Date: 2023.12.10

Notes: The PyTorch template may be updated in the future, but the code in this example may not be affected by those changes.

MNIST

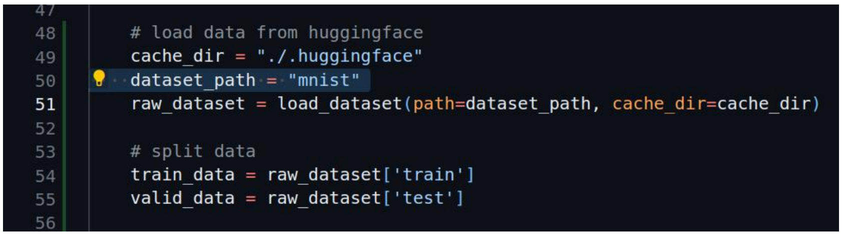
This is the record of how to tweak the pytorch template for this project, as well as what the training procedure looks like.

1. Tweak code

Remove the code unneeded, e.g., the code for nlp and the code about test dataset, as we only have train and valid splits.

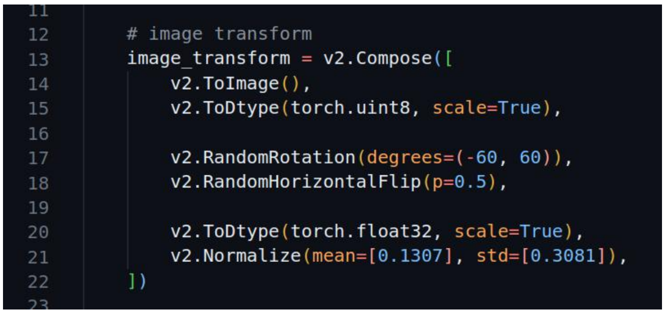
1.1 About data

The MNIST dataset will be loaded from huggingface. Specify the dataset in **./main.py** and load the train and valid splits.



```
    # load data from huggingface
    cache_dir = "./.huggingface"
    dataset_path = "mnist"
    raw_dataset = load_dataset(path=dataset_path, cache_dir=cache_dir)

    # split data
    train_data = raw_dataset['train']
    valid_data = raw_dataset['test']
```

The default function for preprocessing in **./preprocess.py** is fine, we just made some tweak to image transform.

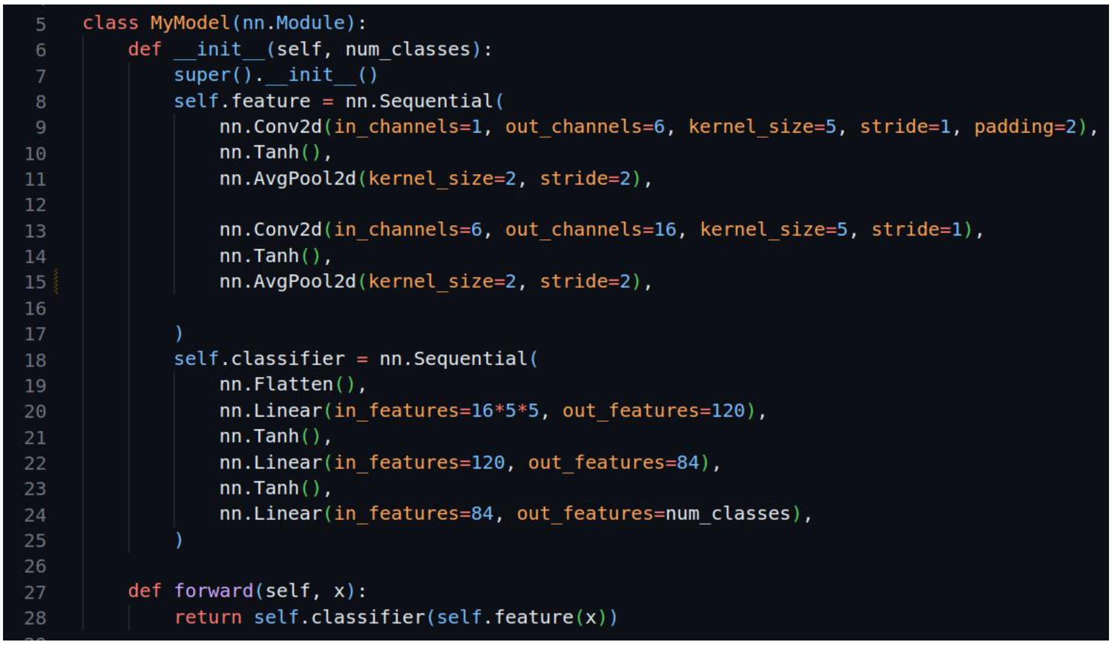


```
    # image transform
    image_transform = v2.Compose([
        v2.ToImage(),
        v2.ToDtype(torch.uint8, scale=True),

        v2.RandomRotation(degrees=(-60, 60)),
        v2.RandomHorizontalFlip(p=0.5),

        v2.ToDtype(torch.float32, scale=True),
        v2.Normalize(mean=[0.1307], std=[0.3081]),
    ])
```

Also, the ImageDataset class in **./dataset.py** could be used directly.

1.2 About model

Implement LeNet5 in **./model.py** ourselves.



```python
class MyModel(nn.Module):
    def __init__(self, num_classes):
        super().__init__()
        self.feature = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=6, kernel_size=5, stride=1, padding=2),
            nn.Tanh(),
            nn.AvgPool2d(kernel_size=2, stride=2),

            nn.Conv2d(in_channels=6, out_channels=16, kernel_size=5, stride=1),
            nn.Tanh(),
            nn.AvgPool2d(kernel_size=2, stride=2),

        )
        self.classifier = nn.Sequential(
            nn.Flatten(),
            nn.Linear(in_features=16*5*5, out_features=120),
            nn.Tanh(),
            nn.Linear(in_features=120, out_features=84),
            nn.Tanh(),
            nn.Linear(in_features=84, out_features=num_classes),
        )

    def forward(self, x):
        return self.classifier(self.feature(x))
```

1.3 About training

By default, the template uses CrossEntropyLoss for criterion, AdamW for optimizer, CosineAnnealingWarmRestarts for lr scheduler, which seems appropriate. So I didn't touch these in **./main.py**.

