Project 7: Investigating Parameter Sharing in Neural Networks (Group 40)

Justine Fastner (s242066), Stefanie Lanz (s242107), Niclas Pokel (s242105), Maximilian Riedl (s241710)
Supervisor: Laurits Fredsgaard Larsen (laula@dtu.dk)
Technical University of Denmark

Project Settings

Modern ensemble methods, such as BatchEnsemble, are renowned for their ability to enhance model accuracy and robustness by leveraging predictions from multiple neural networks. However, scaling the number of ensemble members often results in increased computational demands, posing significant resource challenges [1]. This project investigates novel strategies for reducing the parameter footprint in neural networks, with a focus on enabling computationally efficient training. By exploring parameter-sharing mechanisms within single networks and across network layers, we aim to retain the high predictive performance and reliability inherent to BatchEnsemble while minimizing computational overhead.

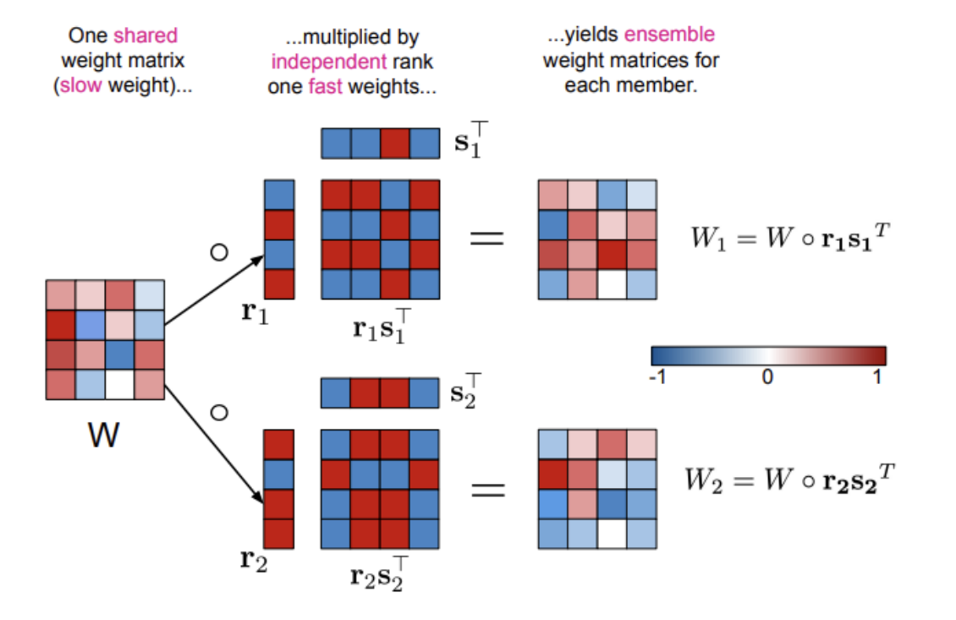
One shared weight matrix (slow weight)...
...multiplied by independent rank one fast weights...
...yields ensemble weight matrices for each member.
$W_1 = W \circ r_1 s_1^T$
$W_2 = W \circ r_2 s_2^T$

Figure 1: BatchEnsemble example with two members [1].

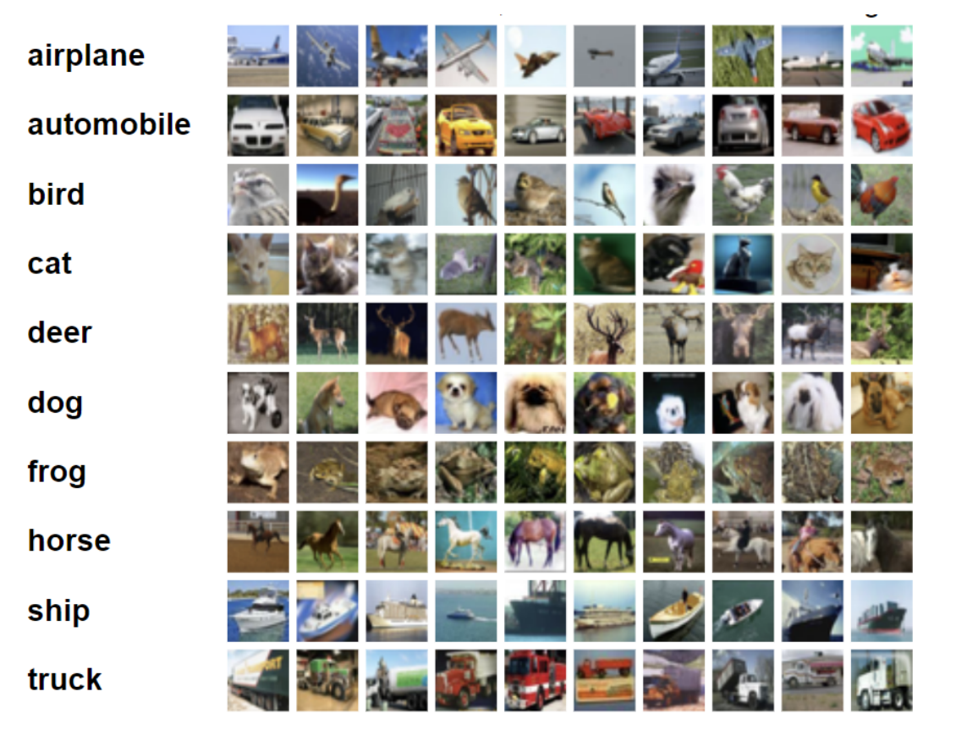

Figure 2: CIFAR-10 dataset with each class and ten random images [2].

Model Specifications

This study utilizes the CIFAR-10 dataset, consisting of 60,000 color images with a resolution of 32x32 pixels, divided into 10 classes (6,000 images per class). The dataset includes 50,000 training images and 10,000 test images, as illustrated in Figure 2.

Our implementation combines Conv2d and BatchNorm2d layers with BatchEnsemble-specific BELinear layers to normalize and enhance the model. We evaluated performance on both simple and complex CNN architectures, including a BatchEnsembleCNN derived from the VGG11 architecture, chosen for its balance of computational efficiency and complexity.

The study also compares the optimization performance of Adam, SGD, and IVON optimizers, as discussed in [3]. Additionally, an initial attempt at parameter sharing across the model was conducted. Figure 1 visualizes the matrix structure and key functionalities of the proposed approach.

Ideas and Experiments

The study explores the effects of varying alpha, gamma, and ensemble sizes on the performance of BatchEnsemble models, with a particular focus on shared members. Performance was visualized using metrics such as heatmaps, box plots and line diagrams. A key area of investigation was the role of shared members, both across layers and within individual layers. Experiments included varying ensemble sizes and parameter sharing across different models. Typer was utilized to compare optimizers and visualize the diversity within ensembles.

Results

Optimizer Testing

For comparison of a single VGG11-network to an ensemble of 8 members (VGG11-networks) we chose three optimizers. Adam and SGD are well known optimizers while IVON is a new optimizer. We wanted to include IVON for a first try and see if further research about using it in BatchEnsembles is reasonable. We used alpha = 1.0 and gamma = 0.2 from the Parameter Testing Experiments.

