DSP Final Project 2020

Bearing health condition classification

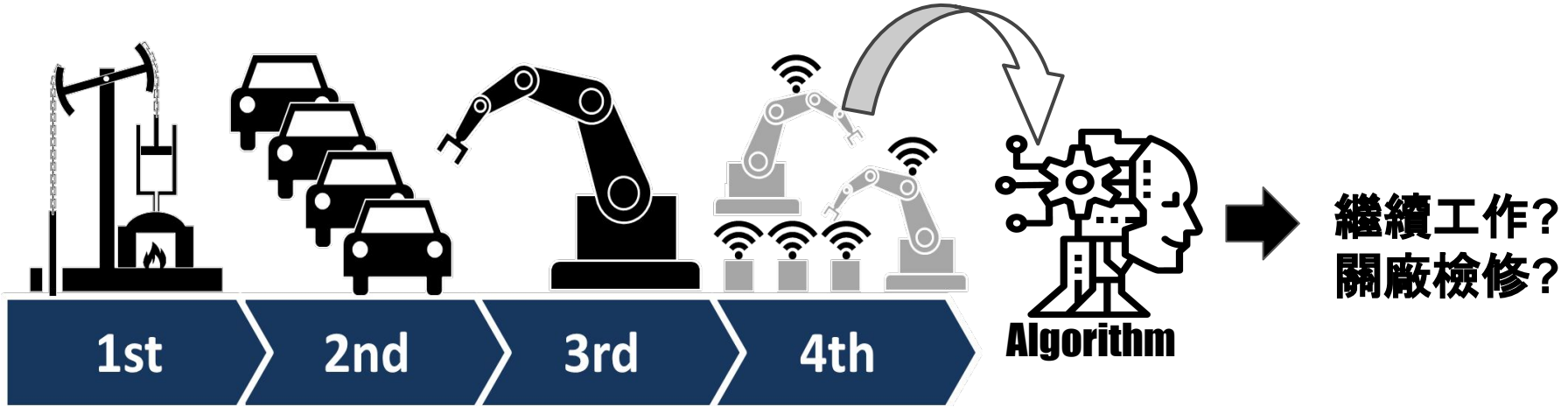
軸承健康狀態分類

Deadline: 2021/01/11 (Mon.) 14:00

TA: Timmy S. T. Wan 萬世澤

Industry 4.0 and big data

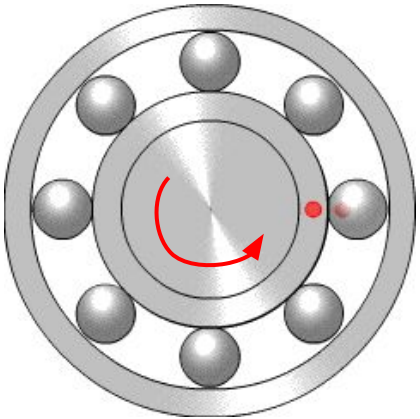
- Industry 4.0 including IoT, AI, robotics, etc., is the trend towards **automation and data exchange** in manufactruing technologoies.
- Big data is the one of core components to realize Industry 4.0.
  - Collect diversified time series generated at a high speed by IoT sensors.
  - **Analyze these data (E.g. monitoring system health status)** to support decision making.



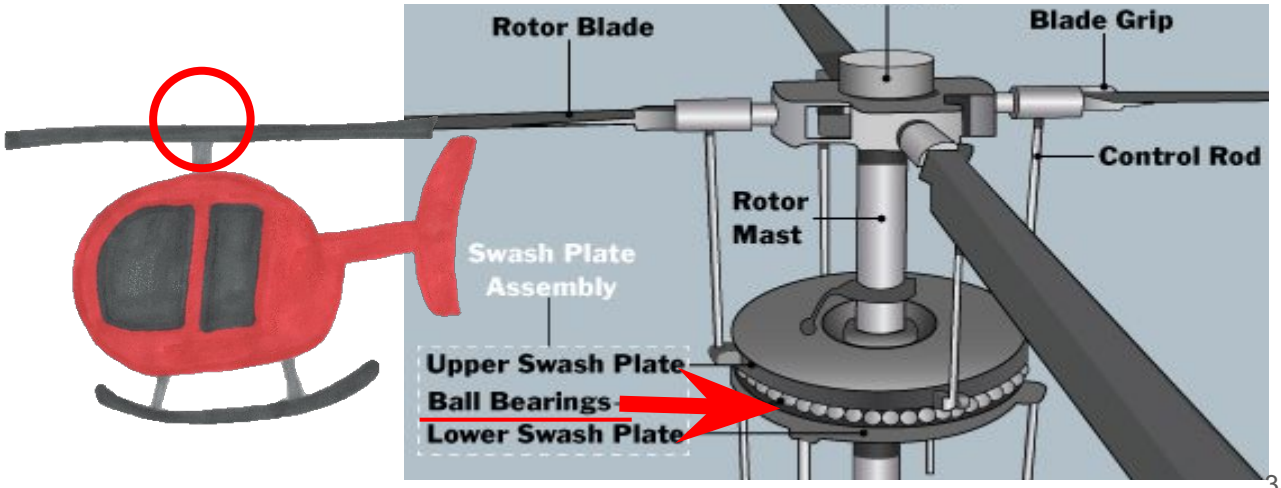
Modified from [Desafios do Financiamento da Previdência na Era do Trabalho 4.0]

Bearing 軸承

- A device that is used to enable rotational or linear movement, while reducing friction(摩擦力) and handling stress. (left figure)
- Wide applicatioins. E.g. helicopter
- It's essential that monitoring bearing health condition to protect the machine from system failure caused by damaged bearing.