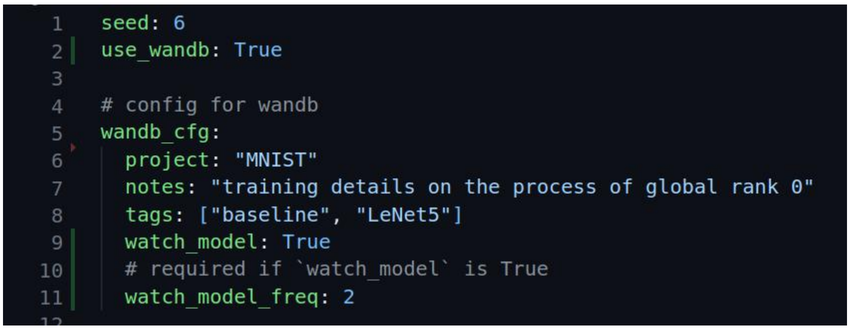
The Trainer in **./trainer.py** is ready-to-use, and it is recommended to use it directly without any alterations.

The template includes accuracy for test method, which just fit our demand in this simple project. So I kept it and didn't add more test methods.

1.4 About config

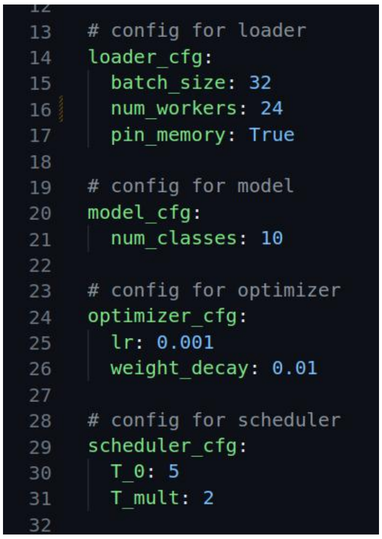
Tweak configurations in **./config.yaml**:

Use wandb to track experiment, set related config.



```yaml
seed: 6
use_wandb: True

# config for wandb
wandb_cfg:
  project: "MNIST"
  notes: "training details on the process of global rank 0"
  tags: ["baseline", "LeNet5"]
  watch_model: True
  # required if `watch_model` is True
  watch_model_freq: 2
```

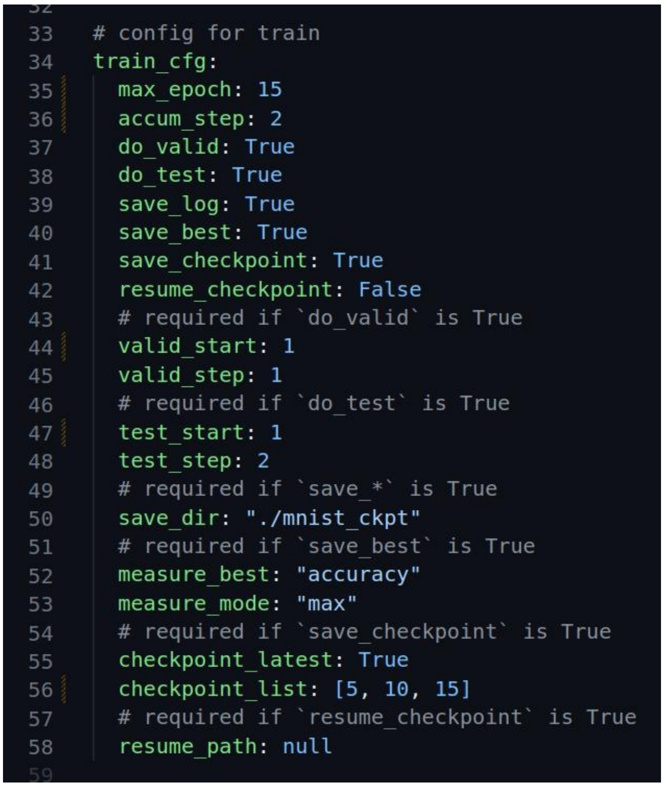
Tweak config for dataloader (e.g., batch_size), model, optimizer (e.g., lr), and lr scheduler.



```yaml
# config for loader
loader_cfg:
  batch_size: 32
  num_workers: 24
  pin_memory: True

# config for model
model_cfg:
  num_classes: 10

# config for optimizer
optimizer_cfg:
  lr: 0.001
  weight_decay: 0.01

# config for scheduler
scheduler_cfg:
  T_0: 5
  T_mult: 2
```

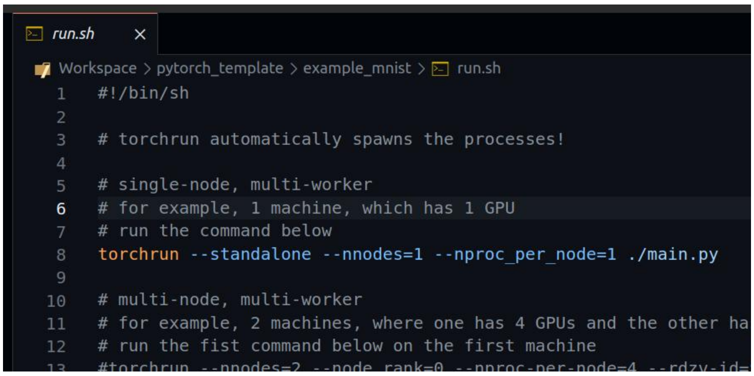
Tweak config for training. Here I made it to:

train up to 15 epochs;
use gradient accumulation at step 2;
do validation and test accuracy;
save logs, best model, and checkpoints during training;
train from scratch rather than from checkpoint;
start validation at epoch 1 and at every 1 epoch;
start testing accuracy at epoch 1 and at every 2 epoch;
save logs and checkpoints to an existing directory;
use accuracy to measure best model;
save latest checkpoint and checkpoints at specified epochs.



```yaml
# config for train
train_cfg:
  max_epoch: 15
  accum_step: 2
  do_valid: True
  do_test: True
  save_log: True
  save_best: True
  save_checkpoint: True
  resume_checkpoint: False
  # required if `do_valid` is True
  valid_start: 1
  valid_step: 1
  # required if `do_test` is True
  test_start: 1
  test_step: 2
  # required if `save_*` is True
  save_dir: "./mnist_ckpt"
  # required if `save_best` is True
  measure_best: "accuracy"
  measure_mode: "max"
  # required if `save_checkpoint` is True
  checkpoint_latest: True
  checkpoint_list: [5, 10, 15]
  # required if `resume_checkpoint` is True
  resume_path: null
```

