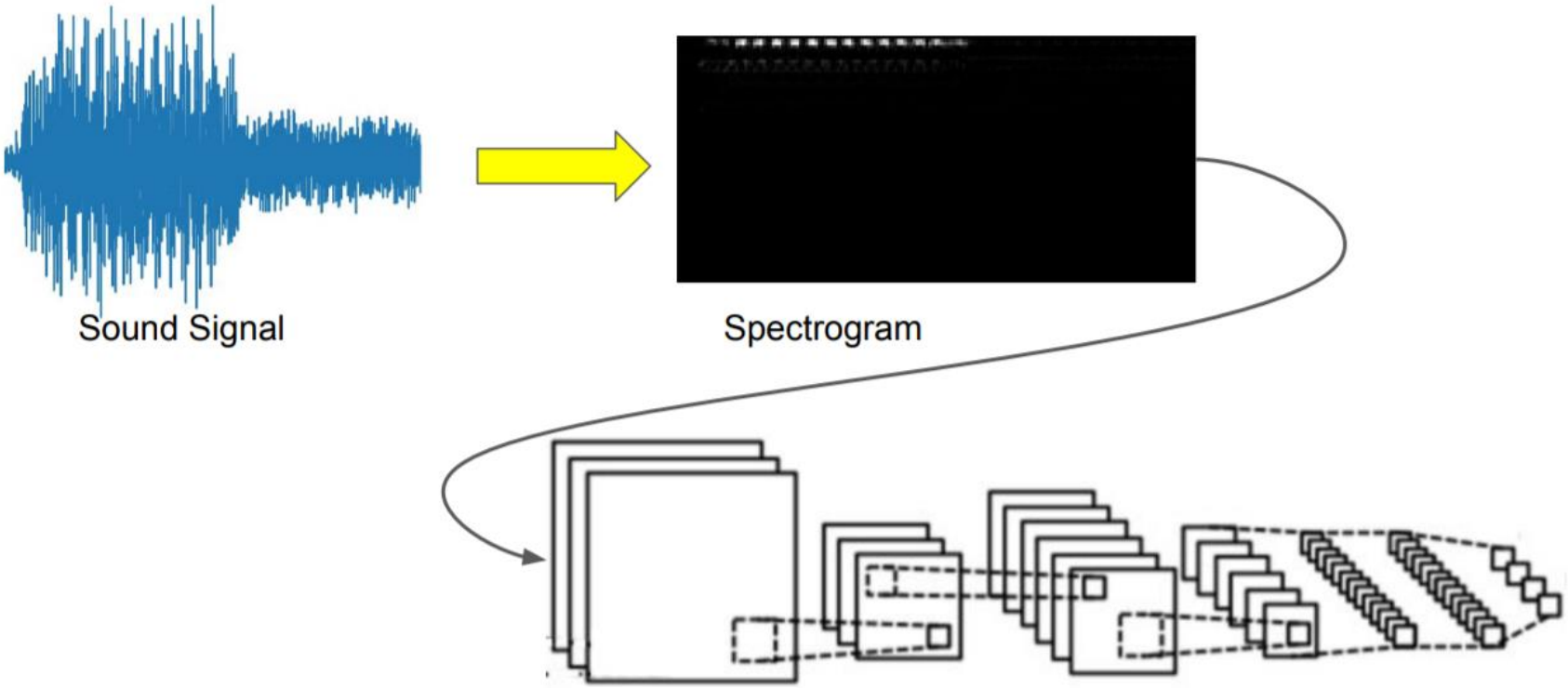
DSP Final Project 2019
Sound Classification

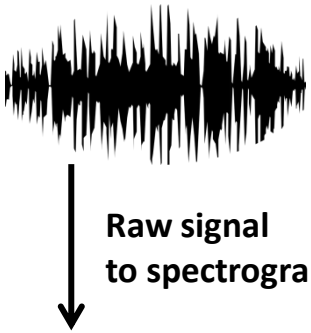
TA: Timmy S. T. Wan 萬世澤

Sound Classification using CNNs



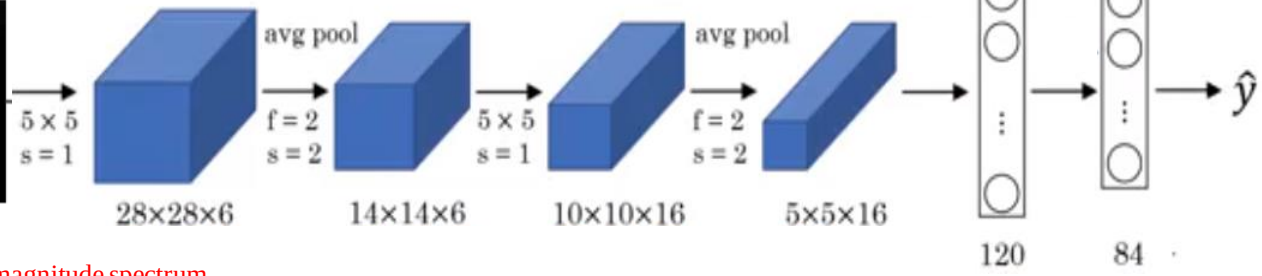
In this homework, you need to train a neural network with spectrograms for 20-classes sound classification problem.

