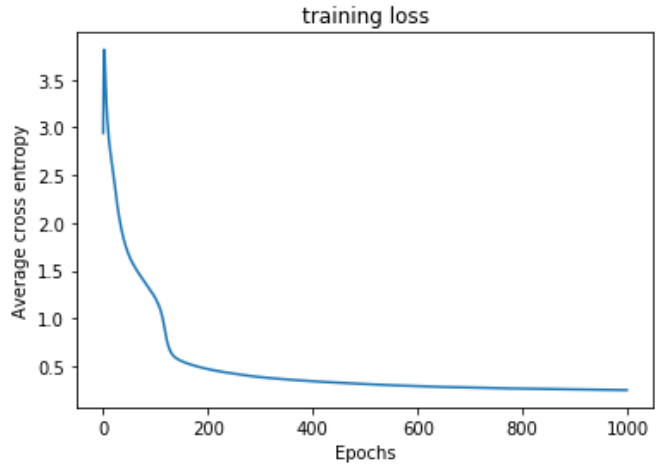
深度學習 HW1

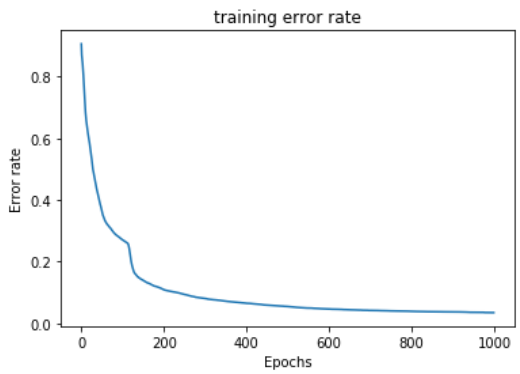
0853402 吳奐萱

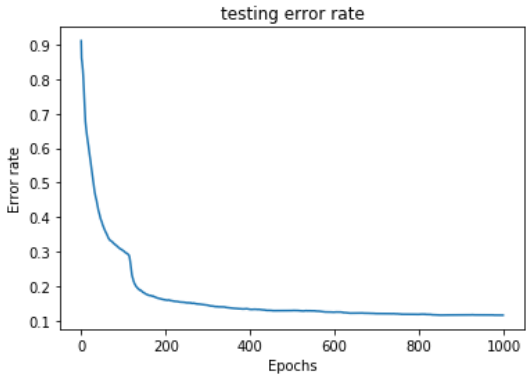
1.DNN 實作

(1) You have to show your (a) learning curve, (b) training error rate and (c) test error rate in the report. You should design the network architecture by yourself.

