trainer_2MSwithunknown_bert_proselflc

November 30, 2022

0.1 Import dependencies

[1]:

```
import os
import time

import pandas
import torch

# The trainer implementation is slightly different
# as we are using bert transformer models
from proselflc.trainer.trainer_cnn_vision_derivedgrad_adaptedfordeeploc import␣
  ↪Trainer

import pprint
print = pprint.PrettyPrinter(indent=4).pprint

%env CUDA_VISIBLE_DEVICES=0
%env CUBLAS_WORKSPACE_CONFIG=:4096:8
%env TOKENIZERS_PARALLELISM=false
```

```
env: CUDA_VISIBLE_DEVICES=0
env: CUBLAS_WORKSPACE_CONFIG=:4096:8
env: TOKENIZERS_PARALLELISM=false
```

0.2 Add params configurations

[2]:

```
params = {}
params.update(
    {
        "data_name": "deeploc_prottrans",
        "num_classes": 2,
        "task_name": "MS-with-unknown",
        #
        "device": "gpu",
        "num_workers": 1,
        #
        # network
        "network_name": "Rostlab_prot_bert_bfd_seq",
        "num_hidden_layers": 2, # only for demo, in our paper, it is 6
        "num_attention_heads": 8, # only for demo, in our paper, it is 16
        #
        "counter": "iteration",
        #
        "batch_size": 4, # only for demo, in our paper, it is 32
        "classes_per_batch": 2,  # at most 2
        #
        "max_seq_length": 434, # smaller -> more noise
        #
        #
        "seed": 123,
    }
)

# data
params["symmetric_noise_rate"] = 0.0  # placeholder only

# for the demo purpose, I set the total epochs to be small.
params["total_epochs"] = 40
params["eval_interval"] = 100 # iterations

# batch
params["sampler"] = "BalancedBatchSampler"

# learning rate and optimisation
params["lr"] = 0.01
params["weight_decay"] = 1e-3
params["lr_scheduler"] = "WarmupMultiStepSchedule"
params["warmup_epochs"] = 0
#
params["momentum"] = 0.9
params["batch_accumu_steps"] = 10
params["milestones"] = [30000]
params["gamma"] = 0.1

# loss settings
params["loss_mode"] = "cross entropy"
params["trust_mode"] = "global*(1-H(p)/H(u))"
params["loss_name"] = "proselflc"
params["transit_time_ratio"] = 0.50
params["exp_base"] = 12
params["logit_soften_T"] = 0.2

print(params)
```

```
{   'batch_accumu_steps': 10,
 'batch_size': 4,
 'classes_per_batch': 2,
 'counter': 'iteration',
 'data_name': 'deeploc_prottrans',
 'device': 'gpu',
 'eval_interval': 100,
 'exp_base': 12,
 'gamma': 0.1,
 'logit_soften_T': 0.2,
 'loss_mode': 'cross entropy',
 'loss_name': 'proselflc',
 'lr': 0.01,
 'lr_scheduler': 'WarmupMultiStepSchedule',
 'max_seq_length': 434,
 'milestones': [30000],
 'momentum': 0.9,
 'network_name': 'Rostlab_prot_bert_bfd_seq',
 'num_attention_heads': 8,
 'num_classes': 2,
 'num_hidden_layers': 2,
 'num_workers': 1,
 'sampler': 'BalancedBatchSampler',
 'seed': 123,
 'symmetric_noise_rate': 0.0,
 'task_name': 'MS-with-unknown',
 'total_epochs': 40,
 'transit_time_ratio': 0.5,
 'trust_mode': 'global*(1-H(p)/H(u))',
 'warmup_epochs': 0,
 'weight_decay': 0.001}
```

0.3 Create the folder to store intermediate and final results

- First, run `sudo mkdir /home/proselflc_experiments/ && sudo chmod -R 777 /home/proselflc_experiments/` so that so you have the write permission.
- Or set `WORK_DIR = "/home/your_username/proselflc_experiments/"`

[3]:

```
WORK_DIR = "/home/proselflc_experiments/"
#
# use the time as a unique experiment identifier
dt_string = time.strftime("%Y%m%d-%H%M%S")
summary_writer_dir = (
    params["loss_name"]
    + "_"
    + dt_string
)
params["summary_writer_dir"] = (
    WORK_DIR
    + "/"
    + params["data_name"]
    + "_symmetric_noise_rate_"
    + str(params["symmetric_noise_rate"])
    + "/"
    + params["network_name"]
    + "/"
    + summary_writer_dir
)
if not os.path.exists(params["summary_writer_dir"]):
    os.makedirs(params["summary_writer_dir"])
```

