

Tester Document

Table of Contents

1. Development Phase

- 1.1 Hyperparameter Tuning
- 1.2 Data Loading and Preprocessing
- 1.3 Model Definition and Hyperparameter Tuning
- 1.4 Model Evaluation
- 1.5 Visualization

2. PyTorch Model

- 2.1 Set Up PyTorch Model
- 2.2 Define Hyperparameters and Model Training
- 2.3 Hyperparameter Tuning with GridSearch
- 2.4 Model Training
- 2.5 Evaluation and Metrics
- 2.6 Model Saving and Loading
- 2.7 Model Optimization
- 2.8 Transfer Learning
- 2.9 Model Pruning
- 2.10 Distributed Training
- 2.11 TensorBoard for Visualization
- 2.12 Deployment and Inference
- 2.13 Quantization
- 2.14 Custom Loss Functions and Optimizers

3. Deployment Phase

- 3.1 Model Exporting
- 3.2 Creating an API for Model Serving

4. Grid Search in PyTorch

- 4.1 Imports and Model Definition
- 4.2 Data Generation and Preprocessing
- 4.3 Hyperparameter Grid Definition
- 4.4 Grid Search and Model Training
- 4.5 Model Evaluation on Validation Set
- 4.6 Tracking Best Hyperparameters

Team Members

- 1. Shravani R S
- 2. Himaja S
- 3. Tarunaa A C
- 4. Karanbir Singh
- 5. Anushka Dutta

Date: 19th December 2024
Prepared By: Shravani R S

1. Development Phase

Objective: Build, train, and evaluate a robust model using hyperparameter tuning and preprocessing techniques.

Test Cases

Test ID	Test Case	Expected Outcome	Pass/Fail
TC1	Verify hyperparameter tuning results.	Optimal combination improves performance.	Pass
TC2	Load and preprocess data correctly.	Data is cleaned, normalized, and ready.	Pass
TC3	Evaluate model performance on validation.	Metrics like accuracy > 80%.	Pass
TC4	Visualize training results effectively.	Loss and accuracy graphs are clear.	Pass

2. PyTorch Model

Objective: Define, train, and optimize a PyTorch model while validating each phase of its development.

Test Cases

Test ID	Test Case	Expected Outcome	Pass/Fail
TC5	Verify model setup.	Model compiles successfully.	Pass
TC6	Define and train with optimal parameters.	Loss decreases steadily during training.	Pass
TC7	Conduct GridSearch for tuning.	Best hyperparameters identified.	Pass
TC8	Save and reload trained model.	Model restores with identical performance.	Pass
TC9	Optimize using advanced techniques.	Training becomes faster and more accurate.	Pass

3. Deployment Phase

Objective: Prepare and deploy the model for real-world usage.

Test Cases

Test ID	Test Case	Expected Outcome	Pass/Fail
TC10	Export the trained model.	Model is saved in ONNX format successfully.	Fail
TC11	Create an API for predictions.	API returns correct predictions quickly.	Pass

4. Grid Search in PyTorch

Objective: Perform GridSearch to identify the best-performing hyperparameters.

Test Cases

Test ID	Test Case	Expected Outcome	Pass/Fail
TC12	Define hyperparameter grid.	All combinations generated successfully.	Pass
TC13	Train models for each parameter set.	Models train without errors.	Pass
TC14	Evaluate on validation set.	Best parameter set identified.	Pass
TC15	Track and log best-performing results.	Logs are accurate and reproducible.	Pass