

1. 請說明你實作的 CNN 模型(best model)，其模型架構、訓練參數量和準確率為何？(1%)

我的模型包含五層 CNN，每一層都是 Conv2d->BatchNorm2d->ReLU->MaxPool2d，再接上 4 層 DNN，DNN 中除了最後一層是結果，其他都加上 50% 的 Dropout 防止 overfitting。如下：

```
self.cnn = nn.Sequential(
    nn.Conv2d(3, 64, 3, 1, 1), # [64, 128, 128]
    nn.BatchNorm2d(64),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0), # [64, 64, 64]

    nn.Conv2d(64, 128, 3, 1, 1), # [128, 64, 64]
    nn.BatchNorm2d(128),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0), # [128, 32, 32]

    nn.Conv2d(128, 256, 3, 1, 1), # [256, 32, 32]
    nn.BatchNorm2d(256),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0), # [256, 16, 16]

    nn.Conv2d(256, 512, 3, 1, 1), # [512, 16, 16]
    nn.BatchNorm2d(512),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0), # [512, 8, 8]

    nn.Conv2d(512, 1024, 3, 1, 1), # [1024, 8, 8]
    nn.BatchNorm2d(1024),
    nn.ReLU(),
    nn.MaxPool2d(2, 2, 0), # [1024, 4, 4]
)
```

```
self.fc = nn.Sequential(
    nn.Linear(1024*4*4, 1024*4),
    nn.ReLU(),
    nn.Dropout(0.5, True),
    nn.Linear(1024*4, 1024),
    nn.ReLU(),
    nn.Dropout(0.5, True),
    nn.Linear(1024, 512),
    nn.ReLU(),
    nn.Linear(512, 11)
)
```

此外也有做 data augmentation，如對所有圖片的 RGB 做 normalize，另外還有隨機的水平和翻轉，旋轉，平移，縮放。如下：

```
train_transform = transforms.Compose([
    transforms.ToPILImage(),
    transforms.RandomHorizontalFlip(), #隨機將圖片水平翻轉
    # transforms.RandomRotation(15), #隨機旋轉圖片
    transforms.RandomAffine(30, (0.1, 0.1), (0.9, 1.1)),
    transforms.ToTensor(), #將圖片轉成 Tensor，並把數值normalize到[0,1](data normalization)
    transforms.Normalize(mean_std[0], mean_std[1]),
])
```

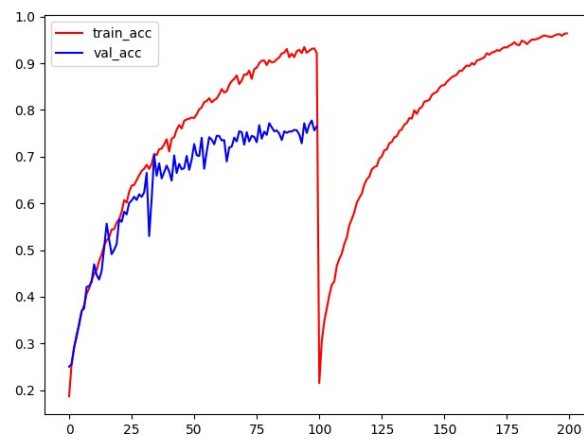
其他參數的部份跟助教的 sample code 都一樣，除了 epoch 調到 100（2 部份 training 都是）。

參數量是以以下 code 計算的：

```
if __name__ == "__main__":  
    model = Classifier()  
    params = model.parameters()  
  
    nParams = sum(p.numel() for p in model.parameters())  
    print(nParams)
```

參數量為 78113291

training accuracy 如下圖



在 kaggle 上的 accuracy 是 0.83502。

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？(1%)

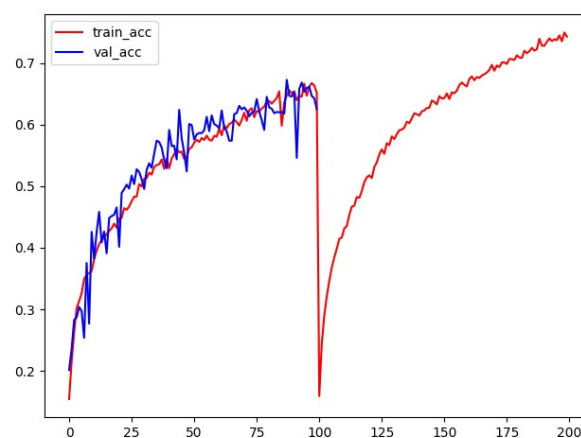
將第一題的 CNN 部份改成如下圖兩層 CNN，channel 的設定是為了使參數量與第一題的 model 差不多。

```
nn.Conv2d(3, 169, 3, 1, 1), # [169, 128, 128]
nn.BatchNorm2d(169),
nn.ReLU(),
nn.MaxPool2d(8, 8, 0), # [169, 16, 16]

nn.Conv2d(169, 4096, 3, 1, 1), # [4096, 16, 16]
nn.BatchNorm2d(4096),
nn.ReLU(),
nn.MaxPool2d(8, 8, 0), # [4096, 2, 2]
```