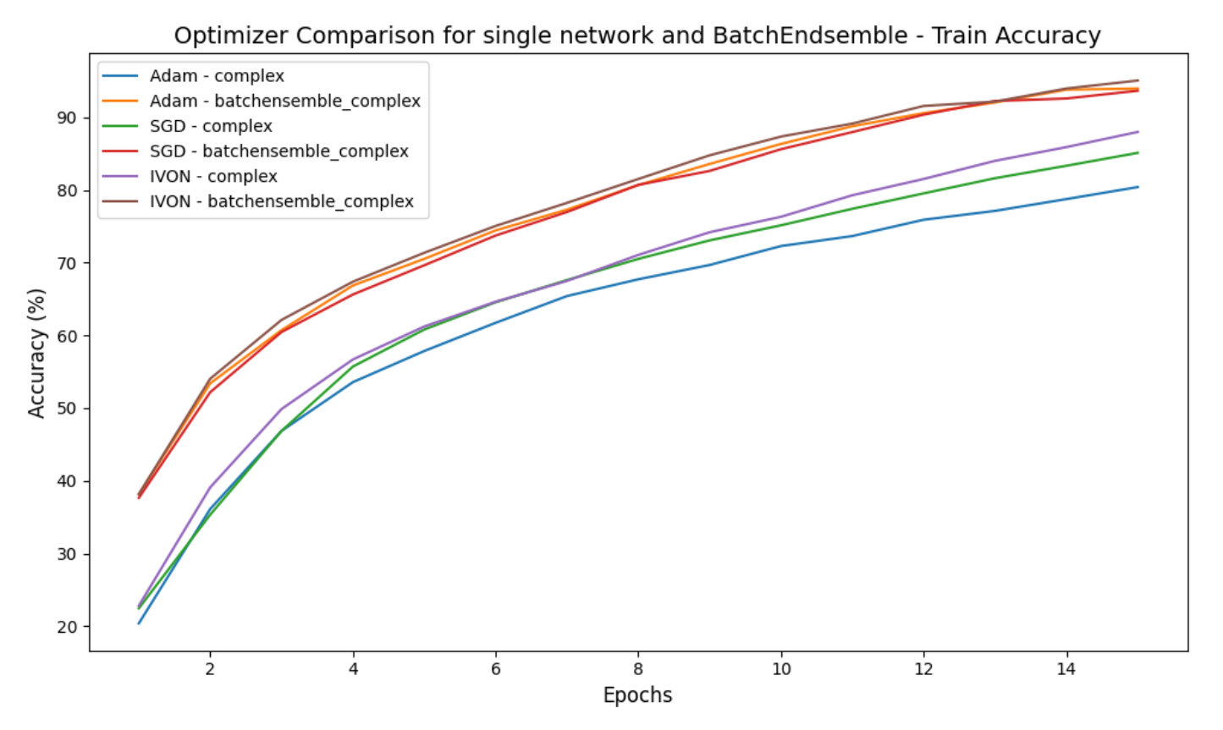

Figure 3: Optimizer Comparison - Train Accuracy

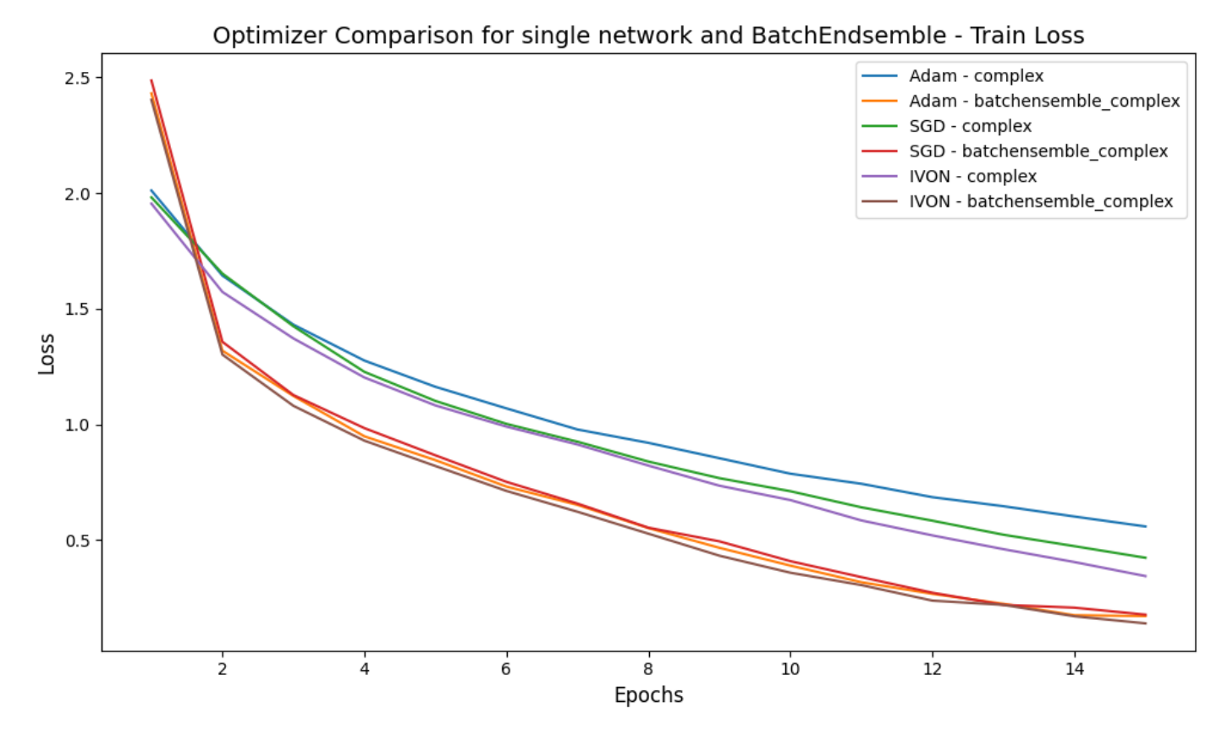

Figure 4: Optimizer Comparison - Train Loss

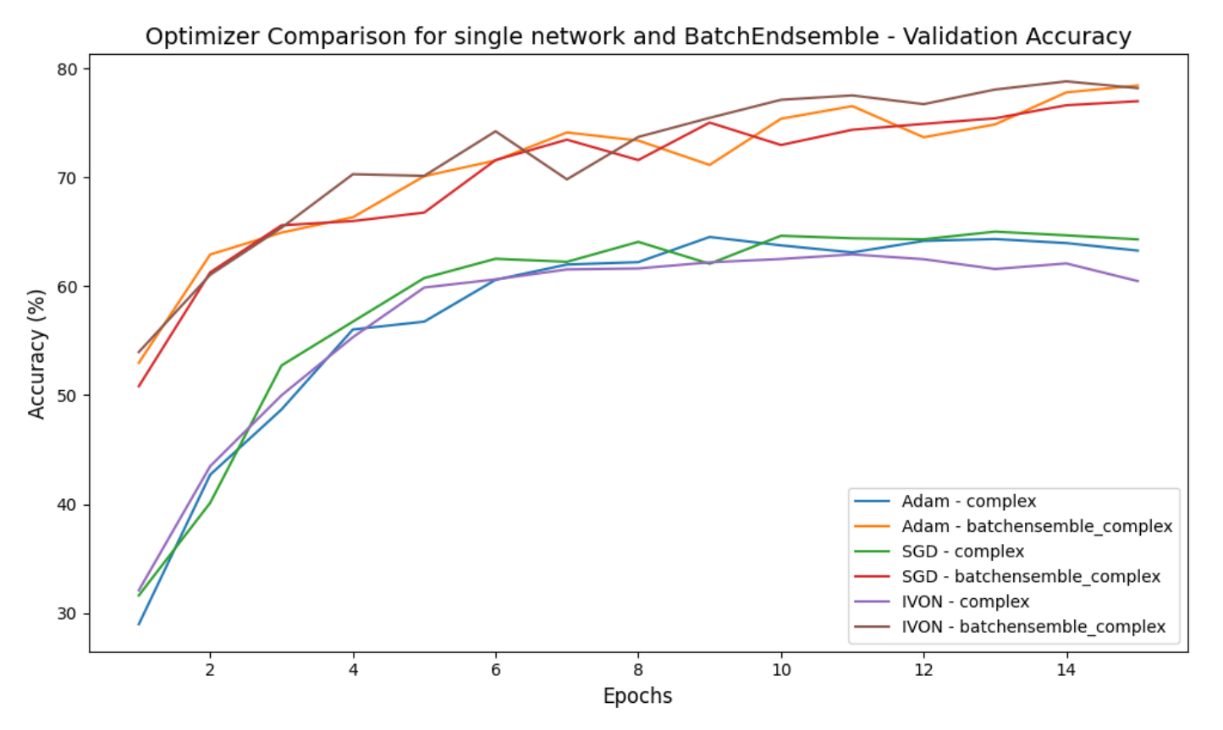

Figure 5: Optimizer Comparison - Validation Accuracy

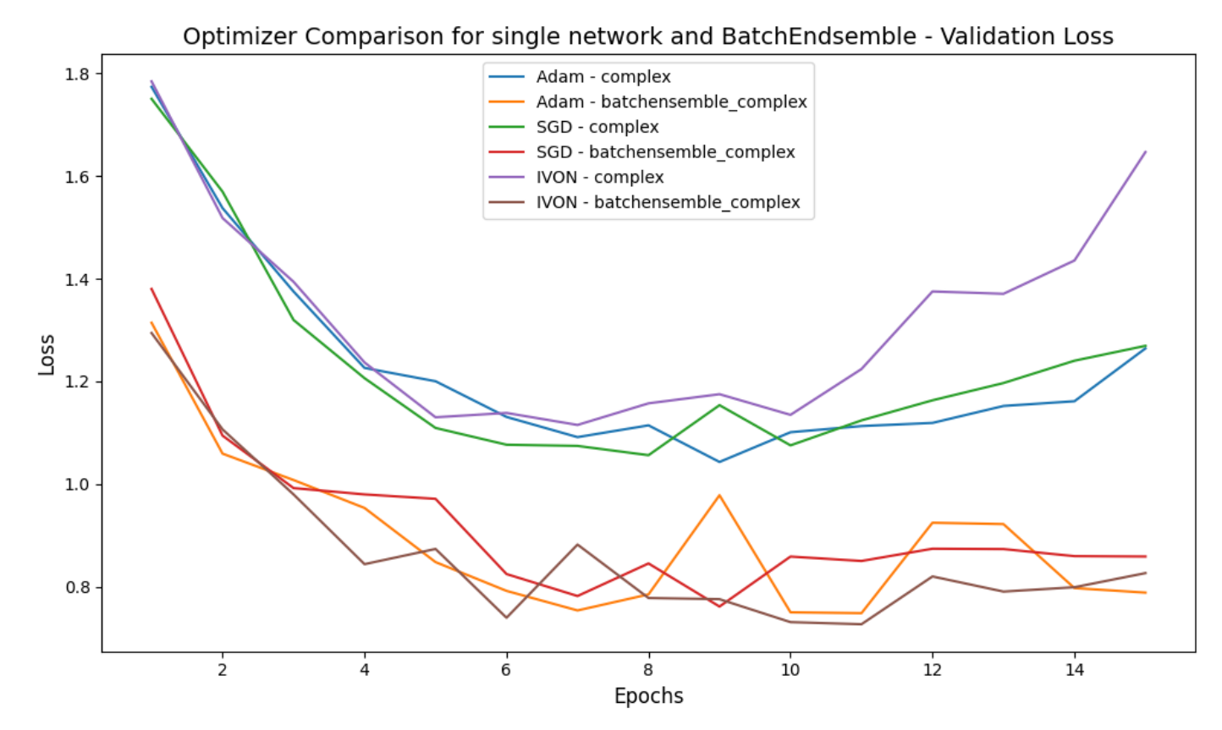

Figure 6: Optimizer Comparison - Validation Loss

Results

Parameter Testing

Comparison of different alpha and gamma values: -1, -0.8, -0.5, -0.2, 0, 0.2, 0.5, 0.8, 1.0. Visualization for test accuracy for last value and for average value (Figure 7 and Figure 8), as well as visualization for losses at last value and average value (Figure 9 and Figure 10).

