
Machine Learning HW11

ML TAs

mlta-2022-spring@googlegroups.com

Outline

- Task Description
- Dataset
- Data & Submission Format
- Report
- Grading Policy
- Baseline Guides
- Regulations

Links

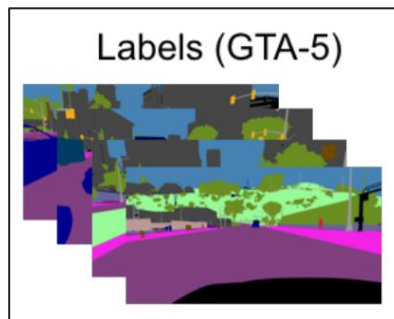
- [Kaggle](#)
- [colab tutorial](#)
- [HW11 dicussion](#)

Due

- Kaggle: 2022/06/03 23:59:59
- Code & Report: 2022/06/03 23:59:59
- **No Late Submission!!!**

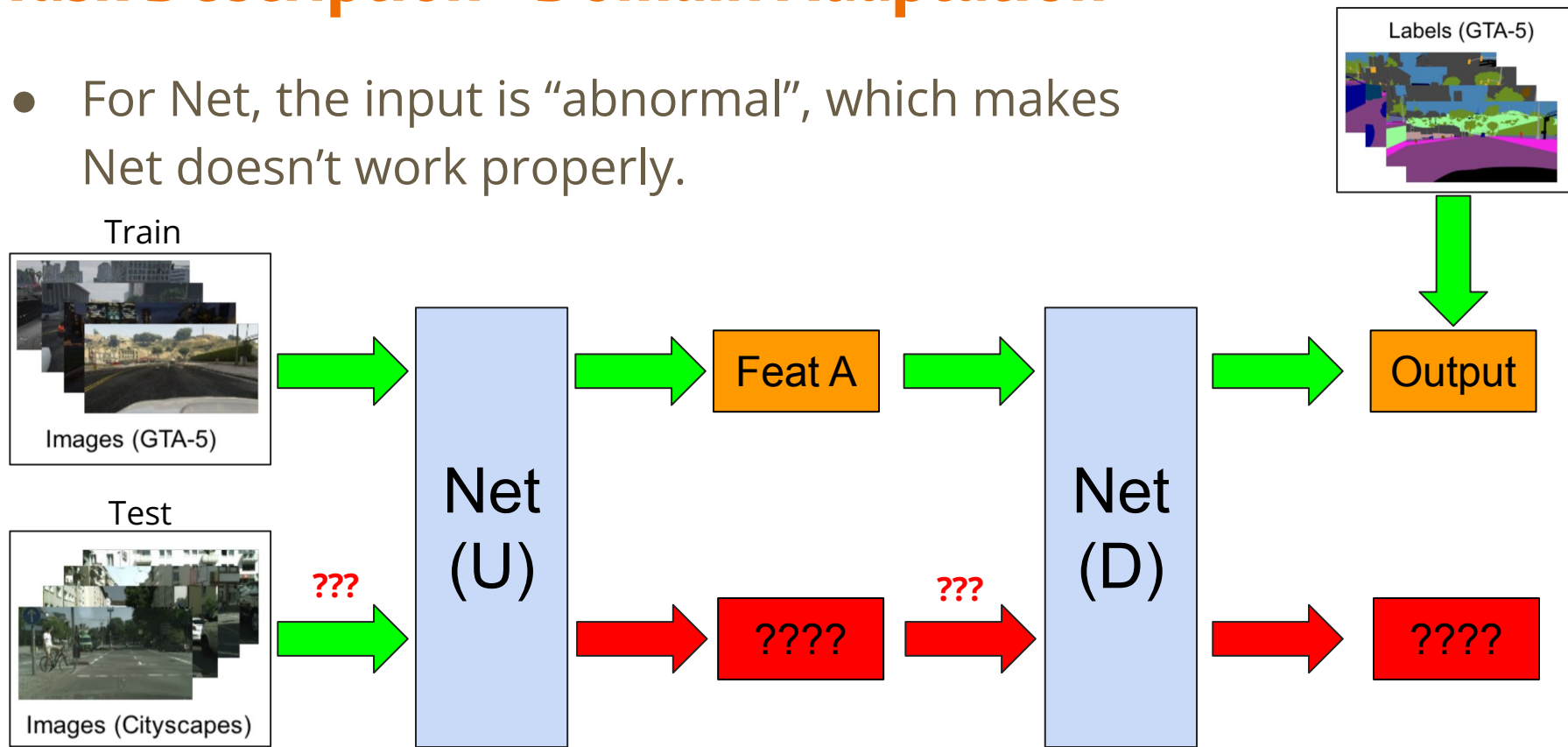
Task Description - Domain Adaptation

- Imagine you want to do tasks related to the 3D environment, and then discover that...
 - 3D images are difficult to mark and therefore expensive.
 - Simulated images (such as simulated scene on GTA-5) are easy to label.
Why not just train on simulated images?



