

- (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？

(Collaborators: B05901158 林緯璋)

CNN model：使用簡化版的VGG，有使用batch normalization和dropout。Conv2D的部分，filter大小皆為(3, 3)，一共有六層，每層filter數量為64-64-128-128-128-128；Dense的部分有三層，unit數量為512-512-7。總參數量為1,875,143。

DNN model：和CNN model一樣有使用batch normalization和dropout。Dense有五層，unit數量為512-512-512-256-7。總參數量為1,845,767。

CNN和DNN model的詳細架構如下：

CNN model		DNN model
Conv2D(64, (3, 3), padding = 'same')	Flatten	Flatten
BatchNormalization	Dense(512)	Dense(512)
Activation('relu')	BatchNormalization	BatchNormalization
Conv2D(64, (3, 3))	Activation('relu')	Activation('relu')
BatchNormalization	Dropout(0.35)	Dropout(0.4)
Activation('relu')	Dense(512)	Dense(512)
MaxPooling2D(2, 2)	BatchNormalization	BatchNormalization
Dropout(0.25)	Activation('relu')	Activation('relu')
Conv2D(128, (3, 3), padding = 'same')	Dropout(0.35)	Dropout(0.4)
Activation('relu')	Dense(7)	Dense(512)
Conv2D(128, (3, 3))	Activation('softmax')	BatchNormalization
Activation('relu')		Activation('relu')
MaxPooling2D(2, 2)		Dropout(0.4)
Dropout(0.25)		Dense(256)
Conv2D(128, (3, 3), padding = 'same')		BatchNormalization
BatchNormalization		Activation('relu')
Activation('relu')		Dropout(0.4)
Conv2D(128, (3, 3))		Dense(7)
BatchNormalization		Activation('softmax')
Activation('relu')		
MaxPooling2D(2, 2)		
Dropout(0.25)		

訓練參數：CNN和DNN model皆使用batch_size = 100, epochs = 20。有使用相同的data normalization和data augmentation。