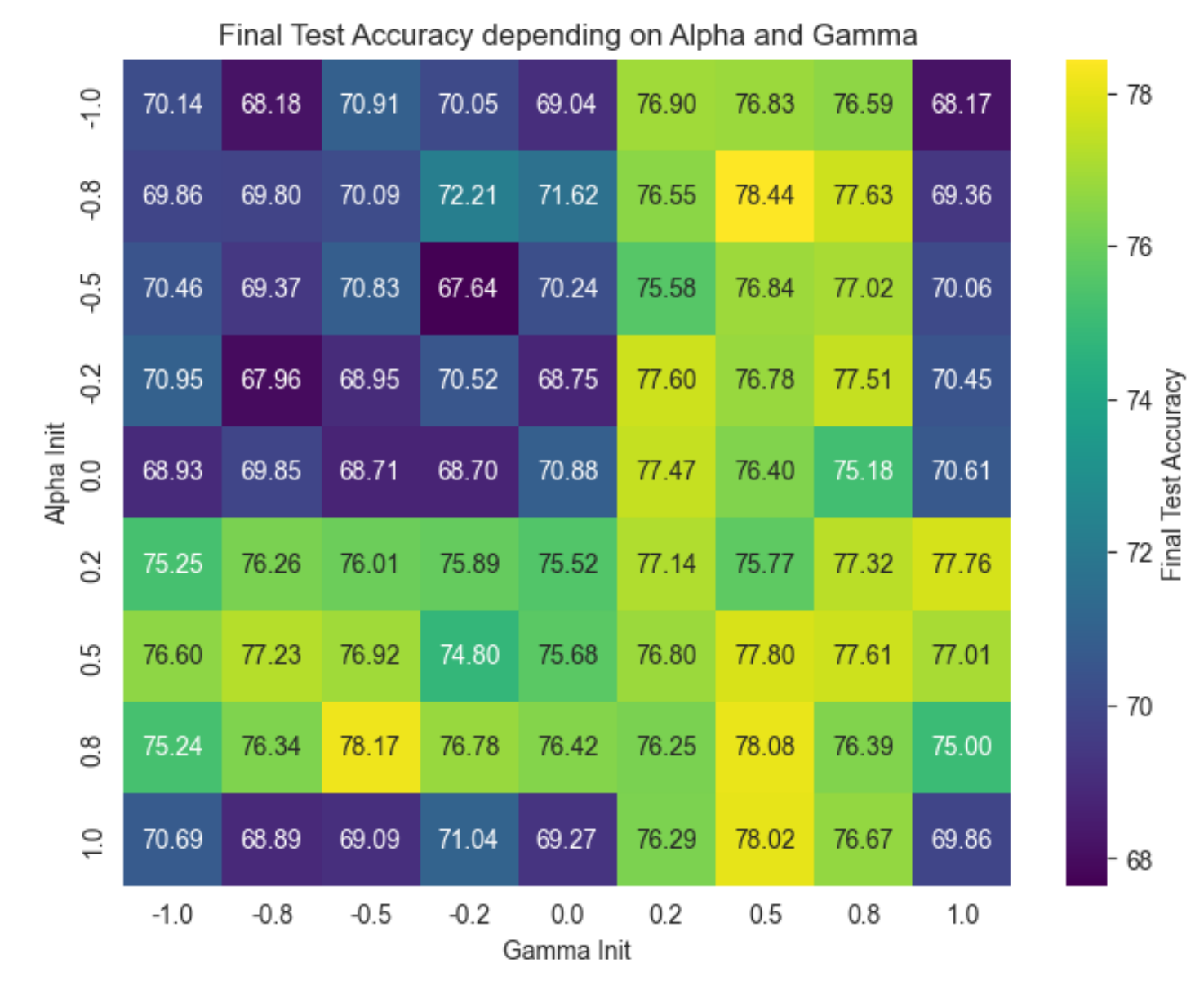

Figure 7: Different Alpha and Gamma for Final Test Accuracy

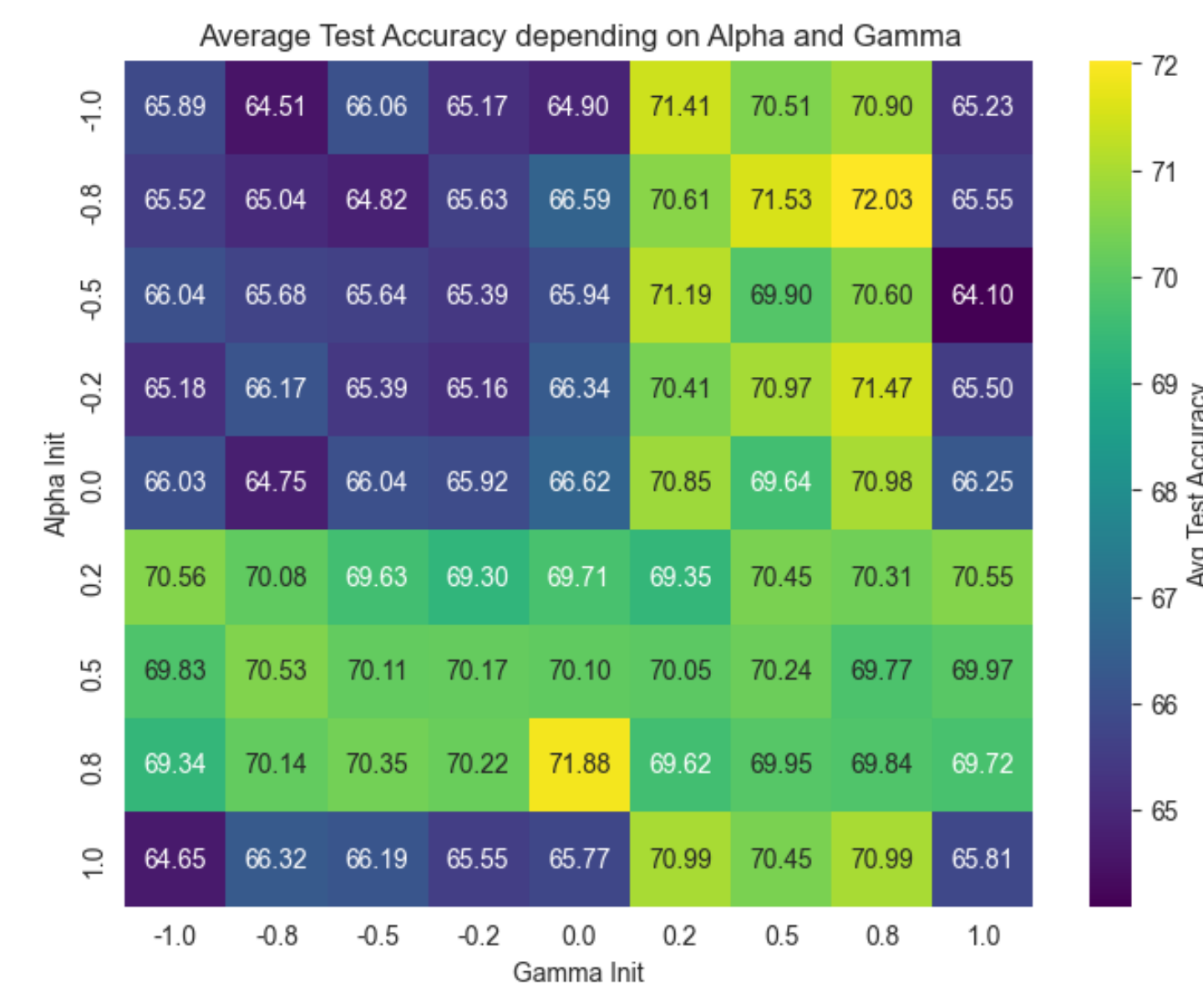

Figure 8: Different Alpha and Gamma for Avg Test Accuracy

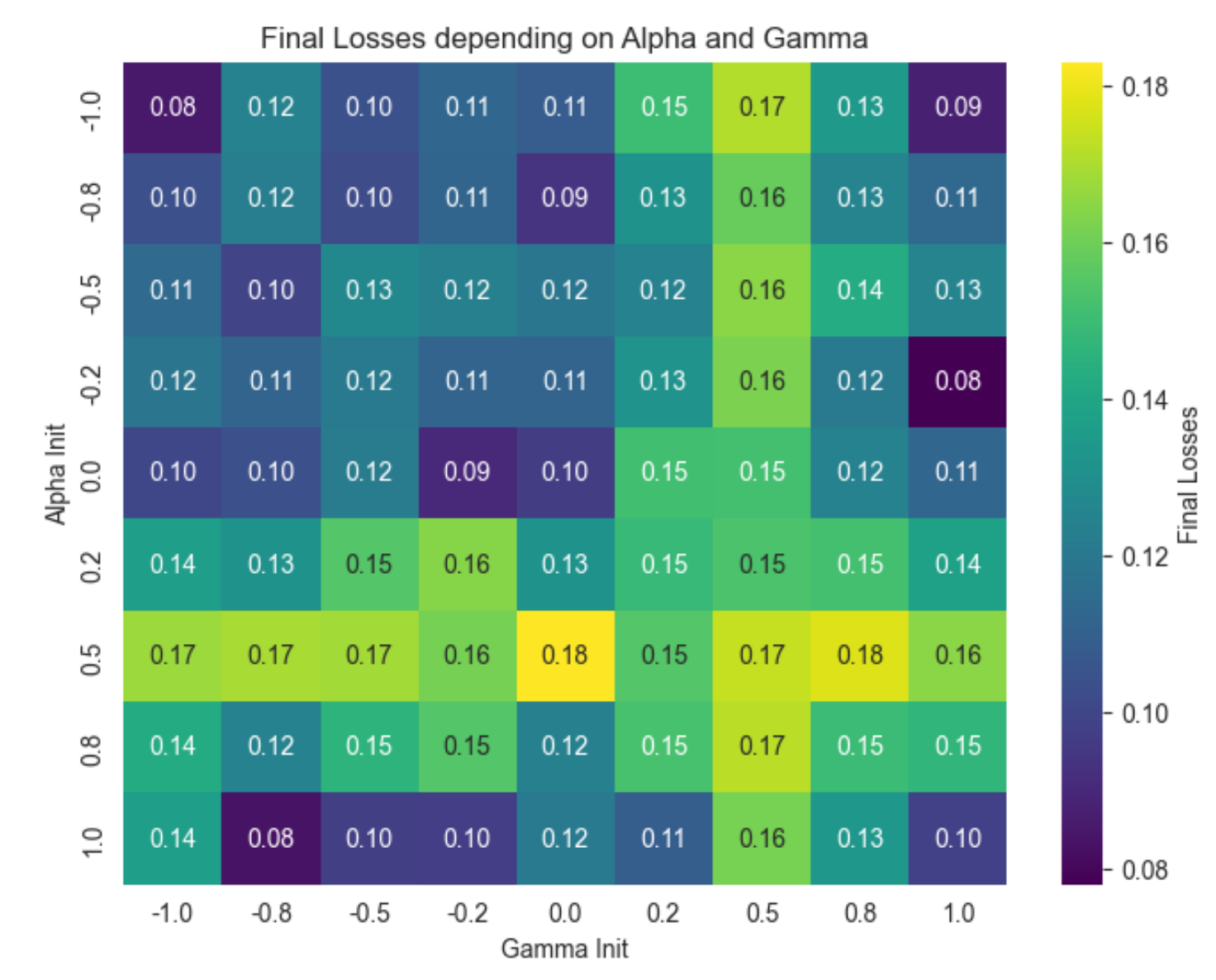

Figure 9: Different Alpha and Gamma for Final Loss

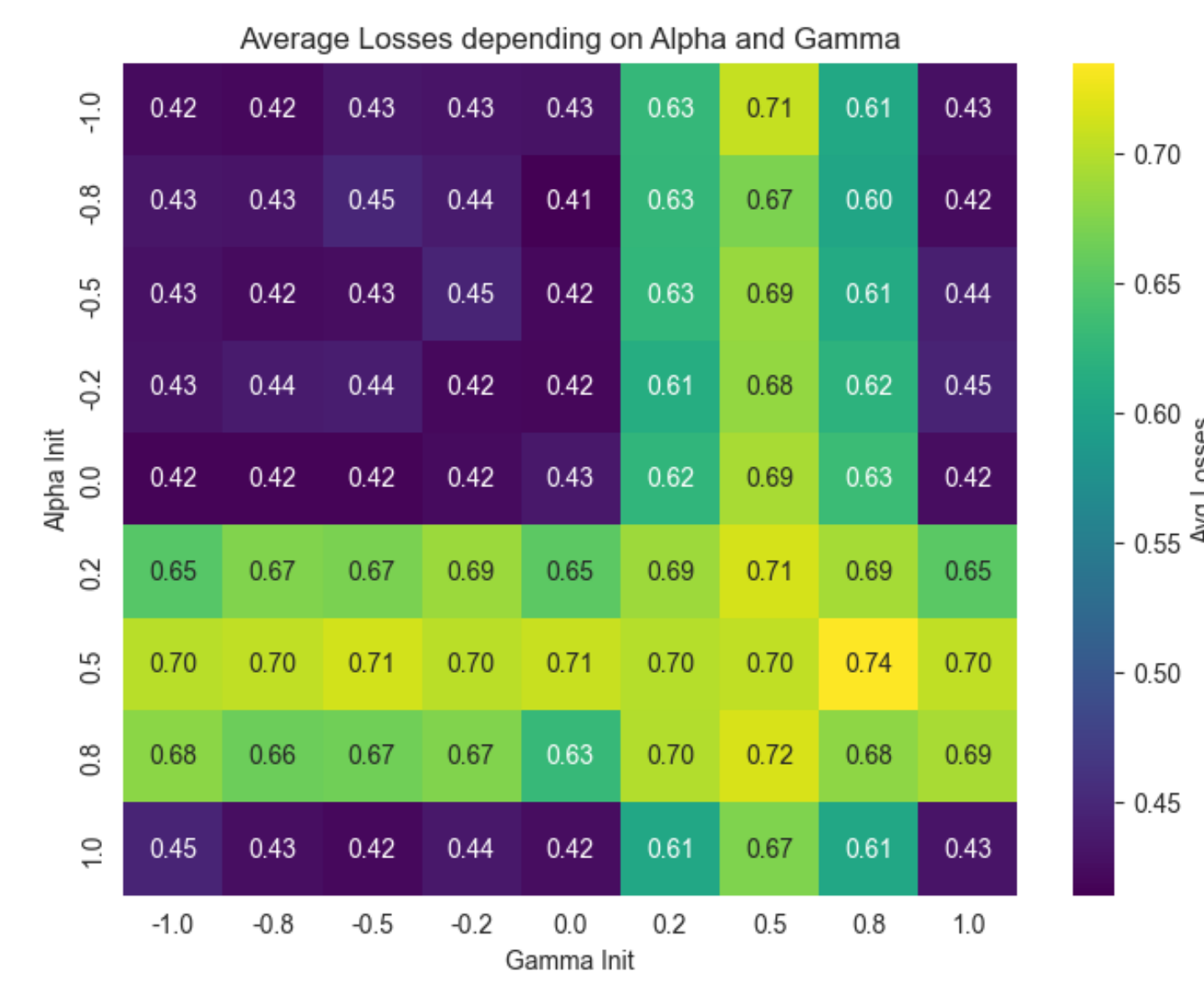

Figure 10: Different Alpha and Gamma for Avg Loss

Comparison of the different ensemble sizes: 2, 3, 4, 5, 6, 7, 8, 9. Additionally depending on different learning rates: 0.02, 0.002, 0.0002. Visualization of the test accuracy for the last value and for the average value (Figure 11), as well as visualization of the losses at the last value and at the average value (Figure 12).

