

(一)專案簡介

本次期末專案我選擇進行 human activity recognition using smartphone data set，其選用的 sensor 為 samsungGalaxy s2 並且進行 12 個種類的分析
採樣頻率為 50(hz)

(二)技術簡介

在此我主要選擇 CNN 之模型設計架構進行分析分類並最終達到 96.3%之準確率我的設計模型如圖 1，而我會在之後解釋為甚麼我的層數及參數要這樣調配，其中有幾個重要的參數 X, Y, Z, W 。

X 為每個遮罩的寬度， Y 為此架構的深度， Z 是當經過 CONVOLUTION 層之後連接的全連接層神經元數， Z 則是 DROPOUT 的參數

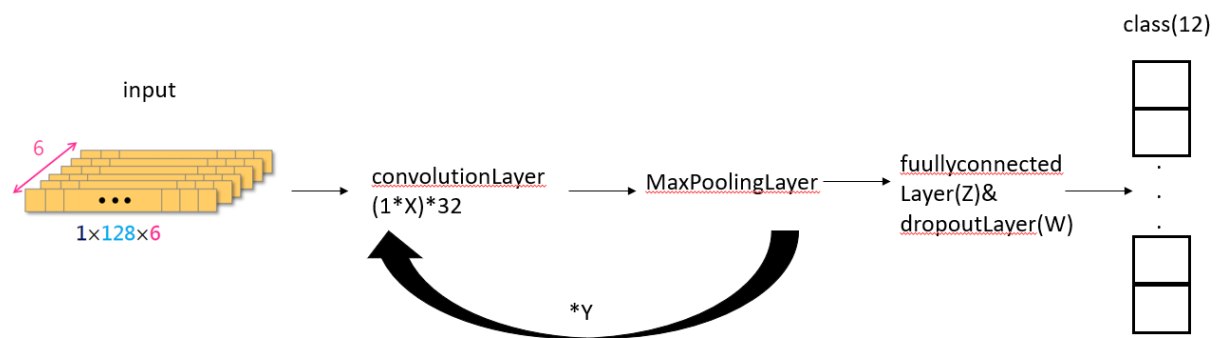


圖 1.設計架構

(三)CNN 設計概念

首先為了選出合適的 fullyconnectedLayer 之層數對如圖 1 中之 z 參數，我做了兩組對比如圖 2，為了快速進行比較我選用 epoch=10 進行對比，filtersize 選擇 $1*6$ ，並且將 y 參數定在 2 層，可以看出當 $z=256$ (fullyconnected layer)時，準確率還是會高一點

Epoch=10	Filter =(1*6)*2	Dropout=0.2	Shuffle=every
Fullyconnected(z)	256	32	64
Accuracy	92.33%	91.81%	89.63%
Loss	0.2017	0.2505	0.2832
time	3:21	3:21	3:21

為了選擇合適的 y 值我們將由實驗 1 的結果將 z 定在 256，並調整 y 值為 2,3,4 進行實驗

Epotch=10	Filter $=(1*6)*3$	Dropout=0.2	Fullyconnect(z)
Y=	4	3	2
Acurancuy	92.1%	92.78%	92.33%
Loss	0.2411	0.2003	0.2017
time	4:31	4:24	3:21

總體來說，當 $Y=3$ 時會有較好的準確率且花費時間也比較適中，但因為 Epoch 只設在 10，因此導致 $Y=4$ 的效果不如預期，但我們在此先選用 $Y=3$ 進行後續的實驗

接下來為了選擇合適的 dropoutLayer 參數，我分別進行了幾個實驗

	Filter $=(1*6)*3$	FullyconnectedLayer=256		Shuffle=every
DropoutLayer	0.2	0.8	0.2	0.8
Epotch	20	20	10	10
Acurancuy	93.64%	93.76%	92.33%	91.92%
Loss	0.1929	0.1711	0.2017	0.2234
time	7:23	7:27	3:21	3:27