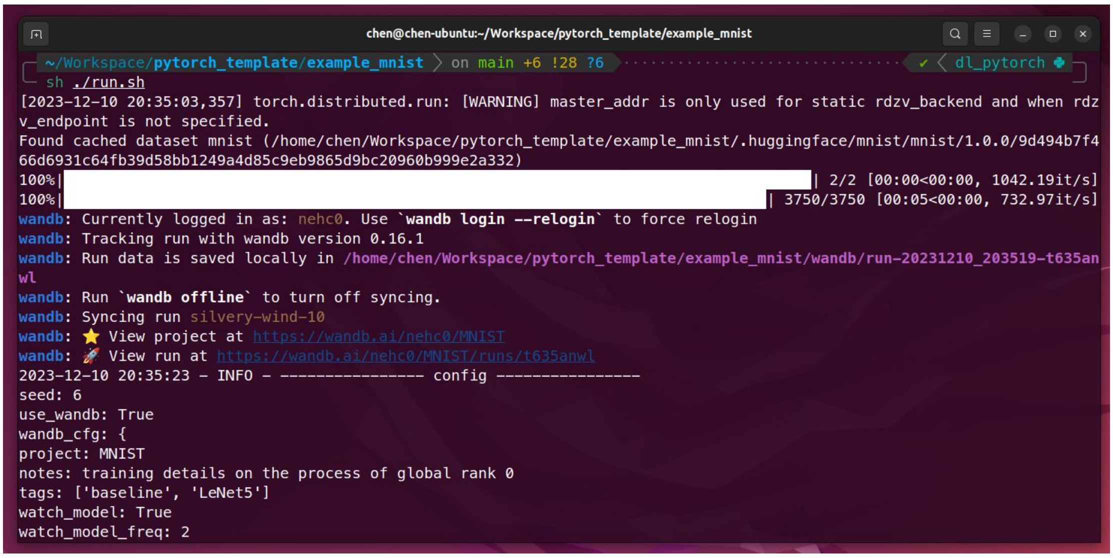
Finally, adjust **./run.sh** based on the machine architecture. I ran it on my laptop with single gpu.



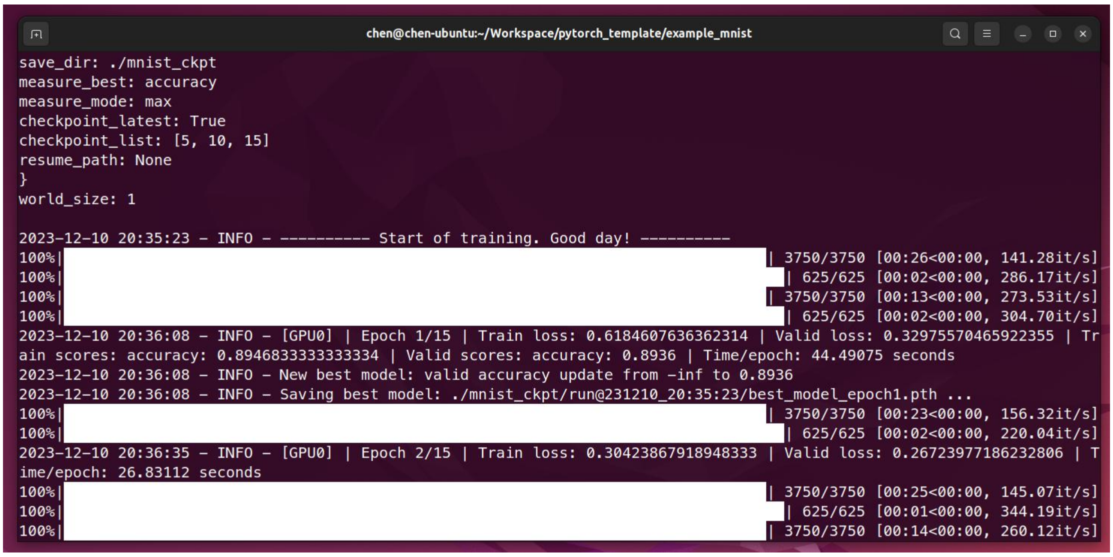
```
#!/bin/sh

# torchrun automatically spawns the processes!

# single-node, multi-worker
# for example, 1 machine, which has 1 GPU
# run the command below
torchrun --standalone --nnodes=1 --nproc_per_node=1 ./main.py

# multi-node, multi-worker
# for example, 2 machines, where one has 4 GPUs and the other ha
# run the fist command below on the first machine
#torchrun --nnodes=2 --node_rank=0 --nproc-per-node=4 --rdzv-id=
```

2. Training appearance

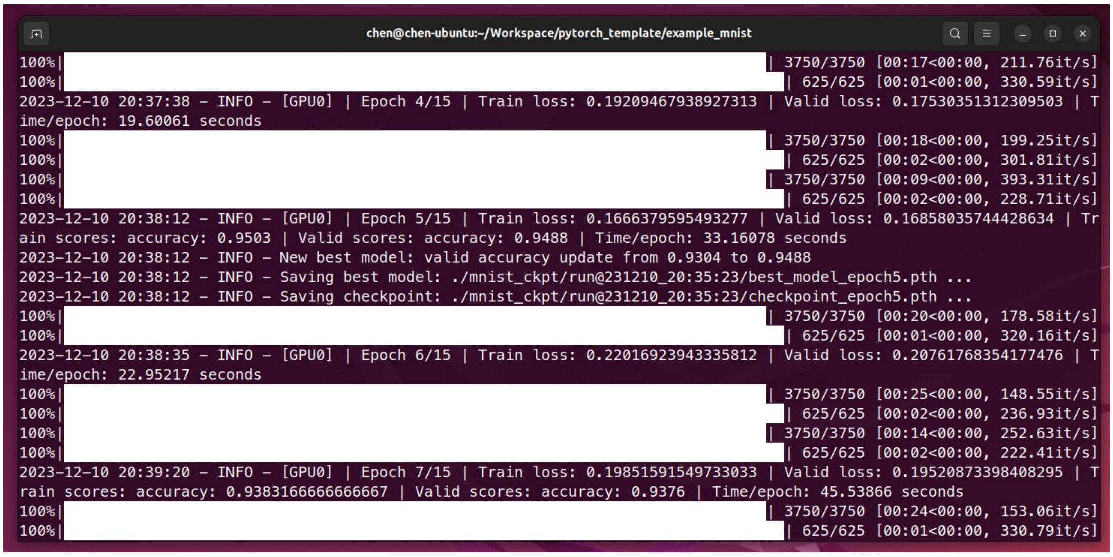
terminal:



```
~/Workspace/pytorch_template/example_mnist > on main +6 !28 ?6 ························ ✓ < dl_pytorch
sh ./run.sh
[2023-12-10 20:35:03,357] torch.distributed.run: [WARNING] master_addr is only used for static rdzv_backend and when rdz
v_endpoint is not specified.
Found cached dataset mnist (/home/chen/Workspace/pytorch_template/example_mnist/.huggingface/mnist/mnist/1.0.0/9d494b7f4
66d6931c64fb39d58bb1249a4d85c9eb9865d9bc20960b999e2a332)
100%|██████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 1042.19it/s]
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:05<00:00, 732.97it/s]
wandb: Currently logged in as: nehc0. Use `wandb login --relogin` to force relogin
wandb: Tracking run with wandb version 0.16.1
wandb: Run data is saved locally in /home/chen/Workspace/pytorch_template/example_mnist/wandb/run-20231210_203519-t635an
wl
wandb: Run `wandb offline` to turn off syncing.
wandb: Syncing run silvery-wind-10
wandb: ⭐ View project at https://wandb.ai/nehc0/MNIST
wandb: 🚀 View run at https://wandb.ai/nehc0/MNIST/runs/t635anwl
2023-12-10 20:35:23 - INFO - ---------------- config ----------------
seed: 6
use_wandb: True
wandb_cfg: {
project: MNIST
notes: training details on the process of global rank 0
tags: ['baseline', 'LeNet5']
watch_model: True
watch_model_freq: 2
```