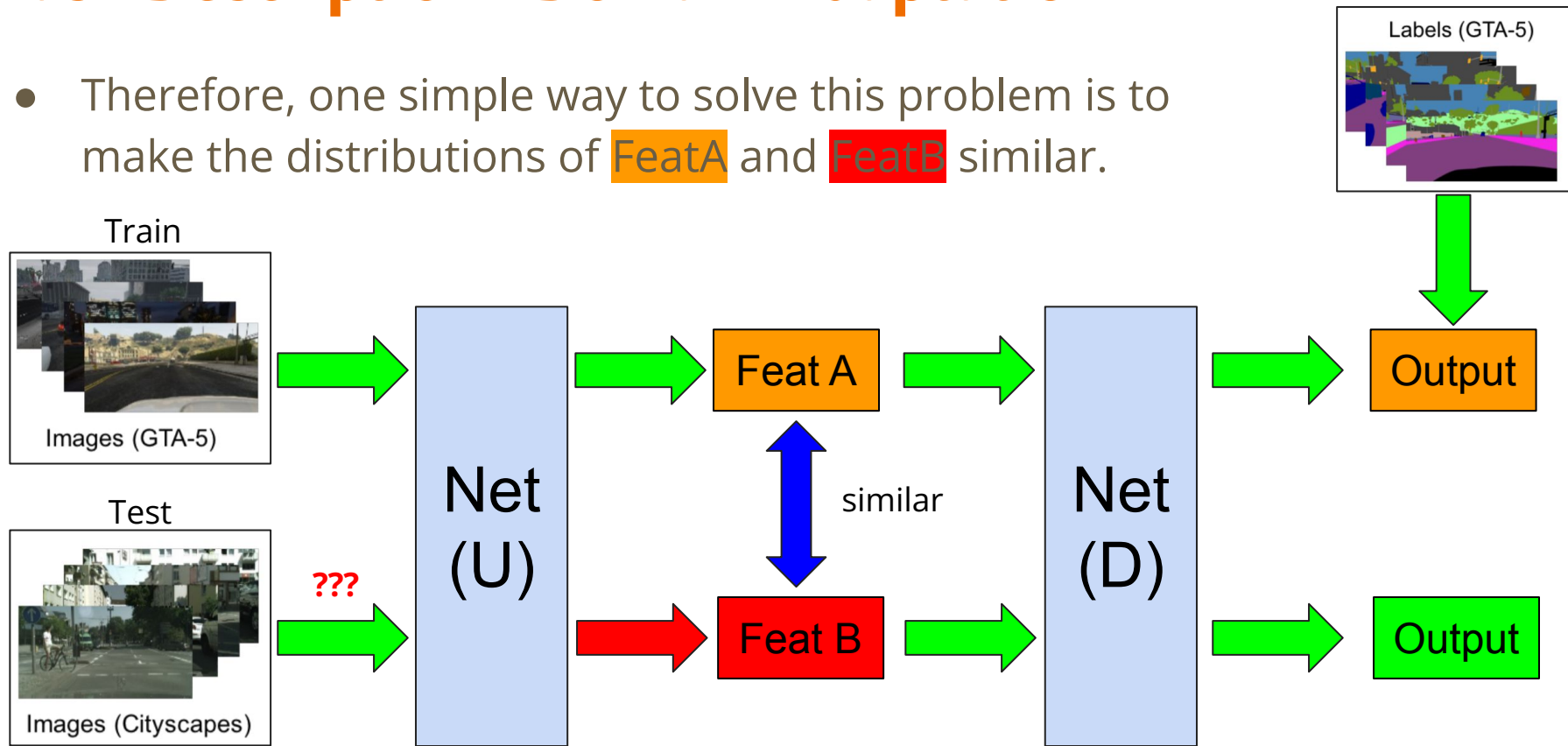
Task Description - Domain Adaptation

- For Net, the input is “abnormal”, which makes Net doesn't work properly.



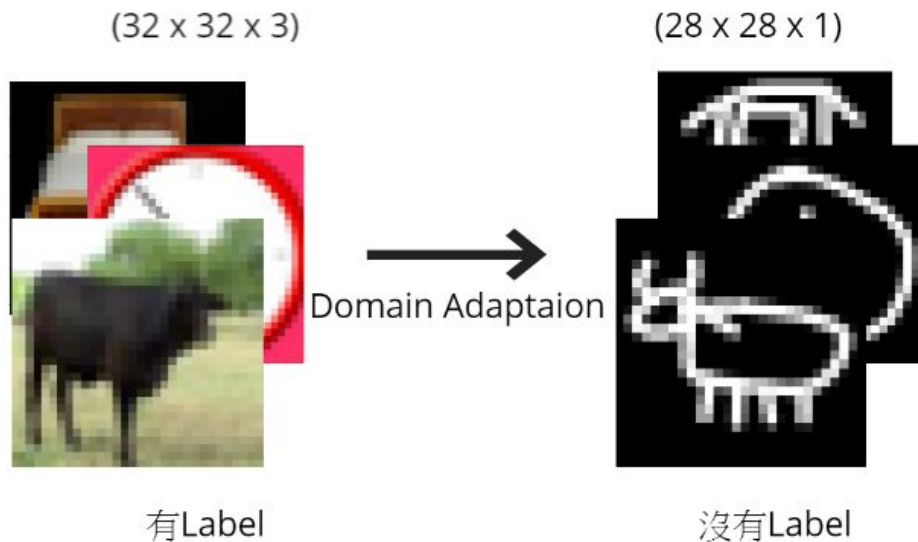
Task Description - Domain Adaptation

- Therefore, one simple way to solve this problem is to make the distributions of **FeatA** and **FeatB** similar.



Task Description - Domain Adaptation

- Our task: Given real images (with labels) and drawing images (without labels), please use domain adaptation technique to make your network predict the drawing images correctly.



Dataset

- Label: 10 classes (numbered from 0 to 9), as following pictures described.
- Training : 5000 (32, 32) RGB real images (with label).
- Testing : 100000 (28, 28) gray scale drawing images.

horse



bed



clock



apple



cat



plane



television



dog



dolphin



spider



Data Format

- Unzip **real_or_drawing.zip**, the data format is as below:
- real_or_drawing/
 - train_data/
 - 0/
 - 0.bmp, 1.bmp ... 499.bmp
 - 1/
 - 500.bmp, 501.bmp ... 999.bmp
 - ... 9/
 - test_data/
 - 0/
 - 00000.bmp
 - 00001.bmp
 - ... 99999.bmp

Data Format

- You can simply use the following code to get dataloader after extracting the zip. (You can apply your own source/target transform function.)

```
source_dataset = ImageFolder('real_or_drawing/train_data', transform=source_transform)
target_dataset = ImageFolder('real_or_drawing/test_data', transform=target_transform)

source_dataloader = DataLoader(source_dataset, batch_size=32, shuffle=True)
target_dataloader = DataLoader(target_dataset, batch_size=32, shuffle=True)
test_dataloader = DataLoader(target_dataset, batch_size=128, shuffle=False)
```

