

# Machine Learning 2020 - Homework 3 Report

學號：b08902100, 系級：資工一, 姓名：江昱勳

1. 請說明你實作的 CNN 模型，其模型架構、訓練參數量和準確率為何？

模型結構如下:

Layer (type)	Output Shape	Param #
=====	=====	=====
Conv2d	[-1, 128, 128, 128]	3,584
BatchNorm2d	[-1, 128, 128, 128]	256
ReLU	[-1, 128, 128, 128]	0
MaxPool2d	[-1, 128, 64, 64]	0
Dropout2d	[-1, 128, 64, 64]	0
Conv2d	[-1, 128, 64, 64]	147,584
BatchNorm2d	[-1, 128, 64, 64]	256
ReLU	[-1, 128, 64, 64]	0
MaxPool2d	[-1, 128, 32, 32]	0
Conv2d	[-1, 256, 32, 32]	295,168
BatchNorm2d	[-1, 256, 32, 32]	512
PReLU	[-1, 256, 32, 32]	1
MaxPool2d	[-1, 256, 16, 16]	0
Dropout2d	[-1, 256, 16, 16]	0
Conv2d	[-1, 512, 16, 16]	1,180,160
BatchNorm2d	[-1, 512, 16, 16]	1,024
ReLU	[-1, 512, 16, 16]	0
MaxPool2d	[-1, 512, 8, 8]	0
Conv2d	[-1, 512, 8, 8]	2,359,808
BatchNorm2d	[-1, 512, 8, 8]	1,024
ReLU	[-1, 512, 8, 8]	0
MaxPool2d	[-1, 512, 4, 4]	0
Linear	[-1, 512]	4,194,816
ReLU	[-1, 512]	0
Linear	[-1, 256]	131,328
ReLU	[-1, 256]	0
Linear	[-1, 128]	32,896
Dropout	[-1, 128]	0
ReLU	[-1, 128]	0
Linear	[-1, 100]	12,900
ReLU	[-1, 100]	0
Linear	[-1, 30]	3,030
PReLU	[-1, 30]	1
Linear	[-1, 11]	341
=====	=====	=====
Total params: 8,364,689		

訓練 400 個 epoch 後在 Kaggle 上的 public score 為 0.84997，在訓練過程中，我將原先的資料複製一份並且分別通過兩種不同的變換如下，而最後我也有 normalize 過。

```
train_transform1 = transforms.Compose([
    transforms.ToPILImage(),
    transforms.RandomChoice([
        transforms.RandomVerticalFlip(),
        transforms.RandomHorizontalFlip(),
        transforms.RandomPerspective()
    ]),
    transforms.RandomChoice([
        transforms.RandomAffine(10),
```

```

        transforms.RandomRotation(40)
    ]),
    transforms.ColorJitter(),
    transforms.ToTensor(),
    transforms.Normalize(
        [77.89311144813877 / 255, 102.3587941606983 / 255, 126.59376063616554 / 255],
        [72.80305392379675 / 255, 75.35438507973123 / 255, 79.31408066842762 / 255]
    )
])
train_transform2 = transforms.Compose([
    transforms.ToPILImage(),
    transforms.RandomOrder([
        transforms.RandomChoice([
            transforms.RandomHorizontalFlip(),
            transforms.RandomPerspective()
        ]),
        transforms.RandomAffine(30),
        transforms.RandomResizedCrop((128, 128), scale=(0.5, 1.0)),
    ]),
    transforms.RandomChoice([
        transforms.ColorJitter(),
        transforms.RandomGrayscale(),
    ]),
    transforms.ToTensor(),
    transforms.RandomErasing(0.2),
    transforms.Normalize(
        [77.89311144813877 / 255, 102.3587941606983 / 255, 126.59376063616554 / 255],
        [72.80305392379675 / 255, 75.35438507973123 / 255, 79.31408066842762 / 255]
    )
])

```

2. 請實作與第一題接近的參數量，但 CNN 深度（CNN 層數）減半的模型，並說明其模型架構、訓練參數量和準確率為何？

模型如下：

Layer (type)	Output Shape	Param #
Conv2d	[-1, 128, 128, 128]	3,584
BatchNorm2d	[-1, 128, 128, 128]	256
ReLU	[-1, 128, 128, 128]	0
MaxPool2d	[-1, 128, 64, 64]	0
Dropout2d	[-1, 128, 64, 64]	0
Conv2d	[-1, 256, 64, 64]	295,168
BatchNorm2d	[-1, 256, 64, 64]	512
ReLU	[-1, 256, 64, 64]	0
MaxPool2d	[-1, 256, 16, 16]	0
Conv2d	[-1, 512, 16, 16]	1,180,160
BatchNorm2d	[-1, 512, 16, 16]	1,024
ReLU	[-1, 512, 16, 16]	0
MaxPool2d	[-1, 512, 4, 4]	0
Linear	[-1, 800]	6,554,400
Dropout	[-1, 800]	0
ReLU	[-1, 800]	0
Linear	[-1, 400]	320,400
PRELU	[-1, 400]	1
Linear	[-1, 256]	102,656
ReLU	[-1, 256]	0
Linear	[-1, 128]	32,896
Dropout	[-1, 128]	0
ReLU	[-1, 128]	0
Linear	[-1, 50]	6,450
ReLU	[-1, 50]	0
Linear	[-1, 30]	1,530
PRELU	[-1, 30]	1

```

      Linear                [-1, 11]                341
=====
Total params: 8,499,379