net1 = Network([784, 60, 30, 10])
min_batch_size = 50
eta = 0.1
epoches = 1000





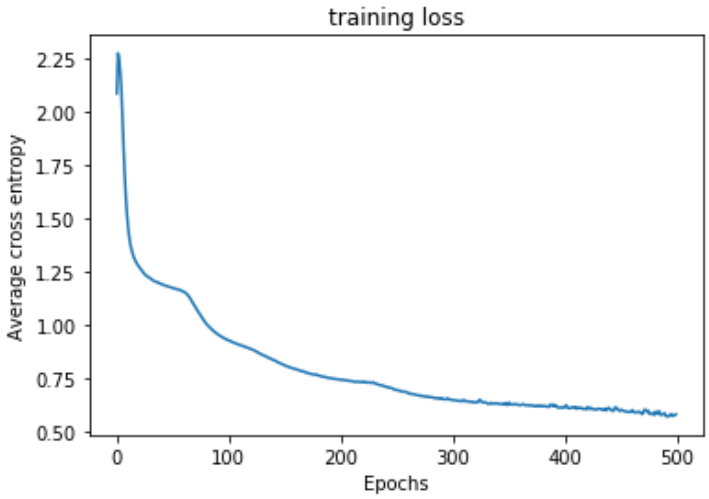


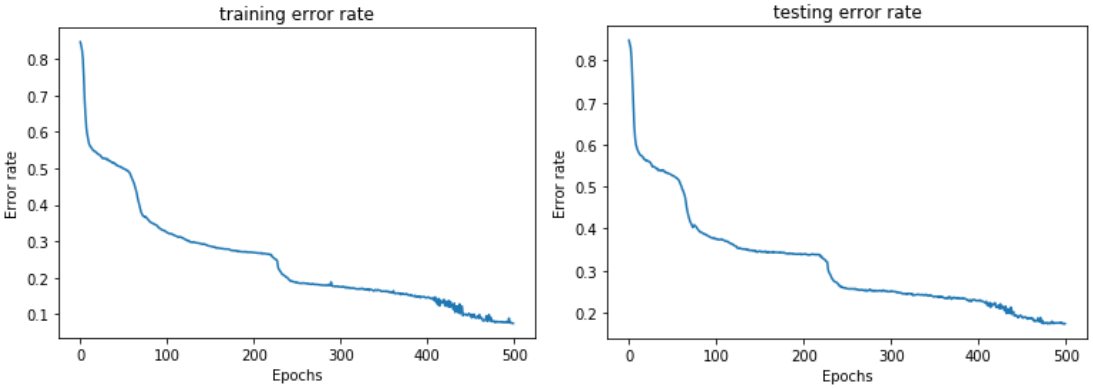
net1 = DNN([784, 30, 10, 2, 10])

min_batch_size = 50

lr = 1.5

epoches = 500

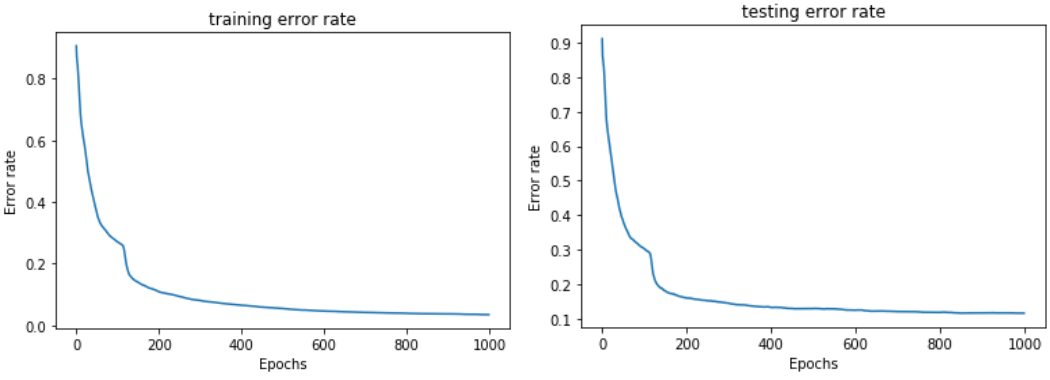




(2) Please perform zero and random initializations for the model weights and compare the corresponding error rates. Are there any difference between two initializations? Please discuss in the report

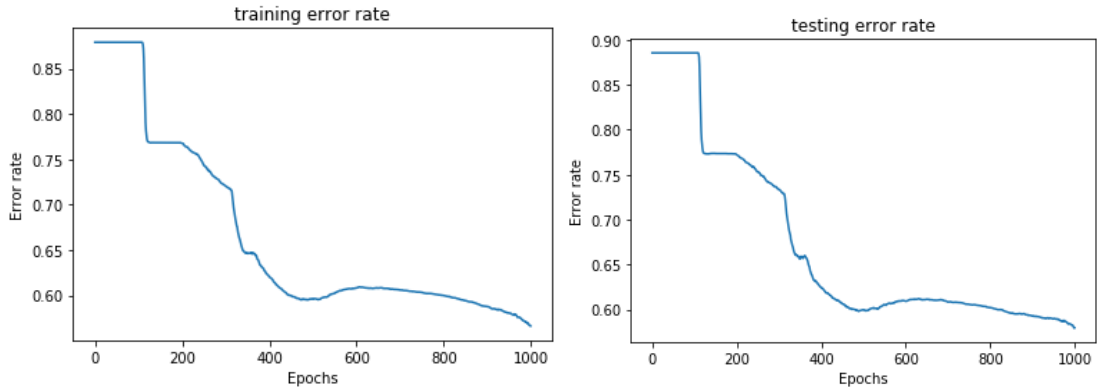
Random:

Random 的錯誤率一開始就有明確的大幅下降，中間也都沒有太多的回升的趨勢，最後收斂的數值差不多在接近 0.1 的位置，很明顯就是比 zero 得好上許多



Zero:

Zero 在最一開始並沒有馬上下降，反而還持平了一段時間才開是大幅下降，此圖看起來 zero 還沒到達收斂的位置，感覺如果多跑幾次還會持續下降，但跑旱 random 相同次數的 epoches，zero 的錯誤率大約在 0.55 左右，可看出 random 下降得比較快，且錯誤率也比較低

