Skolar Machine Learning Project

ImageClassificationusingCNNandMLPonMNISTandFashion MNISTDatasets

Objective:

The goal of this assignment is toi mplement image classification models using Convolutional Neural Networks(CNN) and Multi-LayerPerceptrons(MLP) on two populardatasets: MNIST and Fashion MNIST. Students will explore different architectures and hyperparameters to gain insights into the impact of various configurations model performance.

Instructions:

1.DatasetExploration:

a. Download and load the MNISTand Fashion MNIST datasets.
b. Visualize a few samples from each data set to understand the characteristics of the images.

2. ConvolutionalNeuralNetwork(CNN)Model:

a. Implement a CNN architecture for image classification.

-Use a CNN with the following layers:

1.Convolutional Layer With 32 Filters,kernel size of (3,3),and ReLU activation.

2. MaxPooling layer with pool size (2,2).

3. Convolutional Layer With 32 Filters, kernel size of (3,3), and ReLU activation.

4. MaxPoolinglayerwithpoolsize(2,2).

5. Flatten Layer.

6. Fully Connected (dense) layer with 128 neurons and ReLu activation.

7.Output Layer With 10 Neurons (for MNIST)or 10 neurons (for Fashion MNIST) and softmax activation.

b. Train the CNN model on both MNIST and Fashion MNIST datasets.

-Use a batch size of 64.

-Train for 10 epochs.

-Set the learning rate to 0.001.

c. Experiment with different hyperparameters for the CNN model:

-Explore variations in the number of filters (e.g., 32, 64, 128) in the convolutional layers.

-Try different kernel sizes (e.g., (3,3), (5,5)) for convolutional layers.

-Adjust the learning rate (e.g., 0.001, 0.0001).

3. Multi-LayerPerceptron(MLP):

a. Implement an MLP architecture for image classification.

-UseanMLPwiththefollowinglayers:

1. Flatten Layer.

2. Fully Connected (dense) layer with 128 neurons and ReLU activation.

3. Dropout Layer With A Dropout Rate Of 0.5 (for regularization).

4. Fully connected (dense) layer with 64 neurons and ReLU activation.

5. Output layer with 10 neurons (for MNIST) or 10 neurons (for Fashion MNIST) and softmax activation.

b. Train the MLP model on both MNIST and Fashion MNIST datasets.

-Use a batch size of 64.

-Train for 10 epochs.

-Set the learning rate to0.001.

c. Experiment with different hyperparameters for the MLP model:

-Explore variations in the number of neurons in hidden layers (e.g., 64, 128, 256).

-Try different dropout rates (e.g., 0.3, 0.5) for regularization.

-Adjustthelearningrate(e.g.,0.001,0.0001).

4. Model Comparison and Analysis:

   a. Compare the performance of the CNN and MLP models on both MNIST and Fashion MNIST datasets.
   b. Discuss the impact of different hyperparameters on the model's accuracy, training time, and convergence.
   c. Visualize and analyze the confusion matrix for each model to identify common misclassifications.

5. Bonus (Optional):

   a. Implement data augmentation techniques for image preprocessing.
   b. Explore transfer learning by using pre-trained models (e.g., VGG16, ResNet) on Image Net and fine-tune them for MNIST and Fashion MNIST.

SubmissionGuidelines:

●Prepare a Jupyter Notebook Documenting Your Code, experiments, and analysis.

●Include Visualizations,such as training/validation accuracy and loss curves.

●Discuss Your Findings And Insights In The Notebook.

●Submit Your Assignment By[submission deadline].

EvaluationCriteria:

●Code Implementation And Correctness.

●Experimentation with different hyperparameters.

●Clarity And Completeness Of The Analysis.

●Comparison And Evaluation Of Models.

●Bonus Points For Implementing Optional Tasks