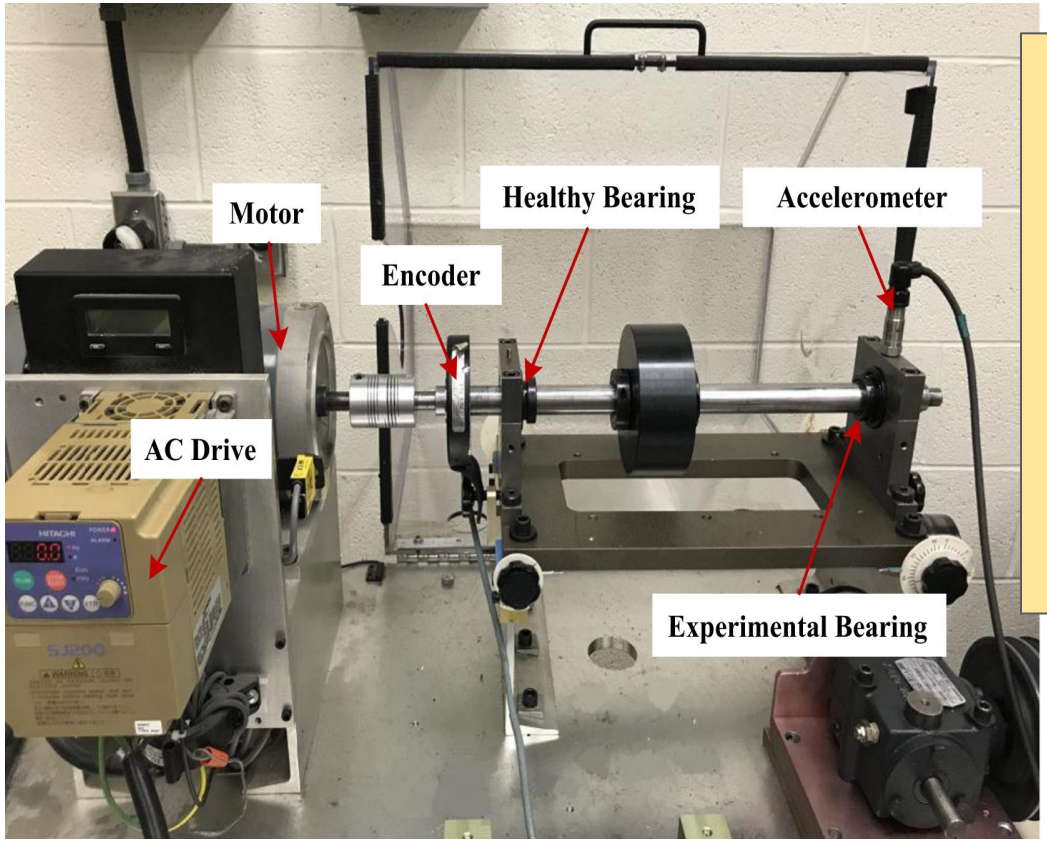


https://en.wikipedia.org/wiki/Ball_bearing



https://science.howstuffworks.com/transport/flight/modern/helicopter5.htm

Machinery Fault Simulator (MFS) 機器故障模擬器



- Collect vibration data using accelerometer (加速規)
- Measure the shaft rotational speed using encoder (速度編碼器)

- The health condition of experimental bearing can be identified by observing both the vibration data and rotational speed data.

Problem definition

- Given both the vibration and rotational speed data for any bearings as inputs, please predict theirs bearing health condition.

- The health condition can be classified into 4 types:
  (1) Health
  (2) Inner fault
  (3) Outer fault
  (4) Ball fault

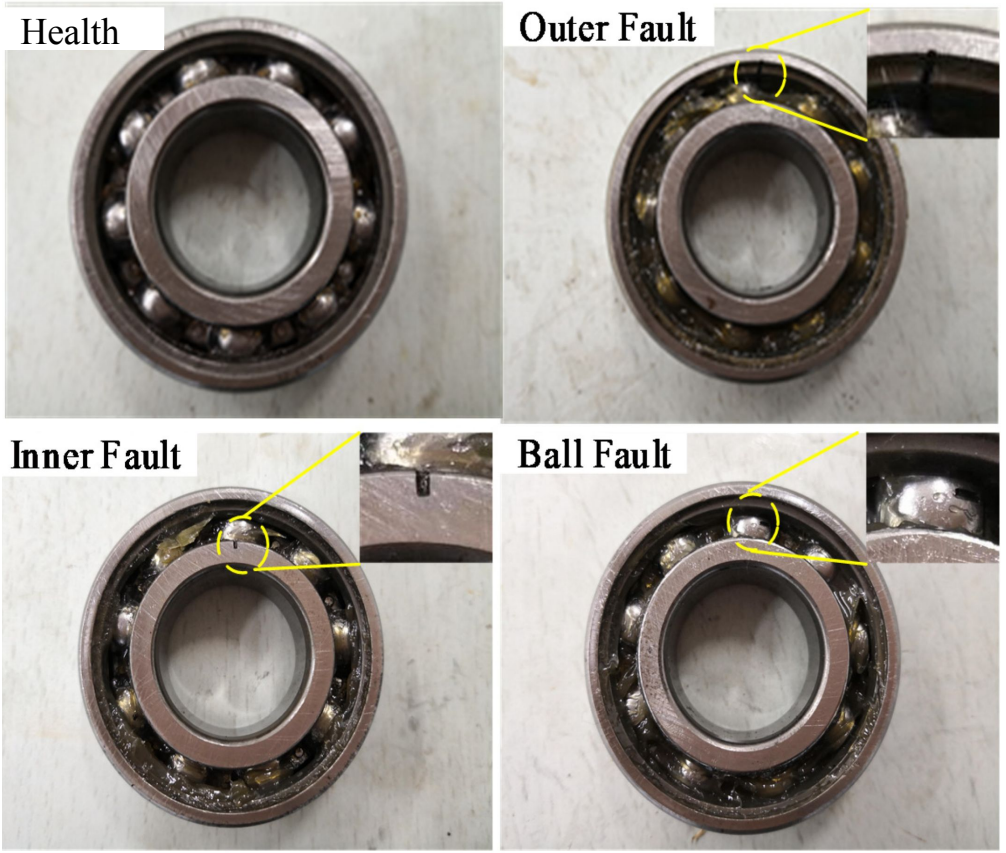
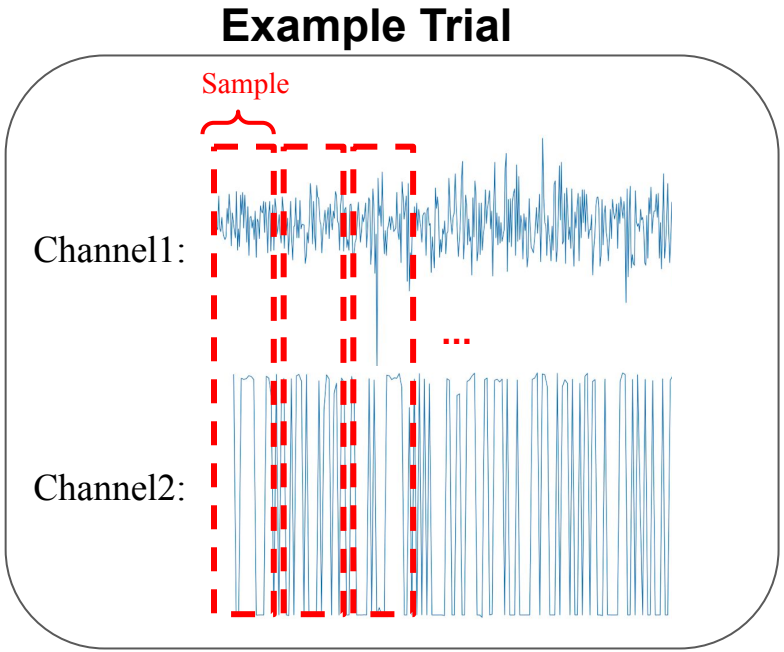


Image courtesy of
"Bearing Fault Diagnosis Based on Improved Convolutional Deep Belief Network"

Data description

- Include 4 health condition types: Health, Inner fault, Outer fault, Ball fault
- There are 12 trials for each type respectively.
  - All trials are collected at 200 kHz for 10s
    - 2M data points for each trial
  - Each trial contains 2 channels:
    - Channel 1: Vibration data
    - Channel 2: Rotational speed data
- Each trial is split into multiple samples.