Sound Classification Example



Raw signal
to spectrogram

Resize



32x32x1 E.g. magnitude spectrum

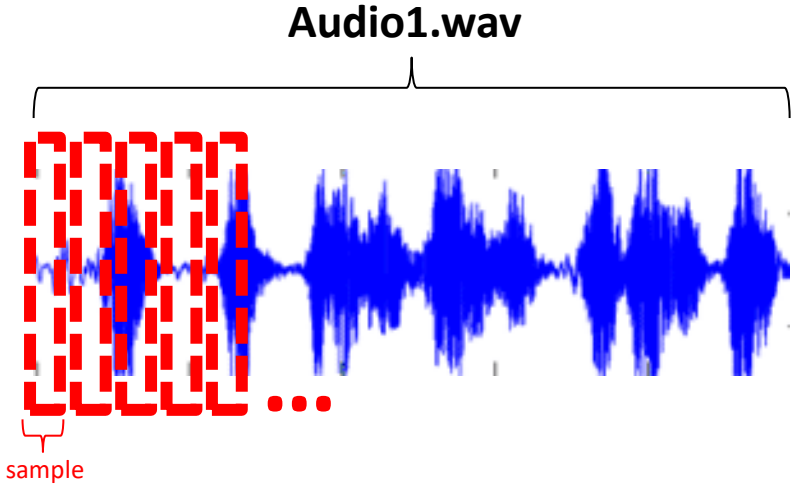
Or

32x32x2 E.g. magnitude spectrum + phase spectrum

Ref. Convolutional Neural Network (CNNs) by Andrew Ng [Full Course] Chapter13 Classic Networks

Dataset

- Sound 20 dataset
  - Include 20 animal and instrument sounds
  - Each class contains at least one or more audio files
  - Each audio file is split into **multiple samples** (see the example below)



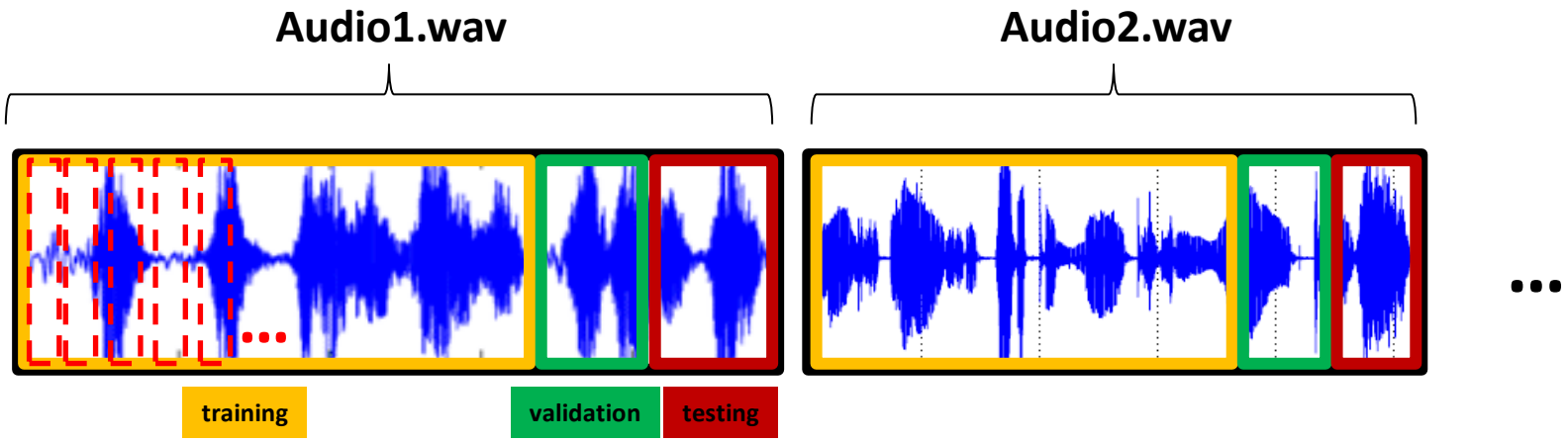
Length of each sample under each class

Label ID	Label Name	Sample length
0	Tettigonioidea1	22050 (0.5s)
1	Tettigonioidea2	22050 (0.5s)
2	drums_Snare	11025(0.25s)
3	Grylloidea1	22050 (0.5s)
4	drums_MidTom	11025(0.25s)
5	drums_HiHat	11025(0.25s)
6	drums_Kick	11025(0.25s)
7	drums_SmallTom	11025(0.25s)
8	guitar_chord2	11025(0.25s)
9	Frog1	11025(0.25s)

Label ID	Label Name	Sample length
10	Frog2	11025(0.25s)
11	drums_FloorTom	11025(0.25s)
12	guitar_7th_fret	11025(0.25s)
13	drums_Rim	11025(0.25s)
14	Grylloidea2	22050(0.5s)
15	guitar_3rd_fret	11025(0.25s)
16	drums_Ride	11025(0.25s)
17	guitar_chord1	11025(0.25s)
18	guitar_9th_fret	11025(0.25s)
19	Frog3	11025(0.25s)

Train, Val and Test set

- For multiple samples in each audio file, we take 70% of the data for training, 15% of the data for validation, and 15% of the data for testing