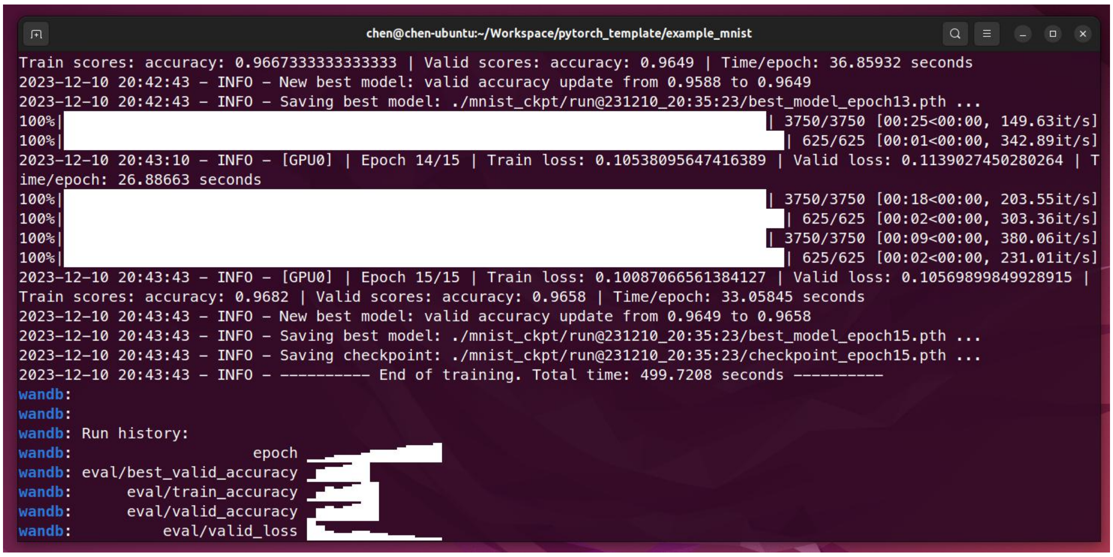


```
save_dir: ./mnist_ckpt
measure_best: accuracy
measure_mode: max
checkpoint_latest: True
checkpoint_list: [5, 10, 15]
resume_path: None
}
world_size: 1

2023-12-10 20:35:23 - INFO - ---------- Start of training. Good day! ----------
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:26<00:00, 141.28it/s]
100%|███████████████████████████████████████████████████████████████████| 625/625 [00:02<00:00, 286.17it/s]
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:13<00:00, 273.53it/s]
100%|███████████████████████████████████████████████████████████████████| 625/625 [00:02<00:00, 304.70it/s]
2023-12-10 20:36:08 - INFO - [GPU0] | Epoch 1/15 | Train loss: 0.6184607636362314 | Valid loss: 0.329755704659223555 | Tr
ain scores: accuracy: 0.8946833333333334 | Valid scores: accuracy: 0.8936 | Time/epoch: 44.49075 seconds
2023-12-10 20:36:08 - INFO - New best model: valid accuracy update from -inf to 0.8936
2023-12-10 20:36:08 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch1.pth ...
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:23<00:00, 156.32it/s]
100%|███████████████████████████████████████████████████████████████████| 625/625 [00:02<00:00, 220.04it/s]
2023-12-10 20:36:35 - INFO - [GPU0] | Epoch 2/15 | Train loss: 0.30423867918948333 | Valid loss: 0.26723977186232806 | T
ime/epoch: 26.83112 seconds
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:25<00:00, 145.07it/s]
100%|███████████████████████████████████████████████████████████████████| 625/625 [00:01<00:00, 344.19it/s]
100%|█████████████████████████████████████████████████████████████████| 3750/3750 [00:14<00:00, 260.12it/s]
```