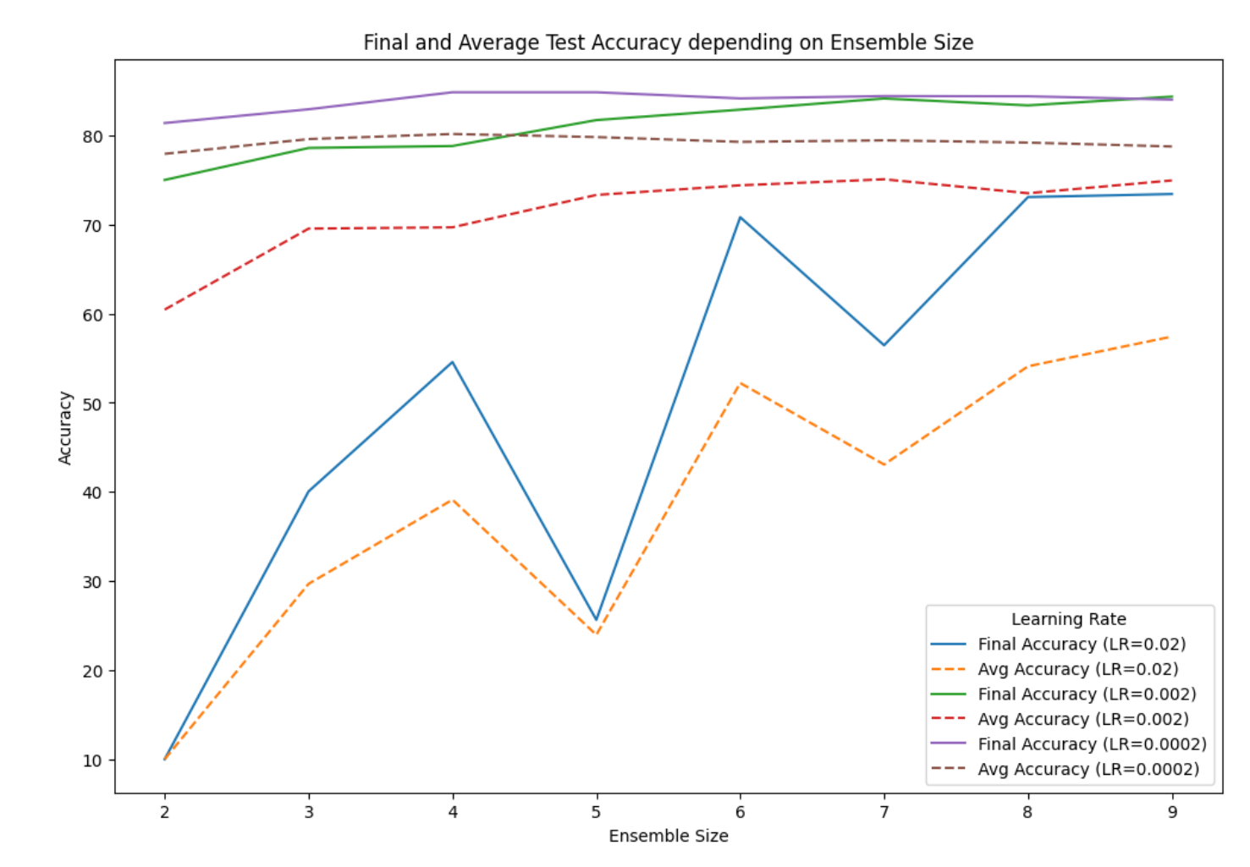

Figure 11: Ensemble Size for Test Accuracy

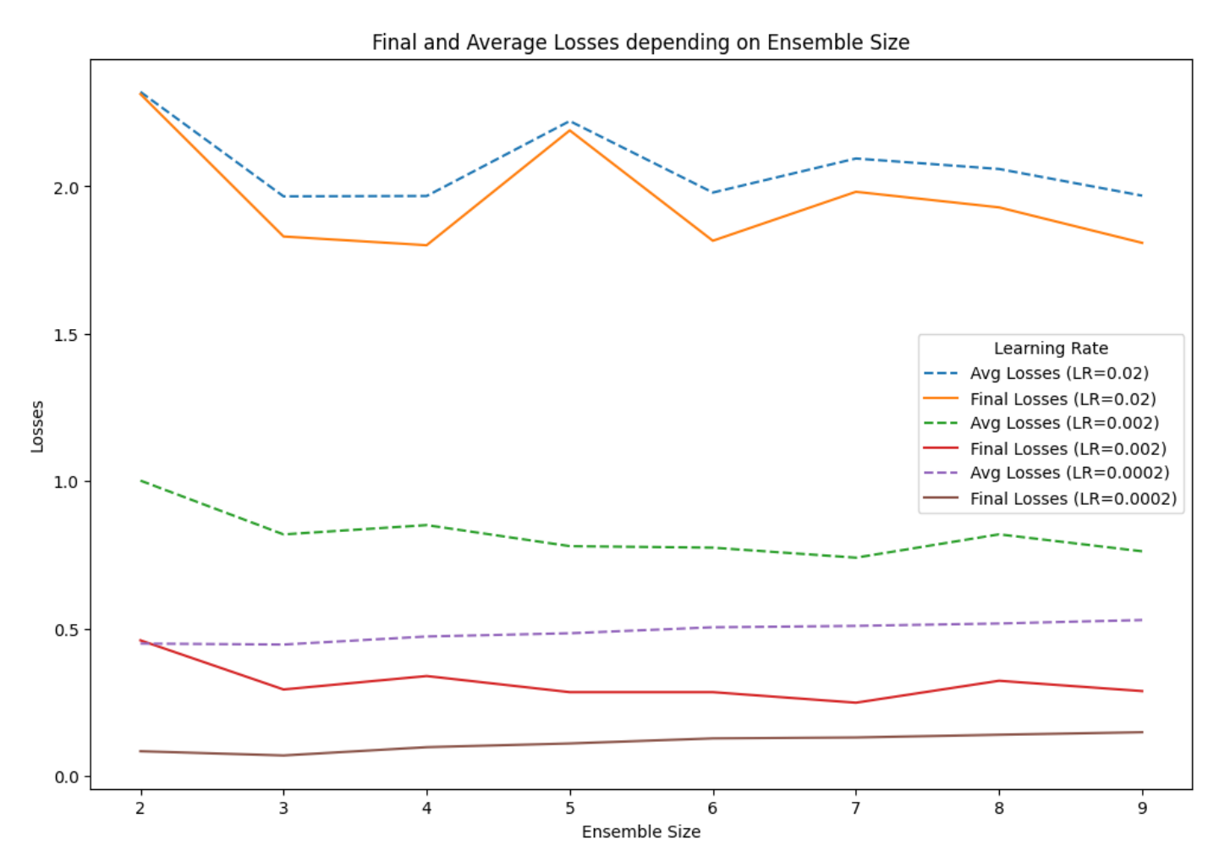

Figure 12: Ensemble Size for Loss

Conclusion

Parameter Testing: The best test accuracy is achieved when either alpha or gamma is between 0.2 and 0.8, while the other value is less critical. Increasing the ensemble size improves test accuracy and reduces losses, and smaller learning rates also result in higher accuracy and lower losses. However, the differences between learning rates of 0.002 and 0.0002, as well as between ensemble sizes of 7, 8, and 9, are minimal.

Optimizer: BatchEnsemble delivers higher accuracy and lower loss for testing, which indicates that an ensemble is leading to better results than a single network. Looking at IVON for the single network and the BatchEnsemble leads to overfit the data. Even more than Adam and SGD. It could be better to stop after 10 epochs instead of 15.

Parameter Sharing: Early layer sharing has a higher performance impact because these layers extract high-level features critical for deeper layers, making their weights more relevant to overall performance. Models with both early and deep layer sharing are less affected, possibly because shared layers focus on extracting simpler yet effective features, while unshared deep layers may attempt to identify overly complex patterns from these simpler outputs. Parameter Sharing, when done on the deeper layers reduces time complexity with minimal impact on the final performance.

Results

Parameter Sharing

To test different degrees and methods of parameter sharing, three sharing regimens with three sub-methods were employed, as shown in the following plot. The weights are shared withing the individual convolutions to avoid dimensional mismatch.