Number of samples under each class

Label ID	Label Name	Train, Val Samples
0	Tettigonioidea1	(607,129)
1	Tettigonioidea2	(388,83)
2	drums_Snare	(907,194)
3	Grylloidea1	(906,192)
4	drums_MidTom	(512,110)
5	drums_HiHat	(593,127)
6	drums_Kick	(484,104)
7	drums_SmallTom	(506,109)
8	guitar_chord2	(963,206)
9	Frog1	(16,8)

Label ID	Label Name	Train, Val Samples
10	Frog2	(20,10)
11	drums_FloorTom	(442,95)
12	guitar_7th_fret	(879,188)
13	drums_Rim	(355,76)
14	Grylloidea2	(503,106)
15	guitar_3rd_fret	(921,197)
16	drums_Ride	(492,106)
17	guitar_chord1	(330,71)
18	guitar_9th_fret	(949,203)
19	Frog3	(20,10)

Total (Train, Val, Test)	(10793,2324,2387)

Provided dataset structure

- Download the DSP_HW.zip (Size: 433MB)
  - https://drive.google.com/file/d/1b4MJ7nbjjsQvvkVZHUNRmq8mOCvFeqzj/view?usp=sharing
- The dataset structure is arranged as below

```
-train
    -drums_FloorTom
        0.npy
        1.npy
        …
    -drums_HiHat
        0.npy
        1.npy
        …
    …
-val
    -drums_FloorTom
        442.npy
        443.npy
        …
    -drums_HiHat
        593.npy
        594.npy
        …
    …
test.npy
npy2wav.py
```

Each .npy file under train and val folder contains the raw mono sound signal recording at 44.1 kHz

```
>>> import numpy as np
>>> signal = np.load('val/drums_FloorTom/442.npy')
>>> print(signal)
[0.03421021 0.03417969 0.03424072 ... 0.00550842 0.00553894 0.00587463]
```

test.npy contains a list of testing data

```
>>> test_data = np.load('test.npy')
>>> print(test_data)
[array([-0.01092529, -0.01101685, -0.01278687, ..., -0.250412  ,
       -0.2587738 , -0.26716614], dtype=float32)
 array([-0.00035095,  0.00704956,  0.00616455, ..., -0.00242615,
       -0.00424194, -0.0110321 ], dtype=float32)
 array([-0.3861084 , -0.3708191 , -0.3531952 , ..., -0.24938965,
       -0.25704956, -0.2659912 ], dtype=float32)
 ...
 array([-0.00686646,  0.00189209, -0.00291443, ..., -0.00187683,
       -0.00186157, -0.00212097], dtype=float32)
 array([-0.08750916, -0.08599854, -0.08857727, ..., -0.00990295,
       -0.00067139,  0.01026917], dtype=float32)
 array([-0.03076172, -0.03131104, -0.03092957, ...,  0.00953674,
        0.00616455,  0.00273132], dtype=float32)]
```

If you want to reconstruct the original audio, please run the script *npy2wav.py*

```
python  npy2wav.py  --labelname  <class_name>
# the script will generate <class_name>.wav in the current folder
# E.g. python  npy2wav.py  --labelname  drums_FloorTom
```

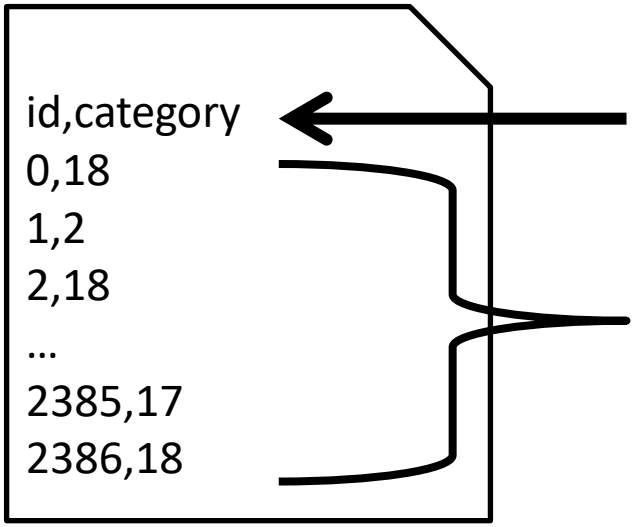
Requirements

1. Implement a CNN network for sound classification
   - The input should be a spectrogram. You need to try at least **2 or more** settings (FFT-Length, Window type, Spectrum type and Overlap) to generate spectrograms for training, validation, and testing sets (10%)
   - Then, train the model with spectrograms. You must train at least **1** model but 2 or more models would be helpful for experimental comparisons. (10%)
2. Show the classification accuracy on the validation set (10%)
3. Submit predictions for the testing set to Kaggle judge system.
   - Report classification accuracy on the public set (10%)
   - Report classification accuracy on the private set. (10%)

   Note. Testing set consists of public set and private set
4. Prepare a report to describe your experimental settings, model configurations or even interesting findings (50%)

Format of test predictions

- Your prediction result should follow the list order of samples in test.npy
- Prediction of each sample should be an integer in 0~19
- Store your results as *.csv
- Follow the format below:



add the header “id,category” by yourself

The first prediction is 18 (i.e. guitar_9th_fret)
The second prediction is 2 (i.e. drums_Snare)
…
The 2386th prediction is 18 (i.e. guitar_9th_fret)

