學號:B04901117 系級:電機三 姓名:毛弘仁

(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?

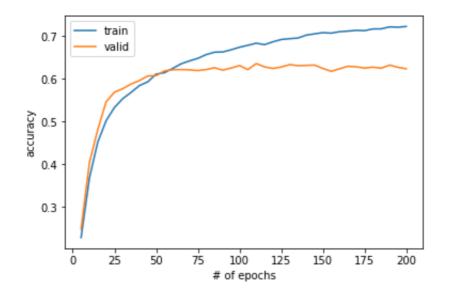
(Collaborators: 無)

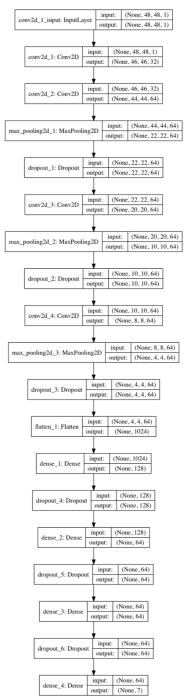
答:

右圖為我過 baseline 的 CNN model,用了三次「convolution 後 max-pooling 再 dropout」,並且再搭上四層的 feedforward network。

使用 Keras 的 default Adadelta,train 了 200 個 epoch,batch size = 128,最後達到 0.62 的 validation accuracy,0.61 的 Kaggle public set accuracy。

其實從圖表中可以發現,大約在 50 epochs 的時候,validation accuracy 就沒有什麼大變化了,或許不需要 train 到 200 epochs。不過比較大的問題可能是 overfitting:當 validation accuracy 停滯後,training accuracy 仍然穩定增加,或許可以再將dropout 比例調高一點試試看。





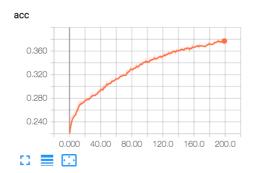
(1%) 承上題,請用與上述 CNN 接近的參數量,實做簡單的 DNN model。其模型架構、訓練過程和準確率為何?試與上題結果做比較,並說明你觀察到了什麼?

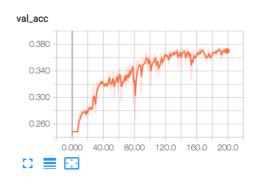
(Collaborators: 無)

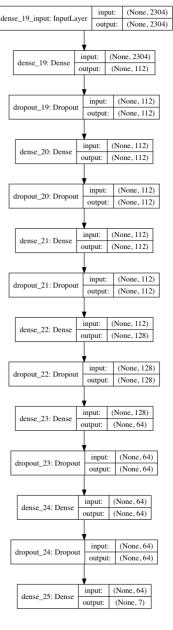
答:

原本的 CNN model 共有 309,664 個參數,我根據這個數字,設計出總參數量 310,208 的 feedforward network: 把 max-pooling 和 convolution 換成 size = 112 的 feedforward layer(共三層),並且刻意保留 CNN model 最後面的(128→64→64→7) feedforward network。如此一來,總參數量差不多、dropout 次數及比例不變,使用的 batch size、optimizer、epochs 也相同。

過了 120 epochs,validation error 維持在 0.37,沒有什麼大變動,需要花的 epoch 數比 CNN 多、noise 比 CNN network 大、最後得到的 accuracy 低很多。不過 train 起來的速度快很多!



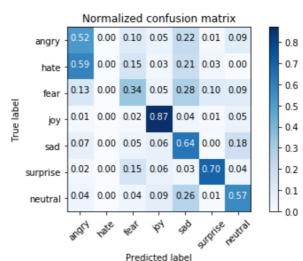




(1%) 觀察答錯的圖片中,哪些 class 彼此間容易用混?[繪出 confusion matrix 分析] (Collaborators: 無)

答:

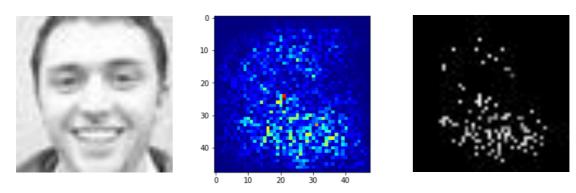
右圖為 validation set(10% training data)的 confusion matrix。從中可以看出 angry 和 hate 是容易混淆的,以日常經驗看,好像不無合理。其實,我的 model 完全無 法判斷 hate 的圖片! fear 的準確率也很低,只有 0.34。準確率最高的則是 joy,到達 0.87,不能再高了。



