

Date: 2023.12.07

The pytorch template may update in the future, but the code in this example is not affected by that.

# IMDb

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 cv, and the code about test dataset, as there's only train and valid data

tweak **./preprocess.py** according to the dataset format and implement label transform

```
8  def preprocess_nlp(  
9      train_text: list[str],  
10     lowercase: bool = True,  
11     rm_punctuation: bool = True,  
12     rm_stopword: bool = False,  
13     lemmatization: bool = True,  
14     min_freq: int = 1,  
15     max_tokens: int = 10000,  
16 ):  
17     """a function for preprocessing in NLP task  
18  
19     return text_transform, label_transform, vocab, tokenizer  
20  
21     """  
22  
23     # create tokenizer  
24     tokenizer = Tokenizer(  
25         lowercase=lowercase,  
26         rm_punctuation=rm_punctuation,  
27         rm_stopword=rm_stopword,  
28         lemmatization=lemmatization,  
29     )  
30  
31     # yield tokens  
32     def yield_tokens(data: list[str]):  
33         for sentence in data:  
34             tokens = tokenizer(sentence)  
35             yield tokens  
36  
37     token_generator = yield_tokens(data=train_text)  
38  
39     # special tokens  
40     specials = ['<unk>', ]  
41  
42     # build vocab  
43     vocab = build_vocab_from_iterator(  
44         iterator=token_generator,
```

tweak **./main.py**, add code to load data, IMDb dataset will be loaded from Huggingface, and I only use a subset to save time

```
47
48     # load data from huggingface
49     cache_dir = "./.huggingface"
50     dataset_path = "imdb"
51
52     imdb_dataset = load_dataset(path=dataset_path, cache_dir=cache_dir)
53     imdb_dataset_shuffled = imdb_dataset.shuffle(seed=0)
54
55     # only use a subset to save time
56     train_texts = imdb_dataset_shuffled['train']['text'][:1000]
57     train_labels = imdb_dataset_shuffled['train']['label'][:1000]
58     valid_texts = imdb_dataset_shuffled['test']['text'][:1000]
59     valid_labels = imdb_dataset_shuffled['test']['label'][:1000]
60
61     # preprocess
62     text_transform, label_transform, _, _ = preprocess_nlp(
63         train_text = train_texts,
64         **config['preprocess_nlp_cfg'],
65     )
66
```

tweak **./dataset.py**, according to the IMDb data format

```
4 class TextDataset(Dataset):
5     def __init__(self, texts, labels, text_transform=None, label_transform=None):
6         assert len(texts) == len(labels)
7         self.texts = texts
8         self.labels = labels
9         self.text_transform = text_transform
10        self.label_transform = label_transform
11
12    def __len__(self):
13        return len(self.labels)
14
15    def __getitem__(self, idx):
16        text = self.texts[idx]
17        label = self.labels[idx]
18        if self.text_transform:
19            text = self.text_transform(text)
20        if self.label_transform:
21            label = self.label_transform(label)
22        return text, label
23
```

tweak **./main.py**, as the TextDataset changed, tweak the part about dataset

```
67     # create datasets
68     train_dataset = TextDataset(
69         texts=train_texts,
70         labels=train_labels,
71         text_transform=text_transform,
72         label_transform=label_transform,
73     )
74     valid_dataset = TextDataset(
75         texts=valid_texts,
76         labels=valid_labels,
77         text_transform=text_transform,
78         label_transform=label_transform,
79     )
80
```

as **./model.py** in template is for text classification task by default, so I just use it directly with no tweak

tweak **./config.yaml**

tweak config for wandb

```
3 # config for wandb
4 wandb_cfg:
5   use_wandb: True
6   project: "IMDb"
7   notes: "training details on the process of global rank 0"
8   tags: ["IMDb dataset", "TransformerEncoder"]
9
```

the configs for preprocess, dataloader and model seem ok, so I just keep these

by default, the template use CrossEntropyLoss for criterion, AdamW for optimizer, CosineAnnealingWarmRestarts for lr scheduler, it seems ok, so I didn't change these, and just tweak the config for optimizer

```
35 # config for optimizer
36 optimizer_cfg:
37   lr: 0.0005
38   weight_decay: 0.001
39
```