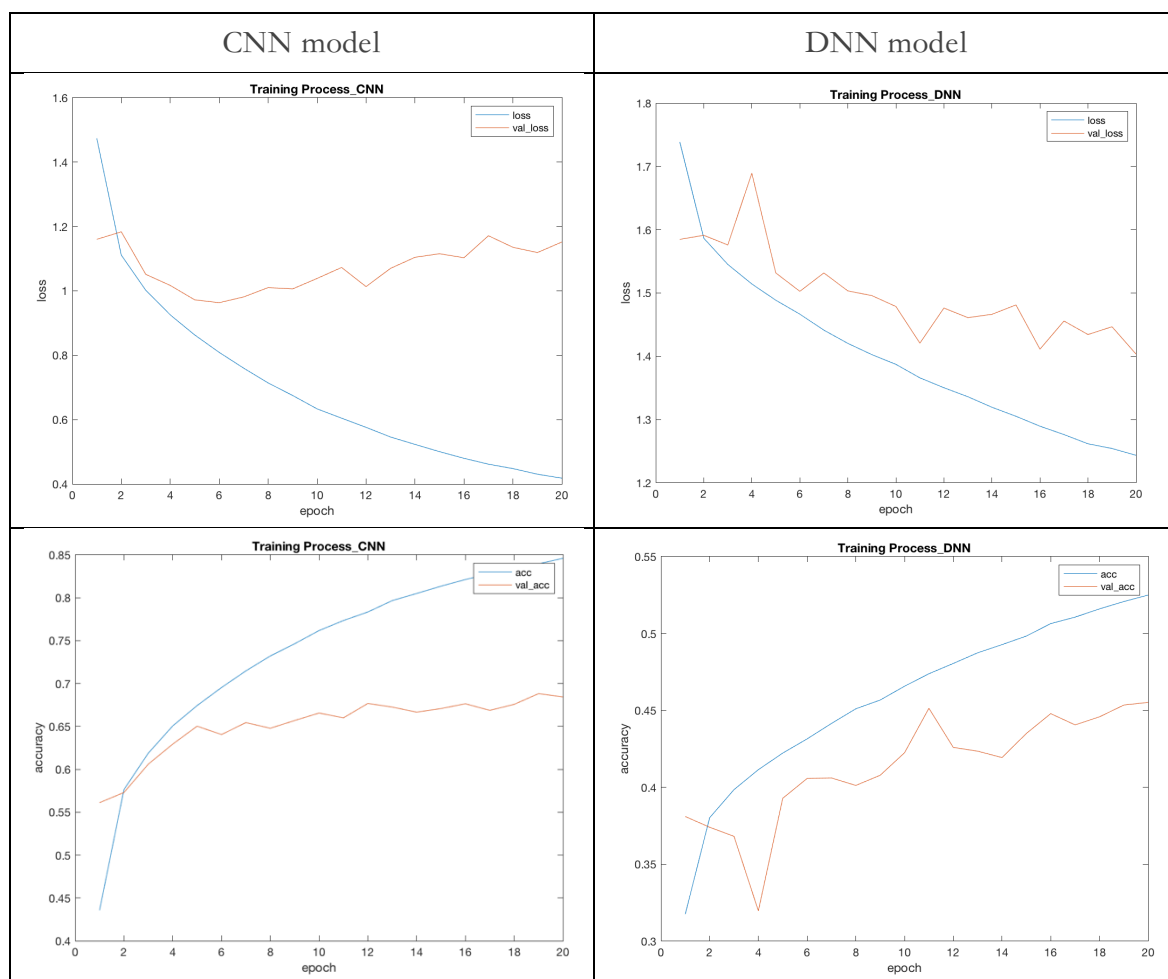
準確率：

	CNN model	DNN model
Public	0.68487	0.46976
Private	0.68208	0.45667

CNN model的表現明顯比DNN model好。

另外，kaggle上準確率最高的結果是使用三個不同的CNN model預測出來的結果做 majority vote。三個model的架構大致相同，只有在最後三層dense的部分，分別使用512-128-7、512-256-7、512-512-7。

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)
(Collaborators: 無)



不論是loss還是accuracy，CNN的表現都比DNN好，而且validation set的準確率在CNN model的趨勢比較穩定。

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？

(Collaborators: 無)

Data normalization: 將data除以255，使所有數字介於0~1之間

Data augmentation: 使用Keras的ImageDataGenerator實作隨機旋轉和水平翻轉，產生五倍的training data

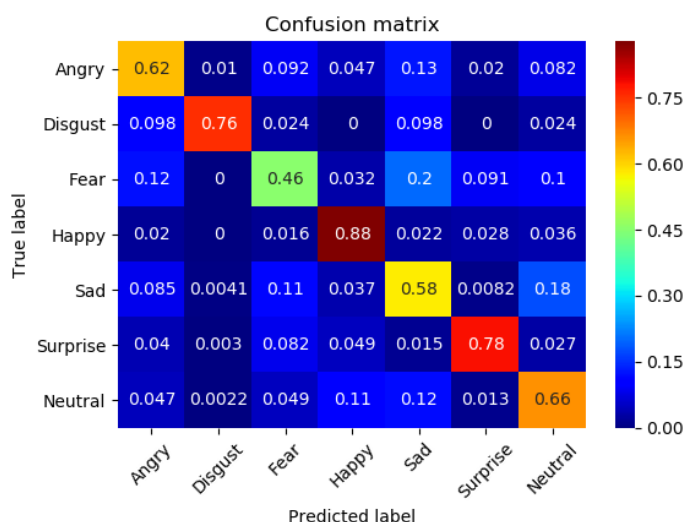
→ ImageDataGenerator(rotation_range = 10, horizontal_flip = True)

	Public	Private
raw	0.64614	0.63109
normalization	0.64363	0.63583
normalization + augmentation	0.68487	0.68208

Normalization對準確率幫助不大，可能是因為model有使用batch normalization，因此是否有做data normalization影響不大。而data augmentation可以大幅提高準確率，大約提高了4%。

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]

(Collaborators: 無)



觀察confusion matrix，容易分類錯誤的有：

Angry誤判成Sad、Fear誤判成Angry、Fear誤判成Sad、Sad誤判成Fear、

Sad誤判成Neutral、Neutral誤判成Happy、Neutral誤判成Sad

因此，我認為Angry、Fear、Sad彼此容易用混，還有Sad和Neutral也容易用混。