

Homework-0

Musical note classification

HW0 TA：李維釗 (Lonian Lee)

Office hour: Tue. 13:20 ~ 14:05 @BL505

Outline

- Rules
- Overview
- Timeline
- Detailed Explanation
- Scoring
- Submission

Rules

- Don't cheat
- Don't use pretrained model
- Don't use extra data
- Don't use the test data while training or tuning your model
- Can use public codes with citation in report

Overview

1. Visualize a Mel-Spectrogram:

Use Python libraries like librosa or torchaudio to create and visualize the mel-spectrogram of an audio file. Briefly describe what the mel-spectrogram shows

2. Train a Traditional Machine Learning Model:

Extract relevant features from the audio data, then train a model using traditional machine learning techniques such as SVM, Random Forest, or k-NN

3. Train a Deep Learning Model:

Train a model using deep learning techniques, such as a CNN or an attention-based model

Timeline

- W1 – 09/05 (Thursday): Announcement of HW0
- W4 – 09/25 (Wednesday 23:59pm): Deadline
 - Late submission: 1 day (-20%), 2 days (-40%), after that (-60%)

Detailed Explanation

- Dataset
- Homework
 - Task 1: Visualize a Mel-Spectrogram
 - Task 2: Train a Traditional Machine Learning Model
 - Task 3: Train a Deep Learning Model
- Evaluation metrics

Dataset: NSynth

- 289,205 for training
12,678 for validation
4,096 for testing
- 4 seconds audio; the note was held for the first three seconds and allowed to decay for the final second
- More detailed information please refer the [source](#)

Jesse Engel et al, “Neural audio synthesis of musical notes with WaveNet autoencoders,” ICML 2017.

Feature	Type	Description
note	int64	A unique integer identifier for the note.
note_str	bytes	A unique string identifier for the note in the format <code><instrument_str>-<pitch>-<velocity></code> .
instrument	int64	A unique, sequential identifier for the instrument the note was synthesized from.
instrument_str	bytes	A unique string identifier for the instrument this note was synthesized from in the format <code><instrument_family_str>-<instrument_production_str>-<instrument_name></code> .
pitch	int64	The 0-based MIDI pitch in the range [0, 127].
velocity	int64	The 0-based MIDI velocity in the range [0, 127].
sample_rate	int64	The samples per second for the <code>audio</code> feature.
audio*	[float]	A list of audio samples represented as floating point values in the range [-1,1].
qualities	[int64]	A binary vector representing which <code>sonic qualities</code> are present in this note.
qualities_str	[bytes]	A list IDs of which qualities are present in this note selected from the <code>sonic qualities list</code> .
instrument_family	int64	The index of the <code>instrument family</code> this instrument is a member of.
instrument_family_str	bytes	The ID of the <code>instrument family</code> this instrument is a member of.
instrument_source	int64	The index of the <code>sonic source</code> for this instrument.
instrument_source_str	bytes	The ID of the <code>sonic source</code> for this instrument.

Dataset: NSynth

- In this homework you need to train models to predict the instrument_family
- The total classes number is 11
- The sota classifier model can reach 75%+ accuracy. [[paper](#)] Some previous works can reach 70%+ accuracy. e.g. [MULE](#), [MERT](#), [MusicCNN](#), etc...

Index	ID
0	bass
1	brass
2	flute
3	guitar
4	keyboard
5	mallet
6	organ
7	reed
8	string
9	synth_lead
10	vocal

Luyu Wang et al, "Towards learning universal audio representations," ICASSP 2022.