Submission Format

- First line should be “id, label”.
- Next **100,000** lines are your predicted labels of test images.
- Evaluate Metrics = Accuracy.

```
1 id, label
2 0,0
3 1,8
4 2,1
5 3,1
6 4,0
7 5,0
8 6,6
9 7,7
10 8,9
11 9,9
```

Grades

- +0.5pt : Simple public baseline (0.44616)
- +0.5pt : Simple private baseline

- +0.5 : Medium public baseline (0.64576)
- +0.5 : Medium private baseline

- +0.5 : Strong public baseline (0.75840)
- +0.5 : Strong private baseline

- +0.5 : Boss public baseline (0.80640)
- +0.5 : Boss private baseline
- +4pt : report submission / +2pt : code submission

Baseline Guides

- Simple Baseline (0.5 + 0.5 pts, $\text{acc} \geq 0.44616$, < 1 hour)
 - Just run the code and submit answer.
- Medium Baseline (0.5 + 0.5 pts, $\text{acc} \geq 0.64576$, 2~4 hours)
 - Set proper λ in DaNN algorithm.
 - Luck, Training more epochs.
- Strong Baseline (0.5 + 0.5 pts, $\text{acc} \geq 0.75840$, 5~6 hours)
 - The Test data is label-balanced, can you make use of this additional information?
 - Luck, Trail & Error :)

*影片中的 **baseline** 和投影片有些微差異, 請以投影片和 kaggle 的分數為主

Baseline Guides

- Boss Baseline (0.5 + 0.5 pts, acc ≥ 0.80640)
 - All the techniques you've learned in CNN.
 - Change optimizer, learning rate, set lr_scheduler, etc...
 - Ensemble the model or output you tried.
 - Implement other advanced adversarial training.
 - For example, [MCD](#) [MSDA](#) [DIRT-T](#)
 - Huh, semi-supervised learning may help, isn't it?
 - What about unsupervised learning? (like [Universal Domain Adaptation?](#))

***影片中的 baseline 和投影片有些微差異, 請以投影片和 kaggle 的分數為主**

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 HW11

[Report Template](#)

Code Submission - NTU COOL

- NTU COOL
 - Deadline: 6/3 (Fri.) 23:59
 - Compress your code and report into <student_ID>_hw11.zip (e.g. b10123456_hw11.zip)
 - We can **only** see your **last submission**.
 - **DO NOT** submit your model or dataset.
 - If your code is not reasonable, your semester grade x 0.9.
- Your .zip file should include only
 - Code: either .py or .ipynb

Report - Gradescope

Question1(+2 pts): Visualize distribution of features **across different classes**.

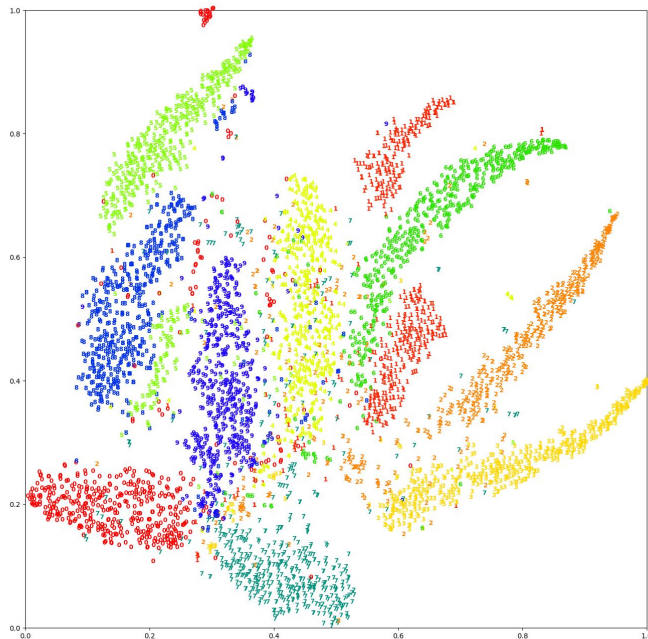
1. Please make t-SNE plot the distribution of **early, middle, final** stage.
 - a. Evaluate the model on training dataset, collect features and labels
 - b. Make **3 t-SNE plots** of the following training phase:
 - i. early stage
 - ii. middle stage
 - iii. final stage
2. Explain and analyze the distribution of features of three stages.
 - a. **Hint: Is is a good feature extractor for classification task? Why or Why not?**

