Assignment Report part-1

Karan Nijhawan

January 8, 2025

Introduction

This report outlines the development and evaluation of a Convolutional Neural Network (CNN) model for a regression task. The dataset consists of three parts, each loaded, preprocessed, and combined for training, validation, and testing. The goal is to predict continuous outputs using image data as input.

Dataset Loading and Preprocessing

The dataset files were stored in NumPy format (`data{i}.npy` for features and `lab{i}.npy` for labels). The following steps were undertaken:

- Loaded individual datasets using the `load_data` function.
- Combined the datasets using `np.concatenate`.
- Expanded the data dimensions to include a channel dimension (grayscale images) using `np.expand_dims`.
- Normalized pixel values to the range [0, 1] by dividing by 255.0.

The combined data was split into training, validation, and test sets using `train_test_split` with a test size of 30% and further splitting the validation set into validation and test sets (33% of the validation set).

Model Architecture

The CNN model, implemented using TensorFlow's Keras API, consists of the following layers:

1. **Convolutional Layer:** 32 filters of size (3, 3) with ReLU activation.
2. **MaxPooling Layer:** Pool size of (2, 2).
3. **Convolutional Layer:** 64 filters of size (3, 3) with ReLU activation.
4. **MaxPooling Layer:** Pool size of (2, 2).

5. **Flatten Layer:** Converts 2D feature maps into a 1D vector.
6. **Dense Layer:** 128 units with ReLU activation.
7. **Dropout Layer:** Dropout rate of 0.5 to reduce overfitting.
8. **Output Layer:** Single neuron with linear activation for regression.

Model Training

The model was compiled with the following configurations:

- **Optimizer:** Adam with a learning rate of 0.001.
- **Loss Function:** Mean Squared Error (MSE).
- **Metric:** Mean Squared Error.

Training was performed for 10 epochs with a batch size of 32. Validation data was used to monitor the performance during training.

Evaluation and Results

Validation Results

The validation loss after training is:

- **Validation Loss:** `12.09`

Learning Curves

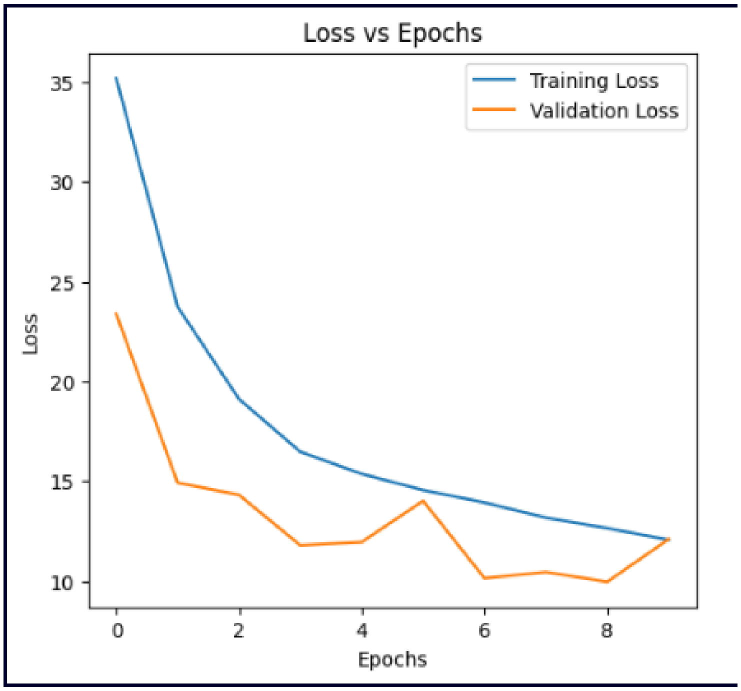

Figure 1: Training and Validation Loss over Epochs

Test Set Evaluation

The model was evaluated on the test set:

- Predictions were rounded to the nearest integer.
- **Test Accuracy:** `10.67 %`