for the train\_cfg, I set max\_epoch to 20, do validation and test accuracy at every epoch, set save\_dir and create the save\_dir manually

```
45 # config for train
46 train_cfg:
47   max_epoch: 20
48   do_valid: True
49   do_test: True
50   save_log: True
51   save_best: True
52   save_checkpoint: True
53   resume_checkpoint: False
54   # required if `do_valid` is True
55   valid_start: 1
56   valid_step: 1
57   # required if `do_test` is True
58   test_start: 1
59   test_step: 1
60   # required if `save_*` is True
61   save_dir: "./imdb_ckpt"
62   # required if `save_best` is True
63   measure_best: "accuracy"
64   measure_mode: "max"
65   # required if `save_checkpoint` is True
66   checkpoint_latest: True
67   checkpoint_list: [10, 15]
68   # required if `resume_checkpoint` is True
```

as for the test method, I just want to test accuracy, which is already implemented by the template, so no change

lastly, tweak `./run.sh`, as I will train on my laptop with single gpu

```
1  #!/bin/sh
2
3  # torchrun automatically spawns the processes!
4
5  # single-node, multi-worker
6  # for example, 1 machine, which has 1 GPU
7  # run the command below
8  torchrun --standalone --nnodes=1 --nproc_per_node=1 ./main.py
9
10 # multi-node, multi-worker
11 # for example, 2 machines, where one has 4 GPUs and the other has
12 # run the first command below on the first machine
13 #torchrun --nnodes=2 --node_rank=0 --nproc-per-node=4 --rdzv-id=$
```

