Machine Learning - 2021Fall - HW3

學號：R10921098 系級：電機所碩一 姓名：楊仁傑

1. (1%) 請以block diagram或是文字的方式說明這次表現最好的model使用哪些layer module(如 Conv/Linear 和各類 normalization layer) 及連接方式(如一般forward 或是使用 skip/residual connection)，並概念性逐項說明選用該 layer module 的理由。

Ans:

參考"Convolutional_Neural_Network_Hyperparameters_optimization_for_Facial_Emotion_Recognition"文中所提出的模型進行修改。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 256)       2560
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 60, 60, 512)       1180160
_________________________________________________________________
batch_normalization (BatchNo (None, 60, 60, 512)       2048
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 512)       0
_________________________________________________________________
dropout (Dropout)            (None, 30, 30, 512)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
batch_normalization_1 (Batch (None, 28, 28, 512)       2048
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 512)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 12, 12, 256)       1179904
_________________________________________________________________
batch_normalization_2 (Batch (None, 12, 12, 256)       1024
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 256)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6, 6, 256)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 4, 4, 512)         1180160
_________________________________________________________________
batch_normalization_3 (Batch (None, 4, 4, 512)         2048
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 512)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 2, 2, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 256)               524544
_________________________________________________________________
batch_normalization_4 (Batch (None, 256)               1024
_________________________________________________________________
dropout_4 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 1799
=================================================================
Total params: 6,437,127
Trainable params: 6,433,031
Non-trainable params: 4,096
_________________________________________________________________
```

[Discussion]

