WEIGHT DIVERGENCE IN CONVOLUTIONAL NEURAL NETWORKS

Akash Rao
New York University
agr8437@nyu.edu

Nicky Kriplani
New York University
ak9100@nyu.edu

Christopher Musco
New York University
cmusco@nyu.edu

ABSTRACT

In this paper, we investigate the behavior of weights in Convolutional Neural Networks when freeing weight sharing. Specifically, we train Convolutional Neural Networks to an optimum and then relax the weight sharing constraint and train further to observe whether the weights diverge and gauge the quality of solutions with diverged weights. We observed that freed weights diverged by a large amount, but the solutions were universally overfits, meaning that they had worse accuracy and robustness. These results support the hypothesis that weight sharing serves to reduce overfitting in Convolutional Neural Networks. All of our code is available at https://github.com/NickyDCFP/Weight-Sharing-Research-Project.

***K*eywords** Weight Sharing · Convolutional Neural Networks · Deep Learning

1 Introduction

Background. Convolutional neural networks (CNNs) are an incredibly common and effective architecture within deep learning that exhibit strong performance in image recognition, NLP, audio processing, and other tasks. One of the key factors behind their success, especially in the computer vision context, is weight sharing, a technique in which kernels within individual layers of the network are constrained to have the same weight. This practice has been touted to decrease overfitting, reduce training time, and increase translational invariance, a measure of a network's resistance to shifting of inputs. Owing to weight sharing's absence in natural systems, though, its necessity in CNNs has been brought into question[1].

Past Work. Our work borrows heavily from an existing paper [1]; we investigate a similar question in a different fashion. In this original paper, the scientists use a variety of approaches to determine weight sharing's effect on translational invariance and overfitting. Crucially, they introduced a new type of network, the Free Convolutional Network (FCN), which is essentially a CNN without weight sharing constraints, i.e just a fully connected layer with all non-local connections severed. The scientists compared the performance of CNNs and FCNs of the same architecture on the MNIST[2] and CIFAR-10[3] datasets with different levels of translational, rotational, noise, and edge noise augmentation. The scientists also tried severing random connections in the FCNs. They collected training and validation accuracy metrics for the networks at different levels of augmentation and

also measured the Euclidean distance between nearby filters in the FCNs. Overall, the scientists found that, at low levels of translational augmentation, both CNNs and FCNs were observed to overfit the training data, but both fits improved at higher levels of augmentation. Additionally, CNNs had overall higher validation accuracy, especially as translational augmentation increased. These results led the scientists to conclude that proper translational augmentation is more of a preventative factor for overfitting than weight sharing. They also concluded that weight sharing does contribute to translational invariance. Finally, the weights of neighboring filters in the FCNs were observed to converge, especially at higher levels of translational augmentation. This led the scientists to conclude that, with properly augmented data, weight sharing occurs naturally, which stands in line with the conclusion that weight sharing and translational invariance are correlated.

Motivations. We took issue with a few elements of this paper. First, we felt that the Euclidean distance metric did not provide a strong enough argument for the emergence of weight sharing. Consider the formula for the squared Euclidean distance between two filters, $\mathbf{x}$ and $\mathbf{y}$.

$$\begin{aligned}\|\mathbf{x}-\mathbf{y}\|_2^2 &= \|\mathbf{x}\|_2^2 - 2\langle \mathbf{x}, \mathbf{y}\rangle + \|\mathbf{y}\|_2^2 \\ &= \|\mathbf{x}\|_2^2 - 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2\cos(\theta) + \|\mathbf{y}\|_2^2\end{aligned}$$

As you can see, the dominant components in this formula are the respective norms of the filters, $\|\mathbf{x}\|_2$ and $\|\mathbf{y}\|_2$, which means that the Euclidean distance may be more representative of the difference in norms of the filters rather than the difference in their spatial orientations. For that reason, we felt that Euclidean distance on its own may not be convincing enough and wanted an accompanying metric, such as cosine distance. Second, we felt that the FCNs were not necessarily given a fair comparison with the CNNs. The FCNs, which had significantly more trainable parameters than the CNNs, were not given any additional training time to catch up to the CNNs in validation accuracy. The different networks were also initialized in exactly the same way, which we did not feel was a proper usage of the FCNs. Our work focuses on examining the efficacy of weight sharing by observing how models behave when weights are freed after an optimum has already been reached with weight sharing. Specifically,

After achieving convergence, do freed weights diverge?