Task1: Visualize a Mel-Spectrogram

- Select 3 different instruments and 3 different pitches and visualize one mel-spectrogram of them
- There should be 9 figures in your report but the format is up to you
- Remember to convert the power to dB
- Please fix the FFT window size=2048, and the hop length=512
- [Librosa](#) and [torchaudio](#) are recommended

Task2: Traditional ML Model

- Train a traditional ML model (e.g. k-NN, SVM, random forest) with **any features** extracted from the audio
- Need to report **how to implement the model** clearly
- Need to report the testing result (not validation result) with **confusion matrix, top1 accuracy, and top3 accuracy**
- Remember to utilize standardization (e.g. mean, std), pooling and normalization to ensure consistent feature scales, reducing overfitting, and improving model stability and performance during training
- [Sklearn](#) is recommended

Task3: Deep Learning Model

- Train a deep learning model (e.g. CNN or attention based model) with **Mel-spectrograms** extracted from the audio as input
 - Need to compare 2 different kinds of inputs: **Mel-spectrograms with or without taking the log**
 - You can choose whatever FFT window size and hop length you like
 - You can choose whatever deep learning model you like
- Need to report **how to implement the model** clearly
- Need to report the testing result (not validation result) with **confusion matrix, top1 accuracy, and top3 accuracy**
- You can use any music tagging model. For a novice, the **short chunk CNN** in this [repo](#) is recommended. (Need to replace the BCE loss to Cross-entropy loss)

Evaluation on the Test Set

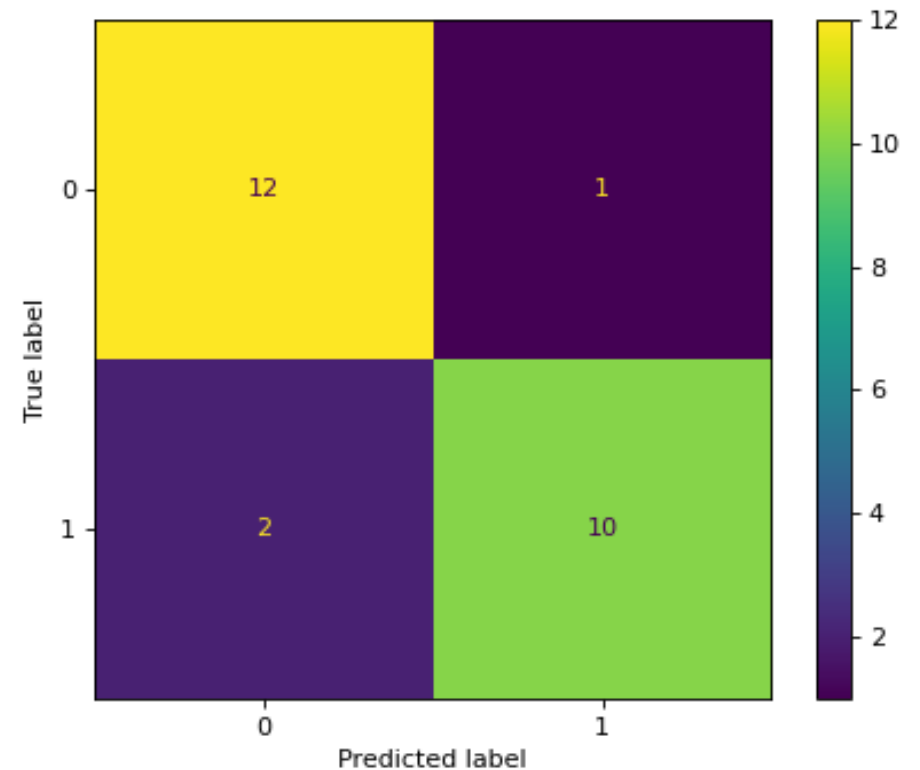
1. Confusion matrix

A table that summarizes the performance of a classification algorithm by comparing the predicted labels with the actual labels

2. Top k accuracy (Top 1 and Top 3)

For a given k, if the correct label is within the top k predicted classes, it's counted as correct; otherwise, it's counted as incorrect

- Only use the test set for evaluation; don't use it at all for training



Scoring

- HW0 accounts for 15% of the total grade
 - Report: 100%

Submission file and details

1. Report (to NTU cool)
 2. Readme file and requirements.txt (to your cloud drive)
 3. Code and one model checkpoint for inference. (to your cloud drive)
- We will randomly select several classmates' code to run inference on your model and run the score on your results, so please ensure that the files you upload include trained model which can successfully execute the entire inference process and the generation results
 - **Don't** upload: training data, testing data, preprocessed data, others model, cache file

