DSC180A Quarter 1 Project: Neural Network Feature Development

Keyu Long kelong@ucsd.edu

Grace Lam pklam@ucsd.edu

Tiffany Yu z5yu@ucsd.edu

Daniel Shi Dshi@ucsd.edu

Licheng Hu l2hu@ucsd.edu

Yian Ma (Mentor) yim013@ucsd.edu

Misha Belkin (Mentor) mbelkin@ucsd.edu

Abstract

The evolution of features during the training process in neural networks represents a critical area of investigation in modern machine learning. Our study aims to focus on the evolution of the features of image data during the training process in convolutional networks by experimenting with training a ResNet-18 and LeNet on the MNIST data. Investigating this evolution involves how features were selected by each filter and become more relevant in representing the underlying patterns within the data. Understanding the trajectory of feature evolution within neural network could offer the insights into its learning mechanisms and unlocking its potential of better performance in more complex tasks.

Code: https://github.com/KULcoder/DSC180A-Project1

1 Introduction

Neural networks are machine learning models that are designed to emulate the human brain's structure and functionality to recognize patterns. Over the years, neural networks have achieved great success and showed extraordinary capabilities of tackling complex problems and extracting meaningful information from data. As the performance of neural networks continuously overtakes traditional machine learning algorithms, it is essential to have a deeper understanding of the intricate mechanisms during their learning processes. The previous studys Beaglehole et al. (2023) and Radhakrishnan et al. (2023) formulated Convolutional Neural Feature Ansatz which the features selected by convolutional networks can be recovered by computing the average gradient outer product of the trained network with respect to image patches given by empirical covariance matrices of filters at any given layer. In this paper, we are going to investigate the learning mechanisms of convolutional neural networks on images from the evolution of its features during the training process start from Convolutional Neural Feature Ansatz.

2 Methods

2.1 Model and Dataset

In this project, we build a 5-layer ResNet-18 and a 5-layer LeNet trained on MNIST (Modified National Institute of Standards and Technology database) dataset. Theses two models was trained over 30 and 10 epochs with learning rate 0.0001 and used Adam for optimization and performed well which the ResNet-18 reached a training accuracy of 98% and a validation accuracy of 98% and the LeNet-5 reached a training accuracy of 95% and a validation accuracy of 95%.

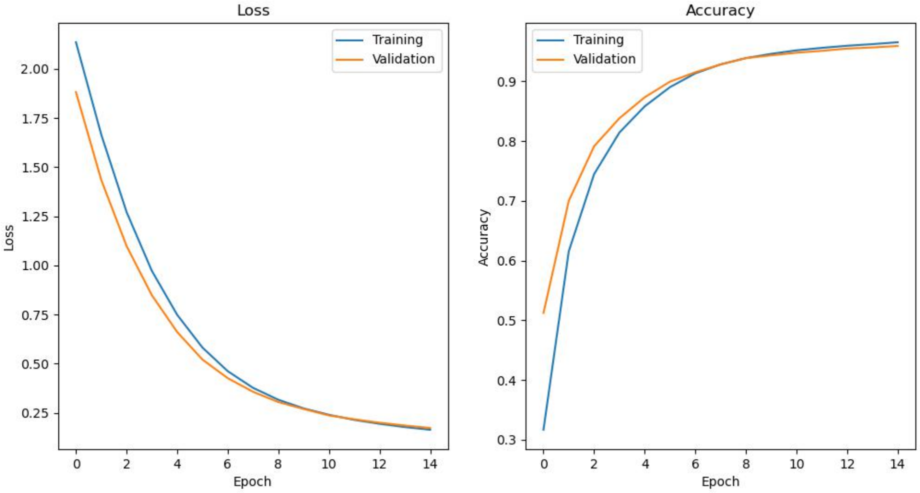

Figure 1: Train and validation loss and accuracy on ResNet-18 with MNIST

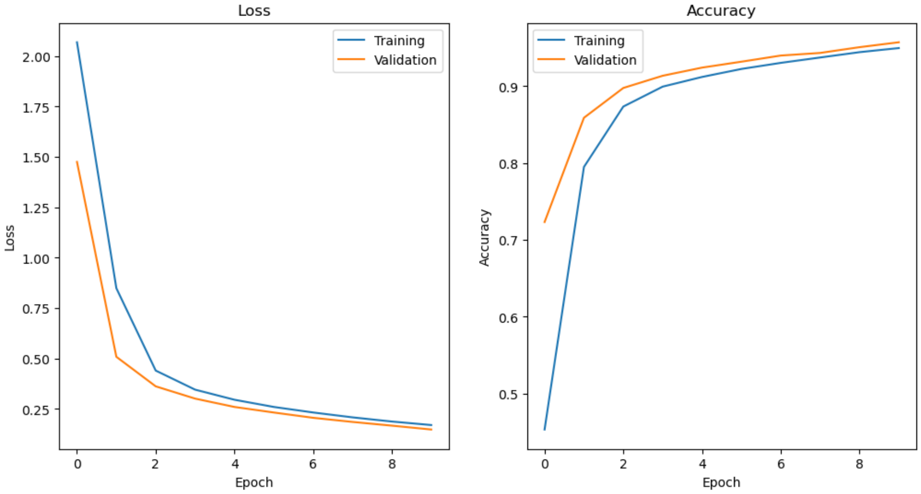

Figure 2: Train and validation loss and accuracy on LeNet-5 with MNIST

2.2 Experiments

After finish training models, we investigated the extracted features process of the network. This analysis is pivotal in getting the insights of the learning process and understanding the how the model understand how model's representation evolved over time.

