Machine Learning HW11

ML TAs
ntu-ml-2021spring-ta@googlegroups.com

Outline

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

Links

- Kaggle
- Data
- colab tutorial (mandarin)
- colab tutorial (english)
- Video Link (Ch / En)

Due

- Kaggle: 2021/06/11 23:59:59
- Code & Report: 2021/06/13 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?

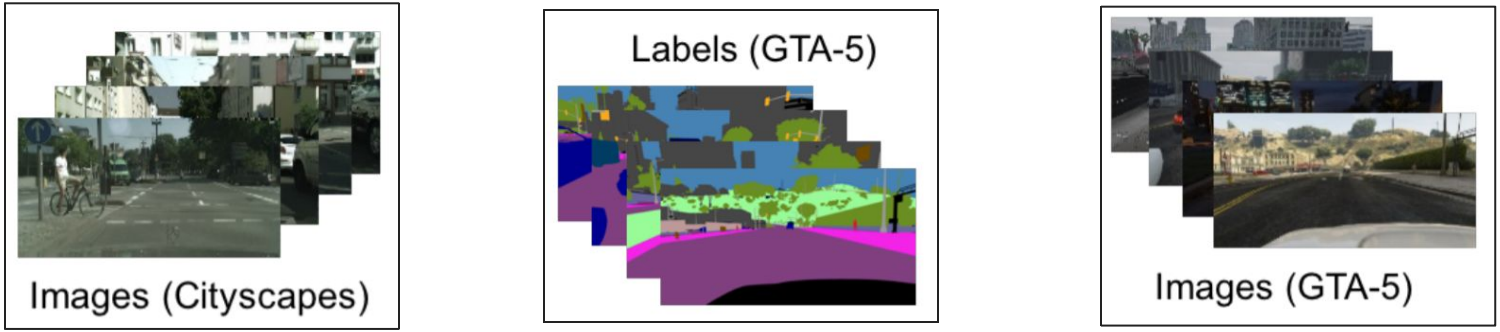


Image Reference: https://arxiv.org/pdf/1810.07911.pdf

Task Description - Domain Adaptation

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

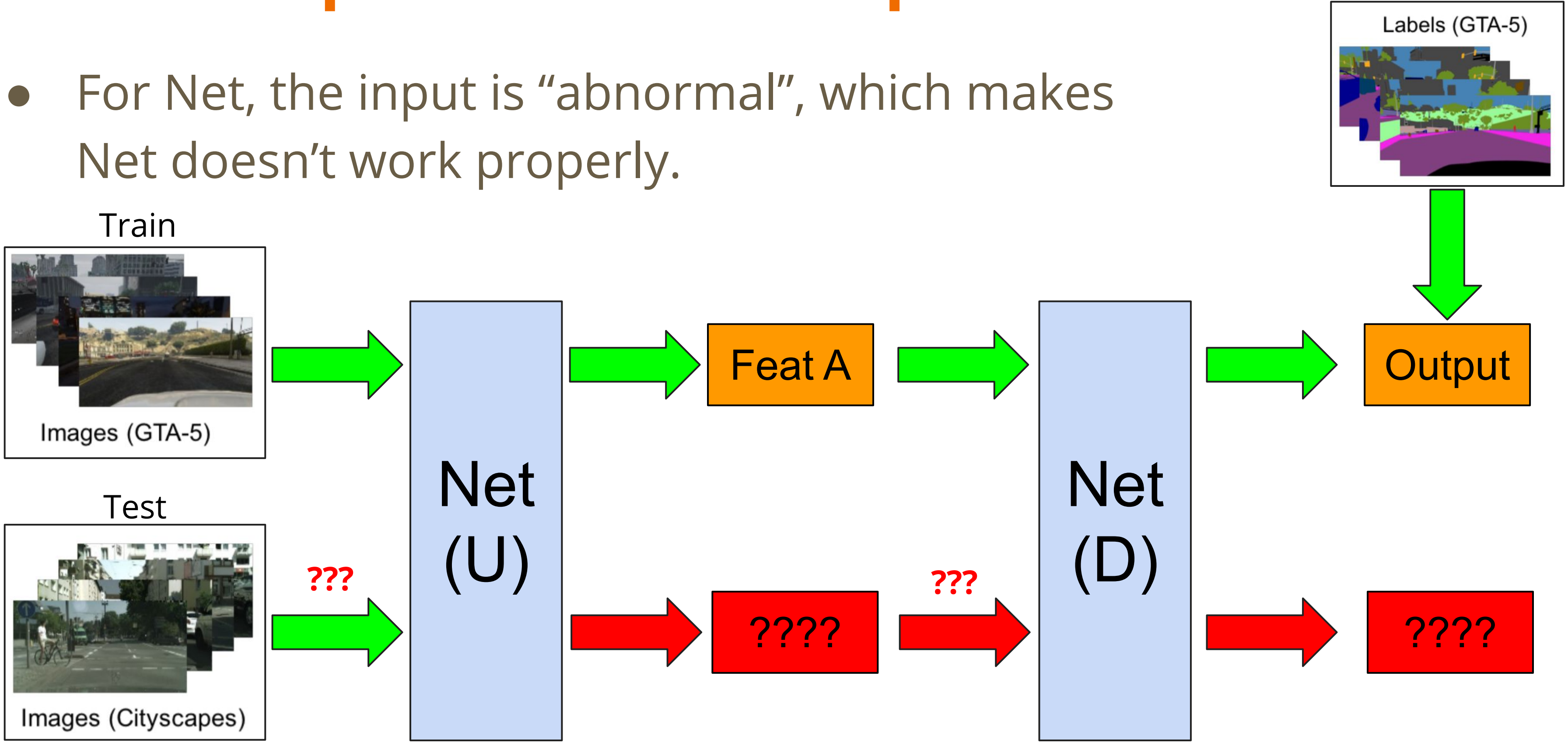
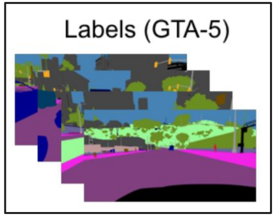
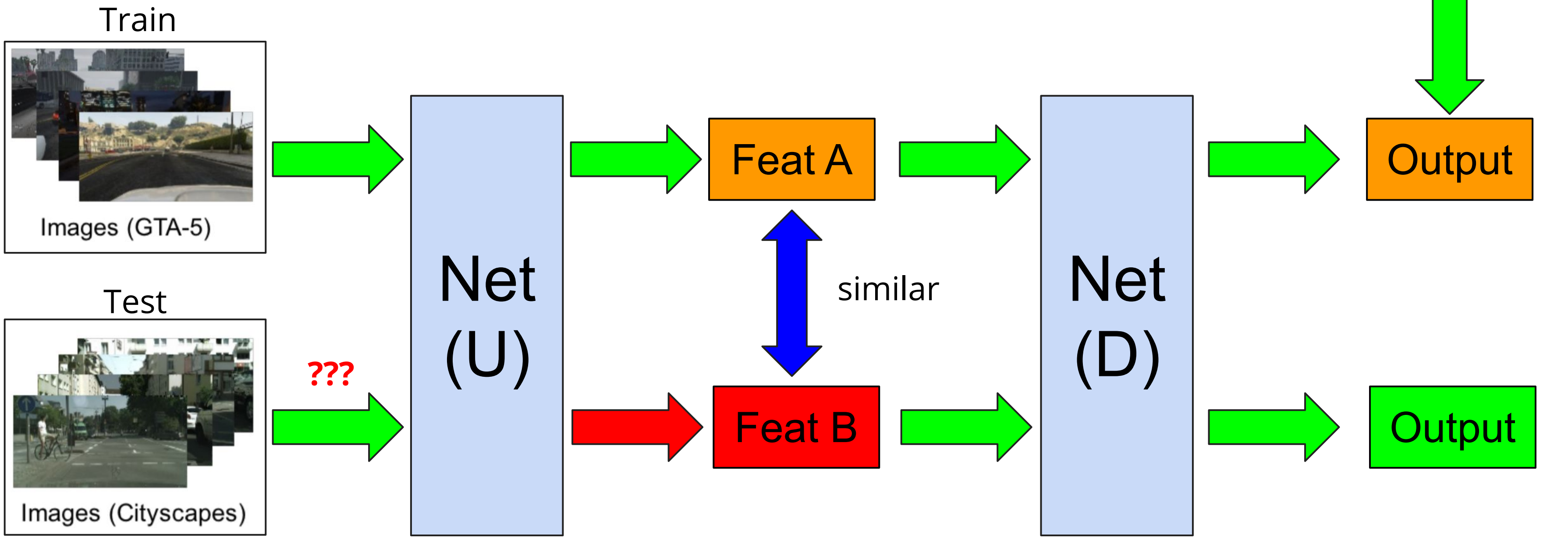