參數量為 78086105。

另外，training accuracy 為下圖：



而 kaggle accuracy 為 0.725。

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？ (1%)

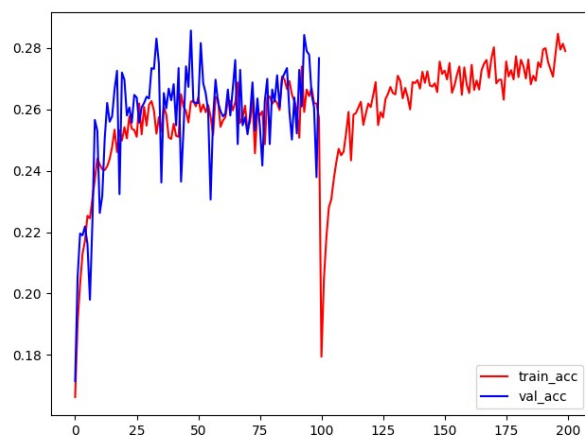
下圖是我的 DNN 模型

```
self.fc = nn.Sequential(  
    nn.Linear(3*128*128, 1546),  
    nn.ReLU(),  
    nn.Dropout(0.5, True),  
    nn.Linear(1546, 1024),  
    nn.ReLU(),  
    nn.Dropout(0.5, True),  
    nn.Linear(1024, 512),  
    nn.ReLU(),  
    nn.Linear(512, 11)  
)
```

channel 的設定一樣是為了使參數量和第一題差不多

參數量為 78105109。

下圖是 train accuracy

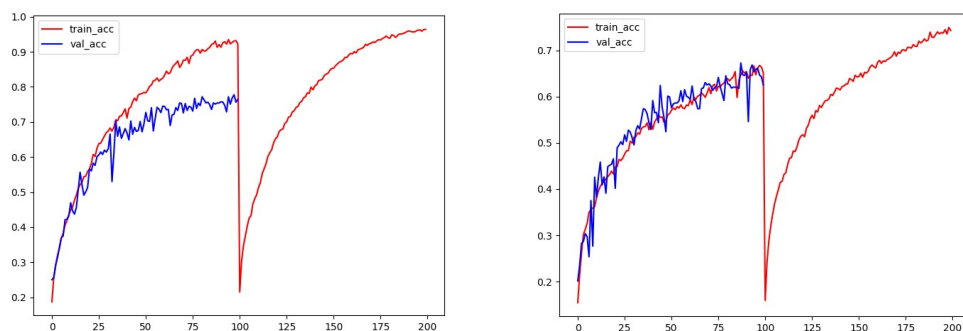


而最後 kaggle 的 accuracy 為 0.249。

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼? (1%)

a) 1 vs. 2

下圖分別是第一題 5 層 CNN 和第二題 2 層 CNN 的 training accuracy 的結果。

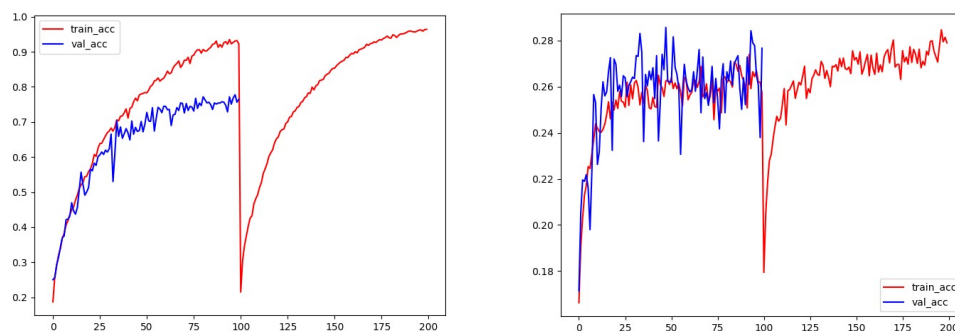


而最後在 kaggle 上的 accuracy 分別為 0.835 和 0.725。

根據實驗結果，相同參數下 5 層 CNN 的 model 比較複雜，可以達到比較高的 accuracy，最後的分數也比較高。但是較容易 overfitting，所以 training accuracy 會明顯低於 validation accuracy。2 層 CNN 的 training accuracy 和 validation accuracy 就非常一致。