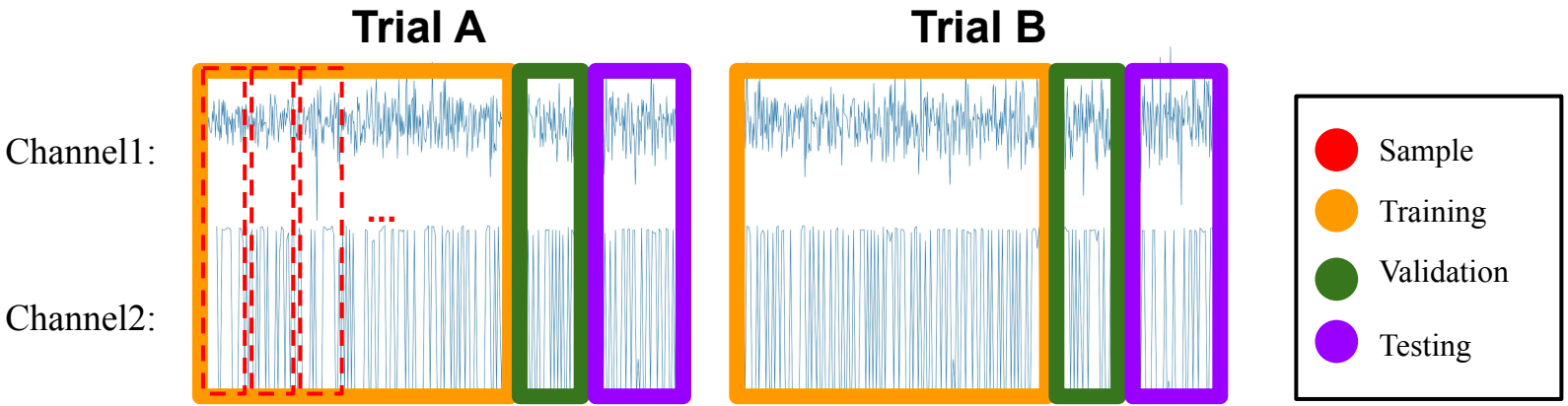


Length of each sample under each type

Class label	Class name	Sample length	Number of samples
0	Health	20000 (0.1s)	1200
1	Inner fault	20000 (0.1s)	1200
2	Outer fault	20000 (0.1s)	1200
3	Ball fault	20000 (0.1s)	1200

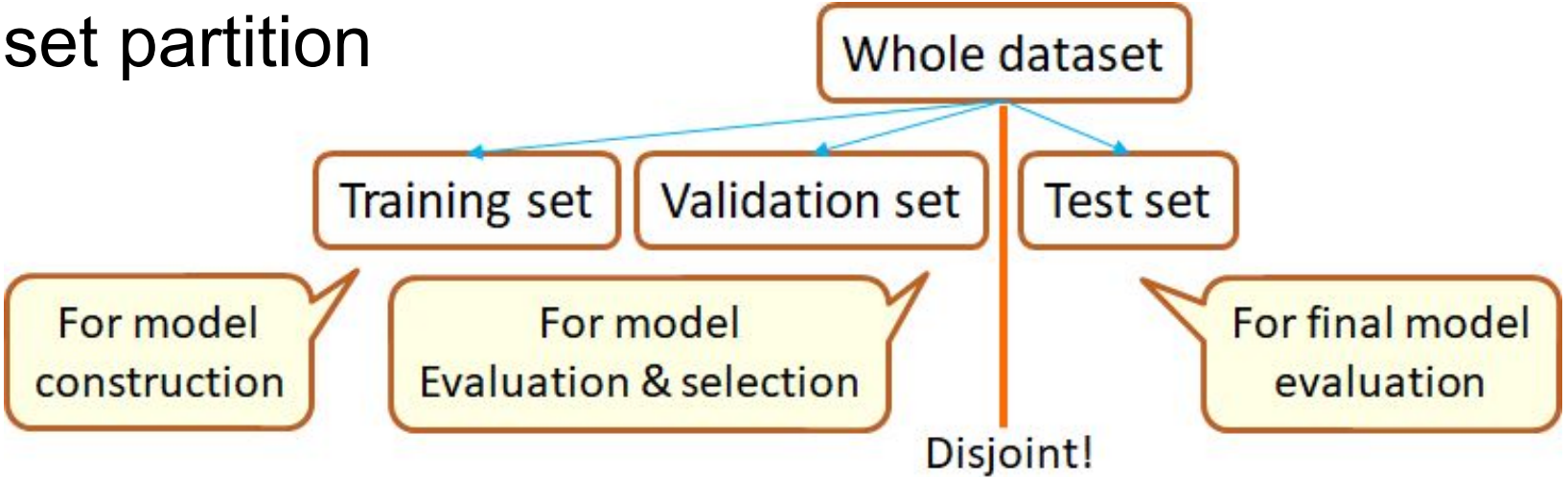
Dataset

- For each sample in each trial, we take:
  - 70% of it for training
  - 10% of it for validation
  - 20% of it for testing



The dataset contains 3360 training samples, 480 validation samples and 960 testing samples in total.

Dataset partition



- Develop a classifier using training set which performs good on validation set. Then, TA will finally check the results of your classifier on the test set, which is not shared with you.
- In some cases, the final classifier can be trained with both training and validation set after model selection. But, in our homework, the validation set should only be used for validation instead of training.

Modified from Fintech 2019 lecture slide provided by Prof. J.-S. Roger Jang

To learn more, please see here.

Dataset structure

- Download the DSP2020_FP.zip. (258 MB)
  https://drive.google.com/file/d/1GsguXWx_rAHQI1uEDraAADIA35f_U8Ym/view?usp=sharing
- Get 6 files by extracting the zip file:

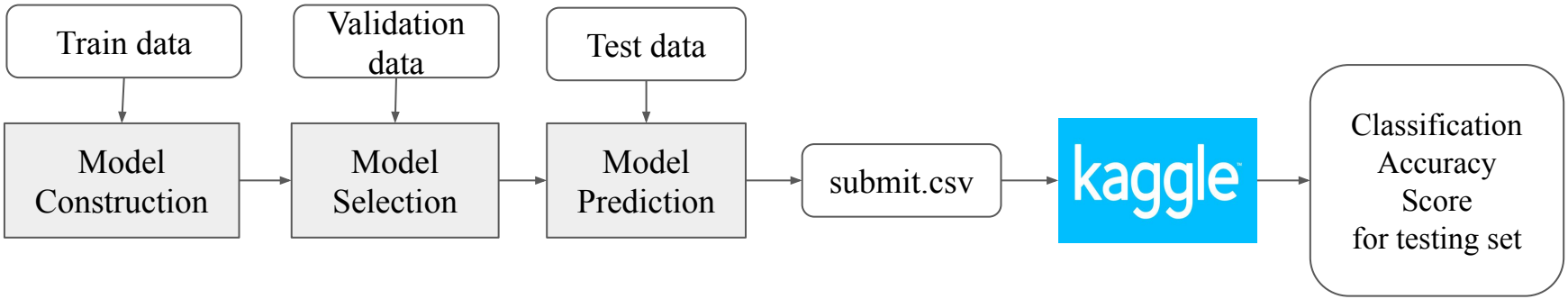
1. traindata.npy
2. trainlabel.npy
3. validationdata.npy
4. validationlabel.npy
5. testdata.npy
6. example_submission.csv

```
import numpy as np
sample, target = np.load("traindata.npy"), np.load("trainlabel.npy") # load training sample and label
print(sample.shape)#(3360,2,20000)=(#sample, #channel, #length)
print(target.shape)#(3360,)=(#sample)
# The label of sample[0] is target[0].
# 即 traindata.npy 的第 n 筆訊號的標籤是 trainlabel.npy 的 第 n 筆資料 (validation 亦比照辦理)
# 訓練完成後, 依 testdata.npy 的順序, 依序完成預測結果
```

- *.npy files are all ordered lists.
- Encounter “UnicodeError: Unpickling a python object failed: UnicodeDecodeError”, please run np.load("traindata.npy",encoding='latin1') instead.

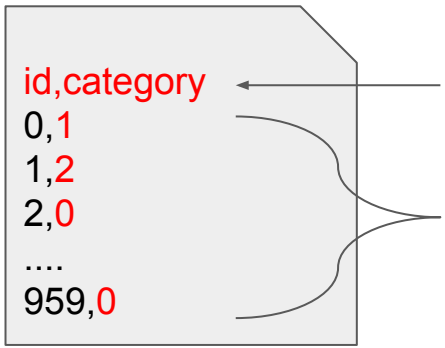
Requirement

1. Implement an algorithm for identifying bearing health condition
   a. **Any approaches (neural nets, statistical methods, etc.) would be accepted.**
   b. **The "single" model size should not exceed 20MB. (請控制模型大小)**
2. Show the classification accuracy on the given validation set.
3. Submit predictions for the testing set to Kaggle judge system.
   a. **Don't train the final model with both training and validation data. (不要拿驗證集來訓練)**
4. Prepare a report to describe your experimental settings, model configurations or even interesting findings.