雖然本次作業在使用此模型的情況下，可以取得不錯的成績。但由於模型中的參數量過大且模型較為複雜，在使用RTX 3080 10GB的環境，在batch_size設置為128時，系統就會提示說顯存不足。此外，在使用增強數據的情況下，訓練50 epochs，約耗時4個小時，對比於原先設計的簡易模型，用時約1個小時，訓練的效率有一定程度的落差。

2. (1%) 嘗試使用 augmentation/early-stopping/ensemble 三種訓練 trick 中的兩種，說明實作細節並比較有無該 trick 對結果表現的影響(validation 或是 testing 擇一即可)。

Ans:

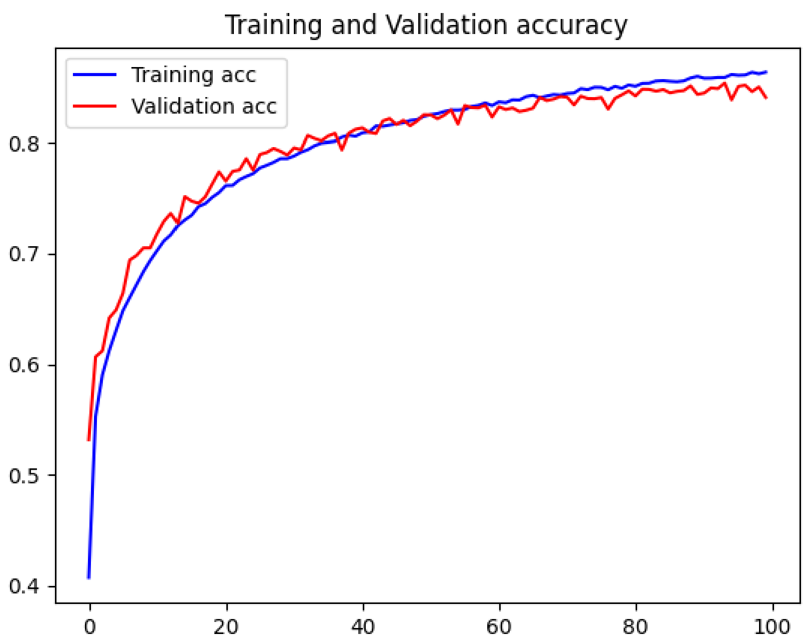
[augmentation]

HW3_v3_Adam_flip.csv 6 days ago by CHIEH add submission details	0.65457	0.67142
HW3_v3_Adam.csv 6 days ago by CHIEH add submission details	0.64342	0.64942

對圖片進行增強，例如說旋轉一定角度，水平翻轉，縮放等等。使用這些數據進行訓練，能保證模型有更多的數據，進而防止overfitting發生。

從上圖中可以看到，在使用同樣的模型架構及epoch下，僅僅是增加了水平翻轉的資料，就使準確率提高了不少。

[Early Stopping]



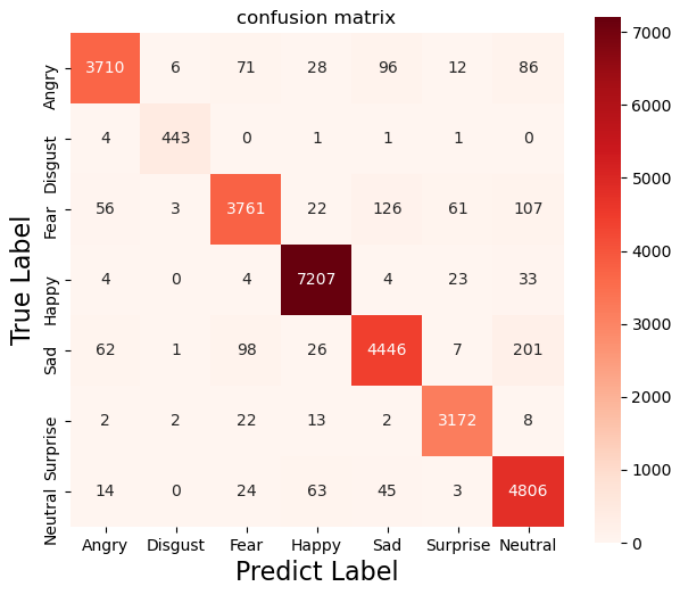
HW3_v4.csv 5 days ago by CHIEH epoch = 50	0.69257	0.69514
HW3_v4.2.csv 4 days ago by CHIEH epoch = 100	0.68400	0.69114

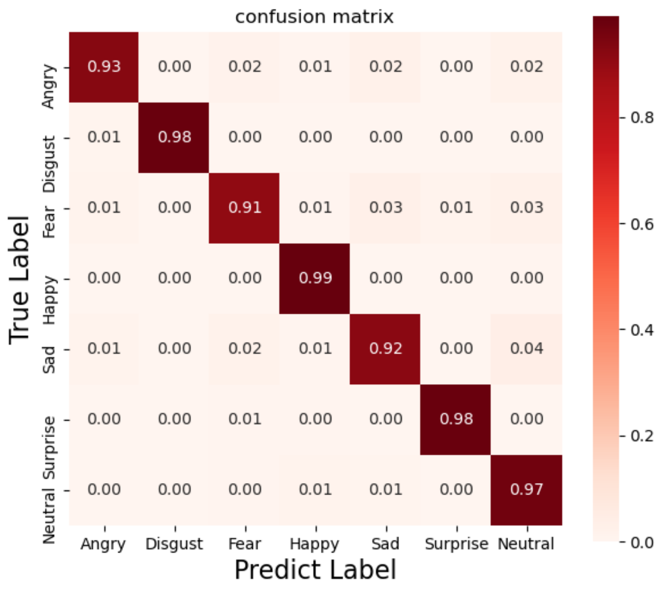
在每一個 epoch 結束時計算validation accuracy，當accuracy不再提高就停止訓練。優點是解決手動設置 epoch 數的問題（節省訓練模型時間），還能防止 overfitting。

由上方的圖中可以得知，validation accuracy在50輪附近就沒有明顯的增加，因此可以看到雖然使用相同模型架構，僅訓練50輪的在Test Data上預測結果比訓練100輪的結果來的好上許多，這也就意味著訓練100輪已經產生了overfitting的結果，因此表現得較差。

3. (1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，並簡單說明。

Ans:





[Discussion]

由confusion matrix中可得知(使用train data做預測)，在判斷Disgust, Happy, Suprise, Neutral這幾種臉部表情普遍有不錯表現，個人認為因為這幾種的臉部表情(嘴巴部分)有著較為明顯的差異，所以模型在訓練過程中可以較為容易的區分出來。而剩下的Angry, Fear, Sad由於表情特徵比較接近，所以可以從confusion matrix中看出，模型容易預測錯誤這幾種表情。

4. (1%) 請統計訓練資料中不同類別的數量比例，並說明：

對 testing 或是 validation 來說，不針對特定類別，直接選擇機率最大的類別會是最好的結果嗎？針對上述內容，是否存在更好的方式來提升表現？例如設置不同條件來選擇預測結果/變更訓練資料抽樣的方式，或是直接回答「否」(但需要給出支持你論點的論述)

Ans:

我認為若直接選擇機率最大的類別不一定是最好的選擇，承第三題所述，在Angry, Fear, Sad這三種情況下，模型容易預測錯誤。換句話而言，可能這三者的預測機率是十分接近的，而若我們只一昧的選取機率最大的情況，那我認為這是一個不好的方法。

因此我認為，我們可以使用複數個模型作為預測，若某次模型中，所有的預測機率並沒有特別突出的，也就是可能有兩到三組十分接近的預測數據的話，我們將其保留，並使用另外一組模型進行預測，進而得到新的預測結果。實施多次測試後，或許能夠有較好的預測結果。

(若多個模型都無法成功預測的話，可能代表此筆資料應該是具有問題或者為資料集中的特例，例如第二次作業中所討論的問題(見下方敘述))

> 從Figure1.和Figure2.中可知，大多數的模型預測結果錯誤是因為落於0.5上下，屬於給定的特徵不足，使得模型無法正確的辨識出該對象收入是否大於50K。少部分(e.g. index=5)才是模型完全預測錯誤，導致輸出結果錯誤，個人認為這應該屬於資料中的特例，因此模型才無法利用現有的特徵去進行判斷。
>
> index=5：
>
> 37 Private 284582 Masters 14 Married-civ-spouse Exec-managerial Wife White Female 0 0 40 United-States <=50K
>
> 我們將index=5的train data叫出來檢查，發現此人在私營企業上班，工作為行政管理階級，同時有碩士學位，並且已婚，美國籍，就現有的特徵而言很難以想像此人的收入<=50K，故承上所述，我認為這應該是屬於資料中的特例，故模型無法正常的預測。(註：此處也有可能是標籤錯誤，但不在討論範圍內)

