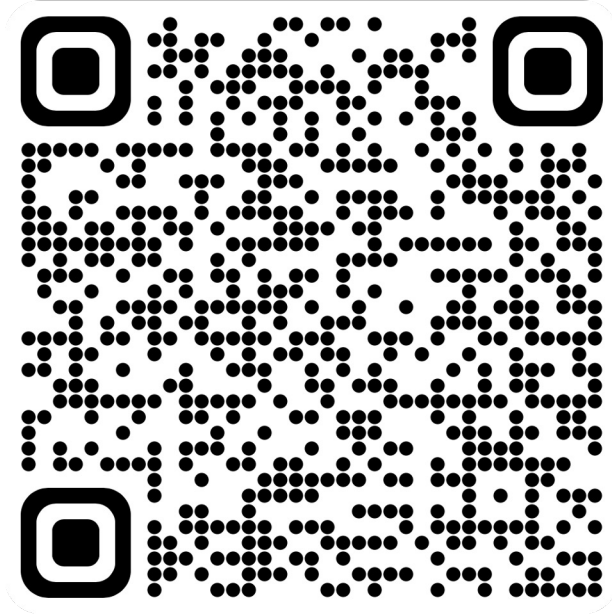


# 加簽表單



[Link](#)

---

---

# Machine Learning HW1

## COVID-19 Cases Prediction

ML TAs

[mlta-2023-spring@googlegroups.com](mailto:mlta-2023-spring@googlegroups.com)

---

---

# Outline

- Objectives
- Task Description
- Data
- Evaluation Metric
- Kaggle
- Grading
- Code Submission
- Hints
- Deadline
- Regulations
- Useful Links

# Objectives

- Solve a regression problem with deep neural networks (DNN).
- Understand basic DNN training tips e.g. hyper-parameter tuning, feature selection, regularization, ...
- Get familiar with PyTorch.

# Task Description (1/2)

- COVID-19 Cases Prediction
- Source: Delphi group @ CMU
  - A daily survey since October 2021 via facebook.

**Try to find out the data and use it to your training is forbidden.**



# Task Description (2/2)

- Given survey results in the **past 3 days** in a specific state in U.S., then predict the percentage of **new tested positive cases** in the 3rd day.



survey



positive cases

Day1



survey



positive cases

Day2



survey



positive cases

Day3

# Data (1/3) – Feature

- States (35, encoded to one-hot vectors)
- COVID-like illness (5)
  - cli, ili ...
- Behavior indicators (5)
  - wearing\_mask, shop\_indoors, restaurant\_indoors, public\_transit ...
- Belief indicators (2)
  - belief\_mask\_effective, belief\_distancing\_effective.

# Data (2/3) – Feature

- Mental indicator (2)
  - worried\_catch\_covid, worried\_finance.
- Environmental indicators (3)
  - other\_masked\_public, other\_distanced\_public ...
- Tested Positive Cases (1)
  - **tested\_positive (this is what we want to predict)**



## Data (3/3) – One-hot Vector

- One-hot Vectors

Vectors with only one element equals to one while others are zero.  
Usually used to encode discrete values.

If state code = AZ  
(Arizona)

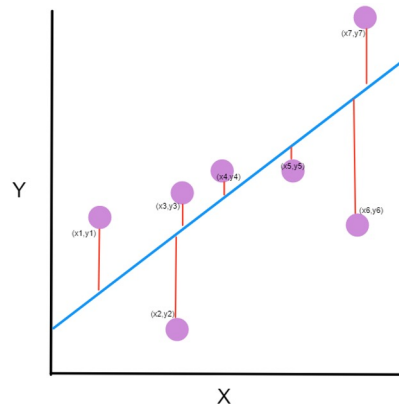
**one-hot encoding**



0	AL (Alabama)
0	AK (Alaska)
1	AZ (Arizona)
0	AR (Arkansas)
⋮	
0	WI (Wisconsin)

# Evaluation Metric

- Mean Squared Error (MSE)



$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

ground truth

your model (prediction)

# Kaggle (1/2) – Format

- Display name: **<student ID>\_<anything>**
  - e.g. b10901000\_public和private也差太多
  - For auditing, don't put student ID in your display name.