Image Reference: https://arxiv.org/pdf/1810.07911.pdf

Task Description - Domain Adaptation

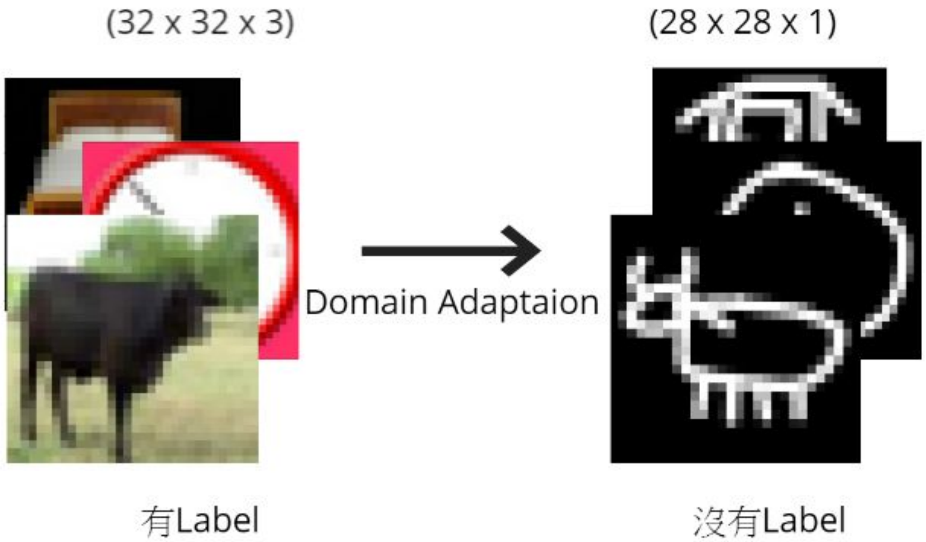
- Therefore, one simple way to solve this problem is to make the distributions of FeatA and FeatB similar.





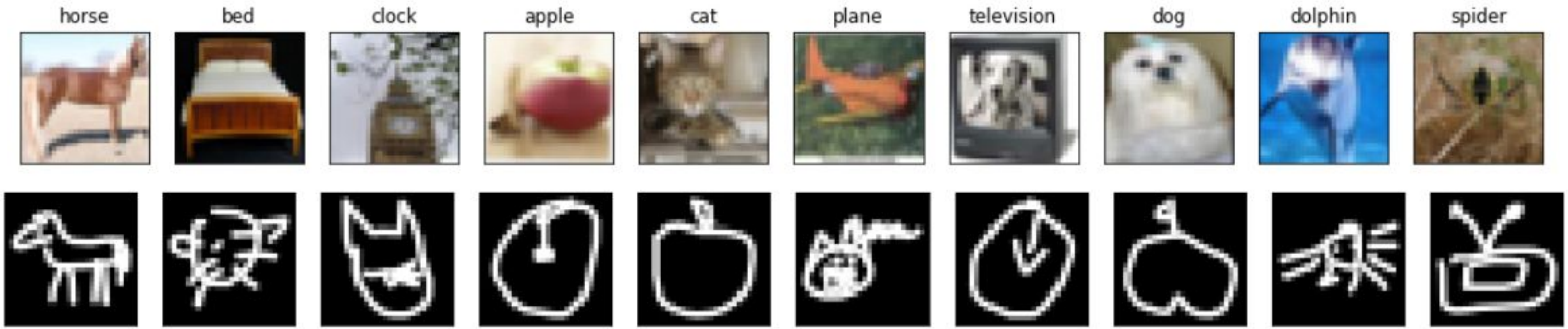
Task Description - Domain Adaptation

- 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 discribed.
- Training : 5000 (32, 32) RGB real images (with label).
- Testing : 100000 (28, 28) gray scale drawing images.



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.

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

Grades

- +4pt : code submission
- +1pt : Simple public baseline (0.41962)
- +1pt : Simple private baseline

- +1 : Medium public baseline (0.59980)
- +1 : Medium private baseline

- +0.75 : Strong public baseline (0.71874)
- +0.75 : Strong private baseline

- +0.25 : Boss public baseline (0.77956)
- +0.25 : Boss private baseline

Baseline Guides

- Simple Basline (2pts, acc≥0.41962, < 1hour)
  - Just run the code and submit answer.