Report

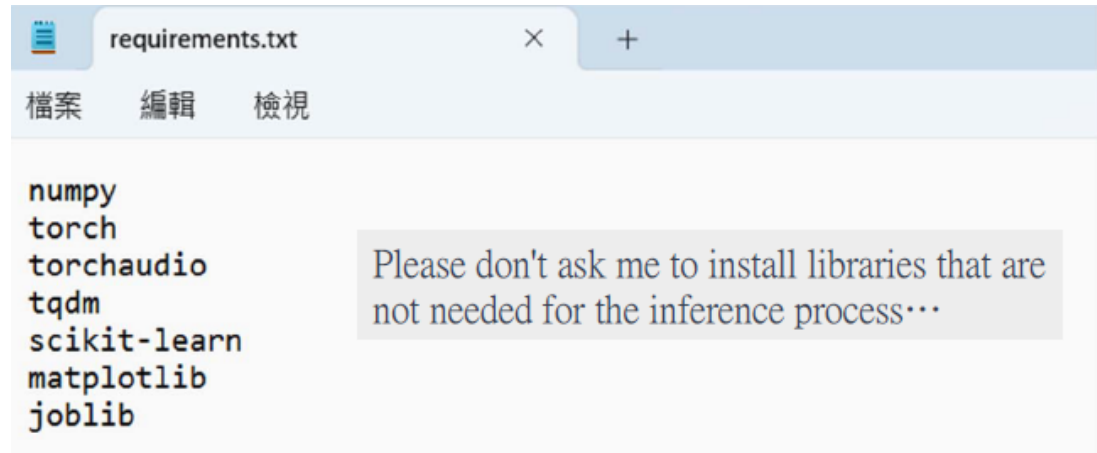
- Write with PPT or PPT-like format (in 16:9 aspect ratio)
- Upload studentID_report.pdf (ex: r12345678_report.pdf)
- Please create a report that is clear and can be understood without the need for oral explanations
- There is **no specific length requirement**, but it should clearly communicate the experiments conducted and their results; approximately 10 pages is a suggested standard, but not a strict limitation

Code

- Upload all your **source code and model** to a cloud drive, **open access permissions**, and then **upload the link to the NTU Cool assignment HW0_report in comments**, as well as include it on the first page of the report
- You will need to upload **requirements.txt**
- I'll run :

```
pip install -r requirements.txt
```

If you have used third-party programs that cannot be installed directly via 'pip install,' please write the URL and install method command by command on **your readme file**.



The screenshot shows a code editor window titled 'requirements.txt'. The editor has a menu bar with '檔案' (File), '編輯' (Edit), and '檢視' (View). The content of the file is a list of Python packages: numpy, torch, torchaudio, tqdm, scikit-learn, matplotlib, and joblib. A light gray tooltip is visible on the right side of the editor, containing the text: 'Please don't ask me to install libraries that are not needed for the inference process...'

```
requirements.txt
numpy
torch
torchaudio
tqdm
scikit-learn
matplotlib
joblib
```

Please don't ask me to install libraries that are not needed for the inference process...

Code

- You will also need to upload **README.txt** or **README.pdf** to guide me on how to perform inference on your model. (I'll use the same test set in Nsynth, ensure I can run your code with the test set path in my device.)
- The inference code should print top 1 and top 3 accuracy

ALL things you need to do before 09/25 23:59

- HW0_report
 - StudentID_report.pdf
 - Cloud drive link
 - README.txt or README.pdf
 - Requirements.txt
 - Codes and model to run inference
 - Others codes

噠噠噠噠噠!



When you encounter problem:

1. Check out all course materials and announcement documents
2. Use the power of the internet and AI
3. Use **Discussions** on NTU COOL
4. Email me weijaw2000@gmail.com or come to office hour