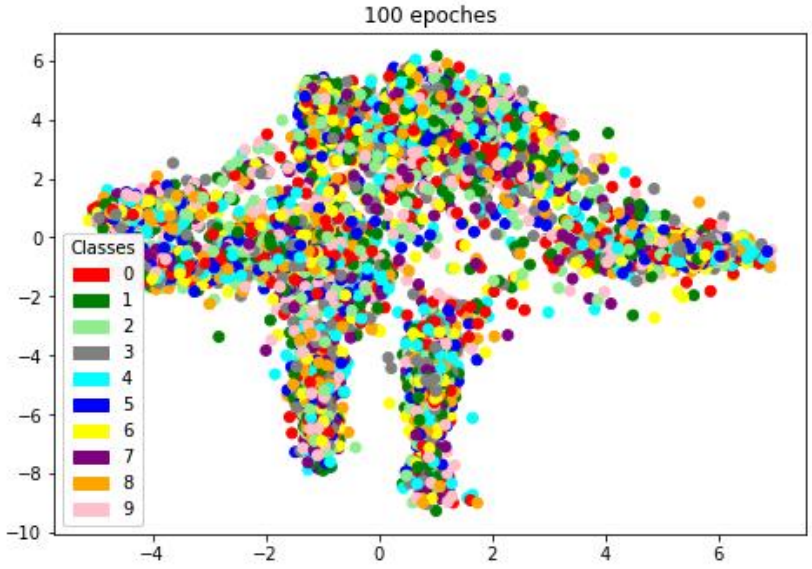


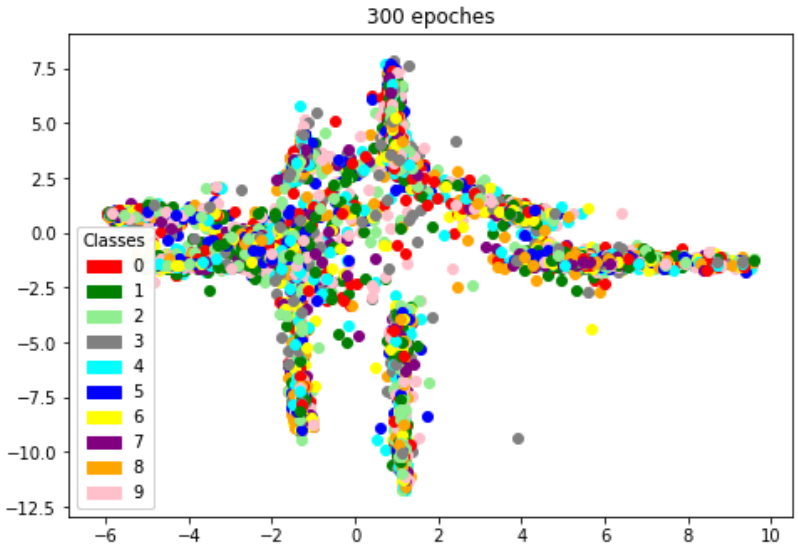
(3) Design your network architecture with the layer of 2 nodes before the output layer.

(1) Plot the distributions of latent features at different training stages. For example, you may show the results when running at 20th and 80th learning epochs.

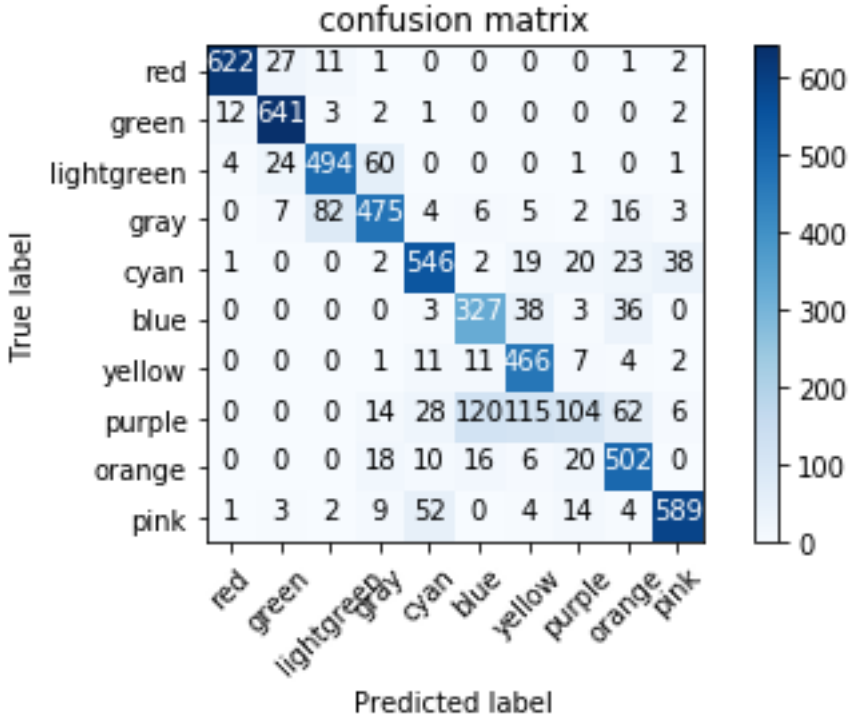
(2) Please discuss the evolution of latent features at different training stage.