5. (3%) Refer to math problem

[(https://hackmd.io/@GfOkB4kgS66YhhM7j6TJew/SJy_akYUK)](https://hackmd.io/@GfOkB4kgS66YhhM7j6TJew/SJy_akYUK)

Convolution

Ans:

$$(B, \lfloor \frac{W + 2p_1 - k_1}{s_1} \rfloor, \lfloor \frac{W + 2p_2 - k_2}{s_2} \rfloor, \text{output\_channels})$$

Batch Normalization

Ans:

$$
\begin{aligned}
\frac{\partial l}{\partial \hat{x}^i} &= \frac{\partial l}{\partial y^i}\frac{\partial y^i}{\partial \hat{x}^i} \\
&= \frac{\partial l}{\partial y^i}\frac{\partial(\gamma \hat{x}^i + \beta)}{\partial \hat{x}^i} \\
&= \frac{\partial l}{\partial y^i}\gamma
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial \sigma_\beta^2} &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}\frac{\partial \hat{x}^i}{\partial \sigma_\beta^2} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}\frac{\partial(\frac{x_i - u_B}{\sqrt{\sigma_\beta^2+\epsilon}})}{\partial \sigma_\beta^2} \\
&= \frac{-1}{2}(\sigma_\beta^2+\epsilon)^{\frac{-3}{2}}\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}(x_i - u_B)^{\frac{1}{2}}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial u_B} &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}\frac{\partial \hat{x}^i}{\partial u_B} + \frac{\partial l}{\partial \sigma_\beta^2}\frac{\partial \sigma_\beta^2}{\partial u_B} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}\frac{-1}{\sqrt{\sigma_\beta^2+\epsilon}} + \frac{\partial l}{\partial \sigma_\beta^2}\frac{-2}{m}\underbrace{\sum_{i=1}^{m}(x_i - u_B)}_{=0}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial \hat{x}^i} &= \frac{\partial l}{\partial \hat{x}^i}\frac{\partial \hat{x}^i}{\partial x^i} + \frac{\partial l}{\partial \sigma_\beta^2}\frac{\partial \sigma_\beta^2}{\partial x^i} + \frac{\partial l}{\partial u_B}\frac{\partial u_B}{\partial x^i} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}^i}\frac{1}{\sqrt{\sigma_\beta^2+\epsilon}} + \frac{\partial l}{\partial \sigma_\beta^2}\frac{2(x_i - u_B)}{m} + \frac{\partial l}{\partial u_B}\frac{1}{m}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial \gamma} &= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \gamma} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial(\gamma \hat{x}^i + \beta)}{\partial \gamma} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\hat{x}^i