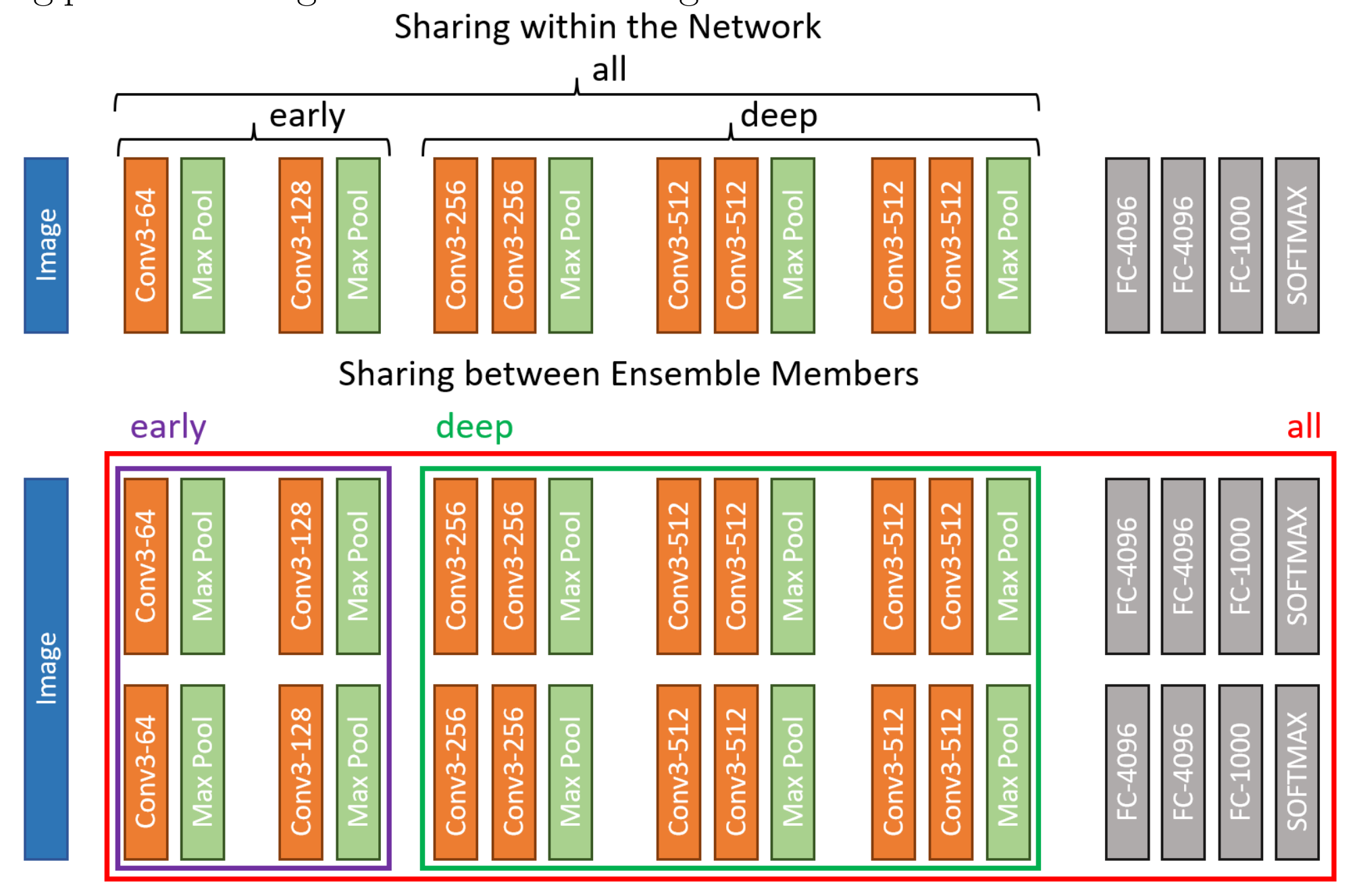

Figure 13: Techniques of Parameter sharing

A third version was also implemented, combining both sharing methods, so sharing weights within the network and between the ensemble members. Results are shown below with 8 ensemble members, alpha and gamma at 0.5 and the ADAM optimizer after 10 epochs.