我的兩個不同 epoch 所呈現的 latent feature 都不是分得很好，但可以明確地看出 300 epoches 有比 100 epoches 形狀在更明顯，應該是有區分得更好的感覺，只是跑的不夠多次。





(4) Please list your confusion matrix and discuss about your results.



2. CNN 實作

(1) Please describe in details how to preprocess images because of the different resolution images and various bounding boxes region in Medical Masks dataset and explain why. You have to submit your preprocessing code.

我先把 train.csv 和 test.csv 讀進來並存成 train 和 test，train 和 test 的處理相同，因此我只介紹 train(且截圖也截 train 的部分)。

1. 把 image 和 label 分別存好，image 使用 resize 的函式全部調成(128, 128)，而

label 則存成 good = 0, none = 1, bad = 2

2. 把照片 transpose 到對的位置((0,3,1,2))，在進行 normalization(除以 255)
3. 最後丟進 pythorch 的套件:Data.TensorDataset，改成正確的 input 格式

```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
training_image = []
training_label = []
testing_image = []
testing_label = []

for i in range(len(train)):
    try:
        image = Image.open("images/images/"+train['filename'][i])
        image = image.numpy()
#         image = cv2.imread("images/images/"+train['filename'][i])
        image_face = cv2.resize(image[train['ymin'][i]:train['ymax'][i],
                                      train['xmin'][i]:train['xmax'][i]], (128,128))
        training_image.append(image_face)
        if train['label'][i] == 'good':
            training_label.append(0)
        elif train['label'][i] == 'none':
            training_label.append(1)
        else:
            training_label.append(2)
#         training_label.append(train['label'][i])
    except:
        pass
```

```
training_image = np.array(training_image)
training_label = np.array(training_label)
testing_image = np.array(testing_image)
testing_label = np.array(testing_label)

training_X = torch.from_numpy(training_image.transpose((0,3,1,2))) #transpose = 改training_image資料
training_X = training_X.float().div(255) # 改成float且範圍改到(0,1)
training_y = torch.from_numpy(training_label).long()
testing_X = torch.from_numpy(testing_image.transpose((0,3,1,2)))
testing_X = testing_X.float().div(255)
testing_y = torch.from_numpy(testing_label).long()
training_dataset = Data.TensorDataset(training_X, training_y)
testing_dataset = Data.TensorDataset(testing_X, testing_y)
```

(2) Please implement a CNN for image recognition by using Medical Masks dataset.

You need to design the network architecture, describe your network architecture and analyze the effect of different settings including stride size and filter size. Plot the learning curve and the accuracy rate of training and test data.