```

在 kaggle 上的分數為 0.84518，跟沒有砍一半的差不多，所以我猜測其實第一題的模型還有一些可以調整的地方。

3. 請實作與第一題接近的參數量，簡單的 DNN 模型，同時也說明其模型架構、訓練參數和準確率為何？

模型如下：

Layer (type)	Output Shape	Param #
Linear	[-1, 180]	8,847,540
ReLU	[-1, 180]	0
Linear	[-1, 128]	23,168
Dropout	[-1, 128]	0
ReLU	[-1, 128]	0
Linear	[-1, 64]	8,256
ReLU	[-1, 64]	0
Linear	[-1, 30]	1,950
PReLU	[-1, 30]	1
Linear	[-1, 11]	341

```

=====
Total params: 8,881,256

```

在 kaggle 上的分數為 0.45845。

4. 請說明由 1-3 題的實驗中你觀察到了什麼？

從第三題我們可以知道在同樣的參數量下，CNN 對於圖片能提供更好的解析能力，而減少 CNN 的層數也會讓模型對於圖像的分辨能力降低，因此可以知道 CNN 對於影像辨識能有很好的幫助。

5. 請嘗試 data normalization 及 data augmentation，說明實作方法並且說明實行前後對準確率有什麼樣的影響？

第一題的模型有使用 data normalization 與 data augmentation，而若是沒有 data augmentation 的話在 kaggle 上的 accuracy 降到 0.64734，可以知道我們的模型很需要較多的資料，而取消 normalization 後，kaggle 上的分數也降到 0.84339 了，可以知道我們的模型也對於資料的品質有一定的要求，只是相比資料量，沒有那麼需要而已。

6. 請說明你實作的 best model，其模型架構、訓練參數量和準確率為何？

參考 VGG 的架構後，將 model 分為捲積層以及全連接層，全連接層的部分由三個線性層組成，中間會使用 batch norm 並且使用 ReLU 作為活化函數；捲積層則是都用大小為三的 kernel，除了第一層以外，接下來每一層都是由兩層 CNN 組成，接著再接一層大小為 2 的 max pool。總訓練參數為 22269707，而在 Kaggle 上的 public score 為 0.88404。

7. 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

從圖 1 中可以看到，這個 model 最容易將 3 跟 0 搞混，1 跟 2 之間也很容易被搞混。

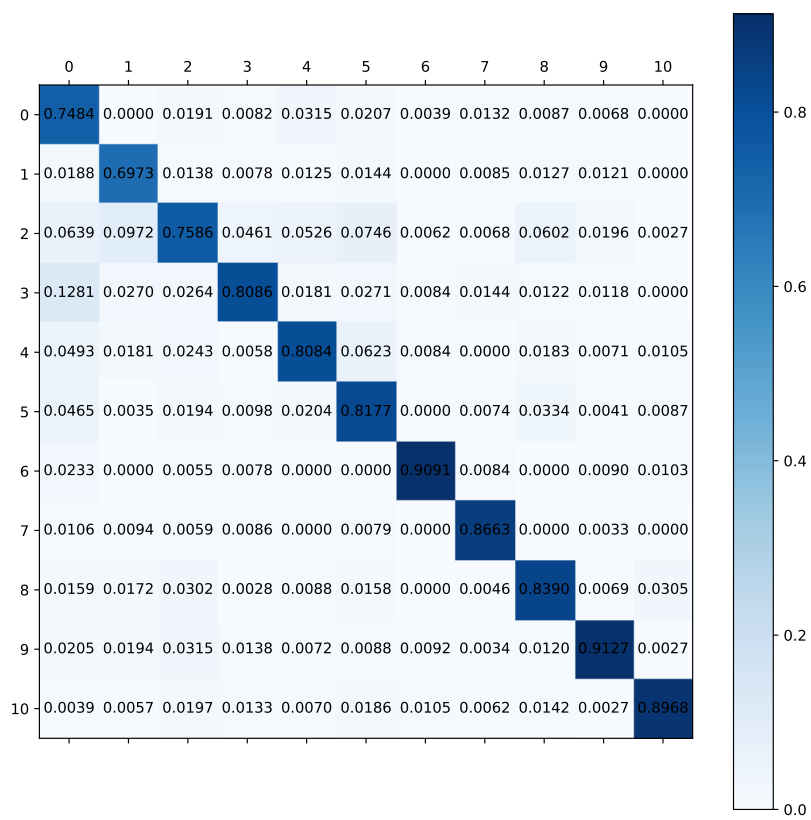


Figure 1: confusion matrix