If so, do diverged weights provide more accurate solutions?

Do diverged weights provide more robust solutions?

2 Methods

Dataset. In the same fashion as the original paper, we focused on computer vision tasks, specifically looking at the CIFAR-10[3] dataset.

Setup. Our tests consisted of using free weights as a fine-tuning mechanism. First, a CNN was trained for 100 epochs (100 was chosen as the epoch count because the authors in [1] were training their models for 200 epochs). Then, an FCN of the same structure as that of the CNN was created. The FCN's weights were initialized using the weights from the trained CNN. Finally, the FCN was trained for a further number of epochs. This setup allowed us to observe the differences in training and validation accuracy when the CNNs and FCNs were forced to converge, starting from an equal playing field. This, in turn, could provide insight into whether neural networks naturally learn weight sharing or if the weights diverge on further training. To this end, we recorded the Euclidean and cosine

distance between nearby filters within the FCN during its fine-tuning to get an idea of how much the weights diverged once unconstrained. We also used early stopping conditioned on when validation reached a plateau and introduced noise augmentation when overfitting was encountered. All of our code is available at `https://github.com/NickyDCFP/Weight-Sharing-Research-Project`.

3 Results

Experiment 0. For our first test, we did not use any augmentation. We simply ran the models, with an upper bound of 500 epochs, until early stopping was triggered. For the training and validation losses of these results, see Figure 1. For the distances between neighboring filters, see Figure 2. The CNNs initially overfit the data, and the FCNs did not seem to improve on the training and validation losses by much. The weights were observed to diverge, but only to a small degree, and they quickly reached a plateau after their initial increase. Because of the observed overfitting, we chose to introduce 30% normal noise augmentation into the data.

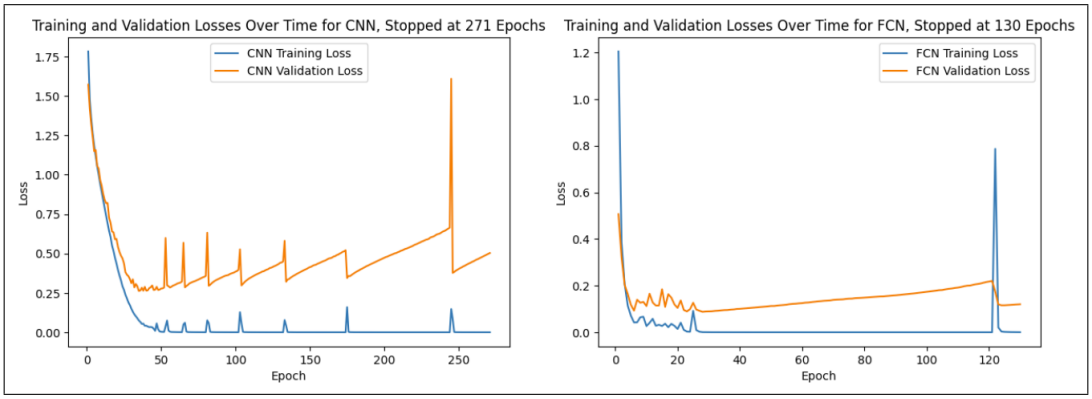

Figure 1: **Training and Validation Losses for CNNs and FCNs from Initial Trial.** This figure shows the training and validation losses without any noise augmentation on the CIFAR-10 dataset. After training the CNN for 271 epochs, early stopping triggered because the validation accuracy hit a plateau. These weights were used to initialize the FCN, which trained for 130 epochs until early stopping triggered again.