[t-SNE 課程連結](#)

Report - Gradescope

3. Example plot & Hints

- SKlearn provide t-SNE function: [link](#)
- Normalize the output before plotting
- [cmap](#) is convenient to map colors



Final Stage

Report - Gradescope

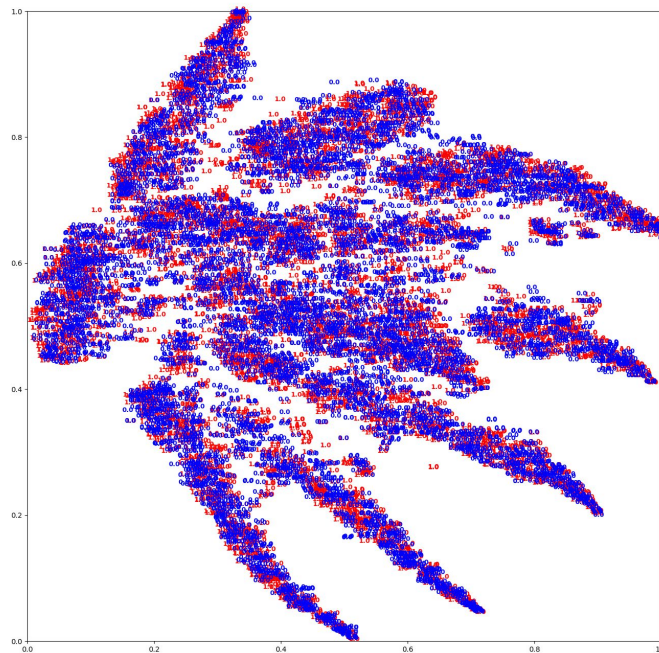
Question2 (+2pts): Visualize distribution of features **across different domains**.

1. Please plot the distribution of early, middle, final stage.
 - a. Evaluate the model on **source dataset** and **target dataset**, collect feature and labels
 - b. **Make 3 plots** of the following training phase:
 - i. early stage
 - ii. middle stage
 - iii. final stage
2. Explain and analyze the distribution of features of three training phases.
 - a. **Hint: Is is a good feature extractor for domain adaption task? Why or Why not?**

Report - Gradescope

3. Example plot & Hints

- The label is related to the domain
e.g. "1" for source and "0" for target
- Target dataset is too large. Just randomly pick
5000 images to evaluate.



Final Stage

Report - Gradescope

- Question1 (+2pts)
 - Include 3 t-SNE plots of different phase accross different classes.
 - Compare the plots and give simple explanation on the distribution of the features.
- Question2 (+2pts)
 - Include 3 t-SNE plots of different phase accross source and target domains.
 - Compare the plots and give simple explanation on the distribution of the features.

