

ML hw3 report

學號：b07901112 系級：電機二 姓名：劉聿珉

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%)

模型架構：

```
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, 512, 3, 1, 1), # [512, 8, 8]  
    nn.BatchNorm2d(512),  
    nn.ReLU(),  
    nn.MaxPool2d(2, 2, 0), # [512, 4, 4]  
  
)  
self.fc = nn.Sequential(  
    nn.Linear(512*4*4, 1024),  
    nn.Dropout(0.6),  
    nn.ReLU(),  
    nn.Linear(1024, 512),  
    nn.Dropout(0.6),  
    nn.ReLU(),  
    nn.Linear(512, 11)  
)
```

Epoch_num = 120, learning rate = 0.0005, batch_size = 128(沒有改動)

準確率：0.83144

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%)

第一題的參數量為 12,833,803 而 CNN 的深度為 5 層，在這題中我將 CNN 的層數變成 2 層模型架構為：

```
self.cnn = nn.Sequential(  
    nn.Conv2d(3, 5, 3, 1, 1),  
    nn.BatchNorm2d(5),  
    nn.ReLU(),  
    nn.MaxPool2d(2, 2, 0),  
  
    nn.Conv2d(5, 12, 3, 1, 1),  
    nn.BatchNorm2d(12),  
    nn.ReLU(),  
    nn.MaxPool2d(2, 2, 0),  
  
)  
  
self.fc = nn.Sequential(  
    nn.Linear(12*32*32, 1024),  
    nn.Dropout(0.6),  
    nn.ReLU(),  
    nn.Linear(1024, 512),  
    nn.Dropout(0.6),  
    nn.ReLU(),  
    nn.Linear(512, 11)  
)
```

參數量變為 13,115,105 參數量與第一題接近。而第一題的 Training acc 為 0.95023，Val acc 為 0.74606。此題的 model(將 CNN 層數減半)的 Train acc 為 0.53375，Val acc 為 0.48688。由實驗結果可以看到將 CNN 的層數減半會使正確率下滑。

要討論這個問題，首先釐清我們在這題中不僅減少了 CNN 的層數，還減少了 filter 的數量以及 maxpool 的數量，filter 的數量減少就像是每一層可以觀察的特徵減少，而 maxpool 的減少會使機器在學習的時候看不到重點，無法將被 filter 發現的特徵放大檢視，以至於看到很多非特徵的畫面，而在實作 CNN 的時候我們往往需要通過很多層的 filter，多次的檢測才可以形成一個接近人眼所看得出的結果，所以將 CNN 的深度減少便會降低畫面的可辨認性。綜合以上特性，將 CNN 的層數減少後雖然參數量相差不多，但卻會使機器在學習的過程中非常沒有效率，無法看到辨別物件的特徵，造成準確率降低的問題。

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%)

第三題的實作比第二題更直接，直接將 CNN 的部分移除，我的 DNN 模型的總參數量為 12,617,483，Train acc 為 0.14990，Val acc 為 0.14577。結構為：

```
self.fc = nn.Sequential(
    nn.Linear(3*128*128, 256),
    nn.Dropout(0.6),
    nn.ReLU(),
    nn.Linear(256, 128),
    nn.Dropout(0.6),
    nn.ReLU(),
    nn.Linear(128, 11)
)
```

由結果可以看到，只有 DNN 在辨認圖像時效果極差，在經過 120 個 epoch 後 train acc 還是大約只有 1.5 成。

這部分的實驗結果之所以會那麼差的原因就好比他將一個圖片的每一個小方格都一視同仁的丟到 model 裡面 train，雖然理論上也是有機會訓練出好的 model 但是這種方法十分的沒有效率，第一題的 model 將圖片經過 CNN 就像是將圖片中的一隻鳥分類成鳥喙、翅膀、爪子的部分，之後再丟給 DNN 藉由這些特徵辨別出他是一隻鳥，而第三題的 model 則是直接跳過 CNN 的部分，這樣會導致在 train 時會把圖片看得太細以致於直接忽略掉判斷所需的重要特徵。

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

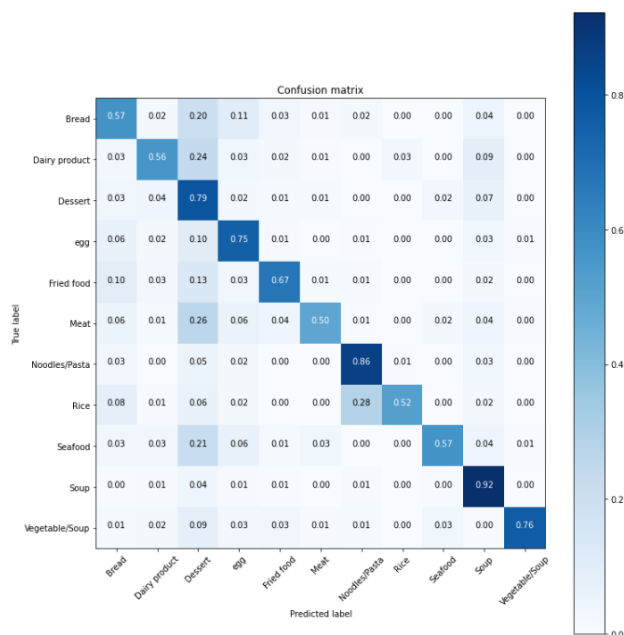
從 1~3 題的實驗中，我發現在圖像辨識中 CNN 是一件非常重要解析圖片的方法，CNN 層太少可能會使機器看得太細使得效率低落，在有限的時間內無法得到令人滿意的結果，但是如果 CNN 層太多的話可能會因為看得太大概反而得到反效果，如何找到一個合適的 CNN 層數以及大小便是需要下功夫的地方了。

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%)

正常的 data : val acc = 0.66501。data augmentation : val acc = 0.74839。data normalization : val acc = 0.64052。data normalization + augmentation val acc = 0.75830 以上的數據 Train acc 都至少有 98% 以上。

有以上結果可以知道 **augmentation** 對於正確率的提升是比較有幫助的，因為對於人眼來說，就算將圖片做反轉或是顛倒我們所判斷出的東西理當也要是同一個，當我們在 **train model** 時理當也要得到相同的結果，將 **data** 做隨機的反轉可以增加 **data** 的隨機性，使得 **data** 更有效。而 **normalization** 的效果並不明顯，再單做 **normalization** 的情況下甚至還便糟(但是不管 **acc** 是上升還是下降，他上升下降的幅度還是非常的少可以說是幾乎沒有差別)，可能是因為這次的 **data** 處理並不是合用 **normalization** 來優化的原因所致。

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%)



由上圖可以知道大致上的判斷其實都是正確的，只是其中還是可以看出一些玄機。像是其他食物很容易被機器判斷成 **dessert**，我猜可能的原因是因為甜點沒有一個固定的形狀以及顏色(甚至可以是牛排形狀的蛋糕之類的)，所以很容易讓機器誤判。而 **Rice** 很容易被看成 **Noodles** 我猜可能是因為顏色相近只有紋路不一樣的，所以很難將這兩類做分別。從表中得出 **Dessert**, **Egg**, **Fried food**, **Noodles**, **Soup**, **Vegetable** 這幾個類別判斷成功的機會比其他類別都還要高，原因可能是特徵明顯辨別度高的原因所致。