

# 加簽表單

- [Link](#)



---

---

# Machine Learning HW1

## COVID-19 Cases Prediction

ML TAs

[mlta-2022-spring@googlegroups.com](mailto:mlta-2022-spring@googlegroups.com)

---

---

# Outline

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

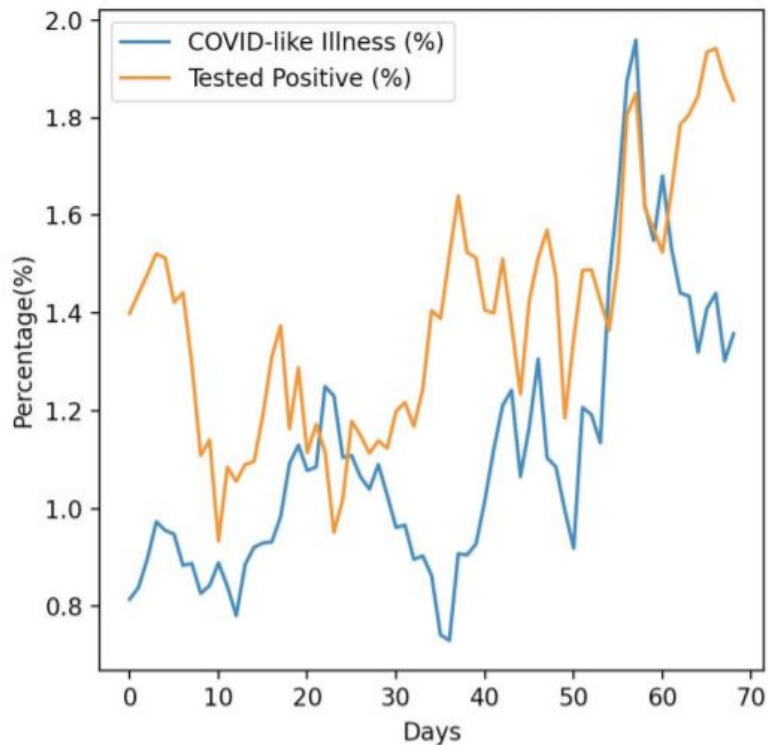
# Objectives

- 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

- COVID-19 Cases Prediction
- Source: Delphi group @ CMU
  - A daily survey since April 2020 via facebook.

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



# Task Description

- Given survey results in the past 5 days in a specific state in U.S., then predict the percentage of new tested positive cases in the 5th day.



survey



positive  
cases

**Day1&2&3&4**



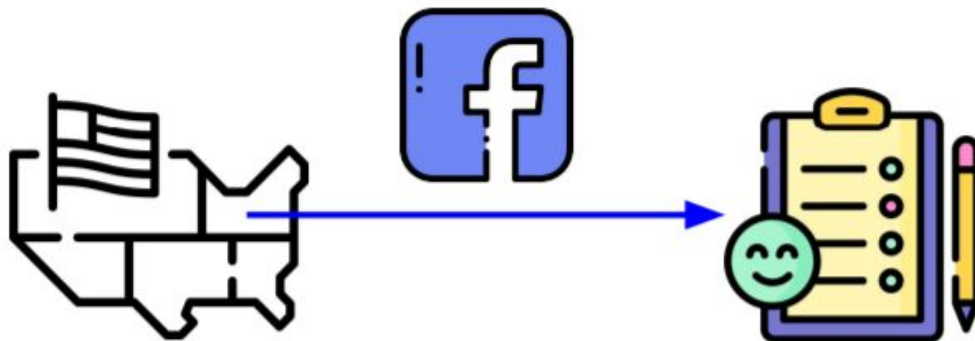
survey



**positive  
cases**

**Day5**

# Data



Conducted surveys via facebook (every day & every state)

Survey: symptoms, COVID-19 testing, social distancing, mental health, demographics, economic effects, ...

# Data

- States (37, encoded to one-hot vectors)
- COVID-like illness (4)
  - cli, ili ...
- Behavior Indicators (8)
  - wearing\_mask, travel\_outside\_state ...
- Mental Health Indicators (3)
  - anxious, depressed ...
- Tested Positive Cases (1)
  - **tested\_positive (this is what we want to predict)**