Prediction format

- Your prediction results should follow the list order of samples in *testdata.npy*
- The prediction for each sample should be an integer in 0~3
- Store your results as *.csv
- Follow the format below:



add the header named **“id,category”**

The first prediction is 1 (i.e. inner fault)
The second prediction is 2 (i.e. outer fault)
The third prediction is 0 (i.e. health)
…
The last prediction is 0 (i.e. health)

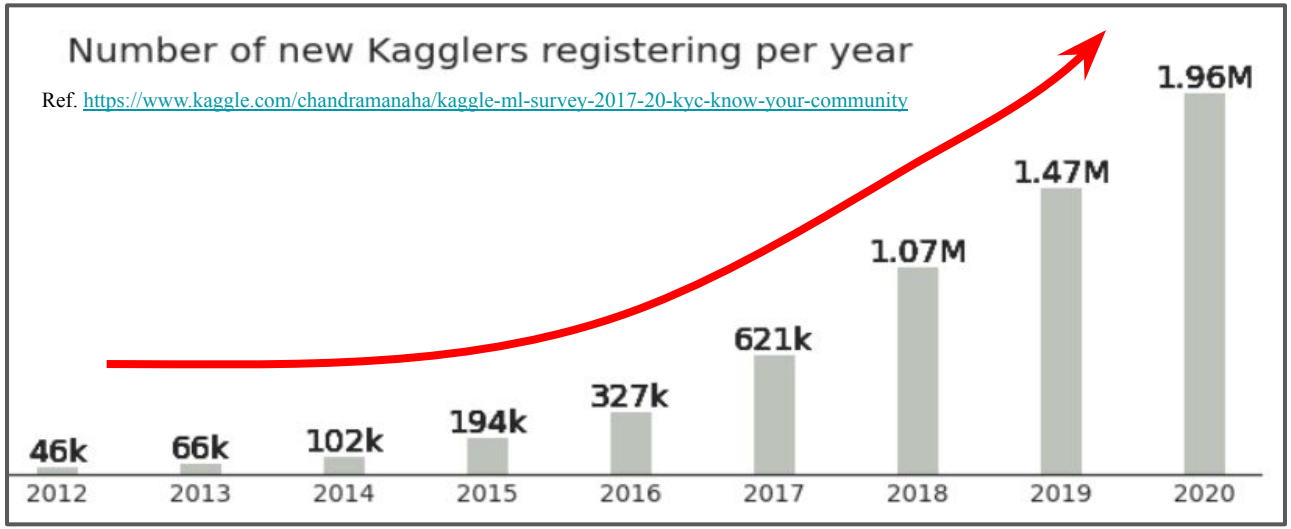
逗號前後不需要空白！

Class Label	Class Name
0	Health
1	Inner fault
2	Outer fault
3	Ball fault

See *example_submisson.csv* for more details!

kaggle

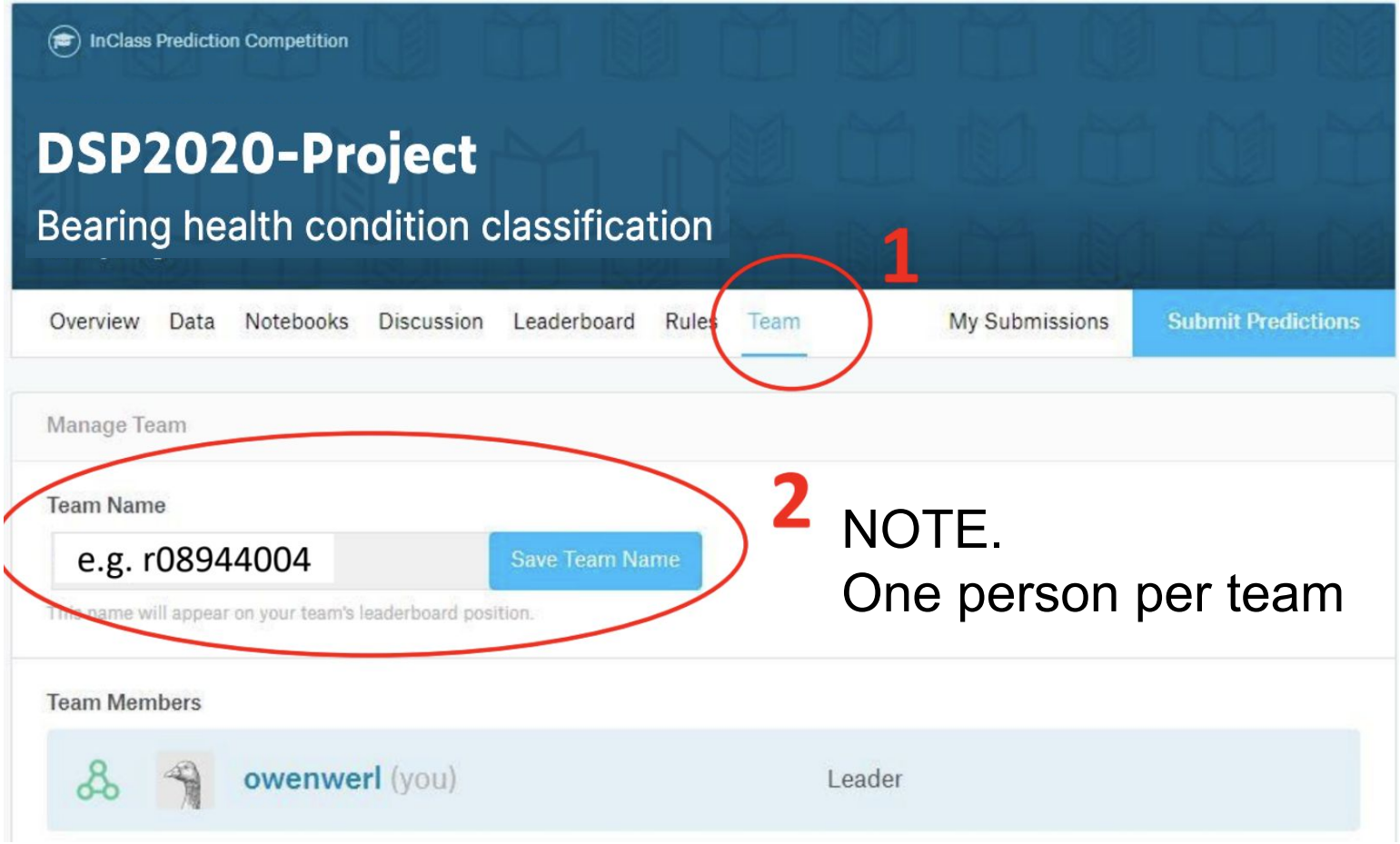
1. An online judge system for data science competiton use.
2. Also can be used for educational purposes like classroom competitions.
3. TA will score your testing predictions using Kaggle for fair judgements.