0.4 Init the trainer

[4]:

```
trainer = Trainer(params=params)
```

```
Some weights of the model checkpoint at Rostlab/prot_bert_bfd were not used when
initializing BertForSequenceClassification:
['bert.encoder.layer.7.output.LayerNorm.bias',
'bert.encoder.layer.27.output.LayerNorm.bias',
'bert.encoder.layer.24.attention.output.dense.bias',
'bert.encoder.layer.13.attention.output.LayerNorm.bias',
'bert.encoder.layer.18.attention.output.dense.bias',
'bert.encoder.layer.28.attention.self.query.weight',
'bert.encoder.layer.26.intermediate.dense.bias',
'bert.encoder.layer.13.attention.output.LayerNorm.weight',
'bert.encoder.layer.19.attention.self.value.bias',
'bert.encoder.layer.14.output.dense.bias',
'bert.encoder.layer.5.attention.output.dense.weight',
'bert.encoder.layer.5.attention.output.dense.bias',
'bert.encoder.layer.16.output.dense.bias',
'bert.encoder.layer.18.output.LayerNorm.bias',
'bert.encoder.layer.28.attention.self.query.bias',
'bert.encoder.layer.7.attention.self.query.weight',
'bert.encoder.layer.18.intermediate.dense.bias',
'bert.encoder.layer.5.attention.self.value.weight',
'bert.encoder.layer.21.attention.self.value.weight',
'bert.encoder.layer.23.output.LayerNorm.bias',
'bert.encoder.layer.9.intermediate.dense.weight',
'bert.encoder.layer.2.intermediate.dense.bias',
'bert.encoder.layer.16.attention.self.value.bias',
'bert.encoder.layer.28.attention.output.dense.weight',
'bert.encoder.layer.4.attention.self.query.bias',
'bert.encoder.layer.9.attention.self.value.bias',
'bert.encoder.layer.3.attention.output.LayerNorm.bias',
'bert.encoder.layer.25.intermediate.dense.weight',
'bert.encoder.layer.7.intermediate.dense.weight',
'bert.encoder.layer.16.attention.self.query.bias',
'bert.encoder.layer.25.output.dense.weight',
'bert.encoder.layer.20.attention.output.dense.weight',
'bert.encoder.layer.22.attention.self.value.bias',
'bert.encoder.layer.24.attention.self.key.weight',
'bert.encoder.layer.9.attention.self.key.weight',
'bert.encoder.layer.14.attention.output.dense.weight',
'cls.seq_relationship.weight', 'bert.encoder.layer.10.output.dense.bias',
'bert.encoder.layer.17.output.dense.weight',
'bert.encoder.layer.10.attention.output.dense.bias',
'bert.encoder.layer.12.intermediate.dense.weight',
'bert.encoder.layer.8.attention.self.value.bias',
'bert.encoder.layer.24.output.LayerNorm.weight',
'bert.encoder.layer.2.attention.output.LayerNorm.weight',
'bert.encoder.layer.16.intermediate.dense.bias',
'bert.encoder.layer.21.attention.self.value.bias',
'bert.encoder.layer.29.attention.self.value.bias',
'bert.encoder.layer.26.attention.self.query.weight',
'bert.encoder.layer.3.output.dense.weight',
'bert.encoder.layer.22.intermediate.dense.bias',
'bert.encoder.layer.26.attention.self.query.bias',
'bert.encoder.layer.5.attention.self.query.weight',
'bert.encoder.layer.24.attention.self.query.bias',
'bert.encoder.layer.10.intermediate.dense.bias',
'bert.encoder.layer.26.intermediate.dense.weight',
'bert.encoder.layer.15.attention.output.dense.weight',
'bert.encoder.layer.12.attention.self.query.weight',
'bert.encoder.layer.15.attention.output.LayerNorm.bias',
'bert.encoder.layer.20.attention.output.LayerNorm.weight',
'bert.encoder.layer.20.intermediate.dense.bias',
'bert.encoder.layer.27.attention.output.dense.bias',
'bert.encoder.layer.11.attention.self.key.bias',
'bert.encoder.layer.6.attention.output.dense.bias',
'bert.encoder.layer.8.output.LayerNorm.weight',
'bert.encoder.layer.2.attention.self.query.bias',
'bert.encoder.layer.6.attention.self.query.weight',
'bert.encoder.layer.23.attention.output.dense.bias',
'bert.encoder.layer.8.attention.output.dense.weight',
'bert.encoder.layer.5.intermediate.dense.weight',
'bert.encoder.layer.21.intermediate.dense.bias',
'bert.encoder.layer.19.output.dense.weight',
'bert.encoder.layer.12.output.LayerNorm.bias',
'bert.encoder.layer.15.attention.self.key.weight',
'bert.encoder.layer.4.attention.output.dense.weight',
'bert.encoder.layer.24.attention.self.value.weight',
'bert.encoder.layer.25.attention.self.value.weight',
'bert.encoder.layer.25.attention.self.value.bias',
'bert.encoder.layer.6.attention.output.LayerNorm.weight',
'bert.encoder.layer.16.attention.output.dense.weight',
'bert.encoder.layer.19.attention.self.key.weight',
'bert.encoder.layer.22.intermediate.dense.weight',
'bert.encoder.layer.13.output.LayerNorm.weight',
'bert.encoder.layer.17.attention.output.LayerNorm.weight',
'bert.encoder.layer.26.attention.output.dense.weight',
'bert.encoder.layer.15.attention.output.dense.bias',
'bert.encoder.layer.8.output.dense.weight',
'bert.encoder.layer.23.attention.self.key.bias',
'bert.encoder.layer.4.intermediate.dense.weight',
'bert.encoder.layer.20.attention.output.dense.bias',
'bert.encoder.layer.2.attention.self.key.weight',