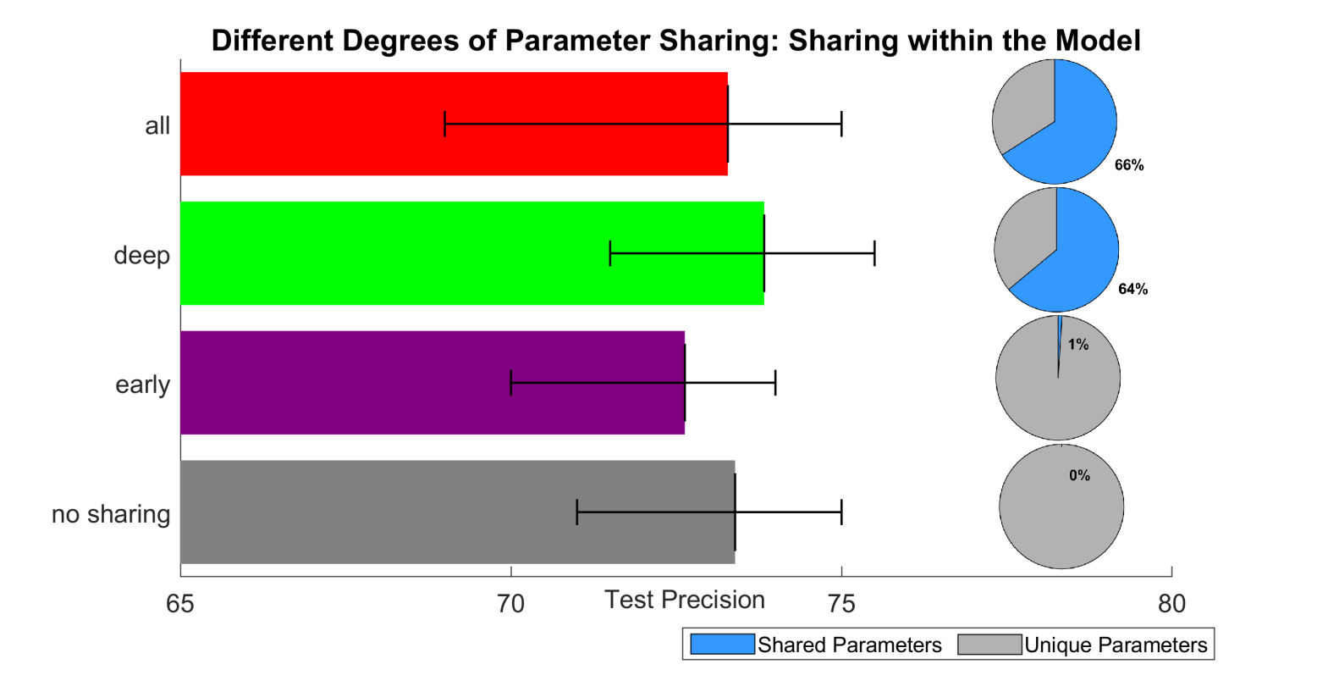

Figure 14: Different Degrees of Parameter Sharing within Model

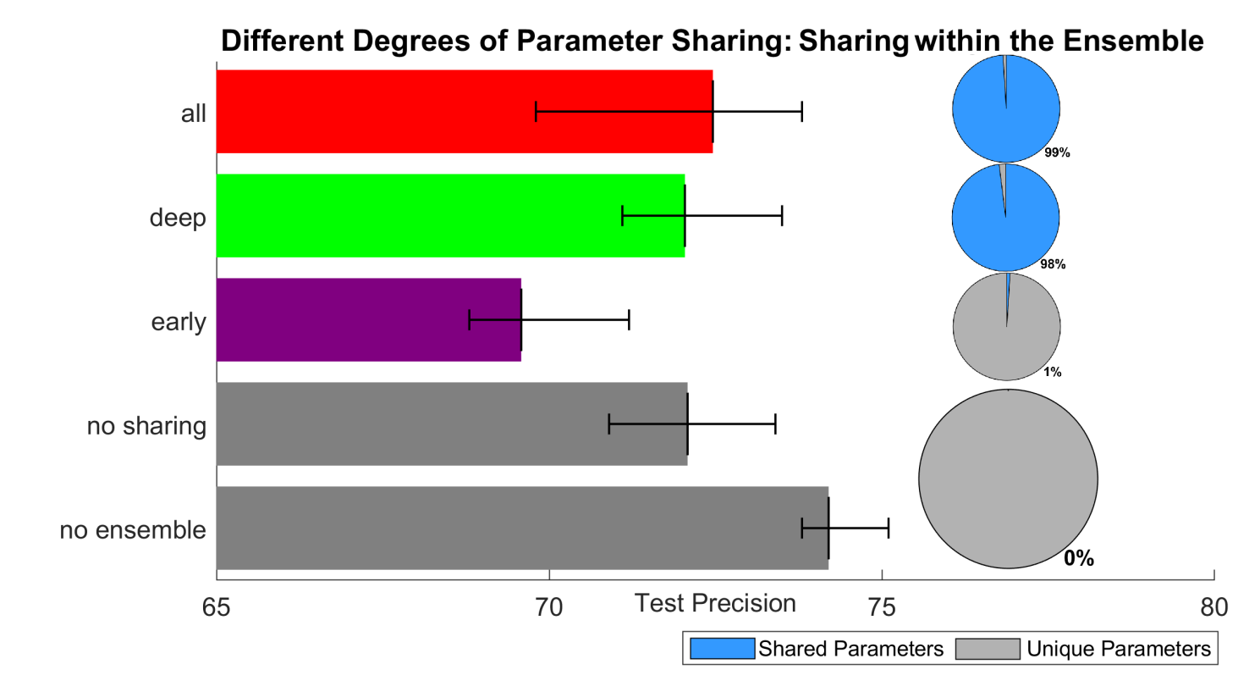

Figure 15: Different Degrees of Parameter Sharing within Ensemble

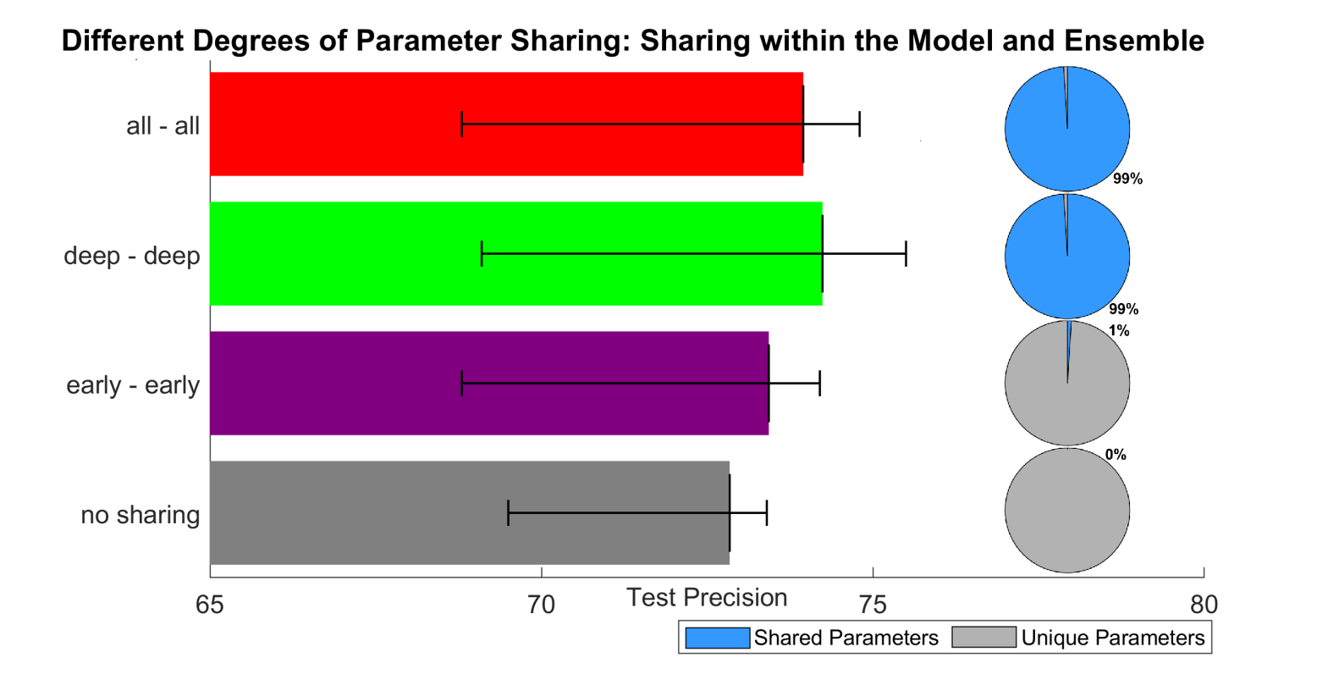

Figure 16: Different Degrees of Parameter Sharing within Model and Ensemble

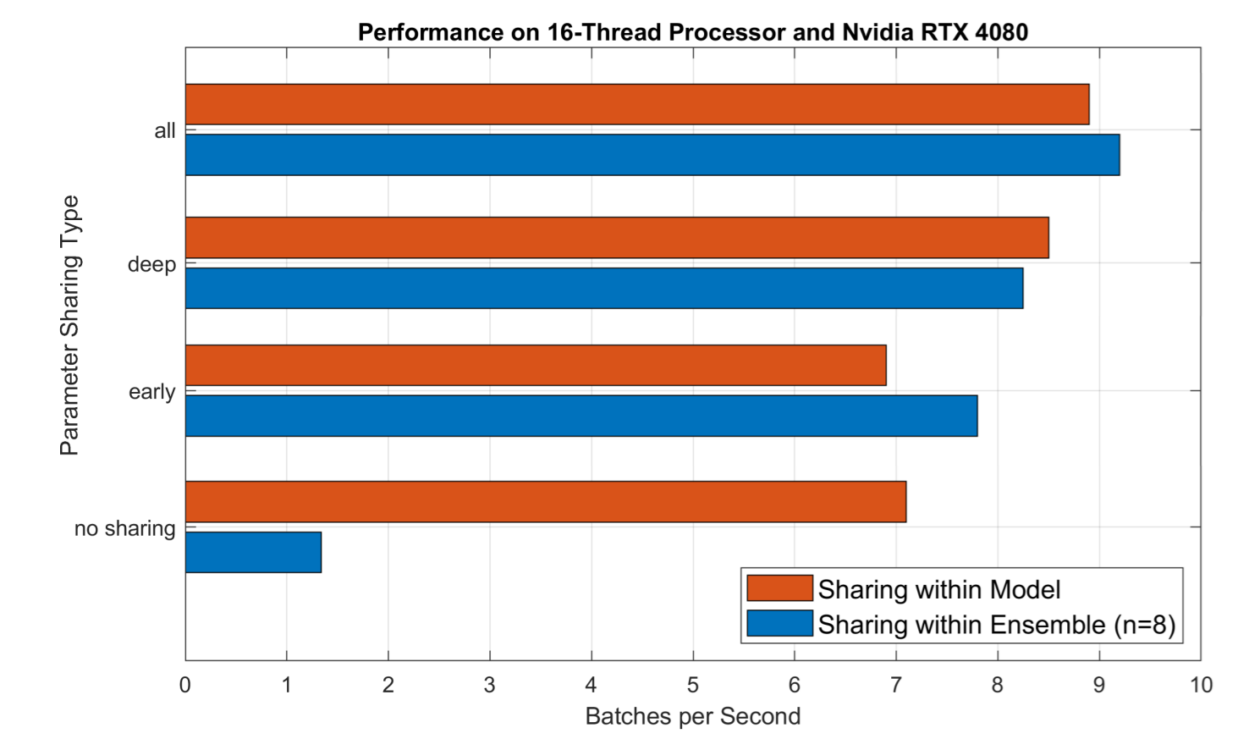

Figure 17: Computed Batches per Second during Training

The experiments were then repeated for sharing within the network and sharing both within the network and the ensemble on the more complex CIFAR-100 dataset, on which the classes were merged to 20 classes to limit complexity. Both cases required gradient clipping on CIFAR-20, which was not necessary in CIFAR-10. Both models performed only on "guessing level" without clipping. The results are shown below.

Sharing within the network			Sharing within the network and the ensemble		
Sharing Intensity	Test Accuracy	Epochs	Sharing Intensity	Test Accuracy	Epochs
No	62.74%	49	No - No	70.62%	13
Early	5.00%	6	Early - Early	68.91%	12
Deep	65.16%	50	Deep - Deep	70.44%	20
All	63.88%	38	All - All	68.22%	13

Table 1: Comparison of sharing strategies within the network and the ensemble

Both sharing techniques showed similar behavior as on CIFAR-10, but the results were much less stable, with some models with early sharing not being trained at all (5 % test accuracy by early stopping with patience of 5).

References

[1] Yeming Wen, Dustin Tran, and Jimmy Ba, "Batchensemble: an alternative approach to efficient ensemble and lifelong learning," *arXiv preprint arXiv:2002.06715*, 2020.

[2] Alex Krizhevsky, "Learning multiple layers of features from tiny images," *https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf*, 2009.

[3] Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, et al., "Variational learning is effective for large deep networks," *arXiv preprint arXiv:2402.17641*, 2024.