Submission to Kaggle

Step 1. Register an account in Kaggle

Step 2. Join the challenge. Invite code is below:

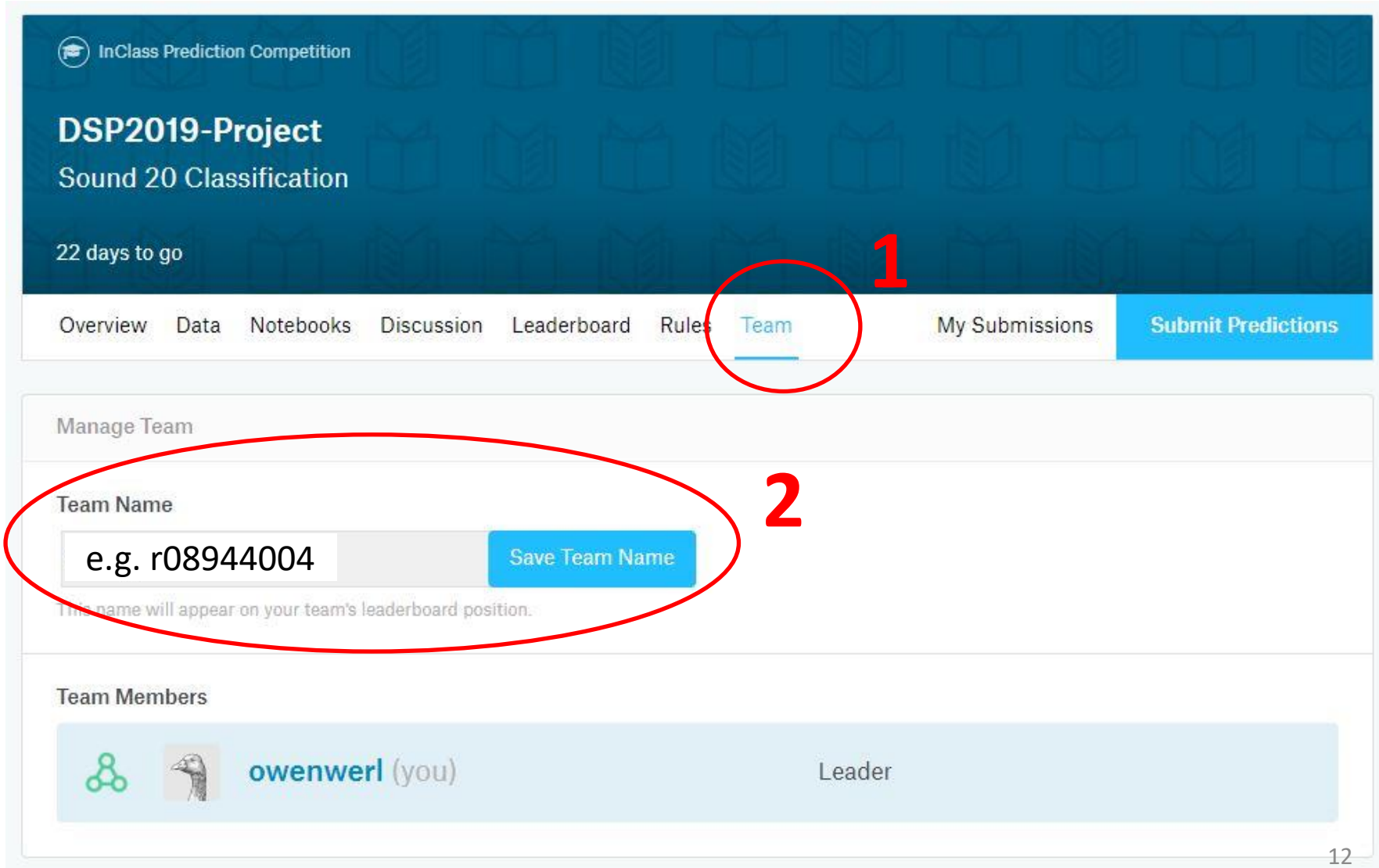
https://www.kaggle.com/t/ad4e41b811764190b634885a68c97300

Challenge Main Page: https://www.kaggle.com/c/dsp2019-pj2
Note. Anyone who joins this challenge will be able to view the main page.