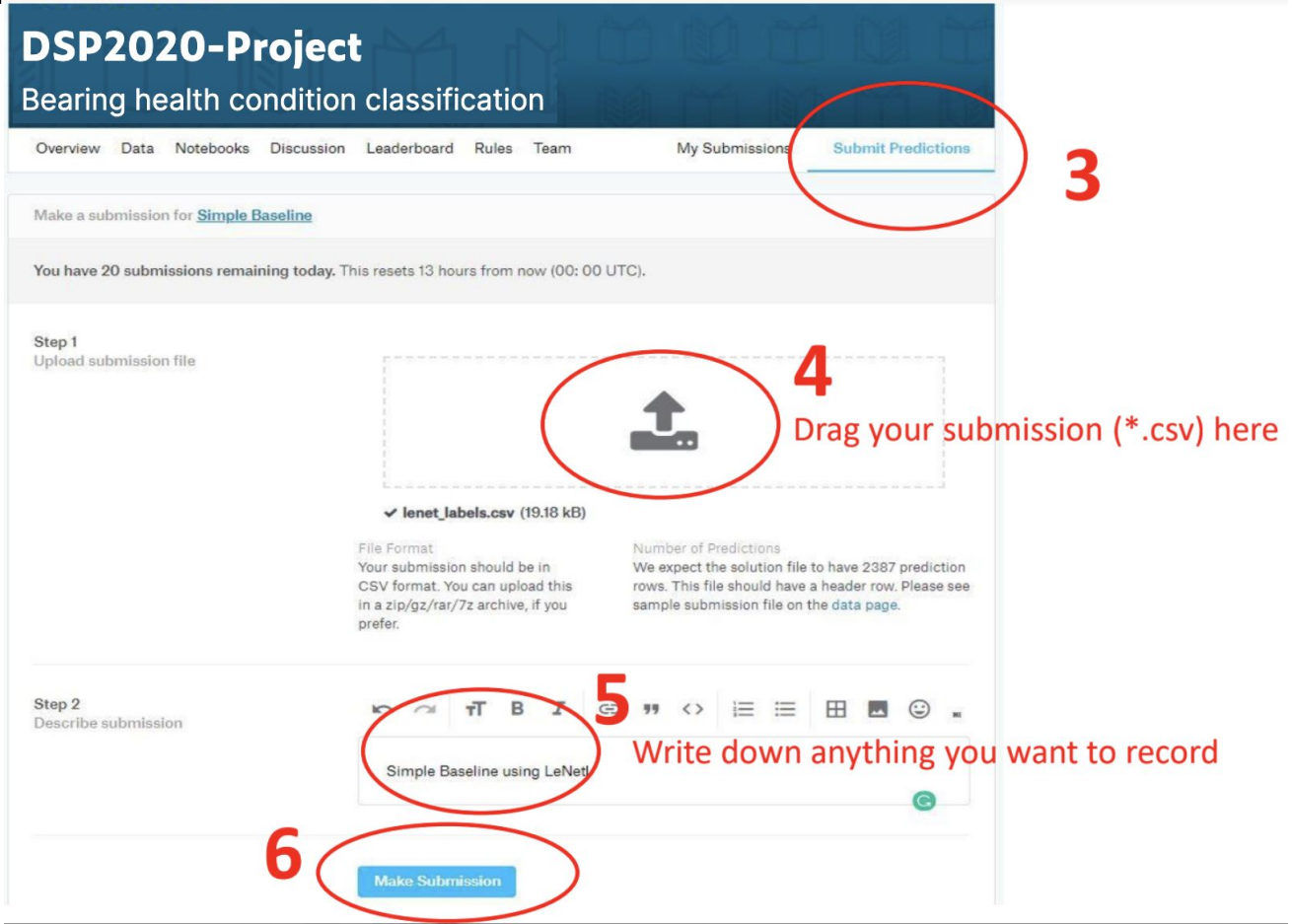
A submission guide to Kaggle

- Follow the instructions to submit:
  a. Register an account in Kaggle
  b. Join the challenge. Your invited code is shown below: https://www.kaggle.com/t/d0f59e5520f840139e10b65e9a93f21d
  c. Change [Team Name] to your student id
  d. Submit predictions!
- Challenge page: https://www.kaggle.com/c/dsp2020fp
  Note: Anyone who joins this challenge will be able to view the main page.

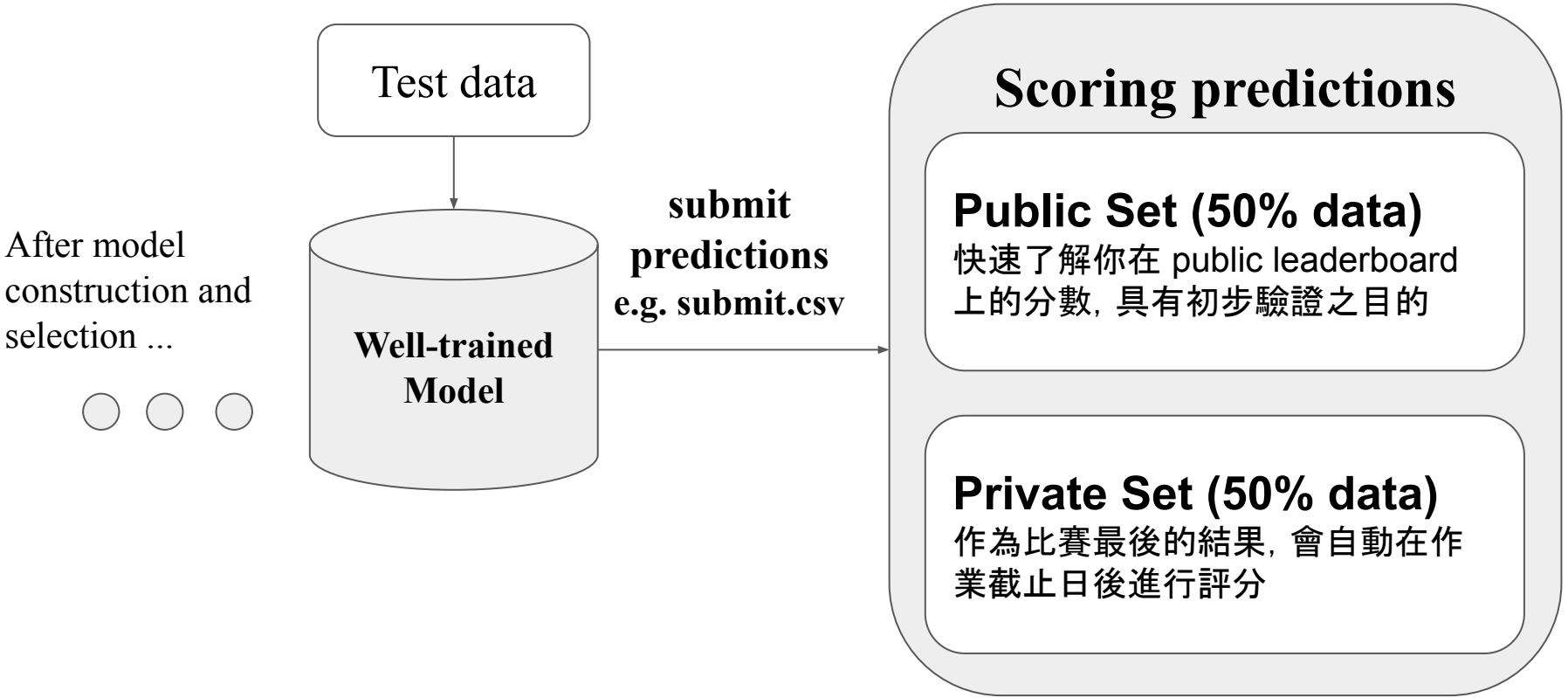
Change your [Team Name] to your student id