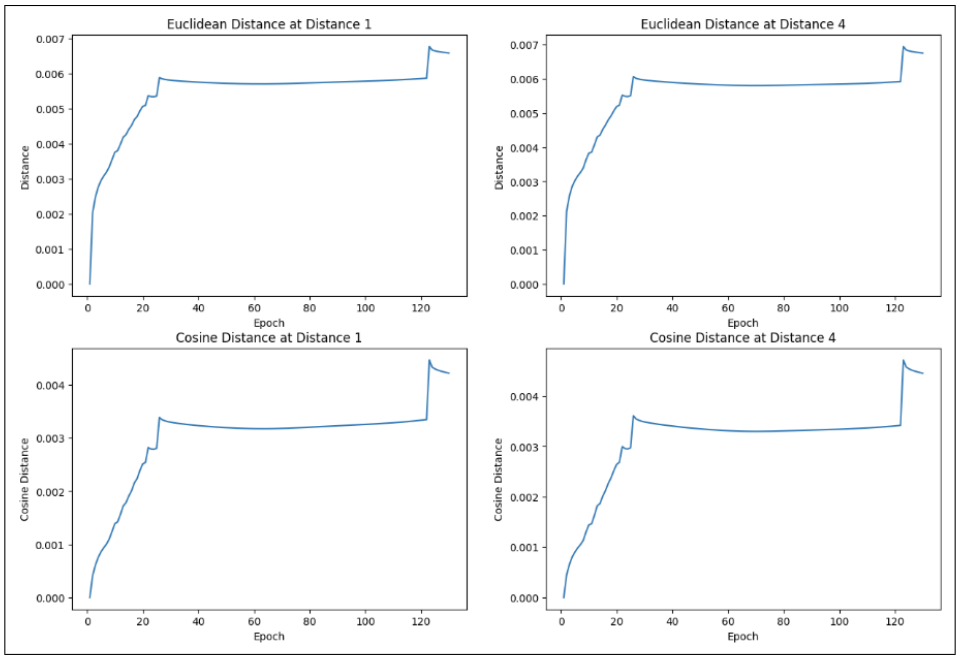

Figure 2: **Euclidean and Cosine Distances Between Neighboring Filters for Initial Trial.** This figure shows the distances between neighboring filters for the trials run in Figure 1.

Experiment 1. Following the results of Experiment 0, we used a 30% noise-augmented CIFAR-10 dataset. The CNN was trained until 100 epochs and then retrained for 100 more epochs. The weights from the 100 epoch CNN were converted to FCN weights and the FCN was trained for a further 200 epochs although early stopping was triggered at around 139 epochs. We still observed heavy overfitting in the FCNs as compared to the CNNs (See Figure 3). We also observed a different curve when it comes to the distances between filters. Figure 4 observes a smooth non-decreasing curve for the euclidean distances and cosine distances between filters at distance 1 and distance 4. This seems to show that the network, when the weight sharing restriction is lifted, tends to move away from weight sharing.

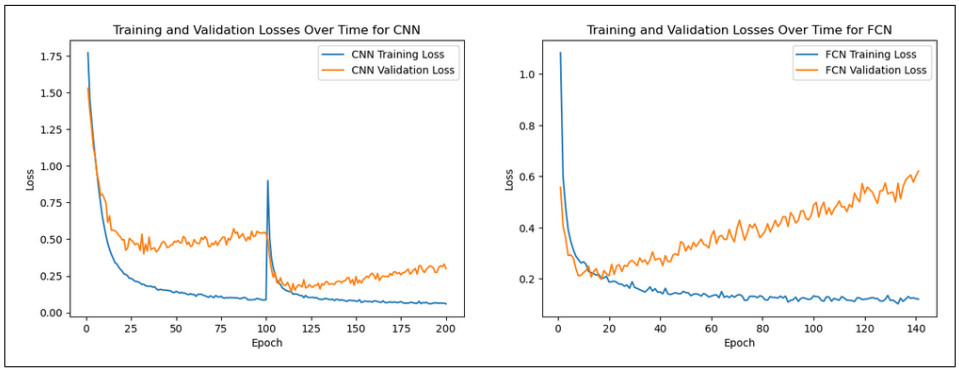

Figure 3: **Training and validation losses for CNN and FCN from Experiment 1.** This figure shows the training and validation losses with 30% noise augmentation on the CIFAR-10 dataset. The CNN weights at 100 epochs were used to initialize the FCN, which trained for 140 epochs until early stopping was triggered. Final CNN validation accuracy achieved: 95.8%. Final FCN validation accuracy achieved: 94%