## 2. train procedure

start training, terminal:

```
chen@chen-ubuntu:~/Workspace/pytorch_template/example_imdb
```

```
~/Workspace/pytorch_template/example_imdb  
sh ./run.sh  
[2023-12-07 11:25:23,149] torch.distributed.run: [WARNING] master_addr is only used for static rdzv_backend and when rdzv_endpoint is not specified.  
Found cached dataset imdb (/home/chen/Workspace/pytorch_template/example_imdb/.huggingface/imdb/plain_text/1.0.0/d613c88cf8fa3bab83b4ded3713f1f74830d1100e171db75bbddb80b3345c9c0)  
100%|██████████████████████████████████████████████████████████████████████████████| 3/3 [00:00<00:00, 778.12it/s]  
Loading cached shuffled indices for dataset at /home/chen/Workspace/pytorch_template/example_imdb/.huggingface/imdb/plain_text/1.0.0/d613c88cf8fa3bab83b4ded3713f1f74830d1100e171db75bbddb80b3345c9c0/cache-79aee49c9f40dc82.arrow  
Loading cached shuffled indices for dataset at /home/chen/Workspace/pytorch_template/example_imdb/.huggingface/imdb/plain_text/1.0.0/d613c88cf8fa3bab83b4ded3713f1f74830d1100e171db75bbddb80b3345c9c0/cache-5a09ddf1bd0fbc8.arrow  
Loading cached shuffled indices for dataset at /home/chen/Workspace/pytorch_template/example_imdb/.huggingface/imdb/plain_text/1.0.0/d613c88cf8fa3bab83b4ded3713f1f74830d1100e171db75bbddb80b3345c9c0/cache-f131e6602007628b.arrow  
/home/chen/anaconda3/envs/dl_pytorch/lib/python3.11/site-packages/torch/nn/modules/transformer.py:282: UserWarning: enable_nested_tensor is True, but self.use_nested_tensor is False because encoder_layer.self_attn.batch_first was not True(use batch_first for better inference performance)  
warnings.warn(f'enable_nested_tensor is True, but self.use_nested_tensor is False because {why_not_sparsity_fast_path}')  
wandb: Currently logged in as: nehc0. Use `wandb login --relogin` to force relogin  
wandb: wandb version 0.16.1 is available! To upgrade, please run:  
wandb: $ pip install wandb --upgrade  
wandb: Tracking run with wandb version 0.15.12  
wandb: Run data is saved locally in /home/chen/Workspace/pytorch_template/example_imdb/wandb/run-20231207_112616-ojzx3gk  
wandb: Run `wandb offline` to turn off syncing.  
wandb: Syncing run toasty-sponge-5  
wandb: 🌟 View project at https://wandb.ai/nehc0/IMDb  
wandb: 🚀 View run at https://wandb.ai/nehc0/IMDb/runs/ojzx3gkk  
100%|██████████████████████████████████████████████████████████████████████████████| 32/32 [00:05<00:00, 5.44it/s]  
2023-12-07 11:26:27 - INFO - ----- config -----  
seed: 6  
wandb_cfg: {  
use_wandb: True  
project: IMDb  
notes: training details on the process of global rank 0
```



```
chen@chen-ubuntu:~/Workspace/pytorch_template/example_imdb
```

```
save_best: True
save_checkpoint: True
resume_checkpoint: False
valid_start: 1
valid_step: 1
test_start: 1
test_step: 1
save_dir: ./imdb_ckpt
measure_best: accuracy
measure_mode: max
checkpoint_latest: True
checkpoint_list: [10, 15]
resume_path: None
}
world_size: 1
```

```
2023-12-07 11:26:27 - INFO - ----- Start of training. Good day! -----
100%|██████████| 32/32 [00:05<00:00, 5.75it/s]
100%|██████████| 32/32 [00:05<00:00, 5.83it/s]
100%|██████████| 32/32 [00:05<00:00, 5.76it/s]
100%|██████████| 32/32 [00:05<00:00, 5.73it/s]
2023-12-07 11:26:50 - INFO - [GPU0] | Epoch 1/20 | Train loss: 0.7045852541923523 | Valid scores: accuracy: 0.507 | Valid scores: accuracy: 0.507 | Time/epoch: 22.2024 seconds
2023-12-07 11:26:50 - INFO - New best model: valid accuracy update from -inf to 0.507
2023-12-07 11:26:50 - INFO - Saving best model: ./imdb_ckpt/run@231207_11:26:27/best_model_epoch1.pth ...
100%|██████████| 32/32 [00:05<00:00, 6.16it/s]
100%|██████████| 32/32 [00:04<00:00, 6.41it/s]
100%|██████████| 32/32 [00:05<00:00, 5.93it/s]
100%|██████████| 32/32 [00:06<00:00, 5.30it/s]
2023-12-07 11:27:11 - INFO - [GPU0] | Epoch 2/20 | Train loss: 0.7534643411636353 | Valid scores: accuracy: 0.507 | Valid scores: accuracy: 0.507 | Time/epoch: 21.62994 seconds
100%|██████████| 32/32 [00:05<00:00, 6.10it/s]
100%|██████████| 32/32 [00:05<00:00, 5.49it/s]
```

```
chen@chen-ubuntu:~/Workspace/pytorch_template/example_imdb
```

```
2023-12-07 11:30:52 - INFO - New best model: valid accuracy update from 0.619 to 0.65
2023-12-07 11:30:52 - INFO - Saving best model: ./imdb_ckpt/run@231207_11:26:27/best_model_epoch12.pth ...
100%|██████████| 32/32 [00:05<00:00, 5.67it/s]
100%|██████████| 32/32 [00:05<00:00, 6.22it/s]
100%|██████████| 32/32 [00:06<00:00, 4.81it/s]
100%|██████████| 32/32 [00:05<00:00, 5.33it/s]
2023-12-07 11:31:15 - INFO - [GPU0] | Epoch 13/20 | Train loss: 0.6626368165016174 | Valid scores: accuracy: 0.719 | Valid scores: accuracy: 0.644 | Time/epoch: 23.4461 seconds
100%|██████████| 32/32 [00:08<00:00, 3.78it/s]
100%|██████████| 32/32 [00:10<00:00, 3.07it/s]
100%|██████████| 32/32 [00:08<00:00, 3.98it/s]
100%|██████████| 32/32 [00:08<00:00, 3.73it/s]
2023-12-07 11:31:51 - INFO - [GPU0] | Epoch 14/20 | Train loss: 0.652633547782898 | Valid scores: accuracy: 0.746 | Valid scores: accuracy: 0.67 | Time/epoch: 35.50088 seconds
2023-12-07 11:31:51 - INFO - New best model: valid accuracy update from 0.65 to 0.67
2023-12-07 11:31:51 - INFO - Saving best model: ./imdb_ckpt/run@231207_11:26:27/best_model_epoch14.pth ...
100%|██████████| 32/32 [00:10<00:00, 2.92it/s]
100%|██████████| 32/32 [00:08<00:00, 3.88it/s]
100%|██████████| 32/32 [00:07<00:00, 4.16it/s]
100%|██████████| 32/32 [00:08<00:00, 3.90it/s]
2023-12-07 11:32:26 - INFO - [GPU0] | Epoch 15/20 | Train loss: 0.6583216190338135 | Valid scores: accuracy: 0.751 | Valid scores: accuracy: 0.666 | Time/epoch: 35.14059 seconds
2023-12-07 11:32:26 - INFO - Saving checkpoint: ./imdb_ckpt/run@231207_11:26:27/checkpoint_epoch15.pth ...
100%|██████████| 32/32 [00:08<00:00, 3.85it/s]
100%|██████████| 32/32 [00:08<00:00, 3.58it/s]
100%|██████████| 32/32 [00:08<00:00, 3.64it/s]
100%|██████████| 32/32 [00:09<00:00, 3.21it/s]
2023-12-07 11:33:02 - INFO - [GPU0] | Epoch 16/20 | Train loss: 0.6219320297241211 | Valid scores: accuracy: 0.59 | Valid scores: accuracy: 0.561 | Time/epoch: 36.00468 seconds
100%|██████████| 32/32 [00:09<00:00, 3.30it/s]
100%|██████████| 32/32 [00:07<00:00, 4.12it/s]
100%|██████████| 32/32 [00:09<00:00, 4.32it/s]
```

```
chen@chen-ubuntu:~/Workspace/pytorch_template/example_imdb
```

```
2023-12-07 11:34:14 - INFO - [GPU0] | Epoch 18/20 | Train loss: 0.6756136417388916 | Valid scores: accuracy: 0.65 | Valid scores: accuracy: 0.601 | Time/epoch: 35.29723 seconds
100%|██████████| 32/32 [00:09<00:00, 3.54it/s]
100%|██████████| 32/32 [00:07<00:00, 4.24it/s]
100%|██████████| 32/32 [00:07<00:00, 4.18it/s]
100%|██████████| 32/32 [00:07<00:00, 4.16it/s]
2023-12-07 11:34:46 - INFO - [GPU0] | Epoch 19/20 | Train loss: 0.6119271516799927 | Valid scores: accuracy: 0.648 | Valid scores: accuracy: 0.596 | Time/epoch: 31.95647 seconds
100%|██████████| 32/32 [00:08<00:00, 3.59it/s]
100%|██████████| 32/32 [00:09<00:00, 3.47it/s]
100%|██████████| 32/32 [00:07<00:00, 4.10it/s]
100%|██████████| 32/32 [00:07<00:00, 4.03it/s]
2023-12-07 11:35:20 - INFO - [GPU0] | Epoch 20/20 | Train loss: 0.6392252445220947 | Valid scores: accuracy: 0.785 | Valid scores: accuracy: 0.702 | Time/epoch: 33.87659 seconds
2023-12-07 11:35:20 - INFO - New best model: valid accuracy update from 0.671 to 0.702
2023-12-07 11:35:20 - INFO - Saving best model: ./imdb_ckpt/run@231207_11:26:27/best_model_epoch20.pth ...
2023-12-07 11:35:20 - INFO - ----- End of training. Total time: 532.45566 seconds -----
wandb: Waiting for W&B process to finish... (success).
wandb:
wandb: Run history:
wandb:      epoch ██████████
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 20
wandb: eval/best_valid_accuracy 0.702
wandb:   eval/train_accuracy 0.785
```