Our investigation began with inputting a random image from the MNIST dataset to see how the the layer output change in the training process. As we feed the random image of digits to the model, we record the output of each layer. It allow us to better understand the transformations and feature representations when inputs passed through each layer of network through visualization. As we seen in figure 3 and figure 4, different filters in LeNet extract information differently for each layer and each layer also capture different patterns for the input image.

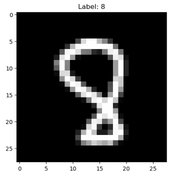

Figure 3: Random image of eight from MNIST

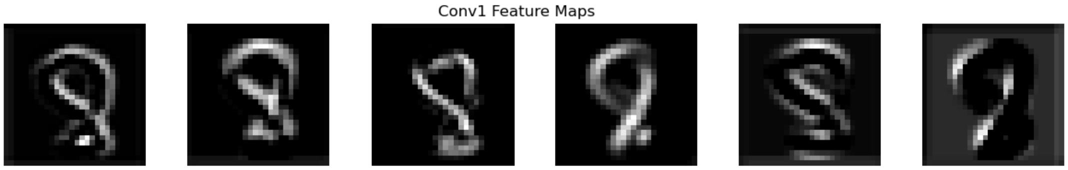

Figure 4: Convolutional output in the first layer for the image 8 for LeNet

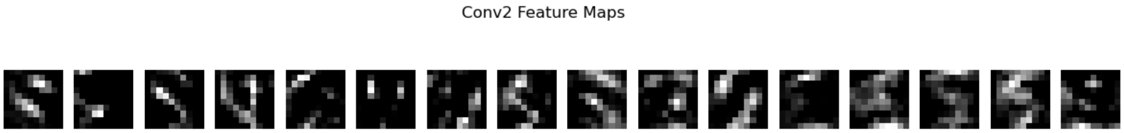

Figure 5: Convolutional output in the second layer for the image 8 for LeNet

Next, we investigated the filter weights by extracting the first convolutional filters in each epoch to see how the weights change through training. This filter typically capture low-level features like edges or simple shape. Extracting and observing these filters in each epoch allowed us to track the evolution of weights and understand how they adapt to training data. As performing the first experiment, we identify that first layer of convolutional networks captures most of the features. Therefore, we focused on investigating the weights evolution of convolutional filters in the first layer. The convolutional filter in first layer has size $64 \times 1 \times 3 \times 3$ and we reshaped it to 64×9. The dot product $W^T W$ is calculated to visualize the Neural Feature Matrix. We use the Neural Feature Matrix to observe the representation of feature extracted by the corresponding layer. As seen in figure 6, the feature becomes more significant along the training process, we can see the it changed from the initial grey map to a map where some features showing up after a few epochs of training.

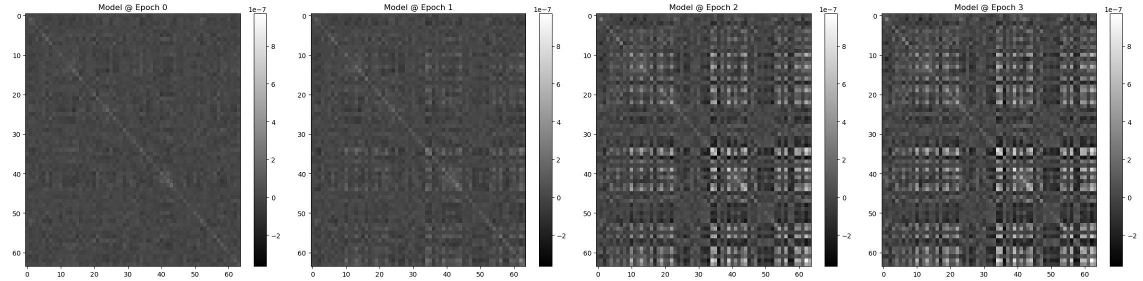

Figure 6: Feature Matrix for ResNet-18 in the first 3 epochs

Table 1 presents how the feature matrix is related to the performance of the model.

Table 1: ResNet-18 Performance in the first 3 epochs

epoch	train accuracy	validation accuracy
1	17.947	10.28
2	61.179	82.58
3	90.162	94.29

3 Conclusion

Our exploration into the feature evolution within convolutional neural networks has provided valuable insights into the internal learning mechanisms of these powerful models. Through the training and analysis of a ResNet-18 and a LeNet-5 on the MNIST dataset, we have observed the progression of feature development across training epochs and layers. Our experiments have shown that these models, though starting from a non-informative state, gradually refine their weights to enhance the discriminative power of their learned representations. In particular, the visualization of the Neural Feature Matrix of the weights in the first convolutional layer has allowed us to observe the development of feature representations, evolving from undifferentiated grey maps to structured patterns that capture the essential features of the input data. These findings implies the importance of visual interpretability in understanding the learning mechanism of neural network models. Beyond this project, we could extend the result in unlocking its potential of better performance in more complex tasks in the future.

References

Beaglehole, Daniel, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. 2023. “Mechanism of feature learning in convolutional neural networks.”

Radhakrishnan, Adityanarayanan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. 2023. “Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features.”