'bert.encoder.layer.9.output.dense.weight',
'bert.encoder.layer.15.attention.self.key.bias',
'bert.encoder.layer.29.output.dense.weight',
'bert.encoder.layer.29.attention.self.key.bias',
'bert.encoder.layer.4.attention.self.query.weight',
'bert.encoder.layer.3.attention.self.key.weight',
'bert.encoder.layer.19.attention.self.value.weight',
'bert.encoder.layer.28.attention.output.dense.bias',
'bert.encoder.layer.9.attention.output.LayerNorm.bias',
'bert.encoder.layer.10.attention.self.value.bias',
'bert.encoder.layer.29.output.LayerNorm.weight',
'bert.encoder.layer.14.attention.self.key.bias',
'bert.encoder.layer.25.attention.self.key.weight',
'bert.encoder.layer.27.intermediate.dense.bias',
'bert.encoder.layer.13.output.dense.weight',
'bert.encoder.layer.5.attention.self.key.weight',
'bert.encoder.layer.8.attention.output.dense.bias',
'bert.encoder.layer.6.output.LayerNorm.weight',
'bert.encoder.layer.18.output.dense.bias',
'bert.encoder.layer.13.attention.output.dense.weight',
'bert.encoder.layer.19.intermediate.dense.bias',
'bert.encoder.layer.12.attention.self.value.weight',
'bert.encoder.layer.6.attention.self.query.bias',
'bert.encoder.layer.6.output.dense.weight',
'bert.encoder.layer.25.attention.self.query.weight',
'bert.encoder.layer.11.attention.output.dense.weight',
'bert.encoder.layer.2.attention.output.dense.bias',
'bert.encoder.layer.15.attention.self.value.bias',
'bert.encoder.layer.14.intermediate.dense.weight',
'bert.encoder.layer.26.attention.self.key.bias',
'bert.encoder.layer.9.attention.self.query.weight',
'bert.encoder.layer.14.intermediate.dense.bias',
'bert.encoder.layer.23.intermediate.dense.bias',
'bert.encoder.layer.20.attention.self.value.weight',
'bert.encoder.layer.25.intermediate.dense.bias',
'bert.encoder.layer.25.output.dense.bias',
'bert.encoder.layer.4.attention.self.value.bias',
'bert.encoder.layer.19.intermediate.dense.weight',
'bert.encoder.layer.14.attention.self.key.weight',
'bert.encoder.layer.26.output.dense.bias',
'bert.encoder.layer.9.attention.self.query.bias',
'bert.encoder.layer.10.attention.output.LayerNorm.weight',
'bert.encoder.layer.5.output.dense.bias',
'bert.encoder.layer.19.attention.self.query.weight',
'bert.encoder.layer.9.attention.self.value.weight',
'bert.encoder.layer.15.intermediate.dense.weight',
'bert.encoder.layer.2.output.dense.bias',
'bert.encoder.layer.19.output.LayerNorm.bias',
'bert.encoder.layer.7.attention.self.key.weight',
'bert.encoder.layer.27.output.dense.bias',
'bert.encoder.layer.13.attention.self.query.weight',
'bert.encoder.layer.24.output.LayerNorm.bias',
'bert.encoder.layer.22.attention.self.query.weight',
'bert.encoder.layer.16.attention.self.key.bias',
'bert.encoder.layer.7.attention.output.LayerNorm.bias',
'cls.predictions.decoder.bias', 'bert.encoder.layer.11.output.LayerNorm.bias',
'bert.encoder.layer.7.output.dense.bias',
'bert.encoder.layer.7.output.LayerNorm.weight',
'bert.encoder.layer.5.attention.self.key.bias',
'bert.encoder.layer.8.output.dense.bias',
'bert.encoder.layer.3.attention.self.key.bias',
'bert.encoder.layer.11.attention.output.dense.bias',
'bert.encoder.layer.6.intermediate.dense.bias',
'bert.encoder.layer.15.attention.output.LayerNorm.weight',
'bert.encoder.layer.26.attention.output.LayerNorm.bias',
'bert.encoder.layer.8.attention.self.query.bias',
'bert.encoder.layer.18.attention.output.LayerNorm.weight',
'bert.encoder.layer.2.output.LayerNorm.weight',
'bert.encoder.layer.24.intermediate.dense.bias',
'bert.encoder.layer.28.attention.output.LayerNorm.bias',
'bert.encoder.layer.22.attention.self.query.bias',
'bert.encoder.layer.26.attention.self.value.bias', 'cls.seq_relationship.bias',
'bert.encoder.layer.8.attention.output.LayerNorm.bias',
'bert.encoder.layer.16.output.LayerNorm.weight',
'bert.encoder.layer.11.intermediate.dense.weight',
'bert.encoder.layer.29.output.LayerNorm.bias',
'bert.encoder.layer.29.attention.output.LayerNorm.weight',
'bert.encoder.layer.4.attention.output.LayerNorm.bias',
'bert.encoder.layer.21.attention.output.LayerNorm.bias',
'bert.encoder.layer.18.attention.self.value.weight',
'bert.encoder.layer.5.output.dense.weight',
'bert.encoder.layer.11.attention.output.LayerNorm.weight',
'bert.encoder.layer.4.attention.self.key.weight',
'bert.encoder.layer.29.attention.self.key.weight',
'bert.encoder.layer.8.attention.self.query.weight',
'bert.encoder.layer.15.output.LayerNorm.bias',
'bert.encoder.layer.6.intermediate.dense.weight',
'bert.encoder.layer.29.attention.output.LayerNorm.bias',
'bert.encoder.layer.6.output.LayerNorm.bias',
'bert.encoder.layer.18.attention.self.query.weight',
'bert.encoder.layer.23.output.dense.bias',
'bert.encoder.layer.21.attention.self.key.bias',
'bert.encoder.layer.13.attention.self.key.weight',
'bert.encoder.layer.27.attention.output.LayerNorm.bias',
