

# Machine Learning HW11 Domain Adaptation Report

R11945071 NTUBEI 莊聰賢

- **Question1 : Visualize distribution of features across different classes.**

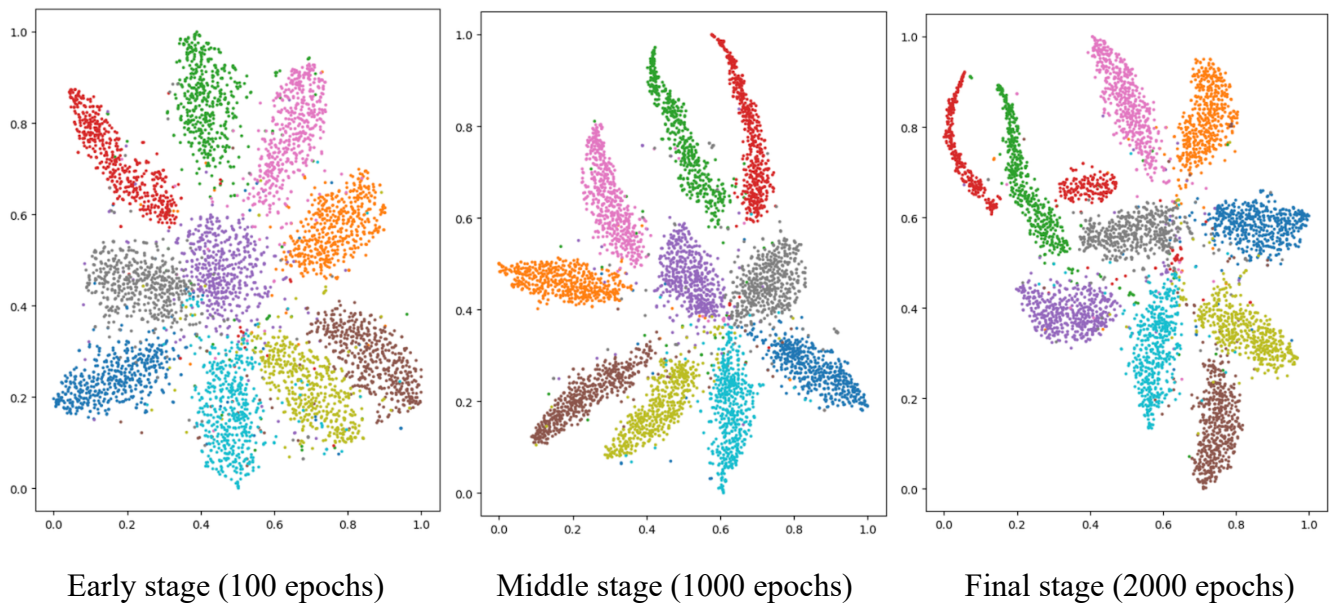
1. Please make t-SNE plot the distribution of **early, middle, final** stage

- a. Evaluate the model on training dataset, collect feature and labels

- b. Make **3 t-SNE plots** of the following training phase:

2. Explain and analyze the distribution of features of three training phases.

- a. Hint: Is this a good feature extractor for classification task? Why or Why not?



