學號：B06902006　系級：資工三 姓名：王俊翔

1. 請描述你實作的模型架構、方法以及 accuracy 為何。其中你的方法必須為 domain adversarial training 系列 (就是你的方法必須要讓輸入 training data & testing data 後的某一層輸出 domain 要相近)。

   Ans:

   我實做的模型為標準的 DANN 模型，feature extractor 為 VGG 類型的 CNN，至於 domain classifier 及 predictor 都是 FCN，沒啥特別的，如下圖所示，至於 training 的方法大都與 sample code 類似，source data 過 Canny Edge Detection，及一些 transform (這裡沒改)，optimizer 用 ADAM，lr 為 1e-3。此外，原 sample code 的 lambda 為 0.1，這裡改為 paper 所述之方法(如下圖式子)，epoch 次數為 1000，accuracy 丟上去後從 0.52836 變成了 0.79430。

$$\lambda_p = \frac{2}{1+\exp(-\gamma \cdot p)} - 1,$$

```
class DomainClassifier(nn.Module):

    def __init__(self):
        super(DomainClassifier, self).__init__()

        self.layer = nn.Sequential(
            nn.Linear(512, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),

            nn.Linear(512, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),

            nn.Linear(512, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),

            nn.Linear(512, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),

            nn.Linear(512, 1),
        )

    def forward(self, h):
        y = self.layer(h)
        return y
```

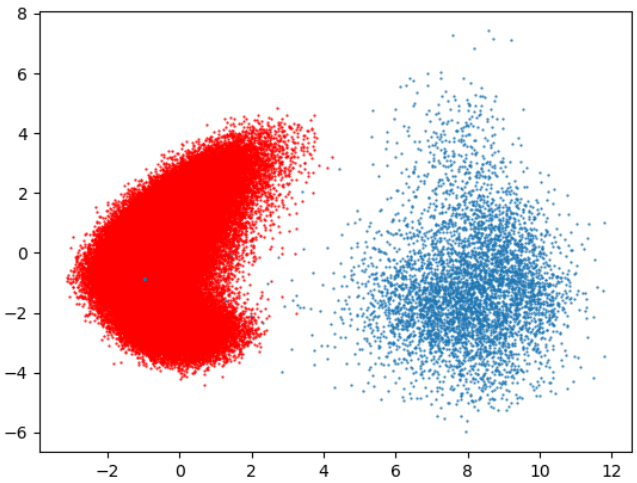
```python
class FeatureExtractor(nn.Module):

    def __init__(self):
        super(FeatureExtractor, self).__init__()

        self.conv = nn.Sequential(
            nn.Conv2d(1, 64, 3, 1, 1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(64, 128, 3, 1, 1),
            nn.BatchNorm2d(128),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(128, 256, 3, 1, 1),
            nn.BatchNorm2d(256),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(256, 256, 3, 1, 1),
            nn.BatchNorm2d(256),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(256, 512, 3, 1, 1),
            nn.BatchNorm2d(512),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(512, 512, 3, 1, 1),
            nn.BatchNorm2d(512),
            nn.ReLU(),
        )

    def forward(self, x):
        x = self.conv(x).squeeze()
        return x
```

```python
class LabelPredictor(nn.Module):

    def __init__(self):
        super(LabelPredictor, self).__init__()

        self.layer = nn.Sequential(
            nn.Linear(512, 512),
            nn.ReLU(),

            nn.Linear(512, 512),
            nn.ReLU(),

            nn.Linear(512, 10),
        )

    def forward(self, h):
        c = self.layer(h)
        return c
```

2. 請視覺化真實圖片以及手繪圖片通過沒有使用 domain adversarial training 的 feature extractor 的 domain 分布圖。

   Ans:

   下圖可以發現沒有使用 domain adversarial training 的 feature extractor domain 分布圖，source(紅)及 target(藍)兩者的分部在用 PCA 縮成兩維後距離相差甚遠，與理論相符。



3. 請視覺化真實圖片以及手繪圖片通過有使用 domain adversarial training 的 feature extractor 的 domain 分布圖。(2%)

   Ans:

   下圖可以發現有使用 domain adversarial training 的 feature extractor domain 分布圖，source(紅)及 target(藍)兩者的分部在用 PCA 縮成兩維後分布非常相似，與理論相符，也就是 domain classifier 難易分辨其圖為 source data 還是 target data。