'bert.encoder.layer.17.attention.self.query.bias',
'bert.encoder.layer.27.intermediate.dense.weight',
'bert.encoder.layer.5.output.LayerNorm.bias',
'bert.encoder.layer.11.output.dense.bias',
'cls.predictions.transform.dense.bias',
'bert.encoder.layer.7.attention.self.value.bias',
'bert.encoder.layer.28.attention.output.LayerNorm.weight',
'bert.encoder.layer.8.intermediate.dense.bias',
'bert.encoder.layer.9.intermediate.dense.bias',
'bert.encoder.layer.12.attention.self.key.weight',
'bert.encoder.layer.2.attention.self.value.weight',
'bert.encoder.layer.19.attention.self.query.bias',
'bert.encoder.layer.10.attention.output.dense.weight',
'bert.encoder.layer.24.attention.self.value.bias',
'bert.encoder.layer.28.intermediate.dense.bias',
'bert.encoder.layer.3.output.dense.bias',
'bert.encoder.layer.12.output.LayerNorm.weight',
'bert.encoder.layer.7.attention.self.key.bias',
'bert.encoder.layer.11.attention.self.query.bias',
'bert.encoder.layer.28.output.LayerNorm.weight',
'bert.encoder.layer.22.output.LayerNorm.bias',
'bert.encoder.layer.18.intermediate.dense.weight',
'bert.encoder.layer.2.attention.self.value.bias',
'bert.encoder.layer.19.attention.self.key.bias',
'bert.encoder.layer.22.attention.output.LayerNorm.bias',
'cls.predictions.decoder.weight',
'bert.encoder.layer.7.attention.self.value.weight',
'bert.encoder.layer.15.output.LayerNorm.weight', 'cls.predictions.bias',
'bert.encoder.layer.13.attention.self.query.bias',
'bert.encoder.layer.11.output.LayerNorm.weight',
'bert.encoder.layer.27.attention.self.value.weight',
'bert.encoder.layer.10.output.dense.weight',
'bert.encoder.layer.2.output.dense.weight',
'bert.encoder.layer.3.output.LayerNorm.weight',
'bert.encoder.layer.15.output.dense.bias',
'bert.encoder.layer.20.intermediate.dense.weight',
'bert.encoder.layer.3.attention.self.value.bias',
'bert.encoder.layer.8.output.LayerNorm.bias',
'bert.encoder.layer.19.attention.output.LayerNorm.weight',
'bert.encoder.layer.21.attention.output.dense.weight',
'bert.encoder.layer.20.output.LayerNorm.bias',
'bert.encoder.layer.21.attention.output.dense.bias',
'bert.encoder.layer.26.attention.self.value.weight',
'bert.encoder.layer.18.attention.output.dense.weight',
'bert.encoder.layer.23.attention.output.LayerNorm.weight',
'bert.encoder.layer.9.output.dense.bias',
'bert.encoder.layer.4.attention.self.key.bias',
'bert.encoder.layer.19.attention.output.dense.bias',
'cls.predictions.transform.LayerNorm.weight',
'bert.encoder.layer.20.attention.self.value.bias',
'bert.encoder.layer.9.attention.output.LayerNorm.weight',
'bert.encoder.layer.7.attention.output.dense.bias',
'bert.encoder.layer.16.intermediate.dense.weight',
'bert.encoder.layer.7.attention.self.query.bias',
'bert.encoder.layer.3.attention.self.query.bias',
'bert.encoder.layer.10.output.LayerNorm.bias',
'bert.encoder.layer.23.attention.self.key.weight',
'bert.encoder.layer.29.intermediate.dense.weight',
'bert.encoder.layer.29.intermediate.dense.bias',
'bert.encoder.layer.28.output.dense.weight',
'bert.encoder.layer.6.attention.self.key.weight',
'bert.encoder.layer.22.attention.self.value.weight',
'bert.encoder.layer.24.output.dense.bias',
'bert.encoder.layer.10.attention.output.LayerNorm.bias',
'bert.encoder.layer.17.attention.output.dense.weight',
'bert.encoder.layer.27.attention.output.LayerNorm.weight',
'bert.encoder.layer.17.attention.self.value.weight',
'bert.encoder.layer.29.attention.self.query.bias',
'bert.encoder.layer.11.attention.self.value.bias',
'bert.encoder.layer.6.attention.self.value.weight',
'bert.encoder.layer.14.output.LayerNorm.weight',
'bert.encoder.layer.26.output.dense.weight',
'bert.encoder.layer.10.intermediate.dense.weight',
'bert.encoder.layer.14.attention.self.value.bias',
'bert.encoder.layer.4.attention.output.dense.bias',
'bert.encoder.layer.20.attention.self.query.weight',
'bert.encoder.layer.28.attention.self.value.weight',
'bert.encoder.layer.11.attention.self.value.weight',
'bert.encoder.layer.3.intermediate.dense.weight',
'bert.encoder.layer.27.attention.self.key.bias',
'bert.encoder.layer.14.output.LayerNorm.bias',
'bert.encoder.layer.11.output.dense.weight',
'bert.encoder.layer.27.output.dense.weight',
'bert.encoder.layer.15.attention.self.value.weight',
'bert.encoder.layer.18.attention.self.key.bias',
'bert.encoder.layer.7.attention.output.LayerNorm.weight',
'bert.encoder.layer.18.attention.output.LayerNorm.bias',
'bert.encoder.layer.23.intermediate.dense.weight',
'bert.encoder.layer.8.attention.self.key.weight',
'bert.encoder.layer.3.attention.self.value.weight',
'bert.encoder.layer.3.attention.output.dense.bias',
'bert.encoder.layer.5.intermediate.dense.bias',
'bert.encoder.layer.9.attention.output.dense.bias',
'bert.encoder.layer.22.attention.output.LayerNorm.weight',
'bert.encoder.layer.24.output.dense.weight',