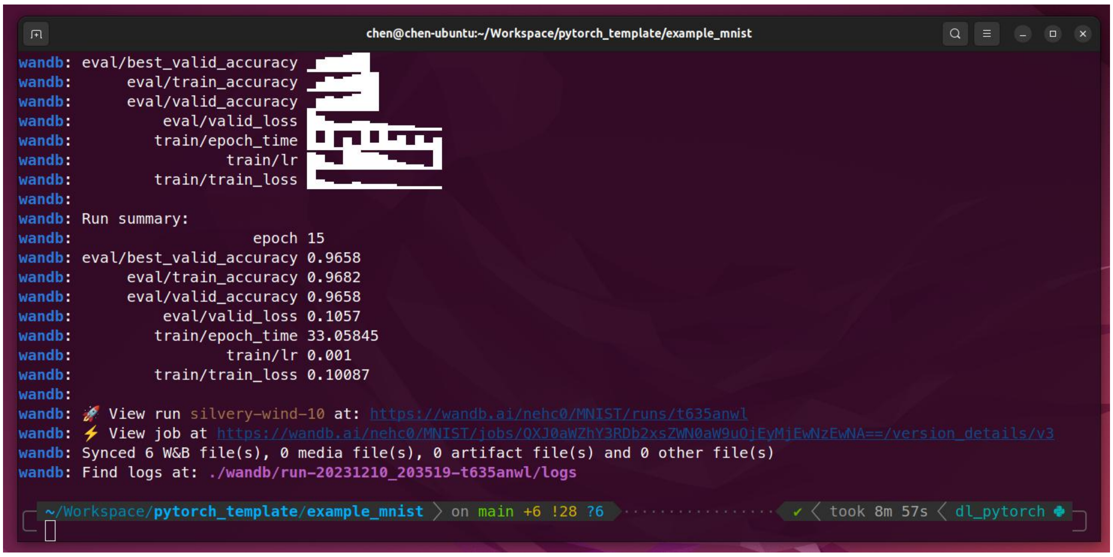
chen@chen-ubuntu:~/Workspace/pytorch_template/example_mnist

```
100%|████████████████████████████████████████| 3750/3750 [00:17<00:00, 211.76it/s]
100%|█████████████████████████████████████████| 625/625 [00:01<00:00, 330.59it/s]
2023-12-10 20:37:38 - INFO - [GPU0] | Epoch 4/15 | Train loss: 0.19209467938927313 | Valid loss: 0.17530351312309503 | T
ime/epoch: 19.60061 seconds
100%|████████████████████████████████████████| 3750/3750 [00:18<00:00, 199.25it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 301.81it/s]
100%|████████████████████████████████████████| 3750/3750 [00:09<00:00, 393.31it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 228.71it/s]
2023-12-10 20:38:12 - INFO - [GPU0] | Epoch 5/15 | Train loss: 0.1666379595493277 | Valid loss: 0.16858035744428634 | Tr
ain scores: accuracy: 0.9503 | Valid scores: accuracy: 0.9488 | Time/epoch: 33.16078 seconds
2023-12-10 20:38:12 - INFO - New best model: valid accuracy update from 0.9304 to 0.9488
2023-12-10 20:38:12 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch5.pth ...
2023-12-10 20:38:12 - INFO - Saving checkpoint: ./mnist_ckpt/run@231210_20:35:23/checkpoint_epoch5.pth ...
100%|████████████████████████████████████████| 3750/3750 [00:20<00:00, 178.58it/s]
100%|█████████████████████████████████████████| 625/625 [00:01<00:00, 320.16it/s]
2023-12-10 20:38:35 - INFO - [GPU0] | Epoch 6/15 | Train loss: 0.22016923943335812 | Valid loss: 0.20761768354177476 | T
ime/epoch: 22.95217 seconds
100%|████████████████████████████████████████| 3750/3750 [00:25<00:00, 148.55it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 236.93it/s]
100%|████████████████████████████████████████| 3750/3750 [00:14<00:00, 252.63it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 222.41it/s]
2023-12-10 20:39:20 - INFO - [GPU0] | Epoch 7/15 | Train loss: 0.19851591549733033 | Valid loss: 0.19520873398408295 | T
rain scores: accuracy: 0.9383166666666667 | Valid scores: accuracy: 0.9376 | Time/epoch: 45.53866 seconds
100%|████████████████████████████████████████| 3750/3750 [00:24<00:00, 153.06it/s]
100%|█████████████████████████████████████████| 625/625 [00:01<00:00, 330.79it/s]
```



chen@chen-ubuntu:~/Workspace/pytorch_template/example_mnist

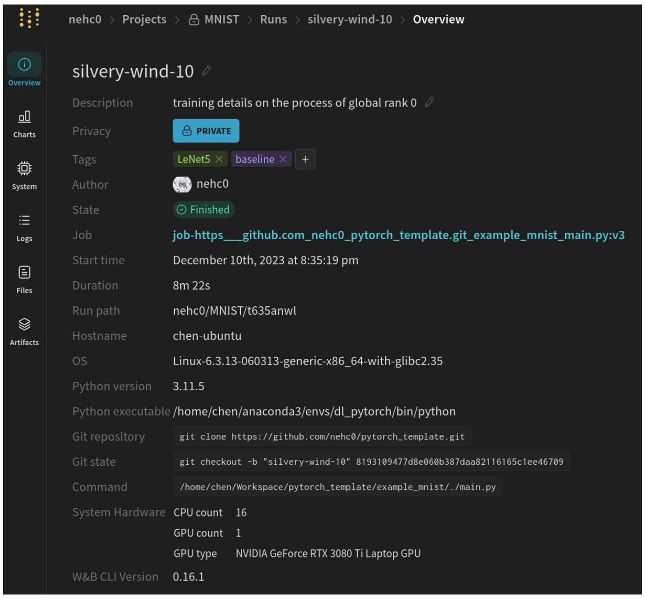
```
Train scores: accuracy: 0.9667333333333333 | Valid scores: accuracy: 0.9649 | Time/epoch: 36.85932 seconds
2023-12-10 20:42:43 - INFO - New best model: valid accuracy update from 0.9588 to 0.9649
2023-12-10 20:42:43 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch13.pth ...
100%|████████████████████████████████████████| 3750/3750 [00:25<00:00, 149.63it/s]
100%|█████████████████████████████████████████| 625/625 [00:01<00:00, 342.89it/s]
2023-12-10 20:43:10 - INFO - [GPU0] | Epoch 14/15 | Train loss: 0.10538095647416389 | Valid loss: 0.1139027450280264 | T
ime/epoch: 26.88663 seconds
100%|████████████████████████████████████████| 3750/3750 [00:18<00:00, 203.55it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 303.36it/s]
100%|████████████████████████████████████████| 3750/3750 [00:09<00:00, 380.06it/s]
100%|█████████████████████████████████████████| 625/625 [00:02<00:00, 231.01it/s]
2023-12-10 20:43:43 - INFO - [GPU0] | Epoch 15/15 | Train loss: 0.10087066561384127 | Valid loss: 0.10569899849928915 |
Train scores: accuracy: 0.9682 | Valid scores: accuracy: 0.9658 | Time/epoch: 33.05845 seconds
2023-12-10 20:43:43 - INFO - New best model: valid accuracy update from 0.9649 to 0.9658
2023-12-10 20:43:43 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch15.pth ...
2023-12-10 20:43:43 - INFO - Saving checkpoint: ./mnist_ckpt/run@231210_20:35:23/checkpoint_epoch15.pth ...
2023-12-10 20:43:43 - INFO - ---------- End of training. Total time: 499.7208 seconds ----------
wandb:
wandb:
wandb: Run history:
wandb:                    epoch
wandb: eval/best_valid_accuracy
wandb:      eval/train_accuracy
wandb:      eval/valid_accuracy
wandb:          eval/valid_loss
```



