Special Project 10/17 HW2
What can we do with Self-Supervised Learning?

Team A
吳冠緯 梁正

Outline

- Briefly talking about Self-Supervised Learning and the tasks
- Processing steps
- Experiment results
- Unsolved problems

Self-Supervised Learning

- labeled data is expensive or hard to access
- learned from (part of) unlabeled data itself
- more like the way human solve the problem

Self-Supervised Learning

- many ways are used in upstream models : masked reconstruction / prediction, contrasive, handcrafted, etc.

- the pretrained upstream model can be fine-tuned on different fields of downstream tasks : recognition, speaker, semantic, etc.

Self-Supervised Learning

- for S3PRL, it supplys a covenient and useful interface to link between upstream models pretraining and downstream tasks fine-tuning

- for NLP tasks, we also need a benchmark corpus like GLUE for text tasks, so we use SUPERB as an universal performannce benchmark

Our tasks

- for HW2, we choose track 1 (1 point), following the description and trying to run several “ upstream and downstream ” pairs

- we focus on content and speaker categories and use fbank as acoustic feature upstream (the basic upstream case), and after trying fbank, we choose other self-supervised pre-trained model as our upstream model (wav2vec 2.0 on PR and ASR)

Our tasks

- for the downstream tasks, we try to run phoneme recognition, automatic speech recognition, speaker diarization, and, keyword spotting

- while attempting to do other tasks, we also meet some problems, and will be mentioned after the experiment results part

Processing steps

Follow the guidilne in s3prl/downstream/docs/supeb.md

- donwload different dataset required by task
- specify the dataset root directory in config.yaml
- use run_downstream.py and specify its arguments

Experiment Result - Phoneme Recognition

```
[Runner] - Resume from /content/s3prl/s3prl/result/downstream/fbank_ctc_test/states-100000.ckpt
[Featurizer] - The selected feature hidden_states's downsample rate is 160
[Runner] - Loading Downstream weights from the previous experiment
[LibriPhone] - Taking the first phoneme sequences for a deterministic behavior.
word -> phonemes: 100% 2620/2620 [00:00<00:00, 7202.73it/s]
test: 100% 2620/2620 [00:26<00:00, 100.08it/s]
test loss: 2.8933188915252686
test wer: 0.8693283586055881
```

Experiment Result - Automatic Speech Recognition

```
[Runner] - Resume from /content/s3prl/s3prl/result/downstream/fbank_asr_test/states-20000.ckpt
[Featurizer] - The selected feature hidden_states's downsample rate is 160
[Runner] - Loading Downstream weights from the previous experiment
ASR dataset test-clean: 100% 2620/2620 [00:00<00:00, 34811.69it/s]
test-clean: 100% 2620/2620 [06:32<00:00,  6.67it/s]
test-clean loss: 0.4147726893424988
test-clean uer: 9.26007660040124
test-clean wer: 37.15367672630318
```

Experiment Result - Speaker Diarization

```
[Runner] - Resume from result/downstream/ExpName/best-states-dev.ckpt
[Featurizer] - The selected feature hidden_states's downsample rate is 160
[Runner] - Loading Downstream weights from the previous experiment
3000  recordings
test: 100% 3000/3000 [01:15<00:00, 39.96it/s]
mode test acc 0.9214757680892944 der 0.10929664969444275
```

Experiment Result - Keyword Spotting

```
[Runner] - Resume from result/downstream/fbank-speech_commands/states-200000.ckpt
[Featurizer] - The selected feature hidden_states's downsample rate is 160
[Runner] - Loading Downstream weights from the previous experiment
test: 100% 97/97 [00:05<00:00, 16.43it/s]
test acc: 0.085037325543654 66
```

- steps=200000
- gradient accumulation = 1 or 2
- w/ or w/o scheduler

Experiment Result - ASR (wav2vec2 as upstream)

```
[Runner] - Resume from /content/s3prl/s3prl/result/downstream/wave2vec2_asr
Using cache found in /root/.cache/torch/hub/s3prl_cache/3a990c945fbe378df95
for https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt
[Featurizer] - Take a list of 13 features and weighted sum them.
[Featurizer] - The selected feature hidden_states's downsample rate is 320
[Runner] - Loading Featurizer weights from the previous experiment
[Runner] - Loading Downstream weights from the previous experiment
ASR dataset test-clean: 100% 2620/2620 [00:00<00:00, 38331.94it/s]
test-clean: 100% 2620/2620 [04:26<00:00,  9.83it/s]
test-clean loss: 0.10173542797565546
test-clean uer: 1.5108955921733094
test-clean wer: 6.929280633012516
```

Experiment Result - PR (wav2vec2 as upstream)

```
word -> phonemes: 100% 2620/2620 [00:00<00:00, 9293.11it/s]
test: 100% 2620/2620 [01:23<00:00, 31.38it/s]
test loss: 0.3006996512413025
test wer: 0.07195816126409108
```

Unsolved Problems (happens on ASV, QbE)

TypeError: type object get multiple values for keyword argument “downstream”

```
[Runner] - Start a new experiment
[Override] - config.downstream_expert.dtwrc.dist_method = cosine_exp
[Override] - config.downstream.datarc.dataset_root = /content/data/quesst14Database
Using cache found in /root/.cache/torch/hub/s3prl_cache/5e6b91abd59b390dc3f89225f0e7d26f5bcb6fac496a08110c8862e3b1bb93e7
for https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt
[Featurizer] - The selected feature hidden_states's downsample rate is 320
Traceback (most recent call last):
  File "run_downstream.py", line 219, in <module>
    main()
  File "run_downstream.py", line 214, in main
    runner = Runner(args, config)
  File "/content/s3prl/s3prl/downstream/runner.py", line 96, in __init__
    self.downstream = self._get_downstream()
  File "/content/s3prl/s3prl/downstream/runner.py", line 189, in _get_downstream
    **vars(self.args)
TypeError: type object got multiple values for keyword argument 'downstream'
```