作業三 Report

學號：B06901020 系級：電機二 姓名：張恆瑞

1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？

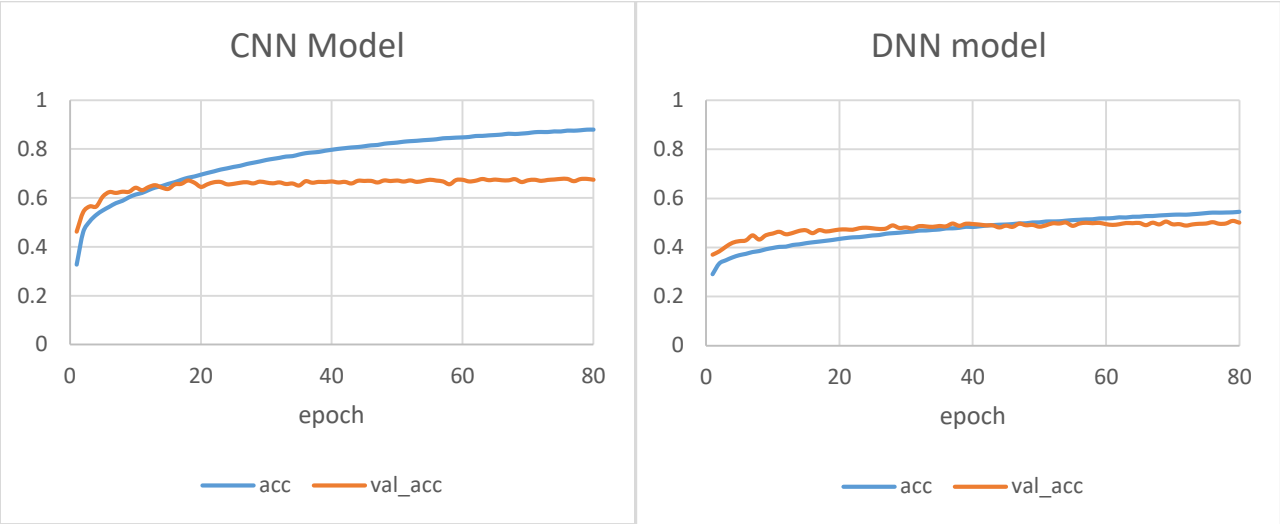
答：我使用的 CNN model 和助教給的 sample code 是幾乎一樣的，只是多加了一些 batch normalization 和 dropout 層，很明顯 CNN 的在影像分類上表現比簡單 DNN 好很多。

後來，我將三個相同架構但利用不同程度 data augmentation 來訓練的 CNN model 做 *ensemble* 得到比原本更高的準確率，使得在 public 和 private 得到 68.8%和 67.5%的準確率。

模型	CNN model	DNN model
架構	nn.Conv2d(1, 64, 4, 2, 1), nn.BatchNorm2d(64), nn.LeakyReLU(0.2), nn.Conv2d(64, 64, 3, 1, 1), nn.BatchNorm2d(64), nn.LeakyReLU(0.2), nn.MaxPool2d(2, 2, 0), nn.Dropout2d(p=0.25), nn.Conv2d(64, 128, 3, 1, 1), nn.BatchNorm2d(128), nn.LeakyReLU(0.2), nn.Conv2d(128, 128, 3, 1, 1), nn.BatchNorm2d(128), nn.LeakyReLU(0.2), nn.MaxPool2d(2, 2, 0), nn.Dropout2d(p=0.25), nn.Conv2d(128, 256, 3, 1, 1), nn.BatchNorm2d(256), nn.LeakyReLU(0.2), nn.Conv2d(256, 256, 3, 1, 1), nn.BatchNorm2d(256), nn.LeakyReLU(0.2), nn.MaxPool2d(2, 2, 0), nn.Dropout2d(p=0.25) nn.Linear(256*3*3, 1024), nn.BatchNorm1d(1024), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(1024, 512), nn.BatchNorm1d(512), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(512, 7)	nn.Linear(48 * 48, 1024), nn.BatchNorm1d(1024), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(1024, 1024), nn.BatchNorm1d(1024), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(1024, 512), nn.BatchNorm1d(512), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(512, 256), nn.BatchNorm1d(256), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(256, 128), nn.BatchNorm1d(128), nn.LeakyReLU(0.2), nn.Dropout(p=0.5), nn.Linear(128, 7)
參數量	4038279	4105735
Val acc	0.67816	0.50888
Public	0.66926	0.49010
Private	0.65338	0.49317

2. (1%) 承上題，請分別畫出這兩個 model 的訓練過程 (i.e., loss/accuracy v.s. epoch)

答：以下兩張圖為 training 的準確率變化，可以觀察出 DNN 始終無法達到 CNN 的準確率。



3. (1%) 請嘗試 data normalization, 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？

答：我利用不同的線性變換自己實作了四種 data augmentation，包括鏡射、旋轉、縮放、斜向伸縮，使得訓練資料變成原本的 5 倍。觀察使用 data normalization 和 data augmentation 後的準確率，可以發現有用 *data augmentation* 後的準確率有大幅提升，但 data normalization 的幫助相對很小。下表為將 puplic 和 private 準確率做平均之後的結果。

	沒有 normalization	有 normalization
沒有 augmentation	0.62218	0.62287
有 augmentation	0.65826	0.66132

4. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

答：藉由 confusion matrix 可以看出，angry 和 hate 較容易被搞混，理由也很簡單，這兩類表情都表現出夾雜厭惡與不悅的樣子，所以被搞混是可以理解的。