'bert.encoder.layer.25.attention.output.LayerNorm.bias',
'bert.encoder.layer.29.attention.output.dense.weight',
'bert.encoder.layer.8.attention.output.LayerNorm.weight',
'bert.encoder.layer.28.output.LayerNorm.bias',
'bert.encoder.layer.29.attention.self.query.weight',
'bert.encoder.layer.2.attention.self.key.bias',
'bert.encoder.layer.24.attention.output.dense.weight',
'bert.encoder.layer.12.intermediate.dense.bias',
'bert.encoder.layer.10.attention.self.key.bias',
'bert.encoder.layer.4.attention.output.LayerNorm.weight',
'bert.encoder.layer.5.attention.self.value.bias',
'bert.encoder.layer.19.output.dense.bias',
'bert.encoder.layer.21.intermediate.dense.weight',
'bert.encoder.layer.6.attention.output.dense.weight',
'bert.encoder.layer.16.attention.self.query.weight',
'bert.encoder.layer.11.attention.self.query.weight',
'bert.encoder.layer.14.attention.self.query.weight',
'bert.encoder.layer.12.attention.self.key.bias',
'bert.encoder.layer.12.output.dense.weight',
'bert.encoder.layer.22.attention.output.dense.weight',
'cls.predictions.transform.dense.weight',
'bert.encoder.layer.29.attention.self.value.weight',
'bert.encoder.layer.2.attention.output.LayerNorm.bias',
'bert.encoder.layer.23.attention.self.query.bias',
'bert.encoder.layer.5.attention.self.query.bias',
'bert.encoder.layer.11.attention.output.LayerNorm.bias',
'bert.encoder.layer.23.attention.self.query.weight',
'bert.encoder.layer.20.attention.output.LayerNorm.bias',
'bert.encoder.layer.24.attention.self.key.bias',
'bert.encoder.layer.13.attention.self.key.bias',
'bert.encoder.layer.21.output.LayerNorm.weight',
'bert.encoder.layer.28.intermediate.dense.weight',
'bert.encoder.layer.13.attention.self.value.bias',
'bert.encoder.layer.16.attention.output.LayerNorm.weight',
'bert.encoder.layer.5.output.LayerNorm.weight',
'bert.encoder.layer.20.output.dense.weight',
'bert.encoder.layer.17.attention.output.dense.bias',
'bert.encoder.layer.22.output.dense.weight',
'bert.encoder.layer.17.output.LayerNorm.bias',
'bert.encoder.layer.24.attention.self.query.weight',
'bert.encoder.layer.17.output.LayerNorm.weight',
'bert.encoder.layer.27.attention.output.dense.weight',
'bert.encoder.layer.21.attention.output.LayerNorm.weight',
'bert.encoder.layer.23.attention.output.LayerNorm.bias',
'bert.encoder.layer.21.attention.self.query.weight',
'bert.encoder.layer.4.output.LayerNorm.bias',
'bert.encoder.layer.22.output.dense.bias',
'bert.encoder.layer.27.output.LayerNorm.weight',
'bert.encoder.layer.14.attention.self.value.weight',
'bert.encoder.layer.13.intermediate.dense.bias',
'cls.predictions.transform.LayerNorm.bias',
'bert.encoder.layer.18.attention.self.key.weight',
'bert.encoder.layer.28.output.dense.bias',
'bert.encoder.layer.20.output.dense.bias',
'bert.encoder.layer.10.attention.self.value.weight',
'bert.encoder.layer.17.output.dense.bias',
'bert.encoder.layer.17.intermediate.dense.bias',
'bert.encoder.layer.17.attention.self.key.bias',
'bert.encoder.layer.12.attention.output.dense.weight',
'bert.encoder.layer.12.output.dense.bias',
'bert.encoder.layer.29.attention.output.dense.bias',
'bert.encoder.layer.6.attention.self.value.bias',
'bert.encoder.layer.15.attention.self.query.bias',
'bert.encoder.layer.6.attention.self.key.bias',
'bert.encoder.layer.12.attention.self.value.bias',
'bert.encoder.layer.17.attention.self.value.bias',
'bert.encoder.layer.3.attention.output.dense.weight',
'bert.encoder.layer.23.attention.self.value.weight',
'bert.encoder.layer.7.attention.output.dense.weight',
'bert.encoder.layer.23.attention.self.value.bias',
'bert.encoder.layer.23.output.dense.weight',
'bert.encoder.layer.27.attention.self.query.weight',
'bert.encoder.layer.15.attention.self.query.weight',
'bert.encoder.layer.18.output.dense.weight',
'bert.encoder.layer.24.attention.output.LayerNorm.bias',
'bert.encoder.layer.10.attention.self.query.bias',
'bert.encoder.layer.14.attention.self.query.bias',
'bert.encoder.layer.16.attention.output.dense.bias',
'bert.encoder.layer.14.attention.output.LayerNorm.bias',
'bert.encoder.layer.14.output.dense.weight',
'bert.encoder.layer.25.attention.self.key.bias',
'bert.encoder.layer.13.intermediate.dense.weight',
'bert.encoder.layer.20.attention.self.key.bias',
'bert.encoder.layer.26.attention.output.LayerNorm.weight',
'bert.encoder.layer.21.output.dense.weight',
'bert.encoder.layer.5.attention.output.LayerNorm.bias',
'bert.encoder.layer.8.intermediate.dense.weight',
'bert.encoder.layer.25.output.LayerNorm.bias',
'bert.encoder.layer.16.attention.output.LayerNorm.bias',
'bert.encoder.layer.12.attention.output.dense.bias',
'bert.encoder.layer.3.attention.self.query.weight',
'bert.encoder.layer.25.attention.output.dense.bias',
'bert.encoder.layer.19.attention.output.LayerNorm.bias',
