a. Methodology

We experimented with a total of 5 methods.

1. Pruning weights according to magnitude layer by layer: For each layer with weights in the network (i.e. Conv2D layer and Denselayer), rank absolute value of weights, set a certain percentage to 0 based on the rank (prune starting from weights with small absolute values). Then, freeze the weights that are 0 in finetuning and keep training the network. The two steps are one iteration and one can do multiple iterations.
2. L1-norm based filter pruning: For each Conv2D layer, compute L1 norm of each filter, prune a certain percentage of filters based on L1 norm (prune starting from small L1 norms).
3. L2-norm based filter pruning: For each Conv2D layer, compute the L2 norm of each filter, prune a certain percentage of filters based on L2 norm (prune starting from small L2 norms).
4. Neuron pruning: For each Dense layer, compute L2 norm of the weight of each neuron, prune a certain percentage of weight vectors based on the L2 norm (prune starting from small L2 norm).
5. Data-Driven Sparse Structure Selection: This method is introduced by the paper titled "Data-Driven Sparse Structure Selection for Deep Neural Networks". The method introduces a new type of parameter called scaling factors, which are applied to the outputs of certain structures within the network, such as neurons, groups, or blocks. These scaling factors are then regularized by sparsity regularizations (L1 regularizations), which encourage them to approach zero during training. After training, we remove the scaling factors and those structures whose corresponding scaling factors' absolute values are below a threshold, and then retrain the model.

b. Comparison

Implementation Ease:

The first four methods, magnitude-based layer-by-layer pruning, L1-norm based filter pruning, L2-norm based filter pruning, and neuron pruning are all relatively straightforward to implement, majorly including steps like calculating a value for each candidate structure and ranking the values and pruning based on the values. However, the iterative fine-tuning process, which can be added on top of all these methods to improve model accuracy, adds a layer of complexity to the overall management of the implementation. The last method, Data-Driven Sparse Structure Selection, is relatively hard to implement, which including steps like implementing new layers (scaling factors) and inserting these into the original models and removing the scaling factors after pruning. This method is data-driven, we need to train the model to get the value of the scaling factors and remove corresponding structures based on the value of the scaling factors.

Performance and Model Score:

In terms of performance, when only considering the Conv2D layer weights, i.e. L1 and L2 norm-based filter prunings, achieve only about 5% sparsity in Conv2D layer (which is only 0.5% sparsity gain of the entire network) without a significant drop in accuracy (meaning maintaining accuracy above 60%). Due to this limited impact, they are not included in the Pareto front graph in our analysis. On the other hand, neuron pruning, which reduces weights structurally by zeroing out all weights associated with a neuron, and magnitude-based pruning, which targets smaller weights, demonstrate more significant effects. As for Data-Driven Sparse Structure Selection, it achieves the best result, with an accuracy of 0.7264, a sparsity of 0.985, and a challenging score of 0.856. We believe this is because of its data-driven nature. It effectively tailors the pruning process to the specific data and architecture of the network, allowing for a more targeted and efficient reduction of redundancy without compromising a lot of the network's accuracy.

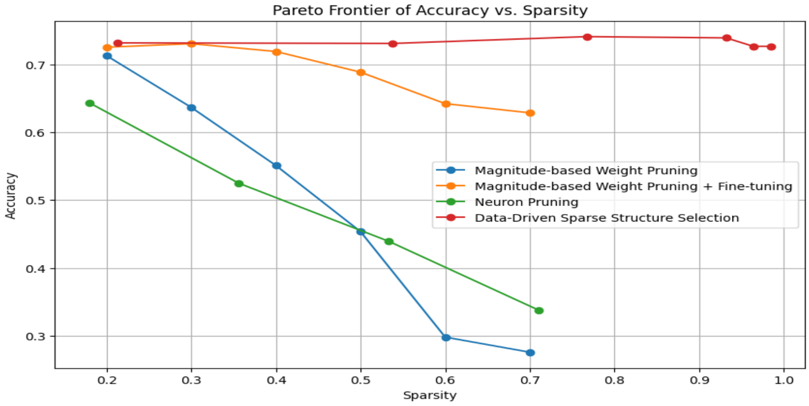
In the Pareto Frontier Analysis, we will show the Pareto Frontier of magnitude-based layer-by-layer pruning, magnitude-based layer-by-layer pruning with one round of fine-tuning, neuron pruning, and Data-Driven Sparse Structure Selection.

Pareto Frontier Analysis:

(We find the Accuracy vs Sparsity plot is easier to observe compared with the Sparsity vs Accuracy plot, so we choose to plot the former.)

The Pareto frontier, illustrating the trade-off between Accuracy and Sparsity, shows similar trends for neuron pruning and magnitude-based pruning. At lower levels of sparsity, unstructured pruning method (magnitude-based pruning) slightly outperform structured method (neuron pruning). Conversely, at higher levels of sparsity, structured method (neuron pruning) show slightly better preservation of accuracy without fine-tuning. With multiple rounds of fine-tuning, both methods can restore accuracy to 70%+. As for Data-Driven Sparse Structure Selection, it demonstrates an impressive trend: as sparsity increases, accuracy remains consistently high. This is likely due to its data-driven nature and ability to selectively prune less impactful structures with knowledge learned from data. It effectively tailors the pruning process to the specific data and architecture of the network, allowing for a more targeted and efficient reduction of redundancy without compromising a lot of the network's accuracy.

PS: For Data-Driven Sparse Structure Selection, the models were retained to reach
convergence after pruning and L1 regularization was used in retraining and weights with absolute values under thresholds are further removed after retraining.



c. Reflection

Did the results match what you expected? What did you learn about model pruning as part of this project?

Yes, the results for Data-Driven Sparse Structure Selection were somewhat surprising but also quite promising. The ability of this method to maintain high accuracy while significantly increasing sparsity challenges conventional expectations about the trade-offs between network efficiency and performance. As we discussed above, we believe this is because of its data-driven nature. We are able to achieve a score of 0.856, with 98.5% sparsity and 72.6% accuracy. As for other methods, they also effectively increase sparsity. However, since they did not learn knowledge from data to prune, we can observe a significant decrease in accuracy as sparsity goes up.

TAKEAWAYS:

High Sparsity with Maintained Accuracy: It was particularly revealing that accuracy could be almost restored to pre-pruning levels even at an extreme sparsity of 98.5%. This highlights the effectiveness of the data-driven pruning techniques.

Importance of Dense Layers: Dense layer has significantly more weights in the given structure, so it has more potential than the Conv layers in terms of pruning.

Redundancy in Dense Layers: Dense layers also structurally have more redundant weights, allowing for a more substantial reduction in their weights without significantly impacting model performance.

Iterative Fine-tuning and Efficiency of Pruning: While iterative fine-tuning significantly restores accuracy post-pruning, it is computationally intensive and time-consuming to achieve very high scores. Although computational cost is not included in the scoring criteria for this mini-project, it is crucial to balance the desired pruning outcomes with the computational resources required. Nevertheless, reducing the number of weights effectively saves storage space and enhances inference speed.

These insights have significantly enhanced our understanding of model pruning, offering strategic directions for future projects.