(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: 無)

答:

以一張 joy 為例,model 是 focus 在嘴巴、鼻子、還有一部分眉毛上。



另外也有在其他圖片上做測試,大部分 focus 的點都是在嘴巴、眼睛周遭。

(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。

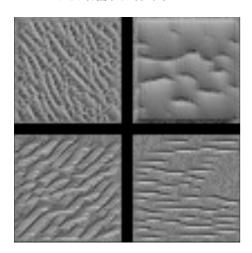
(Collaborators: 無)

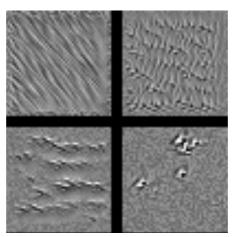
答:

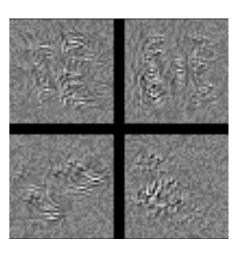
Part 1: Gradient Ascent

下三圖分別是能夠使 conv2d_2、conv2d_3、conv2d_4 中的 filters(每個 layer 各有 64 個 size=(3,3) 的 filters,這裡礙於版面限制只顯示 4 個),有最大 mean 的圖片。 Input image 皆以 random noise 去 initialize。這三層 convolution layers 中,認出 noisy pattern(無法明確看出圖案)的 filter 比例分別為 0.375、0.75、0.3125,我試著將 gradient update 的 step size 調低,但這對於降低"noisy filter"的比例並沒有幫助。經過分析,output mean > 0 的 filter 認出的 pattern 可能是 noise,但 output mean == 0 的 filter 卻有看得清楚的 pattern,因此 output mean 和 filter 是否"noisy"似乎沒有關係。

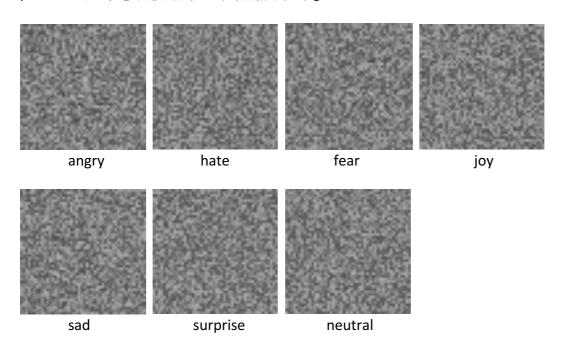
從能夠認出 pattern 的 filter 可看出,越高層的 convolution layer 能認出越複雜、越有層次的圖案。



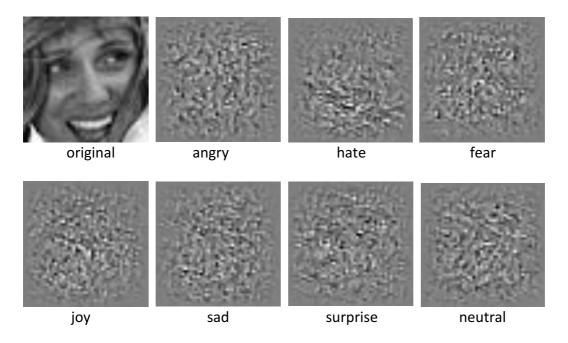




底下的圖是拿最後 softmax layer 做 gradient ascent on input 的結果,其中只有 angry、surprise、neutral 有辦法讓 target output 變成 1,這個結果是讓我有些驚訝,因 為 confusion matrix 在 joy 上準確度最高、angry 準確度最低。圖片當中無法看出明確的 pattern,這呼應宏毅老師上課中所展示的 gradient ascent on MNIST。



底下的圖一樣是做 gradient ascent,不同的是 initial input 不是 noise,而是一張 dataset 裡的圖。好像看得出有些圖案的樣子,但仍然不清楚。



Part 2: Filter Output

下圖是 conv2d_2、conv2d_4 中的部分 filter output, 我將每個 pixel 值加上 100,增加亮度以利觀察。

從結果可看出,低層的 convolution layer 會針對整張圖做比較細的觀察,像是斜線、白色區域、輪廓等等,而越往上層,convolution layer 看到的 feature 越多且越「融合」,因此從 filter output 僅能看出類似 attention 的 map,難以看出原來的 input image 長怎樣,而且因為 max-pooling 的關係,filter output 的維度會越來越小。

