Brock University
Faculty of Mathematics & Science
Department of Computer Science

COSC 4P80 Final Project

Prepared By:
Brett Terpstra #6920201
bt19ex@brocku.ca

Steve Mastrokalos #7276900
sm21ks@brocku.ca

Instructor:
Dave Bockus

January, 2025

Abstract

Deep learning integrates feature extraction directly into the traditional neural network architecture, improving overall performance. This paper explores the benefits of deep learning by using the MNIST handwritten digit dataset. We compare two different network configurations: one with feature extraction and feed-forward classification, and the other with only the feed-forward classification. Our results demonstrate that as the size of the feed-forward network is reduced, its classification performance decreases. However, by leveraging feature extraction, we are able to retain classification power, showing the value of deep learning in improving performance with smaller networks.

Table of Contents

Chapter 1

Introduction

As previously mentioned, deep learning combines feature extraction through convolution and pooling with traditional neural networks, eliminating the need for humans to manually extract features from datasets. Convolution, in essence, is a filtering process where trained filter(s) slides over the input data to extract features and other useful information. Pooling is the subsequent process of taking local samples and selecting either the minimum, maximum, or average of those samples. This step helps identify feature locations and condenses the information produced by the convolution layer.

A typical deep learning pipeline consists of several convolution and pooling layers, followed by a few fully connected layers. In this work, we aim to demonstrate that using a deep learning network configuration can reduce the size of the feed-forward section without compromising program classification performance, thereby highlighting the effectiveness of deep learning.

The MNIST database is a standard benchmark for image-processing neural networks. For our comparison, we will use a modified version of the DLIB deep learning example. This approach allows us to showcase the differences between standard feed-forward neural networks and deep learning networks without requiring expensive GPUs or AI accelerators. While the MNIST dataset is solvable using only feed-forward neural networks, we intend to demonstrate that feature extraction with deep learning can achieve better prediction accuracy, even with smaller classification networks.

Chapter 2

Experimental Setup

The MNIST database comprises grayscale images of size 28x28 pixels, with 60,000 training images and 10,000 test images. For each experiment, we present graphs comparing the average error per epoch for both configurations, alongside a table summarizing the test results of the final network after training. Due to resource constraints, training is limited to a maximum of 100 epochs, and the experiments are averaged over ten runs. Notably, the deep learning configuration requires approximately six hours to complete on a 32-thread workstation.

Our experiments are divided into two phases, each testing a deep learning network alongside its corresponding feed-forward network. To ensure a fair comparison, the feed-forward test focuses exclusively on the feed-forward component of the deep learning network. This approach ensures consistency in variables such as the number of layers or nodes in the feed-forward section, minimizing potential biases and preserving the integrity of the results.

2.1 Experiment 1

The first experiment compares the performance of a deep learning configuration to a baseline feed-forward network, using the example provided in the DLIB C++ library. The deep learning configuration consists of two ReLU-activated convolutional layers, each followed by max-pooling and then a fully connected feed-forward network.

The first convolutional layer uses six filters, each sized 5x5, with a stride of 1x1. The second convolutional layer applies sixteen filters with the same size and stride configuration. After each convolutional operation, the output is passed through a max-pooling layer with a filter size of 2x2 and a stride of 2x2, which reduces the spatial dimensions.

The pooled features are then fed into a three-layer fully connected ReLU-activated feed-forward network. The fully connected layers consist of 120 neurons in the first layer, 84 neurons in the second, and 10 neurons in the final output layer, with each output representing a predicted class for the input image.

For comparison, the baseline configuration omits the convolutional and pooling layers and consists solely of the three-layer feed-forward network. This setup isolates the feed-forward network's performance, enabling a direct comparison with the deep learning configuration.

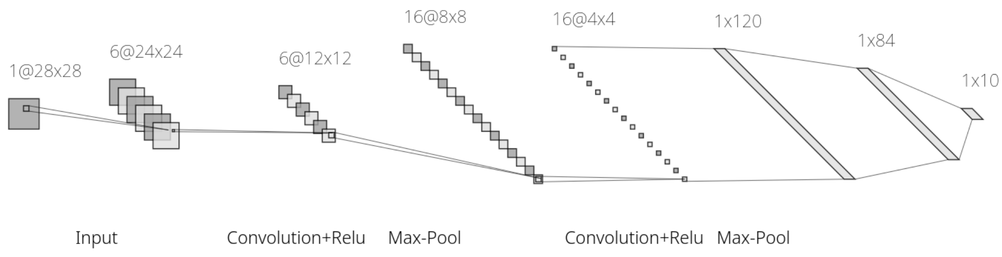

Figure 2.1: Experiment 1 network configuration.

2.2 Experiment 2

The second experiment retains the same convolutional and pooling configurations as in Experiment 1 but modifies the number of neurons in the feed-forward section of the network. This adjustment reduces the number of parameters available for object detection and classification, allowing for an evaluation of how deep learning's hierarchical feature extraction compensates for reduced network capacity.