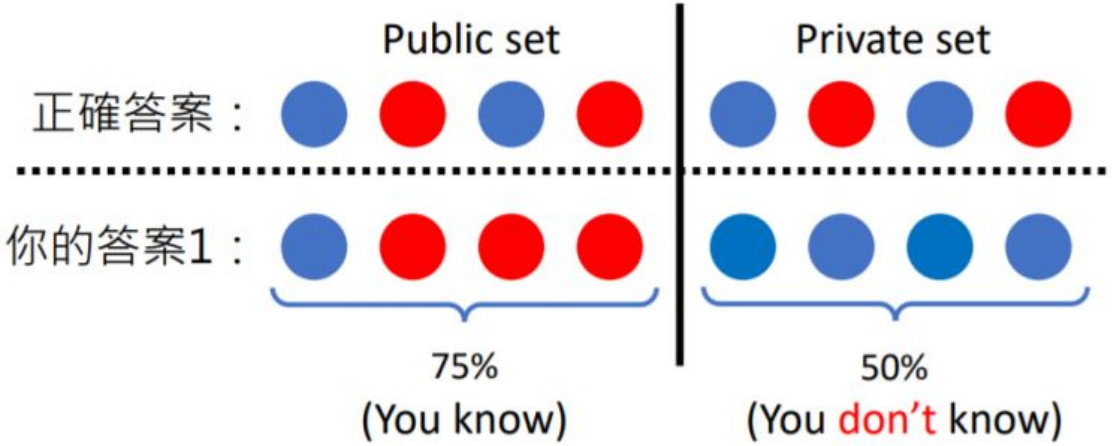
Press [Submit Predictions] to finish the upload process



How Kaggle grades your submission (1/4)



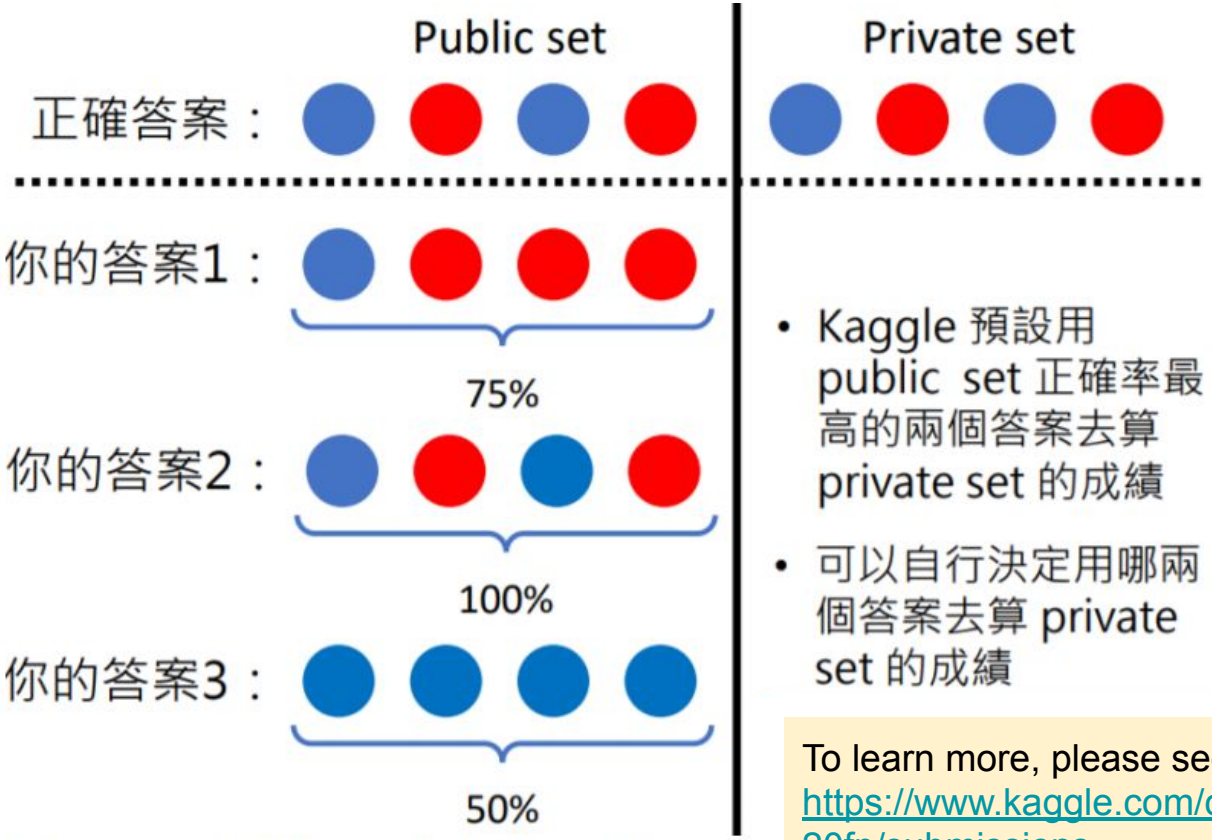
How Kaggle grades your submission (2/4)



- 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:每天有 8 次的上傳限制**

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

How Kaggle grades your submission (3/4)

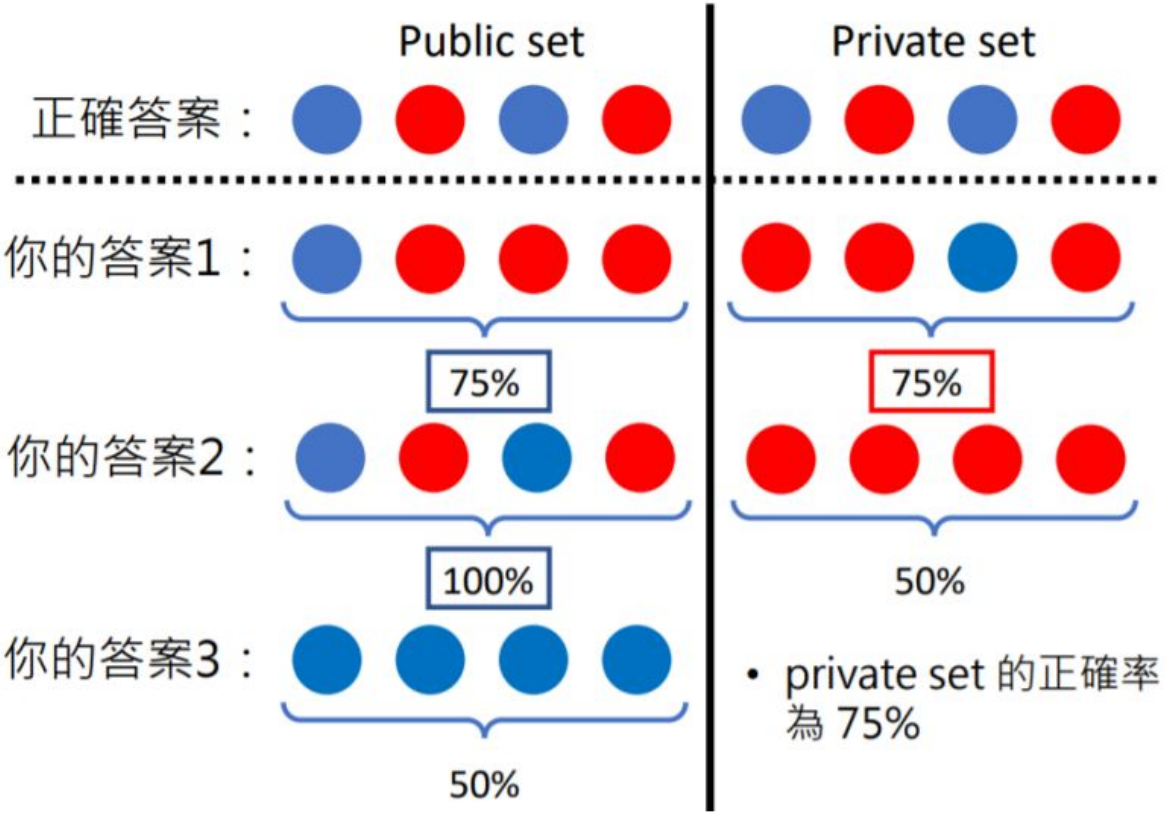


- Kaggle 預設用 public set 正確率最高的兩個答案去算 private set 的成績
- 可以自行決定用哪兩個答案去算 private set 的成績

