

MNIST Hyperparameter Tuning and Experiment Management System

Overview

This project implements an Experiment Management System using Streamlit for a simple ML/DL task to solve the MNIST challenge with PyTorch. The system provides a user-friendly web interface for hyperparameter tuning, running experiments, and managing the overall experimentation process.

Key Components

PyTorch Model

- A PyTorch model (`SimpleNN`) is defined for the MNIST dataset.
- The model architecture includes two fully connected layers.

Hyperparameter Tuning Sidebar

- Users can tune hyperparameters via the Streamlit sidebar:
 - Learning Rate Slider
 - Epochs Slider
 - Batch Size Slider
 - Hidden Units Slider
 - Optimizer Choice Dropdown

Hyperparameter Tuning

Learning Rate

0.01

0.00 0.10

Epochs

20

5 50

Batch Size

64

32 128

Hidden Units

128

64 256

Optimizer

Adam ▼

MLOps Integration with MLflow

- MLflow is used for experiment tracking and logging.
- Hyperparameters, metrics (test accuracy and loss), and the trained model are logged for each experiment.

Training Process and UI Interaction

- Real-time progress updates are displayed during the training process.
- Users can initiate new experiments or select from existing ones.
- The MLflow UI can be launched directly from the Streamlit app.

Displaying Runs from 2 Experiments

[Share](#)

metrics.rmse < 1 and params.model = "tree" Time created State: Active Datasets Sort: Created Columns

Table Chart Evaluation Experimental

	Run Name	Created	Dataset	Duration	Source	Models
<input type="checkbox"/>	resilient-kit-597	33 seconds ago	-		main.py	pytorch
<input type="checkbox"/>	intrigued-ray-366	44 seconds ago	-	8.2s	main.py	pytorch
<input type="checkbox"/>	gregarious-cod-67	51 seconds ago	-	6.6s	main.py	-
<input type="checkbox"/>	unleashed-kite-960	1 hour ago	-		main.py	pytorch

Experiment Management Sidebar

- The Experiment Management sidebar allows users to:
 - Choose New Experiment or Existing Experiment.
 - Input a name for a new experiment.
 - Select from existing experiments.

Select Experiment Type

☒ New Experiment ☐ Existing Experiment

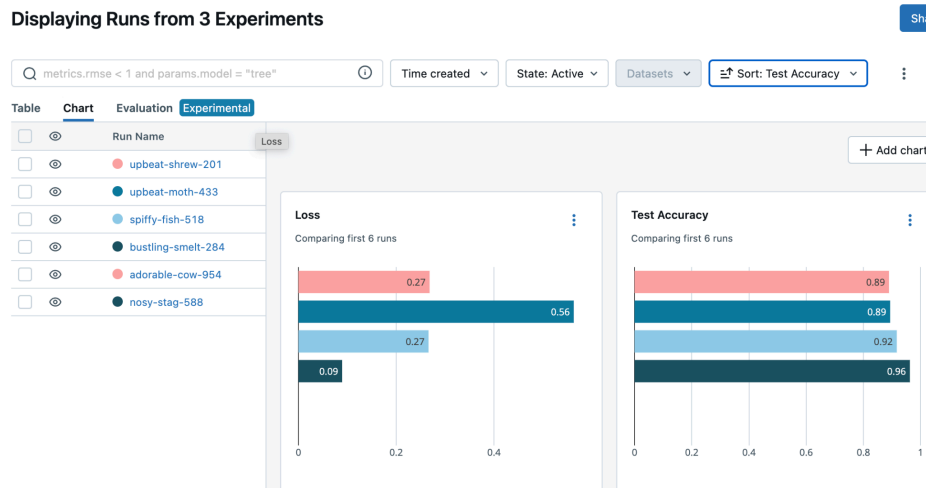
Enter the name for New Experiment

Train this new experiment

Sorting and Resuming

- Experiments can be sorted based on predefined metrics (accuracy and loss) by go to MLFlow UI, choosing the desired experiments, choosing the Chart tab and type of sorting on the left to see the comparison.

Displaying Runs from 3 Experiments



- The UI state is preserved, allowing users to close and reopen the browser without losing the current state.

Avoiding Duplicate Jobs

- The system checks if the exact same job (same hyperparameters) has already been executed, preventing duplicate runs.

How to Use

1. Ensure that all required libraries are installed before running the script. You can install them using `pip install streamlit pandas numpy torch torchvision mlflow`

2. Run the Streamlite app using `streamlit run your_script_name.py`.

ML Flow dashboard: <http://localhost:5000/>

2. Adjust hyperparameters using the sidebar sliders and options.

3. Choose to start a new experiment or select from existing ones.

4. Click the "Train" button to initiate model training.

5. Monitor the training progress in real-time.

6. Optionally, launch the MLflow UI to explore experiment details.

7. The Experiment Management UI allows sorting and filtering of experiments for ease of comparison.

REFERENCES

Hyperparameter Tuning with Optuna and MLflow: Integration of Optuna with MLflow for hyperparameter tuning.

<https://github.com/deepfindr/gnn-project>

<https://www.youtube.com/watch?v=2wEbOmsV028>

<https://github.com/JorgeDSprojects/MLFLOW-Streamlit/tree/main>