chen@chen-ubuntu:~/Workspace/pytorch_template/example_mnist

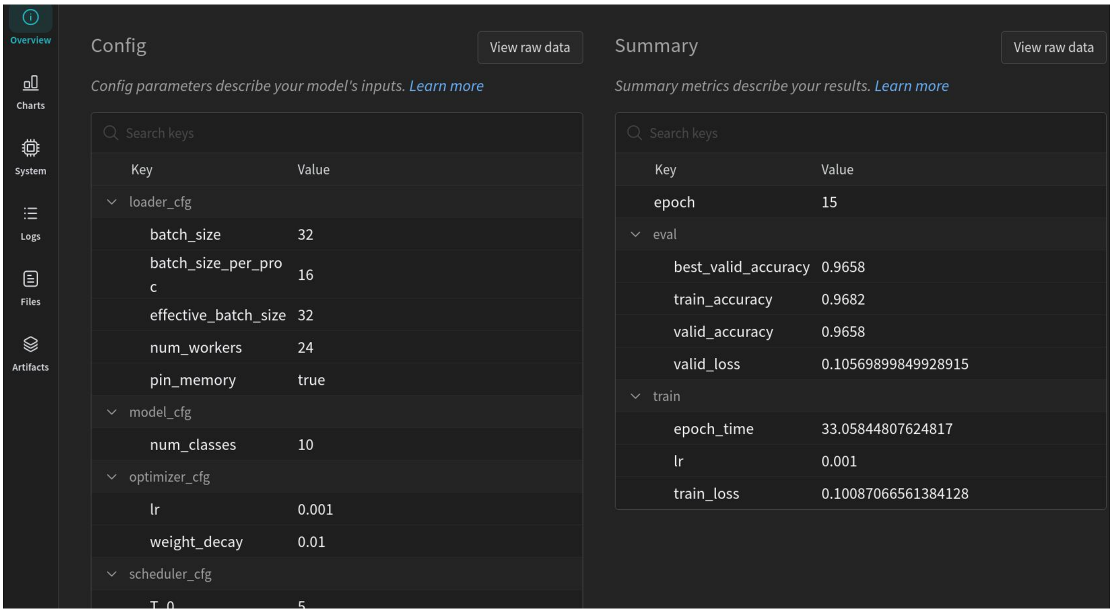
```
wandb: eval/best_valid_accuracy
wandb:      eval/train_accuracy
wandb:      eval/valid_accuracy
wandb:          eval/valid_loss
wandb:         train/epoch_time
wandb:                 train/lr
wandb:         train/train_loss
wandb:
wandb: Run summary:
wandb:                    epoch 15
wandb: eval/best_valid_accuracy 0.9658
wandb:      eval/train_accuracy 0.9682
wandb:      eval/valid_accuracy 0.9658
wandb:          eval/valid_loss 0.1057
wandb:         train/epoch_time 33.05845
wandb:                 train/lr 0.001
wandb:         train/train_loss 0.10087
wandb:
wandb: 🚀 View run silvery-wind-10 at: https://wandb.ai/nehc0/MNIST/runs/t635anwl
wandb: ⚡ View job at https://wandb.ai/nehc0/MNIST/jobs/QXJ0aWZhY3RDb2xsZWN0aW9uOjEyMjEwNzEwNA==/version_details/v3
wandb: Synced 6 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: ./wandb/run-20231210_203519-t635anwl/logs

~/Workspace/pytorch_template/example_mnist 〉 on main +6 !28 ?6 ·················· ✓ 〈 took 8m 57s 〈 dl_pytorch
```

wandb:


nehc0 > Projects > MNIST > Runs > silvery-wind-10 > Overview
Overview
Charts
System
Logs
Files
Artifacts

silvery-wind-10

Description	training details on the process of global rank 0
Privacy	PRIVATE
Tags	LeNet5 × baseline × +
Author	nehc0
State	Finished
Job	job-https___github.com_nehc0_pytorch_template.git_example_mnist_main.py:v3
Start time	December 10th, 2023 at 8:35:19 pm
Duration	8m 22s
Run path	nehc0/MNIST/t635anwl
Hostname	chen-ubuntu
OS	Linux-6.3.13-060313-generic-x86_64-with-glibc2.35
Python version	3.11.5
Python executable	/home/chen/anaconda3/envs/dl_pytorch/bin/python
Git repository	`git clone https://github.com/nehc0/pytorch_template.git`
Git state	`git checkout -b "silvery-wind-10" 8193109477d8e060b387daa82116165c1ee46709`
Command	`/home/chen/Workspace/pytorch_template/example_mnist/./main.py`
System Hardware	CPU count 16 GPU count 1 GPU type NVIDIA GeForce RTX 3080 Ti Laptop GPU
W&B CLI Version	0.16.1



Config

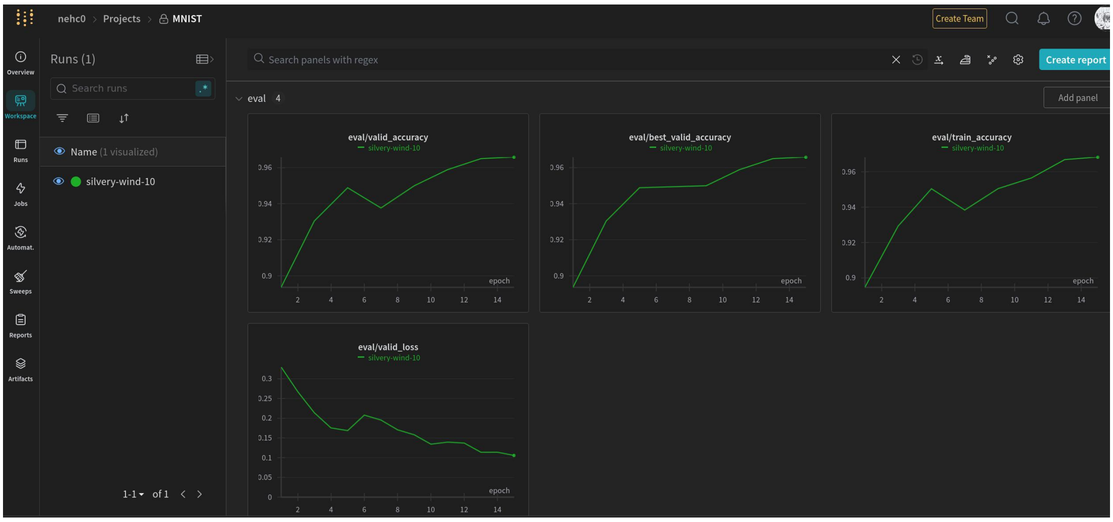
Config parameters describe your model's inputs. Learn more