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Sequential( # input shape:(1, 128, 128)
            nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, stride=1,
padding=1),
#             nn.Conv2d(in_channels=3, out_channels=32, kernel_size=7, stride=1,
padding=3),
            nn.ReLU(),
```

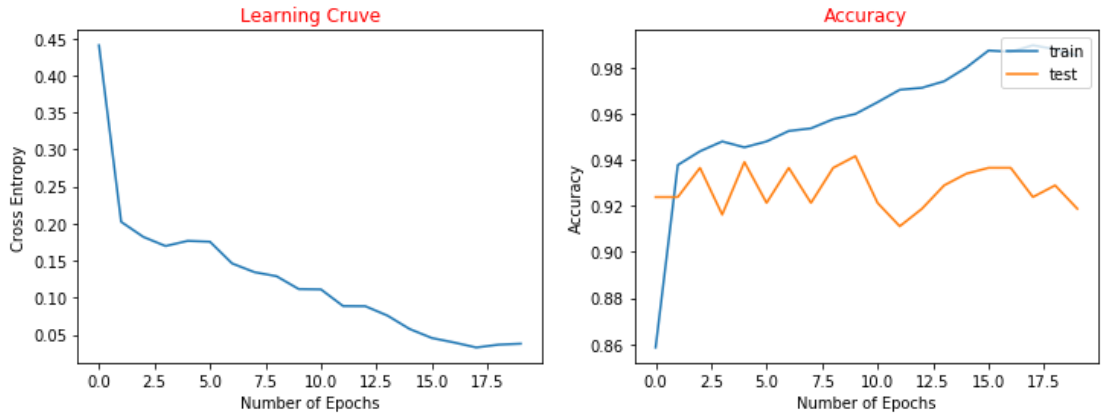
```
                nn.MaxPool2d(kernel_size=2) # output shape: (32, 64, 64)
            )
            self.conv2 = nn.Sequential(
                nn.Conv2d(in_channels=32,    out_channels=64, kernel_size=3,
stride=1, padding=1),
#                   nn.Conv2d(in_channels=32,    out_channels=64, kernel_size=7,
stride=1, padding=3),
                nn.ReLU(),
                nn.MaxPool2d(kernel_size=2) # output shape: (64, 32, 32)
            )
            self.out = nn.Linear(in_features=64 * 32 * 32, out_features=3)

        def forward(self, x):
            x = self.conv1(x)
            x = self.conv2(x)
            # flatten
            x = x.view(x.size()[0], -1) # (batch size, 28*28)
            output = self.out(x)
            return output
```

epoches = 20

batch_size = 50

lr = 0.001

kernel_size=7, stride=1, padding=3

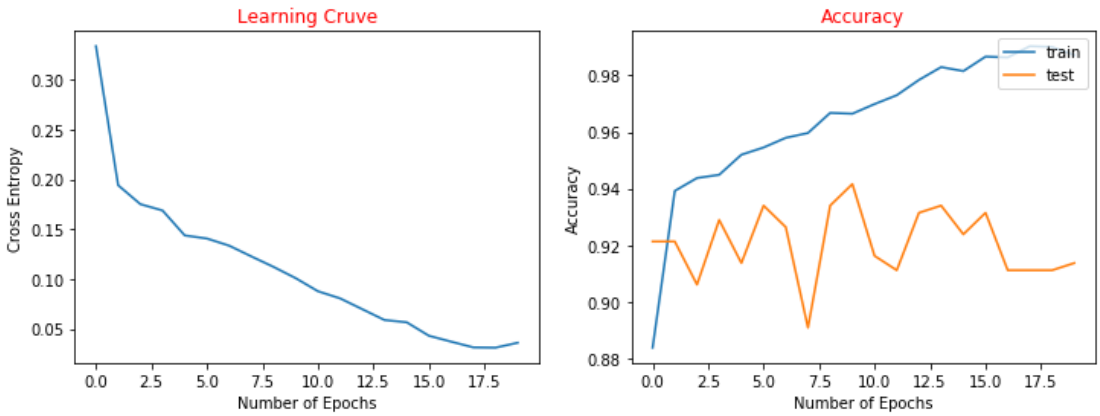


epoches = 20

batch_size = 50

lr = 0.001

kernel_size=3, stride=1, padding=1



(3) Show some examples of classification result, list your accuracy of each classes for both training and test data, and answer the following questions:

kernel_size=3, stride=1, padding=1

good:

【Training】Acc: 81.544%

【Testing】Acc: 70.667%

none

【Training】Acc: 1.333%

【Testing】Acc: 8.000%

bad

【Training】Acc: 20.167%

【Testing】Acc: 26.000%

(1) Which class has the worst classification result and why?

None class 最差，因為他的資料最少

(2) How to solve this problem? (explain and do some experiment to compare the result)

我使用 image data augmentation 的風是去增加 none class 的照片數量，就是把 none class 的照片左右 flip 和上下 flip，而得到的結果準確度也有增加，雖然幅度不大，應該是增加的數量不夠多，或許可以再改亮度或其他的方式去增加資料量。

good:

【Training】Acc: 81.895%

【Testing】Acc: 74.667%

none

【Training】Acc: 3.429%

【Testing】Acc: 4.000%

bad

【Training】Acc: 19.333%

【Testing】Acc: 22.000%

(3) Do some discussion about your results.

我在沒有分三個類別的 train 和 test 得準確度都是偏高的，但分成三類別的時候就會有很明顯的落差，而且準確度也很明確地跟資料量成正比，在經過 none class 的 data augmentation 後準確度也有上升，更可以得出他們的相關性。