可以看出在 epoch=10 時 dropout 設定在 0.2 會比 0.8 來的好，但當 epoch 上升時，則呈現黃金交叉，dropout=0.8 不論是 accuracy 和 loss 都比 dropout=0.2 時更好，我認為這是高的 dropout 增加了 CNN 之強健性，而且我們的 fullyconnectedlayer 比 outputlayer 大了許多因此我們可以設定較高的 dropoutlayer。

再來更重要的是選出 mask 的 X 值，而我們選擇將 X 設為 3 與設為 6 進行對比

Fullyconnect=256	Filter $=(1*6)*3$		Dropout $=0.8$		Shuffle $=every$	
Epotch=	60	50	40	30	20	10
Acurancuy	94.73%	94.16%	93.99%	93.18%	93.47%	92.15%
Loss	0.1476	0.1572	0.1628	0.1768	0.1790	0.2218

Fullyconnect=256	Filter $=(1*3)*3$		Dropout $=0.8$		Shuffle $=every$	最高 95.36%
Epotch=	60	50	40	30	20	10
Acurancuy	94.9%	94.73%	94.33%	93.93%	92.44%	87.74%
Loss	0.1396	0.1447	0.1655	0.1701	0.2034	0.3323

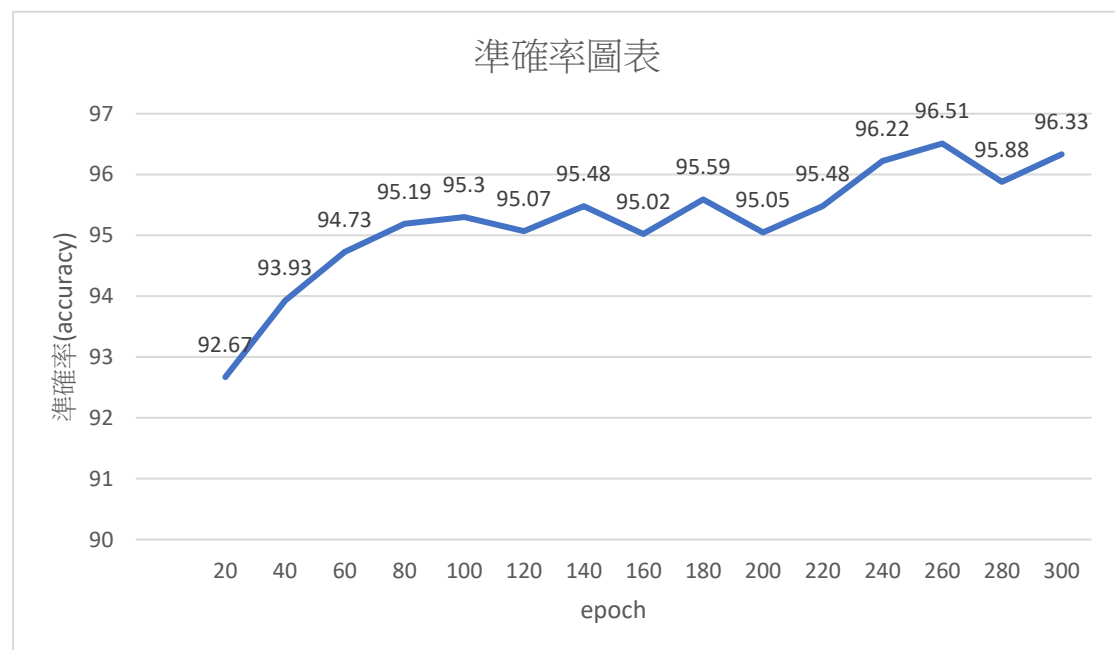
可以看出雖然在 Epoch 比較小時 $X=6$ 的準確率比較高，可是當 Epoch 超過 30 之後， $X=3$ 的準確率都還是表現更好的那一方，因此我認為在更高的 Epoch 時 $X=3$ 的特徵擷取效果更好。