- Medium Baseline (2 pts, acc≥0.59980, 2~4 hours)
  - Set proper λ in DaNN algorithm.
  - Luck, Training more epochs. 原本是設定400個epochs，但1000~2000個epochs比較穩定

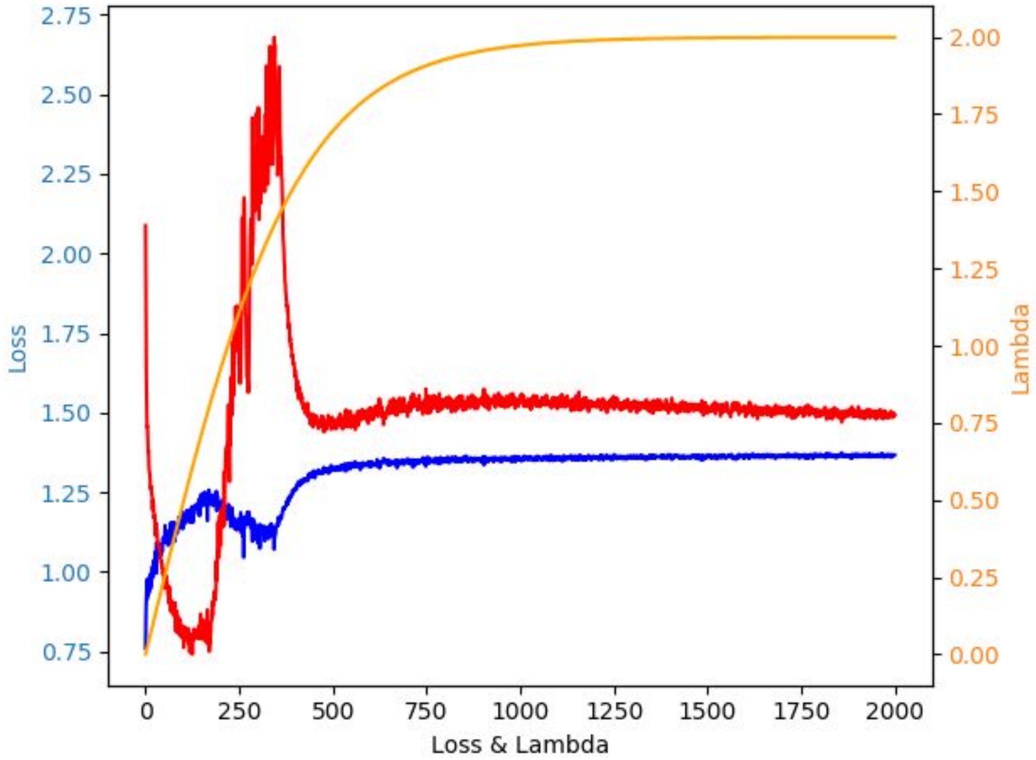
- Strong Baseline (1.5 pts, acc≥0.71874, 5~6 hours)
  - The Test data is label-balanced, can you make use of this additional information?
  - Luck, Trail & Error :)

Baseline Guides

- Boss Baseline (0.5 pts, acc ≥0.77956)
  - 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?)

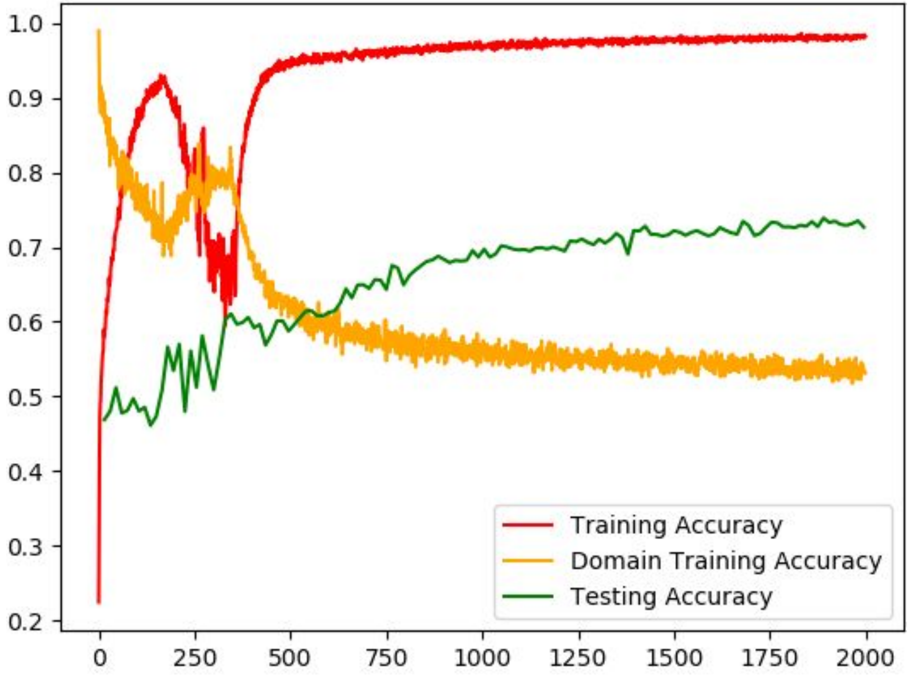
Learning Curve (Loss)