- Submission format: **.csv** file
  - See sample code


```
1 id,tested_positive
2 0,0.0
3 1,0.0
4 2,0.0
5 3,0.0
6 4,0.0
```

- [Kaggle Link](#)

# Kaggle (2/2) – Submission

- You may submit up to **5** results each day (UTC+8, AM 8:00)
- Up to **2** submissions will be considered for the private leaderboard

<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large3_684_compressed.pth, size = 201KB, params: 93139 (rabbit ensemble)	0.65059	0.66341	<input checked="" type="checkbox"/>
<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large3_676_compressed.pth, size = 201KB, params: 93139 (rabbit ensemble)	0.65282	0.65422	<input type="checkbox"/>
<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large2_669_compressed.pth, size = 222KB, params: 103623	0.65394	0.65254	<input checked="" type="checkbox"/>



remember to select **2** results for your final scores before the competition ends!

# Grading (1/5) – Introduction

- In this class, there are **15** assignments.
- Each has **10** points, only count the 10 assignments with the highest points.
- You don't need to do all the assignments. Choose the one you are interested in.

# Grading (2/5) – Introduction






- Most assignment includes leaderboard, gradescope, and code submission.
  - Leaderboard : Kaggle or JudgeBoi (our in-house Kaggle) competition
  - Gradescope : Answer some questions
  - Code submission : Submit the related code of each assignment via NTU COOL
- HW1 doesn't include gradescope.

# Grading (3/5) – Leaderboard

- simple (public) +1 pts
- simple (private) +1 pts
- medium (public) +1 pts
- medium (private) +1 pts
- strong (public) +1 pts
- strong (private) +1 pts
- boss (public) +1 pts
- boss (private) +1 pts
- code submission +2 pts

Total : 10 pts

# Grading (4/5) – Baseline Score

#	Team	Members	Score	Entries	Last
1	傷心咖啡店之歌		0.80786	4	15h
	---- boss baseline ----		0.81456		
	---- strong baseline ----		0.92619		
	---- medium baseline ----		1.15678		
	---- simple baseline ----		1.96993		



# Grading (5/5) – Bonus

- If your ranking in private leaderboard is top 3, you can choose to share a report to NTU COOL and get extra 0.5 pts.
- About the report
  - Your name and student\_ID
  - Methods you used in code
  - Reference
  - in 200 words
  - Deadline is one week later than code submission
  - Please upload to NTU COOL's discussion of HW1

[Report Template](#)

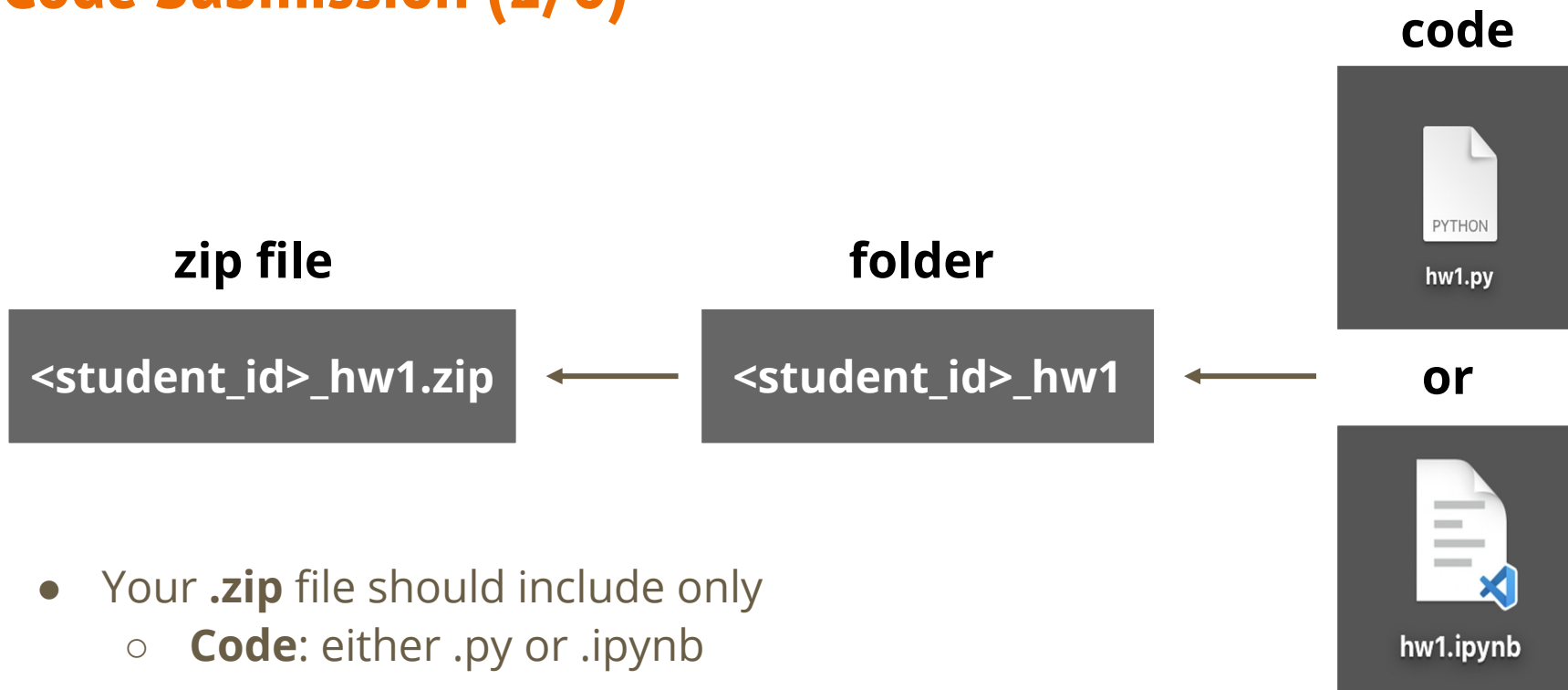
# Code Submission (1/6)

