

Report Homework 3

November 23, 2025

1 Setup Dependencies

The training pipeline relies on several standard Python and PyTorch packages. Note that `timm` and `wandb` are optional.

```
torch
torchvision
numpy
pyyaml
timm # For accessing additional models (optional)
wandb # For Weights & Biases logging (optional)
```

2 How to Run: Configuration and Execution

The script uses a configuration file (YAML) to define training parameters.

2.1 Create a Configuration File

Define the training hyperparameters, dataset, and model in a YAML file (e.g., `config_cifar100.yaml`).

```
dataset: 'CIFAR100'
data_root: 'data'
model: 'resnet50'
pretrained: True
batch_size: 128
epochs: 50
num_workers: 8
lr: 0.00005
optimizer: 'AdamW'
scheduler: 'ReduceLROnPlateau'
weight_decay: 0.0001
early_stopping_patience: 8
grad_clip: 1.0
batch_size_schedule: False
device: 'cuda'
wandb: true
```

2.2 Execute the Training Script

Run the script, specifying the configuration file path with `-config` and the output directory for logs and checkpoints with `-out`.

```
python3 pipeline.py --config config_cifar100.yaml --out out_config_cifar100
```

Arguments:

- `-config` specifies the YAML file containing the training settings.
- `-out` specifies the name of the output directory. This directory will contain:
 - `tb/`: TensorBoard logs.
 - `best.pth`: The checkpoint file for the model with the highest validation accuracy.

3 Results

Results for MNIST: - accuracy 0.9795 - exec. time 3.614 min

```
model: 'MLP'
pretrained: False
batch_size: 64
epochs: 15
num_workers: 4
lr: 0.005
optimizer: 'Adam'
scheduler: 'StepLR'
weight_decay: 0.00001
early_stopping_patience: 5
grad_clip: False
batch_size_schedule: False
device: 'cuda'
```

Results for OxfordIIITPet: - accuracy 0.9483 - exec. time 33.32 min

```
model: 'resnet50'
pretrained: True
batch_size: 32
epochs: 25
num_workers: 8
lr: 0.00005
optimizer: 'AdamW'
scheduler: 'ReduceLROnPlateau'
weight_decay: 0.0001
early_stopping_patience: 5
grad_clip: 1.0
batch_size_schedule: False
device: 'cuda'
```

Results for CIFAR-10: - accuracy 0.8385 - exec. time 1.404 hrs

```

model: 'resnet18'
pretrained: False
batch_size: 128
epochs: 100
num_workers: 8
lr: 0.1
optimizer: 'SGD'
scheduler: 'StepLR'
weight_decay: 0.0005
early_stopping_patience: 10
grad_clip: 0.5
batch_size_schedule: [{'epoch':80, 'batch_size':256}]
device: 'cuda'

```

Results for CIFAR-100:

- for all configurations the batch_size was set to 128
- for configuration 1 and 2 set grad-clip to 1.0
- for configuration 2 batch_size_schedule: *['epoch' : 75, 'batch_size' : 256, 'epoch' : 120, 'batch_size' : 512]*

model	pretrained	epochs	lr	opt	scheduler	train_time (Hrs)	test_acc
resnet50	True	50	0.00005	AdamW	ReduceLR	1.35	0.6373
resnest26d	False	150	0.1	SAM	StepLR	1.06	0.6657
resnet18	False	100	0.1	SGD	StepLR	1.45	0.5442
resnet50	False	100	0.001	Adam	ReduceLR	2.68	0.5868
MLP	False	50	0.003	AdamW	StepLR	1.153	0.1633
resnest14d	False	100	0.1	SGD	ReduceLR	2.493	0.6463
resnest26d	False	100	0.003	AdamW	StepLR	1.33	0.5729
resnet18	True	50	0.0001	AdamW	ReduceLR	1.266	0.5862

Table 1: Experiment hyperparameters and results for CIFAR-100

4 Expected score

First section ≈ 8 points, second section ≈ 2 points, third section 0 points, fourth section 0 points.
In total ≈ 10 points