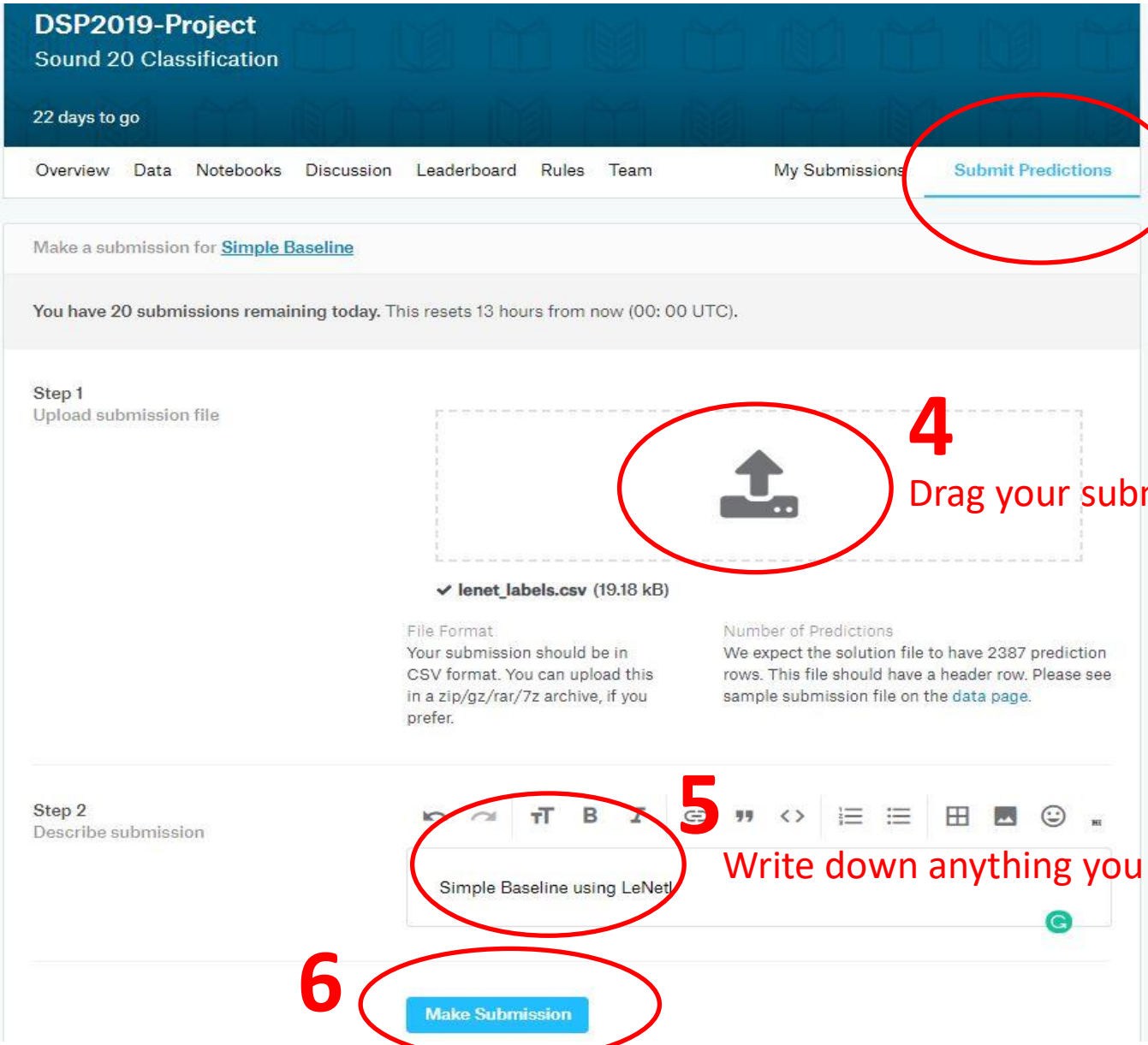
Step 3. Change **[Team Name]** to your student id

Step 4. Submit predictions to finish the upload process

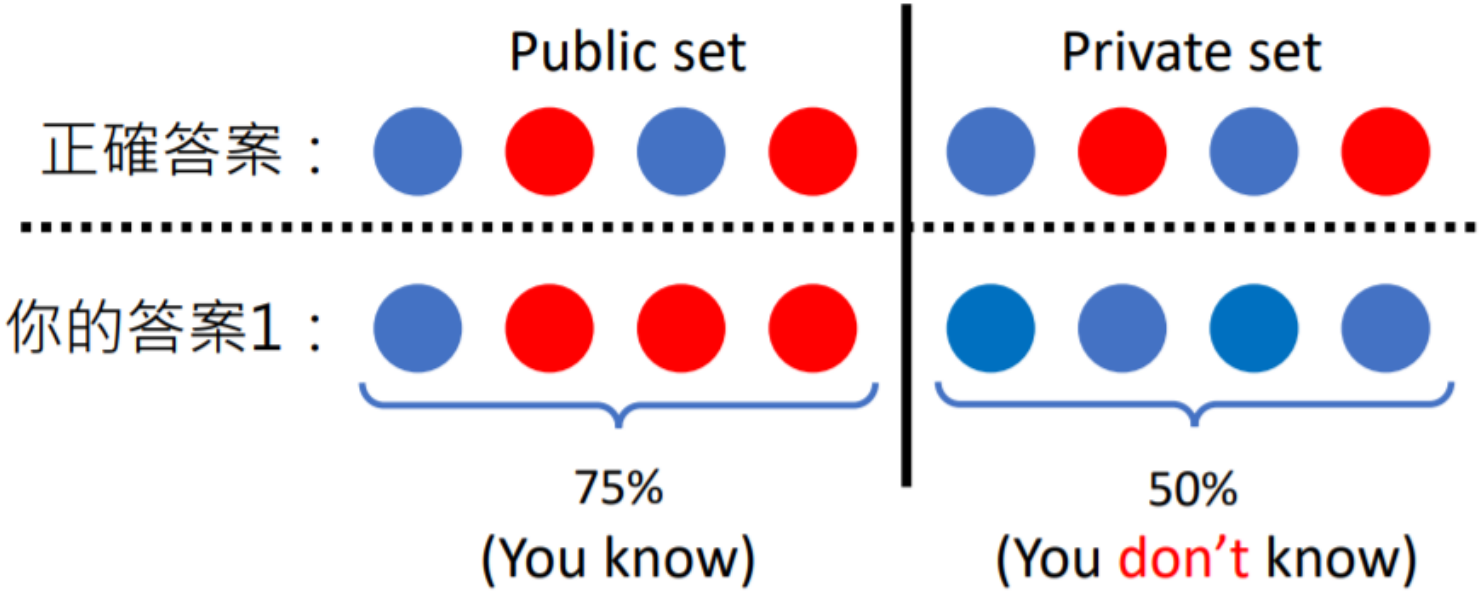
Change your **[Team Name]** to your student id