- NTU COOL
  - Compress your code and pack them into **.zip file**

**<student\_ID>\_hw1.zip**

- **Do not submit models and data**
- **Submit the code you chose in Kaggle (One of the best)**

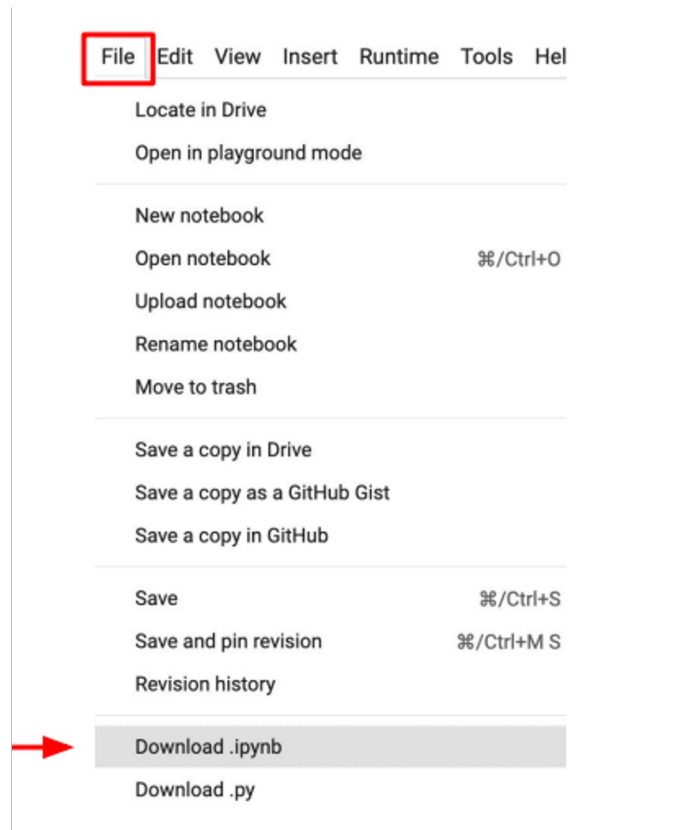
# Code Submission (2/6)



