學號：b05705011　系級：資管四　姓名：楊子霖

1. 請從 Network Pruning/Quantization/Knowledge Distillation/Low Rank Approximation 選擇兩個方法(並詳述)，將同一個大 model 壓縮至同等數量級，並討論其 accuracy 的變化。 (2%)

   大 model 為助教所提供的 student_custom_small.bin，架構為助教所提供的範例程式的 StudentNet (base=16)，儘管此 model 是經由 artitecture design 所設計的架構，但 model 大小仍大於 300000 bytes，大小為 1047430 bytes，因此這邊把它當大 model 來做實驗，在 validation 的 accuracy 為 0.8137。

   方法一：Network Pruning
   使用助教提供的範例程式，並且壓縮成 0.95、0.95^2 ... 0.95^5，以下圖為各個 model 分別在 validation 的 accuracy 以及各自的 model 大小 (單位 bytes)

```
rate 0.9500 epoch   0: train loss: 0.5005, acc 0.8657 valid loss: 1.1221, acc 0.7977
rate 0.9500 epoch   1: train loss: 0.4804, acc 0.8672 valid loss: 1.1258, acc 0.7980
rate 0.9500 epoch   2: train loss: 0.5047, acc 0.8624 valid loss: 1.0881, acc 0.8026
rate 0.9500 epoch   3: train loss: 0.4806, acc 0.8666 valid loss: 1.1302, acc 0.7988
rate 0.9500 epoch   4: train loss: 0.4755, acc 0.8704 valid loss: 1.1416, acc 0.7980
rate 0.9025 epoch   0: train loss: 0.5518, acc 0.8448 valid loss: 1.2178, acc 0.7787
rate 0.9025 epoch   1: train loss: 0.5764, acc 0.8445 valid loss: 1.1651, acc 0.7805
rate 0.9025 epoch   2: train loss: 0.6030, acc 0.8369 valid loss: 1.2207, acc 0.7773
rate 0.9025 epoch   3: train loss: 0.5837, acc 0.8411 valid loss: 1.1740, acc 0.7825
rate 0.9025 epoch   4: train loss: 0.6112, acc 0.8395 valid loss: 1.1937, acc 0.7816
rate 0.8574 epoch   0: train loss: 0.6773, acc 0.8157 valid loss: 1.2299, acc 0.7641
rate 0.8574 epoch   1: train loss: 0.6853, acc 0.8134 valid loss: 1.2007, acc 0.7665
rate 0.8574 epoch   2: train loss: 0.6885, acc 0.8115 valid loss: 1.2535, acc 0.7638
rate 0.8574 epoch   3: train loss: 0.7101, acc 0.8081 valid loss: 1.2231, acc 0.7685
rate 0.8574 epoch   4: train loss: 0.7052, acc 0.8051 valid loss: 1.2108, acc 0.7673
rate 0.8145 epoch   0: train loss: 0.8231, acc 0.7766 valid loss: 1.3034, acc 0.7362
rate 0.8145 epoch   1: train loss: 0.8118, acc 0.7814 valid loss: 1.2549, acc 0.7356
rate 0.8145 epoch   2: train loss: 0.8179, acc 0.7758 valid loss: 1.2616, acc 0.7411
rate 0.8145 epoch   3: train loss: 0.8187, acc 0.7756 valid loss: 1.3189, acc 0.7356
rate 0.8145 epoch   4: train loss: 0.7902, acc 0.7845 valid loss: 1.2561, acc 0.7397
rate 0.7738 epoch   0: train loss: 1.0186, acc 0.7319 valid loss: 1.3942, acc 0.7076
rate 0.7738 epoch   1: train loss: 1.0123, acc 0.7322 valid loss: 1.4754, acc 0.7038
rate 0.7738 epoch   2: train loss: 0.9879, acc 0.7359 valid loss: 1.4279, acc 0.7038
rate 0.7738 epoch   3: train loss: 1.0034, acc 0.7308 valid loss: 1.4188, acc 0.7058
rate 0.7738 epoch   4: train loss: 1.0280, acc 0.7298 valid loss: 1.3628, acc 0.7108
```

```
705204  五  11 14:30 custom_small_rate_0.77378093749999999.bin
759424  五  11 14:28 custom_small_rate_0.81450624999999999.bin
820780  五  11 14:26 custom_small_rate_0.85737499999999999.bin
891100  五  11 14:24 custom_small_rate_0.9025.bin
964300  五  11 14:22 custom_small_rate_0.95.bin
```

   方法二：Weight Quantization
   使用助教提供的範例程式，並將 32 bit tensor array 分別轉成 16 bit 以及 8 bit，而前者 model 在 validation 的 accuracy 仍為 0.813994，後者 model 則為 0.808746，下圖為二 model 的大小。

```
522958  五  11 14:50 16_bit_model.pkl
268471  五  11 14:50 8_bit_model.pkl
```

由此可見，weight quantization 除了壓縮成更小的 model 還可以維持在 0.81 左右的 accuracy，而 network pruning 則效率較差，壓縮成 0.95 accuracy 就掉到 0.8026，何況壓縮成和 8_bit_model.pkl 一樣的大小。