By demonstrating the impact of parameter constraints, this experiment highlights the advantage of feature extraction in the deep learning configuration, particularly when resources in the feed-forward section are limited. This serves to underscore the practical utility of convolutional layers in achieving robust classification performance with smaller, more efficient networks.

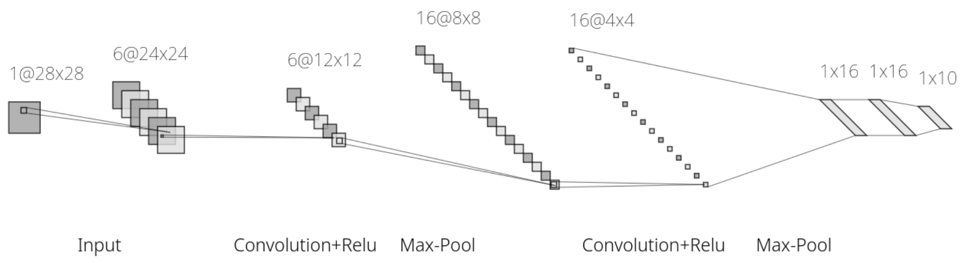

Figure 2.2: Experiment 2 network configuration.

Chapter 3

Results

3.1 Experiment 1

Our testing results, presented in Table 3.1, demonstrate that both classical feed-forward networks and deep learning configurations are capable of achieving near-perfect classification on the testing dataset with a reasonably sized neural network. The deep learning network achieved an exceptional performance, converging in only 20 epochs with a final classification accuracy of 99%. In comparison, the feed-forward network reached an accuracy of 98% but required nearly all the allotted epochs to reach this level of performance. While it is possible that the feed-forward network could match the deep learning network's performance with additional training epochs, resource constraints precluded this, and such an extension was unnecessary for the purposes of Experiment 1.

These findings suggest the presence of an underlying pattern within the Arabic numeral system that can be effectively learned by both approaches. Furthermore, as illustrated in Figure 3.1, the deep learning configuration demonstrated substantially faster convergence, underscoring the efficiency of convolutional layers in feature extraction and their role in expediting convergence in the training process.

It is important to note that throughout this experiment, the deep learning component accounted for the majority of the runtime. This is acceptable, as our primary goal is to demonstrate the benefit of feature extraction on classification performance, rather than to assess computational efficiency.

While deep learning may be outclassed for a relatively simple problem like this, it becomes a more viable option for more complex tasks. In such cases, feature extraction could offer a significant advantage, as the large size of a feed-forward-only network may become impractical.

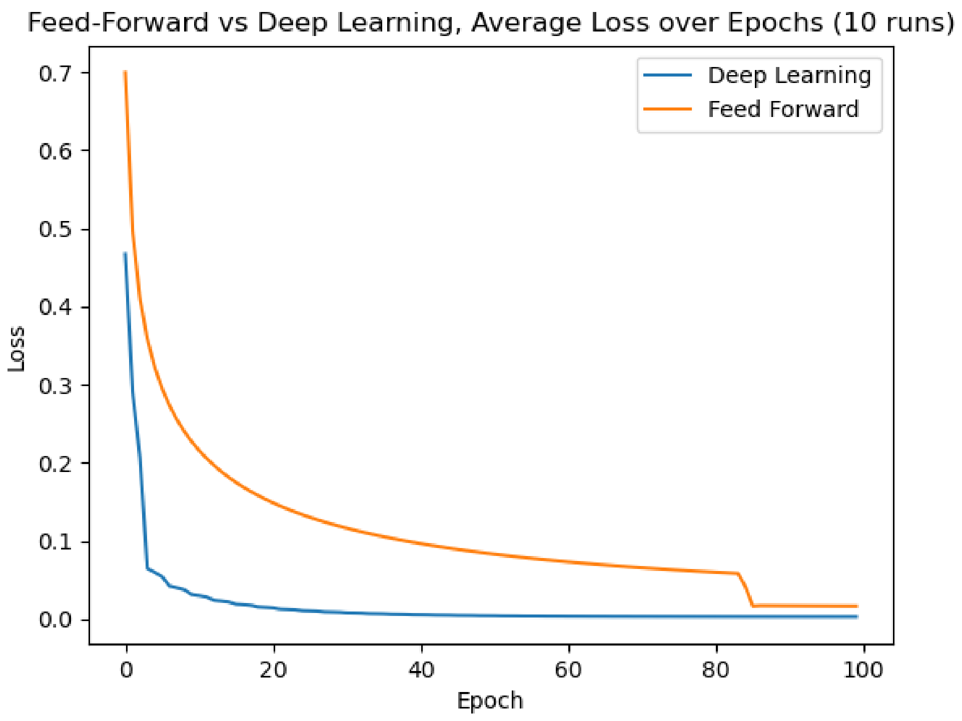

Figure 3.1: Experiment 1 Feed-Forward vs Deep Learning, average loss while training.