Press **[Submit Predictions]** to finish the upload process



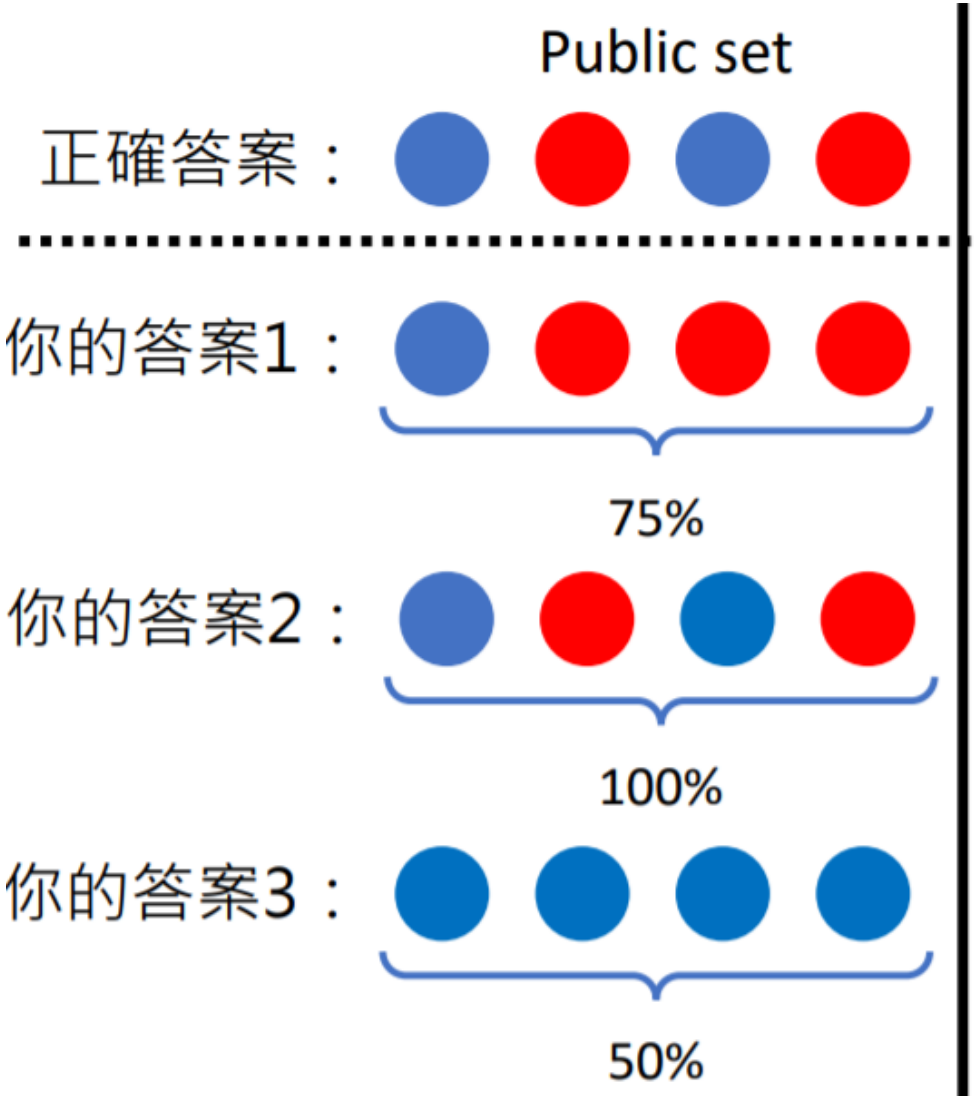
How Kaggle grades your submission



- 作業提供的testing set由public set與private set組成
- 在作業繳交期間，你只能看到public set上的正確率；競賽結束，才能看到private set的正確率
- 在這份作業中，我們將會以模型在public set跟private set的表現作為成績計算的標準
- 注意：每日有20次的上傳限制

