

1. 請比較你本次作業的架構、參數量、結果和原HW3作業架構、參數量、結果做比較。(1%)

本次作業的mobile net將原本的一個convolution layer拆成兩個參數總和較小的Depthwise Separable layer。我的mobile layer有八層convolutional layer，filter數目分別為32, 32, 64, 64, 64, 64, 128, 128, filter大小皆為3，其中前兩層為原本的convolution layer，剩下的convolution layer皆使用Depthwise Separable layer，activation function為Relu。第2、4、6、7、8層convolution layer後面會接上maxpooling來降低圖片大小，最後通過128*7的fully connected layer就直接輸出。在儲存參數前我會以model.half()將參數改以32bit floating point儲存來做weight quantization，再進一步降低大小。

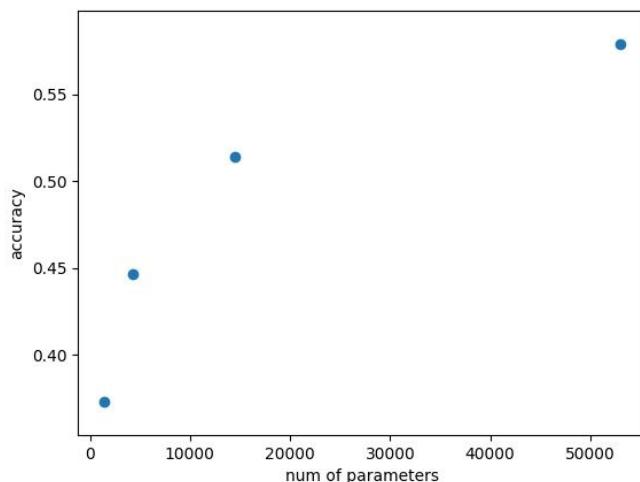
HW3的模型架構為十層convolution layer與兩層fully connected layer。十層convolution layer的filter數目分別為64, 64, 128, 128, 256, 256, 512, 512，而除了第二層convolutional layer的filter大小為5之外其餘皆為3，activation function為Relu。每兩層convolution layer後面會接maxPooling。兩層fully connected layer分別為512*1000、1000*7。

本次實作的mobile net參數為53056，而作業三的參數量為9985432，差距甚大，確實有大量縮減參數量。

Mobile net在kaggle上的成績為public 0.64976, private 0.63611，而HW3 model在kaggle上的成績為public 0.68960, private 0.68877，HW3 model明顯比Mobile net好。

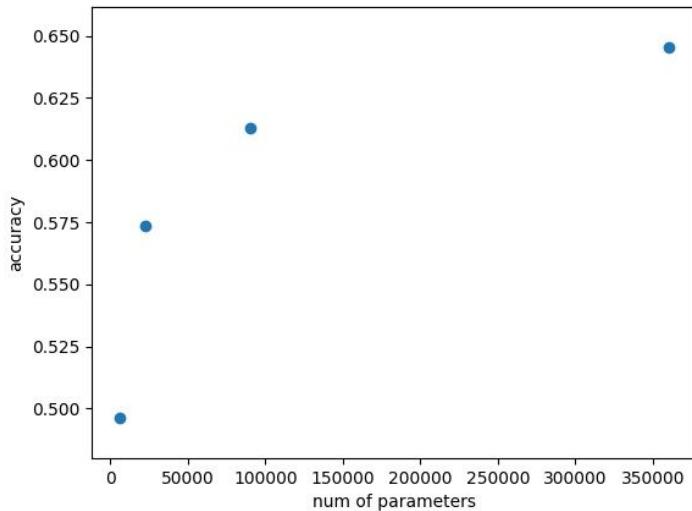
2. 請使用MobileNet的架構，畫出參數量-acc的散布圖（橫軸為參數量，縱軸為accuracy，且至少3個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用train到最好沒關係。）(1%)

下圖為參數量-acc的散布圖，我用training時dataset的1/10切作validation set作為acc依據，我將每個layer的filter數量除以1,2,4,8得到四個不同參數量的model，每個model都train 70個epoch。由圖可以發現參數與準確度為正相關，但是隨著參數量變多acc的成長幅度會慢慢趨緩。

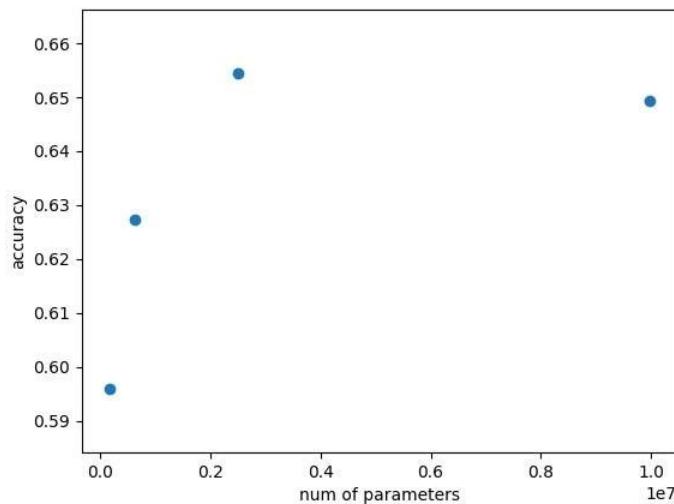


3. 請使用一般CNN的架構，畫出參數量-acc的散布圖（橫軸為參數量，縱軸為accuracy，且至少3個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用train到最好沒關係。）(1%)

下圖為將第二題的Depthwise Separable layer都改成一般Convolution layer的參數量-acc的散布圖，每個model都train 70個epoch。由圖可以發現與第二題大致一樣的現象：參數與準確度為正相關，但是隨著參數量變多acc的成長幅度會慢慢趨緩，只是準確度比第二題高。



另外，下圖為以原本HW3的model將每個layer的filter數量除以1,2,4,8得到四個不同參數量的model做出的參數量-acc的散布圖，每個model都train 70個epoch。雖然大部分現象與上面都相同，但能夠發現最右點的validation accuracy下降了，猜測是因為參數過多開始容易發生overfitting。



4. 請你比較題2和題3的結果，並請針對當參數量相當少的時候，如果兩者參數量相當，兩者的差異，以及你認為為什麼會造成這個原因。(2%)

大致比較第二題第三個點與第三題第一個點的對應準確度，可以發現在相近的參數量之下，mobile net的表現會相對較好。我猜測原因是因為在同樣的參數數量下，第二題簡化了convolutional layer filter所需的參數量，因此每一層 convolutional layer能夠比第三題有更多的filter數目。雖然減少每個filter所需要的參數可能會少許降低每個filter的能力，但是大量的filter數目不只彌補還增加整個layer 的學習能力。因此相對於第三題model，mobile net將weight放在更對的地方，造成效能更好。