以下三題只需要選擇兩者即可，分數取最高的兩個。

2. [Knowledge Distillation] 請嘗試比較以下 validation accuracy (兩個 Teacher Net 由助教提供)以及 student 的總參數量以及架構，並嘗試解釋為甚麼有這樣的結果。你的 Student Net 的參數量必須要小於 Teacher Net 的參數量。(2%)
   x. Teacher net architecture and # of parameters: torchvision's ResNet18, with 11,182,155 parameters.
   y. Student net architecture and # of parameters: 這邊 model 架構為範例程式中的 StudentNet (base=16)，參數量為 256779，並且這邊我是自己重新 train 一遍，epoch 為 300、learning rate 為 0.001，那下圖為其訓練時在 validation 的 accuracy。

```
[296/300] 23.68 sec(s) Train Acc: 0.919015 Loss: 0.007388 | Val Acc: 0.752478 loss: 0.033249
[297/300] 23.64 sec(s) Train Acc: 0.930772 Loss: 0.006545 | Val Acc: 0.754810 loss: 0.033461
[298/300] 23.69 sec(s) Train Acc: 0.926819 Loss: 0.006969 | Val Acc: 0.774052 loss: 0.031657
[299/300] 23.73 sec(s) Train Acc: 0.931583 Loss: 0.006544 | Val Acc: 0.762682 loss: 0.033966
[300/300] 23.71 sec(s) Train Acc: 0.926313 Loss: 0.006904 | Val Acc: 0.760641 loss: 0.033586
```

   可發現此 model 的表現最高來到 0.774 左右。
   a. Teacher net (ResNet18) from scratch: 80.09%
   b. Teacher net (ResNet18) ImageNet pretrained & fine-tune: 88.41%
   c. Your student net from scratch: 77.4 %。
   d. Your student net KD from (a.):

```
epoch 236: train loss: 1.5147, acc 0.8947 valid loss: 2.0637, acc 0.7886
epoch 237: train loss: 1.5433, acc 0.8881 valid loss: 2.2611, acc 0.7889
epoch 238: train loss: 1.5520, acc 0.8917 valid loss: 2.1063, acc 0.7962
epoch 239: train loss: 1.5293, acc 0.8882 valid loss: 2.1212, acc 0.7860
epoch 240: train loss: 1.5369, acc 0.8867 valid loss: 2.2751, acc 0.7741
epoch 241: train loss: 1.5429, acc 0.8857 valid loss: 2.4439, acc 0.7746
epoch 242: train loss: 1.5133, acc 0.8949 valid loss: 1.9761, acc 0.8023
epoch 243: train loss: 1.5500, acc 0.8945 valid loss: 2.1145, acc 0.7901
epoch 244: train loss: 1.5443, acc 0.8911 valid loss: 2.1257, acc 0.7886
```

   epoch 設為 300，最高的 accuracy 為 80.23 %。
   e. Your student net KD from (b.):

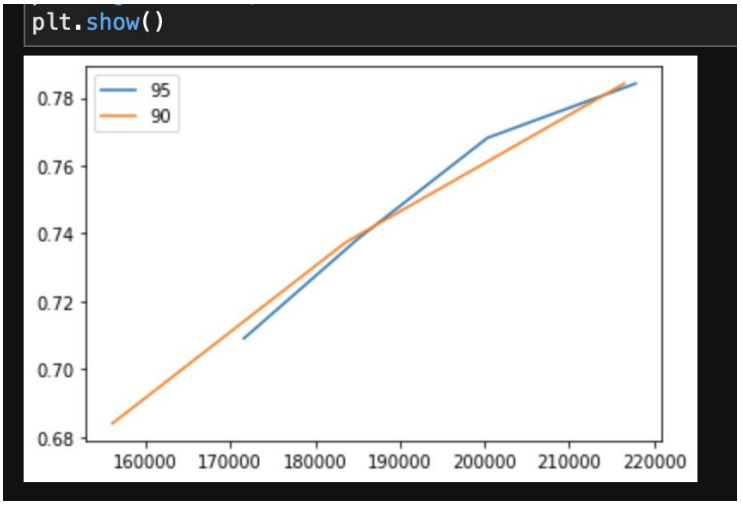
```
epoch 274: train loss: 2.8265, acc 0.9033 valid loss: 4.0207, acc 0.8085
epoch 275: train loss: 2.7699, acc 0.9054 valid loss: 4.2467, acc 0.8041
epoch 276: train loss: 2.7996, acc 0.8997 valid loss: 4.1058, acc 0.8090
epoch 277: train loss: 2.8420, acc 0.9046 valid loss: 3.9152, acc 0.8096
epoch 278: train loss: 2.7931, acc 0.9086 valid loss: 3.7606, acc 0.8224
epoch 279: train loss: 2.7765, acc 0.9018 valid loss: 3.9364, acc 0.8076
```

   epoch 設為 300，最高的 accuracy 為 82.24 %。

   d、e 的結果之所以大於 c 是因為它們每次學習的時候還可以學到哪些類型會比較像一點，而 c 只能學到一張圖片的正確答案。而 e 的結果比 d 好是因為 e 的學習對象的準確率遠高於 d 的學習對象。

3. [Network Pruning] 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。你的圖上應該會有兩條以上的折線。(2%)

   這邊 pruning rate 是抓 0.95、0.90，那這邊我只截取了二邊參數量差不多的部分，可以發現不同的 pruning rate 在 prune 到差不多的參數量時，其預測結果也不會相差太多。



4. [Low Rank Approx / Model Architecture] 請嘗試比較以下 validation accuracy，並且模型大小須接近 1 MB。 (2%)
   a. 原始 CNN model (用一般的 Convolution Layer) 的 accuracy
   b. 將 CNN model 的 Convolution Layer 換成參數量接近的 Depthwise & Pointwise 後的 accuracy
   c. 將 CNN model 的 Convolution Layer 換成參數量接近的 Group Convolution Layer (Group 數量自訂，但不要設為 1 或 in_filters)