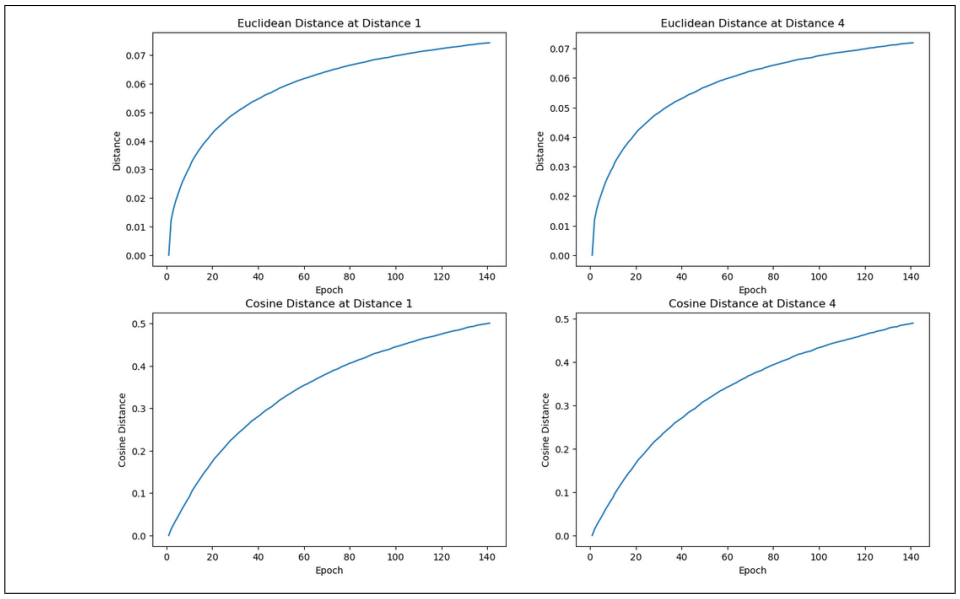

Figure 4: **Euclidean and Cosine Distances Between Neighboring Filters for Experiment 1.** This figure shows the distances between neighboring filters for the trials run in Figure 3.

Experiment 2. We then attempted to run 100 epochs of CNN training and switch to FCN training for a 50% noise augmented version of the CIFAR-10 dataset. This was due to the heavy overfitting observed in the previous experiment. Interestingly, we did not observe much of an improvement in changing the noise augmentation from 30% to 50% (See Figure 5). However the cosine distances were about 0.7 at the end of the training, confirming that we are moving away from the converged structure (See Figure 6)

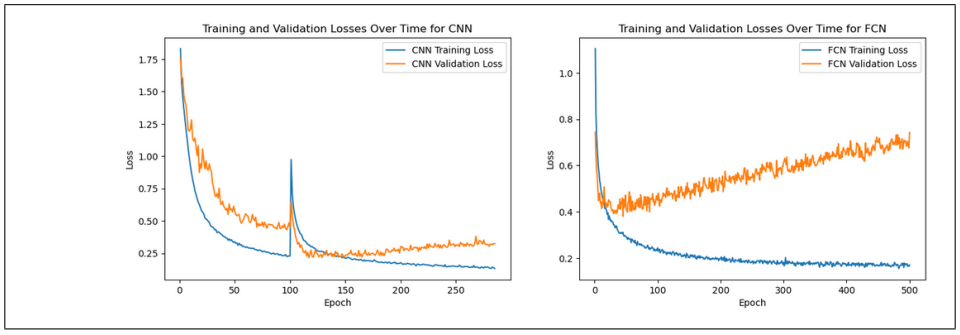

Figure 5: **Training and validation losses for CNN and FCN from Experiment 2.** This figure shows the training and validation losses with 50% noise augmentation on the CIFAR-10 dataset. The CNN weights at 100 epochs were used to initialize the FCN, which trained for 500 more epochs. Early stopping was observed in the CNN at around 270 epochs Final CNN validation accuracy achieved: 94%. Final FCN validation accuracy achieved: 92.6%