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial \beta} &= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \beta} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial(\beta \hat{x}^i + \beta)}{\partial \beta} \\
&= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}
\end{aligned}
$$

Softmax and Cross Entropy

Ans:

As we known,

$$L_t = -y_t \, log\hat{y}_t$$

$$\hat{y}_t = \text{softmax}(z_t) = \frac{e^{z_t}}{\sum_i e^{z_i}}$$

and some differential formulas,

$$
\begin{aligned}
\frac{df(x)g(x)}{dx} &= \frac{df(x)}{dx}g(x) + f(x)\frac{dg(x)}{dx} \\
\frac{d\frac{f(x)}{g(x)}}{dx} &= \frac{\frac{df(x)}{dx}g(x) - f(x)\frac{dg(x)}{dx}}{(g(x))^2}
\end{aligned}
$$

Then, $\frac{\partial L}{\partial z_t}$ can be rewritten as:

$$
\begin{aligned}
\frac{\partial L}{\partial z_t} &= \frac{\partial L}{\partial \hat{y}_t}\frac{\hat{y}_t}{\partial z_t} \\
&= \frac{\partial(-y_t\ log\hat{y}_t)}{\partial \hat{y}_t}\frac{\partial \frac{e^{z_t}}{\sum_i e^{z_i}}}{\partial z_t} \\
&= [\frac{\partial(-y_t)}{\partial \hat{y}_t}(log\ \hat{y}_t) + (-y_t)\frac{\partial(log\ \hat{y}_t)}{\partial \hat{y}_t}][\frac{\frac{\partial e^{z_t}}{\partial z_t}\sum_i e^{z_i} - e^{z_t}\frac{\partial \sum_i e^{z_i}}{\partial z_t}}{(\sum_i e^{z_i})^2}] \\
&= (0 - y_t\frac{1}{\hat{y}_t})(\frac{e^{z_t}\sum_i e^{z_i} - (e^{z_t})^2}{(\sum_i e^{z_i})^2}) \\
&= (\frac{-y_t}{\hat{y}_t})\frac{e^{z_t}}{\sum_i e^{z_i}}(1 - \frac{e^{z_t}}{\sum_i e^{z_i}}) \\
&= (\frac{-y_t}{\hat{y}_t})\hat{y}_t(1 - \hat{y}_t) \\
&= -y_t + y_t\hat{y}_t \\
&= \hat{y}_t - y_t
\end{aligned}
$$

Adaptive learning rate based optimization

Ans:

(a)

As we known,

$$
\begin{cases} m^t = \beta_1 m^{t-1} + (1-\beta_1)g^t \\ v^t = \beta_2 v^{t-1} + (1-\beta_2)(g^t)^2 \end{cases}
$$

If we substitute $[0, t]$ into m_t, we can get:

$$
\begin{aligned}
m_0 &= (1-\beta_1)g^0 \\
m_1 &= \beta_1 m^0 + (1-\beta_1)g^1 \\
&= \beta_1(1-\beta_1)g^0 + (1-\beta_1)g^1 \\
m_2 &= \beta_1 m^1 + (1-\beta_1)g^2 \\
&= \beta_1^2(1-\beta_1)g^0 + \beta_1(1-\beta_1)g^1 + (1-\beta_1)g^2 \\
&\vdots \\
m^t &= \beta_1 m^{t-1} + (1-\beta_1)g^t \\
&= \beta_1^t(1-\beta_1)g^0 + \beta_1^{t-1}(1-\beta_1)g^1 + \cdots + \beta_1(1-\beta_1)g^{t-1} + (1-\beta_1)g^t \\