- Submit pdf to gradescope before **deadline: 2022/6/03 23:59**

Regulations

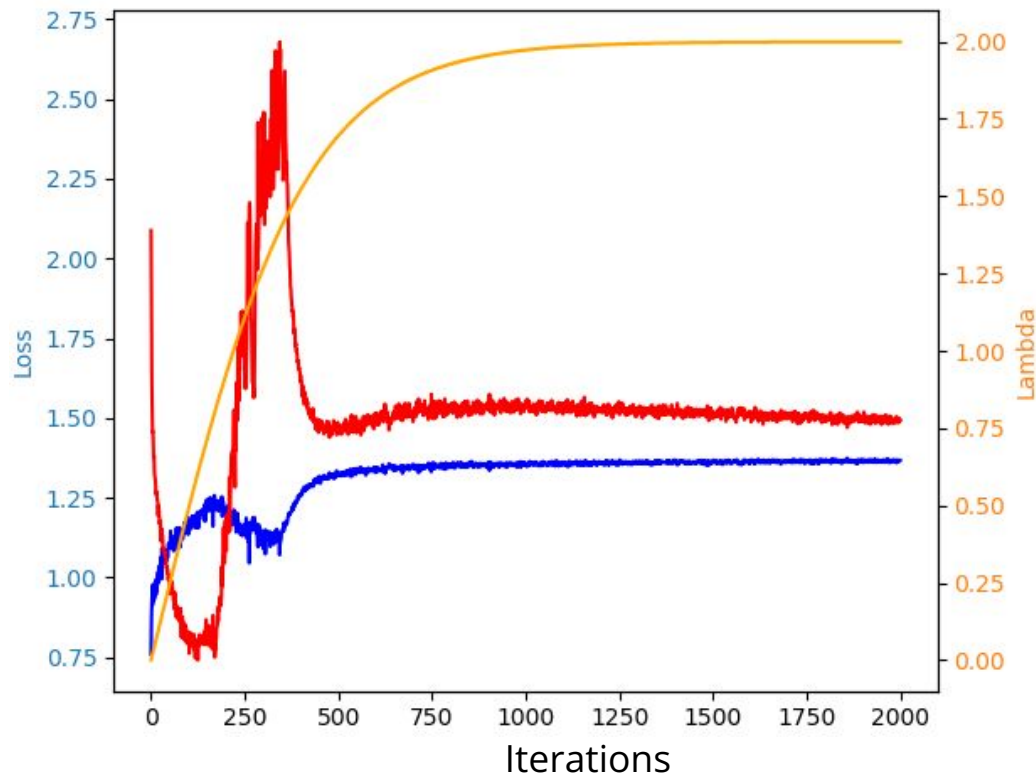
- You should finish your homework on your own.
- You should not modify your prediction files manually
- Do not share codes or prediction files with any living creatures.
- Do not use any approaches to submit your results more than **5 times** a day.
- **Do not search or use additional data or pre-trained 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
 - The title should begin with “[hw11]”

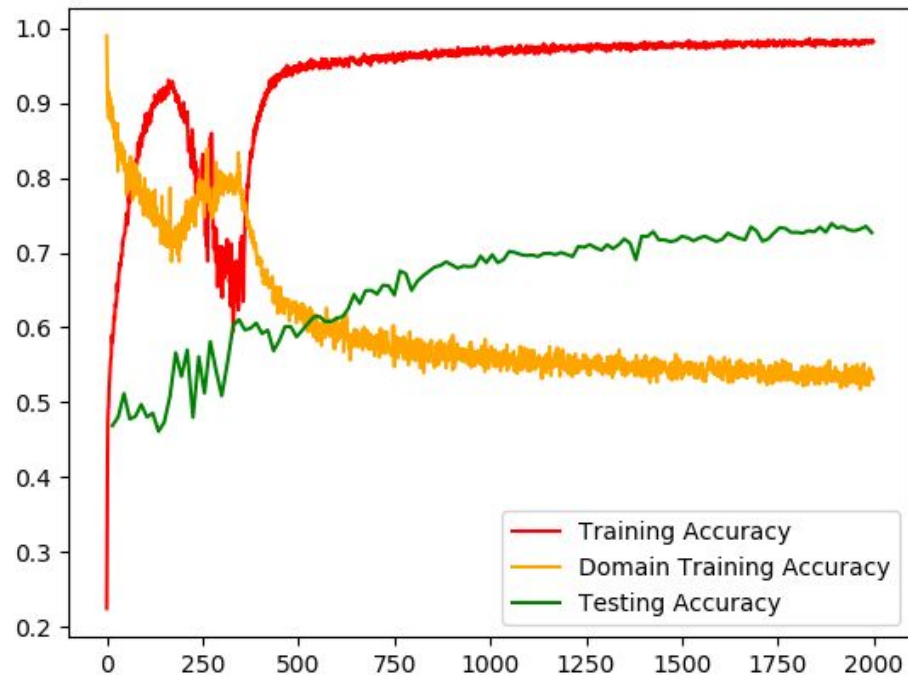
Learning Curve (Loss)

- This image is for reference only.



