Assignment 3 - PyTorch Training Pipeline

1. Setup + How to run

The training script can be executed from the command line and supports a range of configurable arguments that control dataset selection, model architecture, training duration, optimization settings, and data augmentations.
A typical run may be launched using a command such as: `python train_pipeline.py --dataset cifar-100 --model resnet50 --optimizer adamw --scheduler steplr --epochs 30 --batch-size 128 --augmentations crop flip colorjitter`. The script accepts arguments for choosing the dataset (`mnist`, `cifar-10`, `cifar-100`, or `oxfordiiitpet`), the neural network model (e.g., `resnet18`, `resnet50`, `resnet14d, resnet26d, MLP`), the number of training epochs, and the initial batch size. Optimization can be controlled through parameters specifying the optimizer (`sgd`, `adam`, `adamw`, `muon`, or `sam`), the learning rate, momentum, and weight decay, as well as the learning rate scheduler (`steplr`, or `reducelronplateau`). The script additionally supports optional data augmentations such as cropping, flipping, rotation, translation, and color jitter, along with a configurable dataset cache size. The best-performing model, based on validation loss, is saved to the path provided with `--save-path`, and all runs are logged for inspection in TensorBoard.

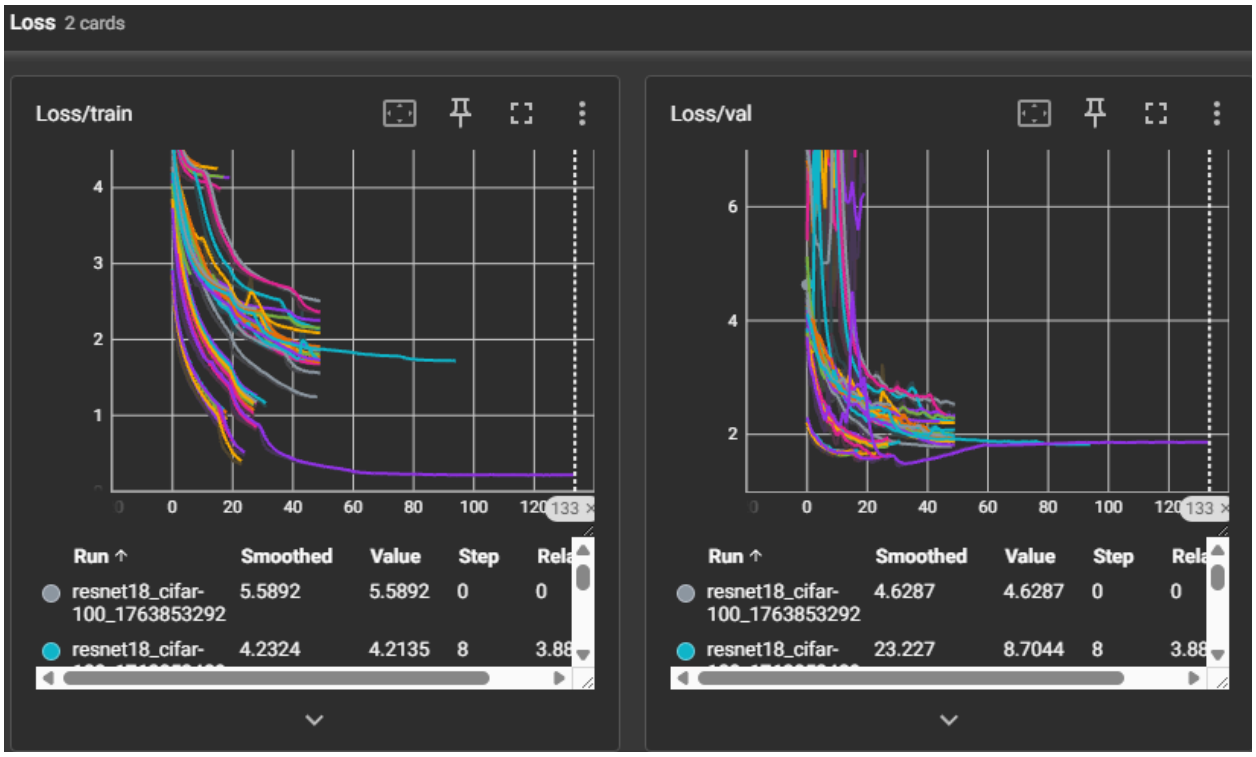
Configuration options

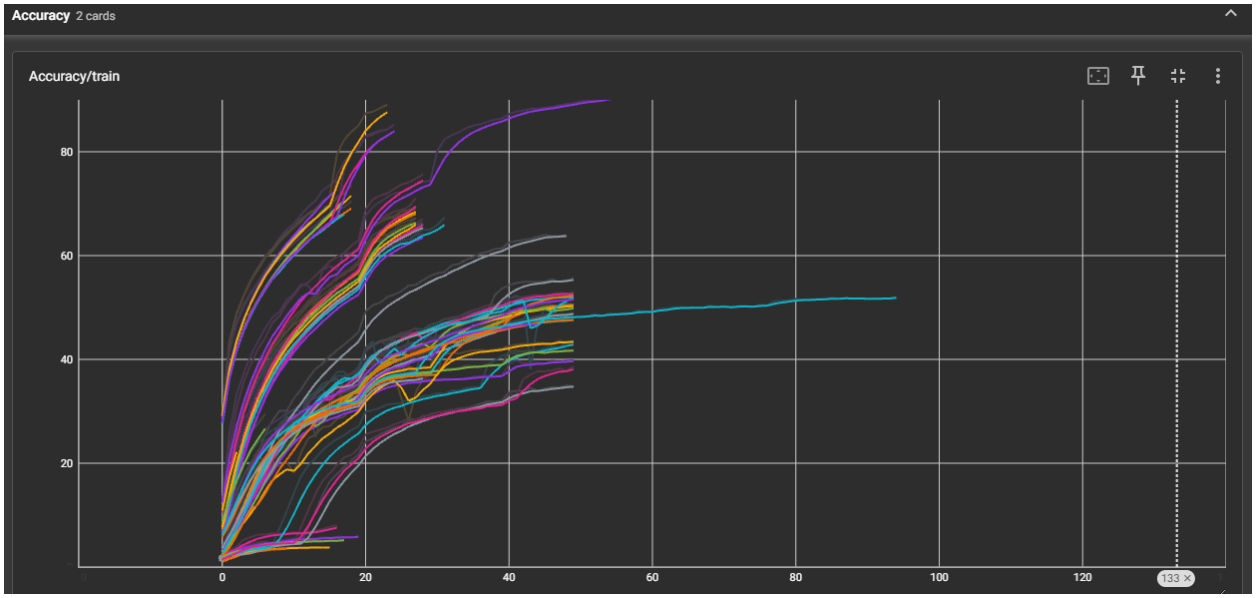
- **`--dataset`**: *str*, default=`"cifar-10"` **Choices**: `"mnist"`, `"cifar-10"`, `"cifar-100"`, `"oxfordiiitpet"`
- **`--model`**: *str*, default=`"resnet18"` **Choices:** `resnet18`, `resnet50`, `resnet14d, resnet26d, MLP`
- **`--epochs`**: *int*, default=`20`
- **`--batch-size`**: *int*, default=`128`
- **`--lr`**: *float*, default=`0.01`
- **`--weight-decay`**: *float*, default=`1e-4`
- **`--momentum`**: *float*, default=`0.9`
- **`--optimizer`**: *str*, default=`"rmsprop"` **Choices**: `"sgd"`, `"adam"`, `"adamw"`, `"muon"`, `"sam"`
- **`--scheduler`**: *str*, default=`"steplr"`
  Choices:`"steplr"`, `"reducelronplateau"`
- **`--augmentations`**: *list of str*, default=`["crop", "flip", "colorjitter"]`
- **`--cache-size`**: *int*, default=`4096`
- **`--allow-cpu`**: *flag*
  Enables CPU execution when CUDA is not available.
- **`--seed`**: *int*, default=`42`
- **`--save-path`**: *str*, default=`"best_model.pth"`