&= (1-\beta_1)(\beta_1^t g^0 + \beta_1^{t-1}g^1 + \cdots + \beta_1 g^{t-1} + g^t)
\end{aligned}
$$

So, we can know that $m^t = (1-\beta_1)\sum_{i=0}^{t}\beta_1^{t-i}g^i$

Using the same method, we can know that $v^t = (1-\beta_2)\sum_{i=0}^{t}\beta_2^{t-i}(g^i)^2$

If g_0 is a set of zero vectors, then m^t, v^t can be rewritten as:

$$
\begin{cases} m^t = (1-\beta_1)\sum_{i=1}^{t}\beta_1^{t-i}g^i \\ v^t = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}(g^i)^2 \end{cases}
$$

Therefore,

$$
\begin{aligned}
A = (1-\beta_1),\ B = \beta_1^{t-i} \\
C = (1-\beta_2),\ D = \beta_2^{t-i}
\end{aligned}
$$

(b)

If $\beta_1 = 0$, then

$$
m^t = \beta_1 m^{t-1} + (1-\beta_1)g^t = g^t
$$

$$
\hat{m}^t = \frac{m^t}{(1-\beta_1^t)} = \frac{g^t}{1} = g^t
$$

As we known,

$$\because (x^n - 1) = (x-1)(x^{n-1} + x^{n-2} + \cdots + x + 1)$$
$$\Rightarrow (1 - x^n) = (1-x)(x^{n-1} + x^{n-2} + \cdots + x + 1)$$

$$\therefore (1 - \beta_2^t) = (1-\beta_2)(\beta_2^{t-1} + \beta_2^{t-2} + \cdots + \beta_2 + 1)$$
$$\Rightarrow (1 - \beta_2^t) = (1-\beta_2)(\sum_{i=1}^{t} \beta_2^{t-i})$$

If $\beta_2 \rightarrow 1$, then

$$\begin{aligned} v^t &= \beta_2 v^{t-1} + (1-\beta_2)(g^t)^2 \\ &= (1-\beta_2)\sum_{i=0}^{t} \beta_2^{t-i}(g^i)^2 \end{aligned}$$

$$\begin{aligned} \hat{v}^t &= \frac{v^t}{(1-\beta_2^t)} \\ &= \frac{(1-\beta_2)\sum_{i=0}^{t} \beta_2^{t-i}(g^i)^2}{(1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i}} \\ &= \frac{\sum_{i=0}^{t} \beta_2^{t-i}(g^i)^2}{\sum_{i=1}^{t} \beta_2^{t-i}} \\ &= \frac{\sum_{i=0}^{t}(g^i)^2}{t} \end{aligned}$$

Finally, we substitute $\hat{m}^t$, $\hat{v}^t$, $\eta = \eta_0 t^{\frac{-1}{2}}$ into $w^t = w^{t-1} - \frac{\eta}{\sqrt{\hat{v}^t}}\hat{m}^t$ to get

$$\begin{aligned} w^t &= w^{t-1} - \frac{\eta}{\sqrt{\hat{v}^t}}\hat{m}^t \\ &= w^{t-1} - \frac{\eta_0 t^{\frac{-1}{2}}}{\sqrt{\frac{\sum_{i=0}^{t}(g^i)^2}{t}}} g^t \\ &= w^{t-1} - \frac{\eta_0 t^{\frac{-1}{2}}}{t^{\frac{-1}{2}}\sqrt{\sum_{i=0}^{t}(g^i)^2}} g^t \\ &= w^{t-1} - \frac{\eta_0}{\sqrt{\sum_{i=0}^{t}(g^i)^2}} g^t \end{aligned}$$

If g_0 is a set of zero vectors, then w^t can be rewritten as:

$$w^t = w^{t-1} - \frac{\eta_0}{\sqrt{\sum_{i=1}^{t}(g^i)^2}} g^t$$