To learn more, please see https://www.kaggle.com/c/dsp2020fp/submissions

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

How Kaggle grades your submission (4/4)



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

Project Report

- The report should cover the following details:
  - Experimental settings
    i. **Input format and preprocessing details**
    E.g. If the input is spectrogram, please describe window size, overlap, etc.
    ii. **Method (open question)**
    E.g. Train a CNN with 3 layers. Minimize cross-entorpy loss using SGD optimizer with learning rate 0.1, etc.
    iii. **Model selection**
    E.g. Submit the model with highest validation accuracy.
  - Report the classification accuracy based on the given validation set.
  - What you have learned for this project.
    E.g. Difficulties you encountered, interesting finding, or special techniques you try.
- **Top-3 winners or people with interesting findings are invited to make an oral presentation on 2021.01.14.**
  - 口頭報告格式不拘，使用書面 內容進行報告亦可

Homework submission

1. Send a mail with a zip file to TA
   a. Mail title: dsp2020_FP_[STUDENT_ID]. E.g. dsp2020_FP_r08944004
   b. Attached filename:
   c. Compress the following items into a zip file named dsp2020_FP_[STUDENT_ID]:
      i. **All** source code (train, test, analyze, etc.) for **reproducibility**
      ii. Program instruction file (README.md)
      iii. Report file: (1) pdf format (2) no longer than 6 pages
2. Submit your predictions to Kaggle.
   a. Remember to upload because your partial score is graded by the judge system.
3. If you have any questions, please mail to iis.sinica.1518@gmail.com
4. Due on 2021.01.11 14:00

Grading policy

1. Project report (80%)
   a. Experimental settings (30%)
   b. Validation accuracy (10%)
   c. What you've learned (40%)
2. Testing set classification accuracy (20%)
   a. Public set classification accuracy (10%)
   b. Private set classification accuracy (10%)

Public / Private set 評分標準 (10%)	
排名分布	得分
前1%~20%	10
前20%~40%	8
前40%~60%	6
前60%~80%	4
前80%~100%	2
未繳交者	**0**

Late submission

- Late work will incur the following penalties.
  - Deduct 20% per day, up to 3 days
  - Late work after 2021.01.14 23:59 will not be accepted!

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

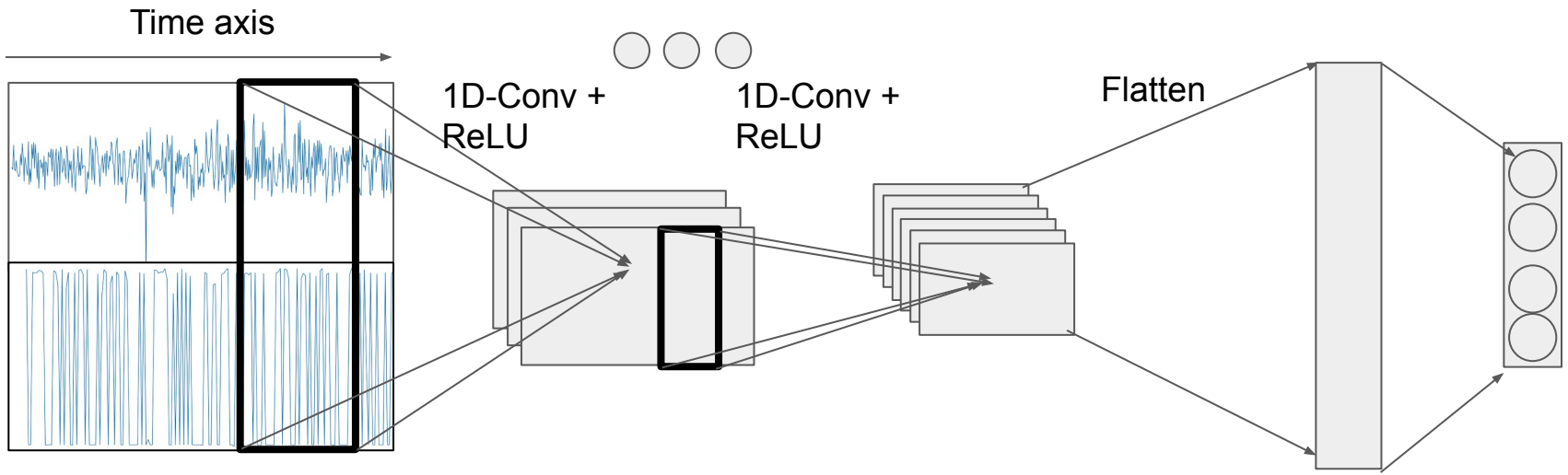
where $\boldsymbol{n}$ *is delaying days and* $\boldsymbol{score}$ *is your original score.*

FAQ

1. Q: 我遲交沒辦法上傳Kaggle怎麼辦?
   A: 該部分將不予評分。
2. Q: 我看不到Private Set上的結果?
   A: 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 2021/01/11 下午 2 點後公布在 Kaggle 網站上。
3. Q: 可以提供baseline嗎? 比baseline低會 0 分嗎?
   A: baseline請見Leaderboard, baseline只是參考,比baseline低並不會0分。

Useful Tips

Baseline: 1-dimensional convolutional neural net



Input raw signal with 2 channels

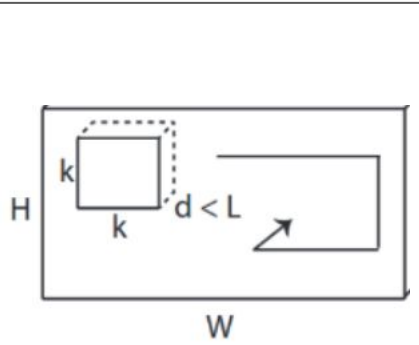
Achieve 64% in the public leaderboard:

- 4 conv layer + 1 fully-connected layer
- Train 10 epochs **(undertrained)**
- Input signals are normalized by Z-score respectively.

System: i5-5200U / 4GB / No GPU
Model size: 550KB
Elapsed Time: 217s for 10 epochs

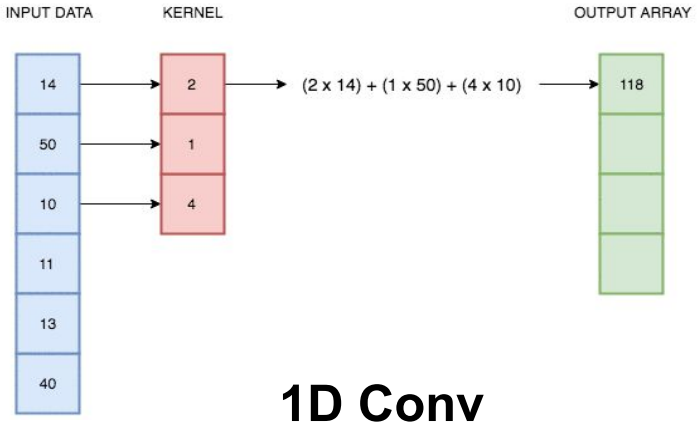
1-dimensional convolution (1-D Conv)

- Almost the same as 2-D Conv
  - 1-D moves in only one direction over time-axis.
  - 2-D Conv moves in two direction.
- Widely used in text analysis.