Ref. ML 2019 assignment slide provided by Hung-yi Lee

How Kaggle grades your submission

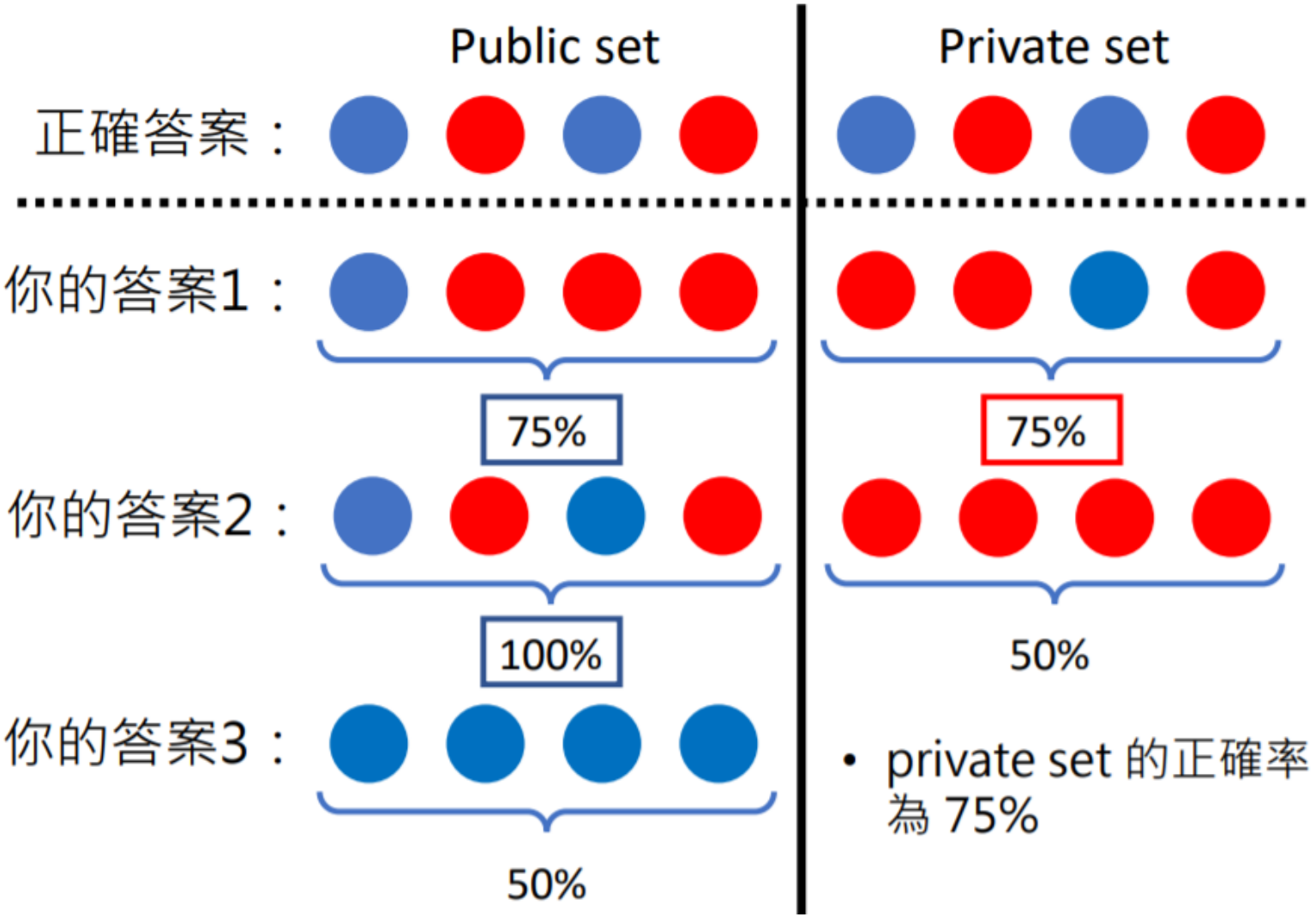


- Kaggle 預設用 public set 正確率最高的兩個答案去算 private set 的成績
- 可以自行決定用哪兩個答案去算 private set 的成績

To learn more, please see:
https://www.kaggle.com/c/dsp2019-pj2/submissions
Ps. Anyone who joins the challenge will be able to view the page.

Ref. ML 2019 assignment slide provided by Hung-yi Lee

How Kaggle grades your submission



Ref. ML 2019 assignment slide provided by Hung-yi Lee

Project Report

- Prepare a report which contains at least the follows:
  - Settings for generating spectrograms (at least **2** settings)
    - Please write down the experimental settings you use. E.g.
      - Exp 1. Generate a **magnitude spectrum** using the hanning window function with a FFT length of 256 and overlapping of 50% between segments
      - Exp 2. Generate a **phase spectrum** using the hanning window function with a FFT length of 256 and overlapping of 50% between segments
  - Settings for your neural network (at least **1** setting)
    - Please include implementation details like architecture(LeNet/VGGNet/…), optimizer(Adam/SGD/…), initialization, learning rate, etc.
    - Report the validation accuracy based on different settings. E.g.
      - Evaluate LeNet model with spectrograms from Exp 1 and Exp 2
  - What you have learned for this project. E.g.
    - Difficulties you encounter, interesting things you find, or special techniques you apply.
- Student with high performance or interesting finding on the report will be invited to make a oral presentation on **2020.01.02**

How to hand in your homework

1. Please send a mail with a **zip** file to TA (iis.sinica.1518@gmail.com)
   - Mail title: DSP2019_ FP_[student_id] (e.g. DSP2019_FP_r08944004)
   - Attached filename: [student_id].zip (e.g. r08944004.zip)
   - It should have at least the following items:
     - Source code and your pretrained model
       - **All** source codes (training, testing, etc.) you use
       - **Don't** hand in generated spectrogram images but a program that produces them
       - To test your result, please develop a **program that load your pre-trained model to make an inference with spectrograms**. TA will assume that **your input is validation data and output is validation accuracy.**
       - **Explain how to execute your program clearly**
     - Electronic files of your report
       - No longer than 4 pages
       - It must be a **pdf** file. Please name it [student_id].pdf (e.g. r08944004.pdf)

2. Submit your predictions to Kaggle judge system.
   - Remember to upload because your partial score is **graded by the judge system**.

