ASSIGNMENT # 3

Names: Saad Mahmood(Fa22-bcs-015)

Jawad Iqbal(Fa22-bcs-008)

Course: Machine Learning

Submitted to: Dr. Saleem Khan

Project Report

Handwritten Digit Recognition (MNIST Dataset)

Introduction:

Handwritten digit recognition is a classic problem in machine learning and computer vision. It involves automatically identifying digits (0-9) from images.

The goal of this project is to develop a machine learning model that accurately classifies handwritten digits using the MNIST dataset.

Data Description:

The MNIST dataset is a benchmark dataset in the field of machine learning. It contains 70,000 grayscale images of handwritten digits, each of size 28x28 pixels. The dataset is divided into 60,000 training samples and 10,000 testing samples. Each image is labeled with the corresponding digit (0-9).

Preprocessing Steps:

1. Normalization: The pixel values (0-255) were scaled to a range of 0 to 1 to improve model efficiency
2. Reshaping: Images were flattened into 1D arrays of size 784 (28x28) to suit the input format of the model.

Methodology:

We employed a Feedforward Neural Network (FNN) for this project. This type of neural network is structured with layers of neurons where information flows in one direction—from the input layer through hidden layers to the output layer.

Reason for Selection:

- FNNs are versatile and effective for classification tasks on structured data like the MNIST dataset.
- They are simple to work with as compared to convolutional neural networks.

Implementation

Feedforward Neural Network Architecture:

- **Framework:** TensorFlow/Keras
- **Layers:**
    - Input Layer: Accepts flattened 1D arrays of 784 features.
    - Hidden Layers: Fully connected layers with ReLU activation functions, each layer with 50 neurons.
    - Output Layer: A dense layer with 10 neurons and softmax activation to output class probabilities.
- **Optimizer**: Adam
- **Loss Function:** Sparse Categorical Cross-Entropy

Code Snippet:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28,28)), #input layer
    keras.layers.Dense(50,activation='relu'), #first hidden layer
    keras.layers.Dense(50,activation='relu'), #second hidden layer
    keras.layers.Dense(10,activation='sigmoid') #output layer
])
model.compile(optimizer='adam',
              loss = 'sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

Results and Analysis

Performance Metrics:

- Training Accuracy: 99.04%
- Testing Accuracy: 97.18%

Visualizations:

- Learning curves showing accuracy and loss over epochs.
- Confusion matrix highlighting misclassified digits.
- Using Heatmap from seaborn library to visualize the confusion matrix.

Insights:

- The FNN achieved high accuracy on the test set, demonstrating its effectiveness for digit recognition.

- The model performed well across most classes but showed minor misclassifications between similar digits such as 4 and 9.

Discussion

Challenges Faced:

- Overfitting: Addressed using dropout layers and data augmentation.
- Hyperparameter Tuning: Selecting the right number of neurons and dropout rates required experimentation.

Lessons Learned:

- Proper architecture design is crucial for achieving good performance.
- Data preprocessing plays a significant role in improving model accuracy and working.

Potential Improvements:

- Experiment with advanced neural networks such as deep FNNs with more layers.
- Use hyperparameter tuning techniques like grid search or Bayesian optimization.
- Integrate feature extraction techniques to enhance model input.

Conclusion

This project successfully demonstrated the use of a feedforward neural network for handwritten digit recognition. The model achieved a high testing accuracy of 97.18%, showcasing the effectiveness of FNNs for structured image data. Future work could involve exploring deeper architectures and applying the model to other digit datasets with greater complexity.