

# ML HW7 report

b07902040 資工二 吳承軒

## 1. 請從 Network Pruning/Quantization/Knowledge Distillation/Low Rank Approximation/Design Architecture 選擇兩者實做並詳述你的方法，將同一個大 model 壓縮至接近相同的參數量，並紀錄其 accuracy。(2%)

big model使用經過Knowledge Distillation的resnet18的pretrain model，student model架構如下：

```
1 base = 16
2 multiplier = [1, 2, 4, 8, 8, 8, 16, 16, 16]
3 bandwidth = [base * m for m in multiplier]
4 cnn=
5     第一層：
6         Conv2d(3, bandwidth[0], 3, 1, 1)
7         BatchNorm2d(bandwidth[0])
8         ReLU6()
9         MaxPool2d(2, 2, 0)
10    第二~四層：
11        Conv2d(bandwidth[i], bandwidth[i], 3, 1, 1, groups=bandwidth[i])
12        BatchNorm2d(bandwidth[i])
13        ReLU6()
14        Conv2d(bandwidth[i], bandwidth[i+1], 1)
15        MaxPool2d(2, 2, 0)
16    第五~八層：
17        Conv2d(bandwidth[i], bandwidth[i], 3, 1, 1, groups=bandwidth[i])
18        BatchNorm2d(bandwidth[i])
19        ReLU6()
20        Conv2d(bandwidth[i], bandwidth[i+1], 1)
21    最後做Global Average Pooling:
22        AdaptiveAvgPool2d((1, 1))
23 fc =
24    Linear(bandwidth[8], 11)
```

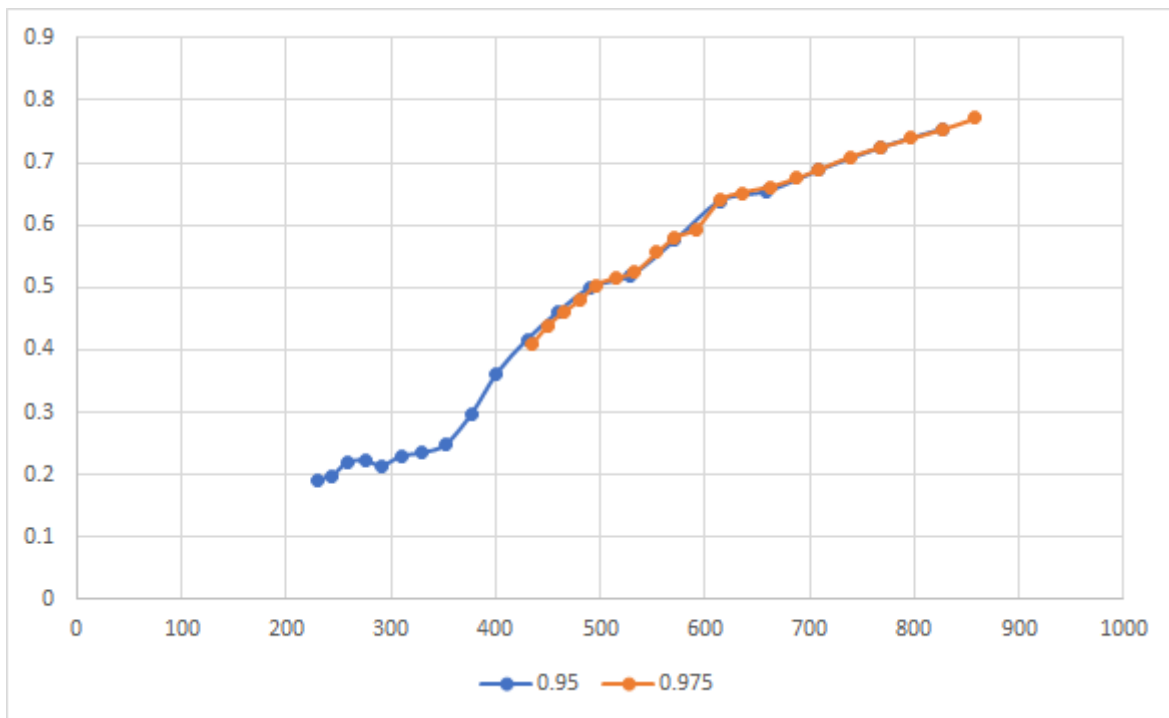
實作Pruning和Quantization：

Pruning:使用colab的code，設定reduce\_ratio為0.35，重複20次，存取val\_acc最好的那次。

Quantization:使用colab的code，直接將參數由32bit的浮點數壓到8bit的整數。

	big_model	Pruning	Quantization
model_size(B)	918038	228878	233234
val_acc	78.71%	18.43%	76.45%