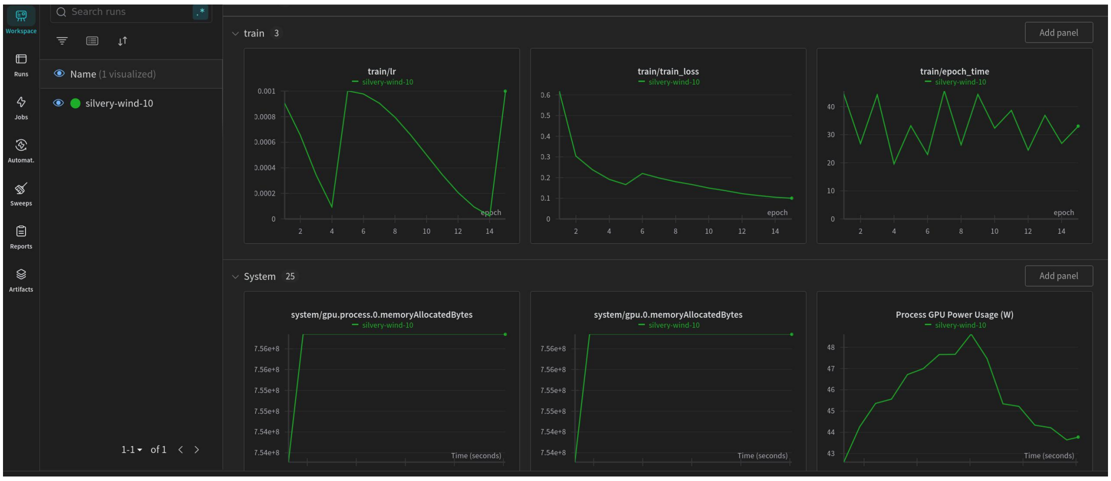
Key	Value
loader_cfg	
batch_size	32
batch_size_per_proc	16
effective_batch_size	32
num_workers	24
pin_memory	true
model_cfg	
num_classes	10
optimizer_cfg	
lr	0.001
weight_decay	0.01
scheduler_cfg	
T_0	5

Summary

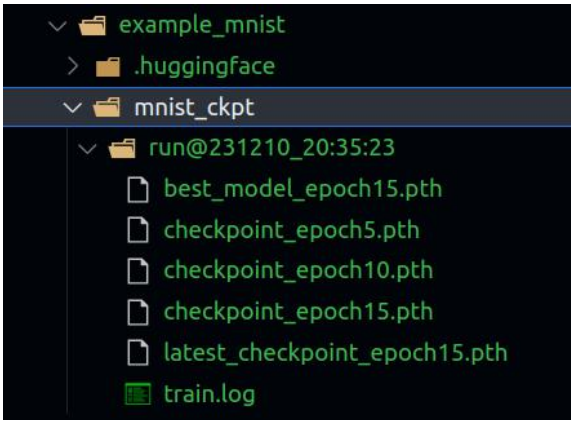
Summary metrics describe your results. Learn more

Key	Value
epoch	15
eval	
best_valid_accuracy	0.9658
train_accuracy	0.9682
valid_accuracy	0.9658
valid_loss	0.10569899849928915
train	
epoch_time	33.05844807624817
lr	0.001
train_loss	0.10087066561384128

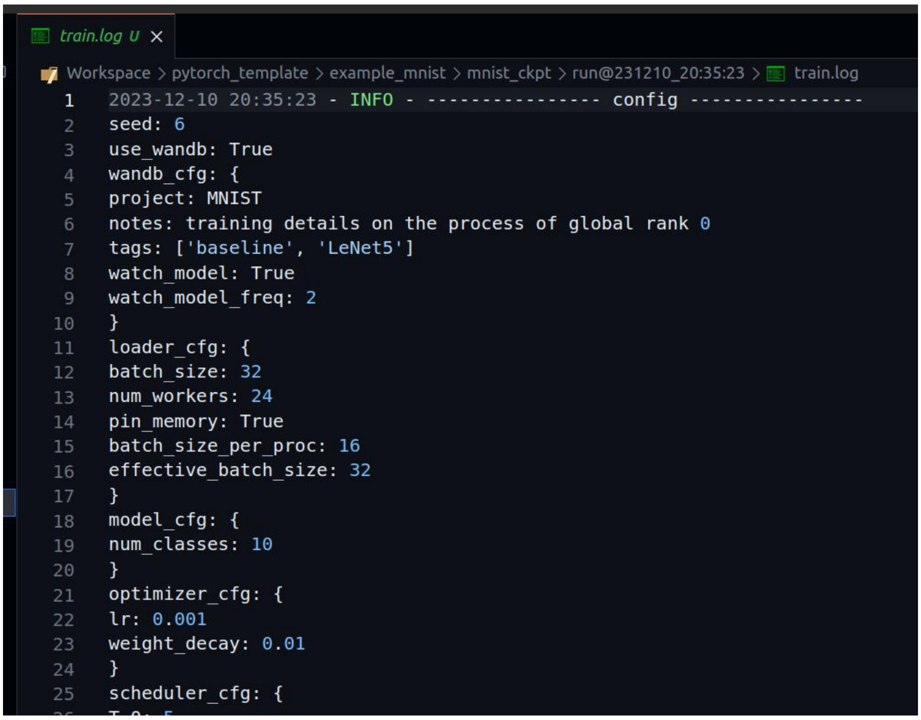




saved logs and checkpoints:



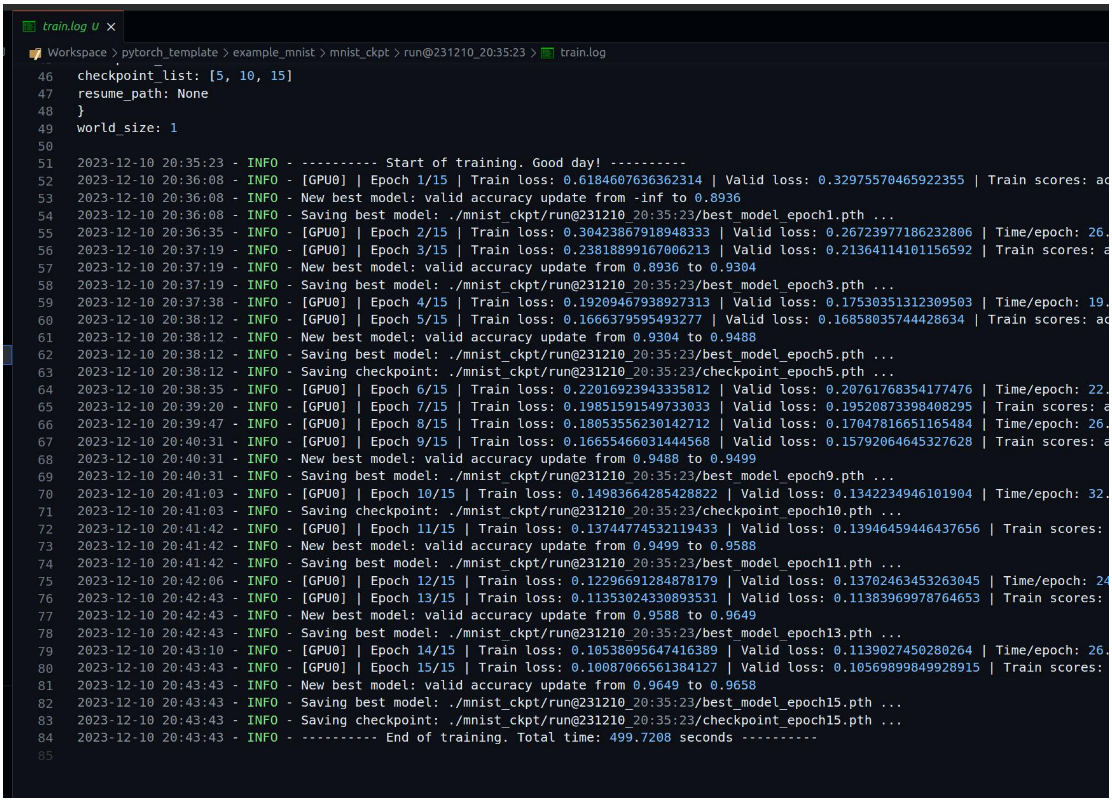
```
example_mnist
  .huggingface
  mnist_ckpt
    run@231210_20:35:23
      best_model_epoch15.pth
      checkpoint_epoch5.pth
      checkpoint_epoch10.pth
      checkpoint_epoch15.pth
      latest_checkpoint_epoch15.pth
      train.log
```