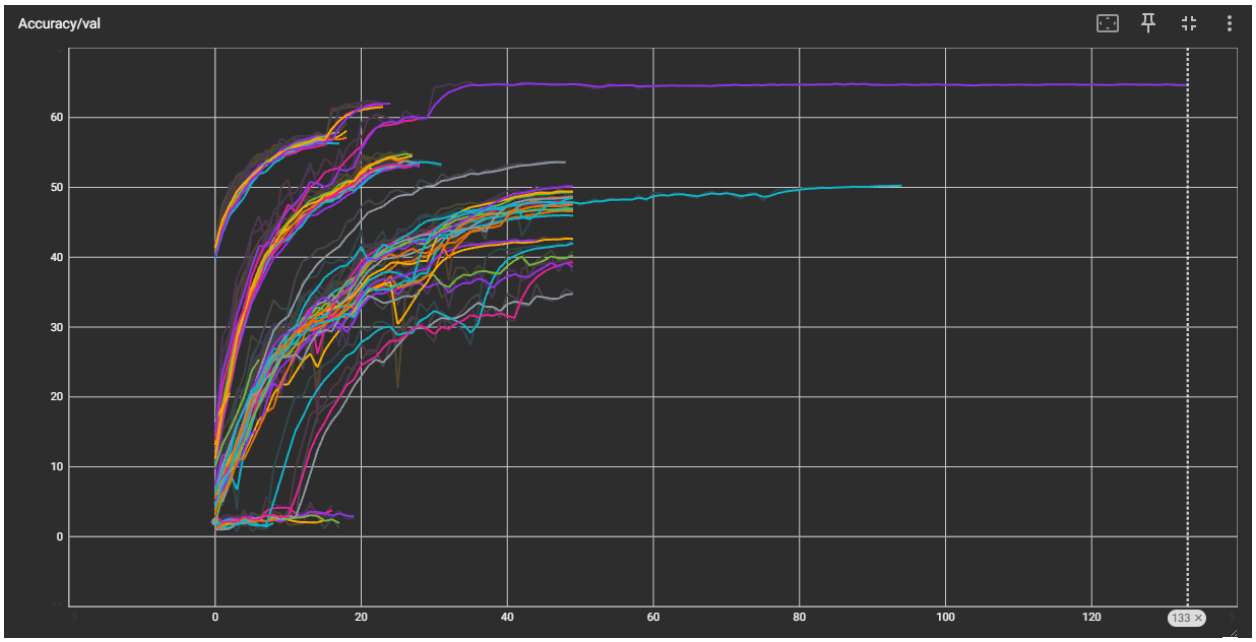
2. Implemented features

Datasets	[“mnist”,”cifar-10”,”cifar-100”,”oxfordiiitpet”]
Models	["resnet18","resnet50","resnet14d","resnet26d","MLP"]
Optimizers	["sgd","adam","adamw","muon","sam"]
Learning rate schedulers	["steplr","reducelronplateau"]
Data augmentation	["crop","flip","colorjitter","rotation","translation"]

3. Results







Pipeline Implementation (8 points)

Requirement	Estimated Points
Device-agnostic pipeline	**1/1**
Datasets are efficient and support data augmentation	**1/1**

Configurable datasets (MNIST, CIFAR-10, CIFAR-100, OxfordIITPet)	**1/1**
Support for resnet18, resnet50, resnest14d, resnest26d, and MLP.	**1/1**
SGD, Adam, AdamW, Muon, SAM	**1/1**
StepLR and ReduceLROnPlateau	**1/1**
Batch-size scheduler	**1/1**
TensorBoard + early stopping	**1/1**