

# ML2019SPRING HW8

## B06902074 資工二 柯宏穎

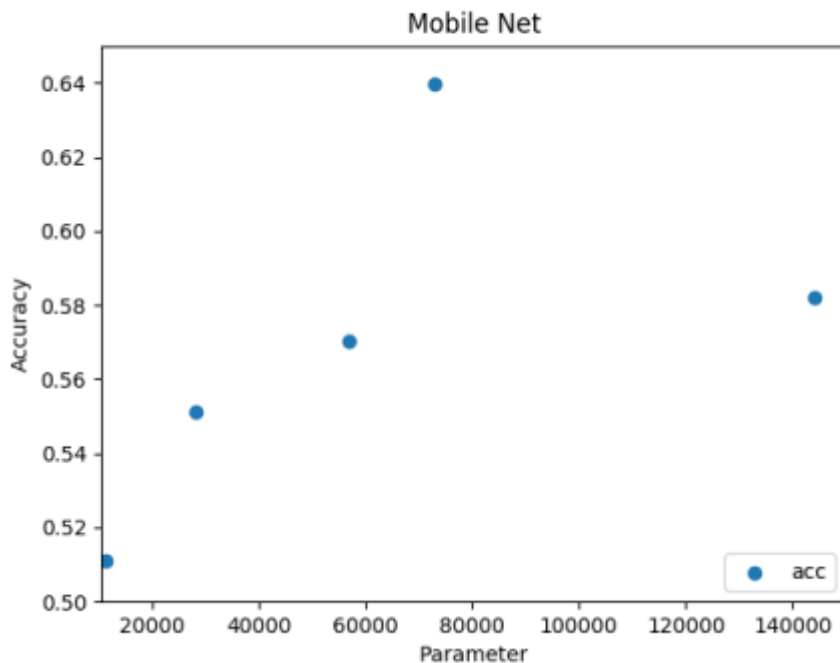
請比較你本次作業的架構，參數量、結果和原HW3作業架構、參數量、結果做比較。

與作業三相比，這次的參數量明顯降低了許多。作業三在架設過程中，完全沒有在考慮model的大小問題，只要能增加performance，架多深，跑多久都讓他跑完。主要架構為兩層 $Cov2D$ 一次 $MaxPooling2D$ ，以 $PReLU$ 當作activation，並做 $BatchNormalization$ ，以此循環三次， $filters$ 數量每組為32, 64, 128，最後接了三層的 $Dense$ ， $units$ 數量來到512與2048，因此參數量非常的龐大，800萬左右個參數。單個模型的準確率大約0.63，其實也沒有到特別的高，最後大多使用 $ensemble$ 來達標。

這次則使用 $DepthwiseConv2D$ 來做，以三層 $DepthwiseConv2D + Conv2D$ 一次 $MaxPooling2D$ 為一組，以 $LeakyReLU$ 當作activation，並一樣地使用 $BatchNormalization$ ， $filters$ 數量每組分別為32, 48, 64，最後再接到一層 $Dense$ 。此種參數量就變低許多，參數量約5萬5左右。為了過 $strong$ ，我另外將精度調整為 $float32$ ，僅僅16kB的 $weights$ ，準確率約0.64307其實是要比之前的來的好。

在這次的training中，我發現其實過深也不是好事，在做下面兩題時，參數量過大反而得到很糟糕的performance，降下來後才有較為正常的預測，適當的調整才能有更好的結果。

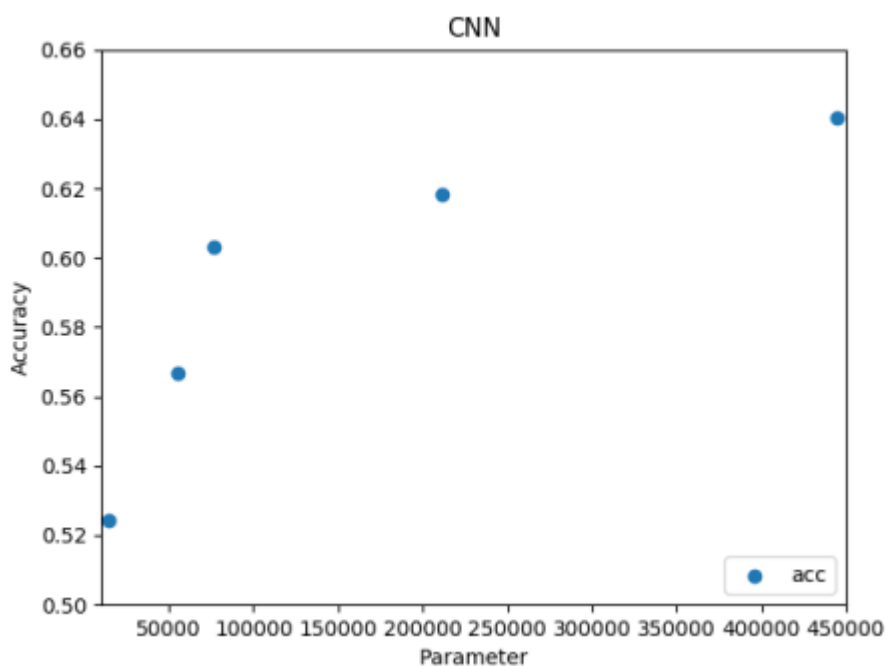
請使用MobileNet的架構，畫出參數量-acc的散布圖（橫軸為參數量，縱軸為accuracy，且至少3個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用train到最好沒關係。）



參數量分別為11255, 28295, 56967, 144191.

我皆只改動 $filters$ 的數量，其餘架構無多做改變。

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。）



參數量分別為14473, 54823, 76725, 211015, 444679.

我皆只改動*filters*的數量，其餘架構無多做改變。

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。

由前兩題可發現，*CNN*的參數量相對多非常多，相對的performance也較好一點。不過在參數量相當時（皆為一萬多與五萬多）時，其實差異不會到太大。在參數少時，相對的*channel*等數量並不會太多，*Mobile Net*相對的影響也就不太大了。不過*Mobile Net*還是讓我們能使用的*filters*數量提升很多，讓我們更有彈性地去調整參數，找到更好，更適合的*model*與參數。