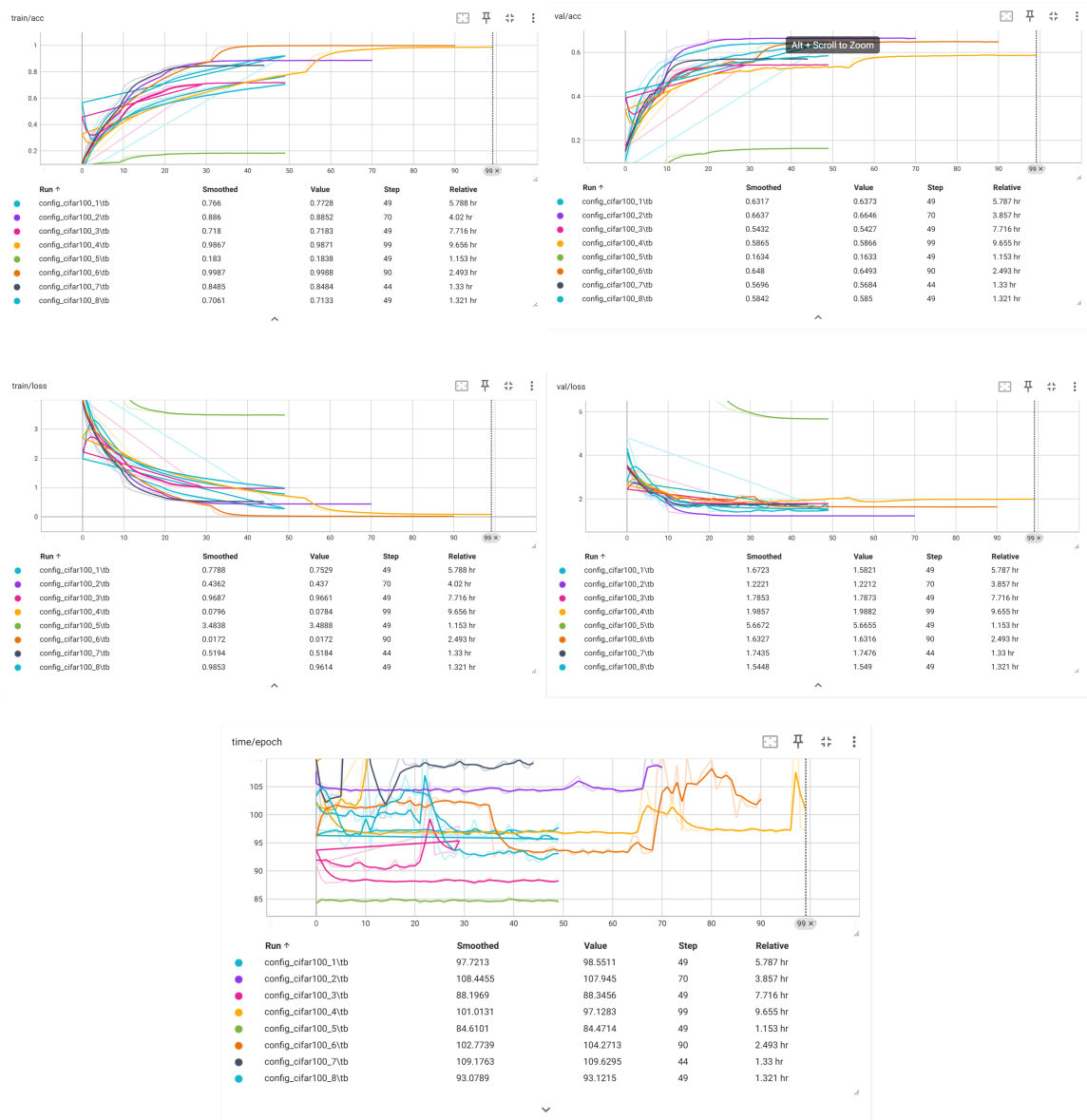


Figure 1: Tensorboard metrics