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 folds feature extraction into the traditional neural network architecture. This paper will use the MNIST handwritten digit dataset to show the benefit in using deep learning techniques.

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 feed-forward neural networks, we intend to demonstrate that deep learning can achieve better classification performance, even on smaller networks.

Chapter 2

Experimental Setup

The MNIST database consists of gray-scale images sized 28x28 with 60k training images and 10k test images. For each experiment we will present graphs of the average error per epoch compared between both configurations alongside a table of test results on the final network after training. Due to resource constraints training is limited to at most 100 epochs and our experiments are averaged over ten runs (the deep learning configuration takes 6 hours to run on my 32 thread workstation).

Our experiments are divided into two parts, each testing a deep learning network alongside its corresponding feed-forward network. For a fair comparison, the feed-forward test focuses explicitly on the feed-forward component of the deep learning network. This ensures that variables such as the number of layers or nodes in the feed-forward section remain consistent, minimizing potential biases and maintaining the integrity of our comparisons.

2.1 Experiment 1

Our first experiment compares using the included example from the DLIB C++ library. Specifically the deep learning test consists of two ReLu convolutions, the first of which has six filters sized 5x5 with a 1x1 stride. The second has sixteen filters with the same size and stride configuration. Each convolution step is passed through a max-pooling stage set to a size of 2x2 with a stride of 2x2. The results are then passed into a three layer fully connected ReLu feed-forward network. The first layer has 120 neurons, second has 84 and the final output layer has 10 neurons, representing the class the network thinks the input image is. The other configuration in this experiment consists of only the three layer fully connected network.

2.2 Experiment 2

The second experiment keeps the same settings on the convolutions and pooling but changes the number of neurons in the layers of the feed forward section of the network. This is meant to demonstrate that as you restrict the number of parameters used for object detection and classification that the feature extractions provided by deep learning are highly beneficial.

Chapter 3

Results

mention paramter counts being unequal and that provides more options for storing information mention that parameter counts being the same doesn't account for differences in "power" of functions.