Machine Learning HW4
Speaker Identification

ML TAs
mlta-2022-spring@googlegroups.com
Outline

● Task Description
● Dataset
● Data Segmentation
● Model Architecture
● Baselines
● Report
● Guidelines
Task Introduction

- **Self-attention**
  - Proposed in GOOGLE's work, *Attention is all you need*. It combines the strengths of RNN (consider the whole sequence) and CNN (processing parallelly).

- **Goal:** Learn how to use Transformer.

Ref. Prof. Hung-Yi Lee
[2021Spring ML] Transformer (1), (2)
Speaker Identification

Task: Multiclass Classification

Predict speaker class from given speech.
Dataset - VoxCeleb2

- Training: 56666 processed audio features with labels.
- Testing: 4000 processed audio features (public & private) without labels.
- Label: 600 classes in total, each class represents a speaker.

VoxCeleb2: [Link]
Data Preprocessing

Ref. Prof. Hung-Yi Lee
[2020Spring DLHLP] Speech Recognition
Data Format

- **Data Directory**
  - metadata.json
  - testdata.json
  - mapping.json
  - uttr-{random string}.pt

- **The information in metadata**
  - "n_mels": The dimension of mel-spectrogram.
  - "speakers": A dictionary.
    - Key: speaker ids
    - Value: "feature_path" and "mel_len"
Data Segmentation During Training

Different length:
Data Segmentation During Training

Different length:

Segment during training
Segment = 2
Model Architecture

Input (preprocessed) -> Encoder -> Features -> Pooling Layer -> Linear Layer -> Prediction

Input Representation: dim=600
Sample Code

- **Link**

- **Baseline Methods**
  - Simple: Run sample code & know how to use Transformer.
  - Medium: Know how to adjust parameters of Transformer.
  - Strong: Construct [Conformer](#), which is a variety of Transformer.
  - Boss: Implement [Self-Attention Pooling](#) & [Additive Margin Softmax](#) to further boost the performance.
Requirements - Simple

- Build a self-attention network to classify speakers with sample code.
- Simple public baseline: 0.60824
- Estimate training time: 30~40 mins on Colab.
Requirements - Medium

- Modify the parameters of the transformer modules in the sample code.
- Medium public baseline: 0.70375

```python
class Classifier(nn.Module):
    def __init__(self, d_model=80, n_spks=600, dropout=0.1):
        super(Classifier, self).__init__()
        # Project the dimension of features from that of input into d_model.
        self.prenet = nn.Linear(40, d_model)
        # TODO:
        # Change Transformer to Conformer.
        self.encoder_layer = nn.TransformerEncoderLayer(d_model=d_model, dim_feedforward=256, nhead=2)
        # self.encoder = nn.TransformerEncoderLayer(self.encoder_layer, num_layers=2)

        # Project the the dimension of features from d_model into speaker nums.
        self.pred_layer = nn.Sequential(
nn.Linear(d_model, d_model),
n.nn.ReLU(),
n.nn.Linear(d_model, n_spks),
        )
```

Estimate training time: 1~1.5 hour on Colab
Requirements - Strong

- Construct **Conformer**, which is a variety of Transformer.
- Strong public baseline: 0.77750

```python
class Classifier(nn.Module):
    def __init__(self, d_model=80, n_spks=600, dropout=0.1):
        super().__init__()
        # Project the dimension of features from that of input into d_model.
        self.prenet = nn.Linear(40, d_model)
        # TODO:
        #   Change Transformer to Conformer.
        self.encoder_layer = nn.TransformerEncoderLayer(
            d_model=d_model, dim_feedforward=256, nhead=2)
        # self.encoder = nn.TransformerEncoder(self.encoder_layer, num_layers=2)

        # Project the dimension of features from d_model into speaker nums.
        self.pred_layer = nn.Sequential(
            nn.Linear(d_model, d_model),
            nn.ReLU(),
            nn.Linear(d_model, n_spks),
        )
```

Estimate training time: 3 ~ 4 hours on Colab
Hints

- **Conformer**

Ref. Prof. Hung-Yi Lee
[2021Spring ML] Network Compression
Requirements - Boss

- Implement **Self-Attention Pooling** & **Additive Margin Softmax** to further boost the performance.
- Public boss baseline: 0.86500
- Estimate training time: about 2~2.5 hours on Kaggle.
Hints

- **Self-Attention Pooling**
Hints

- Additive Margin Softmax
Grading

- Evaluate Metric: @1 Accuracy
- **Simple Baseline** (Public / Private) +0.5 pt / 0.5 pt
- **Medium Baseline** (Public / Private) +0.5 pt / 0.5 pt
- **Strong Baseline** (Public / Private) +0.5 pt / 0.5 pt
- **Boss Baseline** (Public / Private) +0.5 pt / 0.5 pt
- Code Submission +2 pts
- Report +4 pts
Submission Format

- "Id, Category" split by ',' in the first row.
- Followed by 8000 lines of "filename, speaker name" split by ','.
Code Submission

- Submit your code to **NTU COOL** (2 pts).
  - We can only see your **last submission**.
  - Do **NOT** submit the model or dataset.
  - If your codes are not reasonable, your final grade will be **x 0.9**
  - You should compress your code into a single zip file:
    - ex. b08902126_hw4.zip
    - `<Student ID>_hw4.zip`
Report (4 pts)

1. Make a brief introduction about a variant of Transformer. (2 pts)

2. Briefly explain why adding convolutional layers to Transformer can boost performance. (2 pts)
Deadline

- Kaggle: **2022/04/01 23:59 (UTC+8)**
- NTU COOL: **2022/04/01 23:59 (UTC+8)**
- Report: **2022/04/01 23:59 (UTC+8)**
Links

- Kaggle: [link](#)
- Colab: [link](#)
- Data: [link](#)
- Dataset: [link](#)
Regulation

- You should NOT plagiarize, if you use any other resource, you should cite it in the reference. ( *)
- You should NOT modify your prediction files manually.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- Do NOT search or use additional data or pre-trained models.
- Your final grade x 0.9 if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

( *) Academic Ethics Guidelines for Researchers by the Ministry of Science and Technology (MOST)
If any questions, you can ask us via...

- NTU COOL (Recommended)
  - https://cool.ntu.edu.tw/courses/11666
- Email
  - mlta-2022-spring@googlegroups.com
  - The title should begin with “[hw4]”
- TA hour
  - Mandarin: Tuesday, 20:00~21:00
  - English: Friday, 22:00 ~ 23:00
Appendix

- Colab 縮排問題
  - 工具 -> 設定: