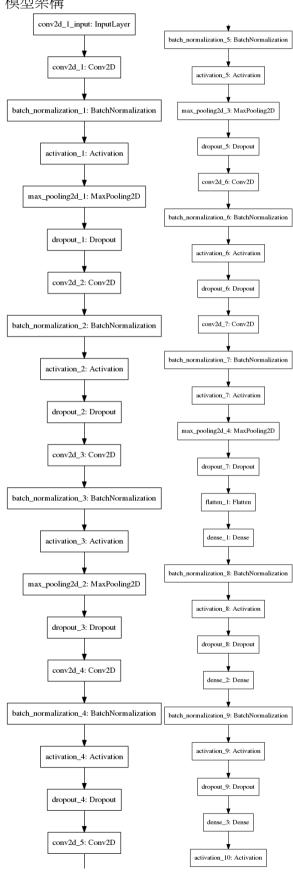
1. (1%) 請說明你實作的 CNN model, 其模型架構、訓練過程和準確率為何? (Collaborators:)

答:

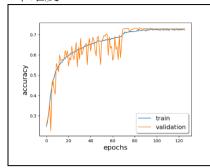




## ii. 訓練過程:

實際上用了 3 個 model 做 voting,分別是上圖的 model (這個有做 semi-supervised learning) 和 TA 的 sample model 以及 xception,validation set 則是採用 training data 的 10% 做 early stopping,另外有用 callback 在訓練中當 val\_loss 不再下降時減少 learning rate。

## iii. 準確度



此圖是 self learning 的準確度,因有加入 batch normalization ,準確度會跳動不定,所以加入 keras.callback 的 ReduceLROnPlateau,以調整 learning rate,約在 epoch 第70 幾,learning rate 做了第一次的調整,於是 learning rate 開始穩定,最後 validation set 的準確度約落在73%。但因為這是 semi-supervised learning,所以實際準確度較 kaggle 低。

2. (1%) 承上題,請用與上述 CNN 接近的參數量,實做簡單的 DNN model。其模型架構、訓練過程和準確率為何?試與上題結果做比較,並說明你觀察到了什麼?

(Collaborators: )

答:

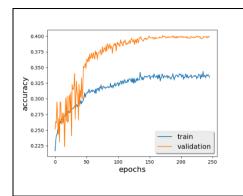
## i. 模型架構

Layer (type)	Output	Shape	Param #
flatten 1 (Flatten)	(None,	2304)	0
dense 1 (Dense)	(None,	9261)	21346605
batch normalization 1 (Batch	(None,	9261)	37044
activation 1 (Activation)	(None,	9261)	
dropout 1 (Dropout)	(None,	9261)	
dense 2 (Dense)	(None,	4608)	42679296
batch normalization 2 (Batch	(None,	4608)	18432
activation 2 (Activation)	(None,	4608)	
dropout 2 (Dropout)	(None,	4608)	
dense 3 (Dense)	(None,	1024)	4719616
batch normalization 3 (Batch	(None,	1024)	4096
activation 3 (Activation)	(None,	1024)	
dropout 3 (Dropout)	(None,	1024)	
dense 4 (Dense)	(None,	1024)	1049600
batch normalization 4 (Batch	(None,	1024)	4096
activation 4 (Activation)	(None,	1024)	
dropout 4 (Dropout)	(None,	1024)	
dense 5 (Dense)	(None,		
activation 5 (Activation)	(None,		
Total params: 69,865,960 Trainable params: 69,834,126 Non-trainable params: 31,834			

#### ii. 訓練過程:

和 CNN 類似,validation set 同樣是 10% 做 early stopping,一樣在訓在訓練中當 val\_loss 不再下降時減少 learning rate。

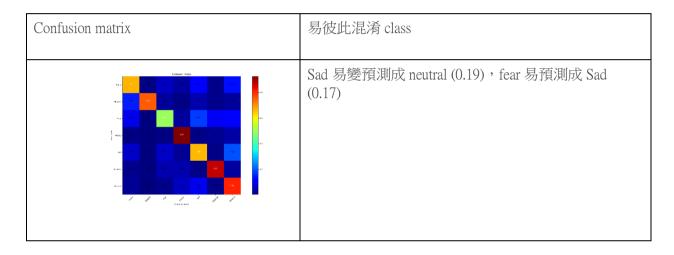
### iii. 準確度



很明顯準確度遠不及 CNN,即使在 validation 的準確度看似不錯,但是實際上在 kaggle 的準確度只有33%。

3. (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: )

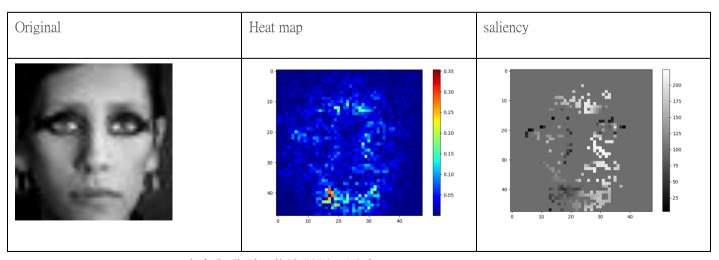
答:



4. (1%) 從(1)(2)可以發現,使用 CNN 的確有些好處,試繪出其 saliency maps,觀察模型在做 classification 時,是 focus 在圖片的哪些部份? (Collaborators: 葉韋辰、黃禹程)

答:

i. 圖



ii. Focus: 大多為嘴型,些許眼眶、眉毛

5. (1%) 承(1)(2),利用上課所提到的 gradient ascent 方法,觀察特定層的 filter 最容易被哪種圖片 activate。 (Collaborators: 葉韋辰、黃禹程) 答:

i. 圖片

Layer 1  April 1 April						
Layer 1	Layer 0		Output of layer0 (Given	image0)	30	
Layer 1					and the same of th	
Layer 1		7/2	7	19	30	
Layer 1			<b>T</b>	9	37	
Layer I			<b>3</b>	3/6	1	
Layer 1			三.	3	= 0	
Layer 1		The state of the s		1	1	
Layer 1    Copyed of type 1 Chien imaged   Copyed of type 1 Ch				9	7	
Layer 1		7			3,	
Layer 1				To.	1	
Layer 1				1	3)	
Layer 1    Corput of layer (Given Imager)   Corput of layer (Given				=		
Layer 1  Layer 1  Cotate of layer (Given image)  Cotate of lay		<u> </u>	and the second	3		
Layer 1			1	=		
Layer 1  Cotput of layer 1 (Given imaged)  Cotput of layer 1 (Given imaged			4			
		food	(====	<b>-</b>	200	
			∃r <sub>6</sub>	. Fa	100	
	Layer 1					
	Layer 1	į.	Output of layer1 (Given	image0)	1.	
	Layer 1	<b>3</b>	Output of layer1 (Given	image0)	<b>3</b> .	
	Layer 1	7. 14. 13.	Output of layer1 (Given	image0)	7. 3. 5.	
	Layer 1		Output of layer1 (Given	imageO)	(7) (7) (8)	
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	imageO)		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		
	Layer 1		Output of layer1 (Given	image()		

# ii. 觀察

第 0 層和第 1 層都是 64 個 filter,這次對 image 0 作圖可發現 layer0 大多還是穩裡表現,但 到 layer 1 的時候幾乎就已經是對眉毛和嘴型,有特別的過濾