'bert.encoder.layer.3.attention.output.LayerNorm.weight',
'bert.encoder.layer.28.attention.self.key.weight',
'bert.encoder.layer.8.attention.self.key.bias',
'bert.encoder.layer.18.attention.self.value.bias',
'bert.encoder.layer.2.attention.self.query.weight',
'bert.encoder.layer.21.output.LayerNorm.bias',
'bert.encoder.layer.22.output.LayerNorm.weight',
'bert.encoder.layer.15.intermediate.dense.bias',
'bert.encoder.layer.22.attention.self.key.weight',
'bert.encoder.layer.23.attention.output.dense.weight',
'bert.encoder.layer.26.attention.self.key.weight',
'bert.encoder.layer.18.output.LayerNorm.weight',
'bert.encoder.layer.7.output.dense.weight',
'bert.encoder.layer.6.attention.output.LayerNorm.bias',
'bert.encoder.layer.27.attention.self.query.bias',
'bert.encoder.layer.10.output.LayerNorm.weight',
'bert.encoder.layer.15.output.dense.weight',
'bert.encoder.layer.28.attention.self.value.bias',
'bert.encoder.layer.27.attention.self.value.bias',
'bert.encoder.layer.24.attention.output.LayerNorm.weight',
'bert.encoder.layer.20.output.LayerNorm.weight',
'bert.encoder.layer.26.output.LayerNorm.bias',
'bert.encoder.layer.21.attention.self.key.weight',
'bert.encoder.layer.16.attention.self.key.weight',
'bert.encoder.layer.2.output.LayerNorm.bias',
'bert.encoder.layer.13.attention.self.value.weight',
'bert.encoder.layer.17.attention.self.query.weight',
'bert.encoder.layer.4.attention.self.value.weight',
'bert.encoder.layer.7.intermediate.dense.bias',
'bert.encoder.layer.25.attention.self.query.bias',
'bert.encoder.layer.11.attention.self.key.weight',
'bert.encoder.layer.14.attention.output.dense.bias',
'bert.encoder.layer.8.attention.self.value.weight',
'bert.encoder.layer.4.output.dense.weight',
'bert.encoder.layer.16.output.dense.weight',
'bert.encoder.layer.13.attention.output.dense.bias',
'bert.encoder.layer.17.attention.self.key.weight',
'bert.encoder.layer.11.intermediate.dense.bias',
'bert.encoder.layer.22.attention.self.key.bias',
'bert.encoder.layer.16.output.LayerNorm.bias',
'bert.encoder.layer.27.attention.self.key.weight',
'bert.encoder.layer.10.attention.self.query.weight',
'bert.encoder.layer.17.attention.output.LayerNorm.bias',
'bert.encoder.layer.26.attention.output.dense.bias',
'bert.encoder.layer.9.output.LayerNorm.weight',
'bert.encoder.layer.21.attention.self.query.bias',
'bert.encoder.layer.25.attention.output.LayerNorm.weight',
'bert.encoder.layer.2.intermediate.dense.weight',
'bert.encoder.layer.10.attention.self.key.weight',
'bert.encoder.layer.20.attention.self.key.weight',
'bert.encoder.layer.2.attention.output.dense.weight',
'bert.encoder.layer.13.output.dense.bias',
'bert.encoder.layer.25.attention.output.dense.weight',
'bert.encoder.layer.25.output.LayerNorm.weight',
'bert.encoder.layer.5.attention.output.LayerNorm.weight',
'bert.encoder.layer.19.attention.output.dense.weight',
'bert.encoder.layer.4.output.dense.bias',
'bert.encoder.layer.26.output.LayerNorm.weight',
'bert.encoder.layer.12.attention.output.LayerNorm.weight',
'bert.encoder.layer.21.output.dense.bias',
'bert.encoder.layer.9.output.LayerNorm.bias',
'bert.encoder.layer.20.attention.self.query.bias',
'bert.encoder.layer.22.attention.output.dense.bias',
'bert.encoder.layer.3.intermediate.dense.bias',
'bert.encoder.layer.14.attention.output.LayerNorm.weight',
'bert.encoder.layer.16.attention.self.value.weight',
'bert.encoder.layer.9.attention.output.dense.weight',
'bert.encoder.layer.24.intermediate.dense.weight',
'bert.encoder.layer.6.output.dense.bias',
'bert.encoder.layer.4.output.LayerNorm.weight',
'bert.encoder.layer.28.attention.self.key.bias',
'bert.encoder.layer.9.attention.self.key.bias',
'bert.encoder.layer.12.attention.self.query.bias',
'bert.encoder.layer.18.attention.self.query.bias',
'bert.encoder.layer.29.output.dense.bias',
'bert.encoder.layer.12.attention.output.LayerNorm.bias',
'bert.encoder.layer.17.intermediate.dense.weight',
'bert.encoder.layer.23.output.LayerNorm.weight',
'bert.encoder.layer.3.output.LayerNorm.bias',
'bert.encoder.layer.4.intermediate.dense.bias',
'bert.encoder.layer.13.output.LayerNorm.bias',
'bert.encoder.layer.19.output.LayerNorm.weight']
- This IS expected if you are initializing BertForSequenceClassification from
the checkpoint of a model trained on another task or with another architecture
(e.g. initializing a BertForSequenceClassification model from a
BertForPreTraining model).
- This IS NOT expected if you are initializing BertForSequenceClassification
from the checkpoint of a model that you expect to be exactly identical
(initializing a BertForSequenceClassification model from a
BertForSequenceClassification model).
```

```
Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at Rostlab/prot_bert_bfd and are newly initialized:
['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