- If you have any questions, please mail to TA
- Due on **2019.12.26 2:00 p.m.**

Late Submission

- Late work will incur the following penalties
  - Deduct 20% per day, up to 3 days
  - late work after 2019.12.29 2:00 p.m. will not be accepted

$$final\_score(n, score) = \begin{cases} score, & n = 0 \\ score * (1 - 0.2 * n), & 0 < n \leq 3 \\ \mathbf{0}, & n > 3 \end{cases}$$

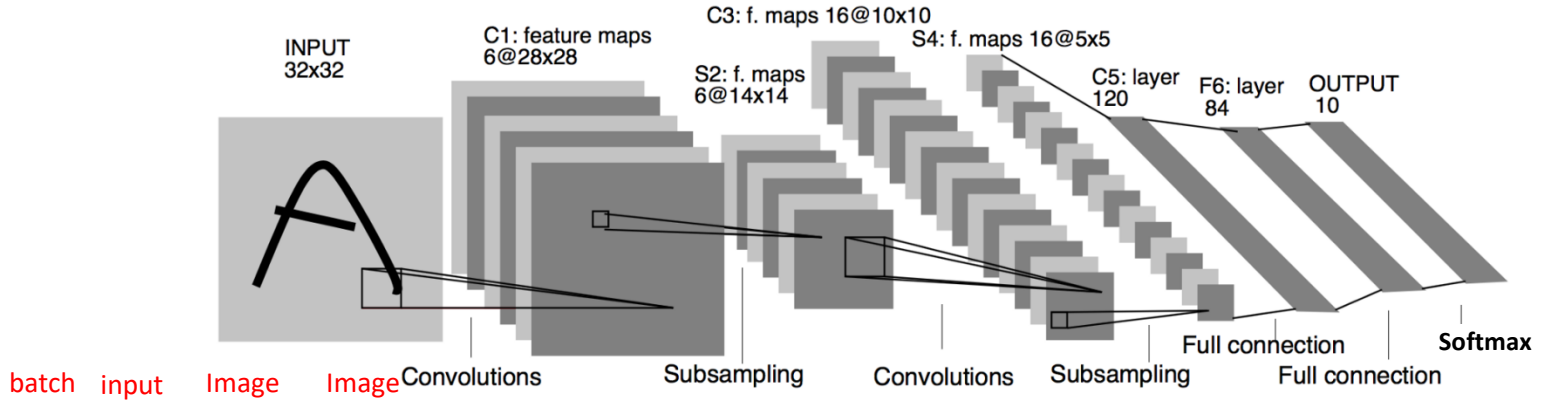
where $\boldsymbol{n}$ *is delaying days and* $\boldsymbol{score}$ *is your original score.*

USEFUL TIPS

Helpful Libraries

- To finish this project, the tools you may use:
  - Scipy
    - Generate spectrograms via scipy.signal.spectrogram
  - Librosa
    - Generate spectrograms via librosa.feature.melspectrogram
  - Pytorch
    - Train and test a neural network

Example. LeNet-5 Architecture



batch size, input channel, Image height, Image width

INPUT **(Nx 1 x32x 32)** ->

output channel, input channel, kernel height, kernel width

CONV with (6, 1, 5, 5) kernels, stride=1, ReLU**(Nx6x28x28)**->

MAXPOOL with (2, 2) window, stride=(2,2), **(Nx6x14x14)** ->

CONV with (16,6,5,5) kernels, stride=1, ReLU **(Nx16x10x10)** ->

MAXPOOL with (2,2) window, stride=(2,2), **(Nx16x5x5)** ->

FLATTEN to **(Nx400)** ->

FC with weight (120, 400), ReLU **(Nx120)** ->

FC with weight (84, 120), ReLU **(Nx84)** ->

FC with weight (10, 84), Softmax **(Nx10)**

FAQ

- 如何載入訓練資料?
- 如何產生預測結果?
- 如何使用前處理?
- 如何載入或儲存你訓練好的模型?
- 如何可視化預測結果?
- 如何訓練模型?
- 如何計算validation accuracy?
- 給定n張圖片，如何產生預測結果?

以上常見問題，助教提供一個簡單的範例程式(數字1,2,3的辨識)供各位參考，建議同學先行參考後再進行本次作業。
範例連結詳見：https://www.kaggle.com/c/dsp2019-pj2/data
※若無法看到連結，請先註冊成為Kaggle會員並加入作業評分網站，方能閱覽。

FAQ (Cont'd)

Q: 我遲交沒辦法上傳Kaggle怎麼辦?

請額外繳交你的結果檔案(*.csv)，並與作業要求之一切檔案一併mail給助教。

Q: 我看不到Private Set上的結果?

在作業截止日前，你只能看到Public Set上的結果；Private Set上的結果會在2019/12/26下午2點後公布結果。

Q: 可以提供baseline嗎? 比baseline低會 0 分嗎?

baseline請見Leaderboard，baseline只是參考，比baseline低並不會0分。