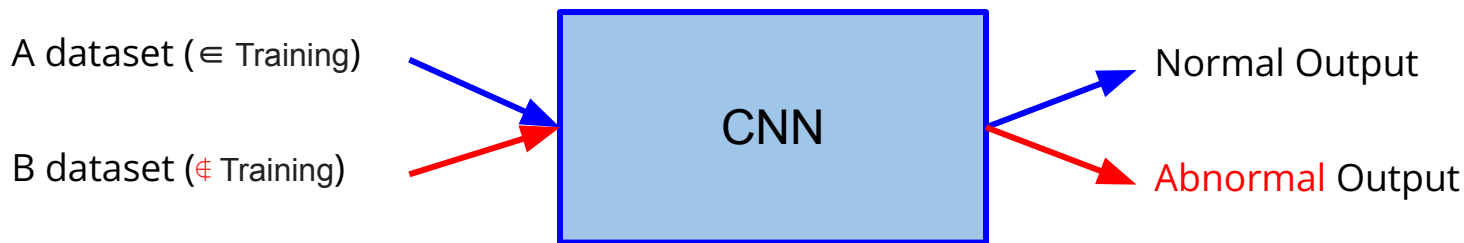
Learning Curve (Accuracy)

- This image is for reference only.
- Note that you cannot access testing accuracy.
- However, this plot tells you that even though the model overfits the training data, the testing accuracy is still improving.



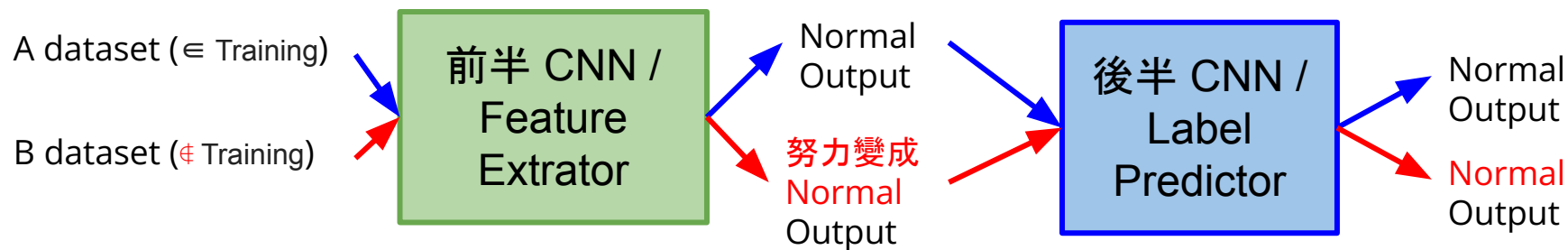
Hidden Guideline - DaNN (1/3)

- 這裡我們介紹最基礎的 DaNN (Domain-Adversarial Training of NNs)。
- 如果一個模型在測試時吃到不是與訓練集同個 distribution 的輸入, 那麼輸出往往會爆走, 如下圖。
- 而為什麼不能讓圖中的 CNN 在輸入 B dataset 輸出正常的 output? 因為你並沒有 B dataset 的 label 使模型學習。



Hidden Guideline - DaNN (2/3)

- 為了因應這樣的情況，DaNN就將 CNN 先拆成兩個部分，並且想辦法讓前半的 CNN 在吃入兩個 A dataset & B dataset 後得到的 distribution 是相近的，那麼後半就會因為輸入是正常的 output，而發揮正常的功用。



Hidden Guideline - DaNN (3/3)

- 而如何讓前半段的模型輸入兩種不同分布的資料，輸出卻是同個分布呢？最簡單的方法就是像 GAN 一樣導入一個 discriminator 來分辨輸入是哪個 dataset，並讓 feature extractor 來騙過 discriminator 即可。