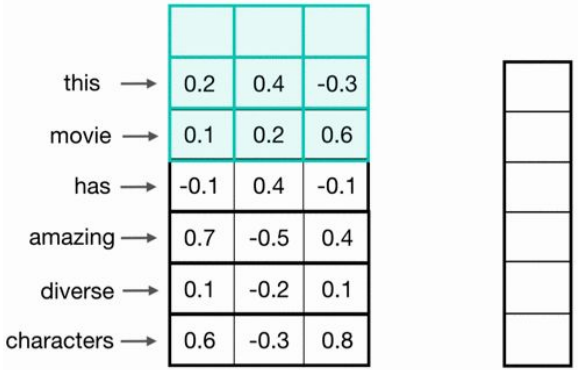
2D Conv

vs.



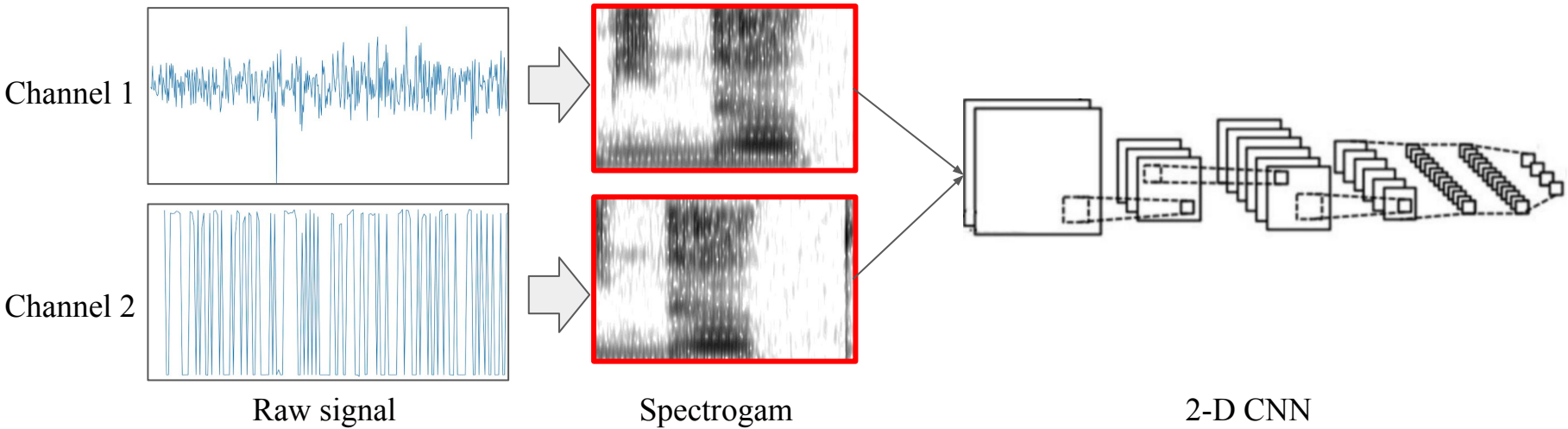
1D Conv

animation url: https://blog.floydhub.com/reading-minds-with-deep-learning/



Animation url:
https://cezannec.github.io/CNN_Text_Classification/

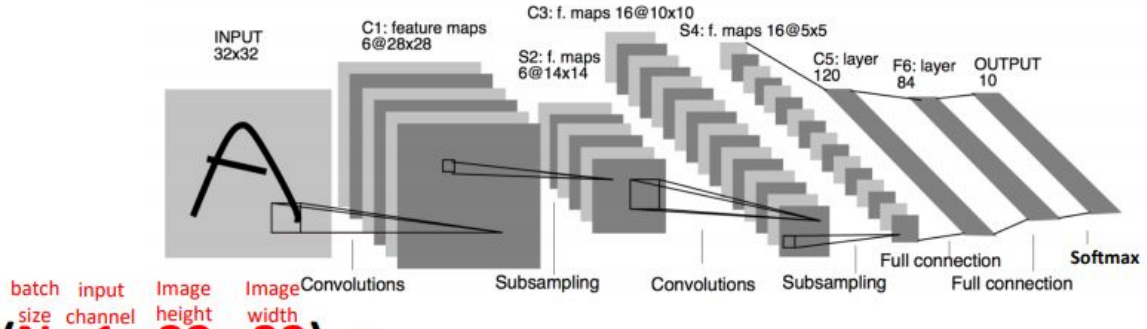
Other reference implementation: Spectrogram classifier

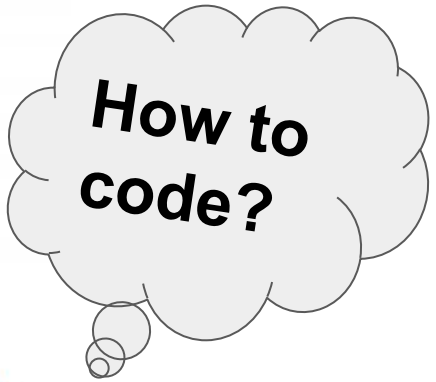


Spectrogram can be magnitude spectrogram, phase spectrogram, etc.

Modified from https://www.youtube.com/watch?v=dZVkygnKh1M

Classification using LeNet





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)**

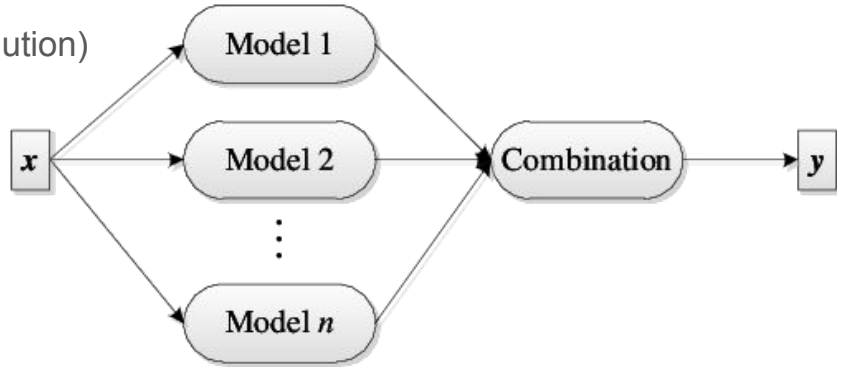
Tutorial: 1,2,3 Classification

1. 如何載入資料?
2. 如何產生預測結果?
3. 如何使用前處理?
4. 如何載入或儲存你訓練好的模型?
5. 如何計算validation classification accuracy?
6. [Extra.] 如何使用 1-D Convolution, 2-D Convolution?
7. [Extra.] 如何產生spectrogram?

關於以上問題，助教已準備一個教學程式供同學參考，請見
https://drive.google.com/file/d/1x0IXTtUQRpkKnJMSEvlKDmF3cejXyZOD/view?usp=sharing
或 https://www.kaggle.com/owenwerl/tutorial 。

Other useful techniques

- Preprocessing
  - Min-max scaling
  - Z-score scaling
  - log scaling (specific for spectorgrams)
  - ...
- Model
  - Convolutional Neural Network (e.g. 1D or 2D convolution)
  - Support Vector Machine (SVM) Classifier
  - ...
- Post-processing
  - Ensemble (e.g. voting, averaging)
  - ...



Helpful Libraries

- Pytorch
  - E.g. nn.Conv2d
    https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html
- Scipy
  - E.g. scipy.signal.spectrogram
    https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html
- Sklearn
  - E.g. sklearn.preprocessing.StandardScaler
    https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- Matplotlib (for data visualization)
- …
- **任何library、github source code只要在報告上註明出處，皆可使用!**