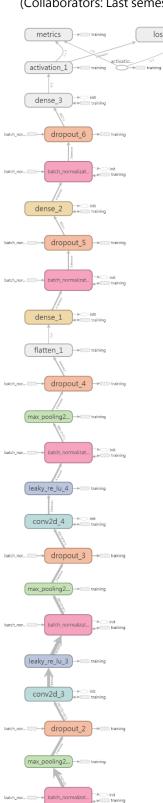
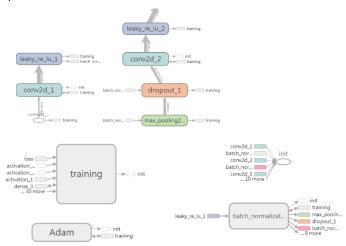
## ML2017FALL HW3 Report

學號:B04902090 系級: 資工三 姓名:施長元

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

(Collaborators: Last semester - windqaq 宋子維、hortune 陳佳佑、This: b04902089 林政豪)





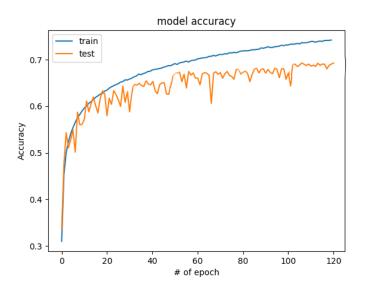
首先是資料的標準化,我將所有像素的資料都除以 255。

之後總共疊了 4 層 CNN Layer·Filter 數量分別是 50, 150, 500, 500·中間 Dropout 0.25, 0.3, 0.3, 0.4。每層都做 kernel\_size 為 (3,3) 的 Conv2D·activation 使用 LeakyReLU·並做 Batch Normalization、以及 pool\_size 為 (2,2)的 MaxPooling2D。

壓平之後,疊了 2 層 Dense 分別為 518, 259、activation 為 LeakyReLU 的 Layer,並也 做了 Batch Normalization 且 Dropout 皆為 0.5。最後用 Dense 為 7、activation 為 softmax 的 Layer 輸出。

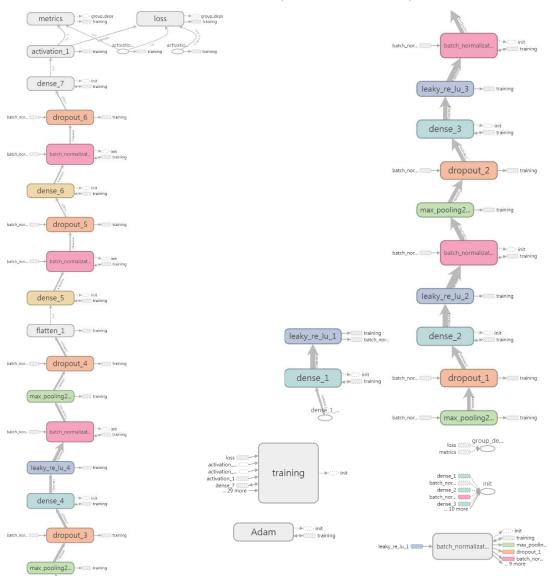
Loss 使用 categorical\_crossentropy · optimizer 使用 adam · 並且有加入 Keras 中的 preprocessing.image · 使 Train data 加入旋轉 10 度內、水平翻轉、上下左右移動 0.2、放大縮小 0.8~1.2 倍、修剪範圍 0.2。

整體 Train 出來的結果 Public 是 0.70242 的準確率。Kaggle 上的最佳 Public: 0.70771 是 使用五個 model: 0.68403、0.68654、 0.68737、0.69322、0.70242 做投票,每個 Model 依據自己的準確率擁有不同權重。

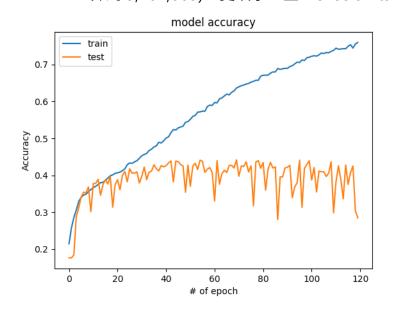


左圖為訓練過程中的準確率圖。 Validation data 為 train data 的後 20%。可以見到,Train data 的準確率不斷上升的同 時,Validation data 在 20 多的 epoch 時就趨緩了,但最終 kaggle 上的準確率卻有 0.7 以 上,推測可能是 Public overfit,或是訓練結果接續 HW1,又再次得偏向 Public 切 出來的 data 了。

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?(Collaborators: None)



Tensorboard 畫出來的圖長得幾乎一樣,因為實作 DNN 的作法跟 CNN 幾乎一模一樣,只是把 Conv2D 改為 Dense 層,其他包括 BatchNormalization、LeakyReLU、MaxPooling2D、Dropout,幾乎該做的都做了。另外,為了將 parameters 弄得一樣 (CNN: 3,398,617,所以把 DNN 弄到 3,402,663),更改了一些 Dense units 的數目。

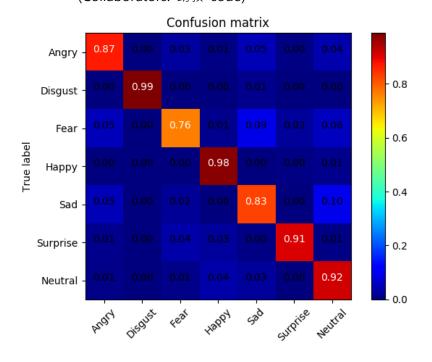


左圖是 DNN 在訓練過程的準確率,不意外的有夠爛;在 20~30 個 epoch 之後就毫不猶豫的 overfit 了。相較 CNN 是差了非常多。CNN 果然是圖片辨識的翹楚。

CNN 跟 DNN 之間的比較,在第五題會順便討論,敬請期待!

## 3. (1%) 觀察答錯的圖片中,哪些 class 彼此間容易用混?

(Collaborators: 助教 code)



左圖是混淆矩陣,政體的結果是不錯的。比較容易混淆的(比較亮的藍色)有兩組,分別是:

Neutral <->Sad

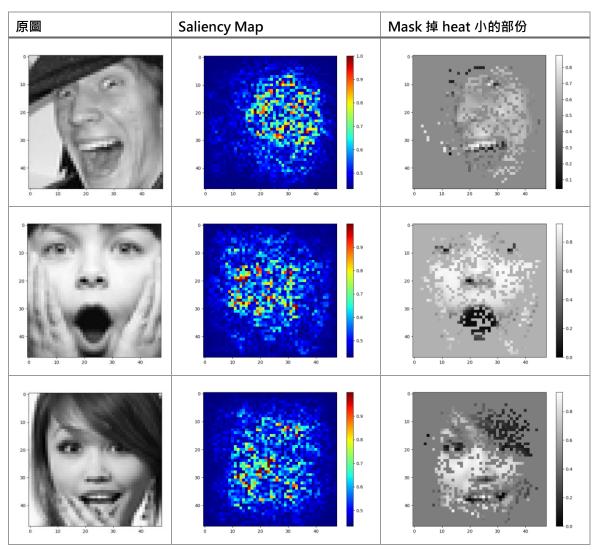
Fear <-> Sad

這其實不難體會,面無表情的人我們常常會覺得他有點鬱鬱寡歡,反之亦然,因此Model會誤判實在是情有可原。

至於害怕跟傷心,個人稍 微使用自己的面部實驗了一 下,傷心的時候,嘴角稍微 再往下一些、眼睛稍微瞪大

一些,就會變成害怕、驚恐的表情。這兩個表情的分界線,本就有些模糊,可能有些測資本就模稜兩可造成 Model 誤判,這亦是情有可原的情況之一。

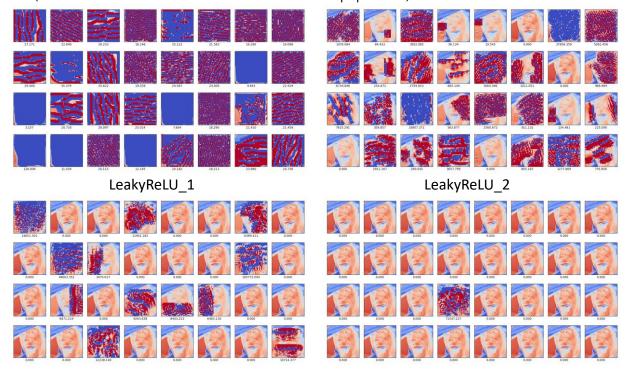
4. (1%) 從(1)(2)可以發現,使用 CNN 的確有些好處,試繪出其 saliency maps,觀察模型在做 classification 時,是 focus 在圖片的哪些部份? (Collaborators: 助教 code)