- Your **.zip** file should include only
  - **Code:** either .py or .ipynb

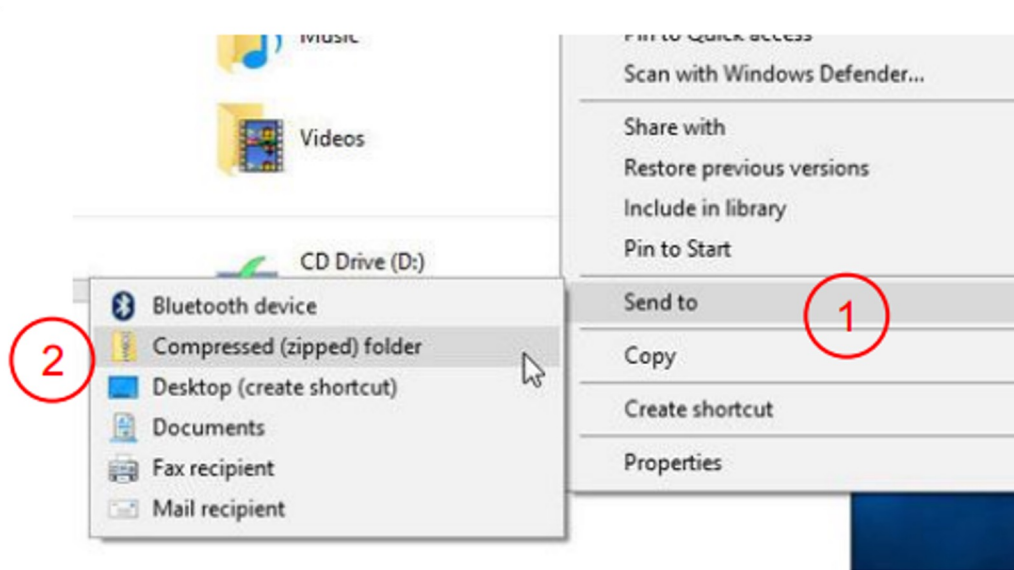
# Code Submission (3/6)

- How to download your code
- From Google Colab



# Code Submission (4/6)

- How to compress your folder?
- Method 1 (for Windows users)
  - <https://support.microsoft.com/en-us/windows/zip-and-unzip-files-f6dde0a7-0fec-8294-e1d3-703ed85e7ebc>



# Code Submission (5/6)

- How to compress your folder?
- Method 2 (for Mac users)
  - <https://support.apple.com/zh-tw/guide/mac-help/mchlp2528/mac>



**Compress “b10901000\_hw1”**

# Code Submission (6/6)

- How to compress your folder?
- Method 3 (command line)

```
zip -r <name>.zip <directory name>
```

e.g.

```
zip -r b10901000_hw1.zip b10901000_hw1
```

# Hints

**Simple** : Just run [sample code](#)

**Medium** : Feature selection

**Strong** : Different optimizers and L2 regularization

**Boss** : Better feature selection, different model architectures and try more hyper-parameters



# Deadline

- Kaggle

**2023/03/01 23:59 (UTC+8)**

- NTU COOL

**2023/03/08 23:59 (UTC+8)**

# Regulations

- You should finish your homework on your own.
- 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 and this HW will get 0 pt** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

# Contact us if you have problems...

- Kaggle Homework 1 Discussion
  - <https://www.kaggle.com/competitions/ml2023spring-hw1/discussion>
- NTU COOL Homework 1 Discussion
  - <https://cool.ntu.edu.tw/courses/24108>
- Email
  - [mlta-2023-spring@googlegroups.com](mailto:mlta-2023-spring@googlegroups.com)
  - The title should begin with “[hw1]”

# Useful Links

- Hung-yi Lee, Gradient Descent (Mandarin)
  - [link1](#), [link2](#), [link3](#), [link4](#)
- Hung-yi Lee, Tips for Training Deep Networks (Mandarin)
  - [link1](#), [link2](#)
- [Pytorch Toolkit](#)
- [Link that can find all things](#)
- [Class webpage](#)

(If Google or Stackoverflow can answer your questions, you may take advantage of them before asking the TAs.)

# FAQ

(1) L2 regularization 除了 sample code 提供的在計算 loss 時處理之外，也可以使用 optimizer 的 `weight_decay` 實現，可參考 [PyTorch](#) 官方文檔

(2) sklearn、TensorFlow、xgboost 是可以使用的（使用額外線上資源請附上 Reference）

(3) 只要 Post-processing 是由程式自動完成，且並未違反規定（如不能使用 pre-trained model、不能直接輸出 `hardcode` 的結果、不能上網爬資料等），都是可以接受的，另外，請記得將後處理的程式一併交上，若沒有交上，將視為違反規定。

(4) 同學只要確認上傳時的檔名正確，COOL 系統內部會在同名的檔案依照版本順序加上編號，忽略即可（如 "學號\_hw1-1.zip" 等）。另外請同學確認最後一次上傳的版本是正確的，我們只會認最新的版本