- 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.



Code Submission - NTU COOL

- NTU COOL
  - Deadline: 6/13 (Sun.) 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: .pdf (only for those who got 10 points)
- [Report template]

Regulations

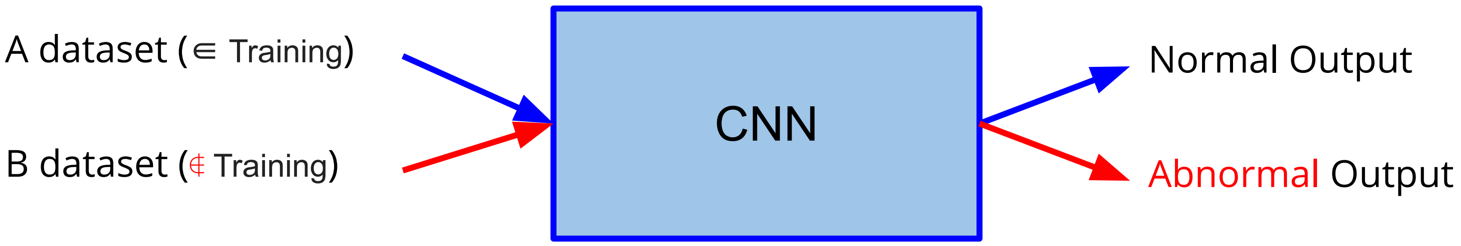
- You should NOT plagiarize, if you use any other resource, you should cite it in the reference. (∗)
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- Do NOT search or use additional data.
- Do NOT search the label or dataset on the Internet.
- **Do NOT use pre-trained models on any image datasets.**
- Your **final grade x 0.9** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

If any questions, you can ask us via...

- NTU COOL (recommended)
  - [Link]
- Email
  - [Link]
  - The title should begin with “[hwX]” (X is the homework number)
- TA hour
  - Each Monday 19:00~21:00 @Room 101, EE2 (電機二館101)
  - Each Friday 13:30~14:20 Before Class @Lecture Hall (綜合大講堂)
  - Each Friday during class

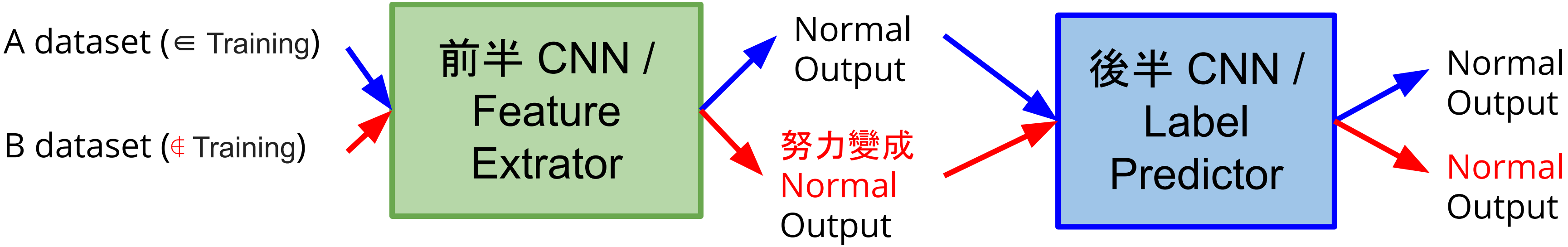
Hidden Guideline - DaNN (1/3)

- 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。
- colab tutorial