b) 1 vs. 3

下圖分別是第一題 5 層 CNN 和第三題只有 DNN 的 training accuracy 的結果。



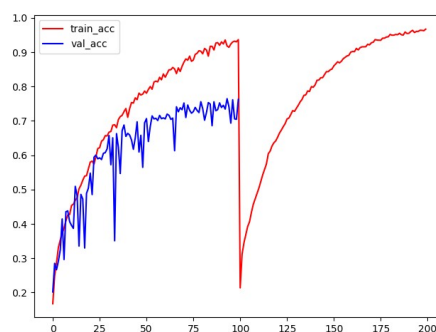
kaggle 的分數分別為 0.835, 0.249。

結果與預期的相同，CNN 對於影像辨識比 DNN 好很多，可以根據圖像的特性有效地重複利用參數以減少參數量。

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？(1%)

a) data normalization

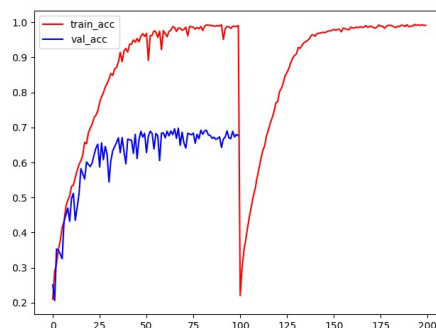
我第一題的版本是有 data normalization 的，normalize 的方法是對 training set 和 validation set 合起來的所有 data 找出 RGB 三個 channel 的 mean, std，然後在 transform 裡 normalize。所以這邊把 normalize 去掉後結果如下圖：



可以看到和第一題沒差多少，kaggle 分數 0.843 甚至更高，可見這次作業 normalize 其實沒什麼用。

b) data augmentation

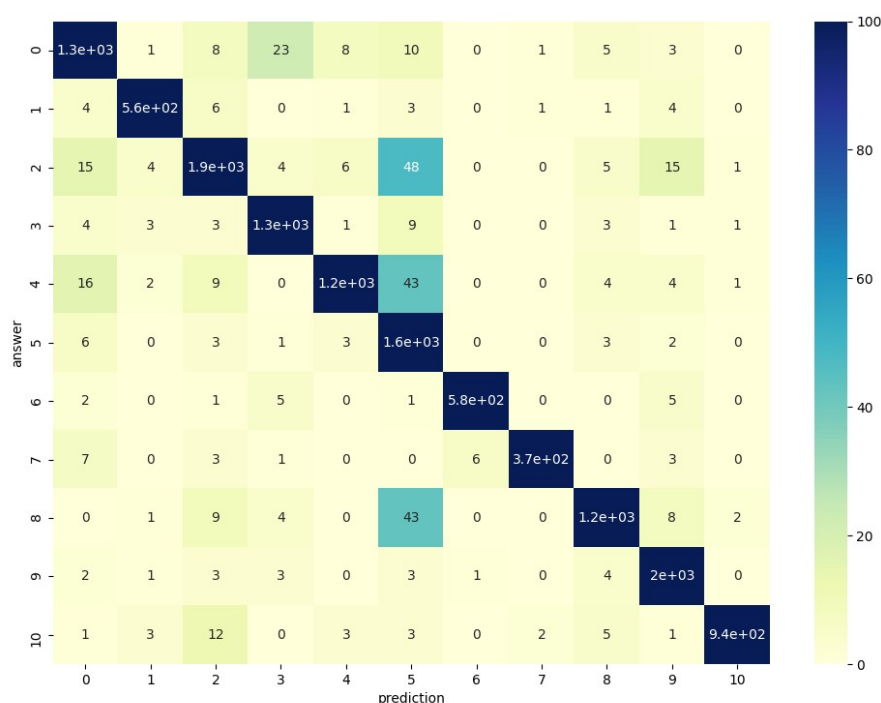
同樣地第一題版本也是有做 data augmentation 的，所以這邊把 randomflip 和 randomaffine 去掉，結果如下：



可以看到 training set 的 accuracy 很高，但是 validation set 的低很多，非常 overfitting，kaggle 的分數也只有 0.758。證明對 image 做 data augmentation 很有用。

6. 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析](1%)

下圖是 confusion matrix



可以看到 2/4/8(dessert/fried food/seafood)被 model 認為是 5(meat)的機率相對很高。

我看了一下 data 覺得 fried food 和 seafood 錯得還還算合理啦。炸的肉到底要算肉還是炸我自己都不知道了，然後 seafood 也是蠻難的，生魚片也是一種肉啊，牡蠣也是一塊肉放在貝殼上面，所以 model 辨識錯可以原諒。甜點就比較奇怪了，不過我發現 data 裡有一個甜點的照片是兩個小孩？？？大概是他們害的。