```
HBox(children=(HTML(value='dataset/deeploc_per_protein_train.csv'),␣
 ↪FloatProgress(value=0.0, max=7417223.0), H…
```

```
HBox(children=(HTML(value='dataset/deeploc_per_protein_test.csv'),␣
 ↪FloatProgress(value=0.0, max=2076681.0), HT…
```

```
unique_class_names: ['M' 'S' 'U']
```

```
/home/amos/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3
.8/site-packages/dask/utils.py:893: FutureWarning: Boolean inputs to the
`inclusive` argument are deprecated infavour of `both` or `neither`.
  return getattr(obj, self.method)(*args, **kwargs)
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, layout=Layout(flex='2'),␣
 ↪max=589.0), HTML(value='')), …
```

```
/home/amos/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3
.8/site-packages/dask/utils.py:893: FutureWarning: Boolean inputs to the
`inclusive` argument are deprecated infavour of `both` or `neither`.
  return getattr(obj, self.method)(*args, **kwargs)
/home/amos/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3
.8/site-packages/dask/utils.py:893: FutureWarning: Boolean inputs to the
`inclusive` argument are deprecated infavour of `both` or `neither`.
  return getattr(obj, self.method)(*args, **kwargs)
```

```
Report has been saved to /home/amos/Dropbox/GeneralCareerDevelop/PersonalBrand/G
ithubRepositories/ProSelfLC_tpami2release/demos_jupyter_notebooks/bert_deeploc/d
ataset//deeploc_eda_reports/MS-with-unknown_train.html!
```

```
/home/amos/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3
.8/site-packages/dask/utils.py:893: FutureWarning: Boolean inputs to the
`inclusive` argument are deprecated infavour of `both` or `neither`.
  return getattr(obj, self.method)(*args, **kwargs)
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, layout=Layout(flex='2'),␣
 ↪max=589.0), HTML(value='')), …
```

```
/home/amos/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3
.8/site-packages/dask/utils.py:893: FutureWarning: Boolean inputs to the
`inclusive` argument are deprecated infavour of `both` or `neither`.
  return getattr(obj, self.method)(*args, **kwargs)
```

```
Report has been saved to /home/amos/Dropbox/GeneralCareerDevelop/PersonalBrand/G
ithubRepositories/ProSelfLC_tpami2release/demos_jupyter_notebooks/bert_deeploc/d
ataset//deeploc_eda_reports/MS-with-unknown_valid or test: the same
dataset.html!
```

0.5 Store the params configurations

For each experiment, we have one unique result folder to store the params configurations and learning curves. Therefore, you can revisit any specific experiment whenever you need without losing any details.

[5]:

```
# params["milestones"] was a list of integers, we convert it to a string before␣
 ↪sinking.
params["milestones"] = str(params["milestones"])
dataframe = pandas.DataFrame(params, index=[0])
dataframe.to_csv(
    params["summary_writer_dir"] + "/params.csv",
    encoding="utf-8",
    index=False,
    sep="\t",
    mode="w",  #
)
```

0.6 Run the trainer and save the final model

[6]:

```
trainer.train()
torch.save(
    trainer.network,
    params["summary_writer_dir"] + "/model.pt",
)
print(
    "The experiment is finished with details sinked in {}".format(
        params["summary_writer_dir"]
    )
)
```