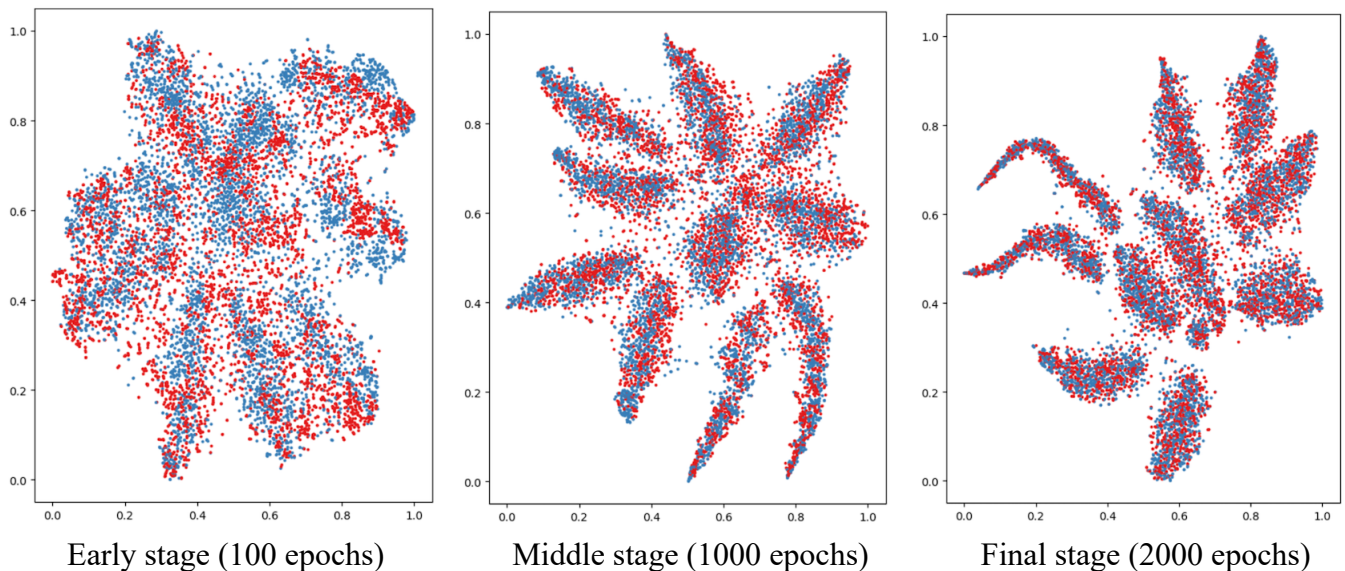
本次 training 更改了 lamb function 使其隨著不同的 epoch 去更新 lamb 的值，另外也加了 AdamW(weightdecay=0.0001)、seed function 等等，public score 分數 0.76686，總共 train 了 2000epochs，那我將 stage 分成 100、1000、2000epochs，分別由 t-SNE 圖去展示 data 分佈的情形。

可以從上面 t-SNE 圖明顯地看到，early stage 中 data point 的趨勢還有點散亂，但可以透過顏色大致分成些許 group，不過我認為在 early stage 還不足以透過 features 將 data points 做適當的分類。

那在 middle stage 可以看到，相較於前一張圖 data points 會來的更緊密一些，尤其紅色 group 有個很明顯的水滴狀線條，這也代表說每個 data group 都有象徵性的 features 從 feature extractors 產出，因此我認為 1000epochs 左右的 modle states 應該有蠻好的 classification 能力。

那在最後期可以發現 data points 會比前一張圖更緊緻一些，這也代表這些 features 更適合拿來去區分出這些不同的 data groups。

- **Question2 : 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:
  2. Explain and analyze the distribution of features of three training phases.
    - a. Hint: Is this a good feature extractor for domain adaption task? Why or Why not?



這次分析不同 domain data points 在 t-SNE 上的 model 和 Question 1 一樣。

由三張圖可以明顯的知道，左邊 early stage 中 source(blue)&target(red)的 data points 分佈上還算蠻散亂的，因此訓練 100epochs 並沒有明顯 domain adaptation 的情形發生，source 跟 target 都還是有各自的 data pattern。

Middle stage 中會發現兩個 domain 的 data points 開始出現一致性的分類趨勢，而訓練出來的 t-SNE 圖與 Question1 也很類似，代表在 1000epochs 已經有 domain adaptation 的效果出現。

在 final stage 中會發現 data points 聚集地更緊緻，而資料分布的趨勢也較中間那張圖更明顯，代表在 final stage 的 training process 中具有更好的 domain adaptation 效果，最後也順利通過 strong 的 baseline。

## ● Modle parameters(100、1000、2000epochs)

```
for epoch in range(num_epochs):
    #Adjust Lambda
    #reference:https://arxiv.org/pdf/1505.07818.pdf
    p = epoch/(num_epochs-1)
    lamb = np.log(1.03+1.68*epoch/num_epochs)
    #lamb = 2/(1+math.exp(-10*p))-1
    train_D_loss, train_F_loss, train_acc = train_epoch(source_dataloader, target_dataloader, lamb)
    torch.save(feature_extractor.state_dict(), f'extractor_model100.bin')
    torch.save(label_predictor.state_dict(), f'predictor_model100.bin')
    print('epoch {:>3d}: train D loss: {:.6.4f}, train F loss: {:.6.4f}, acc {:.6.4f}, lamb{:.4f}'.format(epoch, train_D_loss, train_F_loss, train_acc, lamb))
```