(7):3184. https://doi.org/10.3390/app11073184

[4] C. Blakeney, N. Huish, Y. Yan, and Z. Zong, “Simon Says: Evaluating and Mitigating Bias in Pruned Neural Networks with Knowledge Distillation.” [Online]. Available: https://github.com/codestar12/pruning-distilation-bias (accessed Feb. 24, 2023).

[5] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, “SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability,” Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 6077–6086, 2017

[6] Hooker, Sara, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. "What do compressed deep neural networks forget?." arXiv preprint arXiv:1911.05248 (2019)

[7] Hooker, S., Moorosi, N., Clark, G., Bengio, S., & Denton, E. (2020). Characterising bias in compressed models. arXiv preprint arXiv:2010.03058.

[8] Proskurina, I., Metzler, G., & Velcin, J. (2023, April). The Other Side of Compression: Measuring Bias in Pruned Transformers. In Advances in Intelligent Data Analysis XXI: 21st International Symposium on Intelligent Data Analysis, IDA 2023, Louvain-la-Neuve, Belgium, April 12–14, 2023, Proceedings (pp. 366-378). Cham: Springer Nature Switzerland.