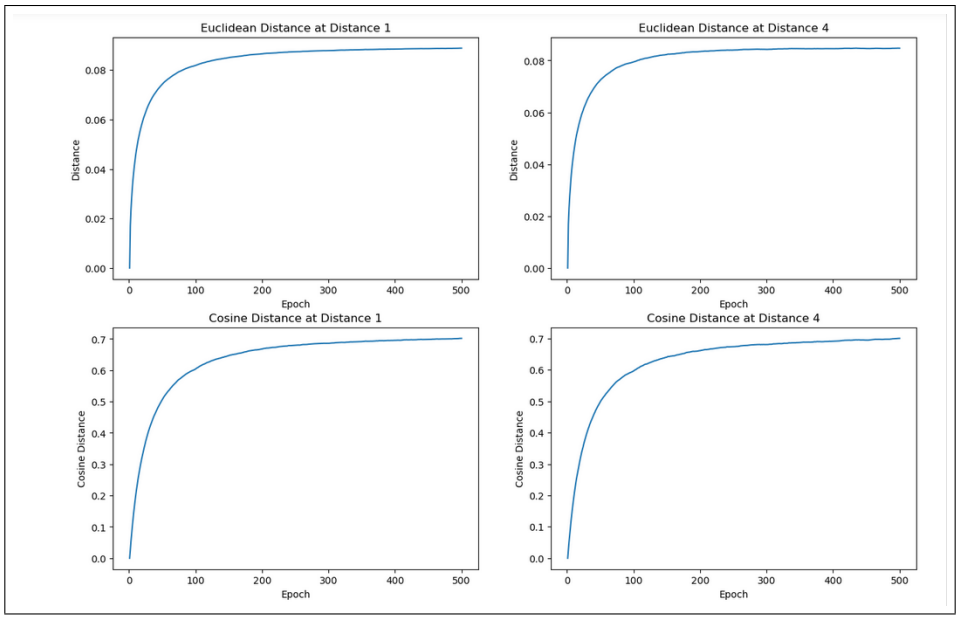

Figure 6: **Euclidean and Cosine Distances Between Neighboring Filters for Experiment 2.** This figure shows the distances between neighboring filters for the trials run in Figure 5.

Experiment 3. Our final experiment tries to address the overfitting issue by reducing the number of epochs trained. We trained a CNN for 50 epochs on a 50% noise augmented CIFAR-10 dataset and then switched to an FCN for a further 50 epochs to see if we observe a similar rapid increase in validation loss. As seen in Figure 7, We observed a trend of small decrease and stabilizing of validation loss in the FCNs, and a steady decrease of training loss. This seems to indicate that possibly the convergence point is very early in the life of the training of these models even at high levels of augmentation.

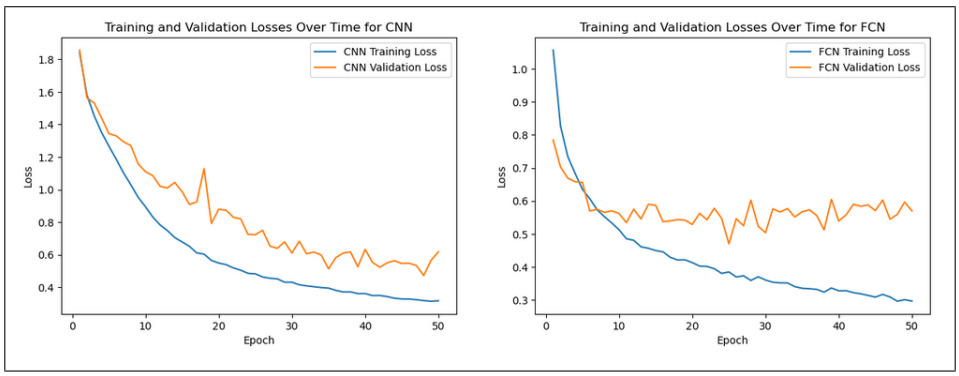

Figure 7: **Training and validation losses for CNN and FCN from Experiment 2.** This figure shows the training and validation losses with 50% noise augmentation on the CIFAR-10 dataset. The CNN weights at 50 epochs were used to initialize the FCN, which trained for 50 more epochs. Final CNN validation accuracy achieved: 83.85%. Final FCN validation accuracy achieved: 85.25%