因為預設 $\text{stride}=1$ ，因此為確保 $\text{stride}=1$ 時效果是最好的，分別進行了其他測試

Fullyconnect=256	Filter = $(1*3)*3$		Dropout =0.8		Shuffle =every	最高 95.36%
Stride=	3	3	3	1	1	1
Epoch=	30	20	10	30	20	10
Accuracy	91.64%	89.69%	85.97%	93.93%	92.44%	87.74%
Loss	0.2432	0.286	0.4109	0.1701	0.2034	0.3323
time	12:40	8:45	4:44	11:45	7:51	3:59

可以看出 epoch 調大之後 $\text{stride}=1$ 的效果還是更好。

通過以上的測試我們選出本次期末 CNN 模型的各個關鍵數據將 $X,Y=3,Z=256$ (FullyconnectedLayer), $W=0.8$ (dropoutLayer)，將 epoch 設為 300，進行最後的訓練，而 option 的設置我以上課所使用的為主，並將優化器從 sgdm 換成 adam，可以看出最後我訓練的結果準確率到達 96.33%，但在第 260 個 epoch 時，已經訓練出更高的 network



(四)混淆矩陣與結論

最後可以看到我們訓練出來的最高準確率可達到 **96.3%**如下圖，而可以看出我們主要錯誤的是在第 4 個 class，單單一格就有 21 個判斷錯誤，可能是其特徵相像。

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	11	12
1	260 14.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
2	0 0.0%	243 13.9%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	3 0.2%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%
3	1 0.1%	0 0.0%	260 14.9%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%
4	0 0.0%	0 0.0%	0 0.0%	258 14.8%	9 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
5	0 0.0%	0 0.0%	0 0.0%	21 1.2%	287 16.4%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	298 17.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.2%	1 0.1%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 0.6%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 0.6%	0 0.0%	0 0.0%	0 0.0%	1 0.1%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 1.0%	0 0.0%	5 0.3%	0 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 0.6%	0 0.0%	2 0.1%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	4 0.2%	0 0.0%	14 0.8%	0 0.0%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 0.3%	0 0.0%	4 0.3%
Precision	100%	97.6%	98.9%	96.6%	92.9%	98.7%	90.9%	90.9%	77.3%	84.6%	73.7%	70.0%
Recall	0.0%	2.4%	1.1%	3.4%	7.1%	1.3%	9.1%	9.1%	22.7%	15.4%	26.3%	30.0%
Accuracy	99.6%	100%	99.6%	92.1%	96.6%	100%	66.7%	100%	73.9%	64.7%	60.9%	77.8%
Macro Avg	0.4%	0.0%	0.4%	7.9%	3.4%	0.0%	33.3%	0.0%	26.1%	35.3%	39.1%	22.2%
Weighted Avg	3.7%											

精確率 (Precision) = $tp/(tp+fp)$ ，即陽性的樣本中有幾個是預測正確的。
如上圖中之紅色框框區域分別代表每個種類的精確率。

召回率 (Recall) = $tp/(tp+fn)$ ，即事實為真的樣本中有幾個是預測正確的。
圖 2 中之藍色框框區域分別代表每個種類的召回率。

(五)心得

我目前是大學四年級控制組的學生，我覺得在大四下學期選修這堂課對於我未來研究所生涯有極大的幫助，在大學期間一直都想往人工智慧與機械本身專業方向進行學習，但機械系本身並沒有開設那麼多的程式實務課，因此有時候都會要去外系修課，而這學期系上研究所開設這堂課對我來說感到非常高興，本課程一開始用一些簡單的數學推倒幫助我們了解其中的原理，之後再慢慢通過作業實作，對於 Matlab 語法與模擬有進一步的認識，並把深度學習的原理透過程式實作，這是在大學部比較少接觸到的，並且最後期末報告設計的也很有巧思，需要蠻多次的測試並思考最符合的策略，並且方法也不是唯一，雖然最後因為硬體原因以及 Matlab 版本不夠新且有些功能不能進行測試，但也讓我學到很多東西，未來也可以應用在很多領域上，其中機器人視覺是我最感興趣的部分，電腦視覺中也有許多方法使用到深度學習這個架構，而且也有很多更深入的數學和方法可以鑽研，希望我未來能在研究所生涯透過大學所學研究順利。