```
Evaluating Network…
Iteration= (100,1)/(66200, 40), lr=0.0100, batch_mean_epsilon=0.0001, valid
batch size=4.0000, batch_mean_gtrust=0.0025, batch_mean_etrust=0.0434
noisy_train: Loss= 0.6890, Accuracy= 0.5501, Entropy= 0.9977, Max_p= 0.5284,
test: Loss= 0.6849, Accuracy= 0.5863, Entropy= 0.9977, Max_p= 0.5284,
Evaluating Network…
Iteration= (200,1)/(66200, 40), lr=0.0100, batch_mean_epsilon=0.0000, valid
batch size=4.0000, batch_mean_gtrust=0.0026, batch_mean_etrust=0.0110
noisy_train: Loss= 0.6904, Accuracy= 0.5501, Entropy= 0.9992, Max_p= 0.5165,
test: Loss= 0.6879, Accuracy= 0.5863, Entropy= 0.9992, Max_p= 0.5165,
Evaluating Network…
```

```
Iteration= (300,1)/(66200, 40), lr=0.0100, batch_mean_epsilon=0.0000, valid
batch size=4.0000, batch_mean_gtrust=0.0026, batch_mean_etrust=0.0022
```

```
Traceback (most recent call last):
  File "/usr/lib/python3.8/multiprocessing/queues.py", line 245, in _feed
    send_bytes(obj)
  File "/usr/lib/python3.8/multiprocessing/connection.py", line 200, in
send_bytes
    self._send_bytes(m[offset:offset + size])
  File "/usr/lib/python3.8/multiprocessing/connection.py", line 411, in
_send_bytes
    self._send(header + buf)
  File "/usr/lib/python3.8/multiprocessing/connection.py", line 368, in _send
    n = write(self._handle, buf)
BrokenPipeError: [Errno 32] Broken pipe
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Input In [6], in <cell line: 1>()
----> 1 trainer.train()
      2 torch.save(
      3     trainer.network,
      4     params["summary_writer_dir"] + "/model.pt",
      5 )
      6 print(
      7     "The experiment is finished with details sinked in {}".format(
      8         params["summary_writer_dir"]
      9     )
     10 )

File ~/Dropbox/GeneralCareerDevelop/PersonalBrand/GithubRepositories/
  ↪ProSelfLC_tpami2release/src/proselflc/trainer/
  ↪trainer_cnn_vision_derivedgrad_adaptedfordeeploc.py:262, in Trainer.train(self)
    258 def train(self) -> None:
    259     # ##############################
    260     for epoch in range(1, self.total_epochs + 1):
    261         # train one epoch
--> 262         self.train_one_epoch(
    263             epoch=epoch,
    264             dataloader=self.traindataloader,
    265         )
    266     # ##############################
    267     self.sink_csv_figures()

File ~/Dropbox/GeneralCareerDevelop/PersonalBrand/GithubRepositories/
  ↪ProSelfLC_tpami2release/src/proselflc/trainer/
  ↪trainer_cnn_vision_derivedgrad_adaptedfordeeploc.py:438, in Trainer.
  ↪train_one_epoch(self, epoch, dataloader)
    436     # for logging, I am still using epoch, only to reduce changes here.
    437     if self.cur_time % self.params["eval_interval"] == 0:
--> 438         self.eval_helper(epoch)
    439         self.network.train()  # self.network.train(mode=True)
    440 #
    441 # final testing

File ~/Dropbox/GeneralCareerDevelop/PersonalBrand/GithubRepositories/
  ↪ProSelfLC_tpami2release/src/proselflc/trainer/
  ↪trainer_cnn_vision_derivedgrad_adaptedfordeeploc.py:485, in Trainer.
  ↪eval_helper(self, epoch)
    482 self.max_p_dynamics["epoch"].append(self.cur_time)
    484 for eval_dataname, eval_dataloader in self.dataloaders.items():
--> 485     (eval_loss, eval_accuracy, eval_entropy, eval_max_p) =␣
  ↪self.evaluation(
    486         dataloader=eval_dataloader,
    487     )
    488     self.loss_dynamics[eval_dataname].append(eval_loss)
    489     self.accuracy_dynamics[eval_dataname].append(eval_accuracy)

File ~/.local/share/virtualenvs/ProSelfLC_tpami2release-X5r1NEDs/lib/python3.8/
  ↪site-packages/torch/autograd/grad_mode.py:28, in _DecoratorContextManager.
  ↪__call__.<locals>.decorate_context(*args, **kwargs)
     25 @functools.wraps(func)
     26 def decorate_context(*args, **kwargs):
     27     with self.__class__():
---> 28         return func(*args, **kwargs)

File ~/Dropbox/GeneralCareerDevelop/PersonalBrand/GithubRepositories/
  ↪ProSelfLC_tpami2release/src/proselflc/trainer/
  ↪trainer_cnn_vision_derivedgrad_adaptedfordeeploc.py:538, in Trainer.
  ↪evaluation(self, dataloader)
    532 loss = cross_entropy(
    533     pred_probs=pred_probs,
    534     target_probs=labels,
    535 )
    536 # ##############################
--> 538 test_loss += loss.item()
    539 _, preds = pred_probs.max(1)
    540 _, annotations = labels.max(1)

KeyboardInterrupt:
```

```
[ ]:
```