從 Mask 之後的圖片來看,Model 切出來的部分大多是五官,但也有第一張圖誤切一隻眼睛、第三張圖把頭髮也列入的情況發生。但整體仍然有算是正確的切出眼鼻口來判斷表情。但看 Saliency Map,紅色的地方其實出現得有些奇怪,比如說第一張圖幾乎沒用到嘴巴的部分;第二章圖只注重鼻子的部分等等。也許鼻子也能表現出表情的某些特徵吧?看在 70%的準確率,我就暫時相信這個 Model 了。

## 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。

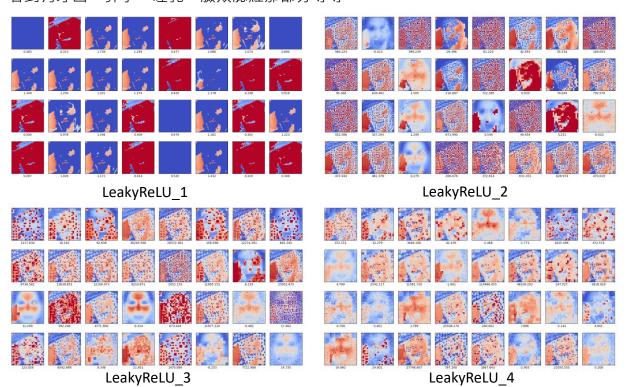
(Collaborators: 助教 code、Last semester - windqaq 宋子維)



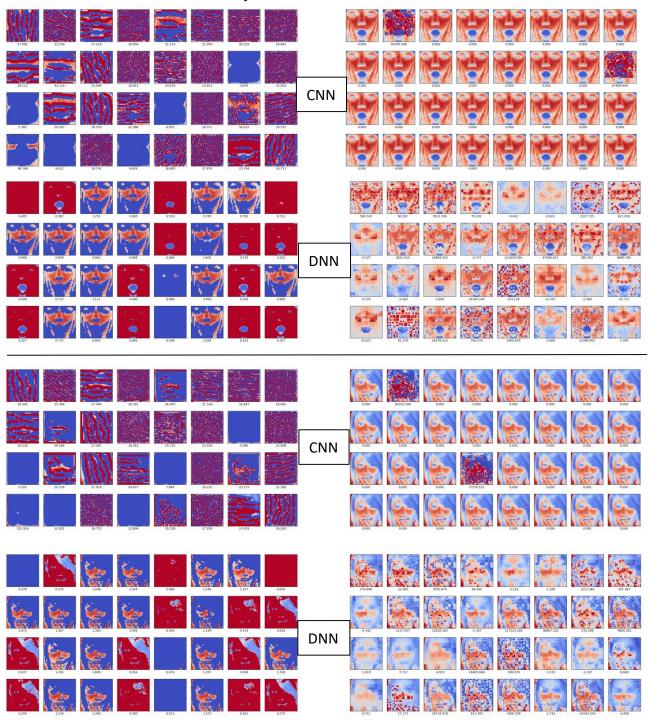
LeakyReLU\_3

LeakyReLU 4

總共有 4 層的 CNN,可以見得,第一層是像是紋理的可能只是一些面部陰影、或是背景的激活。第二層開始出現臉部的特徵,有的 filter 抓到了五官,有的卻往頭髮抓等等,到第四層才穩定的抓到五官的特徵。偏紅的代表激活值高,可以看到有牙齒、鼻子、瞳孔、臉頰腮紅那部分等等。



上圖則是同一張圖在 DNN 的輸出,可以發現, DNN 根本沒有抓到正確的五官,整體還說是有抓到面部的,但總往奇怪的地方抓,所以 DNN 表現遠低於 CNN 是合理的。以下就以 LeakyReLU\_1 及 4 作為代表層:



CNN 可以很明確的看到不同層的不同,DNN 則感覺就只是在面部上尋找特徵 (卻又抓錯)。CNN 在最後一張圖上也有出錯,他把手也抓進去了。