train.log U ×

Workspace > pytorch_template > example_mnist > mnist_ckpt > run@231210_20:35:23 > train.log

```
2023-12-10 20:35:23 - INFO - ---------------- config ----------------
seed: 6
use_wandb: True
wandb_cfg: {
project: MNIST
notes: training details on the process of global rank 0
tags: ['baseline', 'LeNet5']
watch_model: True
watch_model_freq: 2
}
loader_cfg: {
batch_size: 32
num_workers: 24
pin_memory: True
batch_size_per_proc: 16
effective_batch_size: 32
}
model_cfg: {
num_classes: 10
}
optimizer_cfg: {
lr: 0.001
weight_decay: 0.01
}
scheduler_cfg: {
T_0: 5
```



train.log U
Workspace > pytorch_template > example_mnist > mnist_ckpt > run@231210_20:35:23 > train.log

```
checkpoint_list: [5, 10, 15]
resume_path: None
}
world_size: 1

2023-12-10 20:35:23 - INFO - ---------- Start of training. Good day! ----------
2023-12-10 20:36:08 - INFO - [GPU0] | Epoch 1/15 | Train loss: 0.6184607636362314 | Valid loss: 0.32975570465922355 | Train scores: ac
2023-12-10 20:36:08 - INFO - New best model: valid accuracy update from -inf to 0.8936
2023-12-10 20:36:08 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch1.pth ...
2023-12-10 20:36:35 - INFO - [GPU0] | Epoch 2/15 | Train loss: 0.30423867918948333 | Valid loss: 0.26723977186232806 | Time/epoch: 26.
2023-12-10 20:37:19 - INFO - [GPU0] | Epoch 3/15 | Train loss: 0.23818899167006213 | Valid loss: 0.21364114101156592 | Train scores: a
2023-12-10 20:37:19 - INFO - New best model: valid accuracy update from 0.8936 to 0.9304
2023-12-10 20:37:19 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch3.pth ...
2023-12-10 20:37:38 - INFO - [GPU0] | Epoch 4/15 | Train loss: 0.19209467938927313 | Valid loss: 0.17530351312309503 | Time/epoch: 19.
2023-12-10 20:38:12 - INFO - [GPU0] | Epoch 5/15 | Train loss: 0.1666379595493277 | Valid loss: 0.16858035744428634 | Train scores: ac
2023-12-10 20:38:12 - INFO - New best model: valid accuracy update from 0.9304 to 0.9488
2023-12-10 20:38:12 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch5.pth ...
2023-12-10 20:38:12 - INFO - Saving checkpoint: ./mnist_ckpt/run@231210_20:35:23/checkpoint_epoch5.pth ...
2023-12-10 20:38:35 - INFO - [GPU0] | Epoch 6/15 | Train loss: 0.22016923943335812 | Valid loss: 0.20761768354177476 | Time/epoch: 22.
2023-12-10 20:39:20 - INFO - [GPU0] | Epoch 7/15 | Train loss: 0.19851591549733033 | Valid loss: 0.19520873398408295 | Train scores: a
2023-12-10 20:39:47 - INFO - [GPU0] | Epoch 8/15 | Train loss: 0.18053556230142712 | Valid loss: 0.17047816651165484 | Time/epoch: 26.
2023-12-10 20:40:31 - INFO - [GPU0] | Epoch 9/15 | Train loss: 0.16655466031444568 | Valid loss: 0.15792064645327628 | Train scores: a
2023-12-10 20:40:31 - INFO - New best model: valid accuracy update from 0.9488 to 0.9499
2023-12-10 20:40:31 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch9.pth ...
2023-12-10 20:41:03 - INFO - [GPU0] | Epoch 10/15 | Train loss: 0.14983664285428822 | Valid loss: 0.1342234946101904 | Time/epoch: 32.
2023-12-10 20:41:03 - INFO - Saving checkpoint: ./mnist_ckpt/run@231210_20:35:23/checkpoint_epoch10.pth ...
2023-12-10 20:41:42 - INFO - [GPU0] | Epoch 11/15 | Train loss: 0.13744774532119433 | Valid loss: 0.13946459446437656 | Train scores:
2023-12-10 20:41:42 - INFO - New best model: valid accuracy update from 0.9499 to 0.9588
2023-12-10 20:41:42 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch11.pth ...
2023-12-10 20:42:06 - INFO - [GPU0] | Epoch 12/15 | Train loss: 0.12296691284878179 | Valid loss: 0.13702463453263045 | Time/epoch: 24
2023-12-10 20:42:43 - INFO - [GPU0] | Epoch 13/15 | Train loss: 0.11353024330893531 | Valid loss: 0.11383969978764653 | Train scores:
2023-12-10 20:42:43 - INFO - New best model: valid accuracy update from 0.9588 to 0.9649
2023-12-10 20:42:43 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch13.pth ...
2023-12-10 20:43:10 - INFO - [GPU0] | Epoch 14/15 | Train loss: 0.10538095647416389 | Valid loss: 0.1139027450280264 | Time/epoch: 26.
2023-12-10 20:43:43 - INFO - [GPU0] | Epoch 15/15 | Train loss: 0.10087066561384127 | Valid loss: 0.10569899849928915 | Train scores:
2023-12-10 20:43:43 - INFO - New best model: valid accuracy update from 0.9649 to 0.9658
2023-12-10 20:43:43 - INFO - Saving best model: ./mnist_ckpt/run@231210_20:35:23/best_model_epoch15.pth ...
2023-12-10 20:43:43 - INFO - Saving checkpoint: ./mnist_ckpt/run@231210_20:35:23/checkpoint_epoch15.pth ...
2023-12-10 20:43:43 - INFO - ---------- End of training. Total time: 499.7208 seconds ----------
```