Test	Correct	Incorrect	Accuracy (%)
Feed-Forward	9800	199	98
Deep Learning	9898	101	99

Table 3.1: Experiment 1 results on testing dataset. (Averaged over 10 runs)

3.2 Experiment 2

In Experiment 2, the benefits of deep learning are evident, as shown in Table 3.2. Despite the significantly reduced size of the classification network, the deep learning configuration maintains the 99% accuracy achieved in Experiment 1. In contrast, the feed-forward network's performance declines, now achieving only 96% accuracy. Notably, the deep learning network still converges within 20 epochs, similar to Experiment 1. While the feed-forward network roughly converges within 40 epochs and continues to improve throughout the remainder of the run, it never reaches the peak classification accuracy observed in Experiment 1.

It could certainly be argued that the reduced size of the feed-forward network is the primary cause of the observed decrease in performance. However, this only serves to highlight the value of feature extraction. Feature extraction allows for the provision of more information-dense data to the classification network, and, as clearly demonstrated in this experiment, it significantly enhances overall performance. While allowing the feed-forward network more training time might help improve its results, the benefit of feature extraction in the deep learning configuration remains evident.

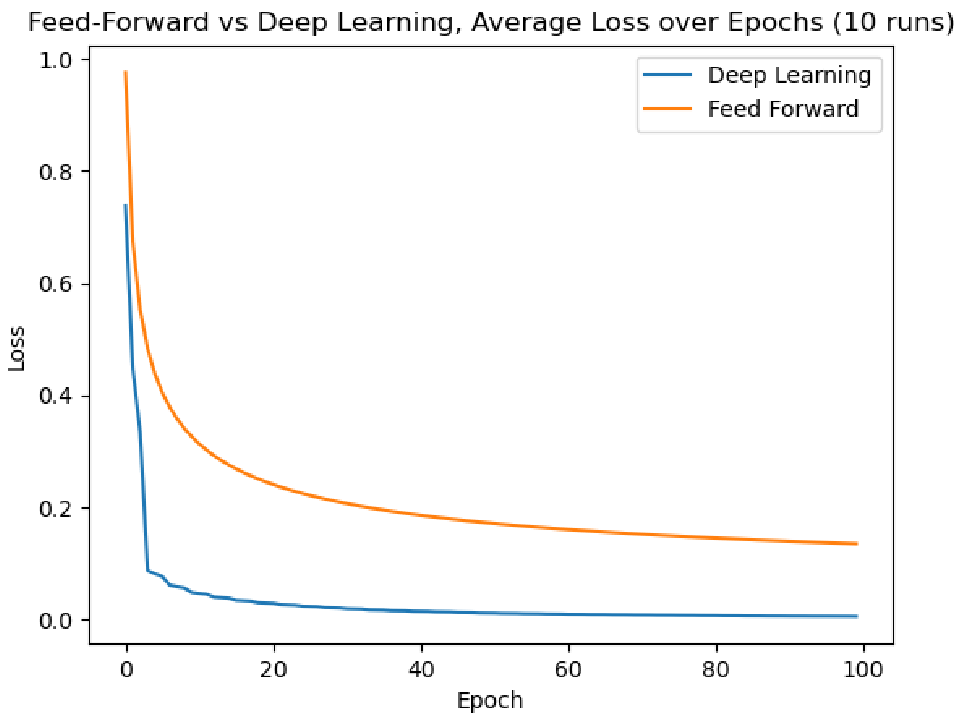

Figure 3.2: Experiment 2 Feed-Forward vs Deep Learning, average loss while training.

Test	Correct	Incorrect	Accuracy (%)
Feed-Forward	9588	411	96
Deep Learning	9887	112	99

Table 3.2: Experiment 2 results on testing dataset. (Averaged over 10 runs)

Chapter 4

Conclusion

We have demonstrated that feature extraction through deep learning is a powerful tool for enhancing neural network performance. This is particularly evident when addressing larger-scale problems, such as the ImageNet1000 dataset, which contains one thousand different classification labels. Instead of training a network to process entire images across all labels, feature extraction allows for a more efficient approach, where the network is trained on extracted features rather than raw data. With a sufficiently large and reasonable deep learning network, problems like ImageNet can be effectively tackled, where traditional feed-forward networks would likely struggle.

However, this work has clear limitations. We only conducted two experiments and did not control for the number of parameters between the feed-forward network and the feature extraction step. Future work could address this by testing with feed-forward networks that have larger and more layers, thus compensating for the parameter discrepancy between the two configurations. Despite these limitations, we believe this paper has successfully shown that feature extraction in deep learning is a valuable and powerful tool.

Bibliography

[1] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.