wandb:

Overview

Charts

System

Logs

Files

nehc0 > Projects > IMDb > Runs > toasty-sponge-5 > Overview

toasty-sponge-5

training details on the process of global rank 0

PRIVATE

IMDb dataset x TransformerEncoder +

nehc0

Running

December 7th, 2023 at 11:26:16 am

6m 10s

nehc0/IMDb/ojzx3gkk

chen-ubuntu

Linux-6.3.13-060313-generic-x86\_64-with-glibc2.35

3.11.5

/home/chen/anaconda3/envs/dl\_pytorch/bin/python

/home/chen/Workspace/pytorch\_template/example\_imdb/./main.py

CPU count 16

GPU count 1

NVIDIA GeForce RTX 3080 Ti Laptop GPU

W&B CLI Version 0.15.12

Configure

View raw data

Summary

Config parameters describe your model's inputs. Learn more

Summary metrics describe your results. Learn more

Search keys

Key

Value

loader\_cfg

batch\_size 32

batch\_size\_per\_proc 32

effective\_batch\_size 32

num\_workers 8

pin\_memory true

model\_cfg

dim\_feedforward 512

dropout 0.1

embed\_dim 128

nhead 2

num\_classes 2

num\_layers 2

vocab\_size 10,000

optimizer\_cfg

lr 0.0005

Search keys

Key

Value

epoch 15

eval

best\_valid\_accuracy 0.67

train\_accuracy 0.751

valid\_accuracy 0.666

valid\_loss 0.6401496529579163

train

epoch\_time 35.140588998794556

lr 0.0005

train\_loss 0.6583216190338135



saved logs and checkpoints:

