Jazmine Brown
L04 CNN
ITAI 2376
Patricia McManus

CNN Lab 04

Introduction

For this assignment, we built a Convolutional Neural Network (CNN) to classify handwritten digits from the famous MNIST dataset. This dataset is a classic in the field of computer vision and provides a great starting point for understanding image classification with deep learning.

The purpose of this assignment is to provide a hands-on experience with building and training a CNN model for image classification. The goal of this assignment is to gain practical insights into the workflow of a building a CNN that will include data preparation, model building, training and evaluation. This assignment also allows me to experiment with different model architectures and hyperparameters to observe their effects on the model's performance.

What new concepts about CNNs did you discover during this lab?

In section 2 of the lab, I learned about defining and building Convolutional Neural Networks (CNNs) using Keras layers. The keras.Input(shape=28,28,1)) layer specifies the shape of the input images as 28x28 pixels with 1 grayscale channel, essentially telling the computer to expect small black and white pictures. The layers.Conv2D and layers.MaxPooling2d layers act like magnifying glasses detecting features in the input images and reducing their spatial dimensions, respectively. The layers.Flatten layer transforms the 2D feature maps into a 1D vector to connect to fully connected layers and the layers.Dropout layer helps prevent overfitting by randomly setting 50% of the input unit to zero during training. These tools work together to help the computer understand the images and confidently classify them as one of the digits (0-9) by providing a probability for each digit.

How does this lab connect to your previous knowledge of neural networks?

Prior to the lab I had some basic previous knowledge of TensorFlow and how it is used to provide comprehensive tools and libraries to build, train and deploy machine learning models. By learning more about Keras, I now understand how the two are related since Keras runs on top of TensorFlow and simplifies the process of building and training neural networks by providing easy to use functions and classes. Keras acts as an interface that allows you to create models quickly and efficiently without getting bogged down in the low-level details. TensorFlow, in the background, handles the heavy lifting and computational tasks required to train and optimize those models. It is almost like driving a car, Keras is the steering wheel and controls how you interact with, while TensorFlow is the engine that powers the vehicle.

What surprised you about working with CNNs and the MNIST dataset?

It was surprising to see how CNNS automatically learn to detect important features in images like edges and shapes without explicitly being programmed to do so. The layers in CNNs, like convolutional and pooling layers work together to create increasingly complex feature maps that make this possible. It was also surprising to see how CNNs demonstrated the ability to accurately classify the digits.

Challenges and Growth:

What specific challenges did you encounter while implementing the CNN?

A challenge I encountered when attempting section 4 of the lab when it asked the student to fill in the blanks for loss function and optimizer. The blanks were already filled with categorical cross entropy as the loss function and adam as the optimizer. I originally wanted to try something different than what was provided so I asked “copilot” for an alternative option and it recommended sparse categorical cross entropy as the loss function and SGD as the optimizer. When I ran cell 5 it did not execute and gave me an error message “The error message "ValueError: Argument output must have rank (ndim) target.ndim - 1. Received: target.shape=(None, 10), output.shape=(None, 10)" indicates a mismatch between the shape of your target variable (y_train) and the expected shape for the sparse_categorical_crossentropy loss function.”

How did you overcome these challenges? What resources or strategies helped you understand difficult concepts?

I overcame the challenge by using Gemini that is embedded in Colab to help explain the error further and it suggested me to change the loss function back to categorical_crossentropy and adam as the optimizer. I was able to successfully train the model and gain a better understanding of loss functions and optimizers.

Personal Development:

How has this lab changed your understanding of deep learning?

By working through the process of building, training, and evaluating a CNN for the MNIST dataset, I gained hands-on experience with the practical aspects of model construction and optimization. I now appreciate the importance of choosing the right loss function and optimizer to effectively train the model and improve its performance.

What aspects of CNNs would you like to explore further?

One aspect of CNNs I'd like to explore further is the process of choosing the most suitable loss function and optimizer for different types of classification tasks. Understanding how different loss functions and optimizers impact the training and performance of the model can provide valuable insights into building more efficient and accurate models. Additionally, experimenting with various CNN architectures and techniques to manage overfitting is something I'm keen to delve deeper into.

If you were already familiar with CNNs, what new perspectives did you gain?

Although I was already familiar with CNNs and TensorFlow, this lab provided new perspectives on the intricacies of model optimization. I learned how different combinations of layers and techniques, such as dropout and data augmentation, can enhance the model's ability to generalize to new data. The detailed exploration of Keras layers and their practical applications in defining and refining CNN models was particularly enlightening.

In conclusion, this assignment provided a valuable hands-on experience in building and training a Convolutional Neural Network (CNN) for image classification using the MNIST dataset. Through this process, I gained practical insights into the workflow of a CNN, including data preparation, model building, training, and evaluation. I discovered new concepts about CNNs, such as the importance of choosing the right loss function and optimizer, and the automatic feature detection capabilities of CNN layers. Despite encountering challenges with alternative loss functions and optimizers, I was able to overcome them with the help of Gemini and successfully train the model. This assignment has deepened my understanding of deep learning and inspired me to further explore the intricacies of CNN architectures and optimization techniques. Overall, this experience has been both enlightening and rewarding, reinforcing the importance of practical experimentation in mastering complex machine learning concepts.