# Data -- 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)

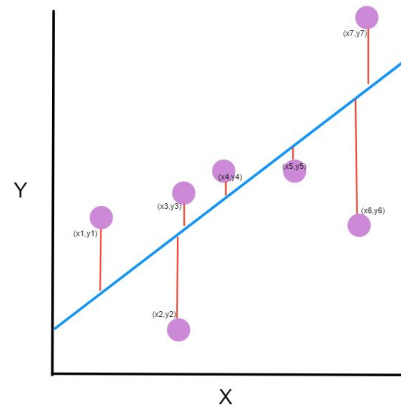
**one-hot encoding**



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


- Display name: **<student ID>\_<anything>**
  - e.g. **b08901000\_public**跟**private**差好多
  - **For auditing, don't put student ID in your displayed name.**
- Submission format: **.csv** file
  - See sample code
- [link](#)

```
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 -- Submission

- You may submit up to **5** results each day (UTC).
- 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

- 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 -- Kaggle

#	Team	Members	Score	Entries	Last	Code
1	TA_RT		0.85800	3	2h	
	---- boss baseline ----		0.86161			
	---- strong baseline ----		1.05728			
	---- medium baseline ----		1.49430			
	---- simple baseline ----		2.28371			

# Grading -- Bonus

- If your ranking in private set 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 same as code submission
  - Please upload to NTU COOL's discussion of HW1

[Report Template](#)

# Code Submission

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

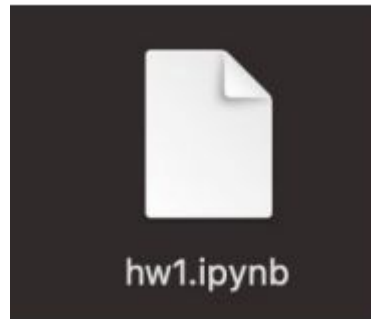
**<student\_ID>\_hw1.zip**

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



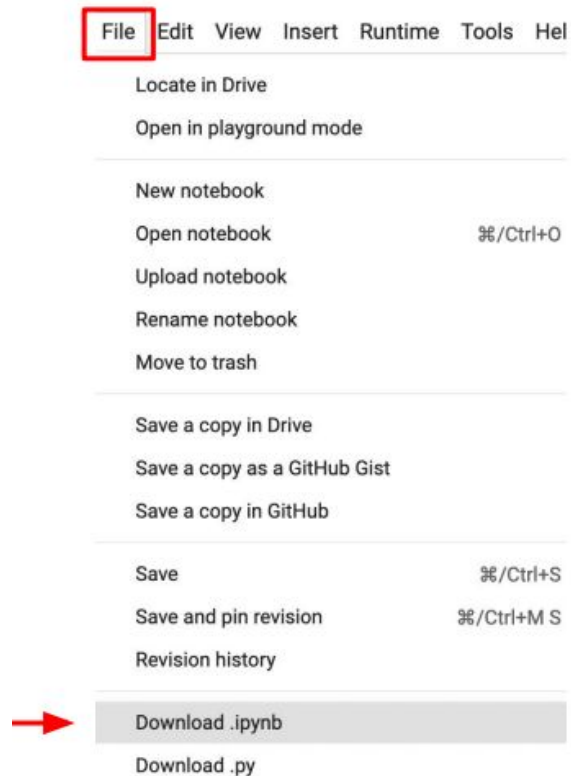
# Code Submission

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



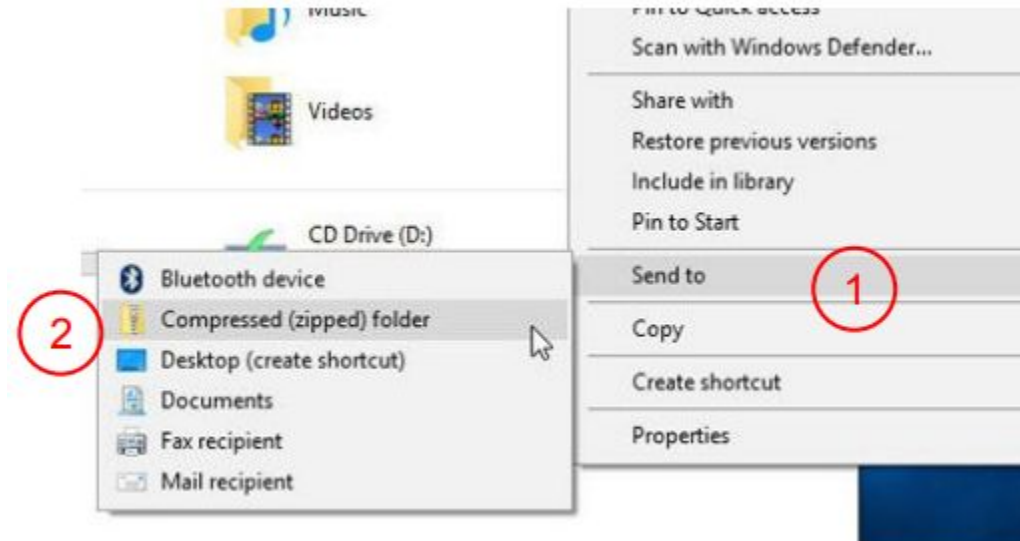
# Code Submission

- How to download your code
- from Google Colab?



# Code Submission

- 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

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



Compress “b06901020\_hw1”

# Code Submission

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

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

e.g.

```
zip -r b06901020_hw1.zip b06901020_hw1
```

# Hints

simple : [sample code](#)

medium : Feature selection

strong : Different model architectures and optimizers

boss : L2 regularization and try more parameters

# Deadlines

- Kaggle

**2022/02/23 23:59 (UTC+8)**

- NTU COOL

**2022/02/27 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...

- NTU COOL (Best way)
  - [link](#)
- Email
  - [mlta-2022-spring@googlegroups.com](mailto:mlta-2022-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](#)

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

# FAQ

## 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(如使用 pre-trained model、直接輸出 hardcode 的結果、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