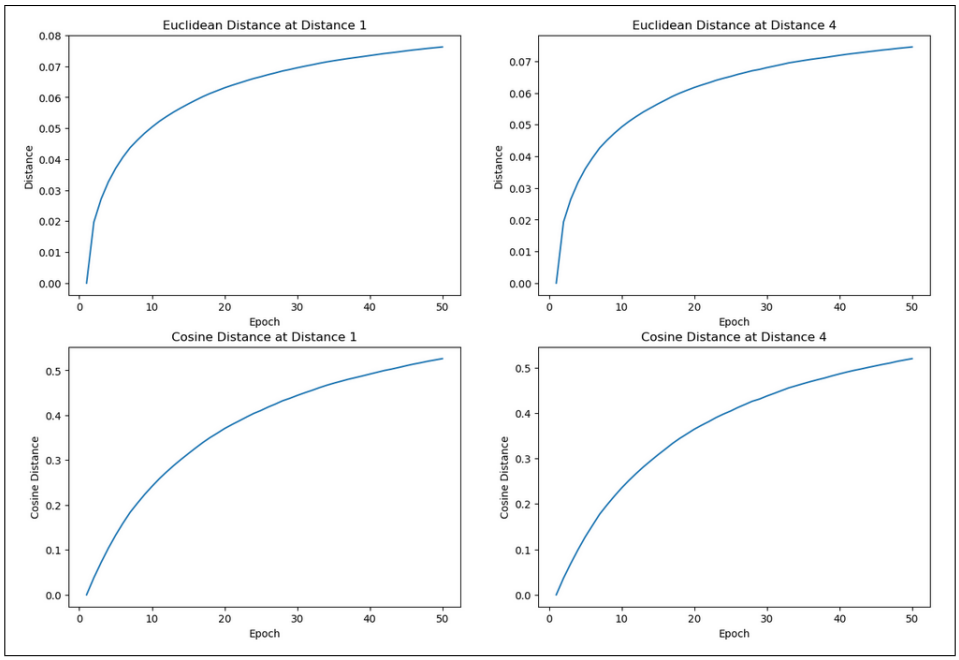

Figure 8: **Euclidean and Cosine Distances Between Neighboring Filters for Experiment 3.** This figure shows the distances between neighboring filters for the trials run in Figure 7.

4 Discussion

Summary. Overall, we found that freed weights do tend to diverge, and quite a bit at that, especially when provided with noisy training data. However, we found that FCNs primarily tended to use their extra parameters to overfit this noise, and that was why their weights would diverge. For this reason, their solutions were both less accurate and less robust than the CNN solutions.

Limitations. It should be noted that we refrained from translationally augmenting our data, which, as was determined in [1], would have severely decreased the overfitting problem. We avoided using translational augmentation because we wanted to remove the possibility that the augmentation would force the weights to remain close in value. Additionally, we did not train on any larger datasets, which would also be more difficult to overfit.

5 Conclusion

Future Work. Potentially the most important next step would be to retry this experiment while forcing the models not to overfit. In the graphs, even some visible overfitting is happening with the CNNs, so it may be useful to train on CIFAR-100[3] or some subset of ImageNet[4], which would make it much more difficult to overfit. Additionally, it could be useful to try translationally augmenting the data and seeing if that reduces the weight divergence, and, if so, by how much.

References

[1] Jordan Ott, Erik Linstead, Nicholas LaHaye, and Pierre Baldi. Learning in the machine: To share or not to share? 2019.

[2] Yann Lecun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. 2012. URL http://yann.lecun.com/exdb/mnist/.

[3] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. 2010. URL https://www.cs.toronto.edu/~kriz/cifar.html.

[4] Li Fei-Fei, Jia Deng, Olga Russakovsky, Alex Berg, and Kai Li. Imagenet. 2021. URL https://www.image-net.org/index.php.