3. 請使用兩種以上的 **pruning rate** 畫出 X 軸為參數量，Y 軸為 **validation accuracy** 的折線圖。你的圖上應會有兩條以上的折線。  
(2%)



pruning rate分別為0.95和0.975，X軸為model size，Y軸為在validation上的正確率。

圖中可以發現model size與正確率呈現高度正相關，且無論pruning rate是0.95還是0.975，在同樣的size下，正確率幾乎是一樣的。

4. 請嘗試比較以下 **validation accuracy**，並且模型大小要接近 **1MB: (2%)**

以下參數量使用torchsummary計算

a. 原始 CNN model (用一般的 Convolution Layer) 的 accuracy

參數量:224,011

```
1 base = 16
2 multiplier = [1, 2, 4, 8, 8]
3 bandwidth = [base * m for m in multiplier]
4 cnn=
5     第一層:
6         Conv2d(3, bandwidth[0], 3, 1, 1)
7         BatchNorm2d(bandwidth[0])
8         ReLU6()
9         MaxPool2d(2, 2, 0)
10    第二~四層:
11        Conv2d(bandwidth[i], bandwidth[i+1], 3, 1, 1)
12        BatchNorm2d(bandwidth[i+1])
13        ReLU6()
14        MaxPool2d(2, 2, 0)
15    第五層:
16        Conv2d(bandwidth[i], bandwidth[i+1], 3, 1, 1)
17        BatchNorm2d(bandwidth[i+1])
18        ReLU6()
```

```

19     最後做Global Average Pooling:
20         AdaptiveAvgPool2d((1, 1))
21     fc =
22         Linear(bandwidth[4], 11)

```

## b. 將 CNN model 的 Convolution Layer 換成總參數量接近的 Depthwise & Pointwise 後的 accuracy

參數量:247,179

```

1     base = 16
2     multiplier = [1, 2, 4, 8, 8, 8, 16, 16, 16]
3     bandwidth = [ base * m for m in multiplier]
4     cnn=
5         第一層:
6             Conv2d(3, bandwidth[0], 3, 1, 1)
7             BatchNorm2d(bandwidth[0])
8             ReLU6()
9             MaxPool2d(2, 2, 0)
10        第二~四層:
11            Conv2d(bandwidth[i], bandwidth[i], 3, 1, 1, groups=bandwidth[i])
12            BatchNorm2d(bandwidth[i])
13            ReLU6()
14            Conv2d(bandwidth[i], bandwidth[i+1], 1)
15            MaxPool2d(2, 2, 0)
16        第五~八層:
17            Conv2d(bandwidth[i], bandwidth[i], 3, 1, 1, groups=bandwidth[i])
18            BatchNorm2d(bandwidth[i])
19            ReLU6()
20            Conv2d(bandwidth[i], bandwidth[i+1], 1)
21        最後做Global Average Pooling:
22            AdaptiveAvgPool2d((1, 1))
23    fc =
24        Linear(bandwidth[8], 11)

```

## c. 將 CNN model 的 Convolution Layer 換成總參數量接近的 Group Convolution Layer (Group 數量自訂，但不要設為 1 或 in\_filters)

參數量:218,651

設定group為2，得到以下model：

```

1     base = 16
2     multiplier = [1, 2, 4, 8, 8, 8]
3     bandwidth = [ base * m for m in multiplier]
4     cnn=
5         第一層:
6             Conv2d(3, bandwidth[0], 3, 1, 1)
7             BatchNorm2d(bandwidth[0])
8             ReLU6()
9             MaxPool2d(2, 2, 0)
10        第二~四層:
11            Conv2d(bandwidth[i], bandwidth[i+1], 3, 1, 1, group=2)
12            BatchNorm2d(bandwidth[i+1])
13            ReLU6()
14            MaxPool2d(2, 2, 0)

```

```

15     第五~六層:
16         Conv2d(bandwidth[i], bandwidth[i+1], 3, 1, 1, group=2)
17         BatchNorm2d(bandwidth[i+1])
18         ReLU6()
19     最後做Global Average Pooling:
20         AdaptiveAvgPool2d((1, 1))
21     fc =
22         Linear(bandwidth[5], 11)

```

在迭代150次後，得到以下val\_acc：

	原始 CNN	Depthwise & Pointwise	Group Convolution
val_acc	0.6804	0.7603	0.7073

可以發現Depthwise & Pointwise的正確率最高，Group Convolution次之，原始 CNN最低，推測是因為Group Convolution可以用更少的參數達成和原始CNN一樣的效果，也意味當參數量相同時，Group Convolution可以更寬/更深，而DW&PW則是Group Convolution的極端，所以減少參數的效果更好。